LUNDS UNIVERSITET

Department of Industrial Management and Logistics
Faculty of Engineering
LTH Lund University

Exploring the use of machine learning to validate country of origin

Anna Klackenberg & Louise Rundqvist
June 2019

Supervisor: Jan Olhager, Professor at LTH
Examiner: Louise Bildsten, Associate senior lecturer at LTH
Company Supervisor: Alexandra Wikström, Sales & Operations Planner at Axis Communications
Company Supervisor: Kaushal Sadashiv, Operations Developer at Axis Communications

Preface

This master thesis constitutes the final project of the Master of Science in Industrial Engineering and Management, at the Faculty of Engineering at Lund University. The study has been conducted in collaboration with Axis Communications in Lund, who has provided both a context for this study, as well as assistance and expertise.

We would like to thank the employees of the Operations department of Axis Communications for the opportunity to conduct this study in collaboration with you. Your openness and benevolence gave us insight to the welcoming air of the company, we thank you for all your support. We would like to address an additional thank you to Robert Lindroth, Alexandra Wikström and Kaushal Sadashiv for your thoughts and input to the development of the project.

Further, we would like to express our gratitude to our supervisor Jan Olhager at the Division of Engineering Logistics at the Faculty of Engineering, Lund University, for the encouragement and guidance you have provided through all stages of the project - thank you!

Lund, June 2019

Anna Klackenberg and Louise Rundqvist

Abstract

Title: Exploring the use of machine learning to validate country of origin

Authors: Anna Klackenberg & Louise Rundqvist

Supervisor: Jan Olhager, Division of Engineering Logistics, Faculty of Engineering, Lund University

Background: In a globalized world, the role of country of origin (CoO) of a product is becoming increasingly important. The CoO of a product may affect where it can be sold. In the case of the US market, rules give preference to domestic products and products from countries which are included in the Trade Agreements Act (TAA). Relating to these regulations, Axis Communications (Axis) is looking to increase sales to US governmental projects, and therefore aspires to map TAA-compliance and verify CoO of their products. As Axis also makes strides to apply machine learning (ML) to their operational work, the project to use ML to verify products' CoO was initiated as a lighthouse project.

Purpose: The purpose of this study is to explore the possibilities of using an ML implementation in the context of validating CoO-related information in a flexible supply chain environment.

Research question: The central research question for this study is:

How can ML be used to validate a product's CoO at Axis Communications?

This question is assessed through exploring the following sub-questions:

RQ1 Which data is needed for a successful ML implementation in the Axis context?
RQ2 How can an ML model be created to validate CoO?
RQ3 How feasible is it to implement an ML solution for validating CoO at Axis?

Delimitations: This master thesis is limited to only consider Axis products for the US market. Information to be used in the ML model is limited to only be retrieved from Axis system for enterprise resource planning (ERP).

Methodology: The study was performed following a design science research approach, in which the problem was explored through developing an ML model to verify CoO. The

research approach was applied through a four-phase research process: preparation phase, observation phase, solution phase and conclusion phase. The preparation phase aimed at finding and articulating the problem to be solved. The observation phase included literature reviews for the topic of ML and CoO, as well as empirical data collection at Axis. The solution phase included data collection and processing, model development and evaluation. The discussion and conclusion were then established in the conclusion phase.

Conclusions: The study mapped which data from the ERP-system was needed for an ML implementation regarding validating CoO. ML models were developed according to the established workflow of ML development. However, further implementation of ML to validate CoO is not recommended. The study identified four limitations to the use of ML in an operational context within Axis:

(1) Lacking storage of historical data
(2) Difficulties in retrieving data from the ERP-system
(3) Lacking experience and knowledge of how to analyze input data for retraining
(4) Lacking experience and knowledge of how to interpret and validate the result

As a result, this study recommends Axis to revise data collection and availability, how the CoO in the ERP-system is to be used, and to explore automation as an aid in working with CoO.

Keywords: *Design Science Research, Country of Origin, Machine Learning, Binary classification, Supervised Learning*

Abbreviations

ACC	Accuracy, an evaluation metric for machine learning.
AUC	Area under the ROC-curve, an evaluation metric for machine learning.
BAA	Buy American Act
CBP	US Customs and Border Protection
CoO	Country of origin
DSR	Design science research
ERP-system	System for enterprise resource planning. In the case of Axis, IFS is used.
FN	False Negative
FP	False Positive
HS	Harmonized System
IFS	System for enterprise resource planning used at Axis.
kNN	k-nearest neighbor, type of machine learning algorithms.
ML	Machine learning
PDG	Product Data Group, department within Axis.
PTG	Product Test Group, department within Axis.
ROC	Receiver operating characteristic, an evaluation metric for machine learning.
ROO	Rules of origin
SVM	Support Vector Machine, type of machine learning algorithms.
TAA	Trade Agreements Act
TN	True Negative
TP	True Positive
UCC	Union Customs Code
WCO	World Customs Organization
WTO	World Trade Organization

Machine learning glossary

Algorithm	Set of instructions. ML algorithms build mathematical models using the sample data, to make predictions.
Bias	How far from the actual observations that the predicted observations are on average.
Classifier	Function of an ML algorithm to a classification problem.
Estimator	Function of an ML algorithm.
Example	One row of a dataset, containing features and possibly a label.
Feature	Variable of the dataset which has been processed to be understood by the ML model.
Hyper parameters	Parameters that must be set before training as they are not learned during, as opposed to model parameters.
Label	Target answer that the ML model is trying to predict.
Labeled example	Example used for training an ML model which has a corresponding label.
Model	The function of a trained ML algorithm.
Overfitting	An overfitted model has high variance and low bias, where it makes errors due to sensitivity to random noise and small variations in the training data.
Underfitting	An underfitted model has high bias and low variance, where it makes errors due to bad assumptions and missing relevant connections between the features and the label.
Unlabeled example	Example used for training an ML model which does not have a corresponding label.
Variance	How scattered the predicted values are from the actual values.

Table of contents

1 Introduction

This initial chapter aims to provide the reader with an introduction and overview of the study. The chapter begins with a general description of the background in section 1.1, followed by an introduction to the problem to be addressed in section 1.2. Next, the purpose, research questions and delimitations of the study are presented in section 1.3 and 1.4 respectively. The introduction is then rounded off with an overview of the chapters of the report in section 1.5.

1.1 Background

In an increasingly globalized world, the role of a product's country of origin (CoO) is important. Countries prefer to import goods from some countries, while other countries are banned, due to factors ranging from quality perceptions to cyber security. Countries view products from varying countries differently, where for example the Middle East see European products as high quality whereas the US prefer goods produced locally in the US. CoO is used to show this economic nationality of goods in global trade, why the importance of certain and correct CoO is increasing. Other global trends affecting the role of CoO are sustainability and ethics, where it is important to be able to trace where, by whom and under which conditions goods have been produced. (Hjelmström, 2019; Lilja Ivarsson and Kos-Hansen, 2019)

CoO is used for trade policy measures such as trade preferences, tariff quotas, anti-dumping measures and trade statistics, but are also important because several policies discriminate between countries. The CoO of a product is determined by rules called Rules of Origin (RoO). The RoO becomes very complex in the global trade where a product often is obtained and processed in several countries before it is ready for sale. (World Trade Organization, n.d.) This complexity is the case for Axis Communications' (Axis) production and supply chain.

Axis is a manufacturer in the security and surveillance industries, providing products and solutions such as network cameras, video encoders and recorders, video management software and analytics (Axis Communications, n.d.). Applications of the products include mobile surveillance, facial recognition, people counting and sound detection, where some product integrate artificial intelligence and machine learning (ML) (Lindroth, 2019).

Axis want their product portfolio available at any time. This goal is accomplished by having a flexible supply chain, for example through changing suppliers when needed. This flexibility causes their products' CoO to change time and again. In government procurement in the US, there are regulations through the Buy American Act (BAA) that aims to protect national businesses and labor by restricting purchasing of foreign goods. However, the Trade Agreements Act (TAA) works as a waiver of BAA and is often used instead. The TAA controls which CoOs that are approved for public purchasing when the product is not from the US. European countries are listed as TAA-compliant but countries such as China and Thailand are not. Any Axis goods that has a CoO that is non-TAA-compliant cannot be sold for US governmental projects. (Axis Communications, 2018a; Hjelmström, 2019)

In order to be able to compete in federally funded programs in the US, Axis has initiated projects aiming at establishing a list of TAA-compliance for Axis products. This has the potential to increase sales and help Axis to maintain market leadership. The aim of the list is to show which products are consistently TAA-compliant, which have the potential to be made TAA-compliant, and which are never compliant. Those goals are dependent on data accuracy of CoO which is why the Axis projects also includes going over practices of implementing rules of changing CoO, as well as methods to detect faulty data. (Axis Communications, 2019, 2018a).

These projects are initiated by the Operations department at Axis. This department also has a wish to explore possible areas of using machine learning (ML) (Lindroth, 2019). When the wish for finding and implementing a method for detecting possibly faulty CoO was expressed, ML seemed to be a possible solution (Hjelmström, 2019). Axis has several projects and implementations where ML is used in Axis products, and now wishes to expand this knowledge into the company's operational functions (Lindroth, 2019).

ML consists of algorithms, systematic sets of procedures, that constructs a model from input data which can make predictions about new, but equally distributed, data (Google developers, 2019a). ML can be used for solving business problems since it can extract knowledge from and find patterns in large amounts of data. The volume and sources of data at companies increase and therefore the complexity of the systems that handle this data grows. Regular software engineering approaches cannot solve this complexity and companies therefore explore the use of ML. The enthusiasm surrounding ML is big, and it is on the peak of inflated expectations according to Gartner's Hype cycle for Data

science and Machine Learning (Krensky and Hare, 2018). This implies that ML is expected to have a transformational benefit within two to five years, where it can have effects on a range of areas in businesses. For now, many companies are still in early stages of exploring the use of ML.

1.2 Problem introduction

In order to become more competitive and shorten response time on the US market, Axis aimed to produce a list of products which are either TAA-compliant, can be made TAA-compliant or are not possible to make TAA-compliant. As Axis started compiling a list of TAA-compliant products, inaccuracies in the existing CoO of the products in the ERP-system were found. The inaccuracies were due to information missing, and data fields not updated when dependent data changed. The discovery of faulty CoO-data initiated a project for mapping and rectifying the problems. After the initial mapping, four grounds were found of why CoO in the ERP-system might not be correct:

(1) Lack of ownership of the data connected to CoO,
(2) Inadequate processes and ways of working regarding CoO,
(3) Information and updates from external stakeholders,
(4) Changes in the product life cycle, e.g. change of supplier (Olander, 2019).

Axis estimates that the corrective measures (introducing ownership, reinforcing work processes and cleaning of the existing data) will take approximately one year to implement (Hjelmström, 2019). However, the project group requested a method for correcting CoOs of products earlier than in a year. In an effort to reduce or aid in the otherwise manual labor, the project group came to the conclusion that an ML model could have potential use in the process of cleaning the CoO-related data.

By indicating products with a high probability of having a faulty CoO, the employees can then manually investigate the products in a prioritized order. Hence, the use of ML is purely intended as an aid, and the definitive determination of CoO is made by an Axis employee.

1.3 Purpose and research questions

The purpose of this study is to explore the possibilities of using an ML implementation in the context of validating CoO-related information in a flexible supply chain environment.

In order to fulfill this purpose, this study aims to answer the following general research question:

How can ML be used to validate a product's CoO at Axis Communications?

This question is assessed through exploring the following sub-questions:

RQ1 Which data is needed for a successful ML implementation in the Axis context?
RQ2 How can an ML model be created to validate CoO?
RQ3 How feasible is it to implement an ML solution for validating CoO at Axis?

1.4 Delimitations

Since the task to validate CoO using ML was requested by a project group which focuses on the US market, the initial prototype will be limited to consider only Axis products on the US market.

Since the experiments are requested by Axis to only consider information from the ERP-system, this study will not consider alternative sources of information.

Given that this is a master thesis, the project will be dimensioned to be finished within 20 weeks, including preparatory work, the main exploration, writing the report and additional academic requirements.

1.5 Report outline

Chapter 1 Introduction

The first chapter provides background, problem introduction, purpose and research questions, as well as the delimitations of the study.

Chapter 2 Methodology

In the second chapter the choice of design science research as research approach is presented, as well as the research process of the study. This is followed by methodology for literature review and empirical data collection. The chapter is then concluded with research quality of the study.

Chapter 3 Theoretical findings on machine learning

This chapter presents an introduction to ML and key concepts of ML development. It provides the theoretical frame of reference to the use of ML. Three ML development workflows presented by major actors within the field are summarized into a generalized workflow, which is later to be applied in the study.

Chapter 4 Theoretical findings on country of origin

This chapter introduces the concept of and the criteria relevant for deciding CoO. The chapter puts special emphasis on EU and US regulations, as these are the regulatory systems relevant for a European company present on the US market.

Chapter 5 Empirical findings

This chapter serves as the environmental context of the study. It presents an overview of Axis current supply chain setup, product structures, external factors affecting CoO of Axis products, and how the company is currently working with CoO. Further, the chapter shows a mapping of how CoO information is currently managed in the ERP-system and challenges regarding the CoO question. The chapter is concluded with the specification requirements of an ML model used to validate CoO information.

Chapter 6 Applying machine learning to country of origin

This chapter presents the process of developing four ML models attempting to validate CoO information in the ERP-system. The development follows the generalized workflow presented in chapter 3 Theoretical findings of machine learning. After collection and analysis of the available data, the problem framing presents what type of ML problem can be used. The chapter then recounts how the preparation of data was performed, how the models were developed, as well as which tools and libraries were used.

Chapter 7 Result and analysis

Chapter 7 presents the general results of performance for the four developed ML models, as well as detailed results on country-level for the model showing the most promising general results. The chapter is concluded with an analysis of the results.

Chapter 8 Discussion

This chapter presents discussions of the study. The discussions start with a section on the limitations of the developed ML model and fulfillment of the requirements presented in chapter 5.2. The discussion is then broadened to the general topic of using ML in the context of the CoO question at Axis.

Chapter 9 Conclusion

This chapter presents the concluding remarks of this study, including fulfillment of purpose and research questions, recommendations for Axis, and areas for future research.

2 Methodology

This chapter presents the methodology used throughout the study. Section 2.1 presents the research approach, which then acts as the foundation of the research process. The following sections dive into how the literature review and empirical data collection was performed, in section 2.2 and 2.3 respectively. The chapter is then concluded with section 2.4 on research quality.

2.1 Research approach and process

According to Holmström, Ketokivi and Hameri (2009), design science is research that seeks to fulfill either of the following purposes:

- Explore new solution alternatives to solve problems
- Explain the explorative process of solution alternatives
- Improve the problem-solving process

The key principle of design science is the interest of developing "a means to an end", where the researcher's focus is problem solving and solution design. The researcher will have a participatory role in the project's environment, instead of having a solely observational role. In the case of design science the means or method is development of an artefact to solve a problem, which Holmström et al (2009) describe as *exploration through design*.

Takeda, Veerkamp, Tomiyama and Yoshikawa (1990) describes the iterative process of conducting design science research in the model of *design research cycle*. The authors divide the process of performing design science research into decisions on two levels: the object level, consisting of decisions of the design artefact, and the action level, consisting of decisions of how to proceed. The research process is divided into five steps, as illustrated in figure 2.1:

(1) Awareness of problem: Identifying and choosing a problem area to explore.
(2) Suggestion: Propose key concepts which can be of use in solving the problem.
(3) Development: Designing possible artefacts aimed at solving the problem.
(4) Evaluation: Evaluate the designed artefact regarding factors such as cost.
(5) Conclusion: Decide on an artefact for further actions.

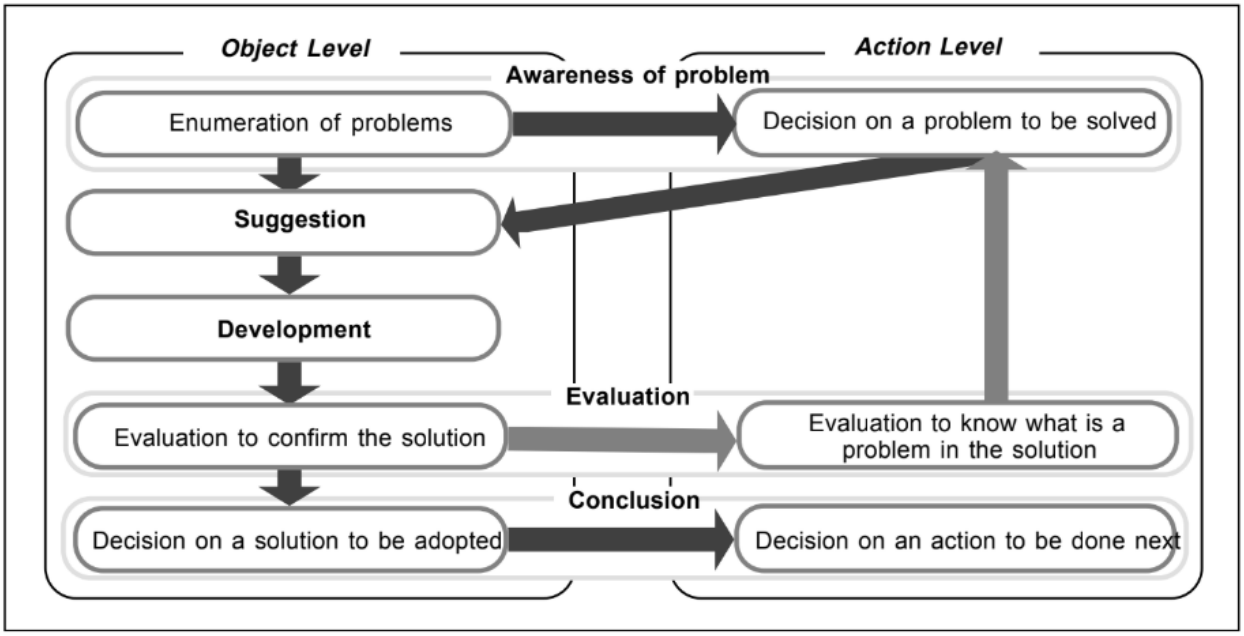

Figure 2.1. The design research cycle. (Takeda et al., 1990)

As this study aims to explore a new area of use for ML, the study falls into the first category of design science as described by Holmström et al. (2009): exploring new solution alternatives to solve problems. Through the experimental development of ML models, the study is therefore both exploratory and problem-solving in its nature. As the study intends to design artefacts to explore the problem, the design science research approach is deemed as a suitable procedure. The research process of the study is divided into four phases: Preparation, Observation, Solution and Conclusion, as shown in figure 2.2. The phases of the research process follows the general format described by Takeda et al. (Takeda et al., 1990).

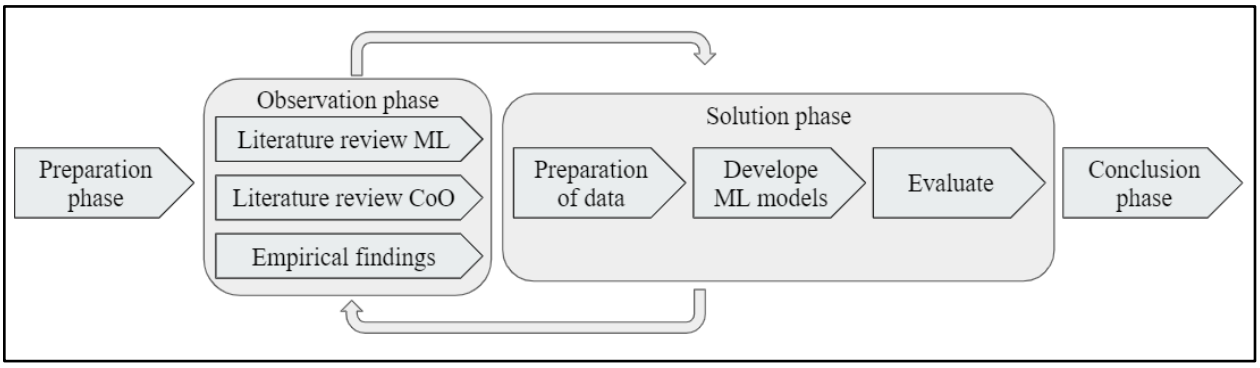

Figure 2.2. The research process.

In order to follow a design science research approach, the research process for this study also adopted elements from the concept of Design Science Research Cycles presented by

Hevner (2007), illustrated in figure 2.3. The framework shows how the relation between the artefact of a design science research project, the contextual environment and the existing knowledge base is established through three cycles: the Relevance Cycle, the Rigor Cycle and the Design Cycle.

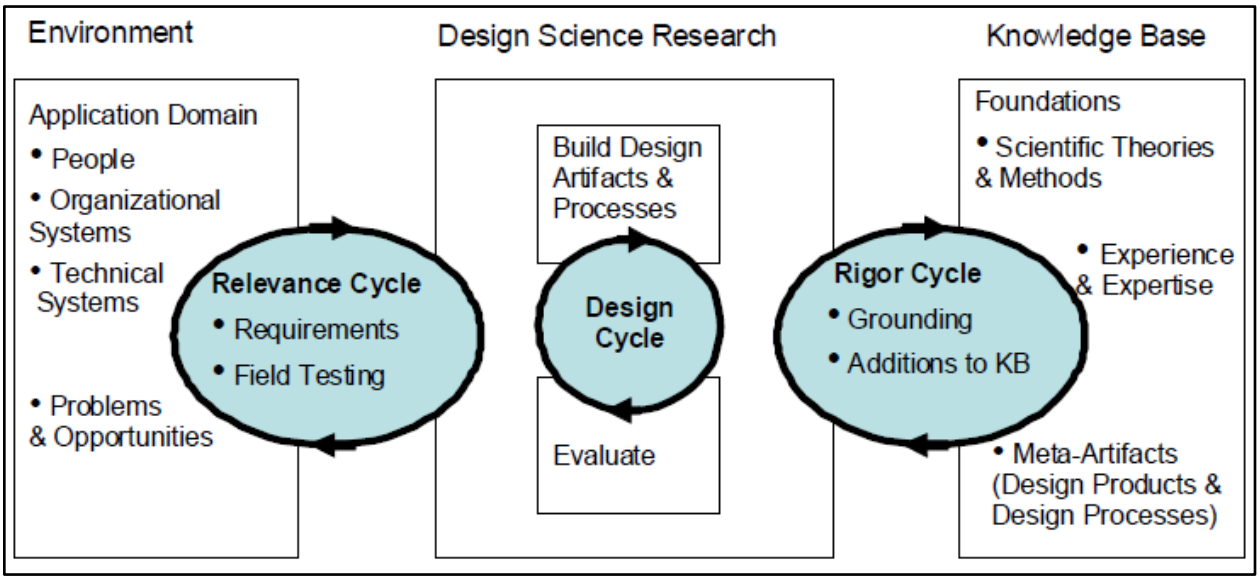

Figure 2.3. The Design Science Research Cycle. (Hevner, 2007)

The aim of the Relevance Cycle (Hevner, 2007) is to provide context to the research, by grounding the design science research in its environment. In the context of this model, the input from the application environment into the design science research project includes exploration of possible opportunities within the environment, analysis of the problem the research is addressing, and also clearly defining which requirements and acceptance criteria are to be used when evaluating the final research results. In turn, the output from research project to the environment consist of field testing, where the artefact is reintroduced into its original environment. Within this study, the requirements and evaluation criteria are established through interviews with Axis employees. The interviews will be conducted as part of the observation phase of the research process. The approach of the interviews is further developed in section 2.3 Methodology for empirical data collection.

The Rigor Cycle (Hevner, 2007) connects the design science research with the existing knowledge base, which consists of scientific theory and method, experience and expertise of the researchers, and artefacts (products and processes) within the organization. The purpose of the Rigor Cycle is to ensure that the design science research creates innovation, and then returns the findings to the knowledge base. Within this study the

knowledge base consisted of employees with technical competence and knowledge of CoO within Axis and a literature review performed during the observation phase of the research process, covering both CoO and ML. The proceedings of the literature review are examined in section 2.2 Methodology for literature review.

The Design Cycle (Hevner, 2007) is the development and evaluation of the solution design or artefact. This is an iterative process, where alternatives are generated, evaluated and improved upon until the artefact fulfills the evaluation criteria. Within this study the Design Cycle and model development was performed during the solution phase of the research process. Methodology of the ML model development is based on the theoretical findings on ML development in chapter 3.

2.2 Methodology for literature review

Academic work starts with looking at what is already known (Silverman, 2010). Both policies and theories as well as already existing research was reviewed for the frame of reference. The purpose was to come up with frameworks for ML and CoO which were used as a basis for motivating the chosen algorithm(s) for the solution phase. Hence, the purpose of the literature review was to investigate if ML has already been applied to similar problems and if not, analyze the result for a potential application of ML. The purpose was further to describe the context that is CoO in order to understand the problem and CoO rules.

The methodology used in the literature review is presented here for transparency. The literature review was conducted iteratively, throughout the study, as new information from the interviews and prototype development arose. Firstly, the aim of the review was to gain an understanding of ML and CoO, find frameworks that are appropriate for the study and is hence descriptive in nature. The scope was to have a limited focus on the narrow focal questions. The literature review had a plan for identifying useful studies which increases transparency. The searching did not cover all studies, due to limitations in time and resources and because the literature review was not the main focus of this study. Inclusion and exclusion criteria was developed and applied to the studies. An analysis was conducted in a structured way. According to Jesson et al. (2011), this literature review approach has elements of a systematic review as well as a traditional review and will be called literature review or simply review.

The plan was developed as following. Questions were developed in order to guide the literature search. They were focal for the literature review since it served to inform the rest of the study and can be found in table 2.1.

Table 2.1. Focal questions for guiding the literature search.

Questions for ML:	What is the definition of ML? Which options of algorithms are there to use ML for validation? Are there any frameworks for conducting an ML project?
Questions for CoO:	How is CoO defined? Are there any differences in definitions depending on market?

Inclusion and exclusion criteria were set up, based on reasoning, as shown in table 2.2.

Table 2.2. Inclusion criteria for literature review.

Type of criteria	**Criteria for inclusion**	**Reasoning**
Language	Swedish or English	English is the dominant language in SCM and ML. Swedish sources was also included since it is the authors' first language.
Timespan	-2019	The area of ML and CoO are developing fast and hence mostly recent literature was included.
Publication type	Peer-reviewed articles Grey literature	If peer-reviewed articles were available, this was used. Since it is fast developing areas, grey literature was also used to a large extent.

Academic peer-reviewed material from academic journals is considered to be the most reliable source since it has been assessed by two or more experts to ensure quality (Jesson et al., 2011). However, all knowledge in the research area could be investigated, in lower rated journals as well as papers that have not yet been peer reviewed, called grey literature. One reason for including grey literature is that the publishing in highly rated journals can take two to three years according to Jesson et al. (2011). For the topic of

ML, grey literature such as lecture notes, documentations, handbooks, websites of big actors, industry journals, technical reports and white papers were included in order to not miss the latest advances. Regarding the topic of CoO, international organizations are the main actors of the topic. Therefore, the information regarding CoO published on their websites were included.

Key articles were firstly sought for that could serve as starting points for further research, by searching for papers written by the same authors, using keywords found in the abstract and attached to the article and articles in the reference list. This is a good way to find further useful articles according to Jesson et al. (2011). The title, abstract and keywords were scanned in the hits that seemed most promising. If those parts could not give enough information, the conclusion was scanned as well to understand if the article was relevant by checking against the inclusion criteria. The relevant literature was skimmed, to make sure that it was relevant and still met the inclusion criteria. The aim was to find main points, core concepts, definitions by underlining relevant parts and taking notes that could later be needed. Finally, the full paper was read in order to understand it fully and quality and trustworthiness was assessed. Assessing trustworthiness was especially important in the cases of grey literature (Höst et al., 2006). Both authors of this study did not read all the articles because of limited time.

Recurring themes were analyzed in order to come up with frameworks, making connections, identifying gaps, mapping and drawing conclusions. Finally, the frame of reference was written, describing the findings and the areas.

2.3 Methodology for empirical data collection

Empirical data was collected in order to understand the environment that is how Axis works with the CoO question. Furthermore, the technical requirements were specified. This data was collected through interviews and complemented with internal documents, internal meetings and informal talks.

Advantages of using interviews as a method for collecting data is that it can give deep and detailed information about the topic according to Denscombe (2010). It is flexible and changes can be made to the questions during the interview depending on what comes up. Interviews can be either unstructured, semi-structured or structured. Structured interviews has predetermined questions and the response options are limited. The interview is highly controlled and standardized which makes the analysis straight forward and hence gets close to quantitative data collection. It is suitable when the purpose of the

interview is to describe or explain the phenomenon (Höst et al., 2006). The unstructured interview has focus on the respondent's thoughts, emotions and experiences. The interview has a topic and the interviewers have prepared interview questions but the questions can change, be rephrased and be asked in different orders (Höst et al., 2006). In the semi-structured interview, the interviewer has a list of questions to be answered but has to be more flexible than in the structured interview, where the interviewee is allowed to speak freely about the topic (Denscombe, 2010).

The types of interviews used in this study are presented in table 2.3. Unstructured interviews were conducted in the preparation and solution phase. In the preparation phase, the purpose was to find an area of research and the researchers did not know much about the subject. As well, help with the technical setup was needed. In the solution phase, the purpose was to get input on the development of the design artefacts and getting input on a suggested solution is one reason for conducting interviews according to Höst et al. (2006). Unstructured interviews were appropriate to use as a data collection method in those phases because they were used for exploration (Denscombe, 2010; Höst et al., 2006).

Table 2.3. Interviews in different phases of the study. Based on Rosengren and Arvidsson (2002) and Höst et al. (2006).

Phase	**Research purpose**	**Interview**	**Selection**	**Aim/Purpose**
Preparation	Exploratory	Unstructured	People with information Technical expert	Find an area of research Getting help with technical set up
Observation	Descriptive	Semi-structured	Key players	Explain the CoO process Articulate requirements and evaluation criteria
Solution	Exploratory	Unstructured	Technical expert	Get input on the development of the ML models

The purpose with the interviews in the observation phase was to describe the CoO process and problem as well as develop requirements for the design artefact for the

solution phase. Insight was needed with explanations and interviews are appropriate for this (Yin, 2009) and chosen as a data collection method in this phase as well. Furthermore, for this purpose, semi-structured interviews are appropriate and hence used (Rosengren and Arvidson, 2002).

The selection of people to interview was in the preparation phase people with information at Axis Operations as well as people with knowledge about ML. In this phase, the organization and its problems were explored to find a suitable problem and area of research. According to Rosengren and Arvidson (2002), it is more a choice of study object than a selection where the purpose is to find the people with the most information and contribution. The next individual to interview was decided by the information collected from previous interviews. This was repeated until an area of research was found.

People with knowledge about the CoO problem were selected in the observation phase. Axis supply chain and the problem with CoO is complex since the process of where things do not work as it should, is not clear. Key players at Axis have unique information about the process, how CoO is used and those were interviewed in the observation phase. The main goal with this research study is not to be able to generalize beyond this case and hence key players could be chosen (Denscombe, 2010). Those key players, and other interviewees as well, were Axis employees and could be reached at Axis office by the researchers. A list of interviews conducted can be found in appendix A.1.

Interview questions that needed answers to were developed for the semi-structured interviews. The starting point for the interview questions was what information was needed from the informants, how much time was needed, how much was already known and how to relate to this knowledge in the interview. The funnel method, starting with broader question and get more specific and end with the most detailed questions (Yin, 2009), were used. An outline of the interview questions was sent in advance in order for the informants to have time to prepare, in accordance with Yin (2009). Interview questions can be found in the interview guide in appendix A.2.

The interviews were conducted by two interviewers, where one was taking a lead interview role and the other was taking a lead collecting role. The interviews were recorded to be able to collect information in new areas that might appear, which is suggested by Höst et al. (2006). After the interviews, summarized transcriptions were made since exact words were not needed. After the transcriptions were made, the

interviews were followed up. Completed interview questions with answers were sent to the interviewee for authentication and correction. Earlier interviews were returned to, and follow-up questions sent by email, when gaps or unclear areas were identified before further interviews were conducted, which is recommended by Voss et al (2002).

Data that employees at Axis have collected and compiled in internal documents have also been used for triangulation and for explaining areas that the interviewees did not have enough knowledge in. Informal meetings and talks have also been taken place for collecting specific data.

2.4 Research quality

Quality of a study can be measured in four dimensions: construct validity, internal validity, external validity and reliability (Yin, 2009).

Construct validity is defined as “the extent to which correct operational measures are established” (Olhager, 2019, p. 45). Tactics used in this study is that several sources of evidence have been used in the empirical data collection phase. Several interviews have been conducted with different informants, in order to validate the information. Similar questions in both of the semi-structured interviews were asked and the same information in meetings, informal meetings and from internal documents was collected. This data triangulation, using multiple data sources of evidence, enhance the construct validity (Yin, 2009). Furthermore, interviewees have confirmed analysis drawn from the interviews and adjusted where the understanding was not correct.

In the ML development phase, the available grey literature were investigated to come up with a process on how to conduct this part of the study. Before, during and after the development phase, the process, questions and issues were discussed with Robin Gustafsson (System developer) for technical guidance and support, to validate the work and results.

Internal validity is defined as “the extent to which a causal relationship can be established” (Olhager, 2019, p. 45). It is not applicable in this study, as it is not explanatory. This study does not aim at explaining why one thing led to another but rather try to describe the problem which the model is aiming at solving. Hence, internal validity is not applicable.

External validity is defined as “the extent to which a study’s finding can be generalized” (Olhager, 2019, p. 45). Yin (2009) recommends to choose more than one case and having a multi-case design if it is possible, to increase theoretical replication and generalizability. Having a multi-case design was not possible in this study. Since this is a company-specific case study, it is hard to determine to what extent the findings are generalizable. However, throughout the report, it is documented and described in what environment, under what conditions and which assumptions are made, to increase the possibility of later determine if it can be generalized beyond this case to another.

Reliability is defined as “the extent to which a study can be repeated, with the same results” (Olhager, 2019, p. 45). Errors and biases should be minimized in a study (Yin, 2009). A folder for the data collected during the interview phase is kept, making it possible to investigate the evidence and information later. Even from informal talks and meetings, notes were taken and kept in folders, which were used in the study, making it possible to review later. Folders are kept with explanatory names, to make it easy to access the right evidence needed later. Folders have for example been divided into which part of the process they were collected and relevant for, i.e. the preparation, observation, solution and conclusion phase. This kind of database increase reliability according to Yin (2009). Also, the two authors of this study were both making the observations, which according to Yin (2009) further increase the reliability.

Logbook was used to collect daily thoughts, ideas and analysis that came to mind when talking to people in an unplanned way. Another reason for using logbook is to document what is done and why to later see how different paths and directions were motivated. (Höst et al., 2006)

Furthermore, reliability and validity in data collected in case research is strengthened by having a well-constructed research protocol, according to Voss (2002). A research protocol consists of the research instruments, the procedures for using those, from where and who data will be collected and centrally the interview questions. It is especially important in a multi-case study (Voss et al., 2002). Since this is a single-case study, the research protocol was not as well-structured.

It was validated by the Axis manager of the TAA-project that the respondents chosen were key informants. Tape-recorders were used during the interviews and the interview was later transcribed and verified by the interviewee(s), in order to be certain about what was said.

3 Theoretical findings on machine learning

This chapter aims at introducing the reader to the topic of ML. First, a general introduction including definitions of ML is presented in section 3.1. It is followed by a description of four common ML algorithms for classification problems in section 3.2. In section 3.3, workflows of three major actors in the field of ML is summarized into a general workflow. The steps of the summarized workflow are then expanded in section 3.4 through 3.9.

3.1 Introduction to machine learning

According to Google Developers (2019a), ML is defined as:

> "[...] a program or system that builds (trains) a predictive model from input data. The system uses the learned model to make useful predictions from new (never-before-seen) data drawn from the same distribution as the one used to train the model."

This means that an ML model identifies and learns patterns and relations between data and previously known answers. When the model is trained, it can then apply the patterns and relations to never-before-seen data in order to predict what the target answer should be. The performance of the model is then evaluated by comparing the predictions with the true answers. This workflow is illustrated in figure 3.1.

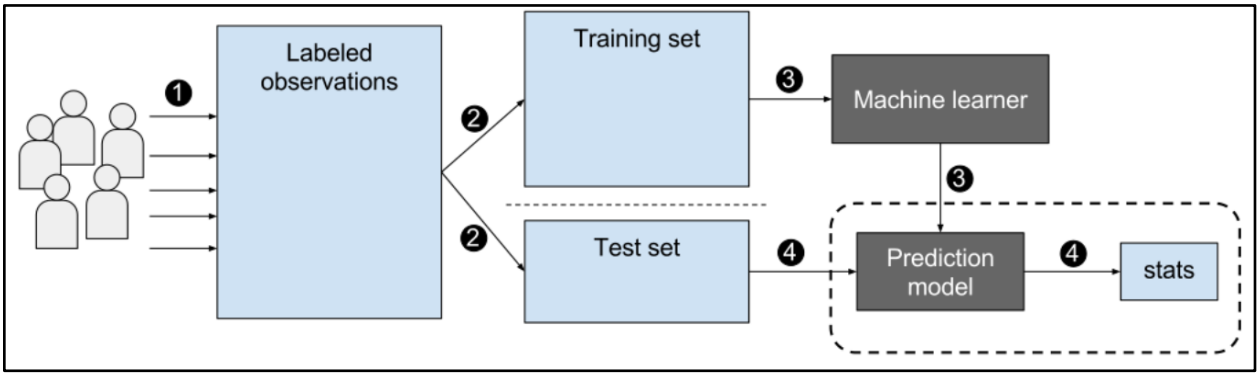

Figure 3.1. The training and evaluation of ML models are based on data with previously known target answers (labeled observations). By learning the patterns on a subset of the data and answers (training set), the ML model can make predictions on never-before-seen data (test set). The predictions are then compared to the true answer for evaluation. (Salian, 2018)

The computational force of the program or system is in so called machine learning algorithms, systematic sets of procedures. MathWorks (2016a) describes ML algorithms as algorithms which "[...] use computational methods to 'learn' information directly from data without relying on a predetermined equation as a model". Mathworks' description puts emphasis on the difference between machine learning and traditional computational methods: all steps of logic in traditional software engineering are known and understandable for humans, while ML systems instead learn and interpret the patterns in the data to form a different kind of logic.

When starting an ML project it is not guaranteed that a usable model will be found (Google developers, 2019b). Google Developers (2019c) even warns of this hype in their Rules of ML: "Rule #1: Don't be afraid to launch a product without machine learning.".

Amazon (2016) lists two scenarios where ML can be used: (1) the rules of the problem cannot be coded in a simple and deterministic way, and (2) the process does not scale, for example if the process includes manual decisions. When following these guidelines, ML can be used to handle large scale problems, and also problems where the answer depends on a large number of factors or rules which are overlapping. In addition, ML also requires a lot of relevant data. If data is not available, ML is not a viable solution (Google developers, 2019b).

ML problems can be categorized into different types of problems depending on the available data and the type of sought after predictions (scikit-learn, 2019a). The most general split is between supervised learning and unsupervised learning. Unsupervised learning aims to find patterns in data, such as clustering data or finding correlations through principal component analysis (Google developers, 2019a). The input data (examples) of unsupervised learning are unlabeled, which means that the input data in training does not have corresponding answers (Google developers, 2019a). Unsupervised learning is not used in this study.

Supervised learning aims to predict an output by learning the link between observed data and the result used in training (scikit-learn, 2019a). There are several subclasses to supervised ML problems, such as classification, where the outcome is a category or label, and regression, where the output is a continuous target variable. In supervised learning each example of the sample data consists of both input parameters (features) and the corresponding answer (label) (scikit-learn, 2019a). This study explores a classification task, thus solving a supervised ML problem.

Figure 3.2 shows different subtypes of classification, depending on the number of possible classification categories in the dataset, as well as number of labels present in each example (Google developers, 2019b).

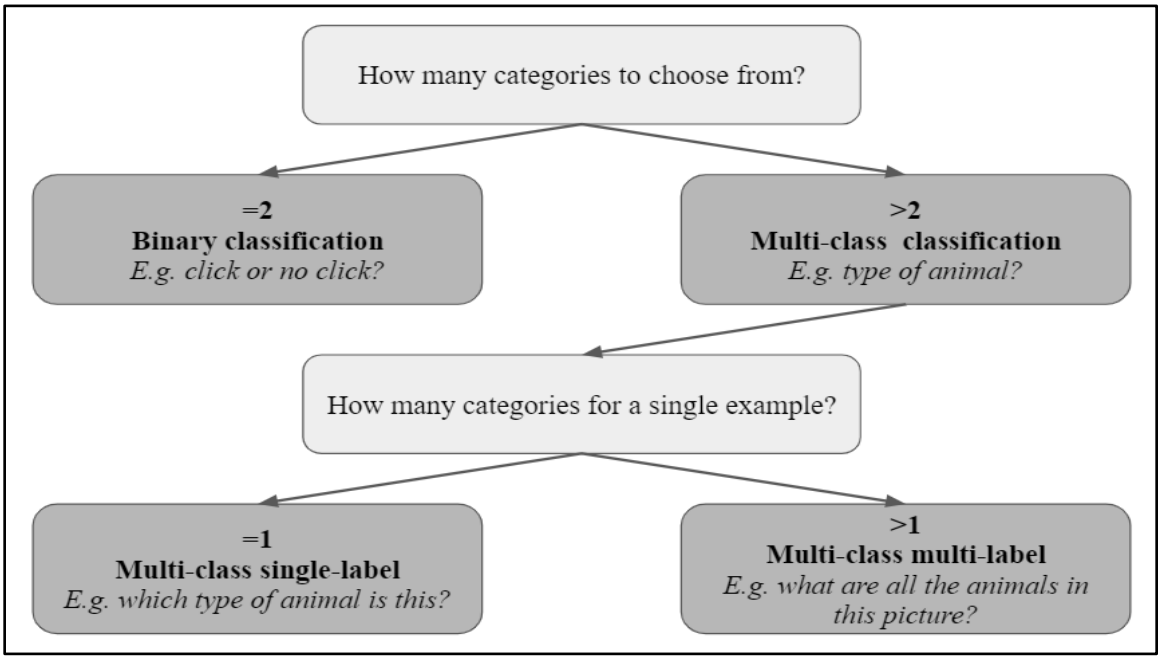

Figure 3.2. Flowchart of how to differentiate between classification subtypes. Adopted from (Google developers, 2019b)

3.2 Algorithms for classification

There are several algorithms available for a classification task, with different benefits and drawbacks. According to Mathworks (Mathworks Inc, n.d.), characteristics of algorithms are for example speed of training, memory usage, predictive accuracy on unseen data, transparency and interpretability. Further, Mathworks has a guide on how to choose initial classification algorithms, when tradeoffs have been decided. Choosing an algorithm is not a straightforward process with one correct answer. Instead it is a process of trial and error where multiple algorithms are explored. Described below are four common algorithms used for classification problems.

Decision tree

The decision tree classifier produces a sequence of rules from training which is then used to classify new data. It is easy to understand, visualize and can manage both categorical and numerical features. It might create complex trees from training, which makes the

model overfit (scikit-learn, 2019a). Further, it is recommended to start with a decision tree as it is fast and easy to understand (Mathworks Inc, n.d.).

The concept of decision trees is that nodes of the trees are features, which each has a set of potential values. The branches from each node represent the values of that feature. Predictions of new observations are made by going with the rules in the tree from the root to a leaf node (Nugues, 2019). During training, the split at each node is often created by choosing the best split or the best random split (scikit-learn, 2019a). The best split could be measured with different metrics such as the information gain. The root of the tree will be the most discriminating feature, while the following nodes are less and less discriminating (Nugues, 2019).

Overfitting occurs when the decision tree is deep, i.e. has many vertical nodes, and when several leaves only have few observations. Pruning or adjusting hyper parameters, such as the maximum depth or minimum number of observations required at a leaf node, can be done to simplify the model. (Nugues, 2019)

Decision trees can be unstable where a small variation in the training data can result in a completely different tree, which can be avoided by using an ensemble method where multiple algorithms are combined (scikit-learn, 2019a).

Random forest

A random forest classifier is an ensemble method, based on the concept that weaker decision trees are combined to shape a stronger ensemble. It is best used when features are categorical or act nonlinearly, and when the training and prediction time is less important. It is more precise than decision trees for many tasks and reduces overfitting (Mathworks Inc, 2019a). Each tree is built from a subset of the total training set. The subset is sampled with replacement, which means that the observations can possibly be chosen several times from the total training set. This results in lower variance and the bias is kept the same, compared to a single decision tree. Furthermore, the split of a node in a tree is not the best split among all features, but instead the best among a random subset of features. (scikit-learn, 2019a)

K-nearest neighbor

The k-nearest neighbor (kNN) algorithm classifies an observation based on classes of their k nearest neighbor observations in the dataset. kNN assumes that observations close to each other are much alike. It is called a non-generalizing method since it just

“remembers” the training data. It uses relatively much memory for training and for making predictions, but is robust to noise. kNN is simple but still often very useful. Hyper parameters commonly tuned are the weight metric and the value k. The weight metric is the weight to assign to observations in the surroundings of the new observation. The weight parameter can be uniform, by distance, or inverse distance. Uniform weights are often used, which means that the nearest neighbors contribute the same, while distance gives preference to neighbors closer to the observation. K is the number of neighbors to consider when deciding on the class of a new observation. A large k makes a more complex classifier and lessens the issue of noise but makes the classification boundaries less definite. (scikit-learn, 2019a)

Support Vector Machine

A Support Vector Machine (SVM) classifies data by finding a hyperplane that divides all observations of one class from those of the other class. The best hyperplane is the one which has the biggest margin between the two classes, when the data can be divided linearly. If the data is not linearly separable, a loss function is used which penalizes observations on the wrong side of the hyperplane. SVMs can use a kernel to transform nonlinearly separable observations into higher dimensions where the data can be linearly divided. (Mathworks Inc, 2019a) Regularization could be adjusted with a hyper parameter, where more regularization means that all observations around the separating hyperplane are used for calculating the margin, whereas with less regularization, only the observations closest to the hyperplane are used. (scikit-learn, 2019a) When making predictions, the new observations are mapped into the space of the trained model, where the observations are predicted to belong to the class on which side they end up (Mathworks Inc, 2019a).

3.3 Workflow of a machine learning project

Though the process of working with ML is an explorative problem solving method of trial and error (Mathworks Inc, 2016b), the general process of using ML to solve a problem can be described in different steps. The steps are iterated over in a heuristic manner to find the best possible model and solution (Amazon Web Services, 2016; Mathworks Inc, 2016b). The number of steps vary between companies offering ML technology solutions. In an effort to find consensus, the work processes presented by Google developers (2019b), Mathworks (2016b) and Amazon Web Services (2016) has been compared and summarized into a generalized workflow presented in table 3.1. The steps of the generalized workflow are presented in the following sections.

Table 3.1. Comparison of workflows used when working with ML, as presented by Google developers (2019b), Mathworks (2016b) and Amazon Web Services (2016). The comparison is summarized into a generalized workflow.

Google Developers	**Mathworks**	**Amazon Web Services**	**Generalized workflow**
(1) Defining an ML problem and propose a solution			Problem framing
(2) Construct dataset	(1) Prepare data	(1) Analyze data	Preparing data and features
(3) Transform data		(2) Split data into training and evaluation data	
		(3) Shuffle training data	
		(4) Process features	
(4) Train a model	(2) Choose algorithm	(5) Train the model	Training the model
	(3) Fit a model		
		(6) Select model parameters	
	(4) Choose a validation method	(7) Evaluate the model performance	Evaluating the model
		(8) Feature selection	Improving the model
		(9) Set threshold for prediction accuracy	
	(5) Examine fit and update until satisfied		Iterate to find best model
(5) Use the model to make predictions	(6) Use fitted model for predictions	(10) Use the model.	Use the model for predictions

3.4 Problem framing

The initial step of using ML is to understand and articulate the problem. Google Developers puts emphasis on the importance of understanding the problem, before the problem is framed into an ML problem (Google developers, 2019b). Understanding the underlying problem aids in establishing what type of ML is relevant. Problem framing then includes establishing the target answer (Amazon Web Services, 2016) and identify what category of ML approaches that can be used to solve the problem (Google developers, 2019b). Examples of how different target answers affect the type of ML problem is shown in table 3.2.

Table 3.2. Examples of target answers for different types of ML problems. Adopted from Google developers (2019b).

Type of ML Problem	Description	Example
Classification	Pick one of N labels	Cat, dog, horse, or bear
Regression	Predict numerical values	Click-through rate
Clustering	Group similar examples	Most relevant documents (unsupervised)

3.5 Preparing data and features

When a problem is formulated, it is time to collect data for the model to be trained and tested on. The data needed for supervised ML consists of examples (also called observations), which in turn contain variables and the corresponding target answer (Amazon Web Services, 2016). A variable is data about the example which directly or indirectly gives information about the example. When collecting sample data, it is important to collect data of all categories. If the sampling data does not contain examples of all categories, the model can neither be trained nor verified to predict targets of the left out categories (Amazon Web Services, 2016). It is recommended that each variable value occurs at least five times in order for the model to learn how this feature affects the label (Google developers, 2019d).

The next step is to analyze the collected data. The predictive power of the model is directly correlated to the quality of the data going into the model (Amazon Web Services, 2016). Amazon (2016) lists two considerations when analyzing data: (1) variable and

target data summaries, and (2) variable-target correlations. The aim of the variable and target data summaries (1) is to understand the values of the data and identify dominant values as well as quality of the data. Amazon lists relevant questions regarding the data summaries as:

- Might there be a problem collecting the data?
- Are all categories equally frequent in the data?
- Are there missing values or invalid data?

The aim of variable-target correlations (2) is to establish the grade of relevance between a variable and target answer.

If the sampling data does not contain examples of all categories, data of those categories can be synthesized. Synthetic data can be produced programmatically to represent real-world data, which the ML algorithm can be trained on. (Yue et al., 2018)

When an acceptable dataset is available, the next step is to process the features. Feature processing is the process of transforming the variables of the raw, collected data into variables which can be used by and is relevant for the model (Amazon Web Services, 2016). These variables which can be used by the ML model are called features. Amazon (2016) mentions six general ways of feature processing:

- Transforming variables into usable features. For example, a variable containing date and time would not be relevant for an ML-model, since this date will never be repeated. However, if the variable is transformed into features such as time of day, day of week and month, then the information of the variable can be used by the model for future predictions made by the model.
- Replacing missing or invalid data. Missing data can be replaced by a default value, mean or median.
- Forming Cartesian products of two variables.
- Non-linear transformations such as splitting continuous numeric values into categories (so called bins). An example is that the value age can be transformed into categories of age groups.
- Domain-specific features. E.g. combining measurements of length and breadth into a feature describing area.
- Variable-specific features, where certain variable types can be processed in order to show structure or context. E.g. creating text features by forming n-grams of text from full sentences: “The fox jumped over the fence” can become unigrams (the, fox, jumped, over, the, fence) or bigrams (the fox, fox jumped, jumped over, over the, the fence).

Another way to process data is through dummy coding. Features might be categorical or continuous numerical. Categorical features can be coded as integers. For example, a country feature [“US”, “Europe”, “Asia”] might be coded as [0, 1, 2]. However, some models might interpret integer representations as ordered and continuous features, instead of categorical. Here, so called dummy coding could be used, where a categorical feature is represented by one binary feature for each value of the categorical feature. For example, the country feature mentioned above might be transformed into US [0, 1], Europe [0,1] and Asia [0, 1]. (scikit-learn, 2019a)

3.6 Training the model

When the features of the observations have been processed into formats which the ML algorithm can handle, it is ready for training. During training, the ML algorithm is provided with the data. The aim of the training is to reach a state where the ML model can generalize on future data and not just on observations used for training. (Amazon Web Services, 2016)

It is important to be able to measure the performance of a trained model. When evaluating the quality of a model’s predictions, the model is generally run on previously unseen data, but where the target answer is known beforehand. Because of this requirement of data for evaluation, some of the initial sample data cannot be used in the training of the model. If the model would be evaluated based on the same data as the model was trained on, the model would be rewarded for “remembering” the training data instead of making generalized predictions from it. (Amazon Web Services, 2016)

Depending on how much sample data is available, there are different ways to split the sample data. If there are big quantities of sample data, it is recommended to provide training data by holding out a certain percentage of the data completely from training, i.e. creating a hold-out set. On the other hand, if the original data source is small, cross-validation is recommended since it maximizes the amount of data used for training. (Mathworks Inc, 2019b)

K-fold cross-validation is when all the sample data is split into k subsets, so called folds. The subsets are held out one at a time and the model is trained and fitted on the other subsets. The model is then tested and evaluated on the hold-out set. The accuracy is saved and the model rejected. This procedure is repeated k times for each subset. The accuracy of the model is the average of the accuracy of each fold. (scikit-learn, 2019a)

A simpler version of this is a procedure called train/test split, where all the data is not used for training. Train/test split is when the entire dataset is split into two subsets: one for training and one for testing (hold-out set) (scikit-learn, 2019a). According to Amazon Web Services (2016), it is common to use 70-80% of the data for training and 20-30% for testing. The split can be made in order (where the first subset of the original data is for training, and second subset is for testing) or randomly (where the order of the original data is not considered). The accuracy has high variance since changing the observations used for training, will often change testing performance (Markham, 2015). However, using train/test split makes it relatively easy to receive further evaluation metrics such as a confusion matrix, which is further described under confusion matrix later in this chapter.

K-fold cross-validation gives a more reliable accuracy with lower variance than train/test split. Finally, a k-fold cross-validation with a hold-out set gives the most reliable performance estimate, as this hold-out set has never been exposed to the algorithm before and works as a double-check to confirm generalizability (Google developers, 2019d).

There are different ways to perform k-fold cross-validation. Stratified k-fold cross-validation is a k-fold cross-validation where the folds are stratified, meaning that each fold contains about the same percentage of observations from each label class. This is illustrated in figure 3.3. Stratified folds are recommended when there is an imbalance in the label class distribution (scikit-learn, 2019a). Amazon (2016) points out that having the same distribution in the test dataset as in the training dataset, as stratified folds aims at, is important because otherwise the model won't learn information regarding all classes which it needs to predict. A random split strategy could also fix this.

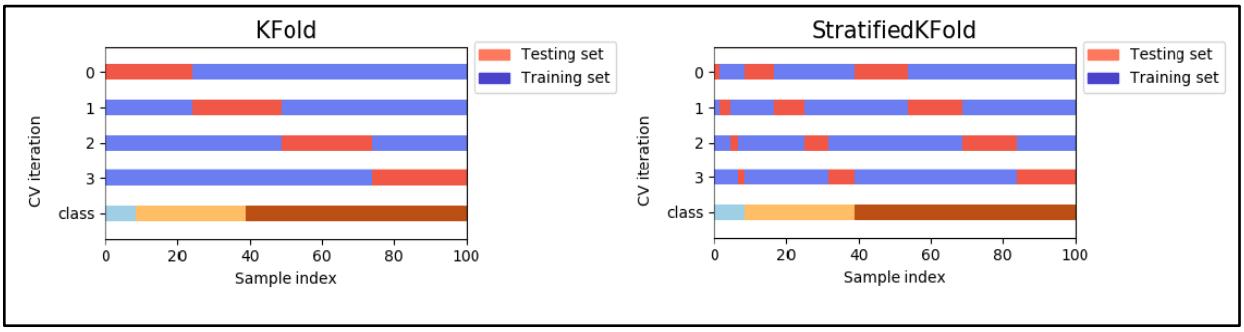

Figure 3.3. K-fold cross-validation and stratified k-fold cross-validation split the dataset differently. One cross-validation (CV) iteration corresponds to one row. Adopted from scikit-learn (2019a)

It is important that the test and training dataset distributions are similar to the distribution of new, unseen data that the ML model is going to be making predictions for in the future (Amazon Web Services, 2016).

To summarize, the evaluation of the model can be split into four different model evaluation procedures:

1. Training and testing on the same data
2. Train/test split
3. K-fold cross-validation
4. K-fold cross-validation with hold-out set

3.7 Evaluating the model

Evaluation of ML models is done using different methods depending on the type of ML task (Amazon Web Services, 2016). Depending on the problem, different metrics are more relevant than others. Following common metrics for evaluating a classification task exist (Amazon Web Services, 2016; Powers, 2007; Sokolova and Lapalme, 2009):

- Confusion matrix
- Accuracy (ACC)
- Precision
- Recall
- F1-measures
- Area under the ROC curve (AUC) - only for binary classification
- Specificity

These metrics mentioned can be used for both multi-class as well as binary classification, except for ROC and AUC. ROC and AUC has no well-developed way of being applied to a multi-class classification task yet, but only applicable for a binary classification task (Sokolova and Lapalme, 2009). In this following section, the metrics will be explained further for a binary classification problem.

A confusion matrix visualizes the performance of an ML algorithm. Predicted classes are represented in each column and the actual classes are represented in the rows, see figure 3.4. The number of true negatives (TN) or correct rejections, true positives (TP) or hits, false negatives (FN) or type II errors, and false positives (FP) or type I errors, are shown in the matrix. It makes it possible to calculate other metrics found below. (Amazon Web Services, 2016)

	Predicted: FALSE	Predicted: TRUE
Actual: FALSE	True negative (TN)	False positive (FP)
Actual: TRUE	False negative (FN)	True positive (TP)

Figure 3.4. A confusion matrix for a binary classification problem.

Accuracy (ACC) is the fraction of correct predictions out of all predictions. It is useful when the label classes are well balanced and not when the dataset consist of a majority of one class. The predictions are compared to the actual answers to give the accuracy of the model (Amazon Web Services, 2016):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Null accuracy is the accuracy that the model would get if it classified all observations as the most occurring class (Markham, 2018). It serves as a baseline accuracy which is useful if the dataset is imbalanced. In the binary classification case, if the dataset for example would have 99 % of the observations belong to one class, the model would get an accuracy of 99 % if predicting that class every time. Other metrics such as prediction and recall mentioned below takes this issue into account, but where precision might still be biased by imbalance (scikit-learn, 2019a).

Precision is a measure of how often a predicted positive value is correct. Precision is also called positive predictive value (PPV) and is defined as the number of TP over the number of TP plus FP (Amazon Web Services, 2016):

$$PPV = \frac{TP}{TP + FP}$$

Recall is the measure of how many positive observations that are predicted as positive among all observations that are actually true. Recall is also called true positive rate (TPR) and is defined as the number of TP over the number of TP plus FN (Amazon Web Services, 2016):

$$TPR = \frac{TP}{TP\ +\ FN}$$

F1 is the harmonic mean of recall and precision (Amazon Web Services, 2016):

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV\ +\ TPR}$$

Neither precision, recall nor F1 take the TN into account. **Specificity** take the TN into account and is also called true negative rate (TNR). Specificity is defined as the number of TN over the number of TN plus FP (Powers, 2007):

$$TNR = \frac{TN}{TN\ +\ FP}$$

Depending on the application and requirements, recall, precision and specificity are not evenly important. For example, having a very high recall and moderate precision is wanted for cancer detection when the focus is on minimizing FNs. The reason is that the most important requirement is to not miss any cancer patient, while it is not as bad to classify a non-cancer case as cancer since it will be examined further. The cancer case that is missed will however not be investigated further and can have devastating consequences. (Amazon Web Services, 2016; Drakos, 2018) As well, having a high specificity in cancer detection and hence correctly reject a non-cancer case as healthy, leads to certainty that a positive result actually is a cancer case (Drakos, 2018).

In another example, having a very high precision and a moderate recall is wanted for email spam classification when the focus is on minimizing FPs. If an email is misclassified as spam, an important email will be lost while if an email is misclassified as not spam, the receiver can delete the email manually. (Amazon Web Services, 2016; Drakos, 2018)

Receiver operating characteristic (ROC) is a curve that visualize how the binary classification model performs with all possible classification thresholds. A classification model returns a probability of a prediction. A classification threshold can range from 0 to 1, meaning a prediction score above the threshold for an observation will be classified as positive and below as negative. By default, many classifiers will have a threshold of 0.5. The ROC curve is created by plotting the TPR versus the false positive rate (FPR) at different thresholds. It is useful when there are two classes in the label and a high-class

imbalance, as opposed to accuracy which is not recommended in case of class imbalance. AUC is the area under the ROC curve and measures the ability to predict a higher score for positives compared to negatives. Hence, it is independent of the threshold. (Amazon Web Services, 2016)

3.8 Improving the model

A model needs to be improved when the performance is not good enough. A model can perform poorly due to problems with the model's fit to the data. If the model is underfitting, it does not show the relationship between the input examples and the target values. Underfitting might be a result of an oversimplification of the model, where the used features are not sufficient to express the full target (Amazon Web Services, 2016). Overfitting on the other hand is when the model matches the training data to a fault, thus performing worse on generalized predictions on never before seen data (Google developers, 2019a). Figure 3.5 visualizes an example of under- and overfitting compared to a balanced model.

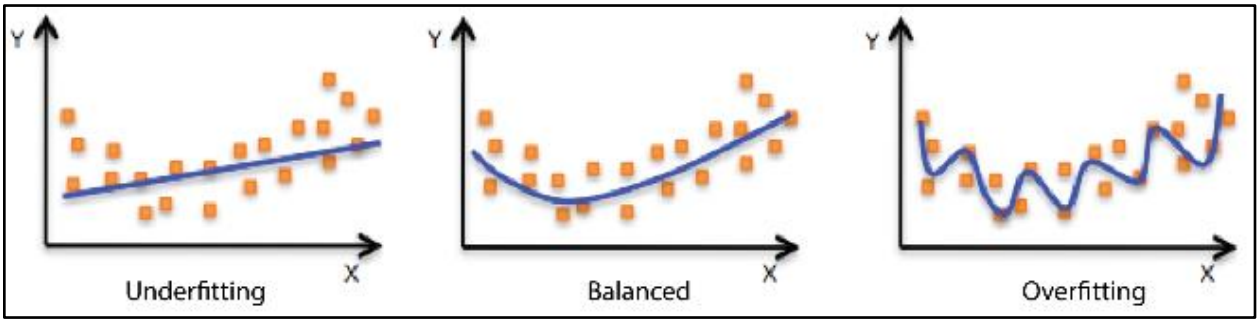

Figure 3.5. Example illustrating a model underfitting and overfitting to training data (Amazon Web Services, 2016).

Briscoe and Feldman (2011) describes a model which overfits as being trained to fit the training data so well that it does not generalize on new data. Generalization describes how well the model performs on new, unseen data and can be shown by comparing train and test accuracy. A model is overfitted when the training accuracy is much higher than the test accuracy. A model is underfitted when both the test and train accuracies are low. The variance increases and the bias decreases when model complexity increases, see figure 3.6.

Amazon (2016) lists the following actions to address imbalance due to poor fitting.

If the model is underfitting to training data:

(1) Add more features and change feature processing to make the features contain other information

(2) Decrease regularization.

If the model is overfitting to training data:

(1) Reduce the number of features used (feature selection)

(2) Increase regularization.

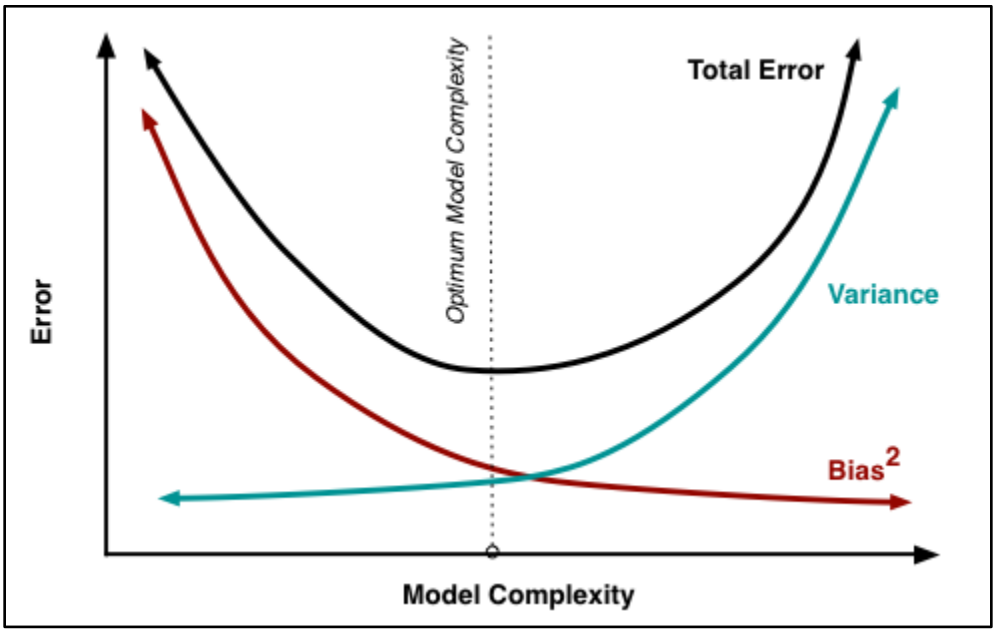

Figure 3.6. Total error of a model connected to model complexity (Fortmann-Roe, 2012).

Accuracy is, as mentioned in section 3.7, a common metric for a binary classification task. If the accuracy is not satisfying, a number of actions can be taken to improve it. Amazon (2016) mentions four areas for improving the accuracy of a model's predictions: (1) Collect data - Increase the number of training examples, (2) Increase the number of passes in training, (3) Feature processing - add more features and improve feature processing, and (4) Model parameter tuning: alter the training parameters used by the learning algorithm.

Cross-validation is often used for the three improvement areas (2), (3) and (4). With cross-validation, the space of hyper parameters can be traversed in search, to find the hyper parameters resulting in the best accuracy. Depending on computational requirements, the search can be performed to varying degrees of exhaustion: all possible values, a given subset of values, or a random subset combining different hyper parameter values. Some hyper parameters contribute a lot to the computational power and complexity of the model, such as hyper parameters influencing the size of the model.

Hyper parameters of big influence are recommended to be tuned, while others can be kept at default values. (scikit-learn, 2019a)

To improve model accuracy, one hyper parameter to influence is the number of passes over the data in training. For a very large dataset, one pass could be enough while for a smaller dataset, it is recommended to try to increase the number of passes. The more passes the closer the algorithm can fit the data, but the more the gain per pass decreases. (Amazon Web Services, 2016)

Cross-validation can also be used for feature selection, where features could be added or removed depending on their relevance in predicting the label. A cross-validation with and without a specific feature could be done, the performances compared, and the model with the best performance chosen. Tree estimators such as decision trees and random forests do feature selection automatically when it is fitted to the training data. Feature importance is computed which can be used to remove less important features. (scikit-learn, 2019a)

When the models have been improved, their performances should be compared in order to choose the most promising model. Kohavi (1995) recommends to use a stratified 10-fold cross-validation method for model selection, i.e. for choosing the model with the most promising performance. When doing so, the variance and bias of the estimation of the accuracy are both low. Further, Kohavi does not recommended to use a hold-out set as the dataset is usually smaller than wanted and needed for using a hold-out set. Creating a hold-out set uses the finite data in an inefficient way.

3.9 Using and retraining the model

When the model is trained, it can be used to make predictions on never-before-seen data. The reliability of the model is dependent on the distribution of the data being of a similar distribution as the training data (Amazon Web Services, 2016). If the distribution of the data source changes over time, the model will need to be retrained on data of the new distribution. This change of distribution is called concept drift (Ditzler et al., 2015) and creates difficulties when predictions get less accurate. Retraining on data can be done continuously or periodically, with varying amount of overhead for monitoring (Amazon Web Services, 2016).

4 Theoretical findings on country of origin

The theoretical findings on CoO from the literature review is found in this chapter. It serves as a frame of reference and starts with defining criteria for CoO in section 4.1. It is followed by regulations on CoO in EU in section 4.2 and in the US in section 4.3.

4.1 Criteria for CoO

All internationally traded goods must have a CoO when declared to customs for import. Attributing one CoO to each product becomes complex in global trade where a product can be obtained and processed in not only one country, but often in several countries before it is ready for sale. Here, rules of origin (ROO) are needed, which contain the criteria required to decide CoO for the different cases: *wholly obtained* and *substantial transformation*. (World Trade Organization, n.d.) Wholly obtained goods are usually goods which are wholly produced or obtained in one country. These are products such as raw materials, animals and harvest. The status of wholly obtained can be determined by a comprehensive list of products or by a definition. Where two or more countries have taken part in the production of the good, the origin is determined by the substantial transformation criterion of ROO. This means that CoO is determined as the country where the last substantial manufacturing has been performed according to World Customs Organization (WCO) and World Trade Organization (WTO) (World Customs Organization, n.d.; World Trade Organization, n.d.).

WTO and WCO have important roles in creating, administering and simplifying the enforcement of ROO in global trade (World Trade Organization, 2019). WTO is an international organization dealing with the global rules of trade, providing assurance and stability for producers and consumers (World Trade Organization, n.d.). WTO administer agreements which are negotiated, signed and ratified by the majority of the world's trading countries. The agreements are contracts that ensure WTO members' trade rights as well as transparent, fair and predictable trade policies. WTO also collaborates with WCO (World Customs Organization, n.d.). WCO represents 183 customs administrations, covering 98 % of world trade and is recognized as being the global customs expert (World Customs Organization, n.d.).

In order to decide what substantial transformation involves, there are several different methods to use: Rules based on (1) the change in tariff classification that is the change of Harmonized System (HS) code, or (2) the value added to the product, or (3) a list of specific manufacturing and processing operations. The rules can be combined or applied

independently. (World Customs Organization, n.d.; World Trade Organization, n.d.) Different governments have different practices when it comes to ROO. However, the requirement of substantial transformation is recognized among WTO members as necessary for determining CoO, but what it involves may differ. (World Trade Organization, n.d.)

The HS code is a multipurpose international product nomenclature developed by WCO, which serves as a universal economic classification of goods. It is used by governments and international organizations, resulting in HS codes being used for classification by over 98% of international traded goods. (World Customs Organization, n.d.)

In order to make ROO objective, understandable and predictable, WTO Agreement on Rules of Origin established the Harmonization Work Programme (HWP) in 1995 (World Trade Organization, n.d., n.d.). In the HWP, it was agreed that for each HS code, a rule should be established that defines last substantial transformation. If this rule is fulfilled, the product will get the general CoO of where it underwent the last, substantial transformation. Since the programme is not yet completed, general ROO differs among countries. The importing country decides which rules apply upon import. (Axis Communications, 2018b; World Customs Organization, n.d.)

4.2 EU regulation on CoO

The criteria for obtaining origin in the EU is stated in the Union Customs Code (UCC), which provides a framework for customs rules and procedures (Axis Communications, 2018b). When production of goods involves more than one country, the European parliament and the Council of the European Union (European Parliament, Council of the European Union, 2013, p. 31, article 60) states that:

> "[the origin will be] the country or territory where they underwent their last, substantial, economically-justified processing or working, in an undertaking equipped for that purpose, resulting in the manufacture of a new product or representing an important stage of manufacture."

There are different rules for different types of goods, listed in the UCC. The rules can be either of the three criteria for substantial transformation mentioned in section 4.1. For goods not mentioned in the UCC, the last substantial transformation is considered to have taken place where the major part of materials originate, on the basis of the value of

materials (Axis Communications, 2018b; European Parliament, Council of the European Union, 2013).

4.3 US regulation on CoO

The US government prefer to procure domestic products over foreign, which has been implemented through laws and regulations (Deloitte, 2017). The Buy American Act (BAA) of 1933 is a federal legislation that requires the US government to favor domestic end products. The BAA is applied for federal procurement unless the Trade Agreements Act (TAA) of 1979 applies. TAA implements trade agreements which guarantees that compliant foreign products will be treated as domestic products, while non-compliant will be prohibited. TAA works as a waiver of BAA (Deloitte, 2017). TAA states that an article is a product of a country only if either of the following applies (Axis Communications, 2018b, p. 4):

> "I. It is wholly the growth, product or manufacture of that country or instrumentality, or
>
> II. In the case of an article which consists in whole or in part of materials from another country or instrumentality, it has been substantially transformed into a new and different article of commerce with a name, character or use distinct from that of the article or articles from which it was so transformed."

All EU member countries are TAA-compliant, while many countries in Asia are not, including China, India and Thailand (PwC, 2018). If products contain components which are non-TAA-compliant, CoO must be determined by the US Customs and Border Protection (CBP) Regulations. CBP is responsible for border control and hence implementing US regulations regarding trade and customs, including CoO requirements. According to the CBP, CoO is defined as (Axis Communications, 2018b, pp. 3–4):

> "[...] the country of manufacture, production or growth of any article of foreign origin entering the United States. Further work or material added to an article in another country must effect a substantial transformation in order to render such other country the 'country of origin' within the meaning of the marking laws and regulations."

When determining CoO, the most critical part for CBP is to decide where, and if, a substantial transformation was made. Factors that CBP considers are the number of components in the product, the components' CoO, the operations that the product undergoes in a country, and if the operations transform the product, for example giving it

a new name, character or use. CBP determines the CoO case-by-case by analyzing legal and technical aspects, production and design, where previous cases serve as rulings. Previous cases have shown that assembly operations are not considered to substantially transform an article. Other cases have shown that for software, the CoO is determined to be the country in which the code and software executable files were created, source code programmed and testing and validation carried out. Software loading into hardware is considered to substantially transform a product, where both the place of development of the software as well as loading is taken into account. (Axis Communications, 2018b)

5 Empirical findings

This chapter presents the findings from the empirical data collection at Axis, mainly from conducted interviews and internal documents. When no references are provided, it was perceived as general knowledge at Axis and hence no specific reference deemed needed. In section 5.1, a description of the current situation of handling the CoO question at Axis is described, followed by Axis requirements for the desired solution in section 5.2.

5.1 Description of current situation at Axis

5.1.1 Axis supply chain networks

As shown in figure 5.1, Axis supply chain consists of several actors at different stages (Lindroth, 2019). The components constituting a product are sourced from component suppliers in the US, Europe and Asia. They are in most cases second tier suppliers, but Axis might also source directly from component suppliers. The components are then put together at one of six contract manufacturers, also called Electronic manufacturing services (EMS). The EMSs are located in Thailand, Poland and Mexico.

The hardware, with or without the software, is then shipped to a configuration and logistics center (CLC), which is the only part of the supply chain that Axis owns. The lead time from suppliers to Axis might be 1-26 weeks but could be up to 52 weeks. At the CLCs, the products are processed from the delivered state into a final sales unit (SU). The processing can include for example assembly, loading of software, testing and packing. Axis has six CLCs which are located in Europe, the US and Thailand. (Lindroth, 2019)

Because of the possible different setups of the upstream supply chain, the definition of the CoO of a product is not unambiguous. After the CLC, the SU is shipped to the customer, which can be a distributor or possibly also a reseller or system integrator. The reseller then sells the products and solutions to end users. Axis has a lead time of 6-10 days to its customers. Axis also has return merchandise authorization (RMA) partners that handles the return flow.

Axis want their product portfolio available at any time which is accomplished by having a flexible supply chain. A flexible supply chain is achieved through: (1) Component suppliers, EMSs and CLCs might be changed for certain products or orders, (2) material might be transferred between CLCs, and (3) the level of completeness which a product is

purchased at might be changed. This flexibility can cause the CoO of products to change repeatedly. (Hjelmström, 2019)

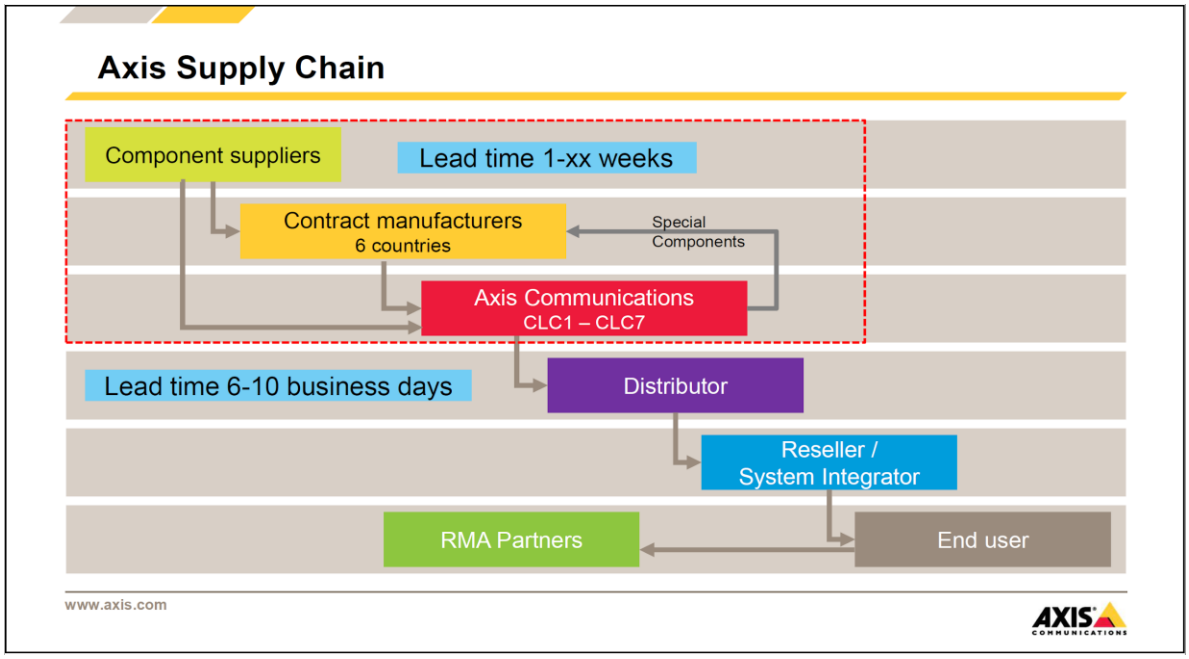

Figure 5.1. Axis supply chain with setups, actors and lead times. (Lindroth, 2019)

5.1.2 Product structure

Axis product portfolio consists of both advanced and more elementary products. Examples of advanced products are network cameras, video encoders and recorders, whereas examples of elementary products are accessories for the advanced products. (Hjelmström, 2019; Lilja Ivarsson and Kos-Hansen, 2019)

An advanced product and its components can be purchased by Axis at different product levels: unit assembly (UA), product unit (PU) and SU. A UA consists of the hardware for an advanced product assembled with all components, excluding the software. A PU is a UA loaded with software. The final product ready for sale is called SU. In the case of an advanced product, an SU is a PU packaged with accessories and other product documentation. In the case of more elementary products an SU is merely a packaged product ready for sale. The product levels are illustrated in figure 5.2. (Gard, 2019)

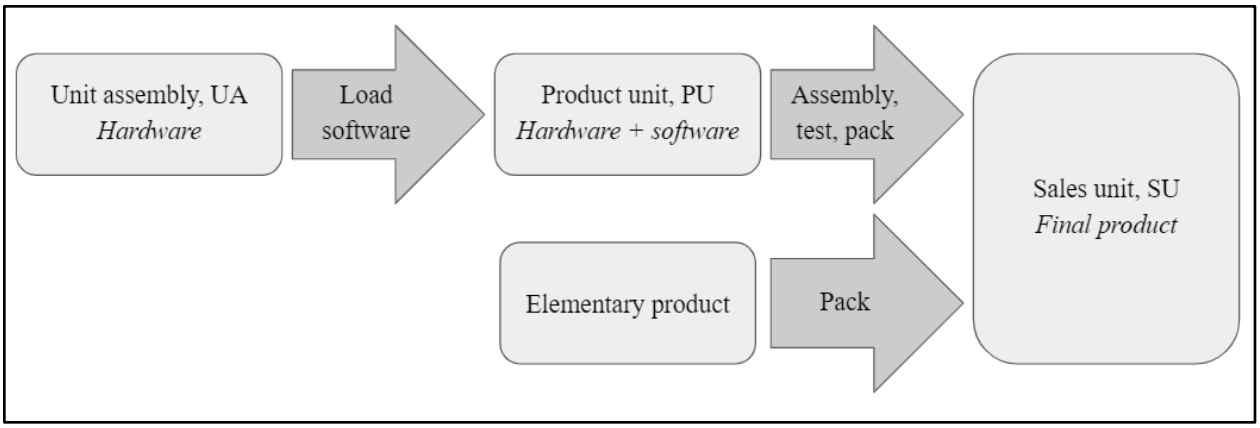

Figure 5.2. Levels of product structure in Axis products.

All articles sold or processed by Axis has a bill of material (BoM), which registers information about all components making up the article. Only components used in the processing operation within Axis are registered. The BoM of an SU can vary from multiple components to only the SU itself, depending on the level of processing performed by Axis. The product structure of an SU is derived from the BoM, by identifying the state of the main component comprising the article. The product structure can be changed because of Axis flexible supply chain. Therefore, a product can be purchased as an SU directly to the CLC or be processed at the CLC from a UA, PU or components. The possible product structures of an SU are illustrated in table 5.1. (Hjelmström, 2019; Lilja Ivarsson and Kos-Hansen, 2019)

Table 5.1. Different alternatives to product structures of how a product is purchased to a CLC and how the SU is categorized when going out of the CLC.

	1	2	3	4
Product type out of the CLC	purchased SU	manufactured SU (from UA)	manufactured SU (from PU)	manufactured SU (from components)
Information in the BoM	SU	List of purchased components including a UA	List of purchased components including a PU	List of purchased components excluding UA and PU

5.1.3 Rules of origin in EU

In order to determine the CoO of an SU, Axis must determine where the products underwent their last, substantial transformation. The processing that occurs at CLCs in the EU does not change the tariff classification, but could still be considered an important stage of manufacture, thus still affecting the CoO. (Hjelmström, 2019)

The CoO of a product depends on its product structure. In the case of products sold by Axis in Europe, there are different ways to define CoO depending on the type of product structure (Hjelmström, 2019):

Purchased product: Since no value is added at an Axis CLC if the product is purchased as a final SU, the CoO is set by the supplier. Examples of products bought in its final state are explosion-protected cameras, as well as elementary products.

Manufactured product with PU as main component: When the main component is purchased as PU, the CoO becomes the country of the supplier, since the packaging and configuration at the CLC is not considered as last substantial transformation.

Manufactured product with UA as main component: If the main component is bought as a UA, the processing and work including uploading of software that occurs at the CLC is considered substantial enough. Therefore, Axis can change the CoO of the SU to the European country where the CLC is located.

Manufactured product with elementary main component: If the most expensive, purchased component is elementary (not a UA or PU), the CoO is set as the country of the component. Examples of this type of manufactured products are elementary products produced in-house.

5.1.4 CoO for products sold on the US market

The US and EU have different rules for deciding CoO, with different requirements for substantial transformation. This implies that Axis has two different supply chain setups for the US market where these rules differ. The setups, together with their effect on CoO for products with different product structures, are shown in figure 5.3. In the first setup, EU rules apply whereas US rules apply in the second setup.

Axis performs different value-adding processes at the CLCs where manufacturing a SU from a UA increases the value the most out of all processes. According to EU rules, this

process of loading software, testing and packing is enough for changing the CoO to the country of the CLC in EU. In contrast, US rules state that this value-adding is not enough to change the CoO to the US, where the CLC is located. Instead, the US rules consider the CoO to be the country of the supplier of the UA, as shown in the second setup in figure 5.3. These differences in the rules imply that the same article can have different CoO depending on the supply chain setup. (Hjelmström, 2019)

However, US rules only considers the CoO which the previous actor of the supply chain defined, before the product entered the US. This is shown in previous CBP rulings, which Axis legal department considers. Therefore, when the UA is processed into a SU at a CLC in EU in the first setup in figure 5.3, the CoO becomes the European country. When the SU is later shipped to the CLC in the US, US rules apply and defines the CoO to remain the European. (Hjelmström, 2019)

Products with the remaining product structures (elementary product, PU or SU as main component) are handled the same in the EU and the US, regardless of the supply chain setup. The manufacturing of SUs from those products is not considered to be enough to meet the substantial transformation requirement and hence the CoO is defined by the supplier. (Hjelmström, 2019)

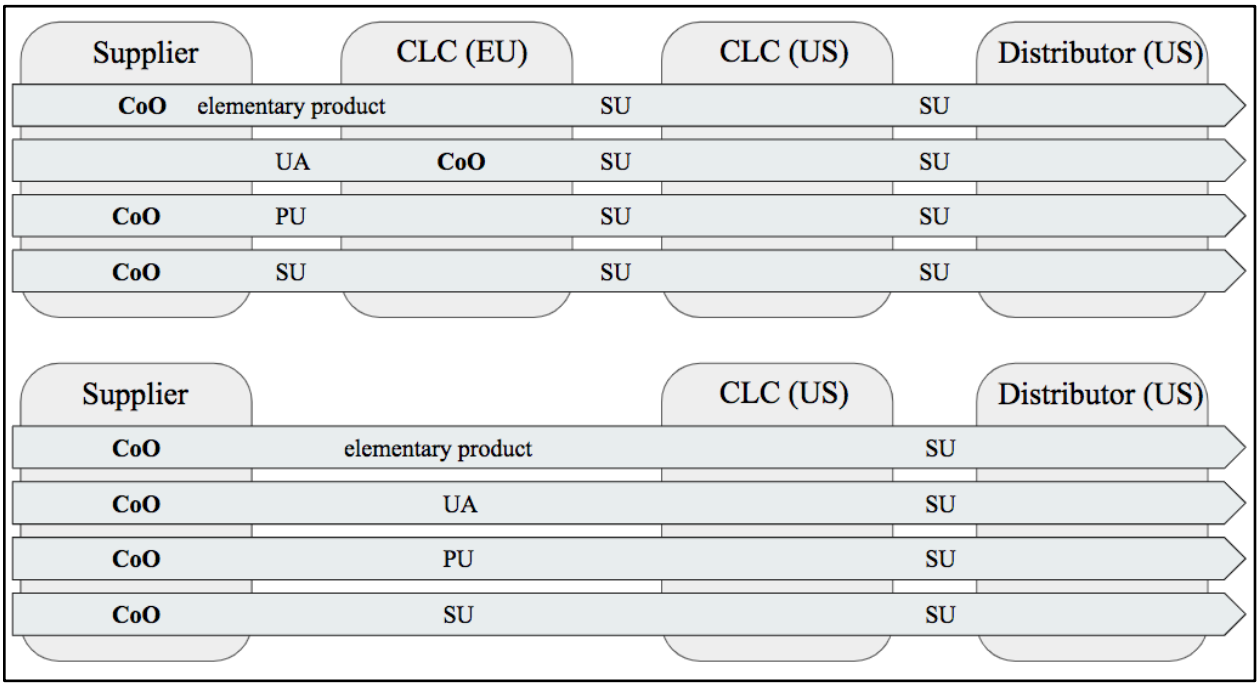

Figure 5.3. Axis two supply chain setups for the US market resulting in different CoO for different articles.

5.1.5 ERP-data relevant for CoO

The CoO-data, among other product related information, is managed and retrievable from Axis ERP-system IFS Applications (IFS). IFS has several views with CoO-related data which are summarized in table 5.2 and described in more detail below (Hjelmström, 2019).

Table 5.2. An overview of which CoO-related information that can be found in which view in IFS.

View in IFS	**Data**
Inventory parts	*Country of origin*: CoO of an article (purchased or manufactured)
Supplier for purchased parts	*Primary supplier*: first CLC for an article (purchased or manufactured) for a market. *Secondary supplier* (if using): second CLC for an article (purchased or manufactured) for a market. *Country of origin*: CoO of a purchased unit
Part cost	Bill of Material for an article containing information about its components: ● Article number ● Article description ● Unit cost ● How the component is attained (purchased or manufactured)
Inventory part availability planning	*Primary supplier purchase*: The first external supplier of an article
Supplier	*Country*: registered address of a supplier or CLC

IFS/Inventory parts

The data field *Country of origin* in IFS/Inventory parts exists for manufactured SUs as well as for purchased components, UAs, PUs and SUs. It is the CoO that gets printed on documents at production and packing, as well as being the current CoO for the product. Hence, this is the data field which needs to be validated.

IFS/Supplier for purchased parts

In IFS/Supplier for purchased parts, the CLC which does the final assembly can be found through knowing the SU article number and the market (US for this study). First, a primary supplier for the market is found. A secondary supplier can be found as well if it exists, i.e. if the supply chain configuration is according to setup 1 in figure 5.3. If a secondary supplier does not exist, the supply chain configuration is according to setup 2 in figure 5.3. Hence, the CLC which does the final assembly, which primarily influences the CoO of the SU can be found in IFS/Supplier for purchased parts. Furthermore, the data field *Country of origin* of an SU, PU, UA or component at a CLC which does final assembly can be found in IFS/Supplier for purchased parts.

IFS/Part cost

After the CLC which does final assembly for the SU is identified, the product structure of the SU at that CLC can be found through analyzing the BoM, found in IFS/Part cost. The possible product structures are found in section 5.1.2. If IFS/Part cost only contains the SU, it is purchased as the SU. If the IFS/Part cost contains several article numbers, the purchased component with the highest unit cost is identified since it is influencing the CoO and is of interest. Here, it is also identified if the most expensive, purchased component is a UA, PU or component.

IFS/Inventory Part Availability Planning

The primary external supplier for an SU at a CLC is identified in IFS/Inventory Part Availability Planning.

IFS/Supplier

The manufacturing address and hence *Country* of suppliers as well as of CLCs is found in IFS/Supplier.

5.1.6 Processes of managing CoO-data in IFS

Data is put in IFS in different data fields and views by different departments at Axis. The departments Sourcing, Purchasing and Product Data Group (PDG) are involved in adding CoO-related data in IFS. Data that directly affects the CoO is managed in the views IFS/Supplier, IFS/Supplier for purchase parts and IFS/Inventory parts in IFS. Sourcing and Purchasing manage data in input data fields whereas PDG is deriving what the CoO in IFS/Inventory parts should be from the input data fields. Table 5.3 shows which departments that manages which CoO-data in IFS. The table is explained further in the coming sections. (Hjelmström, 2019; Olander, 2019)

Table 5.3. Responsibilities of CoO-data in different views in IFS.

Data field in IFS: **Product type:**	*Country* in **Supplier**	*Country of origin* in **Supplier for purchased parts**	*Country of origin* in **Inventory parts**
Elementary product	Sourcing	Purchasing	Uncertain
UA	Sourcing	Purchasing	Uncertain
PU	Sourcing	Purchasing	Uncertain
SU	Sourcing	Purchasing	Sourcing (CLC1), else PDG

***Country* in IFS/Supplier**

The data field *Country* in IFS/Supplier is an input field where Sourcing adds the country of registration of the suppliers. The information is entered when a new supplier is introduced, or the information is changed. A supplier might have the production in one country and might be registered in another country. This is why the data field Country is not always reliable for the use of calculating CoO for a product, as the CoO depends on the country of production. However, this data exists to a large extent, making it a commonly used variable for deciding CoO. The supplier that is currently supplying a CLC for a specific product is called primary supplier. Different CLCs can have different primary suppliers for the same product. (Hjelmström, 2019)

***Country of origin* in IFS/Supplier for purchased parts**

The purchaser responsible for a product at a specific CLC adds CoO for the purchased product based on the article number and site in IFS/Supplier for purchase parts. The purchaser base this data on the country in which the supplier is located. (Olander, 2019)

***Country of origin* in IFS/Inventory parts**

For components, UAs and PUs, the data field *Country of origin* in IFS/Inventory parts is an original data field which Sourcing or Purchasing enter information in, but the responsibilities are not well defined here. The data field for SUs is a derived data field

with the current CoO, where the definition of CoO is based on original data fields in IFS. Sourcing adds *Country of origin* for the SU at CLC1 when a new product is introduced. At other CLCs and in later parts of the product life cycle, PDG is responsible for entering the current CoO. Hence, the responsibility for this data field is shared.

Purchasing and Sourcing adds information in the mentioned original data fields when a new product is launched. They are also responsible for updating this information throughout the product life cycle when changes occur. Derived and dependent data fields, such as *Country of origin* in IFS/Inventory parts for SUs, should be updated when an input data field is updated.

PDG determines and adds *Country of origin* in IFS/Inventory parts for SUs (Hjelmström, 2019). The CoO-data is based on information from Sourcing and Purchasing. PDG determines the current CoO based on multiple considerations: which purchased component in the SU that is the most expensive, what product level the component is purchased at, where the CLC for final assembly is located and hence which ROO that should be applied. The CoO of the SU is determined through either of three methods, depending on which information is available. The methods are written in order of preference:

1. *Country of origin* of the most expensive, purchased component found in IFS/Supplier for purchased parts if it exists, else
2. *Country* of the supplier of the most expensive, purchased component found in IFS/Supplier if it exists, else
3. CoO from XML-files from Product Test Group (PTG), who are responsible for testing of an article.

The process for deriving CoOs for SUs could be time consuming, involving up to six different views in IFS and multiple steps for collecting needed data, which the Axis employees have to handle manually (Hjelmström, 2019).

The CoO-data in the first two steps are sometimes lacking which is why the XML-files are checked as a last option. The XML-files contain, among other information, the label with the CoO for the specific unit of the product. There is no guarantee that the CoO in the XML-file reflects the current supply chain setup, but can also come from returns in the reverse supply chain (Olander, 2019).

5.1.7 Challenges of using CoO-data in IFS

One challenge of the setup of CoO-related data in IFS, is that the data is used by different departments at different times for different uses. The final CoO is used by departments such as sales and customs. Sales are interested in the CoO of the product when making a sale, where the SU might not even be manufactured yet. Sales is the department which is interested in guaranteeing TAA-compliance and this CoO is relevant for this study. Customs on the other hand, are interested in the CoO of products being shipped and delivered. CoO-data in IFS is the promised CoO when an order is placed and not when delivery occurs. This time difference between order and shipping might cause problems, since the supply chain setup might change and the inventory management is lacking.

First-In-First-Out should be applied to inventory, meaning that the components and products should be sent out in the order that they arrived (Lilja Ivarsson and Kos-Hansen, 2019). This is however not always applied in practice. Reasons for this could be that older components are sent to a CLC from other CLCs, leading to a risk of mixing products with different CoOs. There might be inventory in stock from previous suppliers which results in that the actual CoO might not match the CoO in IFS, promised to customers. When the CoO in IFS and in the XML-file handled by PTG does not match, the shipment gets stopped at the CLC and the CoO must be controlled. This could be very time consuming for Axis customs department. (Lilja Ivarsson and Kos-Hansen, 2019)

Input data fields should be updated when internal changes appear, such as revision updates, change of EMS or CLC, change of product level for purchasing and transfers that occur when collecting material for products that are at the end of life (Olander, 2019). Derived and dependent data fields should be updated when an input data field is. However, this is not the case and it is hard to tell which data field contains the correct information as no changes are logged or tracked. Furthermore, the dependencies between input data fields are not connected and it is therefore not guaranteed that all relevant data fields are up to date. This may be the case when the data has only been added in one view, and at one CLC, or when it is updated later in the product life cycle and is only updated in one of the views. (Hjelmström, 2019)

As mentioned, the departments Sourcing, Purchasing and PDG are involved in adding CoO-related data in IFS. It is not always clearly established by whom and when data should be updated. There is a lack of processes and routines on how to handle new information and updates as well as lack of ownership of data. For example, PDG is not

notified when changes in the original data fields have occurred and hence the final CoO could be outdated. (Olander, 2019)

Another factor which is not supported by IFS is external changes of ROO. Accurate CoO-data might become inaccurate after changes of external rules, even though no changes in the supply chain has occurred. This change could affect several products sold at a market and changes have to be handled manually. (Lilja Ivarsson and Kos-Hansen, 2019)

5.2 Axis requirements for the desired solution

The following requirements were presented through interviews with Matilda Hjelmström (Supply Chain Development Manager) (2019). Axis is requesting a tool which can aid in validating CoO of products for the US market by presenting a list of products where the CoO might be faulty. The tool should use ML to predict what the CoO of a product should be and then compare the prediction to the CoO existing in the ERP-system. The output is not required to be fed back into the ERP-system, but is instead to be used manually by Axis employees. The prototype is intended to be used by Axis employees from the department of Sourcing and PDG. Input to the prediction is limited to information available in the ERP-system, information from other systems are not to be considered. Information from the ERP-system should be fetched once a day, and does not need to be continuously updated throughout the day.

The output of the tool is required to include five types of information: the prediction (correct/faulty) of the model, the probability of the prediction, the existing CoO from the system, the input values the prediction was based upon, and also the timestamp of when the values were fetched from the ERP-system. The latter three requirements are intended to increase the traceability of the predictions, in order for the employees not having to investigate why they got the result that they did, as well as increasing the usability of the tool.

The performance of the model will primarily be evaluated based on the accuracy of the predictions, but also on feasibility of implementation, taking into consideration factors such as ease of use for employees.

6 Applying machine learning to country of origin

Applying ML to a problem is often an experimental process of trial and error. This chapter will describe the process and how the workflow in section 3.3 was applied in the context of validating CoO at Axis. It will start with presenting the data that was available and collected in section 6.1, followed by how the problem was framed for using ML in section 6.2 and section 6.3 explains the data preparation and feature processing. Before the ML model training, improving and evaluation is presented in section 6.5, the tools and libraries used for this are described in section 6.4. There is no single ML algorithm that fits every problem and therefore promising algorithms, that seemed suitable for this CoO problem, were tested and are lastly presented in section 6.6.

6.1 Data collection and analysis

Data for the study was limited to articles on the US market (site = INC). An initial list of relevant articles provided by Axis was based on the master price list for the US market for the fall of 2018, which at that point contained 1501 article numbers of physical products. Although data about the products could be viewed in the ERP-system, it could not be retrieved or downloaded in a collected format. Instead, the needed data was provided through a business intelligence tool in combination with processed data from the ERP-system.

When analyzing the initial dataset, the data was distinguished between verified and unverified product data. Verified products are products with cleaned and authenticated data in the ERP-system, as part of establishing a list of TAA-compliance. Data of unverified products have not been cleaned and authenticated. It is therefore uncertain if this unverified data is correct or faulty. The correction and verification of data concentrated on products with high likelihood of being TAA-compliant. This causes the verified and unverified product data to partially cover products from different countries.

As can be seen in table 6.1, 807 out of the 1501 products are verified. The data associated with these products is likely to be correct. The remaining 694 products are unverified, and might therefore still have data which is incorrect or missing. The reliability of the existing data of the unverified products is estimated to be high even before verification (Ekström, 2019).

Table 6.1. Data for verified and unverified products. The four rightmost columns show the number of distinct countries present in the dataset for each data field, showing a divide in which countries are represented in the verified and unverified product data. The default value of - MISSING - is included as a possible value of country.

Labels	**Total number of products**	**Number of CoOs, product level** *[Inventory part]*	**Number of CoOs, component level** *[Inventory part]*	**Number of countries of suppliers** *[Supplier]*	**Number of CoOs, component level** *[Supplier for purchased parts]*
Un-verified	694	18	22	20	22
Verified	807	24	23	18	23
Total	**1501**	**28**	**30**	**22**	**30**

The CoO of products are not evenly spread across the possible countries. When looking at the current CoO of verified products in the ERP-system, products from 24 countries are represented. While the number of data points range from 199 (Czech Republic) to one (Romania among others), and only 6 of the 24 countries have more than 50 examples, this results in a tail of countries with few data entries, as can be seen in figure 6.1.

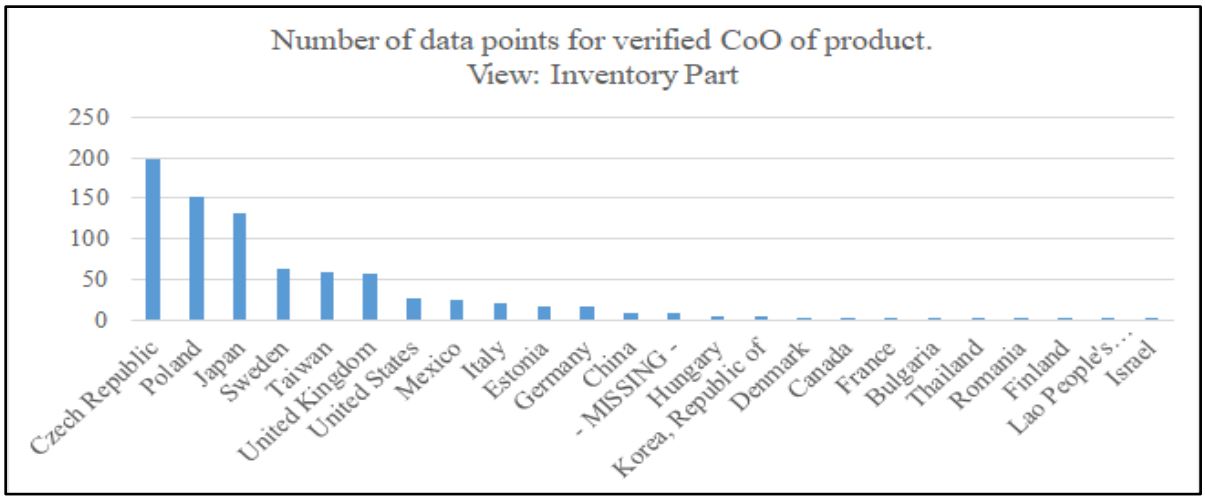

Figure 6.1. Number of data points per country for current CoO-data. Only verified product data is included.

If increasing the dataset for current CoO to include both verified and unverified products, products from 28 countries are represented. Though the dataset would be increased from 807 to 1501 data points, the same pattern of dispersion is present. Figure 6.2 shows that the number of products from each country range from 380 products (China) to one single product (Romania among others), where a majority of the countries still have less than 50 data entries. Similar patterns are present in the product data of the other CoO-related variables, as is illustrated in appendix A3 Countries in the original dataset.

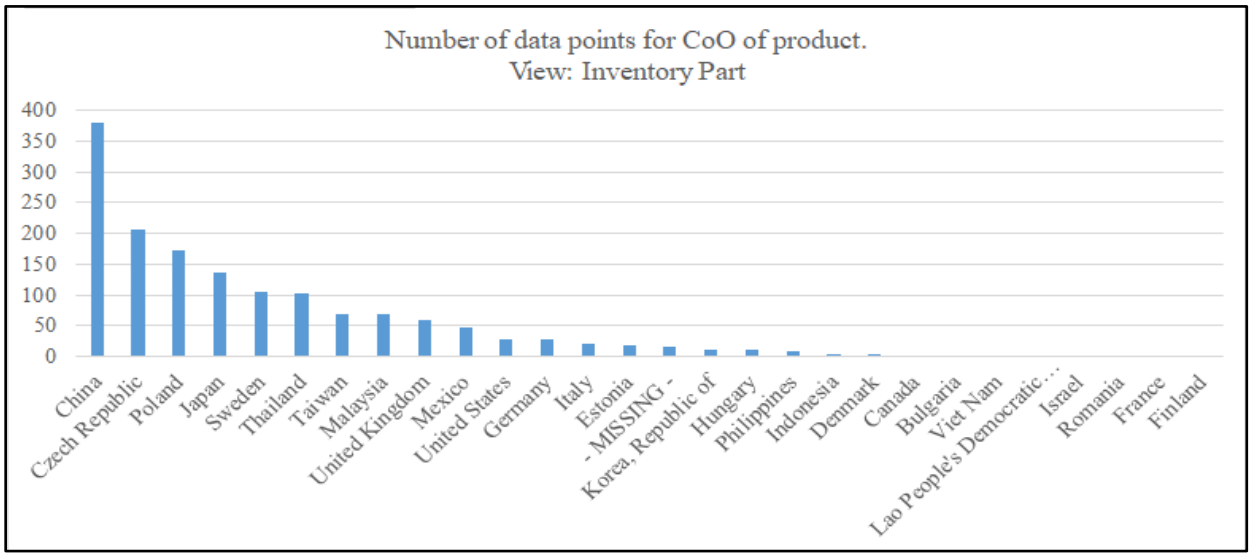

Figure 6.2. Number of data points per country for current CoO-data. Both verified and unverified product data is included.

The dataset includes examples of both supply chain setups used by Axis for the US market (see figure 5.3): 707 of the products have final assembly at a CLC in the EU, while the remaining 794 products have final assembly at a CLC in the US. 148 products have one or more feature values missing and hence are given the value "-MISSING-".

There is no data available of original faulty product data in the ERP-system, partly due to not saving examples while producing the TAA-compliant list, but also the ERP-system not tracking changes. Throughout the control and verification of product data, approximately 5% of the products had a faulty CoO (Ekström, 2019). As a result of the structure of the ERP-system, none of these examples were saved or can be recreated within the resources of this study. Faulty data could possibly be retrieved from the remaining, unverified data. Given the assumption that 5% of the remaining data is faulty, only about 35 true faulty examples could be generated from the remaining unverified data.

6.2 Problem framing of using ML

Axis requests a tool for predicting if the CoO-data in the ERP-system is correct or faulty, thus letting the tool solve a classification problem. The problem can be interpreted to use ML in two ways: (1) Use ML to predict the correct CoO of a product, then compare the predicted value with the current CoO in the ERP-system (IFS/ Inventory part), and (2) Use ML to predict if the CoO-related data in the ERP-system is of a correct format. The two interpretations are depicted in figure 6.3.

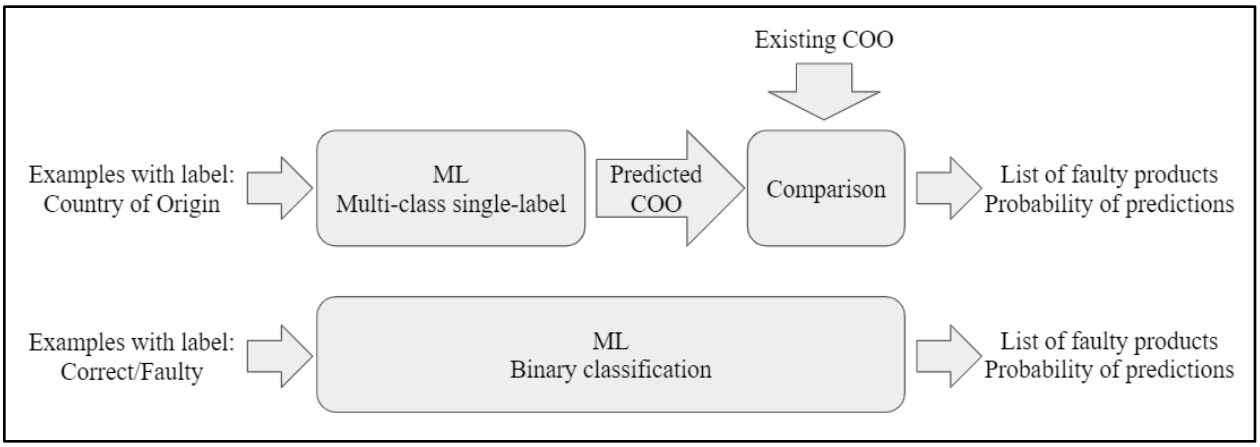

Figure 6.3. Image comparing potential methods of using ML to solve the problem. The upper solution consists of two steps: Using ML (multi-class single-label classification) to predict the CoO and then comparing the predicted CoO with the current CoO. The lower solution uses ML (binary classification) to directly predict if the current CoO and data is of a correct or faulty format.

The first way to interpret the problem aims at predicting the correct CoO of the product. Since a product only has one CoO and there are 27 possible countries present in the raw data for CoO of a product (IFS/Inventory parts), the problem would become a multi-class single-label classification problem with 27 classes, according to the flowchart in figure 3.2 in section 3.1. The output of the ML model would be a predicted CoO, which could then be compared to the existing CoO outside of the ML model. When training the model, the label of the input to the ML model would be the CoO existing in the ERP-system (IFS/Inventory parts).

For the second way of interpreting the problem, ML is used to directly predict if the existing CoO and related data is of a correct or faulty format. This still makes it a classification problem, but the model uses two classes (correct and faulty) instead of using multiple countries as classes, thus making it a binary classification problem. The

existing CoO (IFS/Inventory part) would then be used as a feature, instead of being used as a label as is done in multi-class classification.

The main advantage of using multi-class classification to solve the problem is increased usability and possibly meeting more of Axis requirements. By predicting a country, the model could be used both to verify an existing CoO, but also as an aid in defining a product's CoO in case the existing CoO is faulty. This would aid in solving a bigger part of the problem for Axis, since the tool then could create a list of products with faulty CoO as well as a prediction of the correct CoO. The drawback of using multi-class classification is the increase in complexity of the model. In order to train the model, sufficient amount of data is needed of all labels thus increasing the total amount of data needed. In comparison, binary classification only requires data for two classes, thus making it a significantly lower total amount of data needed. It is however different data needed for the two interpretations. For multi-class classification, there needs to be a sufficient amount of examples of products from all 27 countries. Synthesizing additional data for multi-class classification is difficult given the available CoO-data. Binary classification on the other hand needs both correct data in the form of labeled observations, as well as faulty data. In the absence of true faulty data, it is deemed to be possible to synthesize faulty data by changing CoO of correct examples.

Given the skewed distribution of available examples of current CoO per country, as can be seen in figure 6.2, a majority of the CoOs does not have sufficient amount of data to support the use of multi-class classification. Additionally, the examples where CoO is missing cannot be used in training of a multi-class classification model, thus eliminating 15 examples. In order to use multi-class single-label classification, the amount of data for a majority of the countries would need to be increased, which is not deemed as executable under this study. This study will therefore further explore the possibilities of using binary classification to predict correct or faulty relations in a product's CoO information.

6.3 Preparing data and features

The next step of developing an ML model is to make the collected data usable for the ML algorithms. This is done by first preparing the collected examples in order to set up the data for training and evaluation of the model. Preparing data includes actions such as setting default values where variables are missing, and synthesizing examples if examples are missing. When the dataset is ready for use, the variables are transformed through feature processing to make the information interpretable for the algorithms.

6.3.1 Preparing examples for training and evaluation

Supervised ML models require examples for training and evaluation. In the case of this study, four ways to set up sample data was explored. The ways of setting up data takes different factors into consideration, such as using verified or unverified examples, using true labels or synthesize faulty labels, and which original examples should be used if synthesizing is performed. An overview of the ways to set up data is presented in table 6.2, which is then explained further below.

As presented in the section 6.1 there are 807 verified correct examples in the dataset, but no original examples of faulty data. This eliminates using the first way of setting up the sample dataset. Given the estimation that about 5 percent of the verified data has been faulty and this percentage is expected to be constant (Ekström, 2019), the unverified data (694 unverified examples) is estimated to also mainly be correct. This assumption leads to two implications: (1) First implication is that few new true cases of faulty examples will be discovered. The imbalance between the amount of correct verified data, and the potential true faulty data from the unverified examples (about 35 examples) imposes a hindrance to training and evaluation of a binary classification model. The use of binary classification therefore requires synthesizing of faulty data. (2) The second implication is that all available data (verified and unverified) can be used as training and evaluation data for the model, with a low risk of examples falsely labeled as correct, thus making it possible to use the third and fourth way of setting up the data. Using the full dataset not only increases the amount of original examples available for training and testing, but also increases the number of countries which the model can be used on. As the cleaning of data at Axis puts bigger emphasis on potential products from TAA-compliant countries, non-TAA-compliant countries are more common among the unverified examples than in the verified ones. Including the unverified data will therefore not only increase the number of observations per country, but also increase the total number of countries used by the model.

Table 6.2. Four potential methods to set up data for training and testing explored in this study. Each method is presented with total number of examples available in the final dataset, benefits and drawbacks.

Method to set up data	**Benefit**	**Drawback**
1. Correct and faulty examples from verified data. *Total: 807 correct examples*	+ Most authentic data representation, of both correct and faulty data	- Faulty examples not available - Predominantly TAA-compliant products, limiting the number of countries
2. Correct examples from verified data. Faulty examples synthesized from copy of verified data. *Total: 807 correct + 807 synthesized faulty examples*	+ Authentic representation of correct examples	- Predominantly TAA-compliant products, limiting the number of countries - Risk of dependencies since same original examples are present with both correct and faulty label
3. Original examples from both verified and unverified data. Correct labels for half dataset, synthesize labels for other half. *Total: 751 correct + 750 synthesized faulty examples*	+ Maximizes number of original examples used + Maximizes number of countries present in dataset + Low risk of dependencies, since each example is used once	- Smallest total dataset of the potential methods (2, 3 and 4)
4. Correct examples from verified and unverified data. Faulty examples synthesized from copy of verified and unverified data. *Total: 1501 correct + 1501 synthesized faulty examples*	+ Maximizes number of original examples used + Maximizes number of countries present in dataset + Biggest possible dataset for training and testing	- Risk of dependencies since same original examples are present with both correct and faulty label

When comparing the third and fourth method to set up data, they differ in quantity of samples in the final dataset, what examples are used for synthesizing faulty labels and potential dependencies between examples. The third way of setting up data have low interdependencies between observations, since no parts of observations are duplicated. Instead the third way implies synthesizing faulty examples from half of the available samples, thus keeping the total number of samples the same. Transforming half of the dataset from correct to faulty would be a viable method if the distribution of the examples of each country was more even, but since the distribution is skewed with a number of countries having few observations, then the method might lead to losing valuable insight and introducing undesirable patterns in the data. Though the fourth way of setting up data would duplicate observations through the synthesizing process, the method does not affect the underlying distribution of countries represented in the dataset labeled correct. The fourth way of setting up data also increases the amount of data available for training and testing.

Due to the previously mentioned properties of the original dataset, the sample data were set up according to way four. In the original dataset 148 products have one or more feature value equal to "-MISSING-". These examples were labelled "faulty", in order to train the model to give a warning when any feature value is missing. This resulted in 1353 examples labelled "correct" and 148 labelled "faulty" from the original dataset. In order to synthesize faulty data without missing feature values, the examples labelled "correct" were duplicated and the variable representing the current CoO of the product was replaced with a randomly generated, new CoO. The new CoO was chosen from the 27 possible countries for the product from IFS/Inventory part. All CoOs were generated with the same probability of being generated, controlling to avoid the correct CoO. This resulted in faulty samples with values of all other features being realistic, in addition to a generated CoO of the SU. The full dataset had 1353 examples labeled "correct" and 1501 labeled "faulty", in total 2854 labeled examples.

Table 6.3. Distribution of the labels of binary classification.

Label	Count of each label	Percentage
Faulty	1501	52.59%
Correct	1353	47.41%
Grand Total	**2854**	

As can be seen in the table 6.3, a narrow majority of the processed data is labelled faulty (52.59%). As this is the majority, this percentage also makes up the null accuracy.

6.3.2 Feature processing

In order to make the raw data usable for the ML system, the variables of the data was translated into features. As can be seen in figure 6.4, there are six variables identified as relevant in the ERP-system. The performed feature processing is explained below.

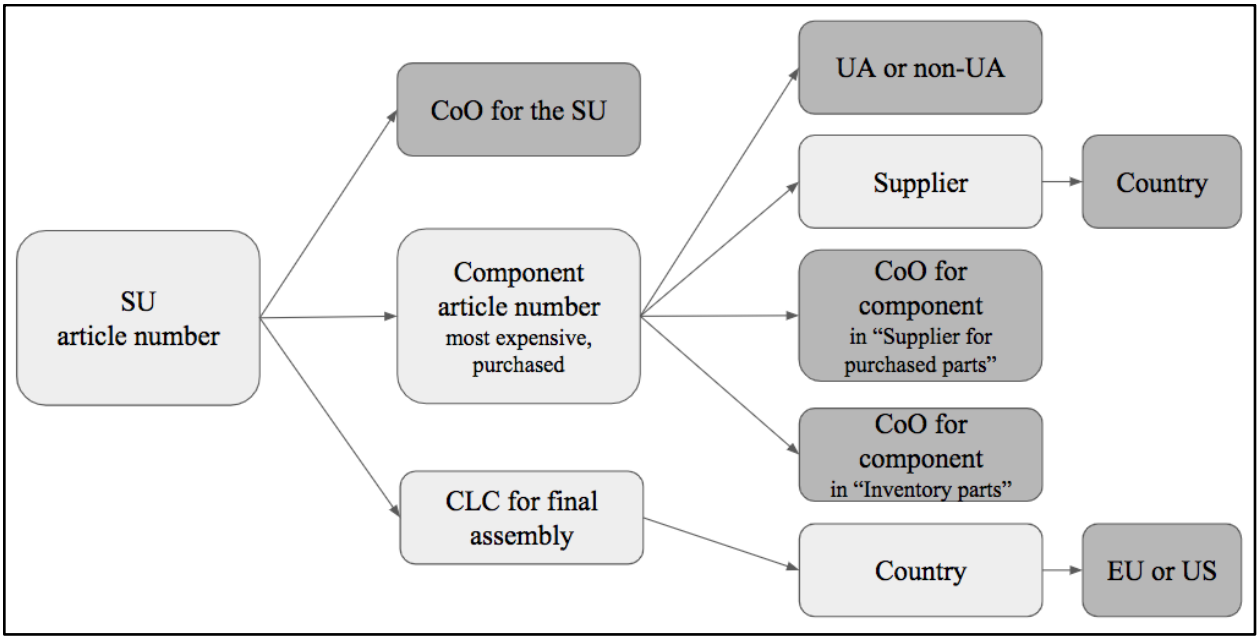

Figure 6.4. Identified, relevant data found in the ERP-system for deciding CoO of an SU article. CoO-relevant data are at the end nodes.

UA or non-UA - *binary feature*

There are no isolated data field in the ERP-system establishing whether or not a component of an article is a UA. Instead, UA is written as part of the component description. For determining an article's CoO, only the purchased component with the highest unit cost is relevant. A binary feature is therefore introduced for that specific component, processing the component description to deduce whether or not the component is a UA. The feature has values '0' and '1', where '0' indicates that the component is not a UA, and '1' is a UA.

EU or US - *binary feature*

As established in section 5.1.4, there are two different supply chain flows of goods to the US: Either through a CLC in Europe and then to a CLC in the US, or directly to a CLC in

the US from the suppliers. In order to differentiate between the two flows of goods, a feature is introduced to show if the country of the CLC managing final assembly is in the US or in Europe, where ‘0’ indicates not in the EU and ‘1’ indicates in the EU.

Country, CoO for product and CoO for component - *categorical features*
The four fields Country (IFS/Supplier), CoO (IFS/Supplier for purchased parts) as well as CoO for component and CoO for the SU (IFS/Inventory parts) are text fields containing a country code. In case the data field is blank, the feature is assigned the default value ‘-MISSING -’.

As can be seen in appendix A.3, table A.3.1, Country (IFS/Supplier) has 22 possible values, CoO (IFS/Supplier for purchased parts) has 30 possible values and CoO (IFS/Inventory parts) 30 possible values for the component and 28 possible values for the SU, where ‘-MISSING-’ is included as a possible value. Many ML algorithms interpret categorical features as ordered and continuous. Since countries and CoOs are neither ordered nor continuous, all features were transformed into binary features using dummy coding. Each feature was transformed into as many features as it had possible values. For example, Country (Supplier) was transformed into 22 binary features, called ‘Supplier country_UNITED STATES’, ‘Supplier country_POLAND’ etc.

The feature processing resulted in 112 features in total.

6.4 Tools and libraries for model development

The development environment used in this study was Jupyter Notebook, which is a web-application where you can develop and share documents including code, visualizations and text. It was used in this study because of its vast support of data science, scientific computing as well as all programming languages. Jupyer notebook is open-source, i.e. free to use and distribute. (jupyter, 2019)

The programming language used was the general-purpose programming language Python. Python is known for its readability and usability. Data science and ML projects are often written in Python due to its many scientific libraries, making the language useful for data analysis. (Pedregosa et al., 2011)

The scikit-learn library for Python contains algorithms for ML such as classification, regression and clustering algorithms (Pedregosa et al., 2011). Estimators in scikit-learn are Python objects that implements a specific ML algorithm. The estimator implements a

‘fit’ method which accepts data array as input and, in the case of supervised learning, an array of labels. An estimator is fitted to the given dataset with the ‘fit’ method to be able to predict labels on future, unseen data. Supervised estimators implement a ‘predict’ method that can predict labels. (scikit-learn, 2019a)

In scikit-learn, hyper parameters are given as arguments to the constructor of the estimator. A GridSearchCV method can be used with an estimator to select and tune the hyper parameters, from a specified parameter grid, to get the best cross-validation score. GridSearchCV has a ‘scoring’ parameter, which is set to accuracy in this case. CV stands for cross-validated because cross-validation is performed with a specified number of folds, without stratification. (scikit-learn, 2019a)

Scikit-learn provides cross-validation iterators, for example Kfold and StratifiedKfold. StratifiedKFold is a cross-validation object where the folds contain a representative percentage of samples from each class in the dataset. The parameter ‘n_splits’ is the number of folds and set to 10 in this case. The parameter ‘shuffle’ is a boolean and decides whether the samples should be shuffled before the splitting. (scikit-learn, 2019a)

The functions accuracy_score and confusion_matrix are metrics imported from scikit-learn’s sklearn.metrics module. accuracy_score was used to get the train and test classification accuracies. accuracy_score has parameters ‘y_true’ and ‘y_pred’ which contain the correct label respectively the predicted labels from the classifier. Confusion_matrix was used to calculate the TNs, FNs, TPs and FPs, and also has ‘y_true’ and ‘y_pred’ as parameters. (scikit-learn, 2019a)

Other libraries used in this study were Pandas and Numpy, which are libraries providing data analysis tools for Python, useful for data science and machine learning. Pandas provides data structures designed to work on labeled data and can read and write from CSV or Excel files into DataFrame objects. A DataFrame object is 2-dimensional size-changeable tabular structure for data manipulation with integrated indexing. Numpy provides an array language where data can be presented as numpy arrays and arithmetic operations can be performed on them, as well as mean values and variances can be calculated. A DataFrame object can be created from NumPy arrays and vice versa. (McKinney, 2019)

6.5 Model training, improving and evaluation

Before training the model, the overall structure of the model evaluation procedure was decided upon based in the procedures mentioned in section 3.6. Out of the four evaluation procedures, k-fold cross-validation without hold-out set was deemed most suitable for the study. As discussed in section 6.1, the underlying dataset was unevenly distributed across countries, resulting in a tail of countries with few sample observations. If data was to be held out of training, as is the case of train/test split and k-fold cross-validation with hold-out, the countries with few sample observations would be at risk of not being represented in either the training set or the testing set. Training and testing on the same data as an evaluation procedure was dismissed due to lack of generalizability.

When a model evaluation procedure was decided upon, a general code shell was built in order to have a common model structure to fairly compare the estimators. A common code shell ensured that both model evaluation procedure and model evaluation metrics were used equally for all estimators. The code shell included (1) initializing the estimator, (2) improving model parameters through hyper parameter tuning, (3) customizing k-fold cross-validation, and (4) taking customized metrics on feature level.

The hyper parameter tuning was done using scikit-learn's method GridSearchCV. The number of parameters and types of parameter values used depended on which estimator was used. The grid search resulted in optimized hyper parameters, which were later used in training the model.

After the hyper parameter tuning, the data was split into folds for k-fold cross-validation. Initially during the experimental phase normal k-fold cross-validation was used, thus splitting into folds traversing through the dataset in a successive order from top to bottom. As the dataset was organized with the correct samples grouped before the faulty samples, normal k-fold cross-validation generally had the same label through all test sets, thus not providing a good representation of the general underlying distribution. Normal k-fold cross-validation was then replaced with the scikit-learn splitter class StratifiedKFold, where the percentage of samples match the distribution of samples in the overall dataset. The stratified k-fold cross-validation was performed with ten splits, as is recommended by Kohavi (1995) mentioned in section 3.8. The observations of each class were shuffled before splitting the samples into folds by using the parameter *shuffle* in StratifiedKFold. This randomized shuffle further decreased the influence of potential patterns from the dataset, in addition to the stratification.

Predictions and metrics were calculated for each fold of the cross-validation. After the estimator had been fitted to the training set of the fold, predictions were made from both the testing set and the training set. The predictions of the testing set were saved with the corresponding observation. The accuracy was calculated for both the testing and training data, by using the scikit-learn metric accuracy_score to compare the true labels and the predicted values, resulting in both a testing accuracy and training accuracy of each fold. A confusion matrix for each fold was calculated through using scikit-learn metric confusion_matrix. To produce the general metrics, the results from each fold were combined. The general confusion matrix was calculated by summarizing the confusion matrices over all folds. Training and testing accuracy were calculated for the whole dataset as mean values over all folds.

On feature level, confusion matrices were calculated for each country present as a current CoO. For each example the country, true label and predicted value were gathered. The true label and predicted value were compared for each example. The results were then summarized per country, showing the confusion matrix for each country through the number of TPs, TNs, FPs and FNs. Metrics of classification accuracy, recall and specificity were calculated from the confusion matrix table.

When the shell structure was finalized, the model was trained and tested on the four potential estimators in scikit-learn: DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier and SVC (a type of SVM). When possible, the estimators were improved upon by using GridSearchCV to find improved model parameters. Since all models showed high classification accuracy, no further improvement efforts were used.

General evaluation metrics used to evaluate model performance of the estimators were classification accuracy of the test set (test accuracy), classification accuracy of the training set (training accuracy) and the confusion matrix based on the testing sets. From the confusion matrix, the recall and specificity were calculated for each estimator. As the primary goal for the ML model is to both validate correct CoO as well as give a warning when the CoO seems to be faulty, TPs and TNs are both important measures. Recall and specificity were therefore chosen as additional metrics. Since no differentiation in preference was made between predicting correct and faulty labels, different classification thresholds were not investigated. Hence, no ROC could be or was relevant to calculate which implies that an AUC was not calculated. As is common for binary classification, test accuracy was used as the main metric for choice of estimator for the final model (Amazon Web Services, 2016). Further metrics were evaluated on the final model in

order to explore the performance on feature level. Since the main objective of the model was to evaluate the current CoO of the ERP-system, performance metrics were used for each country, represented as current CoO. For each country, the metrics used were: the confusion matrix, classification accuracy, recall and specificity. The difference between the test and null accuracy was computed in order to evaluate how well the model had actually learned connections.

6.6 Estimators in scikit-learn

Scikit-learn has a number of estimators suitable for binary classification problems. In this study, the estimators DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier and SVC (scikit-learn, 2019a) were chosen for training due to their appropriate characteristics. The theory in this chapter regarding the estimators is based on scikit-learn's user guide (scikit-learn, 2019a). The estimators were chosen for initial experimentation based on scikit-learns' guide to choosing the right algorithm (scikit-learn, 2019b), the theory in section 3.2 and by recommendations from Gustafsson (2019).

DecisionTreeClassifier
DecisionTreeClassifier can perform both binary-class as well as multi-class classification on datasets. The 'criterion' parameter decides how to measure the quality of a split. Another parameter is 'splitter', deciding how to choose a split at each node, which can be to choose the best split (default) or the best random split. Parameters which decides the size of the tree are among others 'max_depth', 'min_samples_leaf', 'min_samples_split'. The parameters lead to wholly developed trees which can possibly get very big when their default values are used. It could consume lots of memory. Additionally, a big, complex tree could possibly lead to overfitting.

RandomForestClassifier
RandomForestClassifier is an ensemble method, using averaging, and consists of decision tree classifiers on different sub-samples of the total dataset. The sub-samples are the same size as the total dataset, but samples from the training set are drawn with replacement. The split chosen when building a tree, is not based on the best split of all features but the best of a random subset of the features. This randomness leads to increased bias but when averaging, the variance decreases, usually resulting in a better model than a decision tree. The parameter 'n_estimators' is the number of trees in the forest. The default value will change from 10 to 100 and hence the GridSearchCV method was applied with the 'n_estimators' parameter optimized over the values [10, 20, 30...100]. Apart from 'n_estimators' the parameters are the same as for the DecisionTreeClassifier.

KNeighborsClassifier

KNeighborsClassifier implements the k-nearest neighbor majority vote classification. The estimator has eight parameters. ‘n_neighbors’ and ‘weights’ were tuned with GridSearchCV. ‘N_neighbors’ is the value of k and were given options of integers between 1 and 30. ‘weights’ is the weight function used for prediction and possible values are ‘uniform’ and ‘distance’ which were the options given in the search.

SVC

The problem of this case has many features compared to the number of samples and SVMs are usually effective in those cases and hence included. SVC has 14 parameters and ‘C’, ‘kernel’ and ‘gamma’ were chosen to be tuned with GridSearchCV. Since the training was experimental, those parameters seemed most promising and were first tested. ‘C’ is the penalty parameter C of the error term, the regularization parameter, and the default value is 1. ‘Kernel’ is the kernel function and the default is rbf, radial-basis function. ‘Gamma’ is the kernel coefficient if the kernel is ‘rbf’, ‘poly’ or ‘sigmoid’. SVC is very complex and takes lots of computational power to train, which is why ‘C’ as well as ‘gamma’ were only given the options 0.1, 1 and 10 and ‘kernel’ was given the options linear, rbf, poly and sigmoid.

7 Result and analysis

Results of the models after training, improving and evaluation of the potential estimators are presented in section 7.1. It is followed by section 7.2 which contains more detailed results for the final model. Finally, section 7.3 contains the analysis of the results.

7.1 Comparing potential estimators

The potential estimators' results from the training and testing are presented. The test accuracy and train accuracy are shown in figure 7.1 for the DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier and SVC. The results are from stratified 10-fold cross-validation with hyper parameters tuned through separate grid searches.

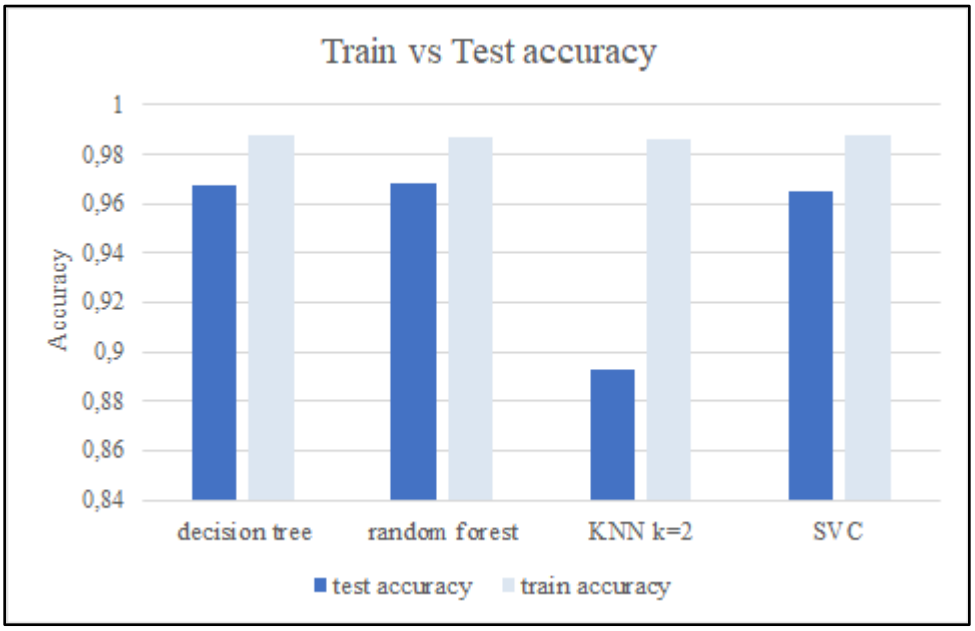

Figure 7.1. Visualization of test and train accuracies for the ML model when using the four potential estimators. The diagram is based on the numbers presented in appendix A.4, table A.4.1.

For the DecisionTreeClassifier, the training and testing on the dataset did not take too long and no overfitting occurred. Hence, no hyper parameter tuning was needed, and the default values were chosen. The result from 10-fold cross-validation is shown in a confusion matrix in figure 7.2.

	Predicted: FALSE	Predicted: TRUE
Actual: FALSE	True negative 1452 (50.9%)	False positive 49 (1.7%)
Actual: TRUE	False negative 43 (1.5%)	True positive 1310 (45.9%)

Figure 7.2. Confusion matrix for the DecisionTreeClassifier.

For the RandomForestClassifier, the grid search resulted in 20 decision trees and the result was satisfying. Hence, no more hyper parameter tuning was needed. The result from 10-fold cross-validation with this hyper parameter value is shown in a confusion matrix in figure 7.3.

	Predicted: FALSE	Predicted: TRUE
Actual: FALSE	True negative 1461 (51.2%)	False positive 40 (1.4%)
Actual: TRUE	False negative 50 (1.7%)	True positive 1303 (45.7%)

Figure 7.3. Confusion matrix for the RandomForestClassifier.

For the KNeighborsClassifier, the grid search resulted in *k* equal to two and *weights* equal to distance. It did not seem as promising as the other estimators and were not investigated further. The result from 10-fold cross-validation with these hyper parameter values is shown in a confusion matrix in figure 7.4.

	Predicted: FALSE	Predicted: TRUE
Actual: FALSE	True negative 1243 (43.6%)	False positive 258 (9.0%)
Actual: TRUE	False negative 47 (1.6%)	True positive 1306 (45.8%)

Figure 7.4. Confusion matrix for the KNeighborsClassifier.

For the SVC, the grid search resulted in *C* equal to 1, *gamma* equal to 1 and *kernel* equal to rbf. The result from 10-fold cross-validation with these hyper parameter values was satisfying and no further hyper parameter tuning was made. The result is shown in a confusion matrix in figure 7.5.

	Predicted: FALSE	Predicted: TRUE
Actual: FALSE	True negative 1470 (51.5%)	False positive 31 (1.1%)
Actual: TRUE	False negative 68 (2.4%)	True positive 1285 (45.0%)

Figure 7.5. Confusion matrix for the SVC.

From the confusion matrices, specificity and recall were calculated for each of the classifiers. The results are shown in table 7.1.

Table 7.1. Specificity and recall for the four classifiers.

	Specificity	**Recall**
DecisionTreeClassifier	0.967	0.968
RandomForestClassifier	0.973	0.963
KNeighborsClassifier	0.828	0.965
SVC	0.979	0.950

7.2 Result for the chosen model: RandomForestClassifier

The RandomForestClassifier was chosen as the final model since it had the best test accuracy out of the four potential classifiers. A sample of the result on null accuracy, test accuracy, recall and specificity for the current CoOs are presented in table 7.2. For the full table of all CoOs of products in the original dataset, see appendix A.4 table A.4.3. The null accuracy is calculated based on the label most prevalent in the examples of that CoO. The metrics test accuracy, recall and specificity are calculated from the confusion matrix of each CoO presented in appendix A.4, table A.4.2.

Table 7.2. Sample of results on feature-level for the chosen RandomForestClassifier model, including number of observations labeled correct and faulty in the original dataset, null accuracy, test accuracy, recall and specificity for each of the current CoOs for the product (IFS/Inventory Part) present in the original dataset. For the full table, see appendix A.4 table A.4.3.

Country of origin	Number of examples with label correct	Number of examples with label faulty	Null accuracy	Test accuracy	Recall	Specificity
- MISSING -	0	15	1	1	0	1
Bulgaria	1	63	0.984	0.984	0	1
Canada	2	62	0.969	0.969	0	1
China	380	37	0.911	0.945	0.965	0.853
Italy	21	46	0.687	1	1	1
Japan	136	37	0.786	0.994	1	0.98
Lao People's Democratic Republic	1	55	0.982	0.982	0	1
Mexico	48	50	0.51	0.98	0.979	0.98
Sweden	104	47	0.689	0.921	0.949	0.865
United Kingdom	60	44	0.577	1	1	1

The difference between the test and the null accuracy was calculated for each of the current CoOs of products for the RandomForestClassifier. The result is shown in figure 7.6.

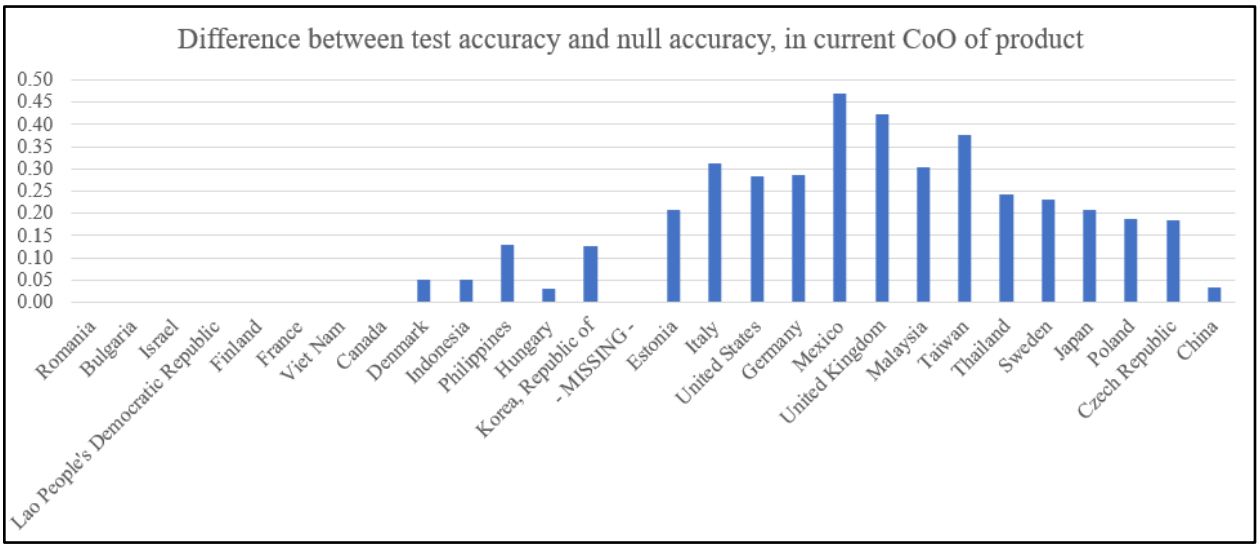

Figure 7.6. Difference between the test accuracy and null accuracy for the specific CoOs present as current CoO of product in the original dataset. Countries are ordered by increasing number of examples in the original dataset.

7.3 Analysis

As is shown in figure 7.1, the test accuracies are high (almost 97 %) for the decision tree, random forest and SVC models and close to the train accuracies of almost 99 %. This indicates that those models generalize well on new data and neither over- nor underfitting occurs. For the kNN model, the train accuracy is also almost 99 %. This high train accuracy is expected, since kNN model make predictions by directly comparing the new dataset to the one the model is trained on. This direct comparison limits the model's generalizability. As the test accuracy (89.3%) is markedly lower than the train accuracy, it implies that the kNN classifier overfits.

The random forest classifier was chosen for the final model based on that it had the highest test accuracy. The random forest classifier, as well as the other classifiers, had accuracies well above the null accuracy at 52.6 %. As mentioned in section 3.7, null accuracy can be used as a baseline to establish the worst possible performance if the model were only to predict the value of the most common class. As the model's test accuracy is well above the null accuracy, it implies that the model performs well.

More metrics than the accuracies were calculated. The values of the confusion matrices made it possible to calculate the specificity and recall values for the classifiers, as presented in table 7.1.

The random forest classifier has high specificity and recall value, both above 96%. The high specificity implies that the model has a low number of FPs, while the high recall implies a low number of FNs. Since both metrics are low, this implies that the number of total errors is low. As mentioned in the section 6.1, only approximately 5% of the verified data has been faulty, which also is accepted as an estimation of the fault rate of the rest of the data. Since there likely will be significantly more correct observations than faulty in the real world, the specificity is a more relevant metric than recall to consider if the model was to be implemented and put to use.

SVC was the model with the highest specificity, as can be seen in table 7.1. This, in combination with the high test accuracy, indicates that SVC could have been a potential alternative to random forest classifier.

The KNeighborsClassifier model has a markedly lower specificity than the other models (82.8%), as shown in table 7.1. This implies that the model fails to mark out faulty observations but instead falsely predicted a larger number of observations to be true, thus making the model a less suitable alternative.

As can be seen in section 7.2, the chosen random forest model has learned that if the feature value is “-MISSING-” in IFS/Inventory part for the SU, the observation should be labelled as faulty. This capacity to identify missing values is considered a minimum requirement of either of the models, since a product with the feature value “-MISSING-” should be corrected. This result is shown in the test accuracy being 100 %. Furthermore, the null accuracy was 100 %. Hence, the difference between the test and null accuracy is zero as can be seen in figure 7.6. However, although the result is promising for the feature of CoO for the product, it was not further investigated how the model performs for other types of features marked as “-MISSING-”.

The chosen model does not perform well on observations with CoOs with few examples, such as Bulgaria. This is partly due to that the assumption was made that when synthesizing the faulty labels, the countries were chosen randomly with the same probability of each country. This random generator with uniform distribution was used even though some countries appear much more frequently than others. For example,

China had 380 examples in the original dataset whereas Bulgaria only had 1 example. Still, Bulgaria was generated 63 times during the synthesizing of the CoO-data, whereas China was generated 37 times, as can be seen in table 7.2. This caused the distribution on CoO-level to change: When China was randomly generated for an observation already having China as a label, it had to regenerate a CoO until it got a label not equal to China. Since China had relatively many observations in the original dataset, not as many faulty observations were generated with CoO equal to China. Even though the dataset for training and testing was close to balanced when considering labels, the dataset on CoO-level was imbalanced.

The synthesizing of faulty data results in the model predicting the observation to be faulty, if an observation has a CoO of a country with only one example in the original dataset. Since CoOs with only one original, correct example have significantly more faulty observations than correct, the model is trained on faulty being the predominant value. For example, the null accuracy for Bulgaria is 98.4% which shows that the dataset for Bulgaria is very imbalanced, with only one observation labeled correct and the rest faulty. When the observation with the label correct is in the test set, the model has been trained that in every case when there is this CoO, it is faulty. This is also the case for Canada, Finland, France, Israel, Lao People's Democratic Republic and Romania. In those cases, the specificity is 100% while the recall is 0%. The reason for this is that the TP and FP are equal to zero since the model was making false predictions on this correct (positive) example, when always predicting faulty (negative) and never correct.

The imbalance in datasets for many CoOs is shown in the values of null accuracies in table 7.2. The difference between the null accuracies and the test accuracies is equal to zero for the CoOs with only one observation in the original dataset, as can be seen in figure 7.6. As well, the same small difference between test and null accuracy exist for CoOs with high imbalance with more correct than faulty examples such as China. To take China as an example, it has a test accuracy of 94.5%, recall of 96.5% and a specificity of 85.3% which is a lower specificity than for other CoOs, implying that more FPs are delivered.

Hence, figure 7.6 indicates that the model might not have learned the right patterns and connections properly for CoOs with imbalanced datasets, even though the test accuracies are high. On the other hand, the CoO with the best results is Mexico. Mexico, has the biggest difference between test and null accuracy, with a high test accuracy of 98.0%. Mexico was the CoO with the most balanced dataset, with a null accuracy of 51.0%.

Furthermore, Mexico has a specificity of 98.0% and recall of 97.9% which in combination with the difference between the test and null accuracy, indicates that the model has learned the pattern well for Mexico. Hence, having a well-balanced dataset might be a success factor for the ML model.

In order for the model to be useful, it would have to perform well on every country which is not the case. After evaluating and interpreting the results, the ML model using the RandomForestClassifier shows potential but is not considered successful when trained on the available data.

8 Discussion

The discussion is divided into two sections. The previous chapter covered the result and analysis of the ML model, forming a foundation for the first section of this chapter: discussing the limitations of the developed ML model in section 8.1 Limitations of the model. The second section, 8.2 Discussion of the CoO-problem, broadens the discussion to the use of ML for the CoO problem in the Axis context.

8.1 Limitations of the model

Though the result of the model evaluation has shown some promising results, there are model limitations. The limitations stem from varying factors, which can be traced to different parts of the workflow of developing the ML solution.

The first limitation of the model arose from the quantity of the sample data available for training and testing. As discussed in section 6.2, the availability of examples for a number of CoOs were limited. Since this made current CoO unsuitable to use as a label, it was not possible to form a multi-class single-label classification problem. Instead, the problem was framed as a binary classification problem, where the relationship between CoO-data was predicted as correct or faulty instead of predicting the CoO of a product. Because of this, the requirement from Axis of predicting a CoO could not be fulfilled.

The limitation in the available data also affected the result of developed ML model. As stated in the theory of section 3.5, each feature value is recommended to occur at least five times. This has not been fulfilled for some of the features. As an example, there were 10 values of CoOs of the product (IFS/Inventory part) with less than five observations in the original dataset. The chosen model cannot give a trustworthy result for these countries, as was shown in the result on specificity, recall and null accuracy, discussed in section 7.3. As the result for these countries is not trustworthy, it narrows down the potential use of the model.

The result might also have been affected by the quality of the original data. During the development of the model, it was discovered that the verified and corrected observations in fact might only have had the data field of the current CoO of the product corrected. As the cleaning only included one of the data fields, the dataset might still include defect values for some features. It was not investigated further to what extent this was the case.

The model might also have been affected by the collection and synthesis of data. The most veridical result would have been achieved through the use of both correct and faulty verified examples. Unfortunately, no true faulty examples had been saved, and were therefore not available. Instead the dataset of both verified and unverified data was used, as well as synthesized faulty examples. Since unverified observations could possibly be faulty, there is a risk that the model was wrongly trained to some extent. Furthermore, in the synthesis of faulty examples all original data was used, resulting in a dataset of almost double size compared to the original dataset. An option to this method of producing data would have been to separate the full dataset into two subsets, and keeping one with original features labeled correct and the other subset with synthesized data labeled faulty. This would have resulted in observations being unique and lowered the risk of the model learning unrealistic connections, but would have halved the total dataset. This option was not investigated further due to time constraints of the study.

The chosen method of synthetization of data might also have introduced limitations to the model. As presented in section 6.1, the features of the original data were not of a uniform distribution. However, the synthesized data was uniformly distributed. When combining the original and synthesized data, the underlying distribution shifted. Due to time limitation of this study, the implications of this shifted distribution have not been researched.

During the feature processing, the two variables regarding which CLCs an article is passing through were transformed into a feature describing whether or not the CLC for final assembly is located in the EU. Since the products used in this study only pass through CLCs in the EU or US, thus excluding all other CLCs, the EU-feature can be interpreted as a binary: EU or US. This way of transforming variables into a feature is a form of simplification of the data, where some information might be lost. The feature now portrays which of the two supply chain setups the product is part of, which has impact on which rules and regulations would be influential when defining the CoO. One weakness of the simplification is that the feature is limited to information about the market of the final CLC, but no information about which specific CLCs are in use. An alternative way of feature processing is to create categorical features of the variable for primary CLC and CLC for final assembly. Though this would provide the ML model with more information, it would also increase the complexity of the model. If using dummy coding, the variable of the primary CLC would be represented by three features (one feature for each CLC on the US market and one feature for information missing). In the same way, the CLC for final assembly would be represented by six features (features

for all CLCs on the EU and US market and one feature for information missing). This way of feature processing would result in 9 features in total, instead of one feature. Due to the time restraint of this study, ML models using the more comprehensive way of feature processing were not explored.

Also on the subject of feature processing, dependencies and correlations among the features were not investigated. However, the decision tree and random forest classifier takes this aspect into consideration. For example, if two features would have had high correlation, only one would be chosen since the other wouldn't contribute to a high information gain.

During training of the model, a possible shortcoming with the model is that the grid search used for finding optimal hyper parameters, was not made with stratified folds. The final cross-validation was made with stratified 10-fold cross-validation. Hence the found optimal hyper parameters might not be optimal for the model when stratification was made. However, the results were satisfying, thus this was not considered an issue.

A limitation of using ML in general is the interpretability of the result. Since ML models learn by identifying patterns in data, it is not certain that the patterns align with programmable logic. This makes predictions of an ML model act like a black box, and it might be difficult to distinguish why a certain prediction was made. The exception to this is the decision tree algorithm, which is the most interpretable of the classifiers used, followed by the random forest. The decision tree can be visualized and hence the steps which have been taken in making a prediction can be visualized too. The importance of each attribute can be shown and given as outputs for both a decision tree and a random forest. This gives transparency to the ML model, which could be of use for the Axis employees.

Axis further requested one output of the ML system to be the probability of the prediction, the existing CoO in the system as well as other input values and the timestamp. Those requests would probably be possible to fulfill. However, this was not investigated further as an implementation is not considered to be useful.

8.2 Discussion of the CoO-problem

The problem identified by Axis was that CoO of a product in the ERP-system is not always correct. In the mapping following the discovery, four general problem areas related to the process of defining a product's CoO were identified:

(1) Lack of ownership of the data connected to CoO,
(2) Inadequate processes and ways of working regarding CoO,
(3) Information and updates from external stakeholders,
(4) Changes in the product life cycle, e.g. change of supplier (Olander, 2019).

This study would argue that the four problem areas are of different nature from each other, and can thereby be divided into different types of problems: Administrational areas of improvement at Axis, and challenges of defining CoO.

Both the first problem area (lack of ownership) and the second problem area (inadequate processes regarding CoO) are administrational in nature. In the case of Axis, the responsibility of managing information and questions regarding CoO has not been allocated to an accountable department or role. Instead, multiple departments within Axis Operations have worked with data related to CoO, more or less unknowingly of what impact changes of the data might have for other users. This has led to loss of control over the data in the ERP-system and how it is used. Though important for the broad topic of working with CoO in a business setting, and for the quality of data in the ERP-system, the administrational task of establishing ownership and ways of working is outside the scope of this study.

The third problem area shows the flexibility and mutability of macro-environmental factors affecting Axis. These changes are results of an ever-changing business context and will by varying degree influence how a product's CoO is defined. As discussed in chapter 4, the definition of a product's CoO can for example differ depending on the rules and regulations of the buying market, as in the case of European and US legislation. Even though initiatives are in progress to unify standards regarding CoO, the standards are under constant development which brings an increased risk of upcoming changes. This risk cannot be eliminated by Axis, but is instead part of the environmental context of which Axis is working in. The problem of changing information from external stakeholders (3) entails two challenges: (a) how to collect information from external stakeholders, and (b) how the new information is represented in the ERP-system. The first challenge (a) regarding how information about changes in the surrounding environment is collected and handled by Axis is an administrative and organizational task. It has low direct impact on the question of using ML in a CoO context and is therefore not included in this study.

The second challenge (b) presents a bigger risk for the use of ML, as macro-environmental changes might affect the distribution of the underlying data in the ERP-system. As discussed, ML require data for training of the model. If there is a shift in the distribution of the underlying data, the correctness of the model's predictions could be negatively affected. In order to recompose the model's predictive power, data of the new distribution will need to be collected in enough quantities before the model can be accurately retrained on it. To be able to use ML for CoO predictions in the Axis context, these changes in the environment needs to be identified, as well as enough data needs to be collected. Given that the US market has a limited number of products spread across a bigger number of CoOs, the manual act of producing data of the new distributions is likely to be more labor intensive than the time saving impact that the use of ML could bring. The challenge of using ML with mutable environmental factors and changing distributions of data is commonly called concept drift. Given the shared interest and the rapid development of the field of ML, this might be a relevant area for future research.

The fourth problem area in regard to CoO is changes in the product life cycles, such as changing primary supplier of a component. As in the case of the third problem area, the challenge of changes in the product life cycles is not a challenge Axis can eliminate. On the contrary, having a flexible supply chain is an asset Axis as a company takes pride in, as described in section 5.1.1.

Changes to a product's upstream supply chain might have a direct impact on the CoO of the product, depending on what components are affected by the change. As discussed in chapter 4, CoO of the product can be decided by the CoO of the most expensive component. If the most expensive component was to be changed, the CoO of the final product would also have to be revised. This flexibility in the supply chain exacerbates two problems identified in this study regarding CoO in the ERP-system within Axis: (a) the lack of support for different departments' varying usages of CoO information and (b) how to manage dependencies between data fields in the ERP-system.

The problem of supporting different usages of CoO information (a) stems from the concept of CoO being relevant for different departments of Axis Operations. Information regarding CoO is used by departments at different places of the internal supply chain, for example by sales and customs. Even though these departments are both interested in a product's CoO based on the same definition of CoO, they might be interested in different batches of the SU. To use an example presented in section 5.1.7, sales and customs use the CoO of a product differently. Even though the two departments are both interested in

CoO of a product with a certain article number, they are in reality interested in two different product units. Given Axis supply chain flexibility, different batches of the same SU can have different CoO. Examples of such flexibility are when SUs in stock were purchased from a different supplier than the current primary supplier, or remanufacturing has made returned SUs ready to be sold again. The current ERP-system does not support CoO information on a batch-level, but instead only one CoO can be shown. In order to neatly support the different usages of CoO in a context of a flexible supply chain, the current ERP-setup need to be revised. However, as this study stems from a wish to simplify the sales process on the US market through verifying TAA-compliance of products, the CoO information in the ERP-system has been interpreted as the CoO used by sales. Given this assumption, that the CoO in the ERP-system has been seen as a CoO set early in Axis internal supply chain, this study has not taken delay time or product flows in the internal supply chain into consideration when exploring the use of ML.

The second problem (b) a flexible supply chain might exacerbate is the impact of dependencies between data fields in the ERP-system. When examining the data fields relevant to a product's CoO, the data fields can be split into two types: input fields, and derived fields. Input fields are data fields where information from the product context is added, such as supplier information or BoMs. This information does not exist in the ERP-system from before. The other type of data field, derived fields, contain information which have been derived from information already in the ERP-system. The data field for current CoO of a SU (IFS/ Inventory part) is a derived data field.

As described in chapter 5, there are multiple departments involved in the process of adding CoO-data in the ERP-system. Depending on the product, the CoO is set by either sourcing, purchasing or PDG. The different departments are working in different views of the ERP-system while some views are shared. Currently, all data fields are treated as input data fields in the ERP-system. Related data can be present in several views without a connection between the data fields, thus increasing the risk of diverging information. Even though information related to CoO is changed, such as primary supplier or CLC for final assembly, there are no routines to update the derived fields, such as current CoO. With a flexible supply chain the risk of changes increases, thus increasing the risk of diverging information.

It is in this stage of the process of deciding CoO where ML potentially could be used to classify whether the relationship between the input fields and derived field (CoO) are of a correct format or faulty. As discussed in the section above, the exploratory experiments

of this study show that ML has the potential to recognize relationships within data related to CoO. However, even though the ML model shows promising results, the use of ML does not fulfill the requirement of usability or time and resources saved for Axis. Furthermore, this study has through the development of the ML model identified four limitations within Axis Operations in manage ML in the context of verifying CoO:

(1) Lacking storage of historical data
(2) Difficulties in retrieving data from the ERP-system
(3) Lacking experience and knowledge of how to analyze input data for retraining
(4) Lacking experience and knowledge of how to interpret and validate the result

Access to data for training and testing is a major linchpin to the use of ML. Not only does the data need to be of big enough quantities, but in the case of classification algorithms, the data also need to contain a representative number of samples, both of all labels and of features. These requirements on data availability is not present in the ERP-system at Axis. The limitation of overall quantity of data is in large due to Axis not saving historical data. Instead, only current data is available. Additionally, Axis flexible supply chain leads to higher risk of features (countries, CoOs etc.) missing from the current data, in case they are not used at present time. As ML needs representation of all of the variables the model should be trained to recognize, the model will not be able to handle products which countries are not present in the training data.

The second limitation to the efficiency of using ML in the CoO context is the retrieving of data. Given the current setup of the ERP-system, there is no straightforward method to easily acquire the data needed for the ML model. Within the limits of this study, easy retrieval of data has not been achieved. This presents a limitation to the use of ML, since a request of the model was that it should be making predictions daily.

The third and fourth limitation to using ML in the CoO context stems from the restrictions in interpretability of data and ML algorithms. On the relation between ML efficiency and interpretability, Vellido, Martín-Guerrero and Lisboa states that "[ML models] can be rendered powerless unless they can be interpreted, and the process of human interpretation follows rules that go well beyond technical prowess" (Vellido et al., 2012, p. 163). When using ML to explore the relationship between data fields in the ERP-system, the predictions are not made based on the rules of how to define CoO, but instead on intrinsic patterns of the data. This results in the ML model being a kind of black box-model, where the low interpretability in turn raises the demands on the person managing

and retraining the model. Maintaining the ML model would therefore require resources not present in the departments the tool was intended for, namely Sourcing and PDG.

With the limitations in usability and efficiency in mind, the overall return on and suitability of using an ML model to verify a product's CoO is unfavorable.

9 Conclusion

In the following chapter the conclusions of the study are presented. Section 9.1 presents the fulfillment of the purpose of the study, followed by answers to the research questions in section 9.2. The chapter is then finished with section 9.3 Recommendations for Axis and section 9.4 Areas for future research.

9.1 Fulfillment of purpose

The purpose of this study was to explore the possibilities of using ML to validate a product's CoO. The study followed a design science research approach, thus combining theory and empirical studies with experimental research. The depiction of the theoretical knowledge was divided into two topics relevant for the study: ML and CoO. The theory on ML proceeded from a comparison of major actors' workflows of ML projects, thus describing areas relevant for developing an ML model. The theoretical chapter on CoO provided an overview of the topic of CoO, especially on the geographical markets relevant for this study. In order to create a context for the subject of defining CoO, an empirical mapping was carried out of how Axis currently works with the CoO question. To combine the theoretical knowledge base and the empirical environment, experimental development of an ML model was performed, comparing the performance of four common ML algorithms. The development of the ML model followed the steps established in the comparison of workflows. After analyzing the model based on its performance and limitations, the use of ML in the context of validating CoO was discussed from the perspective of usability within Axis. The analysis resulted in the identification of limiting factors of using ML in the context of CoO within Axis, as well as recommendations concerning the use of ML and CoO within Axis Operations. Through this approach, the purpose of exploring the use of ML in the context of validating the CoO of a product has been fulfilled.

9.2 Answering the research questions

How can ML be used to validate a product's CoO at Axis Communications?

The general research question was explored through answering the three sub-questions of this study.

RQ1 Which data is needed for a successful ML implementation in the Axis context?
The strength of ML is in its capacity to identify and learn from patterns in data. As such, an ML model is dependent on both the quantity and the quality of the data it has been trained on. The data need to be of big enough quantities in order for it to include a sufficient number of examples of both each label and feature. The quality of data considers factors such as reliability and relevance to the problem. Reliability of the data ensures that the sample dataset used for training and testing of the model is an accurate representation of the full dataset and context of the problem. Relevance of the data can increase the likelihood that patterns are learned of the requested problem. In the case of Axis, data fields from the ERP-system identified as relevant for the topic of CoO are described in section 5.1.5.

As analyzed in section 7.3, having well-balanced datasets on feature level might be a success factor. Having a very unbalanced dataset on feature level, not true to the real-world distribution, has shown to prevent the model from learning the patterns.

The data used and needed in the Axis context is further analyzed in section 6.1, as well as illustrated in figure 6.4. The limitations of the data available was discussed in section 8.1.

RQ2 How can an ML model be created to validate CoO?
Developing an ML system is an explorative process comprised of steps which are iterated over to find the best possible model, given the time and resources available. However, it cannot beforehand be guaranteed that an ML model will be successful.

In order to answer the research question, the workflows of model development from three significant actors within the field of ML were summarized into a generalized workflow of developing ML models, presented in detail in chapter 3. The steps of the workflow are: (1) Problem framing, (2) Preparing data and features, (3) Training the model, (4) Evaluating the model, (5) Improving the model, (6) Iterate to find best model, (7) Use the model for predictions.

The workflow was then applied in an attempt to develop an ML model to validate the CoO of products at Axis. The development of the ML model is described in chapter 6, followed by results and analysis of the model in chapter 7. Limitations of the developed model are discussed in section 8.1.

RQ3 How feasible is it to implement an ML solution for validating CoO at Axis?
Although the use of ML to validate CoO of products is theoretically promising, it is not practically feasible in the context of Axis Operations. This study has identified four limitations to the implementation of an ML system for validating CoO at Axis:

(1) Lacking storage of historical data
(2) Difficulties in retrieving data from the ERP-system
(3) Lacking experience and knowledge of how to analyze input data for retraining
(4) Lacking experience and knowledge of how to interpret and validate the result

The limitations are discussed in detail in section 8.2.

9.3 Recommendations for Axis

From the work, investigations and results made in this study, recommendations for Axis have been identified. As this study came to the conclusion of ML not being a viable option in the context of verifying a product's CoO within Axis Operations, the following recommendations are presented in order to potentially make it easier in the future to both solve the CoO problem, as well as exploring the use of ML in other areas of Axis Operations further.

Revise data collection and data availability: One of the bigger challenges which arose both during the preparation of and during this study was the availability of data. Since all kinds of ML models rely on the quantity and quality of the input data, availability of data in turn becomes a key enabler of exploring ML. As discussed in section 8.2, two factors can ease the limitation of data availability at Axis Operations: (1) collection of data, and (2) retrievability of data. In many cases today, collection of data is limited to retrieving current data. By implementing storage solutions for historical data, data can be used to support future analysis and research. The second factor of data availability is the process of retrieving data. In the case of using CoO-data from the ERP-system, it was difficult to retrieve the data in a fast and easy way. If data is too difficult to obtain, it might obstruct development and retraining of ML models.

Review support of different CoO usages: Even though there is only one data field for current CoO in IFS, CoO information may be used differently depending on which department of Axis that is retrieving it, as discussed in section 8.2. Due to factors such as lead times and reverse logistics, these different usages of CoO might at times result in the departments requesting CoO for two separate batches of SUs, which then can have

different CoO. ML is not useful in this context. In order to minimize risk of corrupting the CoO-data, it is instead relevant to identify the different usages of CoO, and provide an ERP-solution to support them.

Explore automation, instead of ML: As discussed in the section 8.2, the practical use of ML in verifying CoO is limited due to data availability, data accessibility and knowledge-factors. For this reason, the authors recommend Axis to shift focus from verification through ML, but to instead explore a preemptive approach to increased data quality through automation. In the context of verifying CoO of products, all steps of the definition are firmly grounded in predefined rules. However complex the rules might be depending on product and market, the rules can still be described unambiguously. Since the rules are clear, as described in the initial empirical mapping, it is possible that defining CoOs can be automated. Replacing the manual task of defining CoO with automation could lower the risk of the human factor corrupting the data of the ERP-system. An additional benefit of automation compared to ML is interpretability. Since automation is rule-based, the logic is easier to understand for humans. In turn, better interpretability has the potential to lead to better preconditions for maintaining the solution and diminish the need for available data.

9.4 Areas for future research

In the context of using ML to validate a product's CoO, there are a number of improvements and areas for future research which could be developed further:

Elaborated feature processing and feature selection: The performance of an ML model can be altered through feature processing and feature selection. This was not fully explored within the scope of this study. As mentioned in section 8.1, simplifications were made during the feature processing of model development. It might therefore be of interest to increase the number of features used, and thereby increase the information given to an ML model.

Explore possibilities of multi-class classification: ML can potentially be used to predict the CoO of a product, instead of predicting correctness of the CoO existing in the ERP-system. Instead of framing the problem as a binary classification problem as done in this study, it could be framed as a multi-class single-label problem. As discussed in section 6.2, the current data in the ERP-system is not enough to support the training of a multi-class single-label problem. Future research would therefore include exploring alternative

data sources, such as the information flow of PTG (XML-files) or information from customs, such as information regarding HS codes.

The impact of concept drift: As mentioned in the discussion in section 8.2, it can be challenging to define a product's CoO in a global setting, due to the ever changing and mutable nature of the macro-environmental context. In order to use ML in a changing context, updated data of the new state must be available for retraining the model. In the case of Axis, it might be of interest to further research how this challenge of changing distributions in the underlying data (so called concept drift) can be met.

10 References

Amazon Web Services, 2016. Amazon Machine Learning: Developer Guide [WWW Document]. URL https://docs.aws.amazon.com/machine-learning/latest/dg/what-is-amazon-machine-learning.html (accessed 3.5.19).

Axis Communications, 2018a. Project Overview Document: TAACOO.

Axis Communications, 2018b. PM: Country of Origin (CoO) - EU and US.

Axis Communications, 2019. Project Overview Document: The Pathfinder Project.

Axis Communications, n.d. Products & Solutions [WWW Document]. Axis Communications. URL https://www.axis.com/products-and-solutions (accessed 5.6.19).

Briscoe, E., Feldman, J., 2011. Conceptual complexity and the bias/variance tradeoff. Cognition 118, 2–16.

Deloitte, 2017. The Buy American Act: New focus on government contract compliance [WWW Document]. Deloitte. URL https://www2.deloitte.com/us/en/pages/regulatory/articles/2017-the-buy-american-act-new-focus-on-government-contract-compliance.html (accessed 5.6.19).

Denscombe, M., 2010. Good research guide: For small-scale social research projects. Maidenhead : Open Univ. Press.

Ditzler, G., Roveri, M., Alippi, C., Polikar, R., 2015. Learning in Nonstationary Environments: A Survey. IEEE Comput. Intell. Mag. 10, 12–25.

Drakos, G., 2018. How to select the Right Evaluation Metric for Machine Learning Models: Part 3 Classification Metrics [WWW Document]. Towards Data Science. URL https://towardsdatascience.com/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-3-classification-3eac420ec991 (accessed 6.9.19).

Ekström, H., 2019. Informal talk about product data for the US market.

European Parliament, Council of the European Union, 2013. REGULATION (EU) No 952/2013 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 9 October 2013 laying down the Union Customs Code. Official Journal of the European Union.

Fortmann-Roe, S., 2012. Understanding the Bias-Variance Tradeoff [WWW Document]. URL http://scott.fortmann-roe.com/docs/BiasVariance.html (accessed 5.13.19).

Gard, T., 2019. Introduction to Axis Operations.

Google developers, 2019a. Machine Learning Glossary [WWW Document]. Google developers. URL https://developers.google.com/machine-learning/glossary/?authuser=1 (accessed 5.6.19).

Google developers, 2019b. Machine Learning: Problem Framing [WWW Document]. Google developers. URL https://developers.google.com/machine-learning/problem-framing/?authuser=1 (accessed

5.23.19).
Google developers, 2019c. Machine Learning: Rules of ML [WWW Document]. Google developers. URL https://developers.google.com/machine-learning/guides/rules-of-ml/ (accessed 5.23.19).
Google developers, 2019d. Machine Learning: Crash Course [WWW Document]. Google developers. URL https://developers.google.com/machine-learning/crash-course/representation/qualities-of-good-features (accessed 5.7.19).
Gustafsson, R., 2019. Technical advisory.
Hevner, A.R., 2007. A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19, 4.
Hjelmström, M., 2019. Interview regarding the CoO context and requirements of an ML model.
Holmstrom, J., Ketokivi, M., Hameri, A.-P., 2009. Bridging Practice and Theory: A Design Science Approach. Decision Sciences 40, 65–87.
Höst, M., Regnell, B., Runesson, P., 2006. Att genomföra examensarbete, 1. Studentlitteratur AB, Lund.
Jesson, J., Matheson, L., Lacey, F.M., 2011. Doing Your Literature Review: Traditional and Systematic Techniques. SAGE.
jupyter, 2019. jupyter [WWW Document]. jupyter. URL https://jupyter.org/ (accessed 5.15.19).
Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Presented at the International Joint Conference on Artificial Intelligence, Computer Science Department.
Krensky, P., Hare, J., 2018. Hype Cycle for Data Science and Machine Learning, 2018. Gartner.
Lilja Ivarsson, G., Kos-Hansen, O., 2019. Problem mapping.
Lindroth, R., 2019. Introduction to Axis.
Markham, K., 2015. Comparing Model Evaluation Procedures [WWW Document]. Github. URL https://github.com/justmarkham/DAT8/blob/master/other/model_evaluation_comparison.md (accessed 5.15.19).
Markham, K., 2018. Evaluating a classification model [WWW Document]. Github. URL https://github.com/justmarkham/scikit-learn-videos/blob/master/09_classification_metrics.ipynb (accessed 5.15.19).
Mathworks Inc, 2016a. Introducing Machine Learning.
Mathworks Inc, 2016b. Getting Started with Machine Learning.
Mathworks Inc, 2019a. Applying Supervised Learning.
Mathworks Inc, 2019b. Mastering Machine Learning: A Step-by-Step Guide with MATLAB.
Mathworks Inc, n.d. Supervised Learning Workflow and Algorithms [WWW Document]. Mathworks Inc. URL https://se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-

algorithms.html (accessed 5.26.19).

McKinney, W., 2019. pandas: powerful Python data analysis toolkit [WWW Document]. URL https://pandas.pydata.org/pandas-docs/stable/ (accessed 5.13.19).

Nugues, P., 2019. Lecture in course EDAF70 Artificial Intelligence, Lund University.

Olander, M., 2019. Presentations of situation assessment of Country of Origin data.

Olhager, J., 2019. Introduktionskurs till examensarbeten inom teknisk logistik.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Powers, D.M.W.P., 2007. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation.

PwC, 2018. PwC Forensic Services: Up to Speed - Trade Agreements Act Compliance. PwC.

Rosengren, K.E., Arvidson, P., 2002. Sociologisk metodik. Liber.

Salian, I., 2018. NVIDIA Blog: What's the Difference Between Supervised & Unsupervised Learning? [WWW Document]. The Official NVIDIA Blog. URL https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/ (accessed 6.15.19).

scikit-learn, 2019a. scikit-learn user guide [WWW Document]. URL https://scikit-learn.org/stable/user_guide.html (accessed 4.16.19).

scikit-learn, 2019b. Choosing the right estimator [WWW Document]. scikit-learn. URL https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html (accessed 2.15.19).

Silverman, D., 2010. Doing Qualitative Research: A Practical Handbook. SAGE.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45, 427–437.

Takeda, H., Veerkamp, P., Tomiyama, T., Yoshikawa, H., 1990. Modeling Design Processes. AI Magazine 11, 37–48.

Vellido, A., Martín-Guerrero, J.D., Lisboa, P.J.G., 2012. Making machine learning models interpretable. In: ESANN. Citeseer, pp. 163–172.

Voss, C., Tsikriktsis, N., Frohlich, M., 2002. Case research in operations management. Int. J. Oper. Prod. Manage. 22, 195–219.

World Customs Organization, n.d. Rules of Origin - Handbook [WWW Document]. URL http://www.wcoomd.org/~/media/wco/public/global/pdf/topics/origin/overview/origin-handbook/rules-of-origin-handbook.pdf (accessed 4.17.19a).

World Customs Organization, n.d. Partner Organizations [WWW Document]. World Customs

Organization. URL http://www.wcoomd.org/en/about-us/partners/international_organizations.aspx (accessed 5.6.19b).

World Customs Organization, n.d. WCO in brief [WWW Document]. World Customs Organization. URL http://www.wcoomd.org/en/about-us/what-is-the-wco.aspx (accessed 5.6.19c).

World Customs Organization, n.d. What is the Harmonized System (HS)? [WWW Document]. World Customs Organization. URL http://www.wcoomd.org/en/topics/nomenclature/overview/what-is-the-harmonized-system.aspx (accessed 5.6.19d).

World Trade Organization, 2019. The WTO and other organizations [WWW Document]. World Trade Organization. URL https://www.wto.org/english/thewto_e/coher_e/coher_e.htm (accessed 5.15.19).

World Trade Organization, n.d. Trade facilitation [WWW Document]. World Trade Organization. URL https://www.wto.org/english/tratop_e/tradfa_e/tradfa_e.htm (accessed 4.16.19a).

World Trade Organization, n.d. Technical Information on Rules of Origin [WWW Document]. World Trade Organization. URL https://www.wto.org/english/tratop_e/roi_e/roi_info_e.htm (accessed 4.17.19b).

World Trade Organization, n.d. WTO Agreement on Rules of Origin [WWW Document]. URL http://www.wcoomd.org/-/media/wco/public/global/pdf/topics/origin/overview/wto-agreement.pdf?db=web (accessed 4.17.19c).

World Trade Organization, n.d. WTO IN BRIEF [WWW Document]. World Trade Organization. URL https://www.wto.org/english/thewto_e/whatis_e/inbrief_e/inbr_e.htm (accessed 5.6.19d).

World Trade Organization, n.d. Legal texts: the WTO agreements [WWW Document]. World Trade Organization. URL https://www.wto.org/english/docs_e/legal_e/ursum_e.htm (accessed 5.6.19e).

Yin, R.K., 2009. Case Study Research: Design and Methods. SAGE.

Yue, Y., Li, Y., Yi, K., Wu, Z., 2018. Synthetic Data Approach for Classification and Regression. Presented at the 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP), IEEE.

Appendix

A.1 List of conducted interviews

Name	**Title of interviewee**	**Subject**	**Type of interview**	**Date**
Tobias Gard	Sales & Operations Planner	Introduction to Axis Operations	Unstructured	190124
Robert Lindroth	Operations Development Manager	Introduction to Axis	Unstructured	190130
Gisela Lilja Ivarsson & Olga Kos-Hansen	Customs Advisor & Trade Compliance Advisor	Problem mapping	Semi-structured	190313
Matilda Hjelmström	Supply Chain Development Manager	Problem mapping	Semi-structured	190312, 190314, 190318
Robin Gustafsson	System Developer	Technical support	Unstructured	190212
Henrik Ekström	Purchaser	CoO-data cleaning	Unstructured	190412
Robin Gustafsson	System Developer	Validate the evaluation procedure, metrics and result	Unstructured	190510

A.2 Interview guide

Interview guide with interview questions for semi-structured interviews with Axis employees.

Start of interview

Introducera oss själva.
Fråga om medgivande till inspelning.
Berätta om syftet med intervjun.

The interviewee

- Vad är din roll på Axis?

Involvement with CoO

- Vad är din roll i CoO-frågan?
- Vilka andra är involverade i frågan?

The CoO problem

- Skulle du kunna börja med att presentera problematiken kring CoO?
    - Finns det exempel på när det blir problem?
- Vilken roll kan maskininlärning ha, som inte täcks av andra lösningar (exempelvis automatisering)?

Definitions

- Hur definieras CoO för Axis produkter?
- Vad ingår i slutmontering?
- Vilka nivåer kan en produkt vara på? SU - PU - UA?
- Vilka nivåer kan en komponent vara på som går in till CLC-ATP?
    - Vilka typer av SU kommer in?

Understanding of the CoO context

- Hur och när och till vad används CoO?
- Vilka externa krav, exempelvis myndighetskrav, finns?
    - Hur ser skillnaderna ut i europeiska och amerikanska regelverk?
    - Hur påverkas Axis av regelverken?
- Vilka möjliga försörjningskedjor finns till den amerikanska marknaden?

Derivation of the CoO of a product today

- Hur sker arbetet med CoO-data i affärssystemet idag?
    - Hur introduceras data?
    - Vem introducerar data var?
    - Vilka moment är mänskliga moment?
- Hur tas beslut om CoO härleds från leverantörens landsadress (metod 1) eller landet för tillverkning (metod 2)?
- Hur kommer det sig att datan i IFS/Supplier for Purchased Parts (metod 2) har högre risk att vara inkorrekt?
- Vi har förstått det som att en produkt kan var SU, PU eller UA när de kommer in i ett CLC, stämmer detta?
    - Kan du rita upp hur försörjningskedjan ser ut, utifrån problematiken med CoO?
    - Hur ser beslutsprocesserna för CoO för respektive försörjningskedja ut?
    - Var sker största värdeökningen för respektive?
    - Var i kedjorna bedöms CoO definieras?
- Var i beslutsprocessen för CoO finns osäkra moment som ML eventuellt kan vara en del i att hantera?

Possible solution, specification and evaluation

- I beskrivningen avgränsas uppgiften till att endast hantera produkter som initialt har site=INC. Kan vi behålla den begränsningen genom resten av projektet?
- Vilka egenskaper önskas i lösningen? Exempelvis spårbarhet, effektivitet och användarbarhet?
- Finns det krav på lösningen, exempelvis på prestanda och tidsbegränsning?
- Finns det begränsningar för prototypen?
- Hur kommer lösningen utvärderas från Axis sida?
- Vad är det förväntade användningsområdet för prototypen?
    - Vem ska använda den?
    - Finns program som lösningen ska kunna kommunicera med?
- Hur finns datan tillgänglig?
    - Vilket format har datan?
    - Var finns data som kan vara relevant?
- Vilken kvalitet har datan?
- Hur mycket korrekt data finns tillgänglig?
- När kan träningsdata vara redo?
- Hur många produkter finns? Hur många omfattas av problematiken (säljs till den amerikanska marknaden)?

- Hur många länder är produkterna/leverantörerna fördelade på?

End of interview

Berätta att vi kommer återkoppla svaren för att få bekräftat att vi uppfattat det rätt.

A.3 Countries in original dataset

Table and graphical representation of the countries represented in the initial dataset. Along each country and variable is the number of samples for each. All variables were assigned the default value of - MISSING - when data is not available.

Table A.3.1. Countries present in the variables representing a country or CoO, as well as number of data entries for each country.

CoO of product *Inventory part*		**CoO of component** *Inventory part*		**Country of supplier of component** *Supplier*		**CoO of component** *Supplier for purchased parts*	
- MISSING -	15	- MISSING -	26	- MISSING -	53	- MISSING -	141
Bulgaria	1	Austria	2	Bulgaria	2	Austria	2
Canada	2	Bangladesh	1	China	151	Bangladesh	1
China	380	Bulgaria	1	Czech Republic	196	Bulgaria	1
Czech Republic	206	Canada	2	Denmark	4	Canada	2
Denmark	3	China	381	Germany	27	China	355
Estonia	18	Czech Republic	195	Hong kong	106	Czech Republic	184
Finland	1	Denmark	3	Hungary	2	Denmark	3
France	1	Estonia	18	Ireland	14	Estonia	18

Germany	27	Finland	1	Israel	17	Finland	1
Hungary	10	France	1	Italy	24	France	1
Indonesia	3	Germany	26	Japan	36	Germany	28
Israel	1	Hong kong	1	Korea, Republic of	7	Hong kong	1
Italy	21	Hungary	2	Latvia	3	Hungary	2
Japan	136	Indonesia	3	Malaysia	52	Indonesia	3
Korea, Republic of	11	Israel	1	Netherlands	53	Israel	1
Lao People's Democratic Republic	1	Italy	23	Poland	185	Italy	13
Malaysia	69	Japan	137	Sweden	136	Japan	126
Mexico	48	Korea, Republic of	8	Taiwan	62	Korea, Republic of	8
Philippines	8	Lao People's Democratic Republic	2	Thailand	117	Lao People's Democratic Republic	2
Poland	173	Malaysia	62	United Kingdom	67	Malaysia	42

Romania	1	Mexico	47	United States	187	Mexico	47
Sweden	104	Philippines	8			Philippines	8
Taiwan	70	Poland	200			Poland	195
Thailand	103	Romania	3			Romania	3
United Kingdom	60	Sweden	69			Sweden	65
United States	27	Taiwan	65			Taiwan	59
Viet Nam	1	Thailand	117			Thailand	104
		United Kingdom	57			United Kingdom	47
		United States	39			United States	38
Grand Total	**1501**	**Grand Total**	**1501**	**Grand Total**	**1501**	**Grand Total**	**1501**

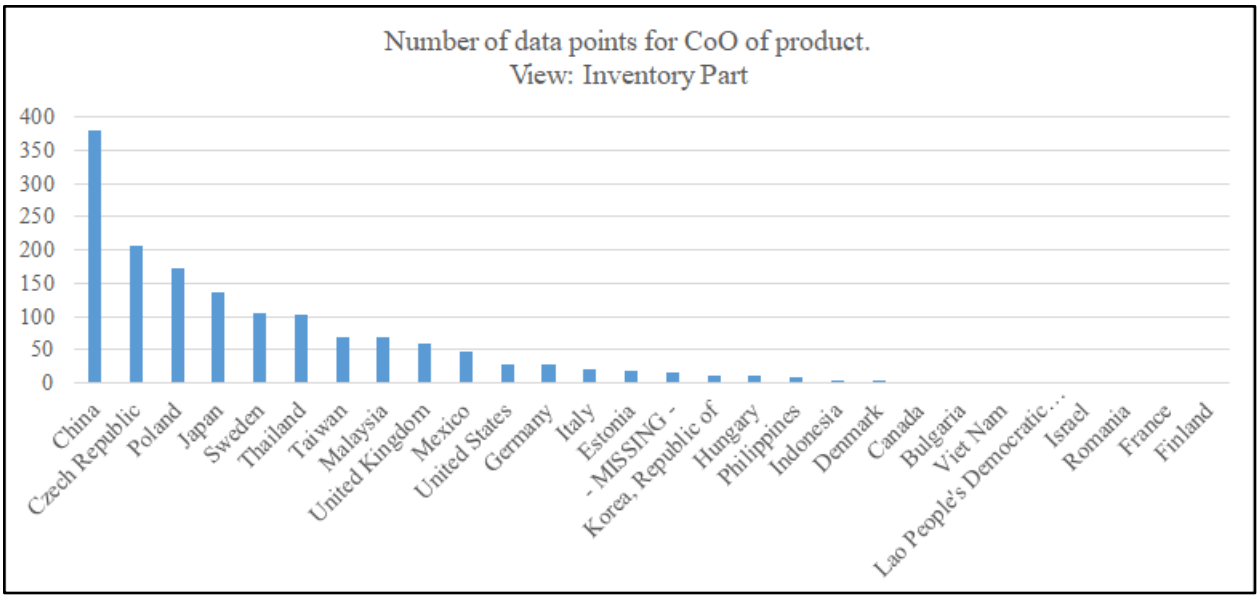

Figure A.3.1. Graphical representation of number of data entries for each CoO of product in Inventory part.

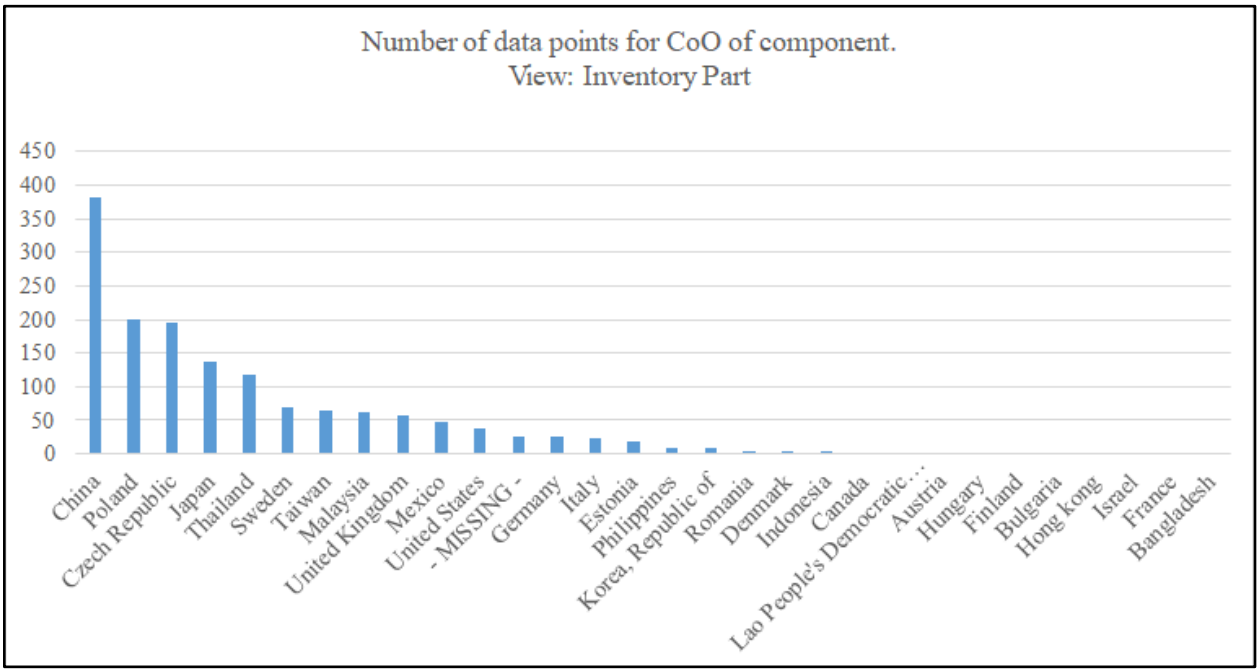

Figure A.3.2. Graphical representation of number of data entries for each CoO of component in Inventory part.

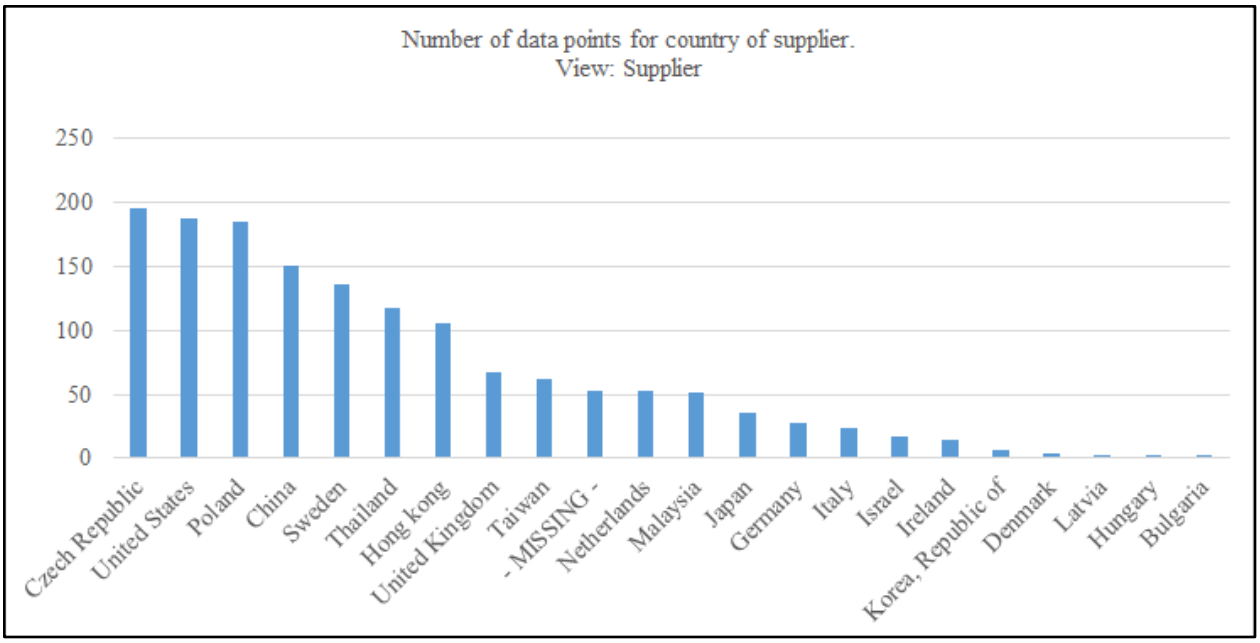

Figure A.3.3. Graphical representation of number of data entries for each country of supplier in Supplier-view.

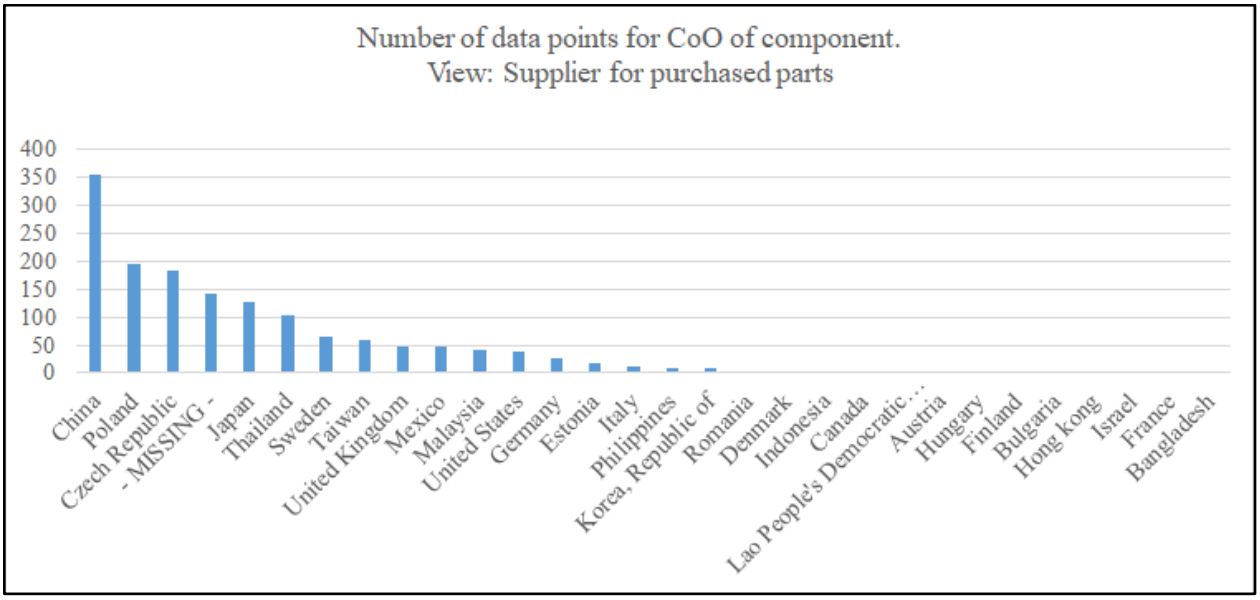

Figure A.3.4. Graphical representation of number of data entries for each CoO of component in Supplier for purchased parts.

A.4 Expanded results

Table A.4.1. Train and test accuracies with variances for the models.

	Train accuracy		**Test accuracy**	
Model	*Mean*	*Variance*	*Mean*	*Variance*
DecisionTreeClassifier	0.987581	0.000001	0.967770	0.000107
RandomForestClassifier	0.987075	0.000000	0.968459	0.000037
KNeighborsClassifier	0.985829	0.000004	0.893110	0.000524
SVC	0.987698	0.000000	0.965313	0.000087

Table A.4.2. Confusion matrices for the feature CoO (IFS/Inventory part) for the model using the RandomForrestClassifier. The TP, TN, FN and FP are labeled as (actual label, predicted value).

Country of origin	**Number of observations**	**True positive** *(1, 1)*	**True negative** *(0, 0)*	**False negative** *(1, 0)*	**False positive** *(0, 1)*
- MISSING -	15	0	15	0	0
Bulgaria	64	0	63	1	0
Canada	64	0	62	2	0
China	417	330	64	12	11
Czech Republic	254	197	56	1	0
Denmark	60	3	57	0	0
Estonia	82	18	63	0	1
Finland	52	0	51	1	0
France	48	0	47	1	0
Germany	84	25	56	2	1
Hungary	63	6	49	4	4
Indonesia	59	3	56	0	0
Israel	61	0	60	1	0
Italy	67	13	54	0	0
Japan	173	124	48	0	1
Korea, Republic of	63	7	53	2	1
Lao People's Democratic Republic	56	0	55	1	0
Malaysia	115	43	61	7	4
Mexico	98	46	50	1	1

Philippines	62	8	54	0	0
Poland	217	163	51	2	1
Romania	66	0	65	1	0
Sweden	151	94	45	5	7
Taiwan	117	61	53	0	3
Thailand	149	88	51	5	5
United Kingdom	104	50	54	0	0
United States	92	24	67	1	0
Viet Nam	1	0	1	0	0

Table A.4.3. Results for the feature current CoO of product (IFS/Inventory part) for the model using the RandomForrestClassifier, including number of observations labeled correct and faulty in the original dataset, null accuracy, test accuracy, recall and specificity for each of the CoOs present in the original dataset.

Country of origin	Number of examples with label correct	Number of examples with label faulty	Null accuracy	Test accuracy	Recall	Specificity
- MISSING -	0	15	1	1	0	1
Bulgaria	1	63	0.984	0.984	0	1
Canada	2	62	0.969	0.969	0	1
China	380	37	0.911	0.945	0.965	0.853
Czech Republic	206	48	0.811	0.996	0.995	1
Denmark	3	57	0.95	1	1	1
Estonia	18	64	0.78	0.988	1	0.984
Finland	1	51	0.981	0.981	0	1
France	1	47	0.979	0.979	0	1
Germany	27	57	0.679	0.964	0.926	0.982
Hungary	10	53	0.841	0.873	0.6	0.925
Indonesia	3	56	0.949	1	1	1
Israel	1	60	0.984	0.984	0	1

Italy	21	46	0.687	1	1	1
Japan	136	37	0.786	0.994	1	0.98
Korea, Republic of	11	52	0.825	0.952	0.778	0.981
Lao People's Democratic Republic	1	55	0.982	0.982	0	1
Malaysia	69	46	0.6	0.904	0.86	0.938
Mexico	48	50	0.51	0.98	0.979	0.98
Philippines	8	54	0.871	1	1	1
Poland	173	44	0.797	0.986	0.988	0.981
Romania	1	65	0.985	0.985	0	1
Sweden	104	47	0.689	0.921	0.949	0.865
Taiwan	70	47	0.598	0.974	1	0.946
Thailand	103	46	0.691	0.933	0.946	0.911
United Kingdom	60	44	0.577	1	1	1
United States	27	65	0.707	0.989	0.96	1
Viet Nam	0	1	1	1	0	1