JOHANNES KEPLER
UNIVERSITY LINZ

Author
Giel van Bree, BSc

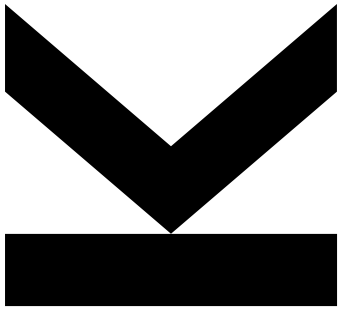
Submission
Institute of Business Informatics – Data & Knowledge Engineering

Thesis Supervisor
Assoz.-Prof. Mag. Dr. Christoph Schütz

Assistant Thesis Supervisor
Simon Staudinger, MSc

Month Year
November 2024

Using Machine Learning to Identify Incorrect Value-Added Tax Reports



Master Thesis
to obtain the academic degree of

Master of Science
in the Master's Program

Economic and Business Analytics

JOHANNES KEPLER UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Preface

This thesis was conducted in cooperation with BDO, a consulting company which is specialised in tax services. Professional support was provided by BDO mainly through two domain experts: Felix Burgstaller for data science expertise, and Felix Schiff for tax knowledge.

This thesis is the basis for a publication in the Proceedings of the 30th Edition of the Americas Conference on Information Systems (AMCIS 2024) (van Bree, Staudinger, Burgstaller, Schiff, & Schütz, 2024).

Abstract

Many companies and organisations worldwide bear the legal responsibility to collect and report value-added tax (VAT) over its sales. Typically, tax reporting is done manually by accountants. The complexities in tax legislation make the reporting of VAT error-prone and time-consuming. As a consequence, it occurs that tax is occasionally reported falsely. A potential solution to improve the compliance with tax regulations is a tax compliance system, i.e., a system that can automatically identify tax statements whereof the tax conditions are potentially reported incorrectly. The objective of this thesis is to design and implement a data-driven tax compliance system using machine learning (ML) techniques, specifically for VAT on both incoming and outgoing invoices. The purpose of the tax compliance system is to support accountants in identifying and so preventing incorrect VAT reporting. Using classification, the tax conditions of invoices can be predicted. The thesis is conducted in cooperation with BDO, a consulting company which is specialised in tax consultancy services, including VAT reporting. Factors that determine the VAT payable are identified by a VAT domain expert. Real-world enterprise resource planning (ERP) data is used to predict the combination of tax conditions, i.e., the tax code. A framework is presented, where tax code deviations are considered anomalies, that is, potentially incorrect tax codes. A tax code deviation is an invoice whereof the predicted tax code deviates from the tax code assigned by the accountant. The data is pre-processed and a classification model is built for outgoing invoices, which can predict tax codes accurately with an accuracy of 98.9%. The classifier has the ability to learn from an ongoing stream of data and from human-generated feedback. This framework is implemented in a system, wherein the classification model actively attempts to identify incorrect VAT reporting by predicting tax codes of a stream of invoice data. Tax code deviations are highlighted in the system, such that users can interactively verify the correct tax code to prevent incorrect VAT reporting. The definite tax codes, as entered by the user, are continuously used to retrain the classifier. The design artifact is this ML-based system to increase VAT compliance.

Keywords: data mining, machine learning, tax compliance, value-added tax

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

1. Introduction

The collection and reporting of value-added tax (VAT) is a legal responsibility for many companies worldwide. Companies can either decide to do the reporting themselves, usually with the use of professional accountants, or to outsource this time-consuming burden to tax consulting companies. Tax legislation is complicated, especially in the globalised world where companies often operate internationally and have to comply with local legislation. Besides, tax legislation can change and

In most companies and tax consulting companies, accountants still do the VAT reporting manually, where erroneous reporting can occur. The problem statement of the thesis is

VAT reporting is prone to errors.

Although accountants are professionals in the domain of VAT reporting, the problem statement holds true. There exists a potential to increase compliance with VAT legislation, or VAT compliance, by the usage of modern data-driven techniques. The thesis aims to develop a system for VAT compliance using data mining methods. The research question of the thesis is therefore as follows:

How can data mining techniques be used to increase VAT compliance?

While the techniques to be applied are data mining techniques, the research methodology to answer the research question is design science.

> *"Design science is concerned with the study, investigation, and accumulation of knowledge about the design process and its constituent operations. It aims to collect, organize, and improve those aspects of thought and information which are available concerning design and to specify and carry out research in those areas of design which are likely to be of value to practical designers and design organisations." (Gregory, 1966, pp. 323-330)*

In the field of design science, the design artifact is an innovative product that provides the solution to a previously unsolved problem, and can be a new application of existing knowledge (Hevner, Salvatore, Park, & Ram, 2004). In the context of this thesis, the artifact must be a solution that helps to increase VAT compliance. The artifact should contribute to the scientific fields of tax compliance and applications of data mining.

The thesis project has been conducted in collaboration with BDO Consulting GmbH, a division of BDO. BDO is a globally operating consulting company. Its consulting activities include, among others, accounting and tax auditing (BDO, 2023). In conclusion, besides aiming to contribute to the mentioned scientific fields, the research has real-world applications and business value. The work is based on real-world data.

The chapter continues with an introduction to *Value-Added Tax*, to obtain an understanding of the background of VAT. Subsequently, a *Motivation* for the thesis is presented.

1.1. Value-Added Tax

VAT is the most common consumption tax worldwide (Sarmento, 2023). Per late 2020, 170 countries issue VAT (OECD, 2022). It is a general consumption tax and it is proportional to the sales price of the good or service. VAT is an indirect tax, meaning that the person liable for the tax and the person bearing the tax are not the same. Namely, VAT aims to tax the final consumer's consumption rather than any intermediaries involved in the distribution chain. Its importance for government budgeting is evident: VAT revenues generated by European Union (EU) countries together sum up to over €1,000 billion per year, contributing for 17.8% of their total tax revenues (Eurostat, 2022). The collection and remitting of these revenues are carried out by the intermediaries, named taxable persons in the context of VAT (facultas.wuv, 2018).

Van Doesem & Nellen (2021) explain that taxable persons face legal obligations to periodically file VAT statements, i.e., practically they act as 'unpaid tax collectors' for the tax authorities, because they collect VAT from their customers in all stages of the production and distribution chain and they pay the collected VAT to the tax authorities. Taxable persons are not limited to businesses: Institutions and (non-profit) organisations are in most countries also required to pay VAT. The authors further explain that within the EU, VAT systems are to a certain extent harmonised through the EU VAT Directive. The Directive serves as a blueprint for how VAT should be applied within EU member states, with the objective to establish a common framework for VAT systems in the EU. It aims for a simple, fair, and consistent VAT system. However, the Directive mostly consists of guidelines, so member states are free to choose their implementation of the VAT Directive, causing certain complexities in VAT regulations to remain. The Directive allows for diversification in terms of applicable tax percentage among goods. Reduced tax rates may be applied to goods and services that are considered essential for human needs. EU member states may choose to regard certain groups of goods or services as zero-rated, meaning that no VAT is payable. Due to the EU VAT Directive, VAT reporting for trade between EU member states is simplified compared to trade with third states, since additional local complexities in legislation may apply.

Depending on local VAT legislation, many exceptions are possible. Nonetheless, transactions can be generalised into a few types of categories (van Doesum & Nellen, 2021). To better comprehend the context, the common types of transactions are illustrated, without delving into details of the complexity of VAT determination, because the focus of the thesis is not on the fiscal aspect.

National transactions: Trade between companies within a single nation is referred to as a national transaction. Whether a transaction is taxable and what tax percentage applies, depends on the local legislation for the type of goods.

Intra-community acquisitions and sales: Trades between companies located in different EU member states are called intra-community acquisitions and sales. Certain tax conditions apply to such trades due to the harmonisation of the EU tax legislation. Northern Ireland has a special status, because despite that Great Britain is not part of the EU, it is considered intra-community for VAT purposes.

Imports and exports: Typically, imports and exports are referred to as trades with businesses from foreign countries. However, in tax definitions and from an EU perspective, imports and exports are trades with businesses from non-EU countries.

Reverse charge: Normally, the liability to pay VAT lies with the seller. Under reverse charge, this liability is shifted to the recipient. This system applies under some circumstances. For example, when the transaction regards a mixture between a good and a service, e.g., when a good is not only bought but also assembled (called Werklieferung in Austrian tax definitions) (Austrian Ministry of Finance, 2023).

Chain and triangular transactions: Chain transactions are transactions where the good is moved from the first taxable person to the last taxable person, while at least one intermediary is involved in the distribution chain (EU VAT Directive, 2023). Intermediaries do not receive the good, however, they sell the good and thus act as an intermediate sales entity. A transaction is only a chain transaction if it is also a cross-border transaction, meaning that the taxable persons in the chain are from at least two different countries c_i with $i \in I$ and $|I| \geq 2$. Invoices are sent down the chain from entity e_n to entity e_{n+1}. *Figure 1.1* provides an illustration of the most basic version of a chain transaction. Note that this example includes only three entities e_n, but infinitely many entities can be involved theoretically, i.e., $n \in N$, $N = \{1, 2, \dots, \infty\}$, and $|N| \geq 3$. Also, more countries may be involved, i.e., $I = \{1, 2, \dots, \infty\}$. Transactions can be cross-border between any two entities e_n.

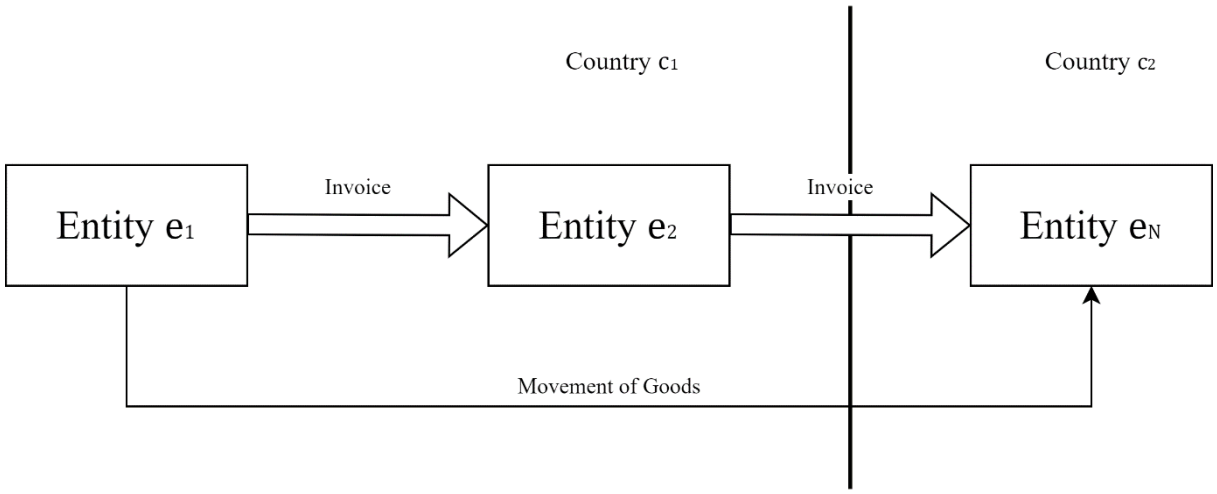


Figure 1.1: Example of a Chain Transaction

Triangular transactions are a subcase of chain transactions, that is, a more specific and more constrained version. Similar to chain transactions, the beginning and end entity e_n of the movement of good determines the length of the triangular transaction. However, there are specifically only three entities e_n involved, i.e., $|N| = 3$, as the name suggests. As a consequence, two invoices are involved in a triangular transaction. All three entities e_n are located in different countries and all these countries are EU countries u_j with $j \in J$ and $|J| = 3$. See *Figure 1.2* for an example of a triangular transaction.

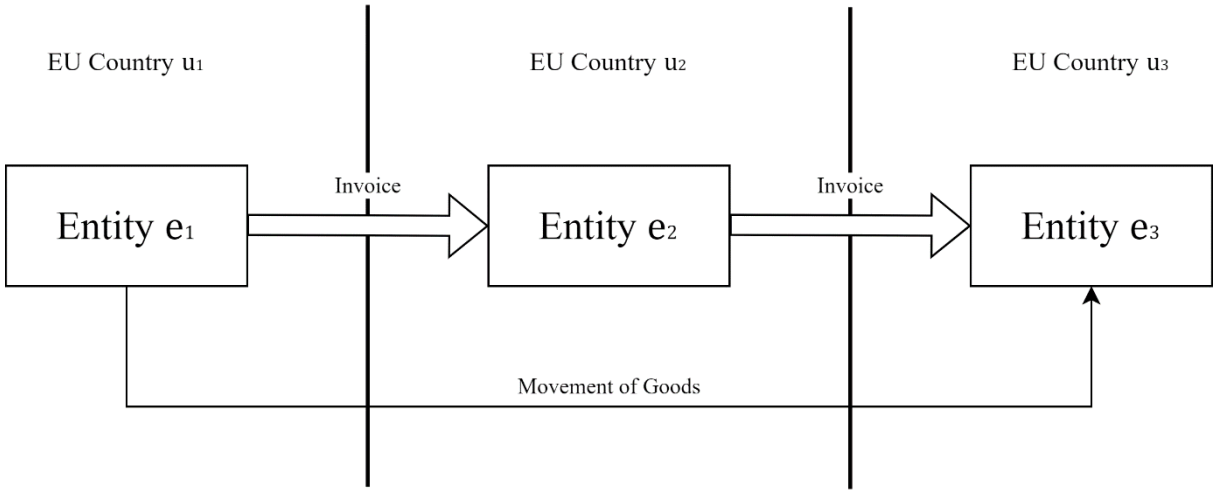


Figure 1.2: Example of a Triangular Transaction

VAT reports contain multiple tax conditions:

- The country where the VAT is to be reported
- The tax type (e.g., input tax, output tax)
- The tax rate (e.g., standard rate, reduced rate, zero rate)
- The tax event (e.g., for specific goods/services or special scenarios)
- Exemptions
- Deductibility
- Etc.

Moreover, less common tax conditions may exist in certain countries, depending on the local legislation. However, the tax conditions stated above generally cover the tax conditions of most countries, especially for EU countries, according to BDO's tax domain expert. Only when all tax conditions of a VAT report are correct, one can speak of a correctly reported VAT statement. In enterprise resource planning (ERP) systems, the combination of tax conditions is commonly stored in so-called tax codes, wherein a single tax code captures a unique combination of tax conditions. A more detailed explanation and examples of tax codes will be provided in *Data Understanding.*

The tax conditions applicable on a transaction depend on the local legislation. As every country has their own tax legislation, it is practically infeasible to state the total set of determinants for tax codes

globally. Nevertheless, the set of tax code determinants from an EU perspective is given by BDO's tax domain expert. Except for extremely rare cases, of which the tax domain expert is unaware, these are the determinants for tax codes from an EU perspective:

- Country of origin of the goods
- Country of destination of the goods
- Whether the transaction regards a good, service, or a mixture (Werklieferung)
- Whether the transaction is a chain or triangular transaction
- Whether the transaction is an input or output transaction
- Whether the seller or buyer is responsible for the transportation of the good
- Whether the good or service has a standard or reduced tax rate

1.2. Motivation

VAT reporting is prone to errors and there is demand for a VAT compliance system to limit erroneous VAT reporting. However, until now it is not clarified specifically why false VAT reporting is problematic. According to BDO's VAT domain expert, wrong VAT reporting can be distinguished into two categories: overreporting and underreporting. Overreporting refers to reporting a higher amount of VAT than legally required. Underreporting refers to the reverse: reporting a lower amount of VAT than what is legally required. Both types of wrong VAT reporting are problematic. For instance, with overreporting a business incurs more VAT costs than necessary. For underreporting, the underreported cash amount needs to be refunded to the tax authorities after the VAT statements have been inspected and underreporting is found. More critically, the taxpayer risks to be considered fraudulent by jurisdictive authorities, as the VAT payment laws have been violated. Typically, this only occurs when the underreporting is structural rather than incidental. If a taxpayer is proven fraudulent, significant fines can follow. The severity of these fines depends on the local legislation, but they are normally serious.

BDO Consulting GmbH, hereafter simply referred to as BDO, provides business-to-business tax consultancy services (BDO, 2023). Specifically relevant to this research are its value-added tax auditing services. Their main activities herein are the auditing and reporting of VAT statements for clients to the tax authorities. Clients provide the necessary documents and data and outsource the VAT reporting duty to BDO. Although the accountants are qualified and skilled, human VAT reporting remains error-prone, especially considering the complexity of tax legislation. To improve tax compliance and limit erroneous VAT reporting, the desire for a supportive tax compliance system was recognised within BDO, which can identify potentially wrongly reported VAT statements.

In conclusion, multiple motivations exist to increase VAT compliance. The problem statement – VAT reporting is prone to errors – has serious incentives to improve the situation by the design of an artifact.

2. State of the Art

The chapter *State of the Art* provides theories and background information about state-of-the-art techniques. First of all, an *Introduction to Data Mining* is given and examples of practical applications are given. Then, the data mining scope is funnelled to the data mining techniques applicable to the thesis. This is followed by the comparison of standard process models used for data mining projects, of which the *Cross-Industry Standard Process for Data Mining* is eventually chosen to use for this thesis. It is therefore further elaborated. At the end, existing *Related Literature* is presented and summarised.

2.1. Introduction to Data Mining

As a consequence of digitalisation and computerisation, we are experiencing an explosive growth of available data, particularly in businesses, but actually in most aspects of society (Han, Pei, & Tong, 2022, p. 1).

Data mining can be defined as

> *"the process of discovering interesting patterns, models, and other kinds of knowledge in large datasets." (Han, Pei, & Tong, 2022, p. 1)*

Furthermore, as Han, Pei, & Tong (2022, p. 2) state in their book *Data Mining: Concepts and Techniques*, data mining is often referred to as knowledge discovery from data (KDD). According to the authors, the goal of data mining is to discover patterns in large amount of data and use these insights. Han, Pei, & Tong (2022, pp. 5-10) explain that data mining tasks can be categorised into descriptive mining and predictive mining. As the names tell, descriptive mining characterizes properties of the data set, while predictive mining aims to discover patterns with the goal to make predictions. Data mining tasks include data summarisation, frequent pattern mining, classification and regression, cluster analysis, and outlier analysis. These data mining tasks are explained in *Table 2.1*.

Table 2.1: Data Mining Tasks (Han, Pei, & Tong, 2022, pp. 5-10)

Data Mining Task	Explanation	Example
Data Summarisation	According to Han, Pei, & Tong (2022, p. 6), summarizing multidimensional data at a high level in order to obtain insights in the data. It is necessary because it is tedious to go over the details of large data sets. Output of data summarisation is often visualised charts, e.g., pie charts, bar charts, histograms, line charts, etc.	In a logistics company, data summarisation can help to understand the distributions of the freight type, freight weight, destinations, etc.

Table 2.1: Data Mining Tasks (Han, Pei, & Tong, 2022, pp. 5-10)

Frequent Pattern Mining	According to Han, Pei, & Tong (2022, pp. 6-7), frequent pattern mining aims to discover patterns in data sets that occur frequently, as the name suggests. Such patterns are items which often appear together, so-called itemsets. With frequent pattern mining, interesting associations and correlations can be found in data sets.	Identifying which products are frequently purchased together in e-commerce. The results can be used for targeted advertising.
Classification and Regression	According to Han, Pei, & Tong (2022, pp. 7-9), classification is "the process of finding a model that describes and distinguishes data classes". By training a classification model on data objects of which the classes are labelled, the model learns to distinguish the different classes. A large variety of classification models exists, for example classification rules (IF-THEN rules), a decision tree, a mathematical formula, or advanced algorithms such as neural networks. Classification models predict categorial labels, whereas regression models predict continuous values. Often, classification is supported and preceded by feature selection, which is the identification of relevant attributes.	With the large availability of sensors and data in modern cars, classification algorithms may be able to predict a car part that is likely to be defect soon. The results can be used to notify the owner to do preventive maintenance.
Cluster Analysis	According to Han, Pei, & Tong (2022, p. 9), clustering analysis, or clustering, does not consult class labels. Often, class labels do not exist (yet) for data. Clustering generates class labels for data, categorizing similar data objects into one cluster. Clusters are usually determined by algorithms that use statistics, which maximize the intraclass similarity and minimize the interclass similarity, i.e., clusters are formed such that data objects within a cluster are similar, while having low similarity to data objects in other clusters.	Clustering customers into different customer segment of customers who are similar in their characteristics.
Outlier Analysis	According to Han, Pei, & Tong (2022, p. 10), an outlier is an unusual data object, being significantly different from most data objects. Outlier analysis, or anomaly mining, is the process of analysing outlier data. Outliers can be detected using statistics and distance measures.	Banks can use outlier analysis to detect credit card frauds, which are the outliers.

Han, Pei, & Tong (2022, pp. 12-15) explain that statistics and machine learning (ML) are the fundamentals of data mining. Furthermore, they explain statistical methods and machine learning in more detail. Namely, statistical methods are used throughout many data mining areas. ML focuses on how computers can learn from data using sophisticated algorithms. It is a fast-growing discipline, where state-of-the-art algorithms are developing rapidly. ML can be distinguished into two classical problems: Supervised learning and unsupervised learning. Whether ML is supervised or unsupervised depends on the availability of class labels. In supervised learning, class labels are required, whereas in unsupervised learning, class labels are not necessary. The typical examples for supervised learning and unsupervised learning are respectively classification and clustering.

The task of classification is particularly of interest to this thesis, since classification can be used to predict the combination of tax conditions, i.e., the tax code. Therefore, the concept of classification is reviewed in more depth, using the definitions from Han, Pei, & Tong (2022, pp. 239-289) for this paragraph. They define classification as “a form of data analysis that extracts models describing important data classes”. They explain that such models, called classifiers, predict categorical (discrete, unordered) class labels. Many classification tasks are binary classification tasks, meaning there are only two classes, e.g., ill or not ill. Multiclass classification tasks have more than two classes, e.g., the kind of illness. When the classes are not equally distributed, one speaks of an imbalanced classification task. For example, it is generally much more common to not be ill than to be ill, so this is an imbalanced classification problem. Classification is a form of supervised learning. The data classification process consists of two steps: A learning step and a classification step. In the learning step, or training step, the classifier is constructed. In the classification step, the classifier predicts class labels for the given data. For the training of the classifier, ML algorithms are often used. The authors explain that in order to rate the effectiveness of the built classifier, the classification model should be evaluated. For the evaluation, generally the following procedure is followed: At start, the data set is split into a training data set and a test data set. A data object belongs either to the training or the test data set. For the learning step, or training step, of the data classification, the training data is used. Then, the classifier predicts the class labels of the test data. The predicted class labels are compared with the actual class labels in order to calculate the accuracy of the classifier. In this context, an accurate prediction means that the predicted class label is equal to the class label given in the data. The reason for splitting the data set into training and test data sets is that the classifier tends to overfit the data. This means that the classifier learns certain anomalies which only exist in the training data but not in the general data. Therefore, the accuracy would likely be overestimated if there is no splitting between training and testing data. To estimate a reliable accuracy that is normally representative for new data objects, the method of splitting the data set into training and testing data is therefore a common practice in data classification.

2.2. Cross-Industry Standard Process for Data Mining

Data mining tasks are complex and require certain steps. Hence, data mining projects often follow data mining process models. Three frequently used data mining process models are:

- KDD
- SEMMA
- CRISP-DM

Although these data mining process models have substantial similarities, subtle differences exist. First of all, the knowledge discovery from data (KDD) process, developed by Fayyad et al. (1996), consists of the steps data selection, data pre-processing, data transformation, data mining, and interpretation/evaluation. The view of Han, Pei, & Tong (2022, pp. 2-4) slightly differs, as they consider the KDD process as a sequence of the following four steps: Data preparation (data cleaning, data integration, data transformation, and data selection), data mining, pattern/model evaluation, and knowledge presentation. The second well-known data mining process model is SEMMA, developed by the SAS Institute, a large statistics and business intelligence software producer. SEMMA stands for sample, explore, modify, model, and assess. Finally, the cross-industry standard process for data mining (CRISP-DM) is a process model developed by an industry consortium of leading data mining users, focusing on practicality for business implementation across industries, as the name reveals. Despite being developed in 1996, it is still the leading process model in data mining projects (Shearer, 2000). The six stages of CRISP-DM are business understanding, data understanding, data preparation, modelling, evaluation, and deployment. In the comparative review of Azevedo & Santos (2008), they conclude that SEMMA and CRISP-DM are both implementations of KDD.

Because of its widely proven practical feasibility for data mining projects across various industries, the CRISP-DM process is the ultimate process guideline to follow for this thesis. Saltz (2021) explored the strengths and weaknesses of CRISP-DM, and particularly praises five strengths: Common sense, cyclical, adoptable, right start, and flexible. Its key strength is that it is easy to understand. The critiques are especially regarding corporate implementation into teamwork and considering corporate stakeholders. Besides that, Saltz argues that CRISP-DM is documentation heavy, indicating that an enormous amount of details would need to be documented, if CRISP-DM is strictly followed. For that reason, some obvious and trivial details are not documented in this thesis. Yet, Saltz recognizes that CRISP-DM is the most frequently used framework for data mining projects. Wirth & Hipp (2000) conducted a data mining project and tested the usefulness of CRISP-DM. They separately documented the evaluation of CRISP-DM and drew the same conclusion as Saltz: CRISP-DM is easy to use. Furthermore, Wirth & Hipp conclude that CRISP-DM is useful for planning, documentation, and communication.

The phases of CRISP-DM – business understanding, data understanding, data preparation, modelling, evaluation, and deployment – are generally executed in the respective sequence, however, this sequence is not strict, as concluded by Wirth & Hipp (2000). Typically, it is even required to move back and forth between the CRISP-DM phases, because the outcome of a phase may require to do so. In *Figure 2.1*, the CRISP-DM process model is displayed. The arrows between phases indicate the flexibility of sequence of the phases. The outer arrows symbolize that data mining is cyclic. Namely, once a solution is implemented, data mining is not finished. A deployed data mining solution may trigger new insights and new business questions, which may be solved by a subsequent data mining project (Wirth & Hipp, 2000). Evidently, data is at the core of CRISP-DM.

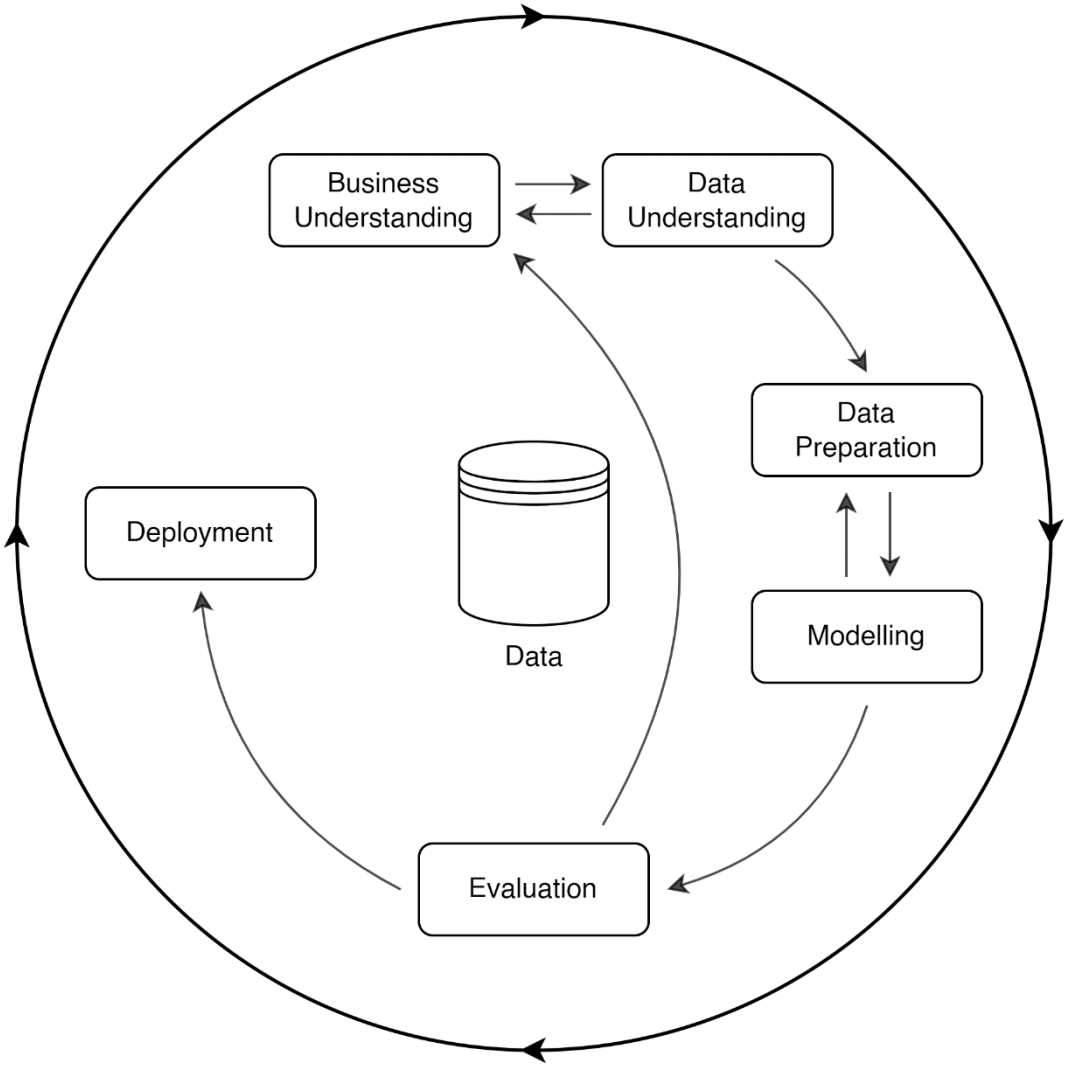


Figure 2.1: Cross-Industry Standard Process for Data Mining (Grigorev, 2021)

Below, the six phases of CRISP-DM are explained, using the framework described by Shearer (2000). CRISP-DM and its phases form the foundation for the structure of this thesis. The CRISP-DM reference model, as depicted in *Figure 2.2*, summarizes the six phases.

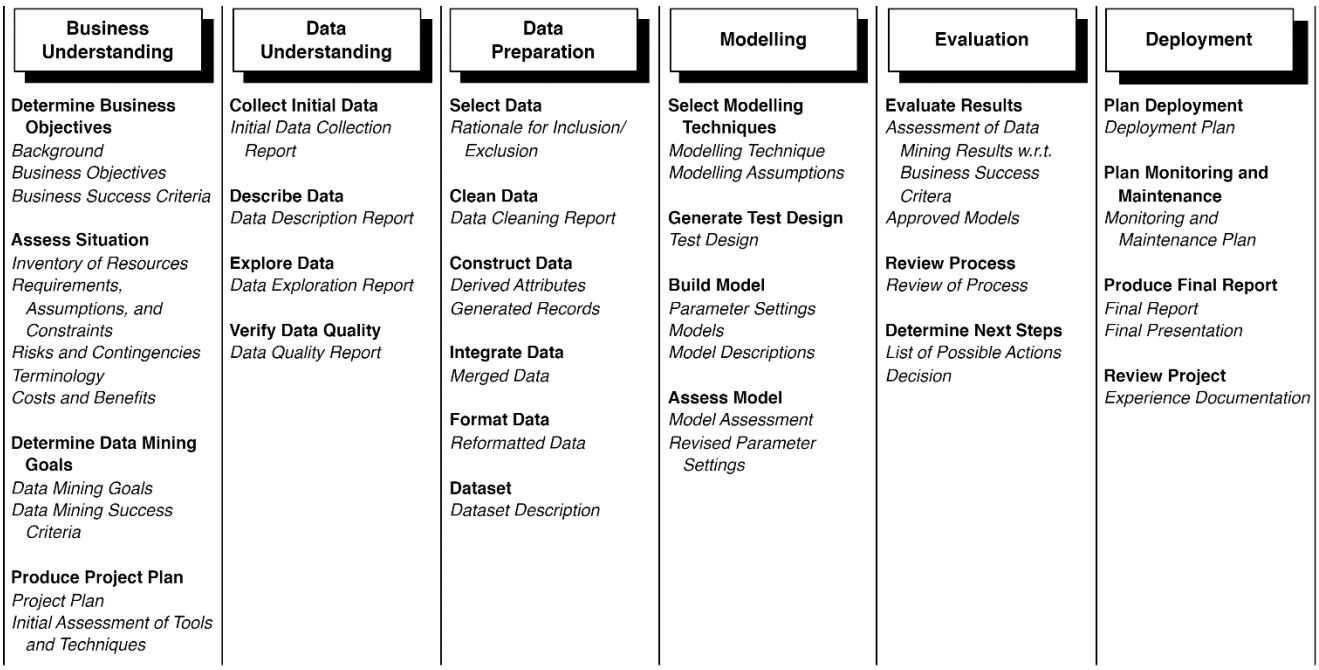


Figure 2.2: Cross-Industry Standard Process for Data Mining Reference Model (Shearer, 2000)

Business Understanding is the first phase of CRISP-DM and refers to understanding the objectives and requirements from a domain-specific perspective (Shearer, 2000). Every domain is different and entails different data understanding challenges. The domain knowledge is converted to a data mining problem. The business understanding phase can be divided into four subsections. Determine business objectives refers to the requirement definitions, i.e., what is ought to be achieved with the data mining. Furthermore, some domain knowledge is gathered to understand the particular background. Assess situation means that the complete environment is evaluated, ranging from available resources to constraints and assumptions. Potential risks are identified here. In determine data mining goals, the business objectives are stated in data mining terminology. Success criteria for these data mining goals are presented. These should be measurable, usually expressed in accuracy. Finally, in produce project plan, the intended project plan is presented, taking into account the current situation and objectives as assessed and determined earlier.

The first step in the **Data Understanding** stage is the initial data collection (Shearer, 2000). As the name says, the data is only initially collected here, and not yet integrated nor pre-processed. Then, in data description, the properties of the relevant data are described, e.g., the data type, data format, data quantity, etc.. Data exploration dives deeper into the data, meaning that the data is explored and initial data patterns are found, often by visualizing data. Finally, in data quality verification, the data quality is measured using metrics such as accuracy, completeness, consistency, timeliness, believability, and interpretability (Han, Pei, & Tong, 2022, pp. 55-56).

Data Preparation regards all necessary data preparation steps prior to the modelling (Shearer, 2000). Decisions, actions, and assumptions regarding data transformation are reported here. The data selection is crucial, as it determines the data to be used in the analysis. Following the data quality assessment in data understanding, data cleaning translates these flaws into actions, with the goal to ensure high data quality. Moreover, the given raw data attributes may not be appropriate to use for the modelling stage. For that reason, data construction of new variables may be necessary. The initially collected data could be stored in multiple source tables and might therefore require data integration using a data model. Normally, data mining tasks require data to be stored in a single table. Finally, data formatting may be necessary. Those are minor modifications to the data without changing its meaning.

The **Modelling** stage aims to select and apply the modelling techniques (Shearer, 2000). Each data mining task has numerous different modelling techniques, each having their pros and cons. Selecting modelling techniques therefore needs to be done carefully and the choice for selected methods and algorithms are argued here. One must generate a test design, wherein the models can be evaluated fairly and without biases. Typically, this is achieved by splitting the data set into training and testing data. Arguments should be given why the selected test design is appropriate and unbiased. Afterwards, the modelling parameters are listed and the rationale for these parameters values is explained. Subsequently, actual model is built.

Evaluation focuses on a thorough evaluation of the model with respect to the business objectives (Shearer, 2000). The performance of the models is compared using performance metrics, while keeping in mind the data mining goals. Evaluate results means that the model is evaluated with an extensive business perspective, trying to identify if there are business reasons why the model might be deficient. The data mining user may want to test the model on a real-world application and evaluate its findings. The step review process focuses on quality assurance, by thoroughly reviewing the complete data mining process to ensure that no important factors are overlooked. Based on the conclusions drawn in the evaluation, one must determine the next steps. If the results are positive, the project likely proceeds to the deployment phase. But, if the results are disappointing, the project leader may decide to terminate the data mining project. Additionally, the parameters of the classifier may be revised and tuned.

The model itself would have little value if it were not deployed. In **Deployment**, the model is deployed such that the client can use it in the decision-making process (Shearer, 2000). Usually, a real-time business solution is deployed to support this process. A deployment plan is made, wherein it is clarified who is responsible for what part of the deployment and when is it scheduled to be executed. One should plan monitoring and maintenance of the data mining solution, to monitor if the data mining results are interpreted and used correctly by the users, who may not all be very familiar with the interpretation of the results. Ultimately, a final report about the project is produced, wherein the project, with its deliverables and results, are documented. The results are often presented to the customer. Finally, a project review is conducted to learn from successes and failures.

2.3. Related Literature

This subchapter focuses on applications of data mining with similar purposes as this thesis, i.e., tax compliance. The used keywords to find the literature are “tax compliance data mining” and “tax compliance machine learning”. The snowballing technique, a procedure wherein cited papers are explored from a tentative set of papers, as presented by Wohlin (2014) are used to find related literature.

As an introduction, research on the potential usefulness of ML in tax compliance is reviewed to obtain a general understanding of the potential of ML within this field. Frey & Osborne (2017) introduced an approach to compute probabilities of susceptibility to computerisation of jobs, among others through ML. Using variables such as perception and manipulation, creative intelligence, and social intelligence, the probabilities of 702 professions are estimated. The professions “accountants and auditors”, “tax preparers”, and “bookkeeping, accounting, and auditing clerks” fall within the top 20% of jobs most susceptible to computerisation. Furthermore, Kaladari & Al-Sartawi (2023) provide a comprehensive analysis of VAT and the role that artificial intelligence (AI) has played, is playing, and will play in its implementation. They state that the techniques organisations are using to calculate VAT liabilities are not limited to rule-based solutions, as AI is already in use for such purposes. However, the tax procedure is not yet end-to-end automated, but in the future that seems feasible. The conclusions drawn by the two reviewed papers show that ML and AI techniques are promising in the field of tax compliance.

The second set of reviewed papers focuses on examples of ML applications for tax compliance. Until today, there is a limited amount of research on this specific topic: Only two papers use ML to predict VAT codes. The paper most similar to this thesis, in terms of objectives and data resources, is *Utilizing Machine Learning Techniques to Reveal VAT Compliance Violations in Accounting Data* by Lahann, Scheid & Fettke (2019), who also acknowledge the absence of papers utilizing ML methods to increase VAT regulation compliance. Most papers in the field of tax regulation compliance focus on the viewpoint of the tax authority and aim to identify fraud, i.e., significant underreporting. An example is the work by González & Velasquez (2013). Lahann, Scheid & Fettke apply ML techniques on real-world SAP data of a chemical and consumer goods company to predict tax codes. Prior to their research, the company used a set of ERP-implemented rules which determine the tax code. Their paper was inspired by the work of Gehrke & Thams (2010), who manually constructed a decision tree for VAT compliance to determine the tax conditions. The main argument for the use of ML techniques by Lahann, Scheid & Fettke originates from the complexity of the rules, which need to be configured and maintained manually by VAT domain experts. The classifier they built is used in a similar framework: The class predicted by the classifier is compared with the class given by the rule-based system. Mismatches are consequently inspected by domain experts. Thus, the role of the classifier is the same: To identify conspicuous transactions. The used input variables for the classifier are presented on a conceptual level in *Table 2.2*, but their technical link with SAP is not shown. The achieved results are encouraging, with a maximum accuracy of 98.7%, achieved using a C4.5 Decision Tree.

Table 2.2: Input Variables Used for the Classifier by Lahann, Scheid & Fettke (2019)

Attribute	Description	Data Type
Tax country code	The country in which the tax is due.	String
Description	Short description of the tax duties.	String
Reverse charge flag	Refers to a sales tax regulation under which the recipient, not the service provider, owes the sales tax.	Boolean
Billing type	Control for the processing of invoices, credit memos, debit memos, and cancellation documents.	String
In-out-flag	Defines whether it is an input or output transactions.	Boolean
Plant country	Specifies the country in which the goods have been manufactured.	String
Customer country	Defines the country of the customer.	String
Material code	Unique material code that assesses the materials involved from a tax point of view.	String
Trading region flag	Characteristics of the trading region.	String
Incoterms	Series of voluntary clauses for the interpretation of standard contractual terms in international trade in goods.	String

The second paper wherein ML techniques are utilised to predict VAT codes is *Automatic Electronic Invoice Classification Using Machine Learning Models* by Bardelli et al. (2020), who uses electronic XML invoices of an Italian professional accounting software developer to predict the account code and the VAT code, so the data resource differs significantly. Unfortunately, the used input variables for the classifier are not mentioned. They have achieved good results with high precision, namely 99.3% for sent invoices and 95.4% for received invoices using a multilayer perceptron algorithm.

3. Business Understanding

At this point, the thesis structure follows the CRISP-DM model. The first stage of CRISP-DM is *Business Understanding*. The aim of this chapter is to elaborate the domain specific background knowledge presented in *Value-Added Tax* into business and data mining objectives. Furthermore, *Assess Situation* assesses the situation wherein these objected are to be achieved.

3.1. Determine Business Objectives

From a business perspective, the objective of the thesis is to develop a VAT compliance system using ML techniques. This system should identify and prevent incorrect VAT reporting. The solution consists of two elements:

- A classification model to predict tax codes
- The deployment of the classification model to prevent incorrect VAT reporting

These two elements of the artifact relate to the explicitly stated goals in the title of the thesis: The identification and prevention of incorrect VAT reporting. Namely, the classification model is ought to do the identification, while the prevention is ought to be achieved by the deployed solution of this classification model.

The objective of the final business solution is to provide a system that detects potential incorrect VAT statements using ML methods and highlights these statements, such that incorrect VAT reporting can be prevented by manual inspections done by accountants. So, the VAT compliance system should serve a supportive function. Note that the automatization of VAT reporting using ML methods is completely out of scope of this thesis.

3.1.1. Classification Model

To prevent incorrect VAT reporting, potentially incorrect VAT reporting must be identified first. As explained in *State of the Art*, classification is “the process of finding a model that describes and distinguishes important data classes” (Han, Pei, & Tong, 2022, p. 7). These models can then predict categorical class labels. Within the context of this thesis, the class label to be predicted by the classification model is the tax code, i.e., the combination of tax conditions. Note that it is not the objective to predict individual tax conditions (e.g., tax type, tax event, tax rate, etc.), but to predict the combinations of all tax conditions at once, which are captured in the tax codes.

The requirements of the classification model are as following:

- Ability to predict tax codes of transactions
- As accurate predictions as possible
- Reasonable computation time
- Ability to learn from user-generated feedback
- Ability to learn from an ongoing stream of data
- Generalizable to other datasets from other companies from other countries
- Should function for all types of transactions
- Should function for both incoming and outgoing invoices
- Should function for all entities involved in chain and triangular transactions

To BDO, the interpretability of the model is less relevant. Accuracy of the model is far more important than the interpretability, because the primary objective is to limit wrong VAT reporting, regardless of whether the technique used is interpretable.

3.1.2. Deployment of the Classification Model

For the classification model to be valuable, it must be deployed in a system wherein the functionality of the classification model is used in an efficient manner to ultimately prevent incorrect VAT reporting. After the classification model has enabled the identification of potentially incorrectly reported VAT statements, the system should provide an interactive platform to manage these potentially incorrect VAT statements. A potentially incorrect VAT statement is defined as a VAT statement where the given tax code does not match the tax code predicted by the classification model. Further explanation about this definition will be given in *Data Understanding*.

The requirements of the interactive platform are as following:

- Relevant invoice data should be displayed
- Invoices with a potentially incorrect tax code should be highlighted
- Functionality to insert user input, i.e., the confirmation of the correct tax code
- Functionality to filter, such that the user can view specific invoices
- Data from an ongoing stream should be displayed, such that recent data is always visible
- User-friendly and intuitive
- To be developed in Celonis

In short, the main objective of the system is to display potentially incorrect VAT statements and highlight those, such that these invoices can be reviewed by the users, who are the domain experts, i.e., accountants. Users can then confirm the correct tax code in the platform. Thus, risky invoices will be double-checked. After a second opinion by the VAT domain expert, it can be assumed confidently that the final class is correct.

Celonis is process mining software and offers multiple data visualisation functionalities. The objective of process mining is to enhance operational processes by systematically using event data (van der Aalst & Carmona, 2022, p. 3). It is a combination of data science and process science and can be categorised into three categories: Process discovery, conformance checking, and process enhancement. Eventually, BDO aims to use the results of this thesis to mine the business process. In this context, that means comparing the processes of correctly classified VAT statement with the processes of incorrectly classified VAT statements, with as objective to discover patterns and learn from them. The results of this thesis should enable to do process mining for this purpose. However, the process mining itself is out of scope for the thesis.

The two elements of the business solution, as described in *Determine Business Objectives*, are clearly connected. For instance, the predictions made by the classification model are displayed in the platform. But, the platform is not only predicated on the classification model. Reversely, the classification model is also built upon the user-generated feedback inserted in the platform, as the classification model learns from the feedback. In conclusion, the two elements of the business objective are interchangeably connected.

3.2. Assess Situation

In this section, the situation and resources available for the completion of the project are described. The fundamental resource for the project is data of good quality, which is necessary throughout the entire project. Access to the data was granted by BDO for a period of six months, so the thesis was time constrained. During this time, the project had to be completed, since data access would expire after these six months. The provided data comes from one of BDO's clients. It regards a database of the well-known ERP system SAP. This database includes dozens of available SAP data tables, which are assumed to have stored the required data attributes. In *Data Understanding*, all details about the data are described.

Support was offered from both BDO's side as well as the university's side. From BDO's side, assistance was particularly provided by a data science domain expert and a VAT domain expert. From the university's side, scientific knowledge in the field of data science was provided by supervisor Assoz.-Prof. Mag. Dr. Christoph Schütz and assistant supervisor Simon Staudinger, MSc.

In terms of systems, access to Celonis was granted by BDO, wherein the SAP data can be displayed. Celonis has a built-in so-called Machine Learning Workbench. Here, the data can be manipulated and the modelling can be done using Python. Celonis also offers dashboarding functionalities where the system can be deployed.

3.3. Determine Data Mining Objectives

From a data mining perspective, the goal is to predict tax codes of invoices as accurately as possible. Note that the correctness of the class labels in the used data set cannot be guaranteed, despite efforts to clean the used data set. However, to the best of our knowledge, the class labels in the data set are mostly correct. The data mining objective is to identify the incorrect tax codes. The architecture to achieve this is illustrated in *Chapter 4.4*. A classification model will be used to make predictions of the tax codes using invoice data. All different tax codes have an equal importance in classifying correctly.

The classification model is successful if it effectively identifies VAT reports which would otherwise have been reported incorrectly.

4. Data Understanding

This chapter serves to provide a comprehensive description of the relevant data. It starts with an *Introduction to SAP Data* and a description of the initial data collection from SAP, followed by a description of the properties of the data. The data description is done separately for the *Order-to-Cash* process and the *Purchase-to-Pay* process, two SAP processes that are explained later. Throughout the chapter, some data quality issues are addressed. Finally, a subchapter is dedicated to understand the target variable, the *Tax Codes*.

4.1. Introduction to SAP Data

The provided data is a collection of data tables originating from an SAP database. SAP provides state-of-the-art ERP systems and is a leading company in this industry. Their ERP system offers a standardised collection of data tables, implementable across companies from different industries. The data is real-world data, originating from an Austrian production company. Due to its large size and international operations, a wide variety of different kind of transactions is available. Unfortunately, this database is from only one of BDO's customers. SAP uses technical names for data tables and data attributes. From these technical names, the meaning of the data can often not be derived, so the SAP Datasheet (2023) is consulted throughout this thesis to understand the data tables and data attributes. The SAP Datasheet is an online SAP object repository with a comprehensive documentation of SAP.

The access to the SAP data is granted through Celonis, which offers functionalities similar to data extraction and data loading, known from extract, transform, and loading (ETL) processes. ETL processes are defined as

> *"processes that extract data from internal and external sources of an organisation, transform these data, and load them into a data warehouse" (Vaisman & Zimányi, 2022)*

The extraction and transformation in Celonis are similar to ETL processes at a higher level, but the purpose of Celonis is not to load them into a data warehouse. The extraction of SAP data in Celonis is done using a standardised SAP data connection, which Celonis has built in for the most used ERP systems. The data extraction is repeated regularly, meaning that the most recent data is available in Celonis. Because the extracted data are raw data, Celonis additionally offers data transformation functionalities. The raw data can be transformed using Structured Query Language (SQL). Using SQL, new data tables can be created from the extracted raw data tables, which is the transformation process.

The data, extracted from SAP, consists of many raw data tables. Prior to understanding the data, some basic SAP knowledge is required. In SAP, there are multiple business processes, for example Recruit-

to-Hire, which is a solution that supports the end-to-end process from attracting, hiring, training, assigning, paying, and retaining employees. As a comprehensive ERP system, SAP offers a large variety of business processes, ready for usage by the SAP client. Besides processes, SAP makes use of modules. A module is a specific functional component, where each module addresses a specific business function. For example, the Sales and Distribution module supports sales and order processing. Typically in SAP, one process consists of multiple modules, and a module consists of multiple data tables. Data tables may be part of more than one module or process. SAP business processes come with the corresponding data tables, wherein the data is stored (SAP, 2023).

The SAP processes of interest to the purpose of this thesis are the Order-to-Cash (informally called O2C) process and Purchase-to-Pay (informally called P2P) process. As can be derived from the name, the Order-to-Cash process provides the necessary tools to manage the entire process from a customer order to receiving cash from the customer. The Purchase-to-Pay process supports the end-to-end process from purchasing goods to paying the vendor. Each of these two processes have their distinct data tables, although there is some overlap of tables which are used in both processes. More about that later. The Order-to-Cash process stores the data for the outgoing invoices (from the SAP client to the customer) and the Purchase-to-Pay process stores the data for incoming invoices (from the vendor to the SAP client).

SAP tables are a collection of standardised and customised tables, i.e., some tables can only be used by SAP clients in a standardised way, and configuration possibilities are limited. On the contrary, other tables require full customisation by the SAP client. Unfortunately, the documentation of degree of customisation of the SAP tables is missing. Furthermore, no contact with BDO's client was possible to inquire about how the SAP data is customised. In conclusion, the extent to which tables are customised remains unknown, however, many essential tables are rather standardised, so it is not a major issue.

4.2. Order-to-Cash

After the identification of tax code determinants in *Value-Added Tax* on a conceptual level, the link between the technical level of these determinants needs to be made. I.e., the SAP data attributes where these determinants are stored should be found. The relevant data tables are all part of the O2C and P2P process. More specifically, the billing documents, good delivery documents, accounting documents, customer master data, vendor master data, and SAP client master data should be the sources for the tax code determinants. Since the O2C and P2P process have mostly distinct data tables, the data is described separately. The available SAP data tables are viewed through the Celonis Data Explorer, which enables data exploration on a summarised level.

To start, an exploratory approach is used to identify all potentially useful data attributes, regardless of data quality, because this will be assessed later this chapter. To do so, the tax code determinants given in *Value-Added Tax* by the VAT domain expert are considered, enriched by the used input variables by Lahann, Scheid & Fettke (2019), as illustrated in *Table 2.2*. Additionally, any other potentially valuable attributes are listed. At this stage, the goal is not to specifically link the attributes to the tax code determinants, but to obtain a first, exploratory impression of the available data.

In *Table 4.1*, the relevant data tables of the O2C process are listed. The SAP technical table names are given with its descriptions. *Table 4.2* provides the initial set of relevant data attributes for the O2C process. Furthermore, the data types and example values are provided. Many data attributes can be found in multiple data tables. Notice that certain tables are closely related, for example a document header table and a document item/segment table. The header tables store data that belongs to for the entire document, while the item tables store item-specific data. For example, an invoice (VBRK) can have multiple invoice lines (VBRP). In SAP, the company code (BUKRS) represents the SAP user itself. Extensive data about the company code can be found in the company master data table (T001). Likewise, comprehensive customer data is stored in the customer master data table (KNA1). These dimension tables store relevant information such as the country code (ISO code) wherein the entity is located, and its VAT registration number, which is a mandatory identifier for companies and organisations for VAT purposes in the EU (EU VAT Directive, 2023). The VAT registration number is an additional indicator for the country, because VAT registration numbers typically have a 2-character prefix, which is in most cases the same as the country's ISO code, especially for EU countries. The only exception is Greece, where VAT registration numbers start with 'EL' (European Commission, 2023).

Table 4.1: Data Tables – Order-to-Cash

Table	Table Description
BSEG	Accounting Document Segment
BKPF	Accounting Document Header
KNA1	Customer Master Data
LIKP	Sales & Distribution Document: Delivery Header Data
LIPS	Sales & Distribution Document: Delivery Item Data
MAKT	Material Descriptions
T001	Company Master Data
T001W	Plants
T007S	Tax Codes Descriptions
VBFA	Sales Document Flow
VBRK	Billing Document: Header Data
VBRP	Billing Document: Item Data

Table 4.2: Data Attributes – Order-to-Cash

Attribute	Attribute Description	Tables	Type	Example
ALAND	Departure Country	VBRP	char(2)	AT
AUGDT	Acc. Doc. Clearing Date	BSEG	date	31-01-2023
AWTYP	Reference Procedure	BKPF	char(5)	VBRK
BSCHL	Posting Key	BSEG	char(2)	11
BUKRS	Company Code	BSEG, VBRK	char(3)	001
EGBLD	Destination Country Goods	BSEG	char(2)	DE
EGLLD	Supplying Country Goods	BSEG	char(2)	AT
ERDAT	Invoice Creation Date	VBRK	date	31-01-2023
INCO1	Incoterms (part 1) – Incoterm	LIKP, VBRK	char(3)	DAP
INCO2	Incoterms (part 2) – Location	LIKP, VBRK	char(28)	Plant Linz
KUNAG	Sold-to-Party	LIKP, VBRK	char(10)	0123456789
KUNNR	Ship-to-Party	BSEG, LIKP	char(10)	0123456789
LAND1	Country / Destination Country	KNA1, T001, T001W, VBRK	char(2)	DE
LANDTX	Tax Reporting Country	VBRK	char(2)	AT
MAKTX	Material Description	MAKT	char(40)	Screwdriver
MATNR	Material Number	BSEG, LIPS, VBFA, VBRP	char(18)	A99999999
MWSKZ	Tax Code	BSEG, VBRP	char(2)	A0
PSTLZ	Postal Code	KNA1	char(10)	4040
ROUTE	Route	LIKP	char(6)	CZDE01
STCEG	VAT Registration Number	KNA1, T001	char(20)	DK99999999
TEXT1	Name for Value-Added Tax	T007S	char(50)	Export
WERKS	Plant	BSEG, LIKP, LIPS, VBRP	char(4)	1001
XEGDR	Triangular Transaction	BSEG	bool	1

4.3. Purchase-to-Pay

To identify the potentially important data attributes, the same exploratory approach is used as for the O2C process. In this approach, the tax code determinants as explained by the VAT domain expert in *Value-Added Tax* are reviewed. The used input variables by Lahann, Scheid & Fettke (2019), as illustrated in *Table 2.2* are considered as well. Finally, any other attributes of interest are included.

The relevant data tables for the P2P process, with its descriptions, can be found in *Table 4.3*. In *Table 4.4,* the relevant data attributes are listed and described. Again, multiple data sources may exist for

data attributes. Note that there are many similar looking attributes, such as LIFNR, LIFRE, and LLIEF. At a later stage, it needs to be decided which of the attributes is the most useful.

Although the attributes from the O2C process and the P2P process differ, some similarities between them can be recognised. In O2C, outgoing invoices are stored in VBRK and VBRP, while in P2P incoming invoices are stored in RBKP and RSEG. EKKO and EKPO can be considered the equivalents of LIKP and LIPS. Nevertheless, the similarly looking tables have different data attributes. As potential input variables for the classification model, the O2C's data attributes look more promising than P2P's. Some potentially useful data attributes in P2P contain only missing values, making them useless. On the other hand, there are some data tables which are used in both the O2C process as well as the P2P process, such as the accounting documents (BKPF and BSEG) and some of the dimension tables (MAKT, T001, T001W, and T007S). However, distinct segments of an accounting document are used for the O2C and the P2P process. Some segments of an accounting document refer to customer invoices, and some other segments refer to vendor invoices. The segments are separated by having distinct posting keys (BSCHL). The posting keys '01', '02', '11', and '12' indicate customer invoices and thus reveal which segments of BSEG belong to the O2C process. Vendor invoices are indicated by the posting keys '21', '22', '31', and '32', so those segments of BSEG belong to the P2P process. Like for the BSEG table, the BKPF table also has specific rows that are associated with the O2C and P2P process separately. For the BKPF table, this is indicated by the reference procedure (AWTYP). When the reference procedure is VBRK, the accounting document is associated with the O2C process and when the reference procedure is RMRP, the accounting document is associated with the P2P process.

Table 4.3: Data Tables – Purchase-to-Pay

Table	Table Description
BSEG	Accounting Document Segment
BKPF	Accounting Document Header
EKKO	Purchasing Document Header
EKPO	Purchasing Document Item
LFA1	Vendor Master Data
MAKT	Material Descriptions
MKPF	Material Document: Header
MSEG	Material Document: Segment
RBKP	Document Header: Invoice Receipt
RSEG	Document Item: Invoice Receipt
T001	Company Master Data
T001W	Plants
T007S	Tax Codes Descriptions

Table 4.4: Data Attributes – Purchase-to-Pay

Attribute	Description	Tables	Type	Example
AUGDT	Acc. Doc. Clearing Date	BSEG	date	31-01-2023
AWTYP	Reference Procedure	BKPF	char(5)	RMRP
BLDAT	Document Date	RBKP	date	31-01-2023
BSCHL	Posting Key	BSEG	char(2)	11
BUKRS	Company Code	BSEG	char(3)	001
EGBLD	Destination Country Goods	BSEG	char(2)	DE
EGLLD	Supplying Country Goods	BSEG	char(2)	AT
INCO1	Incoterms (p. 1) – Incoterm	EKKO	char(3)	DAP
INCO2	Incoterms (p. 2) – Location	EKKO	char(28)	Plant Linz
LIFNR	Vendor's Account Number	BSEG, EKKO, MSEG, RBKP, RSEG	char(10)	0123456789
LIFRE	Different Invoicing Party	EKKO	char(10)	0123456789
LLIEF	Supplying vendor	EKKO, MSEG	char(10)	0123456789
LAND1	Country	LFA1, T001, T001W	char(2)	DE
MAKTX	Material Description	MAKT	char(40)	Screwdriver
MATNR	Material Number	BSEG, EKPO, MSEG, RSEG	char(18)	A99999999
MFRNR	Manufacturer Number	EKPO	char(10)	0123456789
MWSKZ	Tax Code	BSEG, EKPO, RSEG	char(2)	A0
PSTLZ	Postal Code	LFA1	char(10)	4040
STCEG	VAT Registration Number	LFA1, T001	char(20)	DK99999999
TEXT1	Name for Value-Added Tax	T007S	char(50)	Input tax 0%
WERKS	Plant	BSEG, RSEG	char(4)	1001
XEGDR	Triangular Transaction	BSEG	bool	1

4.4. Tax Codes

The tax code is the target variable in the classification task, i.e., the variable to be predicted. In the Chapter *Introduction*, it was mentioned that tax codes are used in many ERP systems, including SAP, to store the combination of tax conditions. In SAP, the tax code on an accounting document is eventually used for the VAT reporting. Every unique tax code triggers a different combination of tax conditions to be reported on the VAT statement. Thus, the tax code on an accounting document is crucial, and whether it is correct determines whether the reported VAT is correct.

In *Table 4.2* and *Table 4.4*, tax codes have already been introduced shortly. The technical SAP name for tax codes is MWSKZ. Tax codes can be found on accounting documents (BSEG), invoice documents (VBRP for O2C and RSEG for P2P) and purchasing documents (EKPO). However, since the tax codes on accounting documents (BSEG) eventually determine the tax conditions to be reported on the VAT statements, the tax code from the accounting document is the most reliable source. The assumption that tax codes on accounting documents are the most reliable is supported by BDO's data science domain expert and the VAT domain expert. Besides, the tax codes in the VBRP and EKPO table contain many missing values, so these sources would be significantly less useful anyway.

A tax code is simple a two-character string, so no meaningful information can be derived from the codes itself. To understand the meaning of the codes, the tax codes are described in table T007S (tax code names). This is the primary and only source for the tax code descriptions, because no further explanation was given by BDO's client about the data. The tax codes and the tax code descriptions are customizable and maintained by the SAP user, i.e., the client of BDO. The accurateness of the tax code descriptions is therefore dependent on how carefully the descriptions are inserted. As a consequence of the customizability, every company uses their own set of tax codes and no uniform tax codes exist. The non-uniformity of tax codes among companies entails further challenges in the generalisation of the classification model, which needs to be considered in the modelling.

Using the tax codes from the accounting documents, the data set contains 36 different tax codes. This may seem like a large multi-class classification task, however, it is less complex than the data set used by Lahann, Scheid & Fettke (2019), which has 250 different tax codes. The class imbalance is great: Some tax codes are used very frequently, while others are rare. In *Figure 4.1*, the class imbalance is visualised by showing the number of invoices per tax code. For readability, only the tax codes from the O2C process are visualised. However, the tax codes of the P2P process are similarly imbalanced.

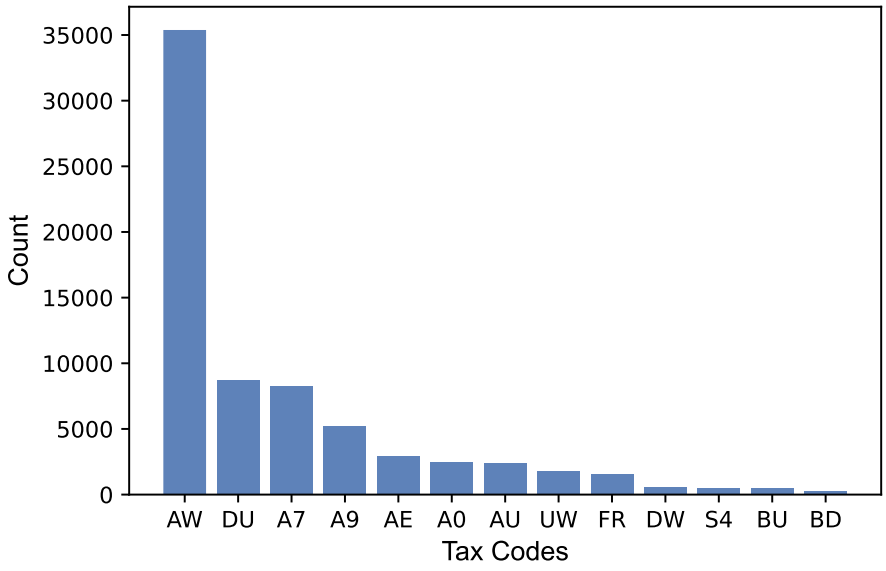


Figure 4.1: Distribution of Tax Codes – Order-to-Cash

The class labels given in the accounting document segment (BSEG) are determined and inserted by the SAP user. In particular, BDO's client has a specialised tax department with accounting professionals who determine the tax code. The VAT domain expert confirms that the tax codes in the data are highly reliable, however, it occurs that tax codes get reported incorrectly, so no certainty about the class label cleanness can be assured. If the correctness of the tax code were to be guaranteed with certainty, there would be no need for this thesis. However, as a requirement for this thesis, the class labels require at least a high data cleanness, in order for the classification model to function properly. Therefore, a crucial assumption is made: The class labels are correct enough to be able to build a reliable classification model. In this context, a reliable classification model does not mean a classification model with high accuracy, but a classification model that identifies incorrect tax codes well. For example, if certain tax codes are structurally incorrectly reported, the classification model would be trained on these incorrect tax codes, and therefore it would probably not identify these invoices as 'potentially wrong'. So, additionally it is assumed that there are no large, structural wrongly used tax codes. The framework for the identification of potentially wrong tax codes is visualised in *Figure 4.2* and is inspired by a similar framework used by Lahann, Scheid & Fettke (2019), although they use a rule-based tax codes instead of manually assigned tax codes by accountants. It is built upon the assumption that the classification model predicts the tax codes correctly, by being trained on a vast majority of correct data. A tax code in the data is potentially incorrect, if the classification model predicts another tax code than given in the data. In that case, an inspection by an accountant is necessary.

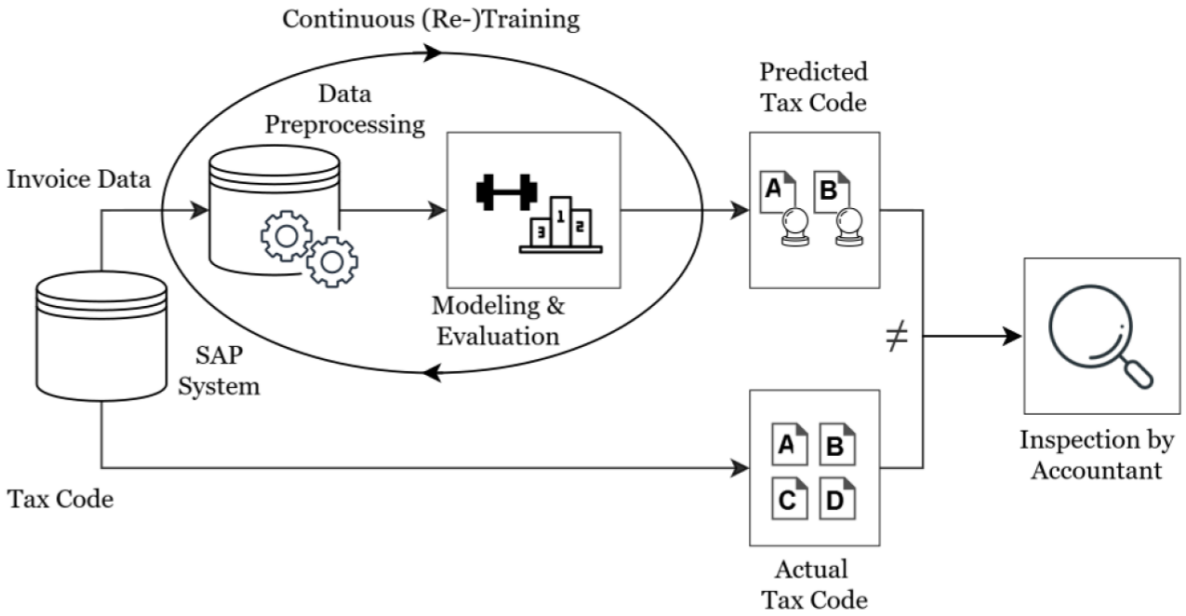


Figure 4.2: System Architecture to Identify Potentially Incorrect Tax Codes

5. Data Preparation

After describing the important data in the previous chapter, this chapter aims to explain all data preparations steps that are required prior to building the classification model. The chapter starts with the *Data Selection*, where the initially selected data is narrowed down to a smaller set of data attributes. This is followed by *Data Construction*, wherein new data attributes are constructed such that the input variables for the classification model are in a more appropriate form. The chapter ends with the *Data Model*, which serves to integrate all necessary data attributes into a single data table. One single data table is namely required to build a model.

5.1. Data Selection

In *Chapter 4.2* and *4.3*, an initial set of data attributes was selected using an exploratory approach. The goal of this subchapter is to reduce this set of attributes by eliminating less valuable attributes. It was found that for many variables multiple different sources exist. Based on reliability and data quality, decisions need to be made about which source to take as leading source.

Order-to-Cash

There may be different data attributes that indicate the country of origin and the country of destination of goods. For example, depending on the interpretation and exact meaning of the variables, ALAND, EGLLD, BUKRS-LAND1, ROUTE, INCO1, INCO2, and WERKS-LAND1 may all describe the origin country of the goods. LAND1, EGBLD, KUNAG-LAND, KUNNR-LAND1, ROUTE, INCO1, and INCO2 could all represent the destination country of the goods. Often, the values for intuitively similar looking variables are different, such as departure country (ALAND) and destination country of the goods (EGBLD). So although many data attributes appear similar, subtle differences exist. The precise difference between similar looking variables cannot always be determined, and unfortunately this information can also not be inquired at owner of the database, which is BDO's client. Therefore, the most reliable data attributes are selected by scanning through the data in consultation with the VAT domain expert to figure out which data attributes are most accurately determining the tax codes. When more than one source exists for a data attribute, the invoice document is always the leading source, because it is the most reliable, according to the VAT and data science domain experts. For all these attributes, the data quality in the invoice documents is always better, having fewer missing values. In *Table 5.1*, the link between tax code determinants and data attributes is made. Sometimes, multiple data attributes can be used. A preference ranking is then mentioned, ranking the most reliable source as highest. Details about how exactly these data attributes can be used are given below the *Table 5.1.*

Table 5.1: Data Attributes as Tax Code Determinants – Order-to-Cash

Tax Code Determinant	Data Table & Data Attribute	Choice
Country of origin of the goods	VBRP.ALAND	1st
	LIKP.ROUTE	2nd
Country of destination of the goods	VBRK.INCO1 & VBRK.INCO2	1st
	VBRK.LAND1	2nd
Type of material (good, service, mix)	VBRP.MATNR & MAKT.MAKTX	1st
Indicator for chain transactions	VBRK.LANDTX & T001W.LAND1	1st
Indicator for triangular transactions	VBRK.ALAND, VBRK.LAND1 & LIKP.ROUTE	1st
Indicator for input/output transaction	All transactions are output in O2C	1st
Responsible for transportation of good	VBRK.INCO1	1st
Indicator for standard/reduced tax rate	VBRP.MATNR & BSEG.MWSKZ	1st

It turns out that the company country attributes are not accurately describing the origin of the goods. Potentially, the company's country (T001.LAND1), the company's VAT registration number (T001.STCEG), the supplying country of goods (BSEG.EGLLD), and the company's plant country (T001W.LAND1) could all have been useful, but probably the company has locations across different countries, and therefore these attributes cannot be used to determine the tax code. On the other hand, the departure country (VBRP.ALAND) and the route (LIKP.ROUTE) determine the tax code quite accurately, however, neither of them is clearly better, although the data of VBRP.ALAND is always available and LIKP.ROUTE has many missing values.

Similar to the company country attributes, there are many customer country attributes which appear to be rather useless, like the customer's country (KNA1.LAND1 related to KUNAG and KUNNR), the customer's VAT registration number (KNA1.STCEG related to KUNAG and KUNNR), and the destination country of goods (BSEG.EGBLD). These attributes are likely useless for the same reason as seen before: The customer may have multiple locations in across different countries, and the customer country only stores one of these countries, which is probably not the country where the goods are actually transported to. The incoterms (VBRK.INCO1 and VBRK.INCO2) turn out to be valuable attributes. Incoterms, or international commercial terms, are a pre-defined set of eleven rules, which define responsibility and liability over goods in business-to-business trade of goods (International Chamber of Commerce, 2023). The incoterm is stored in VBRK.INCO1, and the agreed location where the responsibility over the goods is transferred from seller to buyer is stored in VBRK.INCO2. When the seller is responsible over the entire transportation and delivers the good directly at the customer's warehouse (when VBRK.INCO1 is DAP or DDP), VBRK.INCO2 indicates the good's destination. The destination country (VBRK.LAND1) is also reliable, although less reliable than VBRK.INCO1 and VBRK.INCO2.

The material number (VBRP.MATNR), in combination with its description (MAKT.MAKTX), gives insight into the type of good, i.e., whether the material regards a good, service, or a mixture of a good and service (Werklieferung). By looking at which materials are often used for tax codes that are specifically used for services or mixtures, patterns are found. For example, tax codes for services often have the materials ‘personnel’, ‘service’, or ‘Arbeitsleistung’ (German for ‘labour’). Mixtures of goods and services frequently have the material ‘Montagekosten’ (German for ‘assembly costs’). In conclusion, the material number and its material description can efficiently help to distinguish between goods, services, or mixtures.

Chain and triangular transactions are treated differently compared to other transactions and have their specific tax codes. The conceptual characteristics as illustrated in *Figure 1.1* and *Figure 1.2* help to find possible indicators for chain and triangular transactions. SAP already has a built-in check box on accounting documents to indicate triangular transactions (BSEG.XEGDR). This could perfectly be used as an input variable to indicate triangular transactions, if the goal of the classification model would be to build an as accurate model as possible. But, the primary goal of the classification model is to identify potentially incorrect tax codes. When an accountant uses the check box to mark a transaction as triangular transaction, it can be assumed that the correct tax code is assigned by the accountant, because it was identified that the transaction is triangular in the first place. Regarding triangular transactions, the goal of the classification model is to find triangular transactions when the accountant has failed to see them. Therefore, more subtle indicators for chain transactions are preferred over very obvious indicators, because using very obvious indicators would not help the classification model to identify triangular transactions that are missed by the accountant. A more subtle identifier for chain transactions is within the SAP configurations. According to SAP documentations (2023), the tax departure country (VBRK.LANDTX) generally automatically comes from the plant country (T001W.LAND1). But, in the case of chain transactions, a country other than the plant country needs to be entered as tax departure country. So, whether the tax departure country and the plant country are the same indicates whether the transaction is a chain transaction. After finding a subtle identifier for chain transactions, a subtle identifier for triangular transactions is still required. In the conceptual design of triangular transactions, it was described that three different EU countries are involved in triangular transactions. Using the origin of the goods (VBRP.ALAND), the destination of the goods (VBRK.LAND1), and the route of the goods, the number of countries involved in the transaction can be counted. If the route indicates a different starting of ending point of the goods than VBRP.ALAND and VBRK.LAND1 and all countries are EU countries, it is very likely a triangular transaction.

To distinguish between input and output transaction is extremely simple. Namely, the O2C and P2P process naturally distinguish input and output transactions, as all transactions in the O2C process are output transactions and all transactions in the P2P process are input transactions.

Finally, an identifier for a reduced or standard rate is needed. By using a similar approach as for the type of material, the materials which have a reduced tax rate can be identified. In this data set, no tax codes for reduced tax rates are used. Nonetheless, to make the classification model future-proof and generalizable to other data sets, the possibility of having tax codes with reduced rates should still be taken into account. The set of material numbers (VBRP.MATNR) that have had tax codes with reduced rates in the past form the indicator for reduced tax rates.

Purchase-to-Pay

In *Data Understanding*, it was already found that the P2P process has significantly fewer valuable data attributes available. Yet, this section aims to link the available data attributes to the conceptual tax code determinants. Similarly as in the O2C process, there may be different data attributes that seem to have comparable meanings. For example, depending on the interpretation and exact meaning of the variables, EGLLD, BUKRS-LAND1, INCO1, INCO2, and WERKS-LAND1 could all be indicators for the country of origin of the goods. To find the most useful attribute for every tax code determinant, the data is viewed together with the VAT domain expert, similar to the approach for the O2C process. Also, the invoice document is always the leading source when more than one source exists for certain data attributes, similarly as in the O2C process. In *Table 5.2*, the link between tax code determinants and data attributes is made. A very unfortunate conclusion is that for many tax code determinants, no reliable data attribute is available. Below *Table 5.2*, extensive explanations are given why no valuable data attribute is available for certain tax code determinants.

Table 5.2: Data Attributes as Tax Code Determinants – Purchase-to-Pay

Tax Code Determinant	Data Table & Data Attribute
Country of origin of the goods	No valuable attribute available
Country of destination of the goods	No valuable attribute available
Type of material (good, service, mix)	No valuable attribute available
Indicator for chain transactions	No valuable attribute available
Indicator for triangular transactions	No valuable attribute available
Indicator for input/output transaction	All transactions are input in P2P
Responsible for transportation of good	EKKO.INCO1
Indicator for standard/reduced tax rate	RSEG.MATNR & BSEG.MWSKZ

Although theoretically there are many data attributes that might be a good indicator for the country of origin of the goods, none of them is reliably in line with the tax codes. A small example: When the tax code represents an import (from a non-EU country), the data attributes for origin of goods often contain

an EU country, which could not make sense, according to the VAT domain expert. Or, some of the data attributes contain too many missing values in order to be valuable. The same problem persists for all data attributes that should indicate the destination of the goods. In conclusion, the data quality is insufficient to determine the origin and the destination of the goods.

Moreover, the type of material – good, service, or a mixture – cannot easily be determined. In the O2C process, the material number (VBRP.MATNR) was used in combination with the material description (MAKT.MAKTX). Although there are four different sources for material numbers in the P2P process, none of them have the data quality needed to properly determine the type of material. Most of them have many missing values, with the material number on the invoice (RSEG.MATNR) being the cleanest data attribute, yet it is still not clean enough. Even with the available values, no specific patterns in material numbers can be found for tax codes that regard services or mixtures of goods and services. There are no material descriptions that contain terms such as 'service', 'assembly', its German equivalents, or related terms.

The data attributes used in O2C to identify chain and triangular transactions are not available in P2P, nor are its counterparts. There are also no alternative attributes or SAP configurations that could indicate chain and triangular transactions. As explained, using the check box for triangular transactions (BSEG.XEGD) is undesirable. But, this would not even be feasible, because it only contains missing values in the P2P process. In conclusion, although conceptually it is known what chain and triangular transactions look like, they cannot be identified for the P2P process in this data due to the absence of the necessary data attributes.

The last three tax code determinants in *Table 5.2* are an indicator for an input or output transaction, the responsible entity for the transportation of the goods, and an indicator for a standard or reduced tax rate. For these three tax code determinants, the same logic is used as for the O2C process. Namely, in the P2P process, all transactions are input transactions. The entity responsible for the transportation of the good can again be inferred from the incoterms (EKKO.INCO1), but this data attribute has a substantial number of missing values, so this causes a practical issue. The material number (RSEG.MATNR) in combination with the tax code (BSEG.MWSKZ) can tell about whether the material used is a material that gets a standard or reduced tax rate.

Nevertheless, since too may data attributes are missing or its data quality too poor, the conclusion is that through manual data attribute selection, the required data attributes cannot be found. To ensure that no potentially valuable data attributes are missed, feature selection is additionally done to also statistically search for potentially important features. Feature selection is a method in ML that can find a subset of features to be used for constructing the model, and it has proven to improve models' accuracies (Azhar & Thomas, 2019). There exist multiple statistical measures to compute correlation (or, for categorical variables, dependence). What measure is appropriate depends on the data type of the features x and class label y. The class label y, the tax code, is a categorical attribute. Most of the

features x in the data tables of the P2P process are categorical variables too. Numerical attributes are highly unlikely to be valuable features, so only categorical attributes are screened statistically to look for potentially useful features. A well-known statistical hypothesis test to test the independence between categorical variables is the Pearson chi-square (χ^2) statistic. χ^2 is computed as

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

where o_{ij} is the observed frequency of the joint event (A_i, B_j) and e_{ij} is the expected frequency of (A_i, B_j), which can be computed as

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n},$$

where n is the amount of data tuples, $count(A = a_i)$ is the amount of tuples having value a_i for A, and $count(B = b_j)$ is the amount of tuples having value b_j for B (Han, Pei, & Tong, 2022, pp. 36-38). The higher χ^2, the stronger the dependence between feature x and class label y. χ^2 is computed for every categorical feature x in all tables of the P2P process which has more than one unique value, does not contain only missing values, and has less than 250 unique values. For hundreds of features, χ^2 is computed. The results do not lead to the discovery of potentially valuable features: The highest χ^2 scores are low, and the features x with the highest χ^2 scores are totally irrelevant for the determination of tax codes.

After not only manually screening the data attributes but also statistically, it can be concluded confidently that no reliable classification model for the P2P process could be build with the current set of available data attributes. This is a limitation of the used data set, and with other data sets it should be possible to build a reliable classification model for the P2P process, since conceptually the tax codes are predictable if all data attributes are available and reliable. This means that only the resources to build a classification model for outgoing invoices (O2C) are available, and no classification model can be build for the incoming invoices. After internal discussions at BDO, this conclusion was agreed upon, and it was collectively decided to proceed with only the O2C part of the project and accept that it is infeasible to proceed with the P2P process.

5.2. Data Construction

In *Data Construction*, new data attributes are constructed, which is done by deriving information from existing raw data attributes and transforming them into new data attributes. This is necessary because raw data attributes may not be appropriate to use as input variables in the classification model. For example, some raw data attributes could only be valuable in combination with each other, so a new

data attribute must be constructed from these raw data attributes. Or, raw data attributes may be more specific than necessary, e.g., when the region (EU or non-EU) contains sufficient information, the specific country codes should not be used. This is taken into account during the data construction process. The data attributes from *Table 5.1*, enriched by the tax code determinants from *Value-Added Tax*, serve as a guide towards the final set of data attributes to be used as input variables in the classification model. Below, the input variables are given and its construction is explained.

Tax_Country: Conveniently, VBRK.LANDTX stores the country where the tax is to be reported, which is one of the tax conditions that a tax code consists of. In SAP, this is normally the same country as the supplying plant's country (T001W.LAND1 related to VBRP.WERKS). But, the data attribute can be overwritten manually when the transaction is a chain transaction. The only data transformation necessary is the one-hot encoding, because VBRK.LANDTX is a categorical data attribute and this is often not allowed in ML algorithms. In one-hot encoding, or one-of-*n* representation, every unique value of the categorical data attribute gets one column which is 1 if the attribute is equal to that certain value, and 0 otherwise (Pyle, 1999).

MATNR_Service: In *Data Selection*, it was found that the material number (VBRP.MATNR), in combination with its description (MAKT.MAKTX), indicates the type of good (good, service, or a mixture of both). The materials with the following material descriptions imply services: 'Personnel', ' Service', 'IC Licence', 'Arbeitsleistung' (German for 'work performance), 'IC General Other', and 'IC Segment Other'. The binary data attribute MATNR_Service is 1 if the material description of the involved material is in the set of service indicators, and 0 otherwise. Note that for the set of service indicators presumably differs in other data sets, so the set would need to be revised if the classification is to be used on other data sets.

MATNR_Mixture: Like for services, the material number (VBRP.MATNR) and the material description (MAKT.MAKTX) are used to indicate when the type of good is a mixture between a good and a service, i.e., a good acquisition which is simultaneously assembled. The materials with the description 'Montagekosten' imply such mixtures of a good and service. Similar to MATNR_Service, MATNR_Mixture is a binary data attribute which is 1 if the type of material is a mixture, and 0 otherwise. The indicators for mixtures also would need to be revised when using different data sets in the future. Note that if MATNR_Service and MATNR_Mixture are both 0, the material either regards a good, or the material number is missing.

MATNR_Reduced_Rate: A binary attribute is necessary for material numbers (VBRP.MATNR) that have a reduced tax rate. This can be indicated by looking which material numbers are used for tax codes that regard a reduced tax rate. MATNR_Reduced_Rate is 1 for material numbers that have a tax code for a reduced tax rate, and 0 otherwise. However, in this data set, there are no tax codes for reduced tax rates, so all values for this binary attribute are 0. The attribute is therefore not used for the current data set, but would be necessary for data sets in the future that have reduced tax rates.

Trading_Region: The attributes for the classification model should be generalised as far as possible to not make the classifier learn unnecessarily complex patterns that do not influence the target variable. Hence, the country of origin and country of destination of the good are generalised into trading regions. From an EU perspective, there are three trading regions: Domestic, intra-community, and export. By using these three trading regions instead of the origin and destination country specifically, the information is simplified while no valuable information is lost. On the basis of the categorical data attribute Trading_Region are the departure country (VBRP.ALAND) and the destination country (VBRK.LAND1 and a combination of VBRK.INCO1 and VBRK.INCO2). In *Table 5.1*, it was mentioned that the incoterms (VBRK.INCO1 and VBRK.INCO2) are more reliable indicators for the destination country than VBRK.LAND1. Namely, when VBRK.INCO1 is an incoterm indicating a direct delivery towards the customer (DAP and DDP) (International Chamber of Commerce, 2023), VBRK.INCO2 indicates the city where the goods are delivered. So, when VBRK.INCO1 is DAP or DDP, VBRK.INCO2 is used for the destination of the goods, and otherwise VBRK.LAND1 is used. However, since the destinations in VBRK.INCO2 are city names rather than countries, the corresponding countries need to be derived from the cities. The Python library *geocoder* is used to derive the country ISO code from city names. Geocoder makes use of Nominatim, an open-source geocoding service (Carriere, 2013). However, before the country ISO codes can be derived, VBRK.INCO2 is cleaned first, as it often contains words such as ‘warehouse’, ‘delivery address’, or other geographically irrelevant words which make the country search more difficult. Such words that do not reveal any country information are parsed and removed from the string, such that only the city name remains. Then, geocoder finds the corresponding country ISO code for the city. Furthermore, for cities in the United Kingdom, geocoder checks if these are located in Northern Ireland, because Northern Ireland is considered intra-community for VAT purposes. For VBRK.LAND1, the postal code of the customer (KNA1.PSTLZ linked to VBRK.KUNAG) is used to distinguish between trades to Northern Ireland and the other territories of the United Kingdom. Zip codes in Northern Ireland start with ‘BT’. Note that the assumption is made that if the destination country (VBRK.LAND1) and the customer country (KNA1.LAND1 linked to VBRK.KUNAG) are both United Kingdom, they regard the precise same location within the United Kingdom. Besides using VBRK.INCO1 when it regards a direct delivery to the customer (DAP and DDP), VBRK.INCO1 is also useful to identify exports. There are some incoterms which specifically are for shipments, namely FAS, FOB, CFR, and CIF (International Chamber of Commerce, 2023). When those incoterms are used, the tax code always refers to an export. So, when the incoterms (VBRK.INCO1) indicate a shipment, the trading region is export. Note that this only certainly is true for this data set, because it could be that shipments are also used for domestic or intra-community trades in other data sets. Furthermore, if the country of origin and the destination country are the same, one can speak of a domestic trade region. If the departure country is not the same as the destination country and both of the countries are EU countries, the trade is an intra-community trade. Finally, if the departure country is an EU country and the destination country a non-EU country, the trade is an export. Trading_Region is also one-hot encoded, leading to three new columns, as Trading_Region has three unique values.

N_Countries: The number of countries involved in the trade, N_Countries, helps to identify triangular transactions, as they have exactly three countries involved. To determine the number of countries involved, the departure country (VBRP.ALAND), the destination country (VBRK.LAND1 and a combination of VBRK.INCO1 and VBRK.INCO2), and the route (LIKP.ROUTE) are used. Similarly as for the Trading_Region, the country of VBRK.INCO2 is used as destination country if VBRK.INCO1 is DAP or DDP. Otherwise, VBRK.LAND1 represents the destination country again. The route (LIKP.ROUTE) requires data construction, because LIKP.ROUTE is a 6-character string containing both the start and end point, e.g., ATNL01 for a route from Austria to the Netherlands. Two new data attributes are created: The start of the route, which are the first 2 characters of LIKP.ROUTE, and the end of the route, which are the middle 2 characters of LIKP.ROUTE. Subsequently, the departure country, destination country, and start point of the route are used to determine the number of countries involved. Unlike the starting point of the route, the ending point of the route seems unreliable and not logically in line with the tax codes, which is why only the starting point of the route is used. Thus, the count of unique countries in these three data attributes constitutes N_Countries, and its values are either 1, 2, or 3.

Route_Start_NonEU: Furthermore, the route (LIKP.ROUTE) can be useful to indicate specific cases where the starting point of the transportation of the good is outside the EU. Although the data set is from an Austrian company, this may be the case if this company has a plant outside of the EU which it uses to ship goods from. Route_Start_NonEU is a binary data attribute, 1 if the starting point of the route is outside of the EU, and 0 otherwise.

Route_Start_EU: The cases where Route_Start_NonEU is 0 does not necessarily mean that the starting point of the route is from the EU. LIKP.ROUTE namely has some missing values, so a 0 for Route_Start_NonEU could also indicate that the starting point is unknown. To account for this, the binary data attribute Route_Start_EU is constructed, which is 1 if the starting point of the route is within the EU, and 0 otherwise. Note that if Route_Start_NonEU and Route_Start_EU are both 0, it means that the value for LIKP.ROUTE is missing.

Chain_Transaction: As an indicator for chain transactions, the SAP configuration as explained in *Data Selection* is used. It was mentioned that the country where the tax is to be reported (VBRK.LANDTX) is normally the supplying plant's country (T001W.LAND1 related to VBRP.WERKS. But, VBRK.LANDTX is overwritten manually when there is an additional entity involved in the transaction, i.e., in a chain transaction. Therefore, Chain_Transaction is a binary data attribute, which is 1 if VBRK.LANDTX is equal to T001W.LAND1 related to VBRP.WERKS, and 0 otherwise.

Responsibility_Buyer: A binary data attribute is required to indicate whether the buyer or the seller is responsible for the transportation of the goods in VAT definitions. VAT definitions define that for the following incoterms (VBRK.INCO1), the buyer is responsible for the transportation of the goods: EXW, FCA, FAS, FOB, CFR, CIF, CPT, and CIP. So, Responsibility_Buyer is 1 if the VBRK.INCO1 is in this set of incoterms, and 0 otherwise.

5.3. Data Model

ML algorithms, which are to be used in *Modelling*, require the data attributes to be stored in one single data table. As seen, the necessary data attributes are stored over many different data tables. Therefore, data integration is required. Data integration is defined as

> *"combining data from different sources and bringing it together to ultimately provide a unified view" (Sherman, 2015, p. 14)*

To integrate the data, a data model is required. A data model is

> *"a specification of the data structures and business rules representing business requirements. A data model provides a method to visually communicate the data that is needed, collected, and used by an organisation." (Sherman, 2015, pp. 173-174)*

This definition of data modelling is not to be confused with the modelling stage of CRISP-DM, as they are two separate things. To understand how the tables are related, data models typically visualize the primary keys (PK) and foreign keys (FK) (Sherman, 2015, p. 176). The PK or FK of a data table can either be a single data attribute or a combination of data attributes. The primary key is the unique identifier for a single data record in a table. The foreign key establishes links between tables by referring to another table's primary key. Furthermore, data models typically indicate the cardinality, which is the amount of instances of an entity linked to another entity (Sherman, 2015, p. 181). For example, one invoice document has multiple invoice lines, so this is a one-to-many relationship (also referred to as 1:n). The relationships between the relevant data tables are known from standard SAP data models (SAP, 2023), and the cardinalities are checked by querying and exploring the data. For the O2C process, this leads to the data model as visualised in *Figure 5.1*. The accounting document segment (BSEG) contains the class label y, MWSKZ, and is related to the accounting document header (BKPF) in a one-to-one relationship, which may seem unintuitive at first, but the fact that only specific accounting document segments are used clarifies this. Remind that in *Chapter 4.3*, it was explained that only posting keys (BSCHL) which indicate customer invoices are used, which is for the posting keys '01', '02', '11', and '12'. The tax codes (MWSKZ) are linked to the tax code descriptions (T007S). The accounting document header (BKPF) is linked with the billing document header (VBRK), which contains several important data attributes. The company and customer stated on the invoice are enriched with company data (T001) and customer data (KNA1). Obviously, the billing document header (VBRK) is linked to the billing document items (VBRP). The lines store the plant and the material number, which are enriched with plant data (T001W) and material descriptions (MAKT). The sales document flow (VBFA) indicates how the documents flow in the O2C process. It does not contain any relevant data attributes, but it is necessary to link the billing document item (VBRP) to the sales and distribution (SD) delivery item document (LIPS). This table also does not contain important data attributes, but the SD delivery header document (LIKP) is needed to reach the route attribute (LIKP.ROUTE).

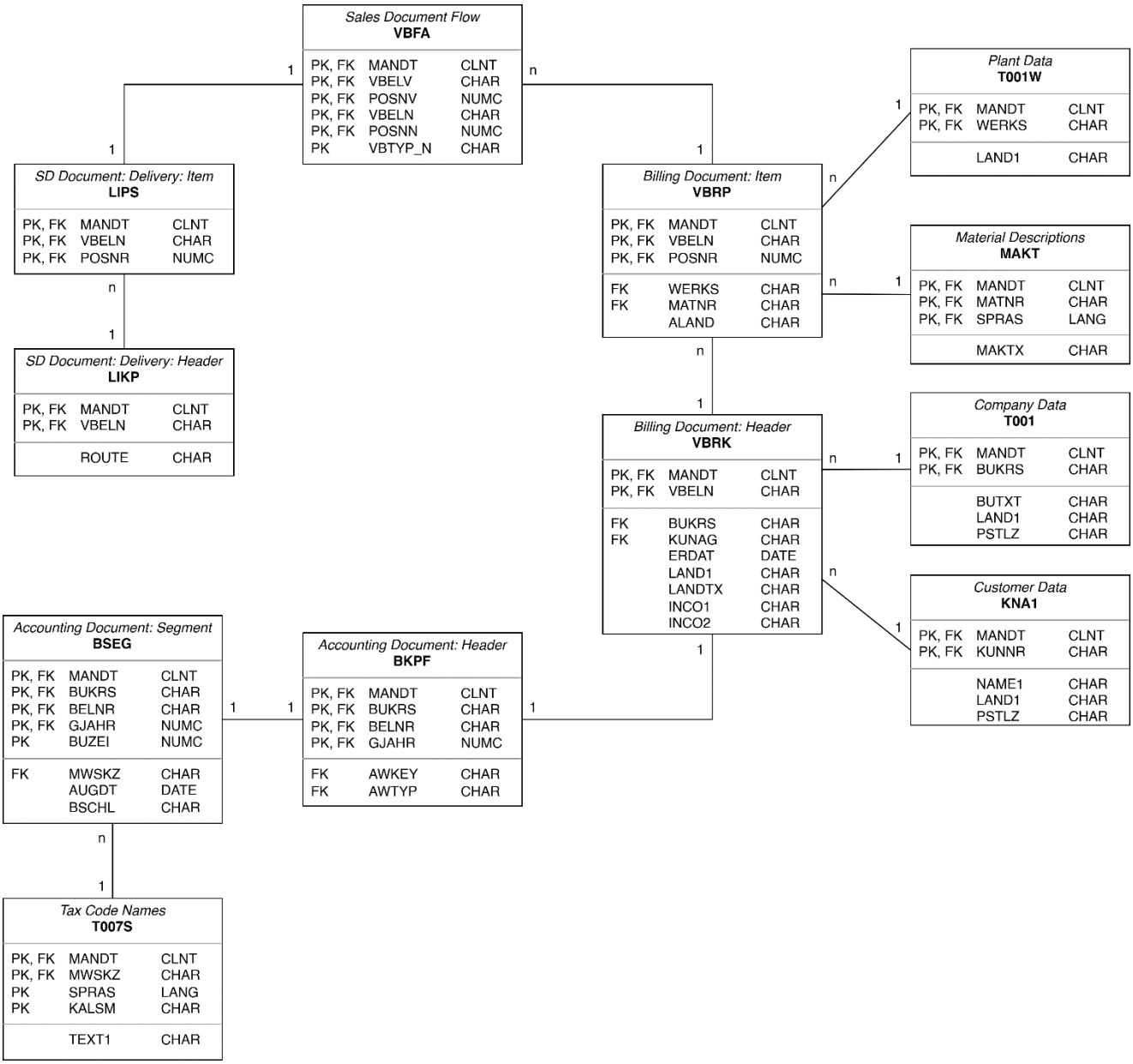


Figure 5.1: Data Model – Order-to-Cash

The data model forms the conceptual foundation for the data integration. Essential to the data integration are the foreign keys, as they establish the relationships between tables. The actual data integration is done in Celonis Data Transformations, which is part of Celonis Data Integration. Herein, SQL queries can be used to transform raw data tables. The raw data tables are connected to Celonis with an SAP connection. Then, SQL is used to create the unified data table, which is named O2C. The SQL query for the creation of this table is shown in *Listing 5.1*.

Listing 5.1: SQL Query to Create Order-to-Cash Table

```
CREATE TABLE O2C AS
  SELECT
    (VBRP.MANDT || VBRP.VBELN || VBRP.POSNR) AS VBRP_KEY, VBRP.MANDT, VBRP.VBELN,
    VBRP.POSNR, VBRK.GJAHR, VBRK.ERDAT, VBRK.BUKRS, T001.BUTXT AS VBRK_BUKRS_BUTXT,
    T001.LAND1 AS VBRK_BUKRS_LAND1, VBRK.KUNAG, KNA1.NAME1 AS VBRK_KUNAG_NAME1,
    KNA1.LAND1 AS VBRK_KUNAG_LAND1, KNA1.PSTLZ AS VBRK_KUNAG_PSTLZ, VBRP.MATNR,
    VBRK.LAND1, VBRK.LANDTX, VBRP.ALAND, VBRK.INCO1, VBRK.INCO2, LIKP.ROUTE,
    LEFT(LIKP.ROUTE, 2) AS ROUTE_FROM, RIGHT(LEFT(LIKP.ROUTE, 4), 2) AS ROUTE_TO,
    BSEG.AUGDT, BSEG.MWSKZ,
    CASE WHEN BSEG.AUGDT IS NULL THEN NULL ELSE 'X' END AS BSEG_AUGDT_CLEARED,
    CASE WHEN VBRK.LANDTX = T001W.LAND1 THEN 1 ELSE 0 END AS LANDTX_IS_WERKS_LAND1
  FROM <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.BSEG
    INNER JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.BKPF ON
      BSEG.MANDT = BKPF.MANDT AND BSEG.BUKRS = BKPF.BUKRS AND
      BSEG.BELNR = BKPF.BELNR AND BSEG.GJAHR = BKPF.GJAHR
    INNER JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.VBRK ON
      VBRK.MANDT = BKPF.MANDT AND VBRK.VBELN = BKPF.AWKEY
    LEFT JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.T001 ON
      VBRK.MANDT = T001.MANDT AND VBRK.BUKRS = T001.BUKRS
    LEFT JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.KNA1 ON
      VBRK.MANDT = KNA1.MANDT AND VBRK.KUNAG = KNA1.KUNNR
    INNER JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.VBRP ON
      VBRP.MANDT = VBRK.MANDT AND VBRP.VBELN = VBRK.VBELN
    LEFT JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.VBFA
      VBRP.MANDT = VBFA.MANDT AND VBRP.VBELN = VBFA.VBELN AND
      VBRP.POSNR = VBFA.POSNN
    LEFT JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.LIPS ON
      LIPS.MANDT = VBFA.MANDT AND LIPS.VBELN = VBFA.VBELV AND
      LIPS.POSNR = VBFA.POSNV
    LEFT JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.LIKP ON
      LIPS.MANDT = LIKP.MANDT AND LIPS.VBELN = LIKP.VBELN
    LEFT JOIN <%=DATASOURCE:EXTRACTIONS_SAP_CONNECTION_SP1%>.T001W ON
      VBRP.MANDT = T001W.MANDT AND VBRP.WERKS = T001W.WERKS
  WHERE BSEG.BSCHL IN ('01', '02', '11', '12') AND BSEG.MWSKZ <> '**' AND
      BKPF.AWTYP = 'VBRK'
  ORDER BY VBRP.MANDT, VBRP.VBELN, VBRP.POSNR;
```

The SQL query from *Listing 5.1* shows that the primary key of the O2C table is the same primary key as the billing document item, i.e., a combination of MANDT, VBELN, and POSNR. This means that one record in the O2C table is one billing document item (VBRP), and it is enriched with all necessary features x, the class label y, and further data attributes that are used in the deployment. Sometimes, an INNER JOIN is used, because records that do not match would be useful for the classification model, because too many important attributes would be missing. In other cases, a LEFT JOIN is more appropriate, because it is desired to keep the initial set of rows, even if the records do not match with the table to be joined. The loss of records as a result of using INNER JOINs is very minimal.

Since the billing document header (VBRK), and not the item (VBRP), is linked to the accounting document header (BKPF), all the items on a single billing document have the same class label y. In SAP, it is technically impossible to have one billing document header where its items have different tax codes. Nevertheless, it was decided to construct the O2C table on the invoice item level (VBRP) instead of the invoice header level (VBRK), because data attributes which are on the invoice item level (VBRP) would need to be aggregated somehow if the O2C table would be build on the invoice header level (VBRK), and this would lead to a loss of information. This specification needs to be kept in consideration when the classification model makes predictions. The O2C data table contains 288,903 data records, i.e., invoice lines, which correspond to 70,581 invoices. The invoices are created between the beginning of 2021 and mid September 2023.

6. Modelling

After the data has carefully been described, selected, cleaned, and constructed, the data is ready for the classification model to be build. This chapter starts with the generation of a *Test Design*, i.e., an environment wherein the models can be evaluated fairly and without bias. Then, different *Classifiers* and its parameters are selected, and the actuals models are built. At the end of the chapter, the framework for *Continuous Retraining* of the classifier is elaborated.

6.1. Test Design

The test design for the modelling is important, because the models need to be evaluated fairly. If the test design leaves room for bias, the evaluated results are not reliable and wrong conclusions may be drawn. As a test design, cross-validation is often used. Labelled data is partitioned into k disjoint subsets of the same size. While one of the k segments is used to test, the other $k-1$ segments are used to train. This repeats k times by selecting each of the k different segments once in the data as a test set. Then, the average accuracy is calculated over the k different test sets (Aggarwal, 2015, p. 336). This method is called k-fold cross-validation and gives a reliable, unbiased accuracy, because every data record is at some point once used in the testing subset and all other subsets are used for training. The specific classification task of the thesis is a multi-class imbalanced classification task, as was visualised in *Figure 4.1*. When the classes are distributed out of balance, it is crucial that the classes are represented proportionally in the training and testing subset. If the classes are subsampled disproportionally, there is the risk of not having enough data records in either the testing or the training subset of certain rare classes. In this training subset, this could lead to the model not properly learning the patterns for that class. For the testing subset, this means that the accuracy for that class cannot be tested properly. Representing class labels proportionally among subsamples of data is called stratified subsampling. In stratified k-fold cross-validation, every class is thus represented proportionally in the testing and training subset of the k folds. Because this technique is the most suitable, it is used for the test design. A value of $k = 10$ is used, which is common in k-fold cross-validation. For a fair comparison between classifiers, exactly the same splits between training and testing data are used every time for all 10 splits.

Before the data is split using k-fold cross-validation, rare classes are deleted from the data set, because these classes would likely not be learned well as a consequence of not having enough data. As a threshold, a rare data class is defined as a class having less than 50 data records. 4 out of 17 classes do not meet this threshold, so 13 classes remain in the data set. These four classes were present in 45 data records and are now omitted from the data set.

Another property to consider in the test design is that in SAP, a single invoice can only have one tax code, i.e., having multiple tax codes among one invoice's different invoice lines is technically not allowed. Yet, a data record in the O2C table represents an invoice line instead of an invoice, implicating that predictions will be made per invoice line rather than per invoice. This would violate the technical constraint in SAP, because multiple different tax codes could be predicted per invoice. Hence, the predictions on the invoice line level would need to be aggregated to the invoice level, such that the technical constraints are met, i.e., such that all invoice lines per invoice have the same tax code. As an aggregation function, it may seem logical at first sight to use the count of the predicted class labels. For example, per invoice, one could assign the tax code which is predicted the most frequent among its invoice lines. This approach is simple but reasonable: The more often a tax code is predicted, the more likely that tax code belongs to the invoice. However, in certain scenarios this aggregation function is not desirable. Namely, sometimes the invoice of a single invoice line may be leading for the whole invoice. This is the case for tax codes related to services and mixtures of goods and services. For an invoice, if one of its invoice lines has a tax code related to a service or a mixture, all its invoice lines have this tax code. Tax code 'AU' is used for services, and 'BU' for mixtures. So, for these two tax codes, the aggregation function is as follows: If one of the invoice lines contains this tax code, all the invoice lines will get this tax code. For all other tax codes, the first aggregation function will be applied, where the tax code with the highest frequency is leading for the invoice. If the count is equal, the alphabetically first tax code will be assigned to all its invoice lines. *Figure 6.1* depicts how the two applied aggregation functions work in a simple example of two invoices, 000001 and 000002, which both have 4 invoice lines. y_{pred} is the predicted class label and y_{pred_agg} is the predicted class label after the application of the aggregation function. For invoice 000001, the count of A9 is 3 and A0 is 1, so y_{pred_agg} of the invoice is A9. For invoice 000002, one of the invoice lines has tax code AU, so y_{pred_agg} of the invoice is AU, regardless of the count. The two aggregation functions are combined in one algorithm and explained in pseudocode in *Algorithm 6.1*. The setup of the aggregation functions is rather simple to generalize to future data sets, as only the initial list $service\_mixture$ would need to be adjusted with the client-specific tax codes that correspond to services and mixtures.

Invoice	Invoice Line	y_{pred}	y_{pred_agg}
000001	001	A9	A9
000001	002	A9	A9
000001	003	A0	A9
000001	004	A9	A9
000002	001	A7	AU
000002	002	AU	AU
000002	003	A7	AU
000002	004	A7	AU
...	...	...	...

Figure 6.1: Aggregation Functions: Examples

Algorithm 6.1: Aggregation Functions for Class Label Aggregation

```
service_mixture = ['AU', 'BU']
y_pred_count = empty dictionary
FOR invoice IN O2C
    FOR invoice_line IN invoice
        IF invoice_line[y_pred] IN service_mixture THEN
            invoice[y_pred_agg] = y_pred
            BREAK
        ELSE
            IF y_pred IN y_pred_count THEN
                y_pred_count[y_pred] += 1
            ELSE
                y_pred_count[y_pred] = 1
            END IF
        END IF
        invoice[y_pred_agg] = max(y_pred_count)
    END FOR
END FOR
```

6.2. Classifiers

The classifiers that will be trained and tested are Gaussian Naïve Bayes (GNB), k-Nearest Neighbour (kNN), Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT). The evaluation of classifiers by Wu et al. (2008) is used for this selection of the chosen classifiers. Although their work is slightly outdated, the authors provide a review of 10 common algorithms in data mining. Later on, the *Evaluation* of this initial set of algorithms provides strong arguments that this set of algorithms is sufficient. To review, this is the set of features x used to train the classifiers:

- Tax_Country
- MATNR_Service
- MATNR_Mixture
- MATNR_Reduced_Rate (only used in future data sets, not in the current data set)
- Trading_Region
- N_Countries
- Route_Start_NonEU
- Route_Start_EU
- Chain_Transaction
- Responsibility_Buyer

With those features x, the objective is to predict the class label y, the tax code (MWSKZ). The classifiers are trained and tested using scikit-learn, which is a commonly used Python library for ML (scikit-learn, 2023). Below, the parameters for the five chosen ML algorithms are stated and it is argued why these parameters are chosen. Gaussian Naïve Bayes does not require any input parameters, as it simply uses the Bayes probability theorem. First of all, the k-Nearest Neighbour classifier (scikit-learn, 2023) uses the following parameters:

- $k = 9$
- Uniform weights of data points
- Euclidian distance

A value of $k = 9$ is large enough to reduce the potential influence of outliers, while still being computationally fast. Namely, when k is small, few outliers may already influence the prediction. This is often undesirable, as outliers frequently are errors. On the other hand, if k is too large, then the neighbourhood may include too many points from other classes (Wu, et al., 2008). A uniform weighting, where all data points have an equal weight, is the standard and therefore logical starting point. Using Euclidian distance instead of Manhattan distance is considered the default in kNN as well, and is hence the chosen parameter.

For the Random Forest classifier (scikit-learn, 2023), the following parameters are used:

- Number of trees = 100
- Gini impurity function is used to measure the quality of a split
- No maximum depth of a tree
- A minimal of 2 samples is required to split an internal node
- A minimal of 1 number of sample in newly created leaves
- No maximum number of leaves

A relatively large number of trees is used, namely 100, because this seemed to generate more accurate results. Also, using Gini impurity as a function to measure the quality of splits works slightly better than using entropy or log loss. Furthermore, the default values are chosen for the tree- and leaf-specific parameters. These default values are the least restrictive and thus allow the most freedom for the trees and leaves.

For the Support Vector Machine classifier (scikit-learn, 2023), the parameters are as follows:

- Linear kernel
- Do not use probability estimates

The different kernels generate the same results, so the most basic kernel – the linear kernel – is used. Furthermore, all default values are used, because there are no reasons against using them in this classification task.

Finally, in the Decision Tree (scikit-learn, 2023), the following parameters are used:

- Gini impurity function is used to measure the quality of a split
- Choose the best split
- No maximum depth of a tree
- A minimal of 2 samples is required to split an internal node
- A minimal of 1 number of sample in newly created leaves
- No maximum number of leaves
- No maximum number of features to consider when looking for the best split

The different measurement function all generated the same accuracy levels. The Gini measure is the default for DTs, and as it works the best for the RF classifier, so it is also used for the DT. Furthermore, the best split is used instead of a random split, because it is much more likely to produce good results and it is therefore also the default. As in the RF, the tree- and leaf-specific parameters are chosen as the least restrictive to allow maximum freedom in terms of the tree and leaves. Those values are also the default in DTs.

One might wonder why manually constructed IF-THEN rules could not be used in this classification problem. One issue with such rules is that they would need to be constructed manually, as Lahann, Scheid & Fettke (2019) also emphasised. However, our data set contains 17 different tax codes, while the data set of Lahann, Scheid & Fettke contains 250 different tax codes, so the complexity is smaller. The major issue with manually constructed rules is that the tax codes are configurable by the SAP user and thus differ from company to company. This would then require a mapping of tax codes, where every user-specific set of tax codes gets mapped to a general set of tax codes which covers all variations of tax codes among data sets. First of all, this mapping of tax codes would be difficult and time-costly. Secondly, it is infeasible to generate a general set of tax codes that cover all possibilities of tax code variations for data sets that are unknown at this stage. In conclusion, manually constructed IF-THEN rules could work for this data set, but is not an efficient solution considering that the classification model should be generalizable to future data sets.

6.3. Continuous Retraining

One of the requirements of the classification model is that it should be able to learn continuously from an ongoing stream of data. In practice, this can be achieved by continuously constructing new versions of the classification model, while continuously extending the original data set with new data from the stream. At the time the classification model is retrained, the new data would then have been incorporated in the training of this model, enabling an updated classification model.

The naïve solution to make a classification model learn from an ongoing stream of data is to retrain the model on the entire dataset, even though the data set is only slightly larger than in the previous learning cycle. E.g., assume a classification model which uses data collected for a period of 5 years, with the requirement to have learnt from the new data every week. The proportion of the new data – one week compared to 5 years - would be insignificant, assuming a consistent amount of data flow. Yet, the classification algorithm needs to use these 5 years of data when being retrained. Therefore, completely reconstructing a classifier continuously on the entire dataset is an inefficient approach. Nevertheless, if computation time allows, this method is not problematic and can still satisfy the requirement to make the classification model learn continuously from a data stream. The computation time may be long if the data set is huge, which is where this approach becomes problematic.

To deal with a growing number of classes, the binary attribute MATNR_Reduced_Rate was introduced in *Data Construction* to capture tax codes for reduced tax rates if they would be used in the future. The set of chosen input variables should normally be able to capture all tax codes. However, it may be that certain companies have industry-specific tax codes which are used for exceptional cases, and additional input variables may need to be constructed to be able to identify these tax codes. But for the current data set, the set of chosen input variables can explain the current tax codes and is as future-proof as possible in terms of being able to identify patterns in new class labels, because at this point, there are no other input variables which may become relevant in the future besides MATNR_Reduced_Rate. So, the classifier should learn from an ongoing data stream. The idea is that the classifier is continuously retrained on the entire dataset, incorporating the new data. The new stream of data is not limited to new invoices; there is also user-generated feedback that the classifier should learn from, i.e., the final tax code to be confirmed by the accountant in the design artifact that is built in *Deployment*. The classifier should be retrained with the tax codes as confirmed by the accountant instead of using the initial class labels.

The objective of learning from an ongoing stream of data is that the artifact quickly adopts to the most recent data. Since the correctness of the class labels cannot be guaranteed, the quality of the model must improve from the user-generated feedback, assuming that after a second check by the accountant the class labels are correct. Hence, it would be a wise idea to assign a greater influence on more recent data, such that the classification model becomes dynamic and it quickly adjusts to recent data.

Assigning higher weights for more recent data points can be done by oversampling recent data (Sammut & Webb, 2017). Oversampling means to duplicate certain data records, so the number of data records increases. By oversampling only recent data, the recent data will then have greater influence on the classifier. To increase the weight of recent data equally for all class labels y, the oversampling is done in a stratified way. It is decided to assign heavier weights to data records from the last 180 days. From these 180 days, every period of 30 days gets a certain weight, where the most recent data records (from the last 30 days) have the heaviest weight. The weighting only starts after the deployed system is in use for a certain period of time, because only then there is user-generated feedback. The number of different splits which have their distinct weights therefore depends on how long the deployed system has been in use. The maximum number of splits is six, such that at a maximum data from the last 180 days have heavier weights. *Algorithm 6.2* shows how the weighting of recent data records is done conceptually in pseudocode. Herein, α is a parameter for the maximum weight that a split can have. The maximum weight is expressed by the amount of data that is oversampled. α is initialised at 0.9, meaning that 90% of the data records are duplicated, so the most recent data split has a data weight of 1.9. Depending on the time (in days) that the system has been in use, $system\_days$, each $data_{split}$ gets assigned a certain $weight$, starting the iteration with the most recent $data_{split}$. Every time after the $weight$ is assigned to a split, the $weight$ is decreased gradually by $weight\_step \;/\; 2$. So, the more recent the $data_{split}$, the heavier the $weight$ that is assigned.

Algorithm 6.2: Weighting of Recent Data Records

```
α = 0.9
n_splits = system_days // 30
IF n_splits > 6 THEN
    n_splits = 6
END IF
weight = α
weight_step = weight / n_splits
FOR split IN RANGE(n_splits)
    assign weight to data_split
    weight −= weight_step / 2
END FOR
```

The assigned $weight$ to each $data_{split}$ thus decreases gradually over time and depends on how long the system is in use. After the system has been in use for at least 180 days, the weight function for data split n is

$$weight_n = 1 + \left(\alpha - (n-1) \times \left(\frac{\alpha}{|N| \times 2}\right)\right)$$

where α is the weight to be added to the data split $n \in N$ and $N = \{1, 2, 3, 4, 5, 6\}$, where $n = 1$ is the data split with from the most recent 30 days. Since only data records from the most recent 180 days are weighted, the weighted data slowly becomes irrelevant as the data set continues to grow. To overcome this issue, only data records from the recent 2 years are kept in the training data set. The influence of the data weights then remains constant as the data set grows. With the given weight function and the chosen parameter value $\alpha = 0.9$, the influence of the data weights on the data set is approximately 9% if the amount of data streaming in is constant over the 2 years. This means that approximately 9% of the data records in the data set are duplicated rows as a result of oversampling in order to assign the weights. The influence may differ depending on the fluctuation in the amount of the data stream over those 2 years. 9% is not inconsiderably large nor inconsiderably small, such that the balance between having a dynamic model and using older data is reasonable. Omitting data records older than 2 years of course causes information loss. But, since predictions on new data are made using existing knowledge, the knowledge is retained over time. The knowledge therefore develops as new data streams in and retains its knowledge as new knowledge is always built upon existing knowledge. By omitting old data from the training data set, there is still a minor risk that important knowledge may be forgotten, however, it is assumed that data from a period of 2 years retains the knowledge if tax codes keep being used. If a tax code is not used for more than 2 years, all the knowledge about this tax code is forgotten. However, that tax code then seems not relevant anymore, and should it become relevant again, then the tax code can be quickly learned again in the business solution which will be presented under *Deployment*. The data weights of the data are visualised in *Figure 6.2*. The graph on the left shows that all data records have an equal weight at the time the system is deployed, and the graph on the right shows how the data records are weighted after the system is deployed for at least 180 days.

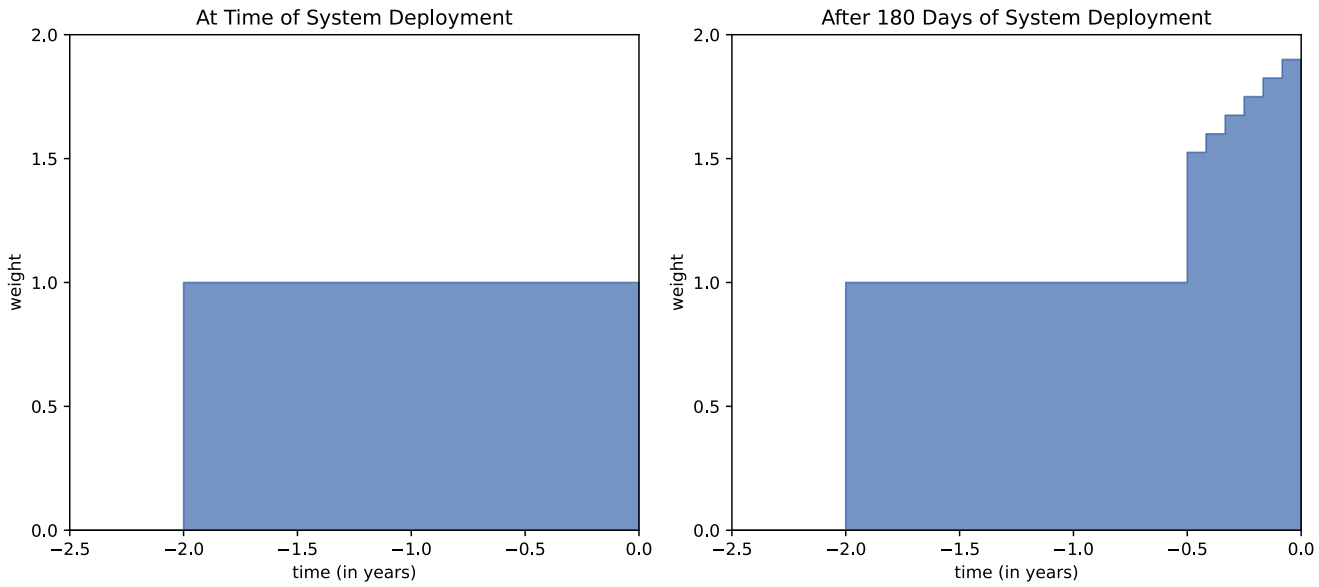


Figure 6.2: Data Weights

As explained at the beginning of this chapter, the naïve approach to continuously retrain the classification model on the entire dataset is the most straightforward to achieve the desired property of being able to learn from new data. But, this naïve approach is not performant and problematic when the dataset is too large. Incremental learning (IL) is a subfield of machine learning that deals with this challenge. Luo et al. (2020) define incremental learning as following:

> *Incremental learning, or continual learning, can continuously acquire new knowledge from new data samples while maintaining most of the previously learned knowledge. IL allows models to adapt and learn in real-time without the need to retrain from scratch each time new data becomes available.*

By not requiring access to the complete data set when updating the classification model, IL is computationally fast, which is the main benefit of IL. In the studied company, using the classification algorithms applied in *Chapter 6.2*, the data set was sufficiently small to not require IL techniques. Implementing a similar classification approach in other organizations may require the use of IL techniques, which we do not explore further in this thesis.

7. Evaluation

This chapter evaluates the performance of the *Classifiers* thoroughly, by making use of performance metrics. The results are evaluated with respect to the business objectives. Based on the results, at least one of the five classifiers is selected to be deployed later. The chapter ends with the *Parameter Optimisation* of the classifier(s) to be deployed.

7.1. Classifiers

All classifiers, with its described parameters, are evaluated within the generated test design, i.e., using stratified k-fold cross-validation with $k = 10$. Common evaluation metrics in classification are the accuracy, precision, recall, and F-score (Grandini, Bagli, & Giorgio Visani, 2020). In multi-class classification tasks, these measures are calculated using the true positives tp_i, false positives fp_i, false negatives fn_i, and true negatives tn_i per class C_i. *Table 7.1* shows the formulas for the evaluation metrics to be used (Sokolova & Lapalme, 2009).

Table 7.1: Metrics for Classification (Sokolova & Lapalme, 2009)

Metric	Formula
Accuracy	$\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}$
Precision	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fp_i)}$
Recall	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fn_i)}$
F-Score	$\frac{(\beta^2 + 1) \times precision \times recall}{\beta^2 \times precision + recall}$

Since k-fold cross-validation with $k = 10$ is used, there are 10 different test subsets, each having their own metric score. Therefore, the averages of these metrics over the 10 test subsets are computed and used for evaluation. *Table 7.2* presents the results of the five classifiers, tested in the generated test design. The standard deviation depicts the standard deviation of the accuracy among the 10 test subsets, to check the variation within the accuracy among the different test subsets. Furthermore, some performance metrics for the computational time are shown. $t_{predict}$ is the computation time in seconds to make predictions on 1,000 data records. t_{train} is the computation time in seconds to train the classifier on 250,000 data records. Some classifiers do not require training. In that case, training means fitting the classifier, i.e., indexing and optimizing the data in the background such that predictions are made fast. For the current data set, those are realistic numbers. The data set would not be much larger

than 250,000 data records and on daily (or weekly) basis, the number of invoices would normally not exceed 1,000, so not more than 1,000 predictions would need to be made at a time. The execution of the Python code is done in the Celonis Machine Learning Workbench, which is a cloud-based server.

Table 7.2: Evaluation of Classifiers

Classifier	Accuracy	St. Dev.	Precision	Recall	F-Score	$t_{predict}$ (s)*	t_{train} (s)**
GNB	0.779	0.002	0.942	0.779	0.803	0.001	0.578
kNN	**0.989**	**0.001**	**0.989**	**0.989**	**0.989**	1.147	0.846
RF	**0.989**	**0.001**	**0.989**	**0.989**	**0.989**	0.010	4.743
SVM	**0.989**	**0.001**	**0.989**	**0.989**	**0.989**	0.201	16.078
DT	**0.989**	**0.001**	**0.989**	**0.989**	**0.989**	**0.000**	**0.349**

* To predict 1,000 data records
** To train/fit the classifier on 250,000 data records

Table 7.2 shows that remarkably, four out of the five chosen classifiers reach the same level of accuracy. It is difficult to state why exactly four out of five classifiers have the same accuracy, however, presumably the reason is that the class labels are such highly distinguished that most of the tested classifiers can reach this level of accuracy that is 'optimal' with the given data set. The presumption is reasonable, knowing that tax codes are based on legislation and there should not be any ambiguity nor indistinct tax codes. All the other accuracy performance metrics are the same for kNN, RF, SVM, and DT. The average accuracy of 0.989 is sufficiently high, such that the number of invoices to be labelled with a warning is reasonable. It shows that the tax codes can be predicted accurately with multiple classifiers. There is therefore no motive to explore more sophisticated classifiers, for example deep learners or gradient boosters. The GNB classifier gives clearly less accurate results. It seems that these four classifiers have reached an optimum level of accuracy that is possible for this data set. The extremely low standard deviations indicates that the results are stable among the 10 different test subsets. In terms of accuracy, there is therefore no single preferred classifier, as kNN, RF, SVM, and DT all perform equally well. Regarding the time performance metrics, there is some difference between the classifiers. The DT performs the fastest considering making predictions as well as its training phase. kNN is the slowest in making predictions, while the SVM classifier has the longest training time. Nevertheless, all classifiers have a reasonable computation time, considering that the code can be executed at night outside working hours. Thus, although the DT has the lowest computation time, there is no preference for a classifier considering the computation time. In conclusion, kNN, RF, SVM, and DT are equally preferred regarding both accuracy metrics and time metrics. But, a decision is required about which classifier to deploy. Due to its simplicity and because it is not too sensitive for outliers, the kNN classifier is chosen to be deployed.

7.2. Parameter Optimisation

Since the kNN classifier will be deployed, it is important that its parameters are tuned to its optimal values. This is done by testing a wide range of possible values for the parameters. The results for every combination of parameters are then computed. The kNN classifier is configurable with the following parameters and possible values:

- k $\in \{1, 2, \dots, n-1\}$
- Weight function $\in \{uniform, distance\}$
- Minkowski metric $\in \{Manhattan, Euclidean\}$

k is the number of neighbours to use, and n is the number of data records in the data table. For the weight function, one may choose to use uniform weights, i.e., every data object has an equal weight. The alternative is to weight data points by the inverse of their distance, meaning that closer neighbours have a greater influence than neighbours which are further away. Finally, the Minkowski metric used to calculate the distances needs to be determined. The Minkowski metric is a way to measure distance, where the options in kNN are Manhattan and Euclidean distance.

A large parameter space is explored, testing all combinations of weight functions, Minkowski metrics, and k values between 3 and 17, in steps of 2. The evaluation is again done using stratified k-fold cross-validation, so the accuracy is the average accuracy among the 10 test subsets. The splits between training and testing data are the same every time. *Figure 7.1* shows the results of the parameter evaluation. The bar charts show the results for using the Manhattan distance (left) and the Euclidean distance (right). To be able to compare the results easier, the horizontal line depicts the maximum accuracy among the tested parameter values. Clearly, there is no preference for the Minkowski distance, as the accuracy results are the same. For the weight function, the results do not significantly differ, but the distance weight function scores slightly better. However, it can be seen that when k is low, the accuracy is lower. When $k \geq 7$, the accuracy does not further improve. In the deployment of the kNN classifier, it is chosen to use $k = 11$, because it is one of the values that achieves the highest accuracy and it is not too small to be very sensitive to outliers. The distance weight function is chosen, because it scores slightly better. Furthermore, the Euclidean distance is used for the Minkowski metric. For the Minkowski metric, there is no preference, so the default – Euclidean distance – is used. To summarise, the used parameters are:

- k $= 11$
- Weight function $= distance$
- Minkowski metric $= Euclidean$

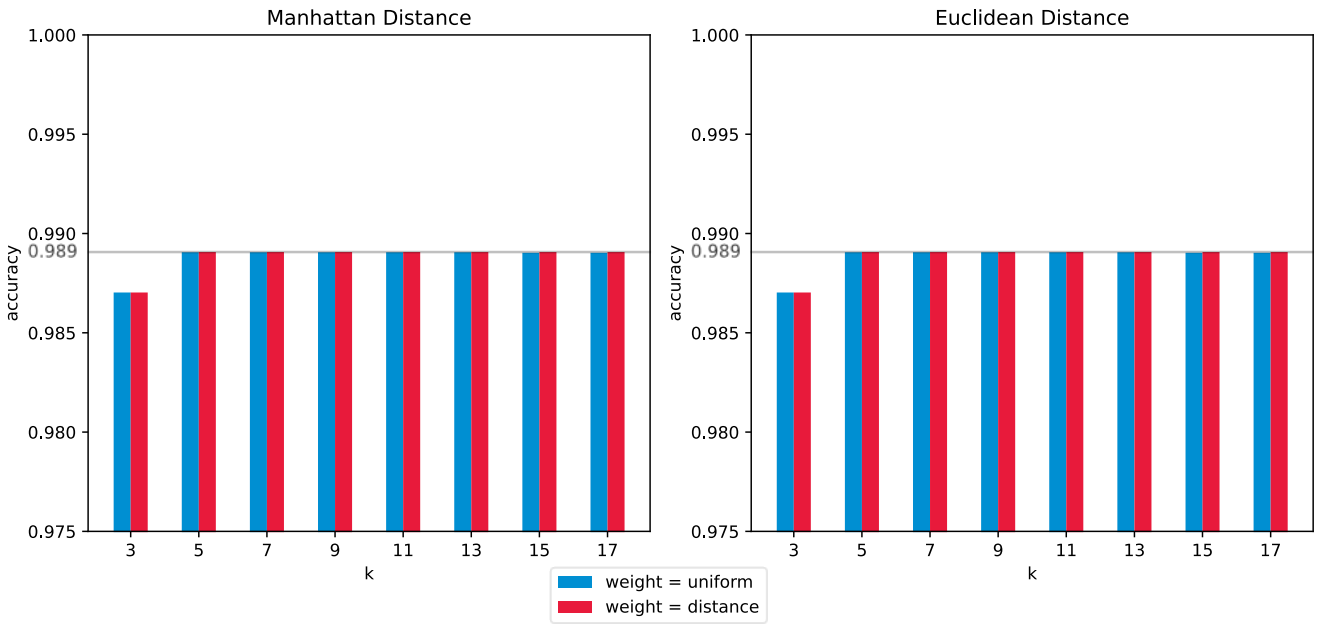


Figure 7.1: kNN Parameter Evaluation

8. Deployment

In *Deployment*, the final phase of CRISP-DM, the kNN classifier is to be deployed in a *System* wherein the classification model can be used by the client to prevent incorrect VAT reporting. It provides a real-time business solution, which is explained in detail. Furthermore, it is mentioned how the system offers a solution to deal with tax legislation changes, which affect the effectiveness of the classifier. This is followed by a description of the *Data Loading* in the system. The monitoring of the business solution is not included, because it was never in scope of the thesis.

8.1. System

The design of the business solution is done while taking into consideration the business requirements from *Determine Business Objectives.* The system should primarily be a practical tool to prevent incorrect VAT reporting. The emphasis in the system should be on invoices for which a warning has been triggered, as illustrated in the framework in *Figure 4.2.* Specifically, a warning means that the invoice is likely to have the incorrect tax code and therefore requires a second check by the accountant. The system provides a platform wherein the user can interactively confirm the correct tax codes of invoices. The confirmed tax code then serves as feedback for the classifier, which is continually reconstructed with the new knowledge of the feedback. The deployment is done in a Celonis View, which is a functionality of Celonis wherein data can be visualised in an interactive manner with the user. Celonis uses Process Query Language (PQL), a domain-specific language of Celonis, tailored towards process data. PQL's syntax is inspired by SQL, but specialised for process-related queries (Celonis, 2023). Within the context of the system to be deployed, PQL is used to calculate measures which are displayed in the user interface (UI). Such measures are KPIs like the number of invoices displayed, the number of potentially wrong tax codes, the number of tax codes confirmed, and the number of tax codes unconfirmed. The UI that has been developed in Celonis is shown in *Figure 8.1*. 'Tax Code – SAP' shows the tax code currently stored in the SAP as initially entered by the accountant, 'Tax Code – Suggestion' shows the tax code as predicted by the classifier, and 'Tax Code - Final' is the tax code to be confirmed in the UI. For intuitiveness purposes, not only the tax code is given, but also the tax code description (T007S.TEXT1).

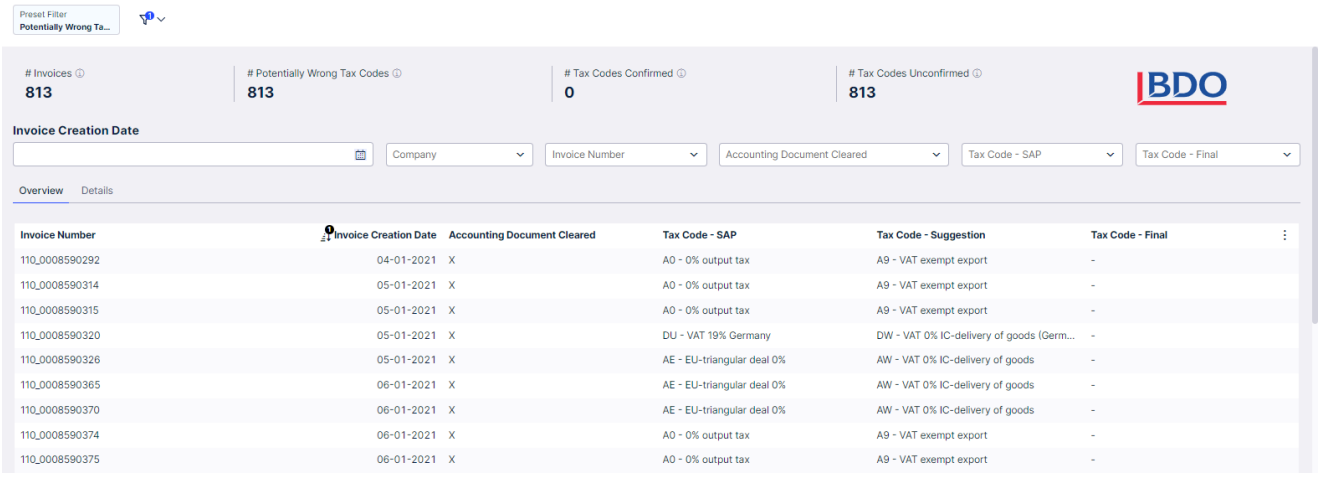


Figure 8.1: Celonis User Interface

By clicking on ‘Tax Code – Final’ of an invoice, a drop-down menu opens with all possible tax codes. The accountant can then select the definite tax code. Tax codes can also be confirmed in bulk by selecting invoices in the table. By default, the UI filters on potentially wrong tax codes, i.e., only invoices are shown for which the current tax code in SAP is different from the tax code as predicted by the classification model. Disabling this filter allows the user to view all invoices, so also the invoices for which no potential risk was identified by the classifier. Furthermore, the user can filter on invoice creation date, company, invoice number, whether the accounting document was cleared or not, the current tax code, and the confirmed tax code. Thus, the accountants can navigate in a user-friendly manner through the invoices that are of interest to them. Accountants can use the invoice number to look up PDF invoice documents, or they can navigate to the tab ‘Details’, in which the input variables for each invoice are shown. The accountant can use both sources to determine what the correct tax code is. This should then be confirmed in ‘Tax Code - Final’, which is used as feedback for the classification model. ‘Tax Code - Final’ is more reliable than ‘Tax Code – SAP’, which is the tax code after a single check by the accountant, while ‘Tax Code – Final’ is checked twice. In the current system, filling in ‘Tax Code – Final’ does not overwrite the tax code in SAP. The attribute only serves to provide feedback for the classifier. So, if ‘Tax Code – Final’ differs from ‘Tax Code - SAP’, the currently stored tax code in SAP is incorrect and it would still need to be edited. This was the current requirement by BDO, as the business solution has only recently been developed and it is therefore desirable to do some testing prior to connecting it to SAP and enabling the tax codes in SAP to be overwritten from within the UI. This functionality may be added in the future by BDO and would make the work of accountants more efficient. Note that confidential information, such as the company name, is excluded from *Figure 8.1.* In the Celonis View, this information is included.

The classifier's ability to learn from an ongoing stream of data is implemented by continuously reconstructing it, i.e., the classifier is retrained daily on the complete data set. The classifier uses the user-entered feedback from 'Tax Code – Final'. As illustrated in *Table 7.2*, the computation time for fitting the kNN classifier is very low: Approximately 1 second. Therefore, sophisticated methods such as incremental learning, as introduced in *Chapter 6.3*, are not necessary to make the classifier able to learn from streaming data. For simplicity reasons, the daily retraining of the classification model on the complete dataset is therefore favourable.

One challenge that has not yet been considered in *Chapter 6.3* is how to deal with changing tax codes due to tax legislation changes. After a change in legislation, a transaction that would previously get tax code A could now get tax code B. Tax legislation typically changes from one day to another, usually at the start of a new month or year, so there is a clear cut-off. With the framework as explained in *Chapter 6.3*, the classifier would need time to correctly predict a transaction which is affected by the tax legislation change. Although the classification model develops rapidly to new data because of the data weights, it still needs plenty of time and it is hence clearly not an effective solution to manage changes in tax legislation. The UI offers a solution to this. Particularly, the user can adjust the 'Tax Code – Final' attributes with the tax codes according to the tax code legislation of today. The user should select all transactions which would have received another tax code with the new legislation. Then, 'Tax Code – Final' should be entered with the tax code that those transactions would have with the new tax legislation. The classifier will then be retrained with the knowledge of the new tax legislation, and this knowledge overwrites the knowledge of the old tax legislation. As a consequence, the classifier would immediately be able to correctly predict those transactions of which the tax code is affected by the tax legislation change. Since the entered tax code in 'Tax Code – Final' does not overwrite the tax code in SAP, this entails no issues. Should BDO in the future enable 'Tax Code – Final' to overwrite the tax code in SAP, then accountants have to be cautious with this approach. Only accounting documents which are cleared, and therefore already reported with the tax code that was in SAP at the time of clearing the accounting document, may have 'Tax Code – Final' overwritten with the tax code that would be applicable after the tax legislation change. The VAT is then already reported, so the tax code cannot be changed anymore.

8.2. Data Loading

For the tax compliance system, a data loading framework is necessary. When the system is deployed, predictions are made on the entire data set. The data is then loaded in the UI. Since the accuracy of kNN in the test design is 0.989, approximately 1% of the invoices would be marked as having a potentially incorrect tax code, assuming that the accuracy of making predictions on the whole data set is similar to the accuracy in the test design. This means that initially, a large amount of invoices with

potentially wrong tax codes are loaded into the UI. So, at the beginning, the accountants would have many invoices to double check. That instantly generates a lot of feedback, from which the classification model would benefit as its quality improves. Following the initial data load, the system needs to be updated on a regular basis, such that new data is loaded in the UI. This will occur on a daily basis, as new invoices are created daily. The process for the daily data load is as follows: First, the new data and the existing data, including the user-generated feedback ('Tax Code – Final'), are loaded. This data set is subsequently pre-processed, which includes data construction, data integration, and assigning the data weights. Then, the kNN classifier is refitted on the data set. Predictions are made for the for new data. If the prediction is correct, i.e., the predicted tax code is the same tax code as the tax code in SAP, then the assumption is made that the tax code is correctly classified. If the tax code is predicted incorrectly, the invoice will be marked as having a potentially incorrect tax code, and these invoices will be highlighted in the UI since by default, the UI filters on invoices with potentially incorrect tax codes. The users can then enter the definite tax code, for either invoices with or without a warning. This process is repeated on a daily basis during the night, because outside working hours the UI can be refreshed securely as no one will be using the UI. The daily data loading process is visualised in *Figure 8.2*. A round-shaped parallelogram stands for a data load, a rectangle means a process, a diamond stands for a decision, and a sharp-shaped parallelogram means input.

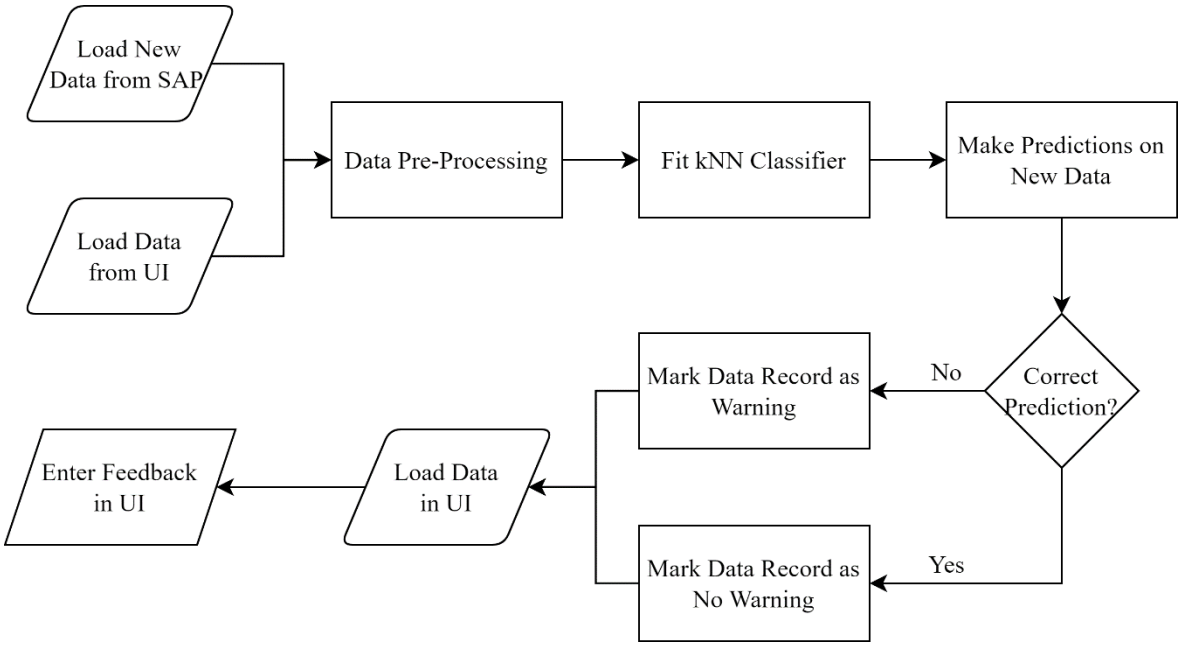


Figure 8.2: Daily Data Load

9. Conclusion

To conclude the work, the problem statement and research question from the *Introduction* are reviewed. The problem statement – *VAT reporting is prone to errors* – has been at the core of the thesis. Although manual VAT reporting remains as error-prone as it was, a tax compliance system has been developed, which can identify and prevent these manually reported incorrect VAT statements. Therefore, overall error-proneness of VAT reporting can be limited with the usage of this tax compliance system. The research question – *How can data mining techniques be used to increase VAT compliance?* – has been answered with the developed framework. This framework, or artifact, has been deployed such that it can readily be used as a tax compliance system. A summary of this artifact is given in the *Summary of Results*. This is followed by the *Contribution* of this work to the research field. Subsequently, the *Limitations* of the thesis are discussed and suggestions for *Future Research* are proposed. At the end, *Legal Aspects* that affect this work are mentioned.

9.1. Summary of Results

Reviewing the objectives from *Determine Business Objectives*, the aim of this thesis was to develop the following:

- A classification model to predict tax codes
- The deployment of the classification model to prevent incorrect VAT reporting

After thorough data understanding and data preparation, the classification model was developed. The data utilised is real-world SAP data from one of BDO's clients. The classifier is developed only for the O2C process, as the required data for the P2P process is unfortunately not available. So, the classifier can only predict tax codes for outgoing invoices, and not for incoming invoices. With an accuracy of 98.9% in the test design, the classifier can accurately predict tax codes. Remember that even though efforts have been made to clean the training dataset, the correctness of class labels cannot be guaranteed. But, to the best of our knowledge, the class labels in the dataset are mostly correct. Four of the five evaluated classifiers score equally good in accuracy metrics, but the kNN classifier is eventually chosen to be deployed. The deployment is done in a Celonis View. In this UI, the invoices with its given tax code and predicted tax code are displayed in a user-friendly manner. Specifically, invoices for which the predicted tax code is different from the given tax code are highlighted, as these tax codes are marked as potentially incorrect. These invoices therefore require a second check by the accountants, which are the users of the UI. The accountants can then confirm the correct tax code in the UI, using relevant invoice data that is shown in the UI. The confirmed tax codes are used as feedback for the kNN classifier, as it is refitted incorporating the feedback. To make the classifier adapt rapidly to new data and let the classifier evolve dynamically, heavier weights are assigned to more recent data

records and the classifier is continuously reconstructed by retraining it on the entire dataset. The objective of doing so is that the classifier can identify incorrect tax codes with increasing accuracy. Although the evaluation has demonstrated that the classifier can accurately predict tax codes, the effectiveness of the classifier in being able to identify incorrect tax codes is not empirically proven, since that can only be done after the UI has been in use for a certain period of time, such that enough evidence is collected. In conclusion, the results include the theory and implementation of the system to prevent incorrect VAT reporting, but its practical functioning is not evaluated.

9.2. Contribution

The thesis applies data mining techniques within the tax domain in a scientific and practical context. The field of tax compliance faces the challenge that VAT may be reported incorrectly on VAT statements. This thesis is an attempt to tackle this issue with the use of ML methods. The results have shown that tax codes can accurately be predicted with multiple classifiers on real-world data. Furthermore, a concept has been proposed to deploy the classifier within BDO, which can potentially be used to increase VAT compliance.

In the research field of data mining and tax compliance, the existing research is limited to the work of Lahann, Scheid & Fettke (2019) and Bardelli et al. (2020). The work of Lahann, Scheid & Fettke (2019) is similar in the sense that they use real-world SAP data to predict tax codes. They also present an approach to find potential anomalies, but their framework differs from the approach suggested in this thesis. A data record is considered a potential anomaly by Lahann, Scheid & Fettke (2019) if the predicted tax code by the classifier deviates from the tax code given by the rule-based system. In this thesis, there is no rule-based system, but the tax codes were determined manually by VAT experts. Furthermore, learning from a data stream is not discussed in neither of the two mentioned papers. The focus of the work of Bardelli et al. (2020) is particularly on the classifiers themselves, and does not propose a framework to deploy a classifier.

9.3. Limitations

The *Summary of Results* showed that not all original objectives have been realised. The data set entails several limitations, causing certain objectives to be infeasible or certain limitations to be taken into consideration. First of all, the classifier only functions for invoices from the O2C process (outgoing invoices) and not for invoices from the P2P process (incoming invoices). The necessary data attributes were simply not available in the data set, causing it to be an unrealistic goal with the given resources.

The final solution is therefore less valuable than originally expected, as no tax compliance system could be developed for incoming invoices.

A second limitation about the data set is the correctness of the class labels y, the tax codes. It is known that they are not entirely correct, because otherwise there would be no need for a tax compliance system as presented in this thesis. But, the exact correctness is not known. In fact, the goal of the thesis is to identify which of these class labels y are incorrect, and subsequently having them corrected by the tax experts. To achieve that effectively, the class labels y should not be too polluted, i.e., having too many incorrect tax codes. Especially structural tax codes mistakes are troublesome for the classifier, since the classifier then learns wrong patterns in the data. Incidental wrong tax codes are less problematic and more likely to be identified by the classifier, as it can very likely be assumed that a vast majority of class labels y is assigned correctly, and thus the kNN classifier would properly classify such invoices. However, the kNN classifier would probably not identify invoices for which the tax codes are assigned structurally incorrect in the data. These invoices would then also not be highlighted as potentially incorrect within the UI, and therefore risk to remain unnoticed. Only if the tax experts interfere and correct such structural incorrect classifications, the classifier can learn from that in the future due to its capability to learn continuously. Yet, the tax codes are assigned by tax experts, so the correctness can be assumed to be high. But, in conclusion, no guarantees can be given, and one needs to be aware of the potential risk.

Another limitation of the data set is its lack of diversity in terms of industries and tax perspectives. The data set only contains data from one Austrian production company. The company operates internationally and has a high diversity of types of transactions, so there is sufficient data to train the classifier for many different scenarios. But, the tax legislation perspective is limited to an Austrian perspective. The kNN classifier is capable to identify new tax codes in the future. However, it uses only the set of input variables as decided in *Data Construction*. The determination of this set of input variables is done consideration all potentially relevant input variables from an international perspective. However, it cannot be guaranteed that some input variables have been overlooked, as some countries could have some very specific and exceptional tax scenarios. Such exceptional scenarios are rare, as the chosen set of input variables covers almost all usual tax scenarios, according to BDO's tax expert. Nevertheless, for the generalizability of the classification model to future data sets (especially non-Austrian), this potential issue should be kept in mind and additional input variables may have to be added.

The data set originates from SAP. SAP systems consist of a combination of standardised and customised tables, so the customised tables differ from each SAP client. The classification model is built considering the configurations of BDO's customer, who is the SAP client. Since no communication with the customer was possible, the specific SAP configurations used remained unknown. It is not precisely clear what data tables are customised and how. Hence, certain assumptions had to be made. When

applying the classification model on new data sets in the future, the data scientist needs to check if the used data attributes are used in the same way.

Finally, the deployed system has not been empirically tested. The effectiveness of the framework for continuous retraining is not evaluated. The theoretical functioning of the system is presented, but not empirically evaluated. This was never in scope of the thesis, as it was originally considered infeasible considering the time constraint of six months. In *Future Research*, this is also addressed.

9.4. Future Research

For future research directions, two proposals are presented. First of all, the possibilities to build a classifier for incoming invoices (P2P process) can be explored. The lack of a classifier for incoming invoices is a limitation of this thesis. On a different and cleaner data set, it is possible to develop such a classifier. Lahann, Scheid & Fettke (2019) have already illustrated that it is possible with SAP data.

Secondly, since the deployed system is not empirically evaluated, it would be interesting to test how effective the presented framework identifies and prevents incorrect VAT reporting. This future research proposal is specifically applicable for BDO, where the system has already been deployed, but can also be evaluated by other researchers who implement this system. The effectiveness of the classifier in identifying potential incorrect tax codes can be evaluated once data has been collected, by tracking how many tax code changes are made after the classifier had proposed a change. Furthermore, the quality of the framework for continuous retraining should be evaluated.

9.5. Legal Aspects

The thesis has been conducted in cooperation with BDO and real-world data is used from one of BDO's clients. It is agreed with BDO that the data set is not published, as it contains confidential data attributes. For illustrative purposes, some non-confidential data attributes and examples are shown throughout this work. However, confidential data attributes, such as company or customer names, are not shown.

Bibliography

Aggarwal, C. C. (2015). *Data Mining: The Textbook.* Springer. doi:10.1007/978-3-319-14142-8

Austrian Ministry of Finance. (2023). *Reverse-charge*. Retrieved October 2023, from Austrian Ministry of Finance: https://www.bmf.gv.at/en/topics/taxation/vat-assessment-refund-n/supplying-in-austria/reverse-charge.html

Azevedo, A., & Santos, M. (2008). KDD, SEMMA and CRISP-DM: A Parallel Overview. *IADIS European Conference Data Mining*, (pp. 182-185). Amsterdam.

Azhar, M., & Thomas, P. (2019). Comparative Review of Feature Selection and Classification modeling. *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)* (pp. 1-9). Mumbai: IEEE. doi:10.1109/ICAC347590.2019.9036816

Bardelli, C., Rondinelli, A., Vecchio, R., & Figini, S. (2020). Automatic Electronic Invoice Classification Using Machine Learning Models. *Machine Learning and Knowledge Extraction, 2*, 617-629. doi:10.3390/make2040033

BDO. (2023). Retrieved October 2023, from BDO: https://www.bdo.at/

Carriere, D. (2013). Retrieved October 2023, from Geocoder: https://geocoder.readthedocs.io/

Celonis. (2023). *Celonis Product Documentation*. Retrieved October 2023, from Celonis: https://docs.celonis.com/

EU VAT Directive. (2023). European Union.

European Commission. (2023). *VAT identification numbers*. Retrieved October 2023, from European Commission: https://taxation-customs.ec.europa.eu/vat-identification-numbers_en

Eurostat. (2022, October 24). *Tax revenue statistics.* Retrieved October 2023, from Eurostat: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Tax_revenue_statistics

facultas.wuv. (2018). Module 4: The Principles of Value Added Taxation. In K. Ubelhoer, S. Pfeiffer, E. Huisman, & E. Schaffer, *Introduction to Austrian Tax Law.* Vienna.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 27-34. doi:10.1145/240455.240464

Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, 254-280. doi:10.1016/j.techfore.2016.08.019

Gehrke, N., & Thams, R. (2010). VAT Compliance: Einhaltung umsatzsteuerlicher Anforderungen beim Prozess-und Datenmanagement in ERP-Systemen. *Multikonferenz Wirtschaftsinformatik 2010* (pp. 569-581). Göttingen: Universitätsverlag Göttingen.

González, P., & Velasquez, J. (2013). Characterization and detection of taxpayers with false invoices using data mining techniques. *Expert Systems with Applications, 40*, 1427-1436. doi:10.1016/j.eswa.2012.08.051

Grandini, M., Bagli, E., & Giorgio Visani. (2020). Metrics for Multi-Class Classification: an Overview. *ArXiv*. doi:10.48550/arXiv.2008.05756

Gregory, S. A. (1966). Design Science. In S. A. Gregory, *The Design Method* (pp. 323-330). Boston: Springer. doi:10.1007/978-1-4899-6331-4_35

Grigorev, A. (2021). *Machine Learning Bookcamp.* Manning.

Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann. doi:10.1016/C2009-0-61819-5

Hevner, A. R., Salvatore, M. T., Park, J., & Ram, S. (2004). Design Science in Information Systems. *MIS Quarterly, 28*(1), 75-105. doi:10.2307/25148625

International Chamber of Commerce. (2023). *Incoterms Rules*. Retrieved October 2023, from International Chamber of Commerce: https://iccwbo.org/business-solutions/incoterms-rules/

Kaladari, N., & Al-Sartawi, A. (2023). The Capabilities of Using Artificial Intelligence for Value-Added Tax. *EAMMIS 2022: From the Internet of Things to the Internet of Ideas: The Role of Artificial Intelligence*, 611-616. doi:10.1007/978-3-031-17746-0_47

Lahann, J., Scheid, M., & Fettke, P. (2019). Utilizing Machine Learning Techniques to Reveal VAT Compliance Violations in Accounting Data. *2019 IEEE 21st Conference on Business Informatics (CBI)* (pp. 1-10). Moscow: IEEE. doi:10.1109/CBI.2019.00008

Luo, Y., Yin, L., Bai, W., & Mao, K. (2020). An Appraisal of Incremental Learning Methods. *Entropy, 22*(11). doi:10.3390/e22111190

OECD. (2022). *Consumption Tax Trends 2022: VAT/GST and Excise, Core Design Features and Trends.* Paris: OECD Publishing. doi:10.1787/6525a942-en

Pyle, D. (1999). *Data Preparation for Data Mining* (1st ed.). San Francisco: Morgan Kaufmann.

Saltz, J. (2021). CRISP-DM for Data Science: Strengths, Weaknesses and Potential Next Steps,. *2021 IEEE International Conference on Big Data (Big Data)* (pp. 2337-2344). Orlando: IEEE. doi:10.1109/BigData52589.2021.9671634

Sammut, C., & Webb, G. I. (2017). *Encyclopedia of Machine Learning and Data Mining* (2nd ed.). New York: Springer. doi:10.1007/978-1-4899-7687-1

SAP. (2023). *SAP*. Retrieved October 2023, from https://www.sap.com/

SAP Datasheet. (2023). Retrieved October 2023, from https://www.sapdatasheet.org/

Sarmento, J. M. (2023). Value-Added Tax. In J. M. Sarmento, *Taxation in Finance and Accounting. Springer Texts in Business and Economics.* Springer International Publishing. doi:10.1007/978-3-031-22097-5_9

scikit-learn. (2023). Retrieved October 2023, from scikit-learn: Machine Learning in Python: https://scikit-learn.org/

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 13-22.

Sherman, R. (2015). *Business Intelligence Guidebook: From Data Integration to Analytics.* Amsterdam: Morgan Kaufmann. doi:https://doi.org/10.1016/C2012-0-06937-2

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 427-437. doi:10.1016/j.ipm.2009.03.002

Vaisman, A., & Zimányi, E. (2022). *Data Warehouse Systems* (2nd ed.). Berlin: Springer. doi:10.1007/978-3-662-65167-4

van Bree, G., Staudinger, S., Burgstaller, F., Schiff, F., & Schütz, C. G. (2024). Using Machine Learning to Identify Incorrect Value-Added Tax Reports. *AMCIS 2024.* Salt Lake City. Retrieved from https://aisel.aisnet.org/amcis2024/acctinfosys/acctinfosys/4

van der Aalst, W., & Carmona, J. (2022). *Process Mining Handbook.* Springer International Publishing. doi:10.1007/978-3-031-08848-3

van Doesum, A., & Nellen, F. (2021). *VAT in a Day.* Nijmegen / Leiden. Retrieved from https://ssrn.com/abstract=3867201

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (pp. 29-40). Manchester: AAAI Press.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *ACM International Conference Proceeding Series*, (pp. 1-10). doi:10.1145/2601248.2601268

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., . . . Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 1-37. doi:10.1007/s10115-007-0114-2

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
CRISP-DM	Cross Industry Standard Process for Data Mining
DT	Decision Tree
ERP	Enterprise Resource Planning
ETL	Extract, Transform, and Loading
EU	European Union
FK	Foreign Key
GNB	Gaussian Naïve Bayes
IL	Incremental Learning
KDD	Knowledge Discovery from Data
kNN	k-Nearest Neighbour
ML	Machine Learning
O2C	Order-to-Cash
P2P	Purchase-to-Pay
PK	Primary Key
PQL	Process Query Language
RF	Random Forest
SD	Sales and Distribution
SEMMA	Sample, Explore, Modify, Model, and Assess
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface
VAT	Value-Added Tax