MASTER'S THESIS

Resilience Mining: Detecting Shadow IT in IT Systems with Data Classification

Huisman, J. (José)

Award date:
2020

Link to publication

Resilience Mining: Detecting Shadow IT in IT Systems with Data Classification

Opleiding:	Open Universiteit, faculteit Management, Science & Technology Masteropleiding Business Process Management & IT
Degree programme:	Open University of the Netherlands, Faculty of Management, Science & Technology Business Process Management & IT master's programme
Course:	IM0602 BPMIT Graduation Assignment Preparation IM9806 Business Process Management and IT Graduation Assignment
Student:	José Huisman
Identification number:	
Date:	26 January 2020
Thesis supervisor:	Dr. Lloyd Rutledge
Second reader:	Dr. Guy Janssens
Version number:	1.0
Status:	Final version

Abstract

This paper shows how and to what extent data mining classification algorithms detect shadow IT in databases. Data classification points out data fields with coded data, indicating agreed procedures. Discovered shadow IT usually exposes gaps or shortcomings in systems used, as well as opportunities for system improvements. Data classification algorithms focus on data to distinguish shadow IT, where other researchers focus on text classification, and association or clustering algorithms. Data classification aims to distinguish data structures in databases that do not follow formal workflows, not accepted or supported by the IT department. On a synthetic dataset, supervised learning with data classification is examined with Naïve Bayes, k-NN and the probabilistic classifiers Decision Trees and Logistic Regression. Due to working with Euclidian distances, k-NN and Support Vector Machines algorithms are not suitable. Classifying the imbalanced dataset often runs into overfitting and other issues. These require special attention and affect the selection of performance metrics. Accuracy, precision, recall, specificity and the area under the curve are evaluated. System improvement suggestions are, for example, to add dedicated code fields instead of a fictitious date to avoid bias and adding a validity period to enrich data and make it more dynamic.

Key terms

Shadow IT, workarounds, data classification, classification algorithms, binary classification and performance measurement

Contents

List of figures

List of tables

1. Introduction

IT systems use is not always in accordance with system design. Creative employees strive for more effective and efficient ways to do their work (Houghton & Kerr, 2015). Shadow IT represents all hardware, software, or any other solutions used by employees in an organization without any formal IT department approval (Silic & Back, 2014). Even when the functionality exceeds the desired requirements of a system and even if users are still satisfied with it, there can still be a need for creating shortcuts and workarounds in striving for increased efficiency. This need results in the creation of shadow IT. However, when it comes to legislation, it might not be governance compliant. Shadow IT is also called "rogue IT" (McLaughlin, 2014), because of its informal and unsanctioned use of IT resources without notice to – or approval from – IT organizations (Behrens, 2009). "Feral IT" another synonym, can spark innovation (Silic, Silic, & Oblakovic, 2016) or can be more efficient than using legacy systems (Behrens, Sedera, Behrens, & Sedera, 2004; Houghton & Kerr, 2015).

Shadow IT can also be risky because of lack of governance compliance and mitigations. This can lead to legislation incompliances, inconsistent business logic, increased risks for data loss or leaks and wasted investment (Silic & Back, 2014). Furthermore shadow IT can undermine the official system (Strong, D. M., & Volkoff, 2004) or even damage organizational data and processes (Oliver, D., Romm, 2002). Shadow IT often arises from good intentions, for example, when IT systems fail to meet user requirements, and creative people who want to do their job in a formal or informal way, ask for workarounds or shortcuts. Workarounds also indicate shortcomings of current IT systems, where the IT manager can benefit from such an invitation for improvement (Kopper, 2017).

Shadow IT and workarounds demonstrate how applications are actually used. Process workarounds are revealed from transactional systems with process mining (Outmazgin & Soffer, 2016). Finding data related workarounds requires other techniques (Outmazgin & Soffer, 2013). Resilience mining shows data diverging from data structures as designed. This divergence evinces the real user needs and system requirements. Workarounds in problem solving are often recognized as opportunities for system improvements (Paparella & Horsham, 2018). Improvement refers to how things are done and making it more effective and efficient. Innovation is when system usage provokes new ideas to create value for an organization (Litwack, 2018).

Data Mining has been used for finding patterns and deriving knowledge from data. Algorithms have been developed and refined to detect shadow IT in a database. Even if an algorithm is best suitable for solving a problem, it is not certain that it is also the best solution for any other problem (Çiğşar & Ünal, 2019). Predictive algorithms as data classification are supervised algorithms with a categorical variable as a target. It distinguishes certain feature values from others (Provost & Fawcett, 2013). Quality aspects concern applying different algorithms, going beyond data and model quality approaches (Domingos, 2012).

Different from other studies on workarounds, this research investigates how shadow IT and workarounds in databases can be found. Databases in our research are highly structured with features and data values as observations, also to be seen as columns and rows. Data values can be characters, text or integers, but no images, sound or other artefacts. Data can be in any data field, even in text notation fields, but we are only focusing on data classification. Examples of shadow IT are data registered in wrong fields or wrong formats. It also occurs while filling out fields according to agreed codes and instructions. A fictional date as "99-99-99" or a future date is common (Outmazgin & Soffer, 2013). By analyzing the appearances and causes of shadow IT, deficiencies or weaknesses of current systems become clear. To get a clear picture of these shortcomings requires much time, effort, knowledge and willingness of employees involved. Therefore, it can be of great value if there are more effective and efficient ways to trace these situations.

In this research, shadow IT in databases should be detected using data classification. Discovered shadow IT can be transformed into suggestions for formal accepted and approved IT. To get there, data classification models are developed. Also, model performance is measured, and classifiers are compared by metrics. For achieving these goals, the main research question is set up as follows:

> *How, to what extent and in what situations can data classification algorithms detect shadow IT in databases, and how can their performance be compared?*

Based on performance measurement the examined algorithms are evaluated and ranked. To achieve the research objective, we consider the following research sub-questions:

RQ1: What data or data structures are indicators for shadow IT in a database?
RQ2: How can classification algorithms predict and detect shadow IT?
RQ3: How can the performance of data classification models be compared?
RQ4: How can detected shadow IT be converted into suggestions for system improvement?

The answers to these questions support the detection of shadow IT in a database. The first question shows how to find data used for shadow IT. The second question results in a selection of classifiers with performance measurement, whereas the third question leads to performance comparison of the models with the selected metrics. The last question brings up suggestions on how to use the findings as a basis for system improvements.

For this research the database from the Dutch Open University course “Development Practical” (Open Universiteit, 2010) is explored. Findings might elicit system improvement suggestions. This paper describes a part of an umbrella research, executed by the Resilience Mining Thesis Circle. Each member examines a type of data mining or text-mining techniques, like classification, association rule mining, clustering and outlier detection. Finally, the individual results fall together in an overall conclusion for the shadow IT problem. The next step in generalization findings is to come up from repeating this analysis on other databases.

The remainder of this report is organized as follows: in the second chapter, the theoretical framework is set up with converging definitions and theoretical concepts of data classification, where in chapter 3 the research approach is explained. Chapter 4 presents findings and results. In chapter 5 discussion flows into conclusions with recommendations for practice and for further research. This paper closes with reflections.

2. Theoretical framework

2.1. Research approach

To know what we are looking for, we first must specify shadow IT and workarounds, and create a way to mark it as such. Then we will narrow our scope to a database. For this experiment data, several classification algorithms are explored and most appropriate are selected. Literature study is used to find answers to the following questions:

1) What characteristics of the database content indicate that it should be flagged as shadow IT?
2) Which data classification algorithms from the most common ones are suited to this research?
3) How can the classifiers' performances be measured and compared?
4) How can shadow IT or workarounds be transformed into system improvement suggestions?

Resources used are the Open Universiteit Library, Google Scholar, and the Fontys Library. Most relevant work is published by ResearchGate.net, Science Direct, Elsevier and SpringerLink. Search terms include "shadow IT" and "workarounds", but also "data classification", "classification algorithms" and "binary classification". For comparing classification algorithms "performance measurement" is used.

2.2. Implementation

A first search round yields about ten articles that are useful to lay a foundation concerning shadow IT. They merely contribute to definitions (Alter, 2014; Kopper & Westner, 2016) of shadow IT and scoping the research. Next, zooming in to databases and what to find there, because of our structured database, we focus on structured data and compliancy with standards like Data Format Description Language (Ibm, Oco, & Ibm, 2011). Non-text data like images, video and audio are out of scope. Free text is ignored because classification of data is emphasized, and text mining requires other techniques.

Regarding literature on data classification, we start from a taxonomy for data mining tasks (Sharda, Delen, & Turban, 2018). Next a lot of work on data classification is based on text mining (Mirończuk & Protasiewicz, 2018), other more specific algorithms (Aher & Lobo, 2014) and more advanced algorithms (Chen, Liu, Gong, & Gao, 2017), other datasets (medical, psychology, social studies, marketing and financial risk) (Çiğşar & Ünal, 2019) and other research approaches. These publications are usable at most in their similarity for our research.

A basic overview of classification algorithms (Wu et al., 2008) points at classifiers to investigate with the tool RapidMiner (Arunadevi, Ramya, & Raja, 2018; Marques & Bernardino, 2013). For each algorithm we use, we search for publications on experiences with approaches, performances and results. To compare our classifiers, different models are developed and optimized. Metrics for comparing classifier performance are accuracy, speed, robustness, scalability, interpretability (Stefanowski, 2008). In this research the confusion matrix is the center from which other metrics are derived (Sossi Alaoui, Farhaoui, & Aksasse, 2017).

2.3. Related circle research

Having a common research objective, the members of the Resilience Mining Thesis Circle approach the mining problem from different perspectives, like focus on data or text. Prediction consists of classification and regression. Classification tasks are supervised and use a categorical variable as target. Regression is supervised and use probabilistic algorithms with a numerical variable as label. Unsupervised models are for association rule mining, clustering and outlier detection. Outlier detection is investigated on both data and text. Using mainly the same or comparable datasets, papers on different techniques are delivered from the circle, as follows:

	Data	Text
Prediction	This paper	(Cate, 2020)
Association Rule Mining	(Sandfort, 2020)	(Rouwendal, 2020)
Clustering	(Spoel, 2020)	(Spronk, 2020)
Outlier Detection	(Koskamp, 2020)	

Table 1 Circle research

The main difference from prediction on free text is the restriction to data and data structures. Whereas the other data focused approaches are unsupervised techniques, classification is the only supervised one.

2.4. Results and conclusions

Shadow IT

Shadow IT is getting more and more attention in scientific research. Kopper distinguishes six types of appearances, where we only dive into workarounds and shadow IT in databases (Kopper & Westner, 2016). A shadow system is described as an alternative to the existing system formally supported by the organization (Behrens, 2009). Workarounds exist because of system constraints in supporting business processes and user tasks (Spierings & Houghton, 2012), and creative users introduce procedures that better fit their individual needs (Huuskonen & Vakkari, 2013). Alter sketches insights on workarounds concerning types, direct effects, perspectives and organizational challenges (Alter, 2014). Hereupon Röder examined the emerging of workarounds, and the related role of IT systems (Röder, Wiesche, & Schermann, 2014). Workarounds can be structured in type, level (individual, team or organization) or intention (positive or negative) (Röder, Wiesche, Schermann, & Krcmar, 2016). Research of the latter illustrated different user intentions (Bækgaard, Lund-Jensen, Azaria, Permien, & Sawari, 2016). With the purpose of systems improvement in mind, we restrict to positive intentions.

Concerning literature question 1, shadow IT in databases manifests e.g. in informal use of data formats and data conventions or abbreviations and certain codes to shorten long text entries, and thus gaining data entry efficiency. It occurs in any unusual or unwanted data, but when it appears often it might not be recognized as anomaly (Chandola, Banerjee, & Kumar, 2017).

Data classification

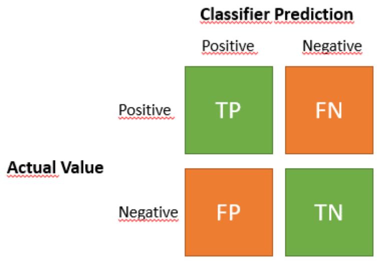
A comparison of open source data mining tools was consulted (Rangra, 2014), as well as a comparative research with both RapidMiner and R (Wowczko, 2015). We decided to use RapidMiner based on this prior research. Starting from most common techniques (Kotu & Deshpande, 2015), and bearing in mind the RapidMiner functionality, the following techniques are part of this research: Naïve Bayes (Patil & Sherekar, 2013), k-NN (Song, Liang, Lu, & Zhao, 2017), Decision Trees (Lim, Loh,

& Shih, 2000) and Support Vector Machines (Wu et al., 2008). These publications were inspirations for set up, execution, evaluation and reporting as well as a reply to the second literature research question. However, the question of how to use classifiers in detecting shadow IT is answered through experiments in chapter 4. That gap is our starting point.

Classification models typically require a target to be set. Examples are flagged by judgement if an occurrence holds shadow IT: *"This is Shadow IT"* or *"This is not Shadow IT"*. As the dataset does not contain such truth labels, they must be added. We use the assignment elaboration from the course and the data generation program to label the examples.

Performance measurement

Data classification here is delimited as binary classification: a sample is classified in only one class without overlap with other classes, it is shadow IT or not (Sokolova & Lapalme, 2009). The confusion matrix shows the performances in terms of true and false predictions. For example, a true positive means that an example is classified as shadow IT, and it actually turns out to be shadow IT.



		Classifier Prediction	
		Positive	Negative
Actual Value	Positive	TP	FN
	Negative	FP	TN

Figure 1: Confusion Matrix

These metrics are derived for classifier evaluation (Powers, 2011; Tharwat, 2018), which is subject of the third literature research question:

Measure	Formula	Evaluation
Accuracy	$\frac{tp + tn}{tp + fn + fp + tn}$	Overall effectiveness of a classifier
Precision	$\frac{tp}{tp + fp}$	Class agreement of the data labels with the positive labels given by the classifier
Recall (Sensitivity)	$\frac{tp}{tp + fn}$	How effectively a classifier identifies positive labels
F-score	$\frac{2 * precision * recall}{precision + recall}$	Combined measure for precision and recall
Specificity	$\frac{tn}{fp + tn}$	Effectiveness of a classifier to identify negative labels
AUC	$\frac{1}{2}\left(\frac{tp}{tp + fn} + \frac{tn}{tn + fp}\right)$	Classifier's ability to avoid false classification

Table 2 Performance metrics

Receiver Operating Characteristics (ROC) are used for organizing classifiers and visualizing their performance (Fawcett, 2006). A more statistical substantiation over classifiers is shown in the Area Under the ROC Curve (AUC) (Provost, Fawcett, & Kohavi, 1998).

Dataset

The dataset for this research is the Betis database from the OU Course Development Practical (Ontwikkelpracticum) ("Development Practical," n.d.). Actually, this training material is based on real life experiences, but was generated programmatically using algorithms and patterns. Before starting our experiment, we were not sure about the data quality, nor were we aware of any shadow IT presence in the data. Course assignment solutions guided us to define our truth labels. The database consists of six tables, and several lists. For our analysis, only the customer (klant) and order (opdracht) tables are of interest, with a focus on the data type fields.

System improvement

In this data classification, detected workarounds inspire system improvements. Suggestions follow for directions, like splitting data fields or adding data entry controls. An answer to this fourth and last literature research question will follow in chapter 4.

Data Mining framework

In order to structure the research the data mining reference model CRISP-DM[1] (Wirth, 1995) is used. Apart from optional iterations, the framework leads to following insights:

1) Business Understanding shines a light on course context, content and objectives,
2) Data Understanding: exploration of the generated database and data quality problems,
3) Data Preparation: selecting features and defining truth labels to mark presence of shadow IT,
4) Modelling of the classifiers with performance metrics and optimizing,
5) Evaluation of the results of the classification models to find the best performance metrics,
6) Deployment: findings of this research can induce system improvement suggestions.

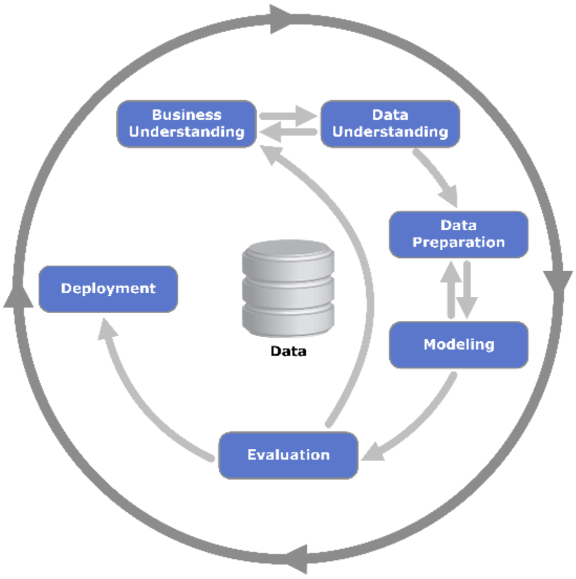


Figure 2 CRISP-DM

2.5. Objective of the follow-up research

We assume that IT systems always contain shadow IT, because over time people are creative in finding shortcuts or workarounds in system use. Alter describes a design system for anticipating and preventing workarounds (Alter, 2015). Being in control over and preventing workarounds is an utopia. Still, the challenge is to find the best classifiers to detect shadow IT. After that, creativity yields system improvement opportunities.

[1] CRISP-DM = CRoss Industry Standard Process for Data Mining

3. Methodology

3.1. Research method

Research insights originate from exploratory classifier development from a qualitative approach. Moreover, pruning, evaluation and statistically comparing the metrics demand a quantitative lens. These mixed methods have been balanced with the feasibility of procedures and techniques (Yin, 2006).

3.2. Classification roadmap

Using CRISP-DM as umbrella, supplemented with other approaches (Kotu & Deshpande, 2015) led to a common roadmap. The following paragraphs explore this roadmap.

Business understanding felt somewhat artificial because the research object was a database that was knowingly coded. Any pattern or classification it contains, was included by design. In other words, checking the software code could have also brought up similar insights. Available course documentation has been used here.

Data Understanding has been done by collecting and cleaning data and preprocessing. Data structures are explored to get familiar with variable types, solve data quality issues like missing values. Outlier detection and visualizing indicated a fictitious entity instance like “2099-01-01” as a transport date for cancelled orders. Certain data entries were limited to an array as data source, and outliers or other natural human entry mistakes are avoided.

Data Preparation lead to further data cleansing by removing some test cases, selecting features, defining predictors and truth labels. Date attributes are converted into a date type, and Boolean attributes were defined as such. Missing values are left unchanged because any solving method can affect potential shadow IT. Therefore, no missing values are imputed, replaced or removed.

In the Modelling phase, the dataset is randomly split into training and testing subsets, with 70 to 30 ratio respectively. Training subsets are used to develop models, including predictors and performance metrics. The testing subset is used for model evaluation. Classifiers are developed using available and suitable algorithms and optimized by parameter pruning.

Evaluation concerns the performance of the model. This is done by measuring, analyzing and visualizing results in ROC and AUC diagrams. Poor findings here could lead to restarting the cycle.

Deployment concerns transposing research results into system improvement suggestions, e.g. data entry controls or data format checks. This will be part of the final discussion, to refer to future research subjects.

In data preprocessing R was used in addition to RapidMiner. The classifiers are built in RapidMiner.

3.3. Sample data, features and truth labels

Sample data is focused mainly on the customer table consisting of 10.432 occurrences and 10 attributes, and the order table consisting of 1.640.059 occurrences and 15 attributes.

To avoid random trial and error, database generator code and course elaborations are used to select potential experiments. Feature selection is based on data structures to be improved (data better assigned to attributes) and data entry checks. For example, in the phone attribute text is used for extra information. In other situations, data is optional where it should be mandatory to avoid missing values. An example is a missing data entry check of the number of packages, weight and the amount, resulting in missing values. Preprocessing uncovers these occurrences and marks them with a truth label.

For data classification, only data or coded text like “unknown” (“onbekend”) where a telephone number is unknown, or a fictitious date “01-01-2099” for cancelled orders are investigated. Therefore typically attributes of type numerical, date and Boolean are in scope, extended with certain coded texts in free text fields. Exploring text fields with text mining techniques is ignored. When data in a note demonstrates shadow IT, it is cleaned as far as possible without text mining techniques.

3.4. Classifiers

Because RapidMiner is our common tool, our classification models are restricted to the RapidMiner assortment. Following classifiers are explored: Naïve Bayes, k-NN, Decision Trees, Logistic Regression and Support Vector Machines. Amongst others, results depend on dataset quality, which also inevitably affects truth labelling. To recognize anomalies, better results are achieved when these occurrences deviate from others. When many occurrences deviate from the norm, that could be interpreted as normal, and not recognized as anomaly (Chandola et al., 2017).

Naïve Bayes

Naïve Bayes (NB) is a probabilistic model that calculates a set of probabilities by counting the frequency and combinations of values in a given dataset. It assumes all attributes to be independent given the value of the class variable (Patil & Sherekar, 2013). In general it achieves high accuracy with high speed, even in larger datasets (Aggarwal, 2015), which is relevant in the order dataset.

K-NN

The K-means clustering technique, k-Nearest Neighbor (k-NN) uses the shortest distances from a centroid to recognize patterns, and is best suited to classify patterns in reduced datasets (Provost & Fawcett, 2013). It is a lazy technique, which requires outlier detection for a better result (Song et al., 2017), and it uses Euclidean distances. The algorithm uses a number of ‘*k*’ iterations with optimizing ‘k’ in the model. This clustering endorses our truth set.

Decision Trees

A decision tree (DT) is a rooted, directed tree, where each node corresponds to a partitioning decision by entropy, and each leaf is mapped to a class label prediction (Aggarwal, 2015). Based on their attributes, instances are segmented into classes with similar values as their target (Provost & Fawcett, 2013).

Logistic Regression

Binary Logistic Regression (LR) is a probabilistic-based classification algorithm, to predict the target class of a variable (Sharda et al., 2018). Logistic Regression applies linear models to class probability estimation (Provost & Fawcett, 2013).

Support Vector Machines

Support Vector Machines (SVM) is also a supervised learning model, where its goal is to find a hyperplane that separates two classes of given samples with a maximal margin (Sossi Alaoui et al., 2017). SVM uses numeric features since it is based on Euclidean distances. SVM classifiers cannot handle missing values, and as missing values might identify shadow IT, SVM is not suitable for our research.

Performance of the classifiers is measured and optimized by parameter pruning and evaluated. Performance metrics on accuracy, recall or sensitivity and specificity are presented in a confusion matrix (Patil & Sherekar, 2013) and visualized with an AUC diagram.

3.5. Validity, reliability and ethical aspects

Different modelling techniques are explored, investigated and compared in order to finally answer the question if data classification is a useful technique for detecting shadow IT. In the area of data classification, basic models are built in several varieties. To broaden our tool assortment, and to reduce the risk that we accidentally get stuck on a model that does not work, we explore models from different algorithms and parameters are varied as far as possible within basic RapidMiner functionality. Working classifiers are optimized and evaluated. To amplify transparency the research process is presented with insight in the modelling process.

Comparing data classification techniques requires a general approach. First, the techniques and tools applied are selected very carefully, regarding a certain prospect and practicability. Second, a structured and standard approach is applied to facilitate reproducibility of the research. Third, the dataset is supposed to have a certain level of data quality, so that it makes sense to experiment with data classification. Data getting starts with acceptance and exploration. A restriction is that only existing techniques are used, no new theory is developed.

This research has been executed with respect for academic ethics code, regarding the research process and the use of data and materials.

4. Results

Conforming CRISP-DM, previous chapters cover the initiation by business understanding. The synthetic course assignment database looks like an order database. User entry errors and shadow IT are simulated programmatically. Data is explored with statistics, visualizations and correlations. Suspicious content that might contain shadow IT are candidate experiments. We chose some of them with different appearances for our data classification.

4.1. Data preparation

This paragraph describes explorative data analysis with R and RapidMiner. This data understanding and preparation lead to a general set up of the experiments and classifiers. The next paragraphs elaborate the experiments towards findings.

Retrieval and statistics

A first course database exploration shows that only the customers and orders tables are relevant to investigate. The tables errormsg, location (plaats) and user (gebruiker) only hold master data, whilst tariff (tarief) contains a calculation instruction. As said, the content of the tables is not originally from work processes. From the source code, we learn how instance data entries are designedly, as the number of packages (pack) is only ranged from one to seven. There are also two test customers and a test order with a street named “a” and “b”, which are removed first. Then attribute types and missing values are explored and statistics of both are tabulated below:

Attribute	Attribute (original)	Description	Type	Min	Max	Missing
NO	NR	Customer number	integer	242	11515	0
NAME	NAAM	Name	polynominal			0
CP	CP	Contact person	polynominal			2172
STREET	STRAAT	Street	polynominal			0
ZIP	PC	Zip code	polynominal			0
HOUSE_NO	HUISNR	House number	polynominal			0
LOCATION	PLAATS	Location	polynominal			0
TEL	TEL	Telephone	polynominal			1974
NOTE	NOTITIE	Notes	polynominal			4101
BLOCK	BLOK	Block code	polynominal	J (62)	N (10370)	0

Table 3 Table customer

The customer table above shows that the customer number is an integer, and all other attributes are text type fields. Values are missing for contact person, telephone and notes. As data absence might be meaningful in this research, we leave them as they are. Further, the block code is a Boolean, with value “J” for blocked customers, and “N” for the others.

Attribute	Attribute (original)	Description	Type	Min	Max	Missing
NO	NR	Order number	integer	1623	1778104	0
D_ORD	DOPDR	Order date	date	15-03-1988	16-07-2010	0
CUST_NO	KLANTNR	Customer number	integer	242	11513	0
PACK	COLLI	Number of packages	integer	1	7	16516
KG	KG	Weight category	real	1	5	49884
STREET	STRAAT	Street	polynominal			0
ZIP	PC	Zip code	polynominal			0

HOUSE_NO	HUISNR	House number	polynominal			0
LOCATION	PLAATS	Location	polynominal			0
D_PLAN	DPLAN	Plan date	date	15-03-1988	14-07-2010	588427
D_TRANS	DTRANS	Transportation date	date	15-03-1988	01-01-2099	2671
NOTE	NOTITIE	Notes	polynominal			1
TICKET	BONBIN	Ticket yes or no	polynominal	J (569194)	N(1070865)	0
EMP	MDW	Employee	polynominal			0
AMOUNT	BEDRAG	Amount	real	0	70	0

Table 4 Table order

In the order table, data types are common: dates are dates, numerical fields are integer or real, and text fields are polynominal. Weight is registered in bins and could be integer, but RapidMiner interprets this as a real. Values are missing for packages, weight, planned date, transportation date and note. From a business perspective an order could be considered as incomplete without packages, weight and planned date, but here they can reveal shadow IT, so they are not removed. The ticket field is a Boolean, filled with "yes" or "no".

Visualizations

Visualizing numerical and date fields gives a quick insight in the date. Figure 3 shows the zip code distribution of customers, with around 1100 until 1200 pointing at locations in the Amsterdam region. This geographical insight is informative, but in itself not an indication for shadow IT.

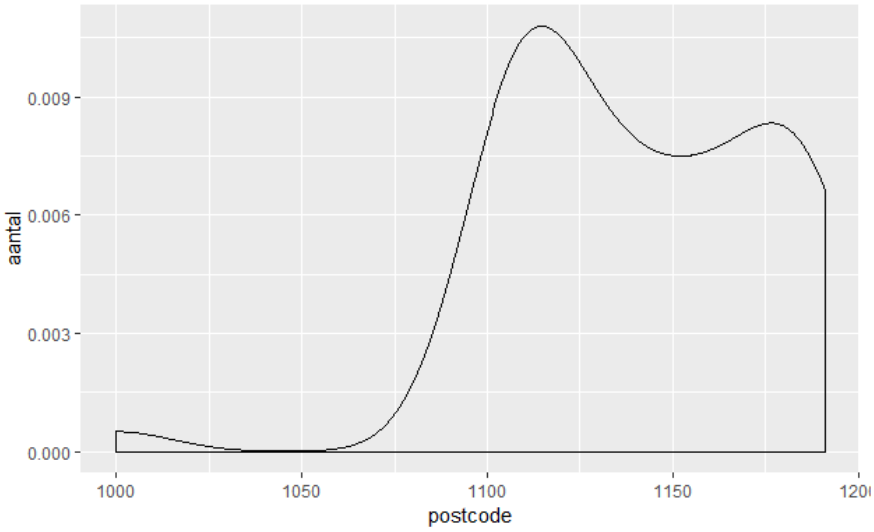


Figure 3 Customer zip code numbers

The histogram of the transport date DTRANS in figure 4 shows a remarkable pattern, because of cancelled orders with the date “01-01-2099”. This occurs in 16258 orders, which is 0.99% of the total order volume. As a suspect shadow IT candidate, this indicates a data imbalance.

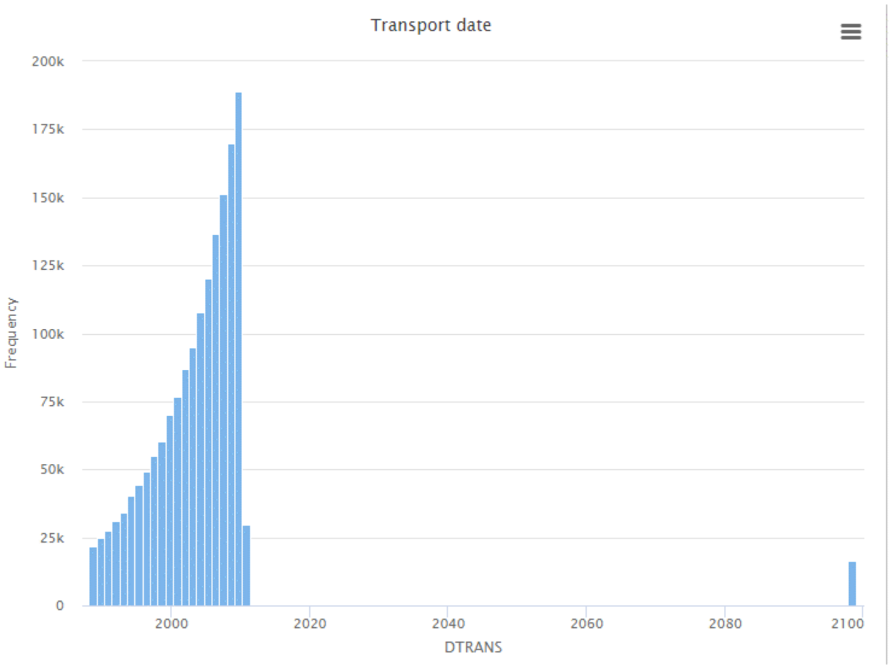
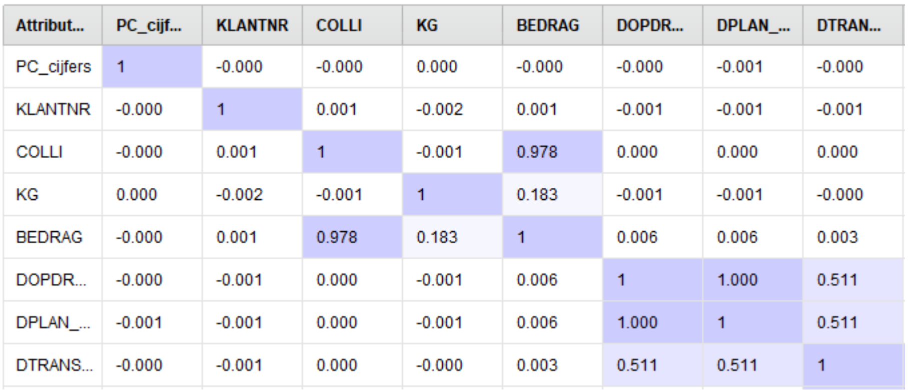


Figure 4 Transport dates histogram

Correlations

Besides visualizations, table 5 shows correlations over the numeric and date fields in the order table:



Attribut...	PC_cijf...	KLANTNR	COLLI	KG	BEDRAG	DOPDR...	DPLAN_...	DTRAN...
PC_cijfers	1	-0.000	-0.000	0.000	-0.000	-0.000	-0.001	-0.000
KLANTNR	-0.000	1	0.001	-0.002	0.001	-0.001	-0.001	-0.001
COLLI	-0.000	0.001	1	-0.001	0.978	0.000	0.000	0.000
KG	0.000	-0.002	-0.001	1	0.183	-0.001	-0.001	-0.000
BEDRAG	-0.000	0.001	0.978	0.183	1	0.006	0.006	0.003
DOPDR...	-0.000	-0.001	0.000	-0.001	0.006	1	1.000	0.511
DPLAN_...	-0.001	-0.001	0.000	-0.001	0.006	1.000	1	0.511
DTRANS...	-0.000	-0.001	0.000	-0.000	0.003	0.511	0.511	1

Table 5 Correlations

There is a strong correlation between PACK and AMOUNT (0.978), which is logical from a business perspective, but not meaningful for our research. Contrary to that expectation, a correlation with KG is only 0.183. Correlation of order date and plan date is evident and logical. However, both dates only correlate with the transport date for 0.511. This already refers to unexpected or uncommon date values. Since there are no striking correlations, there is no specific indication to investigate relationships between attributes. This is one of the criteria in selecting the experiments.

Truth set

Shadow IT is supposed to exist in data fields with unexpected content. It pinpoints an agreed method or appointment, for example, a block code for a customer with a block description in the note field. The note field contains limited different agreed descriptions. Restricting to this code excludes text interpretation and subjectivity. Occurrences that meet the conditions are marked by adding “true” to the Boolean attribute SHADOW_IT, and these are subject for detecting by classifiers.

Connections from data type to free text fields are limited to the block code experiment, where “block” and explanation in “note” are supposed to be connected. The absence of a validity period diminish the practical usefulness. Free text fields need text mining models, which is subject to other circle members’ research (Cate, 2020; Rouwendal, 2020; Spronk, 2020).

Experiments

In data classification the focus is on numeric, Boolean and date attributes. The note field is in the model only in connection with other data fields. Considering our objective to generalize findings, we investigate experiments in different appearances, like text in an expected numerical phone field, a code connected to a description in another field, a fictitious date or missing data. This limited enumeration leads to following experiments:

- Customer telephone: text for unknown or no hearing in an expected numeric phone field (paragraph 4.2)
- Customer block code with block reason in the note field (paragraph 4.3)
- Cancelled orders, which are distinguished by a transportation date of 01-01-2099 (paragraph 4.4)
- Skipped order data entries for crucial data, like COLLI, KG and DPLAN (paragraph 4.5).

Modelling

For each experiment, each classifier mentioned in paragraph 3.4 is developed in RapidMiner and we assessed its performance through 10-fold cross-validation using accuracy, precision, recall, specificity, F-measure and AUC as performance metrics. Most predictor variables in our data are categorical, which obstructs SVM, so we exclude these from our experiments. In each classifier, we use limited number of features; therefore, reduction of complexity by Principal Component Analysis is not necessary. We use the polynominal Boolean attribute SHADOW_IT as target with a value “true” for the marked occurrences and “false” for the others. After labelling the targets, we randomly split the dataset into a training subset of 70% of the data and a test dataset of 30%. Confusion matrices show first results, and if relevant, specific options or peculiarities are described.

Afterwards classifiers’ performances are compared and analyzed. The AUC-score is given in the performance evaluation, instead of AUC diagrams. Findings are generalized as far as relevant.

RapidMiner is used as research platform because of supporting most common classifiers as described in chapter 3.

4.2. Customer telephone

This first experiment uses the customer table. Many customers have text in the telephone attribute that declares an unknown number, or the presence of an alternative. Besides checking coded texts, attributes longer than 15 characters are marked. Those occurrences of TEL are marked with MARK_TEL, which occurred in 1126 examples, or about 11%. This portion of marked occurrences is peculiarly low, and indicates imbalanced data. Despite this, a first model is set up with all attributes and basic parameter settings. With an unsatisfactory result, features are reduced to most relevant, based on human reasoning instead of algorithms.

4.2.1. Naïve Bayes

The lazy model Naïve Bayes developed with a 10-fold cross validation and Laplace correction. Figure 5 shows the basic model layout, which is kept the same over the different algorithms as much as possible. Other models are derived from this by replacing the algorithm in the sub process Cross Validation.

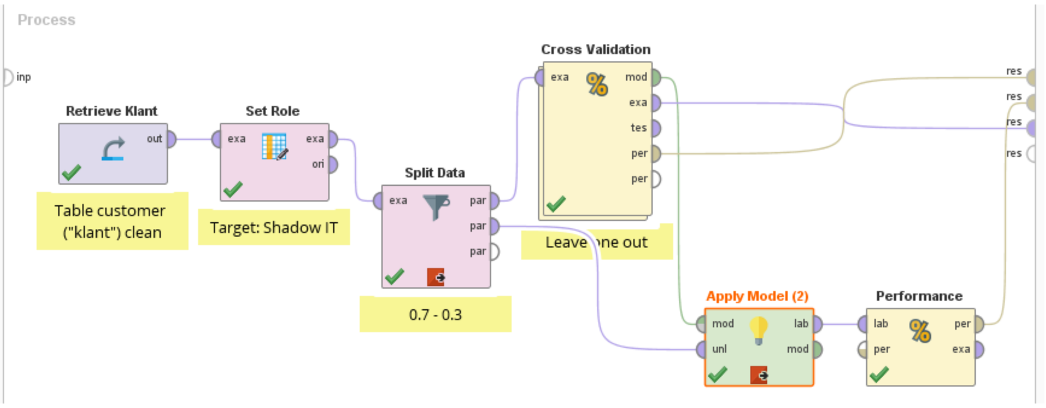
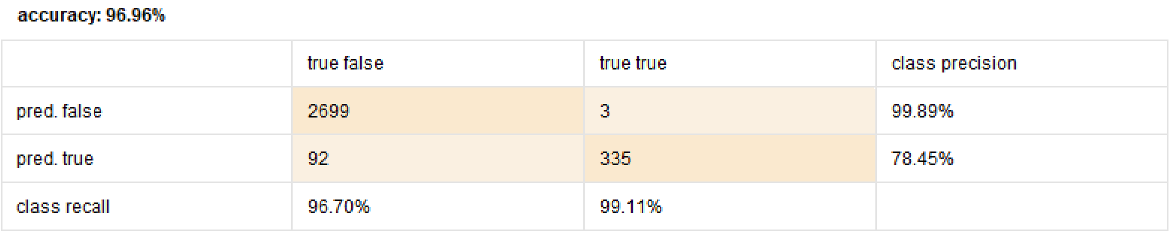


Figure 5 Naive Bayes model telephone

Naïve Bayes with all attributes results in an accuracy of 96.96%, just lower than the from the trained model expected accuracy interval of 97.13 to 98.25. After pruning with leave one out, the best result for this model is this:

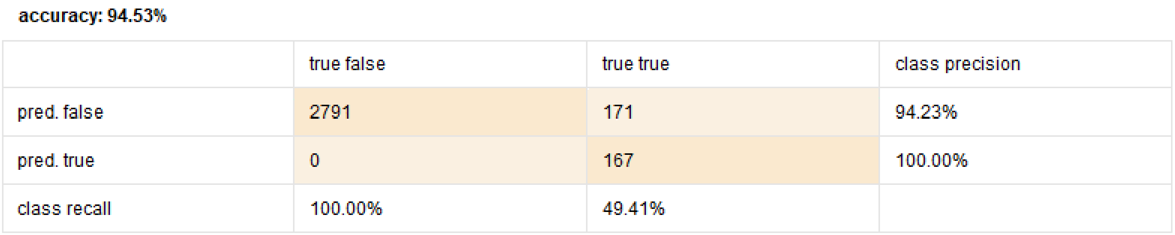


accuracy: 96.96%

	true false	true true	class precision
pred. false	2699	3	99.89%
pred. true	92	335	78.45%
class recall	96.70%	99.11%	

Table 6 Confusion matrix Naive Bayes

Regarding the already low portion of marked shadow IT examples, the objective is to strive for minimalizing the false positives, which is here 92 or 0.9%. Because of the high accuracy, we reduce features to only TEL and SHADOW_IT, and measure the effect. Then accuracy decreases to 94.53%, and false positives are eliminated at the cost of an increased number of false negatives of 171.



accuracy: 94.53%

	true false	true true	class precision
pred. false	2791	171	94.23%
pred. true	0	167	100.00%
class recall	100.00%	49.41%	

Table 7 Confusion matrix Naïve Bayes with only TEL and SHADOW_IT

With the eliminated false positives, a precision and specificity of 100% is achieved at the cost of lower accuracy, F-score and AUC. This feature selection thus performs differently. In paragraph 4.5 we evaluate the metrics and explain which to use.

4.2.2. k-NN

Modelling k-NN requires a small sample for developing the model. Optimizing k triggers many iterations, causing heavier system load. All attributes are used. Once there is basic output, the sample size is enlarged.

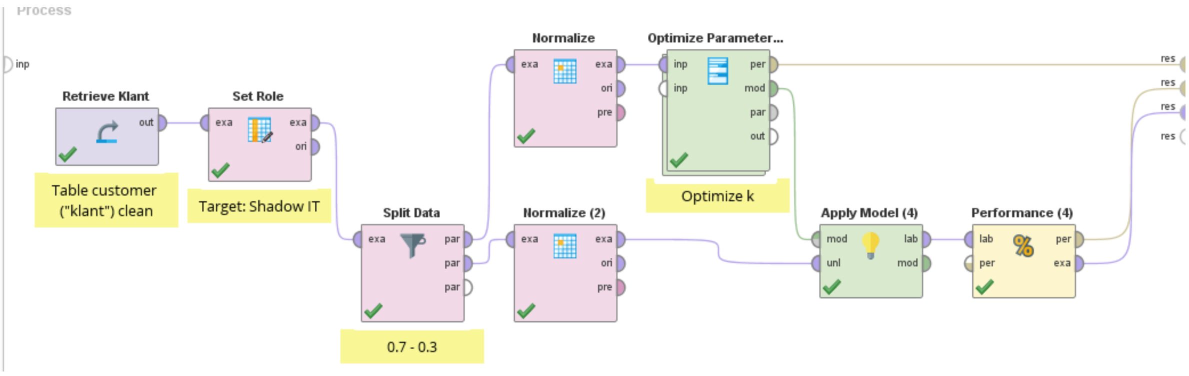
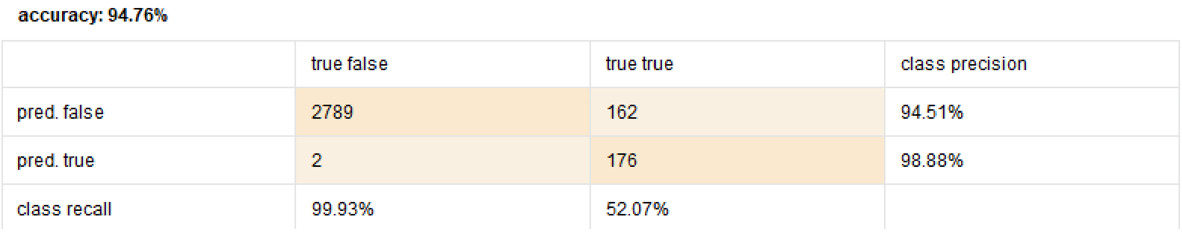


Figure 6 k-NN model telephone in RapidMiner

This accuracy of 94.76% shows a performance that is satisfactory because it fits in the accuracy interval.

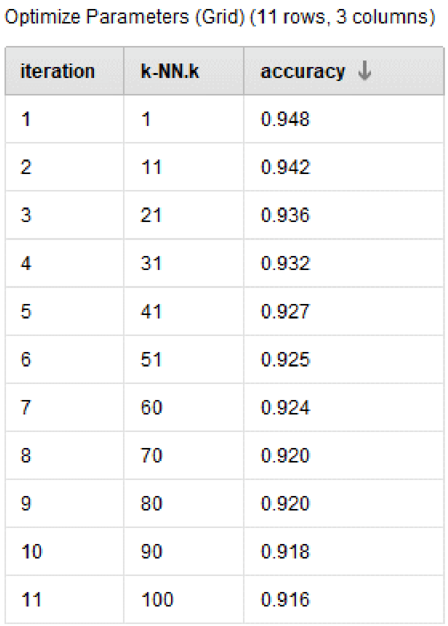


accuracy: 94.76%

	true false	true true	class precision
pred. false	2789	162	94.51%
pred. true	2	176	98.88%
class recall	99.93%	52.07%	

Table 8 Confusion matrix k-NN telephone

The model also optimizes the parameter k, but in the end it shows that ‘k = 1’ delivers best results. The table below shows that the optimal k is 1, so with only one iteration, the accuracy is achieved.



Optimize Parameters (Grid) (11 rows, 3 columns)

iteration	k-NN.k	accuracy ↓
1	1	0.948
2	11	0.942
3	21	0.936
4	31	0.932
5	41	0.927
6	51	0.925
7	60	0.924
8	70	0.920
9	80	0.920
10	90	0.918
11	100	0.916

Table 9 Optimizing k telephone

4.2.3. Decision Trees

As a nonparametric model, Decision Trees is very flexible and therefore often results in overfitting.

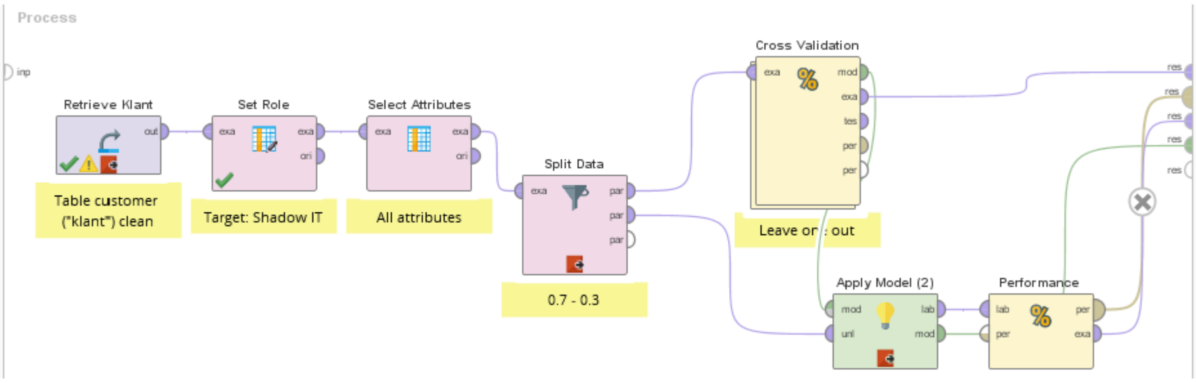
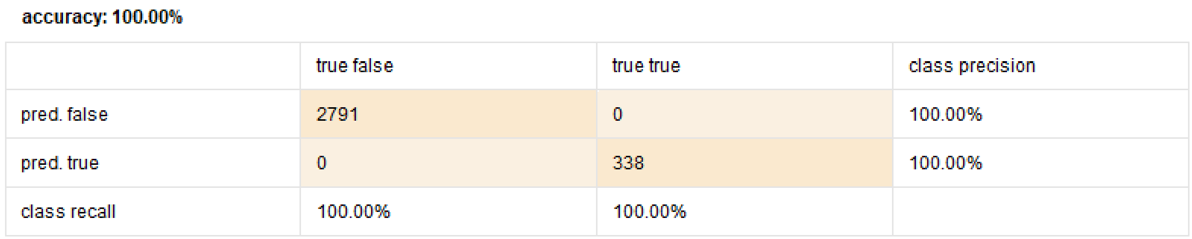


Figure 7 Decision Trees model telephone

The accuracy of this model needs pruning to solve the overfitting issue. Different alternative approaches are investigated, like limited attribute selection, and applying Random Forest for building multiple trees, but the performance does not improve. Even Gradient Boosted Trees with early stopping to avoid overfitting, does not lead to better performance.



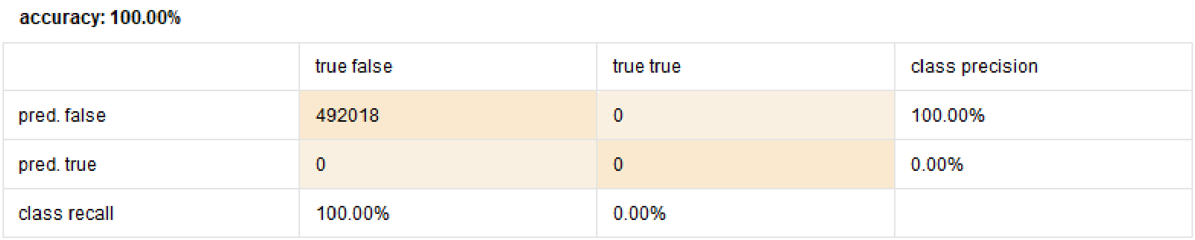
accuracy: 100.00%

	true false	true true	class precision
pred. false	2791	0	100.00%
pred. true	0	338	100.00%
class recall	100.00%	100.00%	

Table 10 Confusion matrix Decision Trees telephone

4.2.4. Logistic Regression

Logistic regression is merely based on probability, and in an imbalanced dataset like this that leads to a 100% accuracy.

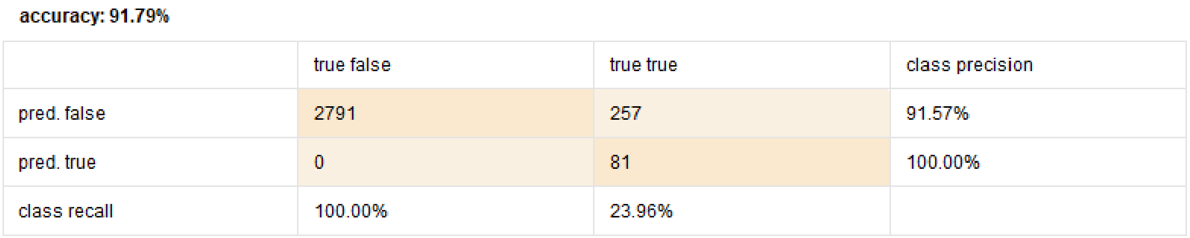


accuracy: 100.00%

	true false	true true	class precision
pred. false	492018	0	100.00%
pred. true	0	0	0.00%
class recall	100.00%	0.00%	

Table 11 Confusion matrix Logistic Regression telephone

As an intervention in this experiment, to see what happens when applying the model on limited features, we only select the attributes TEL and Shadow_ IT. And this model appears to fit with an accuracy of 91.79 into the interval of 90.92 to 91.88:



accuracy: 91.79%

	true false	true true	class precision
pred. false	2791	257	91.57%
pred. true	0	81	100.00%
class recall	100.00%	23.96%	

Table 12 Confusion matrix Logistic Regression with limited features

Here the false positives are zero, which means that there are no occurrences predicted as shadow IT that turned out not to be.

4.2.5. Performances

This first experiment performs with high scores on almost all metrics. A 100% accuracy means that the classifier can perfectly distinguish every shadow IT occurrence. Here the metrics are tabulated:

	Accuracy	Precision	Recall	Specificity	F-score	AUC
Naïve Bayes	96.96	78.45	99.11	96.70	87.58	0.998
Naïve Bayes, with reduced features	94.53	100.00	49.41	100.00	66.14	0.809
k-NN	94.76	98.88	52.07	99.93	68.22	0.500
Decision Trees	100.00	100.00	100.00	100.00	76.83	0.500
Logistic Regression	100.00	100.00	100.00	100.00	100.00	1.000
Logistic Regression - limited features	91.79	100.00	23.96	100.00	38.66	0.675
Random Forest	100.00	100.00	100.00	100.00	100.00	0.500
Gradient Boosted Tree	100.00	100.00	100.00	100.00	100.00	0.500

Table 13 Performance Customer telephone

Usually accuracy is very important to measure the goodness of a model. However, as seen in the table above, all classifiers score high on distinguishing the “TRUE” labels, and there is none or a few false positives, so this does not provide us with relevant information.

What strikes is that these phone data show imbalance data, regarding the low portion of 11%. Pozzolo explains why with unbalanced data accuracy, precision and recall are not suitable. Better metrics are specificity, F-score and AUC (Pozzolo, Caelen, Johnson, & Bontempi, 2015). The specificity shows the correctly classified true negatives or the true shadow IT, which is high in all models. The F-score combines the precision and the recall, thus giving a weighted measure. The AUC represents the degree of separability measure. It shows to what extent the model is capable of distinguishing between classes. The higher the AUC, the better the model can distinguish customer occurrences with or without shadow IT. Comparing these metrics over the classifiers, scores are still high, except for a lower AUC of k-NN and Decision Trees.

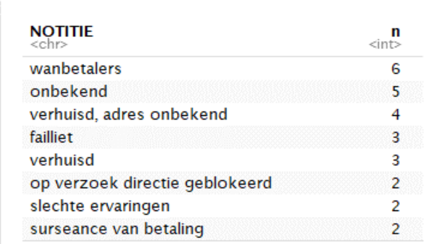
These extremely high scores can mean two things: either the classifiers are all performing very well and able to distinguish every shadow IT occurrence, or all models run into overfitting and are not able to detect the targets at all.

In following experiments, metrics are evaluated on usability.

As far as it is allowed to draw conclusions based on these performance scores, we see that model performances do not deviate substantially. Probability models Decision Trees, Random Forest, Gradient Boosted Trees and Logistic Regression lead to overfitting, and do not provide us with useful performance information. From this, we conclude that this phone field experiment, with unexpected texts in a phone number field, shows best performance with the Naïve Bayes algorithm.

4.3. Customer block

This second experiment starts from blocked customers with block code “yes”, and searches for a block reason in the note. From the 10430 customers 62 or 0.6% are blocked, and at 27 of them or 43.5%, the reason is registered. This is considered to be shadow IT because of its incompleteness, and informal and error prone registration. Only limited reasons occur, so this is considered a coded text, and we search for it without text mining. Customers with a block code “yes” and a block reason in the note are labelled with SHADOW_IT.



NOTITIE <chr>	n <int>
wanbetalers	6
onbekend	5
verhuisd, adres onbekend	4
failliet	3
verhuisd	3
op verzoek directie geblokeerd	2
slechte ervaringen	2
surseance van betaling	2

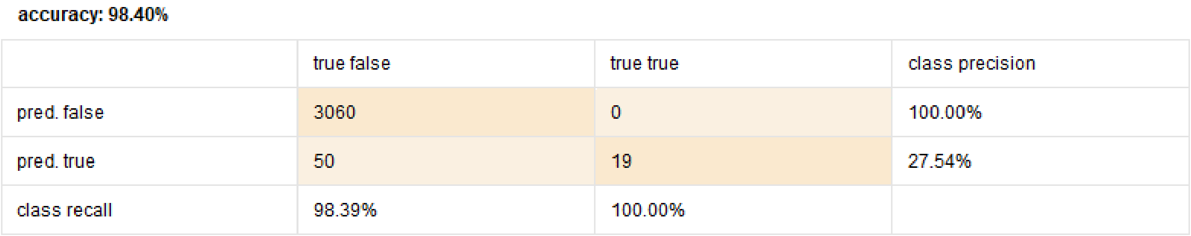
Figure 8 Block reasons

The difference with previous experiment is the connection of data over two attributes. Marking them results in 62 labels, which consequently leads to an imbalance issue again. Therefore, we expect similar performance as in previous experiment.

The models are not repeated here, as they are similar to those in the previous experiment. Explanations are limited to differences with previous experiment.

4.3.1. Naïve Bayes

This model runs with Laplace smoothing. With a small confidence interval of 96.88-98%, that might be caused by using 10-folds instead of ‘leave one out’, we accept the model fit with an accuracy of 98.40%. This model yields 50 false positives, which is acceptable with 1.6%, but no false negatives.



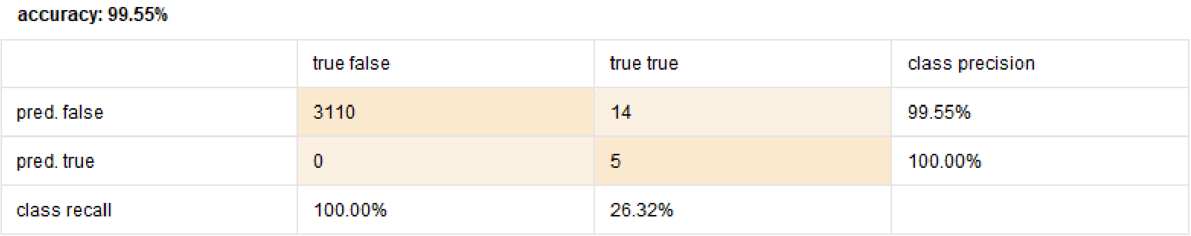
accuracy: 98.40%

	true false	true true	class precision
pred. false	3060	0	100.00%
pred. true	50	19	27.54%
class recall	98.39%	100.00%	

Table 14 Confusion matrix Naïve Bayes blocked customers

4.3.2. k-NN

With many iterations this model has longer runtimes, so the model runs 10-fold and we train it with k=5. There were no false positives. Accuracy of 99.55 is just lower than the confidence interval of 99.60-99.83. But as we did not use leave one out, this is an optimistic interval, and we can accept this result.

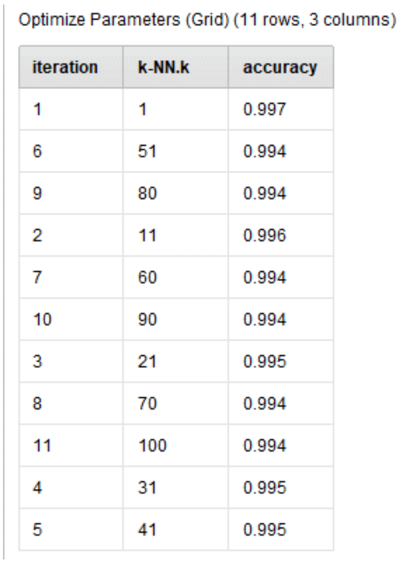


accuracy: 99.55%

	true false	true true	class precision
pred. false	3110	14	99.55%
pred. true	0	5	100.00%
class recall	100.00%	26.32%	

Table 15 Confusion matrix k-NN blocked customers

Optimizing k in the test model comes to 1, with an accuracy of 0.997, which is shown below:



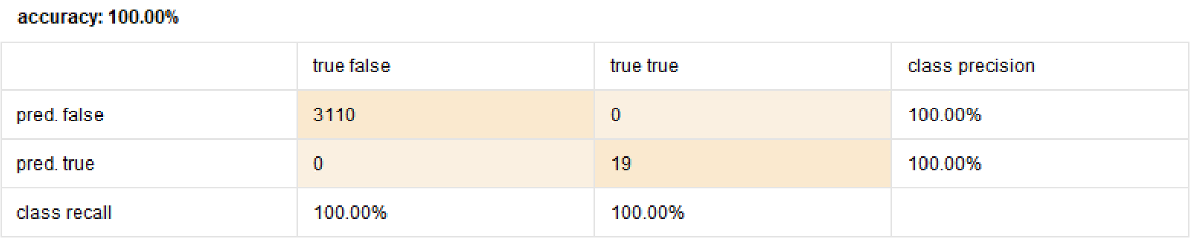
Optimize Parameters (Grid) (11 rows, 3 columns)

iteration	k-NN.k	accuracy
1	1	0.997
6	51	0.994
9	80	0.994
2	11	0.996
7	60	0.994
10	90	0.994
3	21	0.995
8	70	0.994
11	100	0.994
4	31	0.995
5	41	0.995

Table 16 Optimization k for blocked customers

4.3.3. Decision Trees

This model runs with one leave out, and maximal depth of 10, but it generates only one node. As we already could expect, based on previous experiments, this probability based model here also results in overfitting:



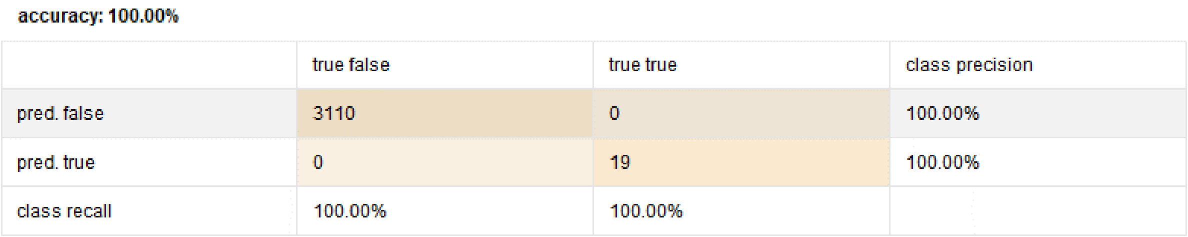
accuracy: 100.00%

	true false	true true	class precision
pred. false	3110	0	100.00%
pred. true	0	19	100.00%
class recall	100.00%	100.00%	

Table 17 Confusion matrix Decision Trees blocked customers

4.3.4. Logistic Regression

Analogue to the Decision Trees model, also this probability model shows overfitting, with a 100% perfect prediction.



accuracy: 100.00%

	true false	true true	class precision
pred. false	3110	0	100.00%
pred. true	0	19	100.00%
class recall	100.00%	100.00%	

Table 18 Confusion matrix Logistic Regression blocked customers

4.3.5. Performances

This experiment takes into account a connection between the Boolean block code, and a description in a free text field. The probabilistic classifiers manifest perfect distinguishing capability, as tabulated here:

	Accuracy	Precision	Recall	Specificity	F-score	AUC
Naïve Bayes	98.40	27.54	100.00	98.39	43.18	1.000
k-NN	99.55	100.00	26.32	100.00	41.67	0.500
Decision Trees	100.00	100.00	100.00	100.00	100.00	0.500
Logistic Regression	100.00	100.00	100.00	100.00	100.00	1.000

Table 19 Block reasons in all customers

Though Naïve Bayes seems to perform satisfactory, but the low precision disqualifies the model. K-NN shows a low recall and AUC, so has a low distinctive capability. Both Decision Trees and Logistic Regression are not usable because of overfitting.

In an alternative to avoid the imbalance situation, we filter out the 62 blocked customers, and classify the 27 customers with a block code in the note field. Here, we start with almost 44% of the examples marked as shadow IT. Doing so, we explore these classifiers in a balanced data situation, with the disadvantage of a small dataset. If we disqualify Naïve Bayes because of its over-performing, than Logistic Regression is best in this case, as tabulated below:

	Accuracy	Precision	Recall	Specificity	F-score	AUC
Naïve Bayes	100.00	100.00	100.00	100.00	100.00	1.000
k-NN	61.11	100.00	12.50	100.00	22.22	0.688
Decision Trees	55.56	50.00	62.50	50.00	55.56	0.656
Logistic Regression	94.44	100.00	87.50	100.00	93.33	1.000

Table 20 Block reasons at blocked customers

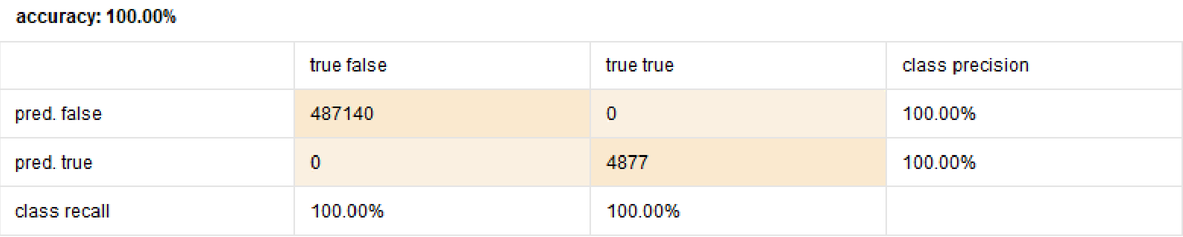
As expected in a smaller dataset, metrics diverge more, but are less reliable. These results are excluded from the final recap.

4.4. Cancelled orders

An agreed indication for cancelled orders is the date “01-01-2099” in the transport date field DTRANS. For finding target occurrences for the date the day number 47116 is used, converted by “epoch” day numbering. The shadow IT occurs in 16258 examples, which is 0.99%. We chose this example because of its origin with a factious data entry, which makes it different from previous experiments. However, after labelling similar situation exists regarding classification. We run our classifiers as before and expect similar results.

4.4.1. Naïve Bayes

This model runs 10-fold with Laplace correction. The confusion matrix below presents high scores again.



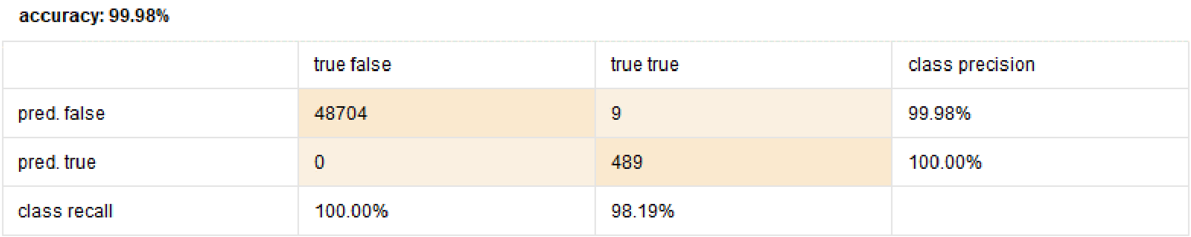
accuracy: 100.00%

	true false	true true	class precision
pred. false	487140	0	100.00%
pred. true	0	4877	100.00%
class recall	100.00%	100.00%	

Table 21 Confusion matrix Naïve Bayes cancelled orders

4.4.2. k-NN

With numerical date attributes and label, k-NN with optimizing k has an extremely long runtime. Selected attributes were limited to DOPDR, DTRANS and examples sampled to 30%. With these restrictions and a long runtime, the accuracy performance scores high again. This restrains us from running the model without sampling.



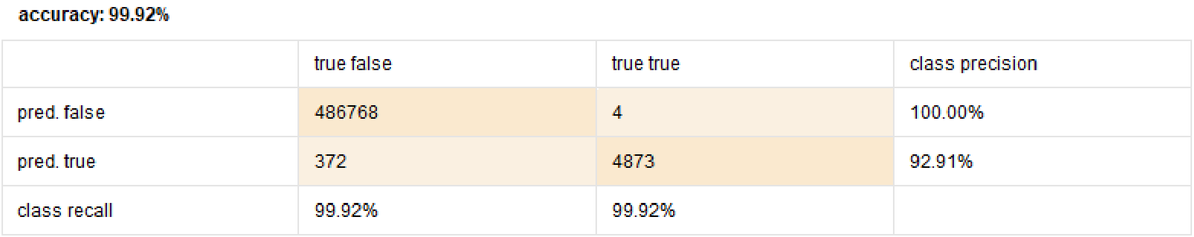
accuracy: 99.98%

	true false	true true	class precision
pred. false	48704	9	99.98%
pred. true	0	489	100.00%
class recall	100.00%	98.19%	

Table 22 Confusion matrix k-NN cancelled orders

4.4.3. Decision Trees

Standard Decision Trees performs 100% with all options we tried. Therefore, we step away from this overfitting by selecting only the order dates DOPDR, DPLAN and DTRANS and using Gradiant Boosted Trees with early stopping the model training. With 5-folds and reduced trees to 10, the performance comes to 99.92%.



accuracy: 99.92%

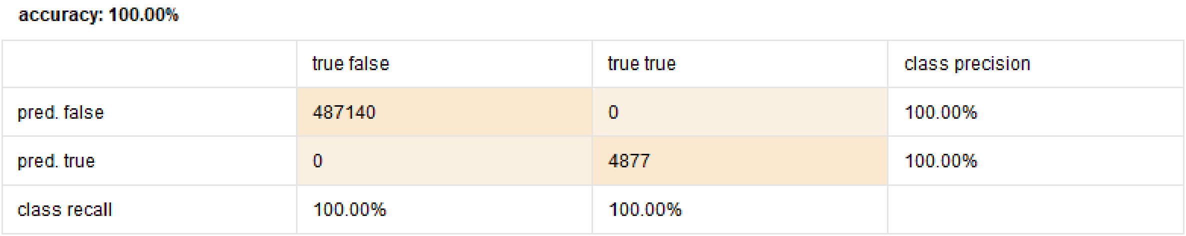
	true false	true true	class precision
pred. false	486768	4	100.00%
pred. true	372	4873	92.91%
class recall	99.92%	99.92%	

Table 23 Confusion matrix Gradiant Boosted Trees cancelled orders

The 372 false positives is 0.00076 of the test data, which is negligible, but compared to the other models is much more. With an AUC of 1, this outperforms the other models.

4.4.4. Logistic Regression

The Logistic Regression model runs with date fields in days, and results in 100% scores.



accuracy: 100.00%

	true false	true true	class precision
pred. false	487140	0	100.00%
pred. true	0	4877	100.00%
class recall	100.00%	100.00%	

Table 24 Confusion matrix Logistic Regression cancelled orders

4.4.5. Performances

Normally we strive for high performances. However, when so many metrics have a value of 100%, we must reconsider the reliability and the message. Here again, high numbers scores embarrass a thorough and weighted evaluation.

As a specific feature of Decision Trees the Gradiant Boosted Trees is added, and despite 372 false positives, this classifier deserves a place in the row.

	Accuracy	Precision	Recall	Specificity	F-score	AUC
Naïve Bayes	100.00	100.00	100.00	100.00	100.00	1.000
k-NN	99.98	100.00	98.19	100.00	99.09	99.98
Decision Trees	100.00	100.00	100.00	100.00	76.83	0.500
Gradiant Boosted Trees	99.92	92.91	99.92	99.92	96.29	1.000
Logistic Regression	100.00	100.00	100.00	100.00	100.00	1.000

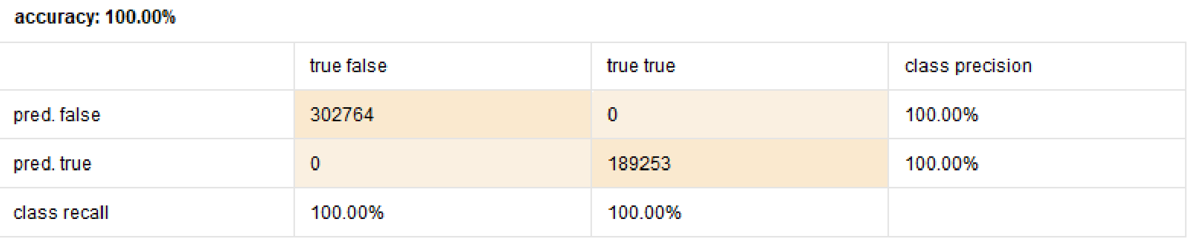
Table 25 Performance evaluation of cancelled orders

4.5. Missing order data

The order table shows unexpected missing values for COLLI, KG and DPLAN. Shipping an order without knowing how many packages or the weight or without planning the transport should be impossible. These occurrences are detected and marked, which is at 630843 orders, or 38.5%. So here, with a portion of almost 40% detected shadow IT, there is no imbalance issue. After labelling, the missing values for COLLI, KG and DPLAN_day are replaced by zero as fictitious number.

4.5.1. Naïve Bayes

With all features this classifier again performs with 100% accuracy for both learning and testing:



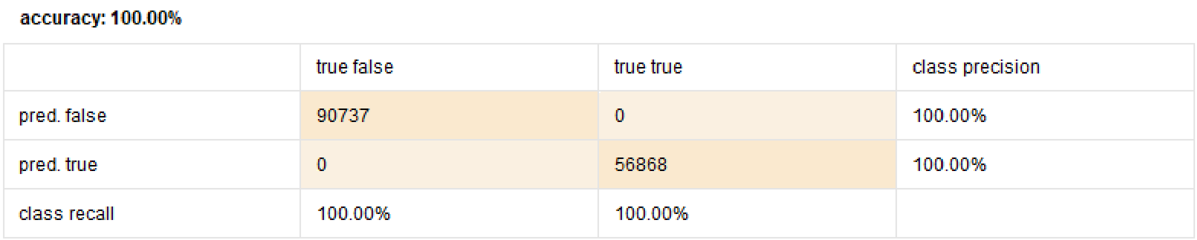
accuracy: 100.00%

	true false	true true	class precision
pred. false	302764	0	100.00%
pred. true	0	189253	100.00%
class recall	100.00%	100.00%	

Table 26 Confusion matrix Naïve Bayes missing order data

4.5.2. k-NN

Due to its nature, as it uses Euclidean distances, here only the numerical features COLLI, KG, DPLAN_day and the label SHADOW_IT are used. With 10-folds, this classifier takes a long runtime and therefore we first sample a 30%, so that its original distribution remains. Then this classifier perfectly classifies as well as other models before. Regarding the long runtimes of this model, we did not run it with a full dataset.

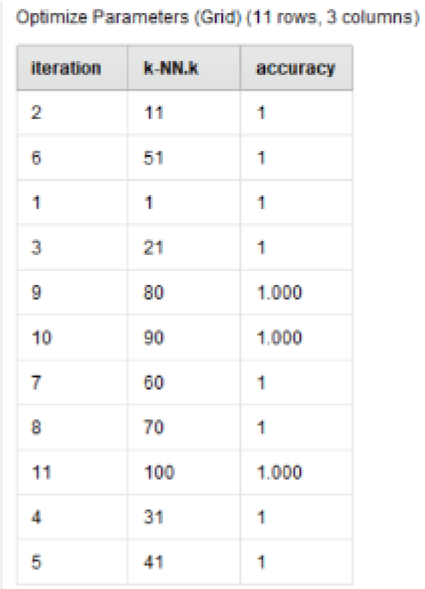


accuracy: 100.00%

	true false	true true	class precision
pred. false	90737	0	100.00%
pred. true	0	56868	100.00%
class recall	100.00%	100.00%	

Table 27 Confusion matrix k-NN missing order data

Optimizing k does not bring up new information, with an equal accuracy for all iterations:



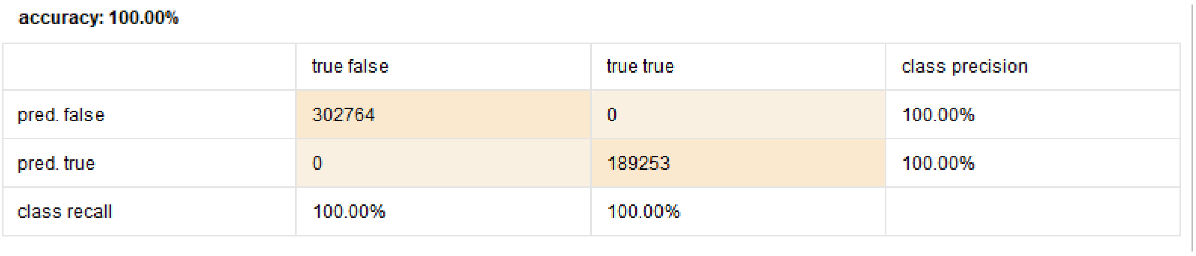
Optimize Parameters (Grid) (11 rows, 3 columns)

iteration	k-NN.k	accuracy
2	11	1
6	51	1
1	1	1
3	21	1
9	80	1.000
10	90	1.000
7	60	1
8	70	1
11	100	1.000
4	31	1
5	41	1

Table 28 Optimizing k for missing order data

4.5.3. Decision Trees

In this experiment, this classifier runs in a basic set-up, again with a 100% performance:



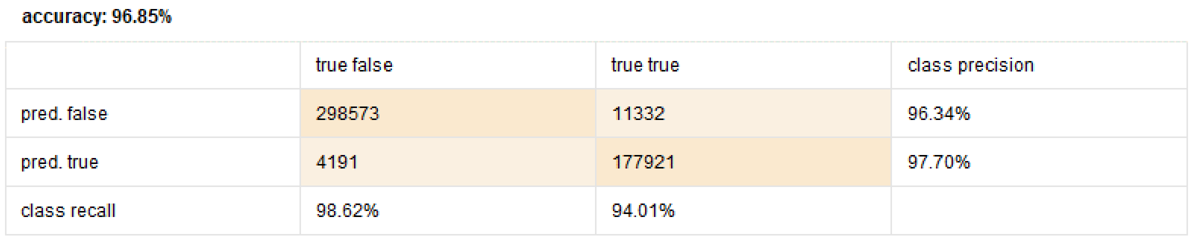
accuracy: 100.00%

	true false	true true	class precision
pred. false	302764	0	100.00%
pred. true	0	189253	100.00%
class recall	100.00%	100.00%	

Table 29 Confusion matrix Decision Trees missing order data

4.5.4. Logistic Regression

This model only uses the numerical features COLLI, KG, DPLAN and the label SHADOW_IT. Here classification shows another confusion matrix. Although performance is not as perfect as for many other models, it gives more confidence and looks more informative:



accuracy: 96.85%

	true false	true true	class precision
pred. false	298573	11332	96.34%
pred. true	4191	177921	97.70%
class recall	98.62%	94.01%	

Table 30 Confusion matrix Logistic Regression missing order data

4.5.5. Performances

This experiment shows that even with balanced data classification models are well capable to detect the marked shadow IT. In this case with common balanced data a first performance metric is the accuracy, whereas the AUC indicates the classification capability as well.

	Accuracy	Precision	Recall	Specificity	F-score	AUC
Naïve Bayes	100.00	100.00	100.00	100.00	100.00	1.000
k-NN	100.00	100.00	100.00	100.00	100.00	1.000
Decision Trees	100.00	100.00	100.00	100.00	100.00	0.500
Logistic Regression	96.85	97.70	94.01	98.62	95.82	0.987

Table 31 Performance evaluation of missing data

The missing data in this experiment are undesirable, because without these it should not be possible to process and transport an order. In practice, this situation could only be caused by procedural errors, which are contrary to common work methods.

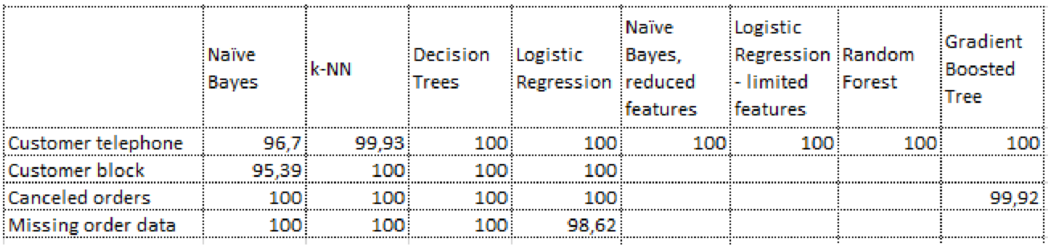
False positives only occur at Logistic Regression. Though all metrics are here just below 100%, they may be more valuable than those of the other models, because they all stay at the 100% level.

4.6. Performance evaluation

Primarily performance is measured with accuracy. RQ3 focuses on how to compare classifiers in general. The difficulty in our data is merely to define and label occurrences with shadow IT. With low label volumes, our experiments deal with unbalanced data. As mentioned earlier, with imbalanced data, better performance measures are specificity, F-score and AUC (Pozzolo et al., 2015).

To evaluate the accuracy, the small blocked customer dataset show high accuracy for Naïve Bayes and Logistic Regression with a high AUC, against much lower performances for k-NN and Decision Trees. In addition, the missing value experiment shows 100% scores in all models. Tables 20 and 31 show these high scores.

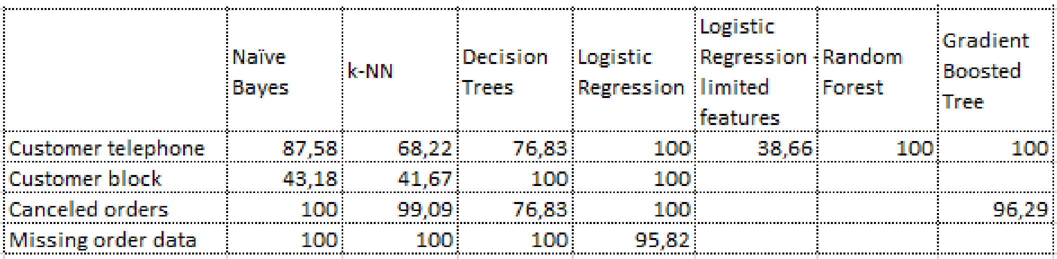
Specificity is high in all experiments, which means that are models are able to distinguish true negatives or the true shadow IT correctly. With quality levels so close, it makes it difficult to point out which model is best. Every classifier at least once outperforms the others. Specificity over the models is summarized below:



	Naïve Bayes	k-NN	Decision Trees	Logistic Regression	Naïve Bayes, reduced features	Logistic Regression - limited features	Random Forest	Gradient Boosted Tree
Customer telephone	96,7	99,93	100	100	100	100	100	100
Customer block	95,39	100	100	100				
Canceled orders	100	100	100	100				99,92
Missing order data	100	100	100	98,62				

Table 32 Specificity recap

In an analogous way, we collect the F-scores over the models. As combination of precision and recall, this is an indication of how well positive labels are distinguished. It shows the extent to which examples with shadow IT are detected by their labels.

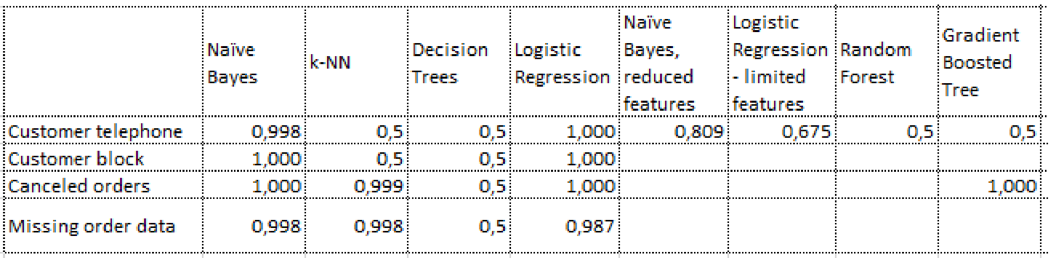


	Naïve Bayes	k-NN	Decision Trees	Logistic Regression	Logistic Regression limited features	Random Forest	Gradient Boosted Tree
Customer telephone	87,58	68,22	76,83	100	38,66	100	100
Customer block	43,18	41,67	100	100			
Canceled orders	100	99,09	76,83	100			96,29
Missing order data	100	100	100	95,82			

Table 33 F-scores recap

Best results are at the order table where missing data are used. An argument for this can be that this experiment uses balanced data in contrast to the other experiments.

Finally, the AUC over the classifiers is compared. The higher the AUC, the better the model's ability to avoid false classification. That is to avoid pointing at examples to be shadow IT, that in the end turns out not to be shadow IT. Following table presents that Naïve Bayes and Logistic Regression perform to 100% or slightly less over all models.



	Naïve Bayes	k-NN	Decision Trees	Logistic Regression	Naïve Bayes, reduced features	Logistic Regression - limited features	Random Forest	Gradient Boosted Tree
Customer telephone	0,998	0,5	0,5	1,000	0,809	0,675	0,5	0,5
Customer block	1,000	0,5	0,5	1,000				
Canceled orders	1,000	0,999	0,5	1,000				1,000
Missing order data	0,998	0,998	0,5	0,987				

Table 34 AUC recap

In general, the classifiers in supervised learning with labelled examples perform perfectly or almost perfectly.

5. Discussion, conclusions and recommendations

Below findings from the experiments are summarized into answers to the research questions and overall conclusions. Afterwards, recommendations for practice are given as well as an outlook to future research.

5.1. Discussion

The counterfeit database that was subject of this research, did not abridge data complexity. In itself this factitious data was implicitly restricted to the developers' creativity. Real data would have contained more natural text variations, because of human behaviour. Still, this synthetic dataset contains patterns and distributions simulating real data. Data quality levels were unknown to us beforehand, as well as data pollution or data entry structures, but every deficiency was deliberately coded.

After exploring the data, for each experiment a truth set is built by labelling examples that meet the conditions, using the database generation code and course documentation as sources. As stated in chapter 3, to our assumption, binary classification requires a supervised approach. In case generator code is not available, and presence of target examples or their sources are not known, in comparable data volumes, a semi supervised technique can be used for labelling. Then labelled and unlabelled data are combined in marking targets, which reduce labelling processes but at the same time might complicate it (Zhu, 2007).

Developing models requires knowledge of classifiers and RapidMiner operators. Moreover, problem solving skills are essential, because you cannot foresee the issues you have to resolve. Selecting potential good algorithms needs more in-depth attention at the start. Not all available classifiers turned out to be usable, for instance SVM appears not to be a good candidate for several reasons. Another example, where according to Powers accuracy is in general a first performance metric (Powers, 2011), here other metrics are required (Saia, 2019).

The first experiments contain less than 1% up to 10% marked examples. This implies imbalanced data issues, which demand for reconsidering used classification methods. From the customer table experiments overfitting seems to be caused by imbalanced data. However, in the latter missing order data experiment, overfitting remains, so that contradicts this causation assumption. In retrospect our search process shows similarities with a fraud detection case (Ngai, Hu, Wong, Chen, & Sun, 2011), where the objective is to detect a small minority (Saia, 2019). Generally speaking, in almost all experiments we elaborate from this dataset, imbalanced data issues occur. Common interventions do not solve the problem. First we mark examples with shadow IT, and with classifiers we try to find those. It is worth it to consider if classifiers really are doing this job all that well, or are we deducting a kind of tautology. This approach deserves a critical re-evaluation and further research. This deficiency interferes a quality response to our research questions.

Another emerging obstacle in our experiments is overfitting models, because they result in 100% performance metrics. Already the learning models show this, and the test performances confirm it. Obviously, the classifiers are able to distinguish labelled examples perfectly. K-fold pruning, early stopping the learning process, or leave one out do not solve this issue. Nor do the models themselves with the RapidMiner log messages provide clear solutions.

When comparing different classifiers, all circumstances and parameters must remain the same as far as possible. For instance, while splitting the data, the same seed should be used, where in our models data with the same ratios, data is randomly split. Other options meant are the sample volume, and optimizing the same parameters. In our experiments, we kept as many attributes as possible within classifier constraints, such as k-NN demanding for numerical features.

The investigated dataset contains static data without a time aspect. Time-dependent validity would enrich data and make it dynamic. Examples such as customer block codes, or an order to be offered a second time, are more informative with a validity period. Workarounds typically suggest a dynamic process flow, as processes create data, which trigger next process steps. With an order number as a case ID, activities and timestamp, process mining as a quantitative technique helps to discover process flows and detect workarounds through systems, better than only data based techniques (Outmazgin & Soffer, 2016).

5.2. Conclusions

This paragraph shows answers to the questions our research was aiming at. The first question, to find out what data or data structures are indicators for shadow IT in a database, explores numeric, Boolean and date data type fields. Free text fields containing data as shadow IT are out of scope. Our dataset has been generated by code, so every potential shadow IT occurrence is also coded. That insight determines the truth set and target examples that we are looking for. The frequencies are, however, still unknown until data preparation. With an unknown content or origin of a dataset, text mining in exploratory data analysis helps to get familiar with the data. In our database, we investigated shadow IT in several appearances and types. Examples are text in a phone number field, which might have been expected numeric type field (4.2), a Boolean with additional description in a free text field (4.3), an order type by use of a fictitious date with consequently artificial lead times (4.4) and missing values for mandatory fields (4.5) to bypass formal procedures. These instances are examples, but not necessarily an exhaustive list. Shadow IT can be in any type of data as long as there is no data entry check to prevent fictitious or nonexistent data or enforce mandatory input.

The second research question is concerned with the prediction and detection of shadow IT using classification algorithms. After labelling shadow IT occurrences, common classifiers selected from RapidMiner are Naïve Bayes, k-NN and the probabilistic Decision Trees, Logistic Regression. SVM only work with numeric attributes, and cannot handle missing values. Therefore these are not suitable. Modelling turns out to be a complicated process, where obstacles are encountered like unbalanced data, overfitting models, runtime issues and unidentified phenomena. Our classifiers do not predict shadow IT for new, unlabelled examples. Once an example is labelled, classifiers recognize it.

Comparing the classifiers performances is the third objective. This originates from the expectation that common metrics would be a good quality indicator. We measure the capability of distinguishing labelled data. Developing classifiers proves not to be a straightforward job with default metrics. Performance interpretation combined with early research confirms that in unbalanced data accuracy, precision and recall are not appropriate for statistical reasons (Pozzolo et al., 2015). Specificity pinpoints the true negatives, which are the shadow IT occurrences. The F-score shows the weighted average of the precision and recall metrics (Saia, 2019). Generally, the AUC scores the classification capability of the model.

The last research question was to find shadow IT and to convert this into system improvement suggestions. Some findings are evident, even without classification models. For example, data fields with essential order information like weight and volumes should be mandatory. At least data entry checks on mandatory data would prevent missing values. In general, a data entry with a required description in a note can be improved by a meaningful data code field instead of a descriptive text.

An example is a meaningful customer block code with a validity period, to signal a timed block with reason in one attribute. The use of a fictional date for a special order type causes artificial lead times and bias. A big and simple improvement is in adding a code for special order types.

5.3. Recommendations for practice

Finding data in this synthetic dataset is characterised by generated patterns and minority of the target occurrences. That makes it comparable to a fraud finding case. When applying our findings in practice, it is recommended to incorporate text mining techniques in the Exploratory Data Analysis. With only data classification finding examples subject to the labelling there is no transparency on what is searched for. With additional text mining also the use of data descriptions in free text fields can be detected as target items, for instance planning and delivery information in the order note. Further is it worthwhile to investigate connections between attributes.

It also can bring op potential connections between attributes, so it is worthwhile to combine results of text mining on free text fields with the data labelling from this research. This might bridge the gap to the research of free text mining classification (Cate, 2020).

This research teaches us that building a truth set is a condition for classifying labelled occurrences. The labelling seems to be the most important job in this research. The classifiers themselves do not really add new information. For instance, recognizing a mail address in a note field is enough motivation to add a dedicated field for that information.

5.4. Recommendations for further research

Looking at restrictions or pitfalls from our research it is worthwhile to find out if models will yield more with real world databases. Before developing classifiers, strategic approach consideration is essential, to verify our assumption that classification requires supervised models. We used RapidMiner to label target occurrences, but perhaps a semi-supervised approach can be a good alternative, where labelled and unlabelled data are combined in a learning model. RapidMiner only offers that option in the SVM algorithm, which was not suitable for our experiments.

Research on how to handle imbalanced data has been done before, but there is no one best way to solve all issues. Findings are affected by the data quality and data balance, so further research on specific characteristics and interventions can be interesting in improving classification performances. Our results suggest resemblance with fraud detection cases, that use classification for minority reports. Further investigation can learn if balancing data in advance can increase the chance of finding the targets. Though the metrics we used already scored very high. Does that really mean that the classifiers perform that well, or are all models overfitted?

For adjusting imbalanced data undersampling and oversampling methods (He & Garcia, 2009) are potential interventions. In undersampling the abundant part of unlabelled occurrences is reduced, for instance by sampling them. With oversampling the amount of shadow IT labelled examples is increased, by repetition, bootstrapping or e.g. with the SMOTE[2] technique (Hall, Kegelmeyer, Chawla, & Bowyer, 2006). RapidMiner supports this technique in the operator 'SMOTE Upsampling', but then new issues are optimizing ratios and system requirements exceed our equipment.

These relatively small shadow IT proportions, sometimes at the level of some percentages or less, impede a flowing modelling process. The imbalanced data required in almost all cases interventions like early stopping model training or reduced feature selection. But with all RapidMiner's options we

[2] SMOTe = Synthetic Minority Over-Sampling Technique

used, performances do not look real. Modelling experiences stimulate curiosity for causalities, which is formulated in new research questions:

- Are our 100% scores indicating overfitting, or just a perfect performance?
- Is there a causality between imbalanced data and overfitting?
- Is supervised learning a good basis for data classification?

5.5. Reflection

All findings are a result of this data exploring project. The use of knowledge and tools was determined by the project scope, where RapidMiner was prescribed. As usual getting familiar with the data was very time consuming. One of the reasons was also to get familiar with the ground rules, to understand what data preparation and modelling actions were available and allowed. Initially we did not want to use the database code to identify the truth set, because it felt like cheating. But then, the job was not doable without this. Regarding the tool we have adhered to RapidMiner, without executing scripts inside.

At first, the objective aims to detect targets in a database with data classification. Searching for targets, we could hardly resist the temptation to seize other techniques like text mining or process mining. In order to keep it pure, we decided to retain only data classification, which implies a supervised learning approach, with initial marking target examples. Then, once the classifiers were applicable within acceptable runtimes, they all performed well. Perhaps too good. It causes some suspicion of our approach and way of working. Repeating our models in a different environment can reduce our suspicion.

For this research, only the given scope is explored, and observations are registered. A final step is to compare our data classification results with those of the other Resilience Mining Circle members.

5.6. Acknowledgement

The Open University facilitates this research with guidance and sources. Thesis supervisor Lloyd Rutledge provided us with a critical guidance on research and thesis, where second reader Guy Janssens challenged to stretch to a next level. My acknowledgements are also for the other members of the Resilience Mining Thesis Circle, for their inspiring discussions, exchanging concepts, strategic ideas, as well as technical support. My colleagues assisted me with feedback and technical contributions, and of course, my family who encourages me and let me spend time and energy to finish this academic work.

References

Aggarwal, C. (2015). *An Introduction to Data Classification: Algorithms and Applications*. CRC Press. Retrieved from https://books.google.nl/books?id=nwQZCwAAQBAJ&lpg=PP1&ots=jkUP4BCXc5&lr&hl=nl&pg=PR9#v=onepage&q&f=false

Aher, S., & Lobo. (2014). Comparative Study of Classification Algorithms for Web Usage Mining, *4*(7), 137–140.

Alter, S. (2014). Theory of Workarounds, (May). https://doi.org/10.17705/1CAIS.03455

Alter, S. (2015). A Workaround Design System for Anticipating. *Designing, and/or Pr*(Enterprise).

Arunadevi, J., Ramya, S., & Raja, M. R. (2018). A study of classification algorithms using Rapidminer, *119*(12), 15977–15988.

Bækgaard, L., Lund-Jensen, R., Azaria, C., Permien, F. H., & Sawari, J. (2016). Feral Information Systems, Shadow Systems, and Workarounds – A Drift in IS Terminology. *Procedia Computer Science*, *100*, 1056–1063. https://doi.org/10.1016/j.procs.2016.09.281

Behrens, S. (2009). Systems : The Good , The Bad and The Ugly. *Commun ACM*, *52*(2), 124–129. https://doi.org/10.1145/1461928.1461960

Behrens, S., Sedera, W., Behrens, M. S., & Sedera, M. W. (2004). Why Do Shadow Systems Exist after an ERP Implementation? *PACIS 2004 Proceedings*, 1713–1726. Retrieved from http://aisel.aisnet.org/pacis2004

Cate, R. ten. (2020). *Detecting shadow IT in free text fields using text mining and classification*.

Chandola, V., Banerjee, A., & Kumar, V. (2017). Anomaly Detection: A Survey. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*, 71–97. https://doi.org/10.1016/B978-0-12-398537-8.00004-3

Chen, T., Liu, S., Gong, D., & Gao, H. (2017). Data classification algorithm for data-intensive computing environments. *Eurasip Journal on Wireless Communications and Networking*, *2017*(1). https://doi.org/10.1186/s13638-017-1002-4

Çiğşar, B., & Ünal, D. (2019). Comparison of Data Mining Classification Algorithms Determining the Default Risk. *Scientific Programming*, *2019*. https://doi.org/10.1155/2019/8706505

Development Practical. (n.d.). Retrieved March 16, 2019, from https://www.ou.nl

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78. https://doi.org/10.1145/2347736.2347755

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Hall, L. O., Kegelmeyer, W. P., Chawla, N. V., & Bowyer, K. W. (2006). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *2009*(Sept. 28), 321–357. https://doi.org/10.1613/jair.953

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data, *21*(9), 1263–1284.

Houghton, L., & Kerr, D. (2015). Feral Information Systems Creation as Sensemaking. *Systems*, *3*(4), 330–347. https://doi.org/10.3390/systems3040330

Huuskonen, S., & Vakkari, P. (2013). '" I Did It My Way "': Social workers as secondary designers of a client information system. *Information Processing and Management*, *49*(1), 380–391. https://doi.org/10.1016/j.ipm.2012.05.003

Ibm, A. W. P., Oco, M. J. B., & Ibm, S. M. H. (2011). Data Format Description Language (DFDL) v1 . 0 Specification, *174*.

Kopper, A. (2017). Perceptions of IT Managers on Shadow IT. *Twenty-Third Americas Conference on Information Systems*, (August), 1–10.

Kopper, A., & Westner, M. (2016). Towards a Taxonomy for Shadow IT. *Twenty-Second Americas Conference on Information Systems*, (August), 1–10. Retrieved from http://aisel.aisnet.org/amcis2016/EndUser/Presentations/3

Koskamp, M. (2020). *Mining for workarounds in information systems using outlier detection*.

Kotu, V., & Deshpande, B. (2015). *Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner*. Elsevier Inc.

Kwon, O., & Sim, J. M. (2013). Effects of data set features on the performances of classification algorithms. *Expert Systems with Applications*, *40*(5), 1847–1857. https://doi.org/10.1016/j.eswa.2012.09.017

Lim, T. S., Loh, W. Y., & Shih, Y. S. (2000). Comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40*(3), 203–228. https://doi.org/10.1023/A:1007608224229

Litwack, K. (2018). Workaround , Improvement , or Innovation. *Journal of PeriAnesthesia Nursing*, *33*(4), 575–576. https://doi.org/10.1016/j.jopan.2018.05.008

Marques, V., & Bernardino, J. (2013). Comparison of Data Mining techniques and tools for data classification, (July). https://doi.org/10.1145/2494444.2494451

McLaughlin, E. (2014). What is rogue IT and what are its benefits and pitfalls. Retrieved from https://searchcio.techtarget.com/tip/What-is-rogue-IT-and-what-are-its-benefits-and-pitfalls

Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*, 36–54. https://doi.org/10.1016/j.eswa.2018.03.058

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection : A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569. https://doi.org/10.1016/j.dss.2010.08.006

Oliver, D., Romm, C. T. (2002). *ERP systems in universities: rationale advanced for their adoption.* Idea Group Publishing, Hershey, PA.

Open Universiteit. Ontwikkelpracticum, Course Ontwikkelpracticum (2010). Retrieved from www.oi.nl

Outmazgin, N., & Soffer, P. (2013). Business Process Workarounds : What Can and Cannot, 48–62.

Outmazgin, N., & Soffer, P. (2016). A process mining-based analysis of business process work-arounds. *Software and Systems Modeling*, *15*(2), 309–323. https://doi.org/10.1007/s10270-014-0420-6

Paparella, S., & Horsham, P. (2018). First - and second-order problem solving : when rework and workarounds become an opportunity for improving safety, *44*(6), 652–654. https://doi.org/10.1016/j.jen.2018.07.008

Patil, T. R., & Sherekar, M. . (2013). Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal Of Computer Science And Applications*, *6*(2), 256–261.

Powers, D. M. W. (2011). EVALUATION : FROM PRECISION , RECALL AND F-MEASURE TO ROC , INFORMEDNESS , MARKEDNESS & CORRELATION, *2*(1), 37–63.

Pozzolo, A. D., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification, (November). https://doi.org/10.1109/SSCI.2015.33

Provost, F., & Fawcett, T. (2013). *Data Science for Business, What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media, Inc.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Induction Algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning (IMC-98)*. Madison.

Rangra, K. (2014). Comparative Study of Data Mining Tools, *4*(6), 216–223.

Röder, N., Wiesche, M., & Schermann, M. (2014). A SITUATIONAL PERSPECTIVE ON WORKAROUNDS IN IT-ENABLED BUSINESS PROCESSES : A MULTIPLE CASE STUDY, (November 2015).

Röder, N., Wiesche, M., Schermann, M., & Krcmar, H. (2016). Toward an ontology of workarounds: A literature review on existing concepts. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2016-March*, 5177–5186. https://doi.org/10.1109/HICSS.2016.640

Rouwendal, J. van. (2020). *Tekst en association rule mining voor detecteren van workarounds in vrije tekst die gestructureerde data-invoer omzeilen*.

Saia, R. (2019). Unbalanced Data Classification in Fraud Detection by Introducing a Multidimensional Space Analysis, (IoTBDS 2018), 978–989. https://doi.org/10.5220/0006663000290040

Sandfort, T. (2020). *Resilience mining: identifying workarounds in IT systems using association rule data mining*.

Sharda, R., Delen, D., & Turban, E. (2018). *Business Intelligence, Analytics and Data Science, A Managerial Perspective*. Pearson.

Silic, M., & Back, A. (2014). Shadow IT - A view from behind the curtain. *Computers and Security*, *45*, 274–283. https://doi.org/10.1016/j.cose.2014.06.007

Silic, M., Silic, D., & Oblakovic, G. (2016). Influence of Shadow IT on Innovation in Organizations. *Complex Systems Informatics and Modeling Quarterly*, (8), 68–80. https://doi.org/10.7250/csimq.2016-8.06

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, *251*, 26–34. https://doi.org/10.1016/j.neucom.2017.04.018

Sossi Alaoui, S., Farhaoui, Y., & Aksasse, B. (2017). A comparative study of the four well-known algorithms in data mining. *Advanced Information Technology, Services and Systems*. https://doi.org/10.1007/978-3-319-69137-4

Spierings, A., & Houghton, L. (2012). What Drives the End User to Build a Feral Information System ? Author Downloaded from Griffith Research Online What Drives the End User to Build a Feral Information System ?

Spoel, P. van der. (2020). *Detecting workarounds with data clustering algorithms to improve Information Systems*.

Spronk, J. (2020). *Mining for workarounds in text fields with clustering algorithms*.

Stefanowski, J. (”Institute of C. S. P. U. of T. (2008). Data Mining - Evaluation of Classifiers.

Strong, D. M., & Volkoff, O. (2004). A roadmap for enterprise system implementation. *Computer*, *37*(6), 22–29. https://doi.org/10.1109/MC.2004.3

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2018.08.003

Wirth, R. (1995). CRISP-DM : Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, (24959), 29–39. https://doi.org/10.1.1.198.5133

Wowczko, I. (2015). Skills and Vacancy Analysis with Data Mining Techniques. *Informatics*, *2*(4), 31–49. https://doi.org/10.3390/informatics2040031

Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., … Steinberg, D. (2008). *Top 10 algorithms in data mining*. *Knowledge and Information Systems* (Vol. 14). https://doi.org/10.1007/s10115-007-0114-2

Yin, R. K. (2006). Case Study Methods.

Zhu, X. (2007). Semi-Supervised Learning Literature Survey Contents.