Effective steering of customer journeys via context-aware recommendations introducing OARA, the Order Aware Recommendation Approach

Goossens, J.A.J.

Award date:
2018

Technische Universiteit Eindhoven University of Technology
Department of Mathematics and Computer Science
Architecture of Information Systems Research Group

Effective Steering of Customer Journeys via Context-Aware Recommendations

Introducing OARA, the Order Aware Recommendation Approach

Joël Goossens

Committee Members:
Dr. Ing. Marwan Hassani
Ir. Mike Neergaard
Dr. Ir. Rik Eshuis

Honorary Mentions:
Ir. Tiblets Demewez

Version 1.0

Eindhoven, September 2018

Abstract

Recently the analysis of customer journeys has been a subject undergoing an intense study. The increase in understanding of customer behaviour serves as an important source of success to many organizations. Current research is however mostly focussed on visualizing these customer journeys to allow them to be more interpretable by humans. A deeper use of customer journey information in prediction and recommendation processes has not been achieved. This thesis aims to take a step forward into that direction by introducing the Order Aware Recommendation Approach (OARA). The main scientific contributions showcased by this approach are (i) increasing performance on prediction and recommendation tasks by taking into account the explicit order of actions in the customer journey, (ii) showing how a visualization of a customer journey can play an important role during predictions and recommendations, and (iii) introducing a way of maximizing recommendations for any tailor-made Key Performance Indicator (KPI) instead of the accuracy-based metrics traditionally used for this task. An extensive experimental evaluation then highlights the potential of OARA against state-of-the-art approaches using a real dataset representing a customer journey of upgrading with multiple products.

Keywords: Recommender system, Data mining, Process mining, Customer journey, Context-Aware Recommendation

Preface

This master thesis is the final work I conducted before graduating from the study program Business Information Systems at the Technical University of Eindhoven (TU/e). The research described here was carried out in a collaboration between the Architecture and Information Systems (AIS) group of TU/e and Signify, previously known as Philips Lighting.

I would firstly like to thank Marwan Hassani for giving me the opportunity to conduct the master's project at Signify as well as for all the feedback he has given me over the past few months. Our discussions and your answers to my questions have furthermore been a great help during this research project. For the same reason I would like to thank Tiblets Demewez who supervised me from the company side as your expertise has also been an invaluable aid. Furthermore, I would like to thank Mike Neergaard for taking over supervisory duties at the end of the project and providing me with insights during the finalization phase, as well as Rik Eshuis for taking the time and effort to be part of the defense committee.

I would also like to express my gratitude to both the friends I have made over the years as well as family members which made the times go by in a much more enjoyable manner. Lastly I want to thank my father, sister, and brother-in-law for always being there for me. Your support has meant a lot to me over all these years and I would not be where I am today if it wasn't for you. Mom, I wish that I have been able to live up to whatever hopes and expectations you may have had for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this introduction the context to which the thesis relates is introduced. The problems tackled are given, as well as the matters considered to be inside and outside the scope of this thesis. Following this, a list of abbreviations of terms used during the thesis is presented and finally the structure of the rest of the thesis is explained.

1.1 Thesis context

In today's society, the number of interactions a customer has with an organization is quite plentiful due to the myriad of ways in which customers are now able to interact with organizations. An example of what leads to this is the growing presence of smart devices that continue to be integrated in streets, houses, and even entire cities [4]. These logged interactions can be seen as a sequence, where each time the customer achieves a certain goal with a specific interaction. Such a sequence of observed events which belongs to a single customer is referred to in this thesis as a customer journey. The analysis of customer journeys can be a huge boon towards improving the organizations, as the key objective is to get an understanding of how the experiences of the customer can be enriched though what marketers call their decision-making process [5].

To properly interpret the customer journey data that organizations possess it is helpful to create a visualization of this information to get an idea of which steps are usually taken in the journeys. Such a representation is called a customer journey map. These artefacts often possess a non-linear structure while reflecting behavioural, emotional and cognitive drives [6]. A mapping in this paper is obtained by means of process mining. The result is known as a process model, of which an example is shown in Figure 1.1. The example shown here is from the website of a music festival. First a customer will have to register him- or herself. Upon completing the registration, they go on to either buy tickets or merchandise from a band. In case tickets are bought, it might occur that the customer wishes to also buy tickets from another band, which is indicated by the arrow to and from itself. A customer can end his or her journey after taking either of these actions, but it might also be the case that they still need to change part of their information, for example their payment credentials. This information can be altered and afterwards everything is set to deliver the tickets and/or merchandise which leads to the end of the customer journey. Note that in this example customers are only able to conduct a single purchase. It might also be the case that one wishes to model all purchases made by a single

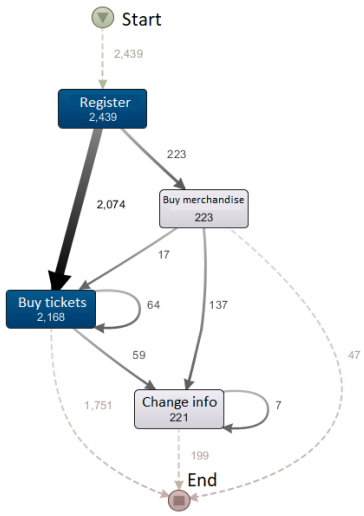


Figure 1.1: Example process model of a customer journey.

customer. In that case the process model would be different and more complex to account for this additional behaviour.

The constant influx of logged user interactions which make up these customer journeys require for customized approaches that are able to act upon such information. As the understanding gained from customer journeys can be applied in many dimensions, so do the approaches which can be built to utilize it. Those particularly relevant for this thesis are the predictions of and recommendations for future events inside these customer journeys. Current state of the art techniques which are able to interpret such information already exist, but these are built for more general purposes. For this reason, this thesis will explore in detail how the information ingrained in the journeys can be utilized to its fullest potential for these two tasks. Based on the results it then becomes apparent whether this information can be a valuable asset to organizations that want to do predictions or recommendations.

1.2 Problem description

Based on the previously described premise more concrete goals can now be introduced. The approach proposed in this paper is called the Order Aware Recommendation Approach, shortened to OARA, and aims to improve upon the current state of the art in three areas.

First, the extraction of a customer journey map by means of process mining is a technique which has been recently contributed in [7]. However, the approach proposed here aims to go beyond simply extracting a model for the customer journeys. The extracted model is used by OARA as an asset to facilitate predictions and recommendations for future steps in the customer journeys in a tailor-made manner. This allows for the value of customer journey data to rise it allows for the utilization of machine learning techniques for these tasks. Usage of these techniques would otherwise involve a large amount of manual labour if it were to be done solely based on the discovered customer journey map.

Secondly, there are currently state of the art predictor algorithms and recommendations systems which can be used to facilitate predictions and recommendations for the customer journey data. Features can be extracted from this data on which these methods are then trained to obtain the corresponding predictions or recommendations. OARA however aims to improve upon these existing methods as they do not take the explicit order information into account which is present in the customer journeys. This information is proposed as a valuable source of contextual information. As such it will be discussed how not taking these details into account can lead to a decrease in quality compared to when this context is applied to the predictions and recommendations.

Finally, there is also the issue that currently the evaluation of recommender systems is mainly focused around prediction accuracy, while other evaluation properties such as novelty are less explored [8]. This is a mismatch with reality. Usually, the goal organizations have when recommending the customer an action is the maximization of a Key Performance Indicator (KPI). This is a value which measures how well an organization is performing on a specific key objective. To provide a solution to the current situation it is shown how one can take KPI maximization into account by using OARA to make sure the recommendations allow an organization to come closer to reaching their goals.

To summarize, the main contributions of this thesis are:

1. Using a process model during predictions and recommendations.

2. The explicit usage of the order of events during predictions and recommendations to potentially increase performance on these tasks.

3. Optimizations of recommendations for a configurable KPI.

While OARA was developed based on these contributions, it should be noted that these contributions can be valuable to organizations which have access to customer journey information in a real-life setting. The following problems then were kept in mind specifically:

1. Providing a clear overview of the steps that are needed to go from the customer journey data to the eventual recommendations. For each step it is then explained which matters should be taken care of and what one should pay attention to make sure the correct methods are applied to reach the maximal potential of OARA. This ensures that organizations will be able to utilize the approach described in this thesis.

2. Achieving high quality predictions for each of the events inside the customer journey that are comparable to or better than current state of the art techniques. The metrics by which quality are assessed can then be configured by the organization as the predictions by OARA can, like most predictor methods, be maximized for any metric.

3. Facilitating recommendations based on values which are of specific importance for a certain organization. The novel way of doing the recommendations makes sure that regardless of the goals of an organization, useful recommendations can be given. This is something that is not always the case for current recommender systems.

1.3 Research Scope

Based on the presented contributions and business problems a number of concrete research questions can be created. These are:

- What are the measurable effects of explicitly taking into account the order in which events occurred during predictions and recommendations?

- Which concepts and techniques need to be applied such that recommendations can be maximized towards any given KPI?

The first research question, as well as the first business problem, require a framework which is built around taking advantage of the order present in customer journeys. This framework is OARA, which consists of a multitude of concepts which are required both to make it possible to take the order into account as well as measuring the effectiveness. However, as time to conduct the research was limited a restriction to the size of the scope needed to be put in place. Matters which are considered in scope are:

- Exploratory data analysis
- Selecting a baseline of customer journey information
- Preprocessing steps
- Prediction steps
- Recommendation steps
- Possible updates to aspects of OARA
- Evaluation

More details on these steps are presented in Chapter 3.

The second research question does not require as large of a setup as the previously given steps contain all required operations to obtain a recommendation. What then remains is maximizing these recommendations for any KPI, for which metrics will be presented. Furthermore, a balance will be struck between the degree to which a recommended action fits the previous behaviour of a user and the degree to which that action increases the KPI.

There are aspects which are left outside of the scope of this thesis. First, there is the process of collecting the actual customer journey data by making devices log the relevant actions. Advice for this is difficult as a business is responsible for determining what kind of actions they want to

monitor for a customer journey. On top of that, are too many devices that can be used to log these actions for there to be general and concrete guidelines for this process.

Additionally there is a limitation of this research that lies in the types of customer journeys used during this thesis. As the research was conducted in cooperation with Signify, the research environment was that of a company. This leads to a mindset that is different from, for example, that of analysing a customer journey of applications for a grant at a government department. As such the methods used, the analyses conducted and the findings obtained will be skewed towards being useful in such an environment. This is not to say that the created approach is not useful outside of a company environment, as the created approach was created with any type of customer journey in mind.

1.4 Used abbreviations

In this section a table is presented with the abbreviations that are used at some point during the thesis.

Abbreviation	**Meaning**
AP	Average Precision
ARCJ	Aggregated Representative Customer Journey
CEI	Customer Engagement Index
CJ	Customer Journey
CLV	Customer Lifetime Value
FN	False Negative
FP	False Positive
ID	Identifier
KPI	Key Performance Indicator
MAF1	Mean Averaged F1
MAP	Mean Average Precision
OARA	Order Aware Recommendation Approach
OCuLaR	Overlapping co-Cluster Recommendation algorithm
RFM	Recency Frequency Monetary values
RCJ	Representative Customer Journey
SAD	Sum of Absolute Differences
SMOTE	Synthetic Minority Over-Sampling
SPD	Streaming Process Discovery
SRCJ	Subset-based Representative Customer Journey
SRD	Sum of Relative Differences
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

Table 1.1: Table of abbreviations used in thesis

1.5 Thesis outline

The rest of this thesis consists of the following sections: Firstly some related work for the rest of the thesis is presented in Chapter 2. In Chapter 3 it is explained in detail how OARA allows for the recommendations to be created. In Chapter 4 a real dataset is used to showcase how OARA can be applied and how well OARA performs in terms of both the predictions and recommendations. To wrap everything in Section 5 the conclusions and future research opportunities are presented.

Chapter 2

Related work

In this chapter all preliminary information which is required to properly understand the rest of the thesis will be explained. First, the main characteristics and types in the field of process mining will be discussed. Afterwards, two mainstream data mining algorithms which are used during the thesis are covered. Then the area of stream data mining is presented where some specific attention is spent on streaming process discovery. Following this, more context on the concept of the customer journey is given, and finally, the concept of a KPI is explained after which several examples are given.

2.1 Process mining

Process mining is a research area which combines the domains of process modelling and analysis with the domains of data mining and machine learning. The goal of this combination is to discover, monitor and improve processes based on knowledge from data which is stored in the event log format regarding the process in question [9]. Event logs show the occurrence of events at a designated point in time, where the event is an action logged by an information system such as the sale of a product. This event is specified to have come from a specific process or instance, also known as case [10]. One such instance or case then encompasses all events belonging to a single customer which can be identified based on an ID.

In Figure 2.1 an overview of the main elements involved in process mining is given. The model aims to give a synopsis of the most important events and to show which events regularly follow each other. This model can however also be enriched such that it allows for analyses regarding different aspects of the real world. An example of such an aspect is the amount of time each event usually takes. These aspects of the real world interplay with IT systems allowing them to be monitored and stored in event logs. The combination of event logs and process models can then be used for the three basic types of process mining [11]:

- **Discovery:** The discovery process is aimed at obtaining a model, usually a process model, based on the events in an event log. It may also be the case that a model is discovered for a different perspective, such as a social network model [9]. The most basic example of an approach which allows for process discovery is the α-algorithm, which constructs a Petri net model that describes the behaviour observed in the event log [12].

- **Conformance:** Checking the conformance involves comparing an a-priori model and the actions which are allowed according to it with the actually observed behaviour that is stored in the event log. Based on this it can be checked how well the allowed actions match with the actions actually taken by the users, and as such to determine where deviations take place as well as how severe they are [13]. This is based on Linear Temporal Logic [14], a type of modal temporal logic where the modalities are referring to time.

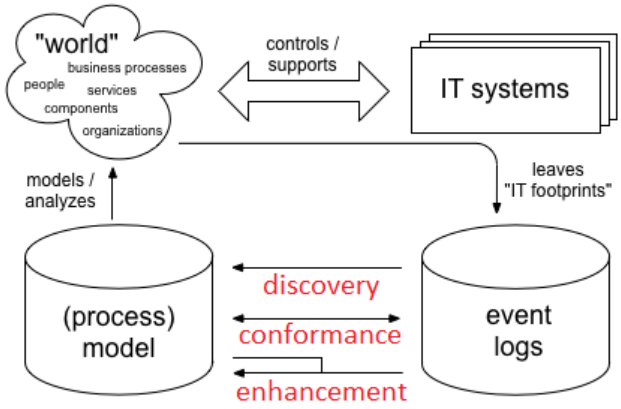


Figure 2.1: An overview of the main elements in process mining.

- **Extensions:** The extensions involve extending an existing a-priori model with information obtained from the event log on matters such as timing, resources, and decisions. An example of such an extension is extending an a-priori model with the average time each events takes. This has been shown to help in effectively identifying bottlenecks in a process [13].

In this thesis these techniques are mainly used to gain a better understanding of the underlying processes in the data. Furthermore, process mining is used to try and find aspects of the data where a split can be placed such that different groups have clearly distinct process models. It also is used to determine the general flow of events to be considered during the predictions and recommendations, which are based on what is allowed according to a previously discovered process model. The environment in which the techniques are used during this thesis is the ProM framework [15].

2.2 Data mining algorithms

Two data mining algorithms are used in Chapter 4, which are Support Vector Machines(SVM) during the predictions of OARA and Gradient Boosting Trees as a competitor for the predictions in the evaluation. In a SVM, a linear decision surface is constructed based on the feature space. This decision surface is then divided into different areas of items which are considered to be similar. These similar items then correspond to one of the classes to be predicted [16].

To further exemplify some of the relevant concepts on SVMs, an example where items exist in a 2-dimensional decision surface is given in Figure 2.2. Two support vectors exist in the decision surface, which separate the items into those which lie above the topmost support vector and below the bottom most support vector. Mathematically the items in this surface have values such that $wx + b$ is less than -1 for the items in the upper region, while it is more than 1 on the bottom. The margin is define based on the shortest distance between two items which lie on opposing ends of the support vectors. When many different types of items exist in the dimensional space the margin can act as an indicator on how similar the types are.

As for the applicability of SVMs, it is interesting to note that it is possible to use support vector machines to cluster unlabeled data [17]. The predictions for which the SVMs are employed in this thesis are already labeled and are non-binary, which is a scenario for which SVMs have been deployed with success in the past in [18].

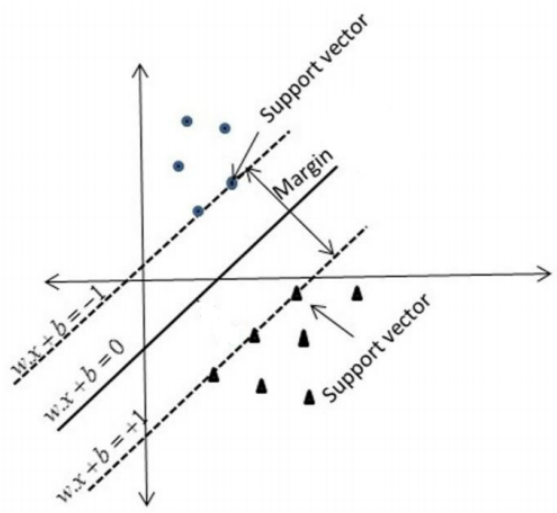


Figure 2.2: Visualization of a Support Vector Machine. Adapted from [1].

Gradient boosting is a technique which tackles classification by creating an ensemble of weaker prediction models. This goes against the more traditional approach of creating a single model which is very strong. The individual models are built in a stage-wise fashion. Afterwards the collection of models is generalized. The generalization is done according to the optimization of a differentiable loss function which can be configured to the classification task at hand [19]. The algorithm by which the models are created during this thesis is a decision tree, more specifically a classification and regression tree [20]. This tree classifies an item based on conditions for feature values that are in place at every split in the tree. In the ensemble each new tree is then trained with an emphasis on the training samples misclassified by the previous tree to get an overall ensemble where different trees are able to correctly classify different samples. This allows for the average strength to rise due to the models in the ensemble covering each other's weakness. Gradient boosting here is considered as a competitor during the evaluation of the predictions. Previous research has been conducted on using this technique for classification in a non-binary setting in [21], where it was able to outperform methods based on pairs of items and regression.

2.3 Stream data mining

Currently data collection technologies, database systems, and the World Wide Web have all become more advanced and widespread. As a result, the amount of data which becomes available drastically increases as well. Stream data mining concerns itself with designing methods which are able to keep up with such data. The data streams in which the information arrives in a streaming setting can be characterized as continuous and typically non-constant [22]. Those data streams are used for storing or processing of the data.

Two main issues arise from such data streams. First, the data they produce is massive or potentially even infinite, and secondly, this data is possibly fast changing. This combination leads to issues with traditional data mining approaches. The reason for this is that these traditional approaches require multiple scans of the data, which is not possible in a streaming setting [23]. Multiple solutions for these issues have been proposed over the years. A collection of several of these solutions was created in [2]. A part of this collection is shown in Figure 2.3. Two main types are identified, which are:

- **Data-based solutions:** These solutions are aimed at summarizing the dataset in some way, or by using a subset from the data which is representative. Both methods reduce the size of

the data and as such make it easier to process.

- **Task-based solutions:** These solutions come in two main flavours. One of them involves modifying an existing data mining technique in such fashion that it fits in the streaming environment. The other is based around introducing an entirely new approach which tackles the computational challenges that come from streaming data.

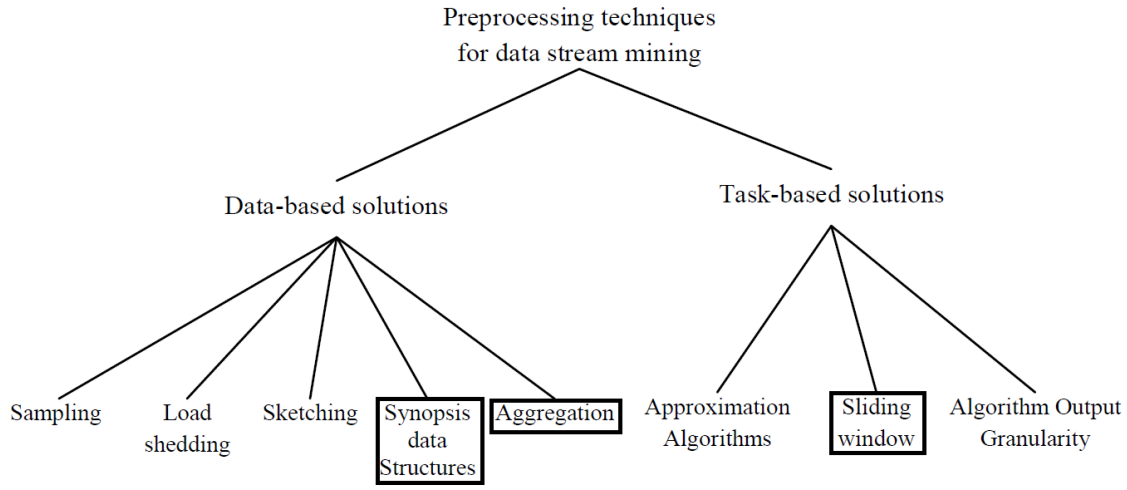


Figure 2.3: Categorization of stream data mining solutions. Adapted from [2].

Obtaining an approach which is compatible with each of the solutions in Figure 2.3 is outside the scope of this thesis, although there are three which are of interest. These have been indicated in Figure 2.3 by a box drawn around them.

First, from the data-based solutions there are the synopsis data structures. This approach tries to identify the most important characteristics of the dataset by applying summary techniques. A downside of this is that less common behaviour gets neglected. However, since rare behaviour is both difficult for predictors to handle and not representative enough to be used during recommendations, this downside is mostly irrelevant for the problem tackled in this thesis.

The second solution is aggregation, where the data from the input stream gets summarized by aggregating it in some manner. A relevant issue here is that highly fluctuating data distributions can affect the method's efficiency [24]. The issue is combated in this thesis by combining it with both the synopsis data structures which will filter out some of the highly unusual and as such highly deviating data, as well as the third solution used in the thesis, the sliding window approach.

The sliding window approach is a task-based solution where when new data arrives, older data either becomes less interesting or is not taken into account during analyses at all. This helps in making sure the most recent, and representative, behaviour gets taken into account. As such the overall fluctuations in data distributions will decline under the assumptions that data which occurred near each other in the time dimension have similar distributions. These solutions will come into play in Sections 3.4 and 4.4.

2.3.1 OCuLaR

The techniques in Figure 2.3 mainly offer high level ways of tackling issues in data stream mining. There are however also more refined approaches specifically built towards adapting to these circumstances. One example of this is the recommender called Overlapping co-Cluster Recommendation algorithm (OCuLaR) introduced in [3]. OCuLaR aims to generate recommendations that are easily interpretable by the users based on data where there is implicit feedback, i.e. no

information is supplied by the users regarding their enjoyment on or motivations for choosing certain products.

The OCuLaR algorithm does this by means of transforming the information of the users and products into tuples of the format (*user*,*item*) inside a matrix R. These tuples are then used to identify overlapping co-clusters, which in this context means a group of users that have bought from a similar group of items. In R, a value of $r_{(user,item)} = 1$ indicates that the user purchased an item in the past. On the other hand, $r_{(user,item)} = 0$ indicates that the degree of interest the user has in the item is unknown as the combination was not observed in the data. For each of the users and items it can then be measured to which degree they are affiliated with each of the identified co-clusters. This information is stored in the vectors v_{user} and v_{item}. The range of values the values inside these vectors ranges from 0 to 1, where 0 indicates that the user or item is not affiliated with the co-cluster and 1 that there is a heavy affiliation. Based on this configuration, if there is a co-cluster a then the probability that a co-cluster generates a new item that suits the user is determined to be $1 - e^{-[v_{user}]_a [v_{item}]_a}$. These values are then the main driving force behind the recommendations, where OCuLaR favors the highest probabilities.

An example of the (*user*, *item*) pairs and co-clusters is visualized in Figure 2.4, where three main co-clusters are identified. These co-clusters indicate similar behaviour in users, which will in most cases leads to higher probabilities that a new item from this co-cluster suits the tastes of the user. The reason for this is that one can compare the items of similar users to find products which a user has not yet bought, but that users similar to him have bought in the past. This is based on the idea that similar users appreciate similar items. As such, users which lie in the same co-cluster are likely to obtain similar recommendations as their peers when using OCuLaR.

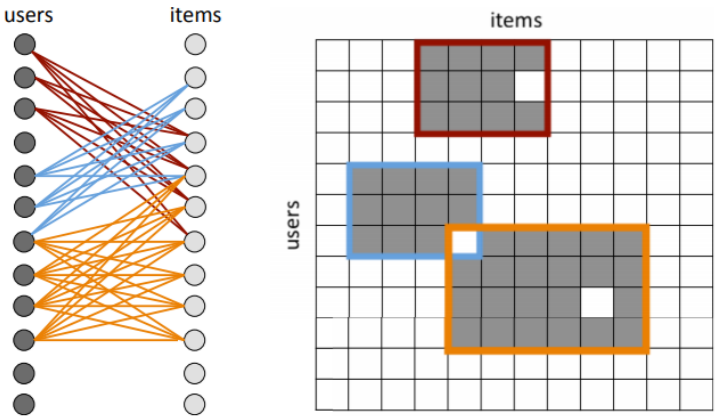


Figure 2.4: Visualization of overlapping user-item co-clusters. Adapted from [3].

2.3.2 Streaming process discovery

A subsection of stream data mining is called Streaming Process Discovery (SPD), which has ties to the process mining domain. SDP among other things concerns itself with obtaining process models on data streams. This task is heavily influenced by concept drift, which is the phenomenon where changes in the behaviour of users can occur over time. Such changes prompt a need for changes in existing process models. In research, there have been multiple attempts to tackle the issue. A subset will be discussed here from which inspiration was taken during the creation of the techniques proposed in this thesis.

The most basic way in which SPD can be handled is by combining a process discovery algorithm with a sliding window. This has previously been put into practice in an adapted version of the Heuristics Miner in [25]. As was previously mentioned, the sliding window can help lessen the deviations in the considered data, counteracting concept drift. There are however some issues which arise from only using this technique. New processes are not necessarily immediately con-

sidered here as the model is unable to be updated for every new incoming event as this leads to too high of a computational burden. Furthermore, all events are handled twice, once for storage and once for mining the model, which is undesirable [25].

To counteract these issues the Heuristics Miner was combined with the 'Lossy Counting' algorithm. The main implication of using this algorithm is that the frequencies of certain aspects of the data such as the actions are recorded and periodically the most uncommon occurrences are deleted from memory. The model can then be more efficiently updated based on these counts, making it a better fit in the streaming domain. Note that it has also been shown that an upper limit to the amount of memory can be put with regard to the number of counts we can keep in memory if so desired [26].

Another way of handling streaming process discovery is by storing cases in a concept known as prefix-trees. In these trees the sequences of all events are stored, where prefixes are also useful based on the idea that unfinished cases are just sub-sequences of finished ones. Storing the information in such a prefix-tree has been reported to allow for the calculation of statistics from the batches [27]. Furthermore, using prefix-trees leads to efficient preprocessing of events in the stream while keeping an upper bound on memory usage [27], similar to the approach using Lossy Counting. Outside of these factors in [27] the prefix-trees are also handled in a way which can give higher importance to newer behaviour. This is useful in situations where older behaviour is less relevant than new behaviour.

2.4 Customer journeys

As described in [28], the term customer journey (shortened to CJ from here on out) is one which is quite widely used in scientific literature yet no common understanding exists with regard to what a CJ exactly entails. Descriptions used in previous research are that a CJ is the cumulation of repeated interactions between the service provider and customer [29], an "engaging story" based on the interactions of a user with the service [30], or a "walk in the customer's shoes" [31]. What all descriptions have in common is that a high importance is placed on the experience of the customer. This is carried over into this thesis as well.

A distinction can be made between CJs which are considered to have clear start and end points [32]. An example of this is an application procedure where the application is either accepted or rejected in the end. There are however also cases where the journey is viewed as a more open ended process [33]. For this thesis the latter interpretation will be used, where the focus lies on processes where one is never certain if the actor will remain engaged in the process or not. An example of this is a customer buying items at a certain store. One can never be sure that the customer will return to this store to buy additional products or not. Therefore, it is unclear when his or her CJ will end. The concept of closing off cases which have not had recent activity is not unprecedented and has been shown to be successful in the past in [27] providing evidence that it can be of use.

A technique for capturing the experiences of the customer in their journey from start to end is customer journey mapping, which tries to map out all parts of the journey in a model. The goal of this visualization is to make it easier to understand, discuss and improve upon the most prevalent CJs observed from the users of a service [34]. The interactions of customers with the organization are called touchpoints and due to increasing prevalence of technology in society these touchpoints are becoming more numerous. Touchpoints are also at times referred to by the terms action or event, which often are more intuitive for those not active in this research field. Both of these terms are also interchangeably used during this thesis, which is in part due to there being an overlap with the process mining research domain. There the term event refers to the same matter, making it often more intuitive than the term touchpoint. Aside from this, users often do not all act in the same manner. As such, the order in which events occur can be rather erratic [35]. Dealing with the different CJs which result due to this variance is one of the challenges that is tackled in this thesis.

While customer journey mappings are mostly focused on obtaining a visualization, a combina-

tion with process mining has been showcased to be possible in [7]. In this paper the events which were relevant to the CJ were retained from the dataset such that a process model could be created upon them. Based on this process model a visualization could be created while also allowing for further analyses on these CJs such as those involving performance and handover of work. Incidentally, two of the tasks proposed as further research in this study happen to be covered by this thesis. These are the creation of new techniques and metrics used for clustering CJs and their representatives, as well as the creation of methods which allow for the prediction of the customer's next action.

The notation for the customer journeys will be borrowed from the process mining domain, since as was shown by [7], a customer journey can be formed based on an event log. The examples on notation given here are based on Figure 1.1. First, a single event in the customer journey called for example *Register* has the following combination of information: $Register = (c, a, t)$. Here, c stands for the case, which is a specific customer, a is the action performed, registering, and t the time at which the action was performed. A customer journey consists of multiple such events and is then denoted as $CJ = \langle Register, BuyTickets \rangle$, where *Register* and $BuyTickets$ are events belonging to the same case ID. Their c values are then also consecutive timestamps. The entire collection of journeys is here equivalent to an event log and is denoted as $Log = \langle Register, BuyTickets \rangle, \langle Register,$
$BuyMerchandise, ChangeInfo \rangle, \langle Register, BuyTickets, BuyTickets, ChangeInfo \rangle$. Note that based on the presence of a loop there is no exhaustive Log which covers all possible customer journeys and that journeys belonging to different customers might be interleaving depending on t.

Considering all possible CJs can be troublesome both during visualizations and analysis. Creating a visualization which covers a lot of different possible orders for a large number of actions in CJs will lead to a model which has many lines flowing between the different actions making it difficult to interpret. During analysis the uncommon CJs can be troublesome during for example predictions, since there is insufficient data for training purposes. To counteract these issues the term Representative Customer Journey(RCJ) is introduced in this thesis. A RCJ aims to represent a relatively large portion of the customer base by itself. Therfore, the actions included in it have to be relatively common as well, since uncommon behaviour is not something which represents the customer base well. Upon identifying a number of RCJs the customer base is analysed based on them instead of all CJs. This is similar to the pruning process in the field of decision trees where the least informative nodes get removed from the tree. The exact construction of the RCJs is discussed in more detail in the Approach Chapter.

2.5 KPI information

The eventual goal of the recommendations is the optimization of a Key Performance Indicator (KPI), which is a measurable value that shows how well a company is performing both with regard to financial and non-financial aspects. The aspects are derived based on previously set goals which are relevant to an organization [36]. Both high and low-level KPIs exist, where the high-level KPIs are aimed at the general performance of an entire company, while the low-level KPIs are aimed at a more specific aspect of them such as sales or marketing. To provide additional intuition on the concept of KPIs three examples of them are discussed in the following subsections. The reason these specific KPI examples are used is that the Customer Lifetime Value and Customer Engagement Index are shown to have potential value during the evaluation of the recommendations, while the Recency, Frequency and Monetary values are later used to segregate the userbase.

2.5.1 Customer Lifetime Value

The Customer Lifetime Value(CLV) is, like the CJs, a concept which has slightly differing definitions across the literature. For example, in [37] it is defined as the present value of the expected

benefits minus the burdens from customers. It has also been defined as the sum of cumulated cash flows minus the weighted average cost of capital per customer in their time with the firm [38]. There is an important distinction to be made here. The first definition includes benefits in a broad manner, i.e. both direct benefits such as the customer buying items and also any positive marketing they may do due to being satisfied with the services. The second definition on the other hand is purely focused on the monetary aspects. In this thesis the CLV will, like in the second definition, be based solely on the buying behaviour. The reason for this is that usually only the concrete actions taken by the users are available in CJs, and as such not enough information is present to use the broader definition.

Many types of contexts have been used to classify customers using CLV. The most common ones are Lost-for-good and Always-a-share [37], membership and non-membership [39], and most predominately contractual and non-contractual settings. In the contractual context there usually is a contract or membership in place such that expected revenues can be forecast with decent accuracy. If there is a constant usage of the service then the expectation is that cumulative profits will increase during the lifetime of the customer [40]. In the non-contractual context the relationship with the customer is less stable. The customer then determines by themselves when to interact with the company, switching to a different company requires little effort, and no contract or membership is in place. This usually leads to the business performing actions such as marketing campaigns to make sure that the customer remains interested in the business over the course of their lifetime.

Based on these two options, both contexts can apply to CJs. There are both contractual services which can have CJs, for example a membership to an internet provider, as well as non-contractual services, such as a buyer at the local grocery store. The effect of what context applies lies mostly in how easy it is to determine the endpoint of a CJ, as in a non-contractual context one does not know when the customer stops being interested [41]. Furthermore, the degree to which customers behave erratically is different. In most cases this is higher for the non-contractual setting as the customer has more freedom there, making it more difficult to model and reason on the CJs.

Outside of the context, there is one final distinction to consider in how to represent the CLV. This distinction lies between if CLV is calculated in a static or dynamic way [41]. In the static approach firstly it is estimated how much longer the customer will remain with the business. Afterwards, for each of the items it is determined how likely it is that the item will be purchased in the remaining time. The issue with this lies mostly in the estimation of the remaining lifetime of a customer, while also not taking into account changes in the customer's behaviour after the estimation. This is fixed in the dynamic approach, where every possible state which a customer can reach can be modeled with a certain KPI value. These states can for example then be based on years, where based on the currently bought products of a customer it gets estimated how likely it is that he ends up with a different set of products in the next year. This dynamic approach meshes very well with the CJs, as there every new touchpoint involves an action which has a certain effect on the CLV as well. Therefore, the dynamic approach is what is used during the thesis.

An example can be found in Figure 2.5, where the CJ is based on a customer buying a TV as well as possible add-ons to it. Buying the TV increases the CLV by 15 points, adding the stereo another 10, and internet access 25 on top of that. The percentages show how likely it is for the customer to follow this path, where churning means that the customer has lost interest in the company. As can be seen on the bottom of Figure 2.5 the CLV is then calculated based on the CLV values in each of the steps and the likelihoods of taking those steps.

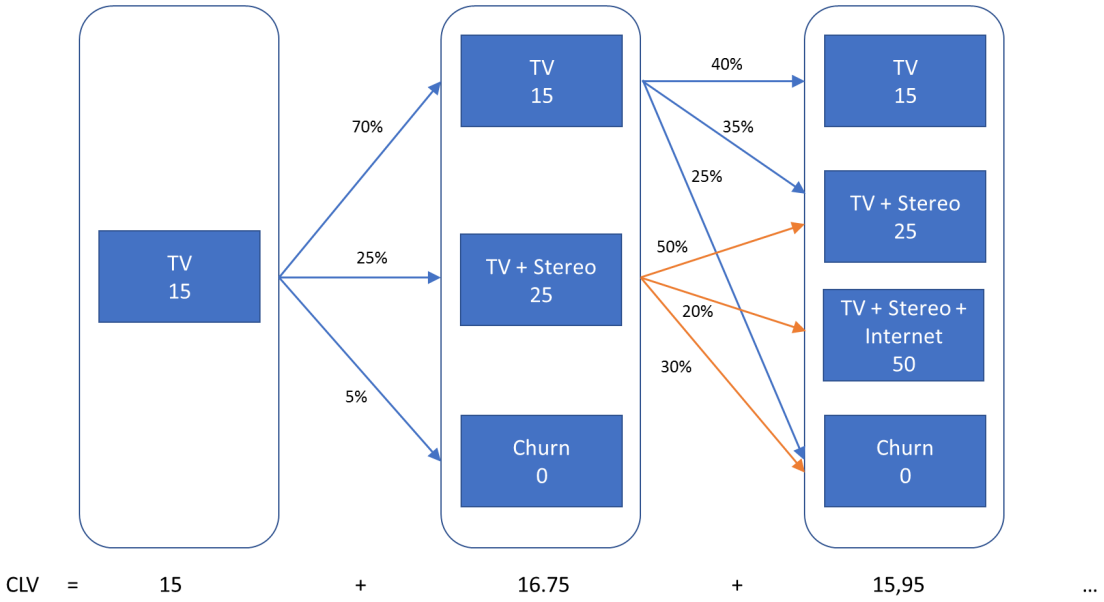


Figure 2.5: An example of the dynamic calculation of the CLV value.

2.5.2 Customer Engagement Index

In general, the Customer Engagement Index (CEI) is a measure which is used to determine how invested the customer currently is in the company. In the literature there is some debate on whether this refers solely to the observable behaviour [42] or also to psychological aspects [43]. Furthermore, a distinction can be made between only using non-purchasing actions [42] and combining these actions with purchasing actions [44]. It should be noted that in cases where psychological aspects are excluded that the CEI heavily resembles the CLV which also aimed to assess the value of a customer by means of the perceived actions. The choice between including psychological information will in most cases primarily be based on the presence of such data, as it is a valuable source of insights. However, in the context of CJs, which are primarily behaviour-based, it will be rare to have this information.

In [45] a framework is given for the CEI. This framework allows for it to be applied in multiple contexts, including the current company environments which are mostly non-linear and dynamic. Furthermore attention in the work of Maslowska et al. goes to recognizing the growing role that non-purchase engagements play in purchase decisions and, more importantly for the setting in this thesis, the rationalization of non-purchase decisions. Several main groups of actions have been identified such as observing, participating and co-creating. Keeping such groupings in mind during the calculation of the CEI can be fruitful as it can help in splitting up the entire non-purchasing behaviour in such smaller groups where the CEI is different. If such differences occur then one can start looking into what causes these differences, and in turn from such understandings insights may arise which can be applied to the less well performing groups to increase the CEI as a whole.

2.5.3 Recency Frequency Monetary value

The Recency, Frequency and Monetary values, often shortened to RFM, is a KPI which is based on how well a customer performs in the recency, frequency and monetary dimensions which has been introduced by Bult and Wansbeek [46]. Recency here means the time interval which has passed between the previously observed interaction of the customer and the present. Frequency involves how often a customer has, possibly in a specific time period, interacted with the business. Monetary value is based on the cumulated amount of money the customer has spent at the business.

In this thesis RFM will be applied for segregation purposes, which is a goal for which it has been used for in success in previous studies [47] [48]. Based on the reported results the segregation has been helpful in [47] by making it easier for decision-makers to identify market segments in a clear manner which in turn allowed for the development of better marketing and sales strategies for customer retention. Outside of clustering, the RFM values have also been shown to be useful in classification tasks. When RFM was combined with other features in the data the classifications performed better than when it was compared to a baseline of less complicated classifications [48].

The clustering algorithm which will be used for this is K-Means++ [49], which is an alteration of the traditional K-Means algorithm that determines the initial points in the center by taking the squared distance from the closest center that was already chosen. In turn, this will in most cases lead to a faster calculation of the clusters while also increasing their potential value [49].

The notation when RFM groups are mentioned during this thesis will be as follows: XYZ, where X represents if the Recency. This recency is then either relatively high, indicated by an H, or low, indicated by a L. Similarly Y indicates the relative Frequency and Z the relative Monetary value. For example, if the HLH group is considered, then the latest event was observed relatively recently, relatively few events were observed in total and the monetary value of the steps taken by the customer is relatively high. This might be achieved due to the few purchases which were observed involving more expensive products.

Chapter 3

OARA: Order Aware Recommendation Approach

In this chapter a recommendation approach called OARA is proposed by which predictions and recommendations can be made for the next events in the CJs. First, in Section 3.1 an overview will be given of OARA to get a general idea of how it functions. Once the main idea is clear, the individual components of the approach are explained in more detail. The value of an initial exploratory data analysis is explained in Section 3.2, after which in Section 3.3 it is covered how the baseline data can be determined. Following this in Section 3.4 the preprocessing which is needed for the predictions and recommendations is described. The concrete steps taken during the predictions and recommendations are given afterwards in respectively Sections 3.5 and 3.6. Finally, in Section 3.7 it is shown how parts of OARA can be updated based on new information such that recommendations remain valid over time and in Section 3.8 a scheme is given for the evaluation of both the recommendations and predictions.

3.1 Overview

A general overview is given of the OARA's components and how these components interact with one another. A visualization of these components can be found in Figure 3.1. The starting point is an optional exploratory data analysis to gain further insights into the dataset, which is only required in case the data scientist is not familiar with the dataset. Afterwards, the baseline customer information is determined. This is the information on which future predictions and recommendations are based. Once the data to be used has been determined some preprocessing is required for it to be usable in the two following components, the predictions and recommendations. The predictions are used in the recommendations, and as such these activities cannot be conducted in parallel once the preprocessing has finished. Once the predictions and recommendations for the customers have been completed, it is shown how updates can be conducted on the representative journeys. To wrap things up, relevant evaluation techniques for both the predictions and evaluations are presented to assess how good the performance is.

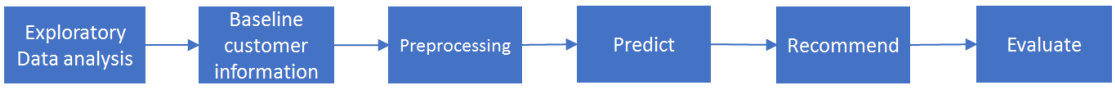


Figure 3.1: A general overview of the components in OARA.

3.2 Exploratory data analysis

The exploratory data analysis acts as the start of the approach, although in some cases it can be better to skip this step and continue towards immediately obtaining the baseline customer information. The reason for this lies in the goal of this exploratory data analysis. This is to discover patterns for both hypothesis development and, more importantly, refinement [50]. In scenarios where the data scientist is already intimately familiar with the features, their distributions and the general information included in the data, an exploratory data analysis will not increase the level of understanding and therefore offer little value. However, if the knowledge of the dataset is not this deep, an exploratory data analysis can help the data scientist understand the data better. This will be a major boon during later steps of OARA as it plays a role in making sure proper decisions can be mad.

For CJ data one of the advocated exploratory data analysis approach is to utilize process mining to create a process map. This allows the data scientist to get an overview of how the actions in the CJs flow into one another. This overview allow for insights into the process described by the CJ to be gained quickly. Examples of such insights are the number of distinct CJs, the degree of similarity between these CJs, how common each distinct CJ is and the average length of the CJs. All of these matters can be derived using process mining and can play a big role in how to use the data moving forward. The reason for this is that different methods should be applied based on these characteristics.

Another aspect for which the exploratory data analysis can be helpful is determining which features heavily influence the mined process models. This information is mainly helpful during the predictions. When the discovered process models significantly differ when split on a certain feature, that is a clear sign that this feature has distinguishing qualities that can be used during machine learning to obtain the correct predictions.

Outside of process mining, one can also use different graphical techniques for numerical features to gain further insights into them. A collection of such techniques has been proposed in [51]. One of the techniques mentioned in [51] is the usage of box plots to obtain the mean as well as how much the values usually deviate from this. It also identifies if there are relatively many or few outliers. Similarly, a histogram can provide an intuition regarding the distribution which a feature may follow by showcasing where most values are concentrated and how they spread out. An example of both of these graphical techniques is given in Figure 3.2. In the box plot there seem to be 3 outliers while most values are centered around 0, and based on the histogram the data seems to mostly follow a normal distribution.

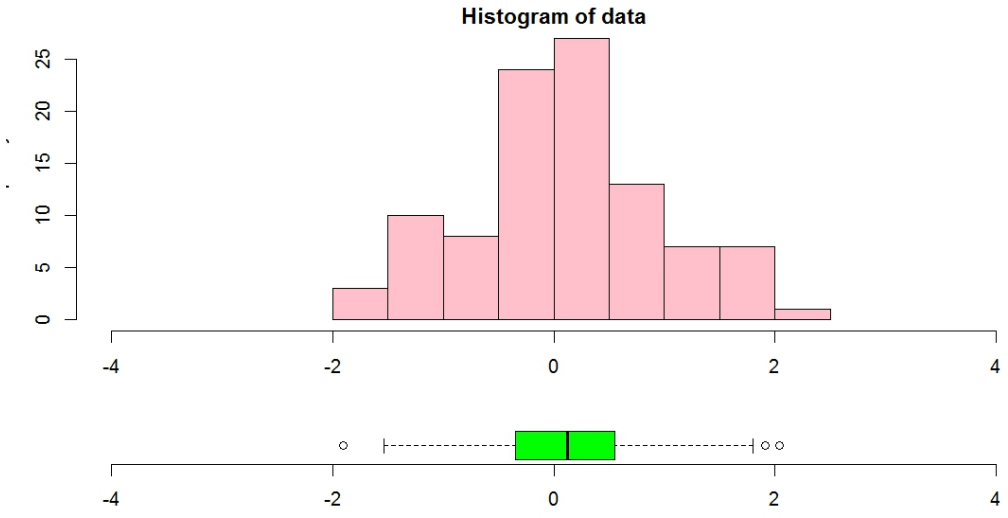


Figure 3.2: An example of a histogram and box plot.

3.3 Baseline customer information

What is important when determining the baseline customer information is to make sure that the data is both from the correct time frame and that it contains the information needed to properly do the predictions and recommendations. The latter requires there to be features which allow for the classes to be distinguished from one another. These matters are here discussed in further detail.

The selection of the time frame involves determining the correct period of time from which information on the CJs should be considered. A balance should be struck between past and recent behaviour, as both can be troublesome if they are the primary focus. Considering too much past behaviour can lead to a situation where older CJs do not represent behaviour which new customers would exhibit. An example of this is when the old CJs involve products which are not in production anymore, in which case new CJs are unable to take the same actions. On the other hand one should also be careful to not focus only on the newest CJs, as this can lead to a situation where a wide arrangement of possible CJs are left unconsidered. Seasonal effects are an example of this, where are certain product may only be bought during winter such as snow tires. As such, in most cases it will be best to try and include as much information as possible while monitoring for example per month if the oldest CJs being included do still involve the same product types, actions and other distinguishing attributes.

Another aspect to take into account during the selection of the time frame is making sure that customers have actually 'finished' their CJ. As was pointed out in Section 2.4 the CJs can be open-ended. The CJs then lack an official final event that signifies the end of the relationship between a customer and an organization. As such customers should have ample time to determine if they want to take a certain action or not, while also not cutting off journeys which contain an action that happens to require a large amount of time in real life. An example of this is sending physical mail overseas. The exploratory data analysis can be helpful here, as an advocated activity of it is examining the distribution of times it usually takes customers to do an action. Based on for example the mean and variance, or percentile information, which were calculated during it it is possible to decide upon a satisfactory time-frame after which the CJs can be considered to have finished.

As previously mentioned the data should facilitate proper predictions and recommendations, which makes it important to carefully consider the information to be included. The features will be derived from this information, which are important aspects of the data on a per customer level. Features which do not properly distinguish between the different outcomes will lead to poor results for both predictions and recommendations. If the dataset lacks such distinguishing features one can try to enrich the current dataset with additional context information that can be helpful in segregating the user base. An example is the inclusion of demographic data for predictions on purchases in a clothing store, where there are a lot of differences in the CJs based on age and gender.

3.4 Preprocessing

The preprocessing stage consists out of a multitude of steps. These steps can be divided in the preprocessing which is needed for predictions and those needed for recommendations. Note that the data should be in an event log format for these steps to proceed smoothly, so the mapping between CJs to process mining described in Section 2.4 is utilized here. As the process of transforming data into such a format is case and implementation specific this is left outside the scope of this thesis. In Figure 3.3 the general overview of the steps taken during preprocessing can be seen. First, the relevant features are extracted after which for the predictions the users are segregated. Each partition then has a process model mined for it. Preprocessing required for the recommendations includes obtaining KPI information, while also several steps are taken to ensure the correct RCJs are in place based on how homogeneous the population is.

As was mentioned in the explanation of Figure 3.3 the first step is the process of feature

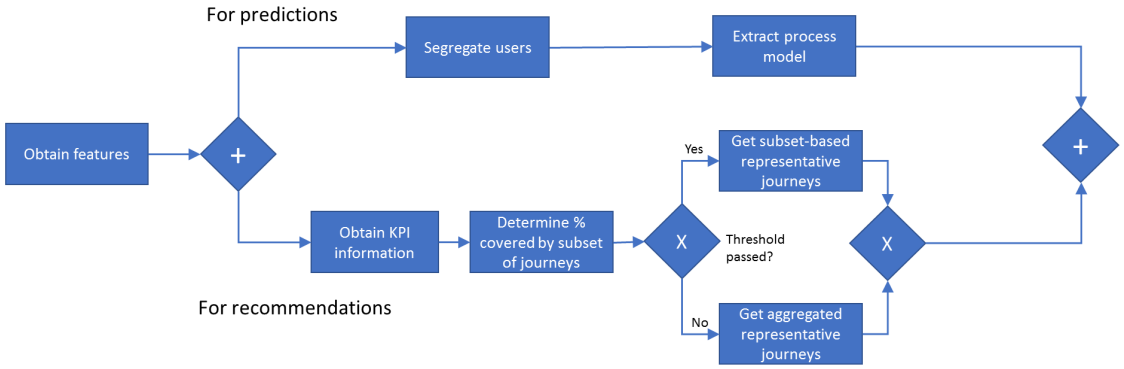


Figure 3.3: A general overview of the steps taken during preprocessing.

extraction based on the journeys in the baseline data. There are parts of the data which can immediately be used as a feature, such as the gender of a customer. However, in most cases the data will need to be altered before it is in a format usable for machine learning. Examples include aggregating information from the last month of the user's activity, encoding slightly different strings to have the same label, and marking if an event was conducted on a public holiday. The creation of the features will be domain specific and discussions with domain experts can lead to insights with regard to where special attention should be placed. The feature creation step is very important, as a small number of great features can quickly outperform a multitude of mediocre ones if the small number of features very accurately indicate the behaviour of customers.

3.4.1 Preprocessing for predictions

The first step taken during preprocessing for the predictions is segregating the users. The goal of segregating the customer base is to make it easier for the used classification algorithm to predict the next action of the customer correctly by isolating observed behaviour into smaller subsets. These more cohesive subsets contain similar examples which allow the predictor to be optimized better. The reason for this is that predictors have an easy time predicting a smaller group of samples with similar behaviour, while a large and varied collection of behaviour will lead to a predictor that does not perform very well. Segregation then allows the predictor to specialize in determining the small differences in the subset rather than finding rules which work properly for all CJs.

This allocating into subsets can be done based on domain knowledge, where for example there could be bronze, silver and gold customers which have different permissions and usage patterns. Alternatively the segregation can also be done based on previously identified features which happen to be an indicator for certain behaviour, such as children not watching violent programs which air late at night on TV in which case the age is an indicator of behaviour. The former option of the domain knowledge is usually preferred, as finding out which combination of features properly segregates the data can be a lengthy manual process. Furthermore, in many cases the discovered relations will already be known by a domain expert, further decreasing the efficiency of determining the distinguishing features manually.

Once the behaviour has been split into subsets, a process model is derived for each of the segregated parts of the user base. This is done to further streamline the journeys considered for predictions. While the customer base has already been slightly streamlined by the segregation, there are still samples inside these partitions which are very rare and specific enough that they do not occur often enough for a predictor to learn them properly. The process model can then be configured to properly generalize the CJs.

There are many scientific tools available to do this, most notably ProM [15] and RapidMiner [52]. The main important point is to strike a good balance between interpretability and

complexity. The model should not be overly simplistic to the point where important parts of the customer journey are left out, while also making sure that the included events are common enough that they can be learned and predicted properly by a machine learning algorithm. The optional exploratory data analysis can help a great deal here, as this will aid in finding a proper balance. Upon having collected the process model, only those journeys which fit into the process model are used from the baseline customer information during the predictions.

3.4.2 Preprocessing for recommendations

While the preprocessing steps during prediction are aimed at making sure predictions can be conducted well, the steps for recommendations are focused on creating artefacts that facilitate sensible recommendations for a given customer. These steps are explained in further detail in the following subsections.

Determining the KPI

As mentioned previously the goal of the recommendations is the maximization of a certain KPI, of which examples can be found in Section 2.5. The first task to take care of is therefore to determine which KPI should be maximized. The KPI will in many cases be domain specific and should follow the general rule that their value serves as a good indicator for how well the business is performing in a certain aspect, e.g. customer engagement when using the Customer Engagement Index. If the domain for some reason traditionally lacks KPIs, or domain expertise is lacking, then starting from the CLV value can be of use. The CLV is a broad enough concept that it can be altered to suit most situations. Once the KPI has been determined, it is calculated for all customers in the baseline customer set.

Determining coverage using subset

The recommendation a user receives is ideally one which leads to the highest possible KPI value. However, in that case every user would receive the same recommendation, as there is a specific sequence of actions which will maximize the KPI. The reason that this is not feasible is that this does not keep in mind the preferences of the users, and as such will lead to recommendations that are unlikely to be followed by all users.

Recommendations are therefore mainly based on behaviour which is both observed often and in line with past behaviour from the customer. The reason for basing recommendations mostly on common behaviour is that a new customer is more likely to follow what most have done before them, and as such recommending mostly common behaviour increases the likelihood of the customer following through on the recommended action. Note that a decrease in overall population also occurs as a side effect of matching the CJs to the discovered process models, yet here once again the entire baseline customer data is used. The reason for this is that while there are customer journeys which may be too specific to be properly predicted due to there being insufficient training samples, the increase in variety can be useful for matching the CJ to the RCJs with more precision.

Obtaining most of the general behaviour in a summarized format is what the following preprocessing steps of the approach aim to achieve. This is slightly similar to the steps in the preprocessing of the predictions, but the key difference is here that a generalization for the entire customer base is required instead of per segregated group. A relatively easy way in which common behaviour can be summarized is by only considering the CJs which are observed most often. This is an approach which is in line with the synopsis data structures described in Section 2.3. If this can already represent most of the customer base then that is the preferred course of action for the next step. However, to make sure that enough of the customer base is represented by this subset of CJs, a coverage threshold is put in place. What is meant here by the term coverage threshold is a minimum percentage of all CJs which need to be represented by the subset taken into account. The formula by which the coverage is determined is as follows:

$$Coverage = \frac{\sum_{i=1}^{m} weight(i)}{n} \tag{3.1}$$

n is here simply the number of CJs and m corresponds to a maximum for the number of different variants of CJs to be considered. CJ variants are a unique sequence of actions in the CJs. This maximum is in place to make sure that the threshold is not reached by simply taking a very large number of possible CJ variants into account, as this would defeat the purpose of generalizing the considered behaviour. The *weight* of such a CJ variant is determined by the number of CJs which fit the same actions as in the variant.

The exploratory data analysis, if conducted, should have provided some intuition on how heterogeneous the current user base is. Furthermore some leeway should be given with respect to the timing at which actions are performed in these variants if this is relevant for the features, as it is extremely unlikely that all actions will be taken at exactly the same time in a CJ. The amount of leeway depends on the CJ in question, as giving a week of leeway to a CJ which only consists of 3 days is nonsensical, while it may be a reasonable amount of time in a journey that lasts a year.

To give an example of how using the coverage can work out, consider the situation in which the data consists of Table 3.1. Here there are 5 customers which only take 2 actions during their CJ, which are done in 4 different orders resulting in different KPI values. $\langle B, C\rangle$ is the only CJ variant with $weight = 2$ so if for example the coverage threshold would be put at 65% and at Equation 3.1 $m = 2$ the threshold will not be passed, as $Coverage = \frac{2+1}{5} = 0.6$. On the other hand, if the threshold would either be below 60% or $n > 2$ then the coverage threshold could be passed.

Customer ID	**Customer Journey**	KPI
1	$\langle A, A\rangle$	7
2	$\langle A, B\rangle$	5
3	$\langle B, C\rangle$	4
4	$\langle A, C\rangle$	6
5	$\langle B, C\rangle$	4

Table 3.1: Data for coverage threshold example

Subset-based representative customer journeys

If the coverage threshold was passed, then the customer base can be properly represented by a sufficiently small subset of the observed behaviour. The Subset-based Representative Customer Journeys (SRCJ) are built up of the most commonly observed behaviour based on the CJ variants considered during the previous step. As such, the synopsis data structures discussed during Section 2.3 fit the current dataset well and are therefore the used preprocessing technique for data stream mining purposes.

What needs to be taken into account from the subset of considered CJ variants are firstly the weights, which as previously noted are based on how many customers are actually represented by a specific CJ variant. This will be used later during the updating process to make sure that only the most common CJ variants remain in the subset. Secondly, the resulting KPIs of the CJ variants are required, which can simply be looked up based on the previously calculated values. **The reason the KPIs are needed here are that in cases where two variants would have the same weight the one with a higher KPI value is preferred.**

To continue on from the example based on Table 3.1, let the coverage threshold be at 50% with 2 CJ variants being allowed such the coverage of 60% surpasses the threshold leading to the subset-based representative journeys being allowed. The representative journeys would then be as shown in Table 3.2. Note that $\langle A, A\rangle$, $\langle A, B\rangle$ and $\langle A, C\rangle$ all have $weight = 1$, and $\langle A, A\rangle$ leads to the highest KPI value. As such it wins the tiebreaker and is used as a SRCJ.

Representative Journey	KPI	Weight
$\langle B, C \rangle$	4	2
$\langle A, A \rangle$	7	1

Table 3.2: Data for subset-based representative journeys example

Aggregated representative customer journeys

If the coverage threshold was not passed, something different from the subset-based approach must be done to obtain the RCJs, as the behaviour is too volatile to be summarized by means of a subset. In this case a different data-based solution is applied which is based on the aggregation method instead of the synopsis data structures. The Aggregated Representative Customer Journeys (ARCJ) aim to provide RCJs by firstly grouping together the different CJ variants based on how well they perform with regard to the KPI. Once the groups are formed, inside each KPI based group group $\sum_{j=1}^{l} \frac{\sum_{i=1}^{n} Action(j)}{n}$ is calculated. Here n is once again the number of CJs, l is the number of possible actions observed in the journeys and $Action(j)$ is based on the j'th action which may or may not be present inside the each of the journeys inside the group. As one can tell from the formula, the averages then form the ARCJs as the cumulative actions are divided by n. Furthermore they are then ranked based on the KPI value obtained from following them and have their weights recorded just like the SRCJs.

This way of obtaining the RCJs is less precise than the SRCJs as values are based on averages, which can lead to for example 2.5 products being bought. In these cases rounding may be needed, and it is domain specific how to handle this. It may for example be the case that you do not want a loan which is paid off for 51% to be rounded upwards to 100% as there is still a substantial amount which needs to be paid.

To further illustrate the concept of the ARCJs consider the CJs in Table 3.3. As in the example of the SRCJs the CJ consists purely out of 2 actions and in this case all actions involve some action A which can occur multiple times. The ARCJ would have for Action 1 the value $(1A + 1A + 3A)/3 = 1.67A$ while for Action 2 it would be $(1A + 2A + 3A)/3 = 2A$. As 3 CJs are included here, the weight of the ARCJ will be 3.

Customer ID	Customer journey
1	$\langle 1A, 1A \rangle$
2	$\langle 1A, 2A \rangle$
3	$\langle 3A, 3A \rangle$

Table 3.3: Data for aggregation based representative journeys example

3.5 Predictions

Predictions are conducted with regard to the next event which occurs in the customer journey. These predictions are based on the features which can be extracted from past events in the customer journey. Note that the predictions also include the option of predicting a customer journey to end. This is primarily interesting for journeys which do not have a set ending point, as in that case one predicts at which point the customer loses interest in continuing their journey. This is an avenue usually left unexplored for recommendation systems, where the main focus lies on monitoring the events actually logged by the system. This knowledge of a customer losing interest can however be very useful, as it can for example be used as an indicator that a special offer should be sent to a customer to rekindle their interest.

Predictions should be conducted using an algorithm that allows for multiple options to be returned with a certain likelihood. The reason for this is that often there is a vast range of options

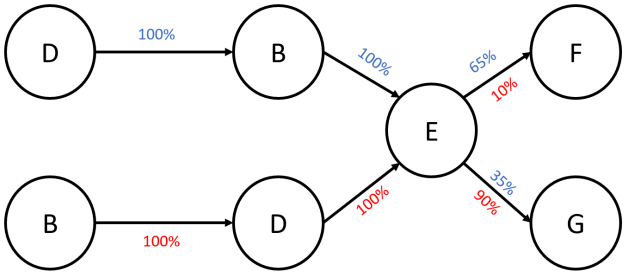


Figure 3.4: Example of order influencing event probability.

in the available paths which lie inside the customer journey. In such cases only providing a single option can lead to poor predictive qualities. It can then be reasonable and valuable to take a larger number of predictions into account which all have a relatively high potential of being useful as opposed to using only a single one. This will decrease the probability that the prediction does not contain any items which the user is not interested in.

Based on these conditions, the way in which OARA does predictions is delved into deeper. As the Order Aware part of the name indicates, the order in which the events have occurred inside the customer journey is taken into account here. This means that the past is not considered to be a bag of unordered events such as for example in OCuLaR [3]. This added structure improves distinguishability of the information used for predictions based on the assumption that customers who have followed the same trail in their CJ have a high likelihood of taking similar actions in the future as well.

An example of how taking the order into account can help is given in Figure 3.4. Here the order in which actions B and D are conducted has a high influence on what occurs after event E. If for example $\langle B, D, E\rangle$ is observed then the next event is F with a probability of 65%, while observing $\langle D, B, E\rangle$ lowers this to only 10%. Here the order clearly influences future choices, as without the order one would only observe that events B, D and E occurred and have a harder time determining if F or G will follow.

Algorithm 1 ObtainPredictors

Input: Training customer journeys n
Output: Predictors $predictorsArray$

```
Initialize sequencesArray, featuresArray,
outcomesArray, predictorsArray
for i = 0 to len(n) do
    for j = 0 to len(sequence(i)) − 1 do
        presequence = first j events of sequence(i)
        if presequence not in sequencesArray then
            Add presequence to sequenesArray
        end if
        Obtain features(presequence) and add to featuresArray
        Add nextEvent(presequence) to outcomesArray
    end for
end for
for seq in sequencesArray do
    Fit predictor to featuresArray(seq) and outcomesArray(seq)
    Add predictor to predictorsArray
end for
return predictorsArray
```

To properly do predictions for any sequence observed in the CJs, OARA employs predictors for each of these sequences. One should keep in mind that this process is done for each of the RFM groups, and that there are sequences which are present in multiple RFM groups. The reason for not using a more general predictor is that the outcomes can be significantly different for a sequence based on the RFM group in which it is observed. Keeping this in mind the predictors need to be trained for all observed sequences before proper predictions can be conducted for new customer journeys.

This process is described in pseudocode in Algorithm 1. Here the sequence is the current path of a customer journey, e.g. $\langle A, B, C \rangle$. In that case the length of the sequence is 3, and the for loop from Line 3 to Line 10 gets executed twice. Two presequences are identified at Line 4, namely $\langle A \rangle$ and $\langle A, B \rangle$. First, it is checked if these presequences were already observed in Lines 5-7, as an array of all presequences is required for later use. Afterwards, in Line 8 the features are extracted based on the data available at these points of the journey. Similarly, Line 9 indicates that the event observed afterwards is stored as well, which is used for training the predictor and during the evaluation as the ground truth. Once all customer journeys have been checked in this manner, Lines 12-14 show that a predictor is trained on the features and outcomes of the sequences in the base customer data. When the journeys contain a large number of events which are very varied, it can be useful to not take into account the longer subsequences. This should be done when there is too little training information available for them to properly train the classifiers. Based on preliminary tests in these cases performance will increase if the subsequence is decreased in size.

Algorithm 2 OARA Prediction method

Input: Predictors P, Customer journeys n
Output: Predictions $predictionsArray$
1: Initialize $sequencesArray, featuresArray,$ $predictionsArray$
2: **for** $i = 0$ to $len(n)$ **do**
3: Add $sequence(i)$ to $sequencesArray$
4: Obtain $features(i)$ and add to $featuresArray$
5: **end for**
6: **for** $j = 0$ to $len(sequencesArray)$ **do**
7: Obtain $prediction(j)$ based on predictor $P[sequencesArray[j]]$ using $featuresArray[j]$ and add $prediction(j)$ to $predictionsArray$
8: **end for**
9: **return** $predictionsArray$

Once these predictors are obtained predictions can be conducted on new samples, which is described in Algorithm 2. Here the loop from Line 2 to Line 5 indicates that the current sequence of the new customer journeys is obtained as well as their features. Afterwards the actual prediction is conducted for each of the sequences in the loop from Line 6 to Line 8. Here the prediction is conducted based on the previously learned predictor, which was tailored towards the same sequence. This gives a prediction that is optimized specifically for the order of events observed. The top X predictions with regard to what is most likely to be the next step in the CJ are then retrieved. X is a configurable amount based on how many possibilities one wants to consider. If customers are believed to be receptive of recommendations which are slightly out of their comfort zone, X can be set relatively high. Alternatively, in cases where the customers are very conservative with respect to their choices, X should be set relatively low. Multiple predictions are then likely to include involve actions which they are not eager to take.

The optimization of predictors is a tough task in general, and the ones created here follow this trend. Two generic issues were observed to be present relatively frequently in the CJ data. The first of these issues is imbalanced data. For data to be considered imbalanced, the distribution of possible upcoming actions is heavily skewed towards a small subset of majority classes. This causes there to be an insufficient number of examples for the machine learning algorithms to properly

learn how to predict the minority classes. A classic example of this is fraud detection, where 99.9% of the cases are legitimate and the system needs to identify the minority 0.01% cases which are fraudulent. There are a number of ways in which imbalanced data can be dealt with. The most effective one is often to collect additional samples for the minority classes in the field, but this is an option which is not always available due to for example time restrictions. Another alternative, which is always readily available, is the application of under- and oversampling.

Undersampling is a technique which reduces the number of samples from the majority class such that it is more in line with the minority samples. Common undersampling methods include only using cluster centroids [53], and removing Tomek links [54]. Cluster centroids contain the most representative variations of the majority class, which if it feature values would be visualized lie at the center. The removal of Tomek links in the aims to get rid of data that has feature values which lie between values already observed in other members of the class as these samples do not add much distinguishing information to the majority class. So if for example only a single feature exist, and for it values 0, 0.5 and 1 are observed, then 0.5 would be removed as it lies between 0 and 1. Oversampling serves as the counterpart of undersampling in that it aims to increase the sample size for the minority classes. Two common ways in which this is done is by applying the Synthetic Minority Over-Sampling(SMOTE) technique [55] or by use of the Adaptive Synthetic Sampling Approach for Imbalanced Learning method [56]. Both create new samples for the minority classes based on feature values which lie closely to the feature values observed in the existing samples.

A second issue which can be encountered is that there is a lack of information during the early stages of the CJ. In that case the customer has not taken many actions yet which can be used to distinguish between them. In these scenarios multiple CJs which are the same may not lead to the same next action, which is problematic for the machine learning algorithms. These algorithms lack a proper indication on which action to predict in these cases due to the lack of distinguishing features. The main solution to this issue is to try and find additional features which would indicate differences in CJs already at the early stages, so that they may help differentiate between the journeys. It should however always be taken into account that the early stages of the CJ will in many cases be the most difficult ones to predict, and as such that a slight decrease in predictive quality at this point compared to the rest of the CJ may be unavoidable.

3.6 Recommendations

The recommendations for future actions in the CJs consist of two main steps. First, there is the process of obtaining the distances between the CJs and the previously created RCJs to see how much they differ. Afterwards, a combination of these derived distances as well as the previously obtained predictions are used to determine the actual recommendations. The general outline of how the recommendations using OARA are done is given in the pseudocode of Algorithm 3, where the split between the two parts lies at Line 9. The loop from Line 4 to Line 8 indicates that the distances between the sequence used during predictions and the RCJs are obtained, while the loop from Line 11 to Line 19 shows how the recommendations are obtained. Of main interest here is Lines 14 and 15, where certain conditions should be met for a recommendation to be valid. How this is done exactly is covered in the following subsections.

3.6.1 Obtaining customer journey distances

As was mentioned in Subsection 3.4, recommendations should be in line with previously observed behaviour from the users. Therefore a distance measure is required to determine how much the CJ of a certain users differs from each of the representative CJs on which the recommended actions will be based. The formal definition of a distance metric sets the following conditions for it to be valid [57], where x, y, z are some measurable elements:

1. **Non-negativity**: The distance between x and y can never be less than 0.
2. **Symmetry**: The distance from x to y is the same as the distance from y to x.

3. **Triangle Inequality**: The distance from x to z can never be higher than the sum of the distance from x to y and the distance from y to z.

4. **Identity of Indiscernibles**: The distance from x to y can only be 0 if $x = y$.

What distance measure is most appropriate depends on the scenario in which the recommendations are needed. In this thesis two distance measures are showcased which are suitable in different scenarios.

Algorithm 3 OARA Recommendation method

Input: Representative journeys RCJ, Predictions P, Prediction Sequences PS, Conditions C
Output: Recommendations $recommendationsArray$

```
Initialize distanceArray, recommendationsArray
for i = 0 to len(P) do
    Initialize distancesArray
    /* Loop over all journeys to obtain all distances */
    for j = 0 to len(RCJ) do
        Obtain distance(i)(j) between PS(i) and RCJ(j)
        Append distance(i)(j) to distancesArray
    end for
    Initialize foundRecc = False
    /* Loop over all journeys to obtain the recommendations */
    for k = len(RCJ) to 0 do
        if foundRecc == False then
            currentDist = distancesArray(i)(k)
            if C based on currentDist are met then
                Get recommendation based on P(i) and RJC(k) and add to
                recommendationsArray
                foundRecc = True
            end if
        end if
    end for
end for
return recommendationsArray
```

The distance based on features

For the first scenario the data has a decently sized set of features available on which the predictions are based. In this case the proposed distance measure is partly based on the Sum of Absolute Differences(SAD). The SAD is defined as:

$$SAD = \sum_{i=1}^{n} |CJ(i) - RCJ(i)| \tag{3.2}$$

Here n the number of features used during predictions while $CJ(i)$ and $RCJ(i)$ indicate the value of the i'th feature value of the CJ and RCJ respectively. The SAD would then be calculated for each of the RCJs to know which ones are relatively close, if any.

It is however not the case that only the absolute differences should be taken into account. If this were to be done, features which naturally have higher values will almost always have higher absolute differences as well. This creates a situation where not every feature has the same influence on the distance. To counteract this, every absolute difference is divided based on the average value observed in the RCJs for that feature. This is known as the relative difference, and as such the

	Event 1			Event 2				Event 3	
RCJ ID	A	B	C	A	D	E	F	C	F
1	0	0.5	0.7	0	0.9	0.8	0	0	0
2	1	0	0.3	1	0.5	0	0.6	0	0
3	1	1	0.4	0	0.3	0.7	0	0.7	0.9

Table 3.4: Data for sequence based distance example

metric used here can be called the Sum of Relative Differences (SRD). The SRD is defined as:

$$SRD = \sum_{i=1}^{n} \frac{|CJ(i) - RCJ(i)|}{\frac{\sum_{k=1}^{o} AllRCJ(k)(i)}{o}} \tag{3.3}$$

Where n is once more the number of features, o the number of RCJs and $AllRCJ(k)(i)$ relates to feature i of the k'th RCJ. To give an example, if the average value for two features is 100 and 1, and the actual observed absolute differences are 110 and 1.1, both will have a SRD value of 0.1, since it becomes $(110 - 100)/100 = 0.1$ and $(1.1 - 1)/1 = 0.1$.

The distance based on sequence

In the second scenario the data is relatively sparse and as such does not contain enough information to obtain a sufficiently large range of effective features can be obtained between the CJs. In this case a different distance measure can be used. This measure is based on the sequence of events observed in the CJ up to the point where a recommendation needs to be made. The way that this is calculated is by taking all possible actions into account for each of the events which have occurred. In the CJ which needs a recommendation all executed events get a value of 1. For the RCJ, the average is taken for each of the events. If for example in the same KPI group it is true that $L = \langle A, B\rangle, \langle A\rangle, \langle A, B\rangle$, then for the first event A would have a score of 1 and for the second event B would have a score of 0.67. Based on these values the absolute differences are once more used as a measure of the distance, which means Equation 3.2 can be applied. n would here be equal to the sum of all available types per event, while $CJ(i)$ and $RCJ(i)$ would be i'th event in the collection of all those types per event.

To clarify this further an example is given based on a CJ $\langle A, (D, E), F\rangle$ and the RCJs in Table 3.4. Comparing the CJ to the first RCJ, an SAD value is obtained of $1 + 0.5 + 0.7 + 0 + 0.1 + 0.2 + 0 + 0 + 1 = 3.5$. When compared in the same manner to the second and third RCJ the SAD values are 2.4 and 3.1. So based on the sequence the best matching RCJ would be the second one.

Neither of the two presented distance measures is necessarily superior to the other, as depending on the situation either can be more suitable. If for example all customers would follow the exact same sequence of types at each event up to the point where a recommendation is needed, but major differences exist in the next event based on their gender and income, then using the distance based on the features is more appropriate. Similarly, when all feature values are almost completely the same but different sequences of events were followed which influence the next event, the distance based on the sequence increases in value. As such, the exact distance measure will always need to be tuned to the problem at hand.

3.6.2 Obtaining recommendations

Having obtained the predictions and distances, the final part of the recommendation process can be started, which is determining the actual recommendation to give to the customer. There are a large number of ways in which this can be done based on the information obtained at this point, however, there are two general rules which should be followed:

1. Recommendations are never given based on the lowest ranked RCJ. The reason for this is that pushing the customer towards this path will never lead to KPI maximization.

2. A 'non-action', or finishing the CJ, is never recommended. What is meant by this is that if in one of the RCJs the recommended action is for example the act of not buying any more products since the customers in the RCJ were satisfied at that point, then that action will not be considered.

These two rules on their own are not enough to properly do the recommendations, as they do not give any concrete directions in case both are followed. An example of what can be done for recommendations instead is combining these two rules with a condition where it is only allowed to recommend an action from a RCJ as long as it is in the top 10 most likely next actions according to the predictions. This would however remove the possibility for an action to be recommended which does not occur in the predictions. This can be troublesome both when only a small number of predictions are considered and when new types of products need to be taken into account which did not exist in the training data of the predictors.

As a solution to this issue there are ways of balancing the information from the predictions and RCJs. An example of proposed here, which is also included in Algorithm 3, is to use the distances. 'Distance tiers' can be created which have specific conditions based on how close the CJ is to a RCJ. These conditions can be configured to directly use the events in the RCJs to circumvent the mentioned problem of only basing recommendations on predicted events. To further clarify this an example is presented based on the conditions provided in Table 3.5.

In this example there are 4 tiers, which are based on how close a CJ is compared to a specific RCJ. The customer is scored on each RCJ in this manner, so it could be that a customer is in the 'Best' distance tier for one of the RCJs while they are in the 'Poor' distance tier for a different one. The percentage thresholds and number of tiers which are taken into account can be varied based on what is desired and suitable. The same is of course true for the conditions, which aim to limit the usage of actions from the RCJs and predictions in some way based on how relatively distant the CJ is. In this example whenever the journey matches a RCJ very well, i.e. it is in the shortest 15% of the distances, the two most likely actions from the RCJ are recommended. In the next tier the recommendation consists of all matches between the three most likely predictions that matches with the actions available in RCJ. The 'Decent' tier works in a similar manner except with the top prediction. Finally, if the CJ performs worse than the 85'th percentile in the training data, then no recommendation is given based on that RCJ. The reason for this is that the actions do not line up well enough for it to be reasonable to assume that the customer would follow an action from that RCJ.

Distance Tier	**Distance%**	**Condition**
Best	0-15	Recommend 2 most likely events based on representative journey
Good	16-50	Check for match in top 3 predictions
Decent	51-85	Check for match top predictions
Poor	86-100+	Do not use action from this representative customer journey

Table 3.5: Data for distance tiers example

Once distance tiers are obtained between each of the RCJs and all of the CJs which require a recommendation, the recommended action can be obtained. As can be seen in Line 11 of Algorithm 3 the RCJs are traversed in reverse order, where the highest ranked RCJ leads to the highest KPI value. Starting from the highest ranked journey it is checked which distance-based condition needs to be met for the action in that RCJ to be recommended. If this condition is met then the action from that RCJ indeed becomes the recommended action, while if the condition is not met, the condition gets checked for the second highest rated journey. This continues until one of the conditions is met.

One thing to keep in mind here is that there is the unlikely scenario that none of the conditions can be met for a certain CJ, for example due it always being in the 'Poor' distance tier. Because of this there also needs to be a default course of action in place. An example of this is recommending

based on the highest ranked RCJ. This can be reasonable as none of the possible recommendations seem to fit this customer very well and this allows for at least some sort of advice which can lead to an advancement of the KPI. A different possibility is recommending based on the action conducted in the RCJ which was closest to the current CJ.

Based on this setup for the recommendation, there a number of relations which need to be taken into account when configuring recommendations:

- Recommending more based on the RCJs than the predictions will lead to higher KPI values in theory. The predictions are based on behaviour which is relatively common and likely to occur without intervention, while the actions from the higher ranked RCJs are specifically centered around a certain KPI instead. The main way in which this can be achieved is by shifting the threshold of the distance tier(s) which involves directly recommending based on the RCJ.

- The amount of most likely predictions one allows in the conditions affects the eventual KPI. Allowing for more predictions to be matched with the action in the RCJ will firstly lead to more matching actions in general, and secondly to more possible matches in the higher ranked RCJs theoretically getting them closer to the highest possible KPI. This however comes at the cost of the recommendations becoming less closely related to the previous behaviour of the customer. This should be considered carefully as a poor recommendation due to over-enthusiasm on KPI maximization may have an adverse effect due to a poor match between the recommendation and the taste of the customer.

- Similarly to the previous point, the chosen default action also influences the potential KPI. A scenario where people are likely to follow any recommendation will incentivize having the default action coming from a higher ranked RCJ, whereas a scenario where people are less open minded will favour the use of a RCJ that is either very common or that is relatively close to that of the customer instead.

All in all the recommendation process involves a number of factors which can be configured to the situations which are encountered. The main issue lies in finding a balance between maximizing the KPI while also making sure that the recommendation makes sense for the customer, and an example of how this can be achieved is given in the Case Study chapter.

3.7 Updates to OARA

As more information from the customers becomes available over time, there are certain parts of the system which may become outdated unless they are updated. To counteract this and make sure that the provided recommendations are in line with recent customer behaviour, updates can be conducted on the RCJs. Note that these updates are most often needed in scenarios where data comes in rapidly, e.g. a streaming setting.

The RCJs taken into account need to be changed based on the new behaviour to make sure that recommendations remain relevant. It should be noted that updates to the RCJs can only be conducted once one is reasonably certain that the CJ has ended. This is similar to the process of selecting the baseline customer information, and as such CJs should only be used for updates once a certain amount of time has passed. The updates to the RCJs here do not require a complete recalculation of the RCJs as they can be updated based on the weights. As such the approach is still in line with the concept of a "one pass algorithm" [58]. The required updates for the subset-based and aggregated RCJs are different, and will be discussed separately.

3.7.1 Updating subset-based representative journeys

For the SRCJs the weights of the CJ variants get updated based on the variant to which the newly finished journeys belong. This may change which CJ variants are most commonly observed. If this is the case, the variants which are included in the subset are changed to accommodate this.

Continuing on from the previous example involving the data in Table 3.2, assume that five new CJs finished, which resulted in the data in Table 3.6.

Customer ID	**Customer Journey**	KPI
1	$\langle A, A \rangle$	7
2	$\langle A, B \rangle$	5
3	$\langle B, C \rangle$	4
4	$\langle A, C \rangle$	6
5	$\langle B, C \rangle$	4
6	$\langle A, A \rangle$	7
7	$\langle A, A \rangle$	7
8	$\langle A, C \rangle$	6
9	$\langle D, D \rangle$	10
10	$\langle A, C \rangle$	6

Table 3.6: Data for subset-based representative journey update example

If the amount of allowed variants still remains at 2 then both of the variants considered previously are now not in the subset any more. The 2 variants which would now make up the SRCJs are:

1. $\langle A, A \rangle$, which has weight 3 and a KPI value of 7.
2. $\langle A, C \rangle$, which has weight 3 and a KPI value of 6.

This process is repeated whenever a new batch of finished CJs comes in to make sure that the SRCJs remain relevant. It should be noted that in a situation where new trends heavily influence the actions in a customer journey, that the newly observed CJs can be given additional weight such that they would rise to the position of a RCJ more swiftly. This makes sure that customers do not receive recommendations based on items which have already gone out of fashion. This increase in weight for new CJs can also be applied to the ARCJs, which will now be discussed.

3.7.2 Updating aggregated representative journeys

Like the SRCJs, the ARCJs allow for updates on the newly finished customer journeys. For the ARCJs the averages get updated based on the new CJs which are in the same KPI based group as them. The influence of the new customer journeys on the values in the RCJs is based on the weight of the RCJ to make sure that the influence is proportional compared to that of the previous customer journeys.

Based on the data in Table 3.3 the ARCJ had 1.67A for Action 1 and 2A for Action 2. Now assume that there is a new journey $\langle 1A, 5A \rangle$ which finished and is in the same KPI based group. The situation shown in Table 3.7 would be what results. The values in the new aggregated journey are calculated as follows: for Action 1 it becomes $(3 * 1.67A + A)/4 = 1.51A$ and for Action 2 it is $(3 * 2A + 5A)/4 = 2.75A$.

Customer ID	**Customer Journey**	**Weight**
Old aggregated	$\langle 1.67A, 2A \rangle$	3
New journey	$\langle 1A, 5A \rangle$	1
New aggregated	$\langle 1.51A, 2.75A \rangle$	4

Table 3.7: Data for aggregation based representative journeys update example

As such the recommended amounts and types get updated just like in the subset-based counterpart. Therefore it has been shown that no matter which type of RCJS used the journeys can be updated based on newly observed behaviour.

3.8 Evaluations

The evaluation is conducted on two fronts, which are the same as during the preprocessing, i.e. the predictions and recommendations. The goal of these concepts is different, as predictions aim to properly predict the next event of the customer's journey based on their past actions, while recommendations aim to turn this most likely event into an event which maximizes the KPI. Since the goals are different, evaluations need to be done in a different manner as well. First, the evaluation of the predictions will be covered, and the recommendations are considered afterwards.

3.8.1 Evaluating predictions

The evaluation of the predictions is mostly done through well known metrics which are based on a confusion matrix. An example of such a matrix is shown in Figure 3.5. Note that in this example there would only be two possible future actions in the CJ, which are here referred to as A and B. TP stands for True Positives, which are cases which were predicted to do action A and where this was the case. TN stands for True Negatives, which indicate that a case was predicted to follow action B, which was correct. FN stands for False Negative and indicates that a case was predicted to do action B while they did action A. Lastly there are FP, False Positives, which act as a counterpart in that a case was predicted to do action A while they did action B.

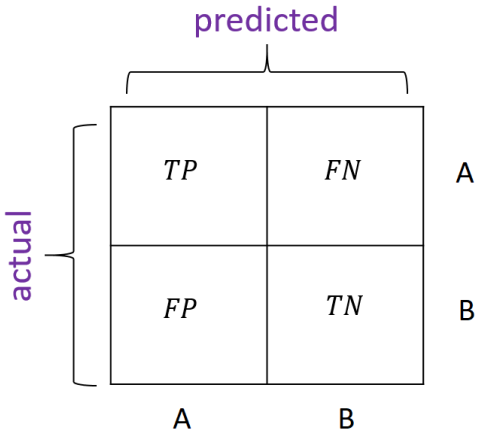


Figure 3.5: General setup of a confusion matrix.

Based on the values observed in the confusion matrix, several metrics can be computed. The ones listed here are relevant to the evaluation of the case study but are by no means an exhaustive collection of all metrics which can be relevant.

Recall when based on the confusion matrix has the following formula:

$$Recall = TP/(TP + FN) \tag{3.4}$$

This metric aims to make sure all cases of a certain class are identified, while disregarding how many cases need to be misclassified for that to be the case. If for example all predictions were to come from a single class, then all elements from that class are identified and as such recall becomes 1.

Precision when based on the confusion matrix has the following formula:

$$Precision = TP/(TP + FP) \tag{3.5}$$

This metric is primarily focused on making sure that there is no accidental misclassification on elements of this type, while not paying attention to how many of the elements of this type are

identified. Precision would for example be 1 even if only a single element in the class was correctly predicted to be from that class while every other element, including all other elements from that class, get predicted to be from a different class.

There is however a potential issue with precision in cases where multiple predictions are considered relevant, which has been showcased to be the case during the explanation of the conditions for the recommendations. The issue lies in the fact that precision punishes additional predictions when the correct prediction has already been conducted. To amend this Average Precision can be used, which only updates when a change in recall is observed. The formula then is:

$$AP = \sum_{i=1}^{n} Precision(i)\Delta Recall(i) \tag{3.6}$$

The Δ symbol here denotes a change in Recall, and as such could also be seen as a Boolean which checks if the $i-1$'th Recall value was the same as $Recall(i)$. If it is, it acts as a 0 such that the sum is not increased, while if it is different it becomes a 1 and the sum does increase. Furthermore, it is important to note that these AP values are based on the user, and not on the types of actions. To give an example, consider the action $\langle(B, C)\rangle$ as the ground truth while the top 3 most likely predictions were A, B and C. Then AP at prediction 1 is 0 as the prediction is wrong so $AP = 0 * 0$. At prediction 2 the prediction is correct resulting in an increase in recall of 0.5 and precision of 0.5 as well, so $AP = 0 * 0 + 0.5 * 0.5 = 0.25$. Finally at the third prediction which is also correct it becomes $AP = 0 * 0 + 0.5 * 0.5 + 0.67 * 0.5 = 0.585$. The AP values are then calculated for all the journeys to get an idea of the overall performance. The average of the sum of all the AP values is called the MAP which follows the formula:

$$MAP = \sum_{i=1}^{n} AP(i)/n \tag{3.7}$$

The F1-score is a metric which aims to combine the viewpoints of precision and recall into a single, more balanced metric. It is normally defined as $2*(Precision*Recall)/(Precision+Recall)$,

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{3.8}$$

though it is adapted here to

$$MAF1 = 2 * \frac{MAP*Recall}{MAP+Recall} \tag{3.9}$$

to take care of the issue with the multiple predictions. MAF1 stands for the Mean Averaged F1 score indicating the use of MAP instead of regular Precision. As one can see in the formulas precision and recall are of equal importance here. Interpretation of the score is slightly difficult as the only real guideline is that a higher $MAF1$-score is better, and there is no cut-off point which determines if the $MAF1$-score is good or bad. Inherently the $MAF1$-score does not take into account the True Negatives into account, which is a flaw of the metric.

Usage of these measures aims to give a relatively balanced idea on how well the predictions perform on different fronts. If there are certain metrics which are more important in a certain scenario then the classifiers can be configured to specifically increase that metric. There is no general metric here which is more significant to the CJs, and if the domain also does not impose any clear favourites then the best course of action is likely to maximize based on the $MAF1$ score. The reason for this is that it gives a good impression on how well a predictor is able to perform due to it taking both precision and recall into account.

3.8.2 Evaluating recommendations

Similar to the predictions, the evaluation of the recommendations can be done in a number of ways. It should however be noted that the metrics used for evaluating the predictions are in many cases not going to fit the recommendations, since as previously stated their goals are different.

With that in mind, there are still a number of factors which influence how recommendations can be evaluated.

Firstly there is the distinction between explicit and implicit feedback given from the users with regard to how much they liked a certain action. Explicit feedback involves customers explicitly providing their opinion on something by for example rating it between 1 and 5 stars, or by means of a thumbs up/down button. This type of feedback allows one to know the feelings a customer experienced without too many assumptions. This increases the amount of trust one can place in the feedback of the users feelings during evaluation. The issue however lies in the availability of such data. Especially within the context of CJs, which is focussed on the actions naturally taken by the users, this will not be accessible in many cases. This is where implicit feedback comes into play as a solution.

Implicit feedback indirectly reflect the users opinion by observing user behaviour [59]. An example of it is the purchasing history, where someone may have bought many books from the same author in the past. This usually indicates that there is a relatively high chance that the customer may buy another of their books in the future. The main advantage of implicit feedback is that it is always available, however, there are some prime characteristics which set it apart from explicit feedback [60]:

- There is no negative feedback. Users will not perform actions which are not to their liking, and as such only actions which they enjoy show up in the data. As such it is not known how heavily disliked the non-observed actions are.
- Implicit feedback is inherently noisy, as the true motives behind the actions can only be guessed.
- Explicit feedback measures preference as ratings are done on some scale, while implicit feedback measures confidence as a higher value purely means that the user has acted more often.
- Evaluation of implicit feedback requires a custom measure to deal with aspects such as the availability to take actions and how to deal with two actions which are only available during the same time period.

As implicit feedback will in most cases be required in the evaluation of recommendations for the CJs, these characteristics should be carefully taken into account for as far as this is possible. However, most of them are inherent flaws of implicit feedback which cannot be addressed. This makes implicit feedback slightly less reliable when compared to explicit feedback. It should be noted that characteristics which fall under the final point can already be taken care of outside of metric scores. An example of this is availability of new products which can be accounted for in OARA by means of updates to the RCJs.

Aside from explicit and implicit feedback, another aspect in which the evaluations can be split is based on if the evaluation was conducted in an online or offline experimental setting. In an online setting actual users interact with the system and receive the recommendations. This allows for observations on the actual reactions users have to these recommendations, and as such one can be certain how the recommendation affects the customers in their journeys. The most well-known example of online testing is A/B-testing, where independently and with equal probability a user is either assigned a recommendation or not. This allows for the benefit of the recommendations to be observed based on the outcomes [61], which in this case refers to the actions taken by the users and how they increase the KPI.

In an offline setting, this real-time interaction is not present and instead previously collected interaction data is used similar to how a train and test set in a prediction setting. In such an offline setting a part of the available customer data is not used while training the recommender, and the assessment of the quality of the recommender is based on how well the recommended actions fit the customer data previously left out. Offline experiments are generally easier to conduct, however, there are some issues regarding them. First, they are mostly focused on a narrow set of questions such as the predictive power of algorithms. Secondly, it is impossible to directly measure

the influence of the recommender on the user behaviour in such a setting since it is not actually deployed [8].

Based on these possible configurations, the most optimal setting would be one where explicit feedback and online evaluations are available, as in that case the most accurate information is obtained. As was however pointed out this is not always possible, especially due to explicit feedback being hard to come by for CJs. In such cases, alternative evaluation methods are to be used which still allow for the recommendations to be evaluated.

An example of a metric which works in an offline setting with implicit feedback that aims to maximize a customizable KPI is introduced here. It follows the following formula:

$$TotalKPI = \sum_{i=1}^{n} KPI(recc(i)) \tag{3.10}$$

Here n is the number of customers which have been recommended a next step in the journey, and the KPI is calculated based on the actions recommended by a specific recommender system. Note that this operates under the restriction that a KPI should be used which can be calculated at any point in the customer journey, much like the dynamic CLV value calculation showcased in Figure 2.5. If this is adhered to, then *TotalKPI* allows for an estimation of the recommendation's effect on this KPI under the assumption that the customers always follow the recommendation. This assumption is different from what one can expect to see in real life, and is mostly in place due to the lack of any prior research on how often recommendations are followed up on by customers. In case one wishes to be more realistic, then one can for example assume that only half of the recommended events are followed by the customers. For the remaining cases one could then take the KPI from the ground truth.

Keeping this in mind it is possible to obtain $TotalKPI$ for as many recommendations as needed to compare how different configuration for the recommendations affect these differences. By changing the configurations in this way the configuration can be found which increases the KPI the most, which could then be used in a real life setting afterwards. It also allows for a comparison between different recommender systems to see which one leads to the highest theoretical KPI value.

To give an example consider the CEI discussed in Chapter II for a set of Twitter users. In this case the CEI is based on observable and non-purchasing behaviour in the form of the number of tweets sent per day. Each action a user can take inside Twitter could then influence the chance that they will send tweets by a certain amount, such as for example setting a profile picture or viewing the profiles of other users. Maximizing the number of tweets sent would then involve recommending the actions which increase it the most, while also making sure that they make sense in context. For example, it does not make sense to suggest a user to change their profile picture every day when it is determined that setting a profile picture has a large positive influence on the probability that a tweet will be sent.

Chapter 4

Case study

In this chapter the results of applying the approach on a real life dataset will be presented. First, the details of the setting of the case study will be introduced. Upon finishing this, the application of OARA for this specific setting is given. This includes specifics on the preprocessing, prediction and recommendation processes. Afterwards, an evaluation of the prediction and recommendation tasks is given. Here the performance of OARA is compared to that of both a traditional machine learning approach in gradient boosted trees as well as a recommender system in the form of OCuLaR. Note that updates to the RCJs are left out due to the lack of any interesting practical details being involved as the selection and distribution of items remains almost completely static in the data of this case study.

4.1 Dataset information

The dataset used for this case study was provided by Signify and is based around one of their products. The product in question requires an initial purchase of a base product to which upgrades can be attached in the future. These further purchases are entirely optional and conducted solely based on the interest the customer has in the product. In this dataset there is no clear end to the journey, as a customer could theoretically always keep buying additional items. A real life example which would fit such a scenario is the purchase of a laptop, where the customer can additionally buy items such a mouse, a carrying bag, or a new battery afterwards. One thing to note here is that the data was anonymized, and as such customers cannot be identified nor the actual products which were bought. As such a limitation of this dataset is that inferences based on domain knowledge are relatively difficult here.

Taking this into account, the following main statistics apply to the data used during the case study: There are 35060 CJs, which contain a total of 141510 events. These CJs are made up out of 271 possible activities that lead to 9127 different variants. These cases were gathered in the time period from 01-10-2017 until 14-05-2018. In this scenario a relatively large number of the customer journeys end early, with 38% buying only the bare necessities after which the user discovers that the product doesn't suit their preferences well enough. A consequence of this is that almost all of the 9127 variants are then contained in the remaining 62% of the journeys. As such this leads to a situation where in some cases the dominant action is that of a non-upgrade. This brings about the problem of imbalanced data discussed in Section 3.5, which is attempted to be tackled here in Section 4.5.

Attribute name	Data type
Customer ID	String
Connection time	DateTime
Event type	Integer
Product type	String
Product subtype	String
Number of products	Integer
Price	Integer

Table 4.1: Table of the attributes in the case study dataset

The data which was used contains a number of attributes per event, which are listed in Table with their data type. In a more detailed explanation, the attributes are as follows:

- Customer ID: A unique sequence of letters and numbers by which a customer can be identified. Note that a single ID here can relate to other users in the same location which use the product as well.
- Connection time: Date and time at which a product was added to the system.
- Event type: Type of the event, which signifies at which stage of the CJ the customer currently is. This value is unbounded due to the existence of a potentially infinite number of upgrades. Event 1 here signifies the installation of a prerequisite product which is the same for all customers, Event 2 indicates that the starting product was installed, and Event 3 any possibly immediate upgrades installed at the same time as the starting product. Higher numbered event types indicate further, separate upgrades.
- Product type and subtype: The type and subtype of the product which was added at a specific step in the journey. In the dataset used for this thesis the type and subtype have an almost 1 to 1 mapping to one another as there are only very few subtype variants.
- Number of products: The amount of products of a specific (sub)type which were added.
- Price: Each product type falls into a price category, with there being 4 different categories. Categories are used instead of concrete values as pricing can evolve over time, as this makes them easier to use than constantly changing the price based on a change log.

This combination of information allows for the data to be represented as an event log and interpreted as a CJ. As was previously defined, a single event can be described as $T = (c, a, t)$. For this data c = Customer ID, a = (Event type, Product Type), and t = Connection time. The CJ is then built up out of multiple of such events and the event log out of multiple CJs. Note that here a is built up solely from an event type and product type. It is not further specified by means of the number of products, which is a deliberate choice driven by the increase in generality being needed to obtain reasonable amounts of training data for the predictors. In case sufficient data is present, it is encouraged to try and create an event as precise as possible as the increase in information can help distinguish the customer base.

4.2 Exploratory data analysis

In this section some of the most prominent discoveries during the exploratory data analysis are presented. First, to obtain an idea of how streamlined the process usually is a process model was discovered which included 75% of all events and 25% of the paths. The events are here conform with the format described in the previous section. These events and paths were the most common ones observed. The discovered model is shown in Figure 4.1, and is known as a 'spaghetti-like model'. In such a model there are many lines crossing each other making the complete picture

messy and difficult to interpret. The cause of this is a partly due to a large number of events but mostly due to the very large number of possible paths of actions. This indicates that there are very few restrictions regarding which events are allowed to follow up on each other in the CJs. This issue reinforces the proposed actions during preprocessing of segregating the customer base, as well as mining a process model tailor-made for the resulting groups. Both of these allow for the overall behaviour to be less varied, making it more easily interpretable.

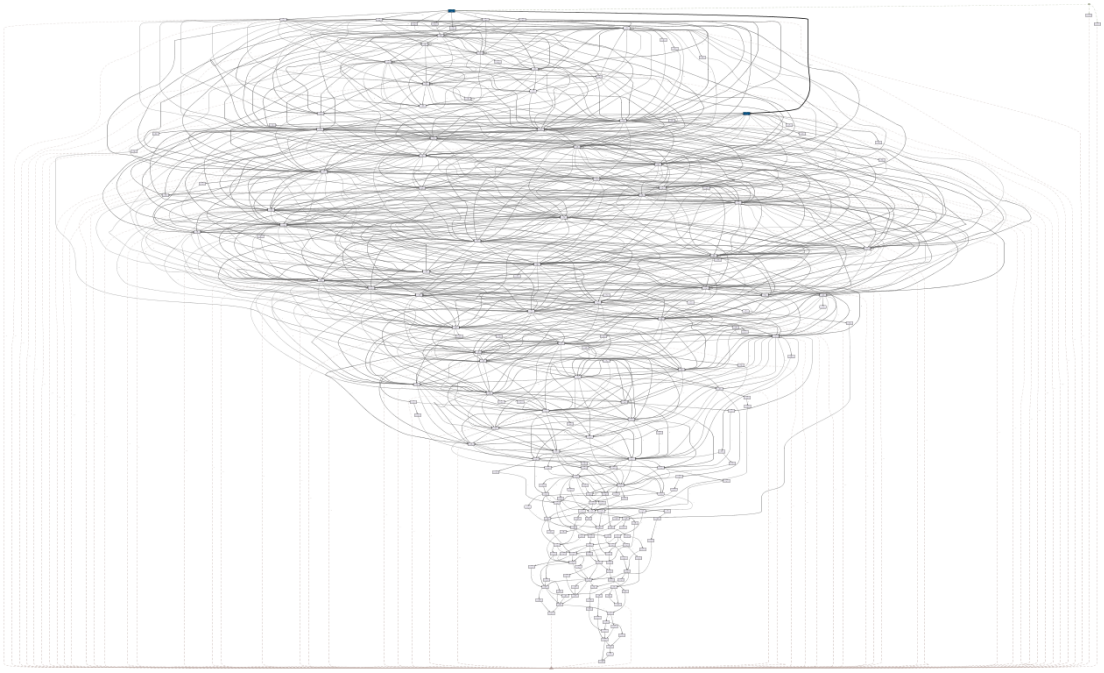


Figure 4.1: Process model including 75% of the events and 25% of the paths.

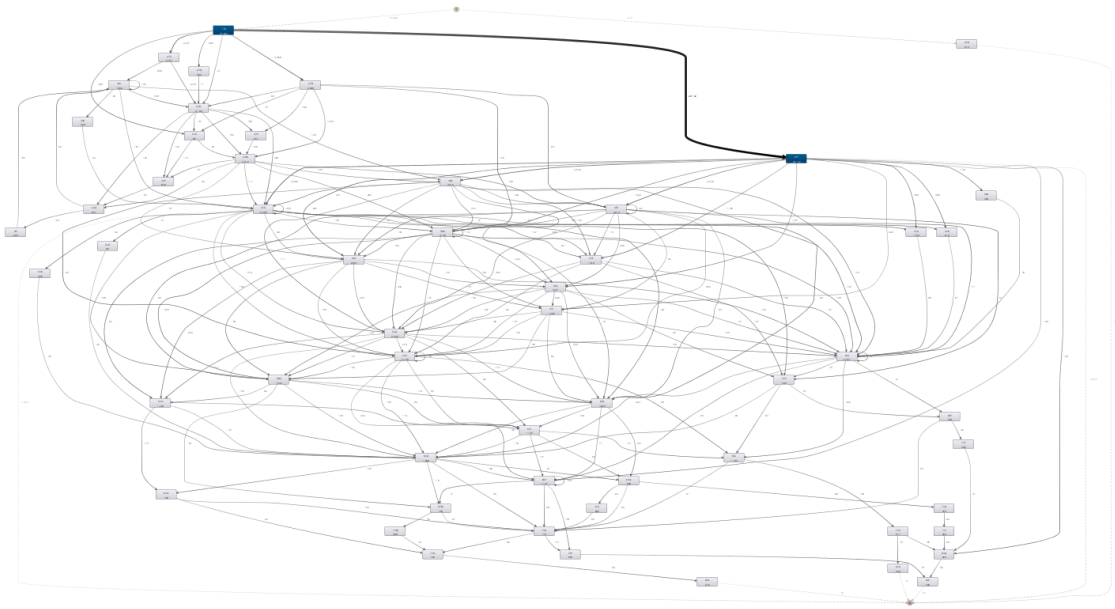


Figure 4.2: Process model including 20% of the events and 5% of the paths.

To however still get an initial impression of the events and paths which are very common, a process model was created based on the most common 20% of the events and 5% of the paths. The resulting model is shown in Figure 4.2. Based on this model it becomes clear that the second event of the journey usually involves a product of type P, whereas R, C and Q are also somewhat popular. Past this second event starts the large amount of diversity that causes Figure 4.1 to be so difficult to read. This is caused by a very large number of events being reachable from each of these second events. As such customers usually do not limit themselves to only a small selection of products based on the initial purchases. The variation still present in Figure 4.2 is mainly caused by events of Event Types 4 and 5. These are the first two upgrades bought upon finalizing the initial setup of the customer. Near the bottom of the model there are also some events which have types 6 and 7, but these are scarcer due to fewer customer journeys reaching this many upgrades. It should be noted that there are higher numbers of event types as well, but these are not observed often enough for them to be included based on the thresholds set for this process model.

Outside of process mining it was checked how certain values developed over time, i.e. if there were some specific trends which might need to be addressed. However, no significant changes in the number of items nor the (sub)types of items bought were observed in the CJs which were included in the dataset. This has as a consequence that older information is, much like newer information, representative of the customer body. Similarly the average length of the CJs was relatively consistent over the period considered in the dataset, remaining closely around 30 days at all times. Note that this average is heavily influenced by a relatively large number of very short CJs.

4.3 Baseline customer information

Based on the characteristics of the dataset it is possible to derive the needed and best preprocessing steps. First, there's the issue that the customer journeys are open-ended. As such an end point needs to be decided. In the exploratory data analysis the elapsed time between two actions was calculated inside each of the CJs, and a good cutoff point was then determined to be the 85th percentile, which resulted in a time period of 98 days. It should be noted that this was based on a larger and slightly older, but in terms of observed customer journeys very similar, dataset. This was done to gain more confidence in what a good cutoff point would be. What this means here is that only journeys which did not see any activity after their final event for at least 98 days are included here. The reason the 85th quantile was taken is to strike a balance between preserving enough data to properly train a model on and prematurely ending a case. The value may appear slightly high compared to the average length of around 30 days in the recent dataset, yet a similar average length was also present in the bigger and older dataset due to the influence of shorter CJs.

The features which are relevant here are slightly scarce. Keeping in mind that no features regarding the order of events are required as they are already taken into account by OARA, the following features were found to have a positive effect on the predictive power:

- Cumulative price
- Cumulative number of products added
- Duration of journey in days
- Cumulative number of upgrades performed

To give a small explanation for the final feature, it can be the case that a single Event type involves upgrades of multiple types, and this feature keeps track of that. Note that there were attempts to use other features as well, such as the day of the week and month, the current season, holiday information, firmware upgrade information, and releases of own-brand and competitor products. None of these features were consistently increasing predictive powers, and as such they were left out.

4.4 Preprocessing

The preprocessing steps will be handled in the same order as during the previous chapter. However, firstly it should be noted that the steps to transform the data from the users devices into an event log format are taken care of by the pipeline in Signify and as such no additional efforts were required to use it. The features described were all quite easy to extract as they did not require an outside information source that needed to be converted to the correct format. This would have been the case if for example the firmware information would have been relevant. For the features here simply the price, number of products, timestamps and number of upgrades observed up to a certain point in the journey can be obtained based on the data naturally included in the CJ here which. As such the mentioned matters here are trivial enough that no further explanation holds any interesting insights.

The next point to take care of is the segregation of the users, which was done according to the RFM values of the CJs in combination with the K-Means++ algorithm. As was pointed out in Chapter 2 splitting the data using this combination has been successful in the past. In total 8 RFM-groups were created here based on if a CJ scored relatively High or Low in the areas of Recency, Frequency and Monetary value. Recency was determined by the number of days observed between the final day in the dataset and the day at which the final event in a CJ occurred, Frequency by the number of distinct Event Types observed, and Monetary value by the cumulative sum of all Price values.

Based on this all combinations based on a relative High or Low score on each of the RFM values are considered. Not all groups have an equal number of cases in them, as can be seen in Table 4.2. The *LLL* group contains the most journeys. This means that there are quite a lot of people who tried out the product that did not find it enough to their liking to continue the journey. The *HLL* is similar but slightly smaller, and only covers the newest journeys which still have a higher potential to change to a different group in the near future. There are also quite a lot of samples available from the *HHH* group, which are the best customers. These customers have bought frequently in the past while the last purchase was also relatively recent, indicating lingering interest in the product. As a final note, the least CJs fall into the groups where frequency is high but monetary is low, as it is requires many purchases of cheap items.

RFM Group:	LLL	HLL	LHL	LLH	HHL	HLH	LHH	HHH
#Customer Journeys	11999	6498	645	2298	1020	2222	2746	7432

Table 4.2: Distribution of the CJs over the RFM groups.

Based on these RFM groups process models can be built and extracted for them to only take the most relevant journeys from these groups. In Figure 4.3 a process model including the most common 75% of the events and 25% of the paths for RFM Group *LLL* is shown. Compared to Figure 4.1 this model is much clearer and easier to understand due to only the cohesive behaviour of this RFM group being taken into account. Furthermore, the *LLL* group mostly includes CJs which are short, as the frequency is low, which also leads to a more compact model. The *HLL*, *LLH* and *HLH* groups all have relatively similar short models.

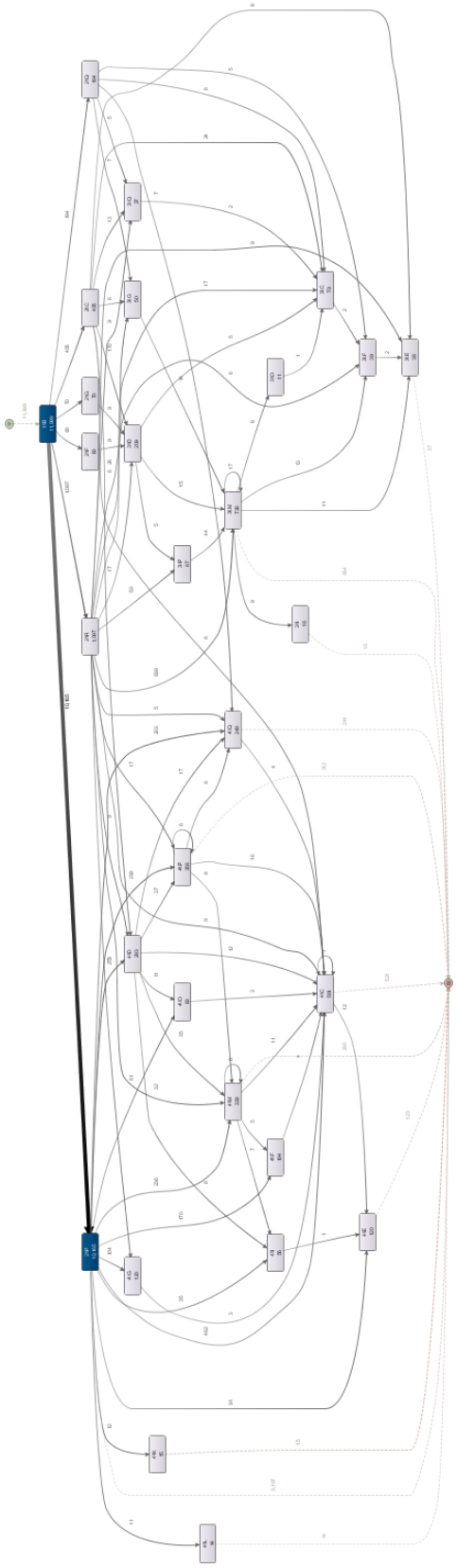


Figure 4.3: Process model of Group*LLL* including 75% of the events and 25% of the paths.

If however an RFM group with relatively high frequency is considered then the model becomes much more expansive and convoluted. An example is given in Figure 4.4, which is the process model including the most common 75% of the events and 25% of the paths of the *HHH* RFM group. This model is a lot more similar to the one in Figure 4.1 due to the journeys being longer. The similarity is further reinforced by the high monetary value indicating that journeys where multiple items are bought at a single event are included too. It should be noted that given an adequate number of samples per journey even such a model can still be useful for predictions. In that case for each of the many paths through the model enough training samples need to exist to properly train the predictors. This would however require for a very large baseline dataset, and in most cases the more appropriate course of action will be to reduce the % of events and paths included in the model to create a more general model. It should be noted that the other RFM groups with high frequency have slightly less complicated models but are still much more expansive than their infrequent counterparts.

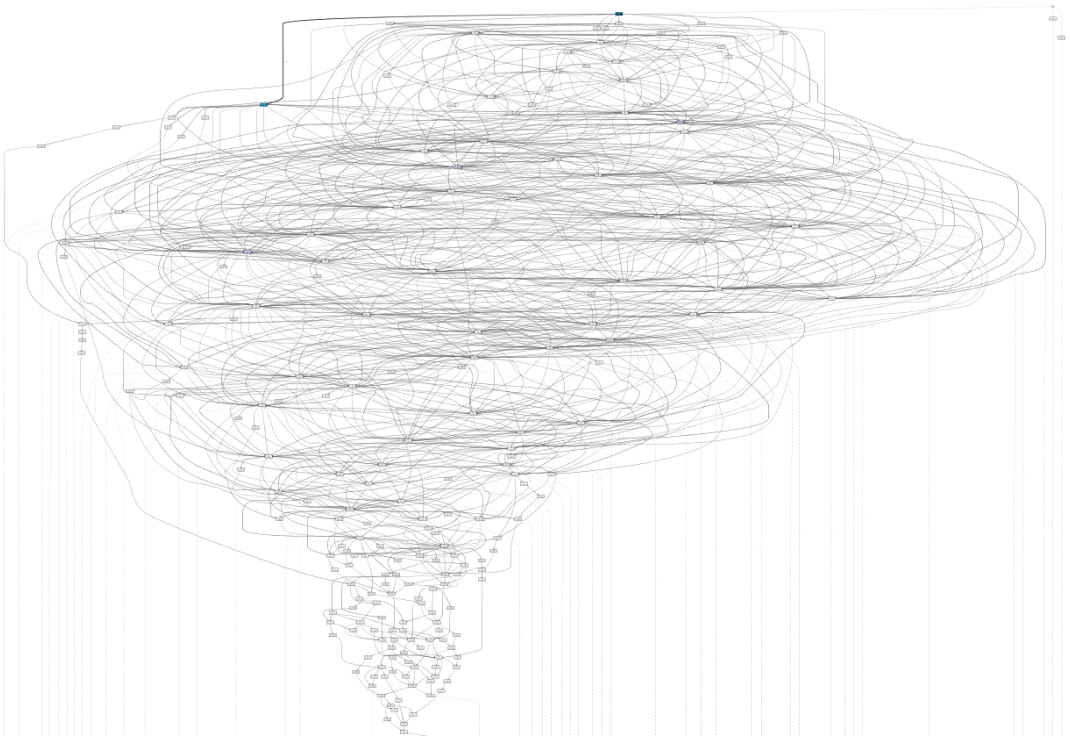


Figure 4.4: Process model of Group*HHH* including 75% of the events and 25% of the paths.

This has covered all preprocessing steps for the predictions, which means only the recommendation steps are left. Firstly any relevant KPIs need to be determined and extracted. In this case these are the CLV and versatility values as they are used during the evaluation. Based on the specification in [41] the CLV is here non-contractual, i.e. customer can always stop buying products whenever they lose interest. Furthermore the behaviour is more erratic than in a contractual setting, and dynamic, which means each action has an effect on the KPI. Furthermore it is viewed here from a monetary perspective, and as such the KPI is based on the revenue obtainable per step in the customer journey according to the Price parameter.

Versatility is concerned with obtaining as many different product types as possible in a single CJ, and is therefore measured by the number of distinct types observed. This can for example be useful in a use case where a customer is determined to remain interested for a longer period of time during their journey when they are able to experience multiple aspects of the system. Both of these metrics are then obtained for each of the CJs in the baseline.

The RCJ format used for this dataset was the ARCJ. The reason for this is the large number

of possible paths highlighted in the discussion of Figure 4.1, which cannot be properly generalized by a reasonable number of CJ variants. More specifically, the coverage threshold was set at 75% and the number of allowed CJ variants at 15. The 15 variants were able to cover only 54.98% of all CJs, with 36.09% coming from the relatively trivial journey $\langle 1B, 2P \rangle$ which does not have any events that can be predicted. As such the use of the ARCJ is a better fit for the data in this case study.

To give an example of how the ARCJs were determined the process is here explained when using the CLV as the KPI of choice. The number of ARCJs used was firstly set 10 to obtain a relatively even distribution of CJs per category. The ranges of CLV values per ARCJ are given in Table 4.3. As most of the journeys are not very long, they do not have time to amass a lot of products and as such the CLV values are slightly similar. The largest differences exist between the journeys which involve a lot of products. These are located in the higher ranked ARCJs where the CLV range is wider. Since the actual calculation of the ARCJs is trivial and has been showcased previously, further details are not included here.

ARCJ	1	2	3	4	5	6	7	8	9	10
CLV Range	0-2	3	4	5	6	7	8-9	10-11	12-17	18-132

Table 4.3: CLV values per ARCJ.

4.5 Predictions

The main idea behind the predictions was already explained by means of Algorithms 1 and 2. The features which can be used by the predictors have been previously introduced, which makes the task that remains here the specification of the used predictor. In this scenario this is a Support Vector Machine(SVM), where it is used for predictions in a non-binary setting much like in [21]. The main reason that this machine learning method was chosen is that it is not necessarily heavily affected by a relatively small sample size, as shown by [62]. Small sample sizes are relatively frequent in this dataset due to there being a lot of different paths available in the CJs. Furthermore SVMs also showed to be among the most promising methods in preliminary tests, and as such this method seemed like a good choice for the use case.

As was previously mentioned, the issue of imbalanced data was present in this data as well. This is partly caused by the RFM groups with low frequency, as there were already a lot of CJs that ended after the fourth Event Type. The ends of CJs also became more common in high frequency RFM groups as later events in them get predicted. An attempt was made at remedying this issue by both underfitting and overfitting the data as well as combining the two. When underfitting was applied to the dataset this was done by only considering the cluster centroids similar adapted from [53]. Overfitting was done according by means of SMOTE [55]. The underfitting and overfitting was then conducted on the CJs belonging to a specific RFM group as a whole, and not based on the CJs that followed a particular presequence.

Using either of these approaches did not significantly affect performance on the test set, although there was a noticeable difference on the training data where scores did slightly decrease due to the prediction task becoming more difficult as there was less of a bias towards the majority class to exploit. This result was not in line with expectations given the fact that these methods have increased performance in similar situations as is showcased in [63]. Two reasons were identified for this. First, the relatively few feature values are at times very similar even between the different classes. In this case both undersampling and oversampling are not really able to increase the distinguishability. Secondly, it was the case for rarer presequences that the feature values in the training data did not cover enough ground such that the examples in the test data could be predicted by the predictor as they were too different. In such a case letting SMOTE create more training samples based on observations in the training data does not effectively help predictions on the test samples. A combination of these factors leads to a situation where undersampling and

oversampling were not able to increase distinguishability of the classes in the imbalanced setting.

Aside from this it should be mentioned that in the cases where frequency was high, a possible issue arose. As was showcased by Figure 4.4, there are a lot of different paths to consider for the later predictions if the entire presequence is taken into account. As such to obtain a reasonable base of training samples on this dataset a smaller presequence is needed. This was here reduced to purely the previous event, as any longer sequence in these RFM groups would already lead to situations where there too few samples to reliably predict on.

Aside from this the SVMs were all trained in a cross-validated manner while optimizing parameters for the methods based on the previously introduced MAF1 metric. Once the predictors were obtained for each of the presequences observed in the data, the trained predictors were used to obtain the top 5 most likely predictions. As will become clear in the next sections not all of them are used during recommendations, as some are included for evaluation purposes.

4.6 Recommendations

Similar to the predictions, the recommendations follow the process described by pseudocode in the corresponding Algorithm 3 very closely. The main things which need to be addressed are the distances, conditions and default action.

As the number of features is relatively low for this dataset, the sequence was used as a means to determine the distance between CJs and RCJs, as this holds more distinguishing power than the features. Aside from the low number of them, as was previously mentioned there are also cases where feature values can be similar between classes. This further reinforces the usage of the sequence based distance for this case study. The manner in which the distances were calculated is the same as described in Section 3.6.1 and as such not further discussed here.

The conditions used were the same as those given in Table 3.5. The first reason for this is that it allows for a fair share of very similar journeys to get a direct recommendation, which makes sure that not only the more commonly observed actions in the CJs were utilized. Secondly, most of the other CJs are given at least a chance to match in some regard with the higher ranked RCJs to increase their similarity to these RCJs. This can lead to them becoming similar enough to them that more, and larger recommendations can come from these prioritized RCJs in the future.

The default action was set to recommending the two most likely actions based on the highest ranked RCJ. The previously made claim that this is a rare event is backed up by the case study data, where at worst in an RFM group it was used in 2.24% of the cases. This was mostly observed in the *LLL* and *HLL* groups where given very specific presequences the only observed next event was the end of the CJ, in which case this was the sole prediction to be made by the predictor. As this is not a valid recommendation, no other choice is available but to use the default recommendation.

4.7 Experimental Evaluation

The predictions and recommendations on the CJs have been conducted in a multitude of ways to facilitate an overview of how different approaches were able to tackle this dataset. The competitors which are compared are OARA, gradient boosting trees [19] and OCuLaR [3]. The reason gradient boosting trees were chosen to represent the traditional methods over any other well known machine learning algorithm is twofold. First, based on preliminary results this method was most promising and as such considered as a valid competitor. Secondly, gradient boosting trees are theoretically able to deal with the unbalanced data issue mentioned earlier. This is caused by one tree, or group of trees, overfitting on part of the data which they specialize in. Since then information from all trees is taken into account, overall a more balanced prediction can be given. OCuLaR was used as it is a recommender which is able to operate under the same circumstances as OARA, while also being a very recently developed state of the art technique that sets a good example of where current techniques stand.

The metrics used for comparison here are the previously introduced *Recall*, MAP and $MAF1$ metrics based on Equations 3.4, 3.7 and 3.9 respectively. This gives an insight both on how well the predictors perform in the recall as in the precision domains, while the $MAF1$ score offers a balance between the two. The areas which are covered during this evaluation are the predictions of the very next event, predictions where the event is allowed to be in the slightly more remote future, performance of predictions when additional context data is available, and finally performance on recommendations of the next event.

In the experimental evaluation the *HLL* and *HHH* RFM groups are explicitly taken into account, while metric scores for the other RFM groups can be found in Appendix A. Groups *HLL* and *HHH* were chosen as they contained the a relatively large number of CJs while also being most interesting from a business perspective. The *HLL* group covers the customers that only recently started their journey and still have to determine if they appreciate the product. As such this group holds a lot of potential value if their interest can be retained. Conversely, the *HHH* group involves the 'best' customers that are currently still interested in the product. These customers have purchased a relatively large number of products, indicating that if relevant products can be recommended to them they are likely to also be interested in those products. Furthermore, these two groups were used as they are representative of one of the big distinguishing factors between the RFM groups. This factor lies in the relative frequency, as low frequency journeys only contain journeys that reach up to the fourth event type. A consequence of this is that the action of stopping the CJ is more frequent during these predictions when compared to the predictions for the high frequency journeys. In these journeys the end of the journeys become more prevalent at the 5th, and predominantly the 6th event type.

4.7.1 Predicting the next event

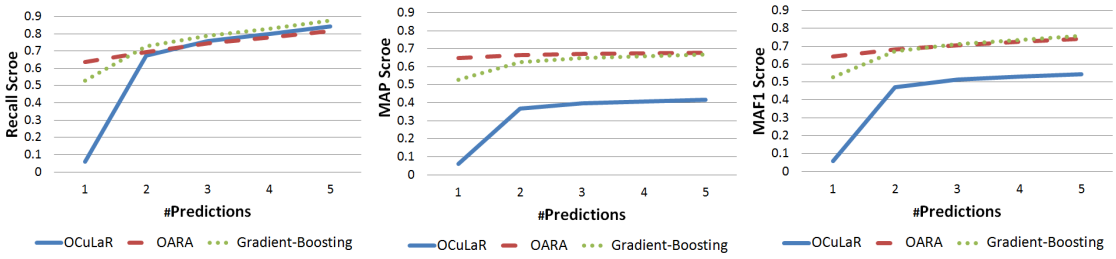
The main objective of the predictions is determining the very next event in the customer journey. For this reason the Recall, MAP and $MAF1$ score has been obtained for the 5 top-most predictions for each of the 3 predictors, which can be found in Figures 4.5-4.8.

HLL Event 4

Figure 4.5 shows the metric scores for the prediction of the 4th event in the *HLL* group. Starting from Subfigure 4.5a, all predictors perform on a similar level once 2 or more predictions are allowed. At the first prediction OARA slightly outperforms gradient-boosting trees, and both of these methods heavily outperform OCuLaR. The cause for this lies in OCuLaR not being able to take feature information into account, which is important for the identification of the relatively large group of stopping journeys, as older CJs in this category are more likely to not continue. This involves the group of customers who relatively recently bought their product who do not return to expand upon it, as they do not have any further interest or are already satisfied with the base product. In terms of MAP Subfigure 4.5b showcases a similar scenario as Subfigure 4.5a, although here the increase in MAP at the threshold of allowing 2 predictions is lessened due to there already having been an erroneous prediction. Overall then Subfigure 4.5c, which showcases the $MAF1$ values, gives a balanced overview based on the previous figures from which it becomes apparent that both OARA and gradient-boosting trees are a valid choice in this situation. OARA has a slight edge if few predictions are required, while gradient-boosting trees are favored if multiple are allowed.

HHH Event 4

Based on the performance in Figure 4.6, all algorithms perform quite poorly here. The performance is especially bad if few predictions are allowed. This is caused by there being a large number of options to choose from in this RFM group, as was shown in Figure 4.4. In this group there is no main path which all journeys follow, and insufficient data is available at this point in the journeys to properly distinguish between them to find the correct prediction. As such both OARA and

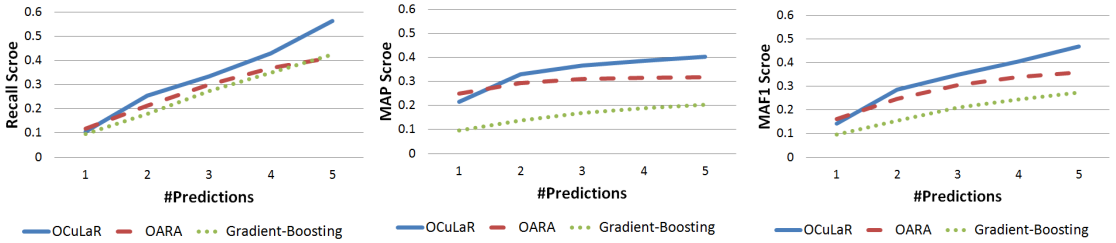


(a) Recall (b) MAP (c) MAF1

Figure 4.5: Metrics for predictions on event 4 of group *HLL*

gradient-boosting trees which take the features into account, as well as OCuLaR which bases it purely on the observed items per customer have large issues during the prediction phase here.

This however does not mean that there is no favorite in this scenario. For recall, as shown in Figure 4.6a all methods perform equally well given a single prediction, and increase in a similar fashion with OCuLaR improving slightly faster than the competitors. For MAP a larger difference can be observed in Figure 4.6b, where OCuLaR and OARA outperform the gradient-boosting trees, and where OCuLaR also rises above OARA as the allowed predictions increased. Overall then, as shown by Figure 4.6c, OCuLaR is able to more effectively gain insights based purely on the bought products in this scenario. OARA is then still able to outperform gradient-boosting trees due to the additional context information from the order of events aiding slightly in the predictions.



(a) Recall (b) MAP (c) MAF1

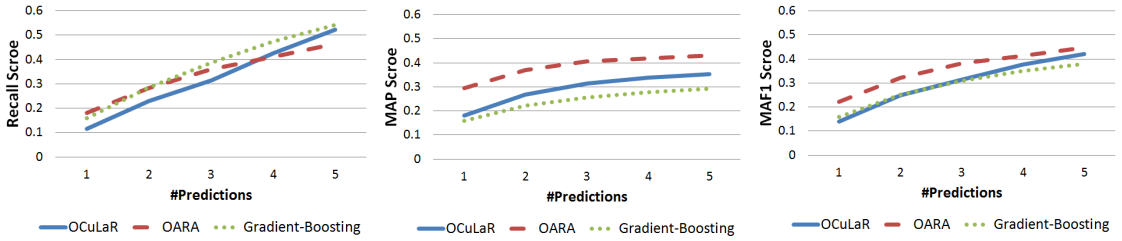
Figure 4.6: Metrics for predictions on event 4 of group *HHH*

HHH Event 5

In terms of overall performance, the values achieved between Figure 4.6 and 4.7 are quite similar for OCuLaR, while OARA and gradient-boosting trees start to perform better. This is due to the additional information that became available due to the additional actions, as well as there being slightly less paths to consider during the prediction of this event. This in turn allows for more training data to be available for the predictors. For recall there are no big differences between the scores of the three methods regardless of the number of allowed predictions, as showcased by Figure 4.7a. The real difference here is in the precision domain shown in Figure 4.7b, where OARA is able to outperform both OCuLaR and gradient boosting trees, showcasing the value the context added by the order of events can have. In terms of general performance the same is true, as the $MAF1$ scores in Figure 4.7c are for any number of predictions the highest when using OARA.

HHH Event 6

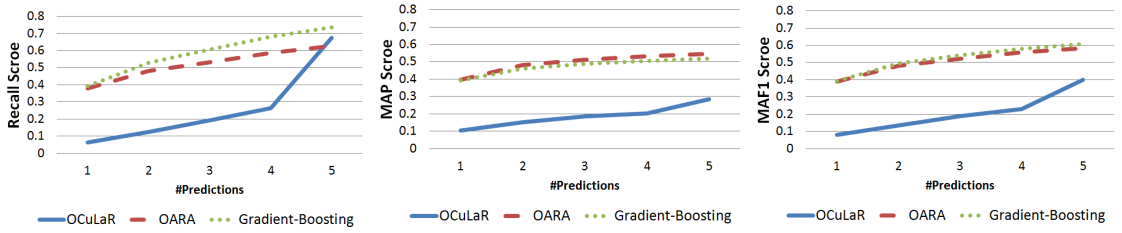
The scores in Figure 4.8 paint a similar situation as those in Figure 4.5 in the sense that OARA and gradient-boosting trees outperform OCuLaR. This is mainly caused by an increase of journeys which are ended that OCuLaR has trouble identifying. This time the increase in recall observable



(a) Recall (b) MAP (c) MAF1

Figure 4.7: Metrics for predictions on event 5 of group *HHH*

in Figure 4.8a also occurs at only the fourth-best prediction instead of the second best. The cause for this as well as the slightly lower recall scores for OARA and gradient-boosting trees is once more the increased diversity when comparing the *HHH* to the *HLL* RFM group. Gradient-boosting trees then are slightly better at adapting to this increase in variety than OARA if multiple predictions are allowed. The development in MAP in Figure 4.8b is identical between OARA and gradient-boosting trees, while being notably worse for OCuLaR due to the many missclassifications. As such also the $MAF1$ scores in Figure 4.8c are significantly worse for OCuLaR here, while OARA and gradient-boosting trees both perform well with a slight edge towards the gradient-boosting trees due to the improvements in recall over multiple predictions.



(a) Recall (b) MAP (c) MAF1

Figure 4.8: Metrics for predictions on event 6 of group *HHH*

Performance on other RFM groups

The *Recall*, MAP and $MAF1$ scores when 1 or 5 predictions are allowed are given for the other RFM groups in Appendix A.1 using OARA, Gradient-Boosing and OCuLaR. The trends observed there mainly mirror what was showcased here. For example in the LHH group, where there are also many products being bought over multiple actions similar to the HHH group, gradient-boosting is once again able to outperform OARA and both perform better than OCuLaR. In particular it seems to be the case that in cases where the monetary value is high, which in this dataset mostly indicates the presence of multiple purchases, OARA seems to perform slightly worse than Gradient-Boosting. This is likely caused by the feature information adding additional value if it is considered as a whole, as opposed to splitting it up like how it is used in the predictions using OARA. As such additional samples may alleviate this issue. OARA is however still able to adapt to such situation better than OCuLaR, which lacks the information to properly predict here.

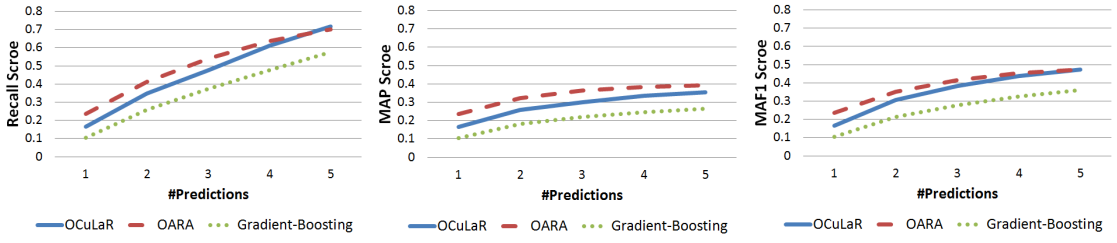
Outside of this, in the remaining groups the performance between OARA and gradient-boosting trees is relatively comparable. When one method reaches high metric values the other is able to follow suit, with some slight differences in which of the two end up reaching higher values. The same is true for OCuLaR in the recall dimension, but it seems to perform slightly worse with regard to MAP. This seems to be caused by it being slightly less refined in scenarios where the journey is still relatively early on, in which case there are not too many products available to

distinguish the different users by. This causes the overall performance as measured by the $MAF1$ score to be slightly lower for this method as well.

4.7.2 Using a span

While most often the prediction needs to be correct immediately, it can also be the case that a prediction is allowed to be correct in the near instead of the immediate future. This is useful when there is a reason to believe that a group of actions will be conducted in the near future without the order being set in stone. An example of this is a user of an online music service who has already bought 4 albums of a single artist. In this case one can be relatively certain they will buy another album of that artist, but not which of the remaining ones. To cope with such scenarios the concept of a span is introduced, which refers to a timespan during which it is allowed for a prediction to be valid. To clarify, if $span = 3$ is present, then if the predicted action shows up either in the next event, the event after that, or the event following that then the prediction is considered to be correct. Furthermore, the predictions in the previous subsection can be interpreted as having a span of 1. Usage of a *span* for sequences of events is not unprecedented and has been used with success in the past in [64], from which is adapted to the current situation.

The effect of using the span for the predictors here is exemplified in Figure 4.9, where metrics are given for the prediction of the 4th event in the *HHH* group when a span of 3 is used. Compared to the scores achieved in Figure 4.6, the recall values in Figure 4.9a are higher for all predictors. This is caused by there being more leniency in when a prediction is considered correct. Especially OARA profits from this as it is now able to outperform OCuLaR up until the point where 5 predictions are allowed. This increase in possibly correct values here has a slightly adverse effect on the MAP when comparing Figure 4.6b and Figure 4.9b. This is due to there being scenarios where additional types of products might be valid predictions. When considering both recall and MAP in the $MAF1$ score in Figure 4.9c, the improvements in recall for OARA allow it to also outperform OCuLaR on this front. As such this experimental evaluation has showcased that in scenarios where predictions do not need to be effective immediately, that OARA can be an effective option.



(a) Recall (b) MAP (c) MAF1

Figure 4.9: Metric scores for the 4th event in group *HHH* with a span of 3.

To further showcase the possible positive effect a span can have on the metrics the recall, MAP and $MAF1$ scores for predictions on the 4th event of the remaining RFM groups can be found in Appendix A.2. Compared to the scores achieved in Appendix A.1 the recall and $MAF1$ scores are once again a lot higher for each of the RFM groups due to the increase in leniency. Big increases are mostly noticeable in the *LLL*, *HLL* and *LLH* groups which is caused by the premature endings in these groups. As such a prediction of the journey ending will always be correct as none of the journeys inside these groups ever reach the 6th event type. Furthermore the range of possible actions is also relatively small in these groups, further slightly increasing the ease of prediction. As such the increases for these groups should be taken with a grain of salt. Furthermore, this is also the reason why the *HHH* group was previously used in the more in-depth showcase. Due to the aforementioned reasons the negative effect on MAP present for the predictions of the 4th event in group *HHH* are also not affecting it as much, leading to the MAP scores being higher

Table 4.4: Comparison of metric scores based on the presence of context data

		HLL-4@1	HLL-4@5	HHH-4@1	HHH-4@5	HHH-5@1	HHH-5@5	HHH-6@1	HHH-6@5
Recall	**No context data**	0.636	0.817	0.119	0.414	0.179	0.465	0.380	0.625
	With context data	0.647	0.844	0.105	0.417	0.241	0.489	0.378	0.636
MAP	**No context data**	0.649	0.677	0.249	0.319	0.294	0.432	0.396	0.544
	With context data	0.655	0.738	0.240	0.415	0.371	0.470	0.387	0.539
MAF1	**No context data**	0.642	0.741	0.161	0.360	0.223	0.447	0.388	0.582
	With context data	0.652	0.783	0.146	0.415	0.291	0.479	0.382	0.583

when the span is set to 3 instead of 1. The other groups also reach higher MAP values this time, as the increase in correctly identified actions is larger than that of the possibly correct values.

4.7.3 Including additional context information

From an intuitive standpoint, it makes sense that to further increase the predictive qualities of a predictor it is helpful to include additional information. This enriches the customer journey by providing an increase in context to the observed events, much like taking the order into account did. To test this hypothesis, an additional dataset was obtained. Due to non-disclosure agreements the details of this additional dataset are not discussed here. The data was employed during predictions in the format of features as well as in the format of additional events. Both had a similar effect on the metrics. However, two issues arose when applying the format of additional events. First, it had an adverse effect on the computation time due to additional models needing to be trained. This is caused by an increase in distinct CJs due to there being more events. Secondly, the additional events led to more situations where distinct CJ variants became so rare that there was insufficient training data to properly facilitate predictions for them. For these reasons the feature format was utilized. Note that since the additional information is not in the format of additional items, that not all existing techniques are able to utilize this supplementary context. OCuLaR for example does not have a natural way of applying the new knowledge during predictions. As such, being able to utilize the additional information is a notable quality of a predictor.

A comparison of the results with and without the added dataset when using OARA for 1 or 5 predictions on the *HLL* and *HHH* groups can be found in Table 4.4. Note that here the span is set to 1 as usual. In terms of recall there do not appear to be too many notable differences between the data with and without context information outside of predictions on the 5th event of group *HHH*. Here the additional context information is quite useful, which is in line with the increase in recall observed when comparing Figures 4.6a and 4.7a. Between these two predictions the additional context information that became available between the 4th and 5th event proved to be valuable to OARA, allowing it to outperform OCuLaR. The additional information introduced here leads to further positive effects. Outside of this it is always the case, also for the other predictions, that additional items can be correctly identified when allowing for multiple predictions when using the new data.

With regard to MAP, there are large improvements for when 5 predictions are allowed outside of the 6th event of the *HHH* group. This signifies that while in many cases there are not too many additional correct predictions, the correct ones are on average obtained in fewer predictions. As such the additional data is a valuable asset if the actions already being identified need to be obtained with more certainty.

Finally, the $MAF1$ scores are in general slightly higher as well. This indicates that general performance has increased due to the additional data. There are however also some small decreases in $MAF1$ which are caused by the models being trained on the new information. This can lead to some incorrect insights if certain relations which happen to be uncommon in the test data happen to be slightly more common in the enriched training data. Outside of this, the additional data introduced during this case study was shown to be mostly effective during the predictions of the earlier events. For later events the effect was lessened, as signified by the negligible changes in metrics for the 6th event of the *HHH* group. Adding even more context information, or using different information on the base data from a different source can then prove useful to improve on the later predictions if this is desirable. As such this experiment has shown that OARA is able

to adapt itself when additional context information becomes available which in turn can lead to improvements of the predictions.

Outcomes for the remaining RFM groups were aggregated into a table in Appendix A.3. The observations there are similar to what was observed for the predictions in the *HLL* and *HHH* groups. Increases in recall for the first prediction like the 5th event of group *HHH* can be observed in for example predictions of the 4th event in groups *LLL* and *LHL*. There are also cases where the additional information does not lead to significant increases such as the for the 4th event in group *LHH*. Furthermore, when five predictions are allowed the recall always increases. Outside of that, the MAP has similar increases for any prediction outside of the 5th event in the *LHL* group where the relatively low number of items to be predicted is not positively affected by the additional knowledge. Due to these increases in both recall and MAP the $MAF1$ scores are in general higher as well, indicating that the used data has an overall positive effect on the prediction process.

4.7.4 Recommendations evaluation on KPIs

As was previously mentioned, the evaluation of the recommendations is done based on how well a KPI can be maximized instead of the regularly used precision based metrics. While an online evaluation would have been preferred due to the increased quality in insights, this was not in the scope of this thesis and as such an offline evaluation was conducted. Furthermore there is no explicit feedback available with regard to how much the customers like their products. As such, the information is here inferred based on implicit feedback. The implicit feedback is based on the types of products which are observed inside a CJ. To compare how well each of the recommenders is able to increase the KPI the previously introduced $TotalKPI$ metric in Equation 3.10 is used.

$TotalKPI$ was calculated for OARA under the positive assumption that all recommendations, which are obtained as described in Section 4.6, are followed. The metric was also obtained for OCuLaR, where it is based on the 2 most likely actions according to this method. The final entity for which the KPI was obtained is the ground truth, which are the actions actually taken by the users without any recommendations affecting them. This was included to see if usage of recommender systems can have any positive effects on the KPI. The first KPI used here was the CLV introduced earlier, which is a KPI that aims to capture how valuable a customer is to an organization measured here by the sum of the prices of observed products.

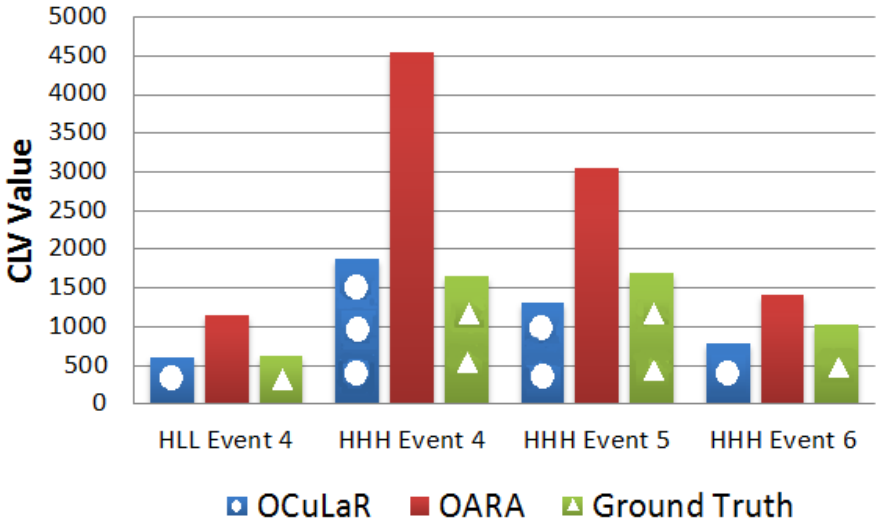


Figure 4.10: Comparison of the CLV values on recommendations.

In Figure 4.10 the CLV values have been calculated for 4 recommendations. This was once

again done for the *HLL* and *HHH* groups. The first two recommendations involving the *HHH* group have higher CLV values for all 3 recommendations methods due to there being more samples to observe for these events. Furthermore, the customers in this category buy a larger number of products than those in the *HLL* group, which further leads to higher CLV values. For each of the 4 recommendations OARA is outperforming the alternatives, which is caused by only OARA optimizing explicitly for the CLV value. There are also times when customers strictly following the OCuLaR recommendation would have a lower CLV value than the ground truth, which is caused by it recommending very heavily based on what is most common. This leads to issues in the *HHH* group, where the buying behaviour is erratic enough that the most common behaviour will not involve proper recommendations for the best of the best customers that heavily increase the KPI.

This does however not mean that the recommendations by OCuLaR are without use, as they should fit the tastes of the customer very well. This could then positively affect their future purchasing behaviour. The same can however be said for OARA, which aims to make sure that the recommendation is in line with the preferences by using the RCJs. Additionally it tries to exert a more direct influence on the KPI by giving a recommendation that immediately increases it as well. Note that if one were to be more pessimistic and assume that only a subset of all recommendations by OARA are actually used, then the relative advantage of OARA would decrease. It would however remain useful since while the sum of CLV values may decrease it is unlikely to sink below the ground truth depending on how the customers are handled for which the recommendation is deemed to fail.

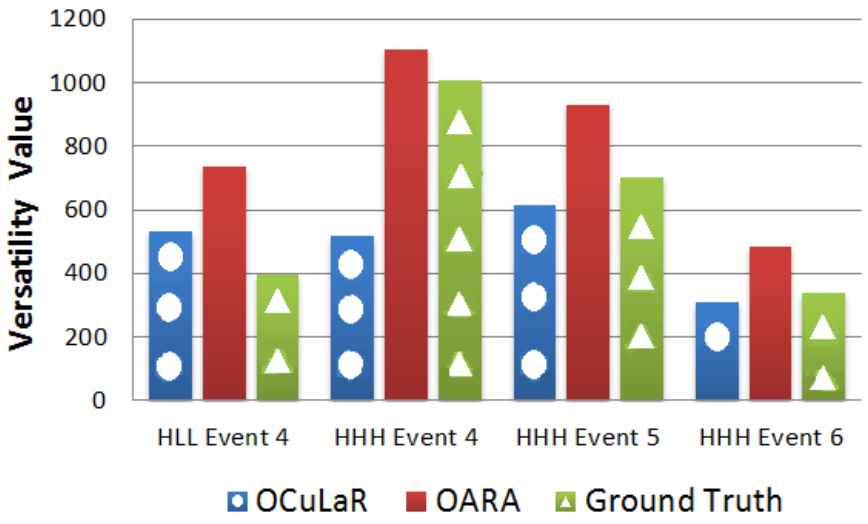


Figure 4.11: Comparison of the versatility values on recommendations.

To showcase that OARA is also able to handle different KPIs, another KPI is used for which $TotalKPI$ is calculated. This other KPI which was also introduced previously and is called versatility, which was measured by the number of different product types a customer acquires in their journey. The highest ranked representative customer journeys are then those which involve the purchase of many different types of products. The result when using this as the KPI when calculating $TotalKPI$ is visualized in Figure 4.11. This is similar to Figure 4.10 in the sense that OARA outperforms both OCuLaR and the ground truth, although this time with slimmer margins. This is caused by it being easier to recommend higher priced products which fit into a CJ than it is to recommend types of products which have not yet been bought. The reason for this is that usually only a smaller subset of of products will suit the tastes of the customer if previously bought products cannot be recommended. OCuLaR also has these issues while, as mentioned previously, not taking KPI maximization into account. This leads to a similar level of

performance as in Figure 4.10. Aside from this, another point of importance in Figure 4.11 is that for the 4th event of group *HHH* the ground truth is very close to the OARA recommendation. The reason for this is that this group of customers naturally buy many different items already. This lessens the need for, and effect of, a recommendation as customers naturally increase their KPI score in this case. As such, there are cases where it may not necessarily be need to take action for improvements on a KPI. On the other hand, recommendations can most certainly be of use if the customer is more conservative in the initial phase of their journey for this KPI and as such those issuing the actual recommendations should carefully consider when they are most useful and needed.

4.8 Case Study Conclusions

Based on the results of the case study, the research questions posed earlier can be answered. First, the answer to the research question 'What are the measurable effects of explicitly taking into account the order in which events occurred during predictions and recommendations?' is given. Regarding the predictions, taking into account the order has a potential positive effect on the metric scores of recall, MAP and $MAF1$ when compared to state of the art alternatives. This is true both when predicting the next action in the immediate and slightly more remote future. Effectiveness was also shown when up to 5 different predictions are allowed in either of those scenarios. As these metrics cover different quality aspects of the predictions, it seems reasonable to assume that there is generally a rise in overall quality when the order is considered during predictions.

Since these predictions play a large role in the recommendation process, this increase also positively affects the quality of the recommendations. A rise in confidence in the predictions therefore directly instigates a rise in the confidence that can be put in the recommendations. Take for example the $MAF1$ score as the general quality metric for the predictions, which increases by 5% when taking the order into account. Any recommendation based on a condition that involves the usage of a prediction should then be 5% more reliable as well. A way to then make the positive effect quantifiable is to consider the increase in metric quality and multiplying it by the fraction of recommendations that use the prediction. However, there is a reason such a metric was not delved deeper into during this thesis. This lies in the flexibility of the conditions used in the recommendations, which make it difficult to properly measure the effect in an offline setting. The conditions determine the importance of the predictions during the recommendations, and as such also the degree to which the effects are properly measurable. Therefore, it is encouraged for future research in this domain to do the evaluation in an online setting to obtain more accurate measurements on the effects on the recommendations. The increase in information is likely to lead to more concretely measurable metrics to properly answer this aspect of the research question.

With regard to the second research question, 'Which concepts and techniques need to be applied such that recommendations can be maximized towards any given KPI?', it was showcased that a combination of factors have made this possible. These factors are the for this thesis introduced RCJs, a distance metric based on the sequence of observed events, a set of conditions which aimed to balance the increase in KPI and the suitability of an action given the previous behaviour of the user, and the CLV and Versatility KPIs. By using all of these concepts the attained KPI values were shown to be higher than that of alternatives. This is an attestation towards the quality of the recommendations, circumventing part of the shortcomings that come with evaluating in an offline setting. Furthermore, this combination of concepts and techniques leaves a lot of freedom for the recommendations to be configured based on the context in which the recommendations are required.

Chapter 5

Conclusions and Future Directions

In this thesis, an approach called OARA was proposed that has proven to allow for predictions and recommendations on datasets which fit the concept of a customer journey. To accomplish this firstly an exploratory data analysis was determined to be of potential value to motivate choices between alternatives at other sections in OARA. The selection of the baseline customer information has then proven to be influenced by the considered timeframe and included information. Upon having selected the baseline information, a multitude of preprocessing steps were presented to be of use mainly for generalization purposes to aid the predictions and recommendations that follow afterwards. Additionally, it was possible to go beyond merely visualizing the journey in a process model by utilizing the model for the prediction and recommendation tasks. Furthermore it was explained in detail how exactly predictions can be done in a manner which takes the order of events into account, and it was also discussed what some of the prominent issues can be during these predictions. The role of the predictions inside OARA was then exhibited during the explanation of the recommendations. In the recommendation process they are combined with the RCJs, a distance measure and a set of conditions to create recommendations that are suitable for a customer and maximize a configurable KPI. The RCJs were shown to be adaptable to new observations. This allows the recommendations to remain relevant even as trends change. Finally, multiple methods were shown that allow for the evaluation of both the predictions and recommendations to determine how well OARA performs. As such, all three contributions mentioned in the Chapter 1 have been addressed in this thesis.

OARA was also applied in a case study to show that the proposed solutions are useful in a real life setting. During the evaluation it became apparent that there are potential positive effects of taking into account the order of events if they exist in a sequential manner. Both in situations where a prediction needs to be accurate in the immediate and slightly more remote future based on a configurable span OARA was shown to achieve good results. It was also made evident that OARA is flexible enough such that it can be further enriched by effective use of an additional source of information to increase predictive qualities. As a final point of the evaluation on the case study the KPI values were shown to be significantly higher if OARA was used to obtain the recommendations, indicating that it can be valuable to explicitly take into account the KPI values during recommendations.

With regard to future work, it is likely to be valuable for researchers to look further into proper evaluation metrics in settings where recommendations are aimed at improving a general KPI. The main shortcoming currently lies in the assumption that the customers will follow recommendations blindly, which was put into place due to there being no prior research on how often customers actually follow the given recommendation. As such a case study of the effectiveness of recommendation could provide a lot of value to the assessment of recommender systems. Additionally, an evaluation in an online setting will also be of value to get a more concrete idea on the quantifiable effect recommendations have on the behaviour in the CJs.

Furthermore, OARA has currently only been employed in a single scenario. As such, deploying it in a different environment will likely lead to further insights on optimizations and generalizations

in areas which were not significant in the scenario of this paper. It is preferable that this scenario includes events that can be conducted in parallel. Another interesting scenario is one where multiple actions are required before a specific different action can be taken, so for example action C is only observed after actions A and B have been taken. The reason these constructs are interesting is that no such patterns existed in the data of the current case study. However, in theory OARA should be able to handle such a sequence of events perfectly given the awareness of past behaviour in the customer journey. As such, a further case study to back up this claim can increase the theoretical power and appeal of OARA.

In a similar vein, additional case studies can be useful. First, the case study in this thesis only applied the ARCJs in practice. As such, a case study based on data that incentivizes the utilization of SRCJs instead of the ARCJs might be interesting to ascertain their viability. Similarly, usage of a dataset where no anonymization is required can be interesting as well. The increase in available context information can potentially lead to a discovery of additional important aspects of the exploratory data analysis. Lastly, it may be compelling to use data where new types of products appear over time. This data could be used to showcase that the OARA is able to adapt to the newer product types to by means of updates.

Bibliography

[1] Krupal S Parikh and Trupti P Shah. Support vector machine–a large margin classifier to diagnose skin illnesses. *Procedia Technology*, 23:369–375, 2016. ix, 7

[2] Mahnoosh Kholghi and Mohammadreza Keyvanpour. An analytical framework for data stream mining techniques based on challenges and requirements. *arXiv preprint arXiv:1105.1950*, 2011. ix, 7, 8

[3] Reinhard Heckel, Michail Vlachos, Thomas Parnell, and Celestine Dünner. Scalable and interpretable product recommendations via overlapping co-clustering. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*, pages 1033–1044. IEEE, 2017. ix, 8, 9, 22, 42

[4] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. Smart cities in europe. *Journal of urban technology*, 18(2):65–82, 2011. 1

[5] Katherine N Lemon and Peter C Verhoef. Understanding customer experience throughout the customer journey. *Journal of Marketing*, 80(6):69–96, 2016. 1

[6] Julia Wolny and Nipawan Charoensuksai. Mapping customer journeys in multichannel decision-making. *Journal of Direct, Data and Digital Marketing Practice*, 15(4):317–326, 2014. 1

[7] Gaël Bernard and Periklis Andritsos. A process mining based model for customer journey mapping. In *Proceedings of the Forum and Doctoral Consortium Papers Presented at the 29th International Conference on Advanced Information Systems Engineering (CAiSE 2017)*, pages 49–56, 2017. 2, 11

[8] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011. 2, 33

[9] Wil Van Der Aalst, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, et al. Process mining manifesto. In *International Conference on Business Process Management*, pages 169–194. Springer, 2011. 5

[10] Boudewijn Frans van Dongen. Process mining and verification, 2007. 5

[11] AJMM Weijters, Wil MP van Der Aalst, and AK Alves De Medeiros. Process mining with the heuristics miner-algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP*, 166:1–34, 2006. 5

[12] Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004. 5

[13] Ronny S Mans, MH Schonenberg, Minseok Song, Wil MP van der Aalst, and Piet JM Bakker. Application of process mining in healthcare–a case study in a dutch hospital. In *International joint conference on biomedical engineering systems and technologies*, pages 425–438. Springer, 2008. 5, 6

[14] Fabrizio Maria Maggi, Marco Montali, Michael Westergaard, and Wil MP Van Der Aalst. Monitoring business constraints with linear temporal logic: An approach based on colored automata. In *International Conference on Business Process Management*, pages 132–147. Springer, 2011. 5

[15] Boudewijn F Van Dongen, Ana Karla A de Medeiros, HMW Verbeek, AJMM Weijters, and Wil MP Van Der Aalst. The prom framework: A new era in process mining tool support. In *International Conference on Application and Theory of Petri Nets*, pages 444–454. Springer, 2005. 6, 18

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 6

[17] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001. 6

[18] Glenn M Fung and Olvi L Mangasarian. Multicategory proximal support vector machine classifiers. *Machine learning*, 59(1-2):77–97, 2005. 6

[19] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 7, 42

[20] Leo Breiman. *Classification and regression trees*. Routledge, 2017. 7

[21] Ping Li, Qiang Wu, and Christopher J Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*, pages 897–904, 2008. 7, 41

[22] Nauman Chaudhry, Kevin Shaw, and Mahdi Abdelguerfi. *Stream data management*, volume 30. Springer Science & Business Media, 2006. 7

[23] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011. 7

[24] Charu C Aggarwal, Jiawei Han, Jianyong Wang, and Philip S Yu. A framework for projected clustering of high dimensional data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 852–863. VLDB Endowment, 2004. 8

[25] Andrea Burattin, Alessandro Sperduti, and Wil MP van der Aalst. Control-flow discovery from event streams. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2420–2427. IEEE, 2014. 9, 10

[26] Giovanni Da San Martino, Nicolo Navarin, and Alessandro Sperduti. A lossy counting based approach for learning on streams of graphs on a budget. In *IJCAI*, pages 1294–1301, 2013. 10

[27] Marwan Hassani, Sergio Siccha, Florian Richter, and Thomas Seidl. Efficient process discovery from event streams using sequential pattern mining. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 1366–1373. IEEE, 2015. 10

[28] Asbjørn Følstad and Knut Kvale. Customer journeys: a systematic literature review. *Journal of Service Theory and Practice*, 28(2):196–227, 2018. 10

[29] Anna Meroni and Daniela Sangiorgi. *Design for services*, pages 83–85. Routledge, 2016. 10

[30] Marc Stickdorn, Jakob Schneider, Kate Andrews, and Adam Lawrence. *This is service design thinking: Basics, tools, cases*, volume 1, pages 17–36. Wiley Hoboken, NJ, 2011. 10

[31] Stefan Holmlid and Shelley Evenson. Bringing service design to service sciences, management and engineering. In *Service science, management and engineering education for the 21st century*, pages 341–345. Springer, 2008. 10

[32] Morris Foster, Susan Whittle, Stuart Smith, and Peter Hyde. Improving the service quality chain. *Managing Service Quality: An International Journal*, 1(1):41–46, 1991. 10

[33] Mirela Elena Nichita, Marcel Vulpoi, and Georgiana Toader. Knowledge management and customer relationship management for accounting services companies. *Chinese Business Review*, 12(6):435–442, 2013. 10

[34] Matthieu Harbich, Gaël Bernard, Pietro Berkes, Benoît Garbinato, and Periklis Andritsos. Discovering customer journey maps using a mixture of markov models. *SIMPDA*, 2017. 10

[35] Satu Peltola, Harri Vainio, and Marko Nieminen. Key factors in developing omnichannel customer experience with finnish retailers. In *International Conference on HCI in Business*, pages 335–346. Springer, 2015. 10

[36] Dragana Velimirović, Milan Velimirović, and Rade Stanković. Role and importance of key performance indicators measurement. *Serbian Journal of Management*, 6(1):63–72, 2011. 11

[37] F Robert Dwyer. Customer lifetime valuation to support marketing decision making. *Journal of interactive marketing*, 11(4):6–13, 1997. 11, 12

[38] V Kumar, Girish Ramani, and Timothy Bohling. Customer lifetime value approaches and best practice applications. *Journal of Interactive marketing*, 18(3):60–72, 2004. 12

[39] Sharad Borle, Siddharth S Singh, and Dipak C Jain. Customer lifetime value measurement. *Management science*, 54(1):100–112, 2008. 12

[40] Werner J Reinartz and Vijay Kumar. On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing. *Journal of marketing*, 64(4):17–35, 2000. 12

[41] Siddarth S Singh and Dipak Jain. Measuring customer lifetime value: models and analysis. *SSRN*, pages 10–17, 2013. 12, 40

[42] Jenny Van Doorn, Katherine N Lemon, Vikas Mittal, Stephan Nass, Doreén Pick, Peter Pirner, and Peter C Verhoef. Customer engagement behavior: Theoretical foundations and research directions. *Journal of service research*, 13(3):253–266, 2010. 13

[43] Roderick J Brodie, Linda D Hollebeek, Biljana Jurić, and Ana Ilić. Customer engagement: Conceptual domain, fundamental propositions, and implications for research. *Journal of service research*, 14(3):252–271, 2011. 13

[44] Monika Kukar-Kinney and Angeline G Close. The determinants of consumers' online shopping cart abandonment. *Journal of the Academy of Marketing Science*, 38(2):240–250, 2010. 13

[45] Ewa Maslowska, Edward C Malthouse, and Tom Collinger. The customer engagement ecosystem. *Journal of Marketing Management*, 32(5-6):469–501, 2016. 13

[46] Jan Roelf Bult and Tom Wansbeek. Optimal selection for direct mail. *Marketing Science*, 14(4):378–394, 1995. 13

[47] Mahboubeh Khajvand, Kiyana Zolfaghar, Sarah Ashoori, and Somayeh Alizadeh. Estimating customer lifetime value based on rfm analysis of customer purchase behavior: Case study. *Procedia Computer Science*, 3:57–63, 2011. 14

[48] Derya Birant. Data mining using rfm analysis. In *Knowledge-oriented applications in data mining*, pages 91–108. InTech, 2011. 14

[49] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 14

[50] John T Behrens. Principles and procedures of exploratory data analysis. *Psychological Methods*, 2(2):131, 1997. 16

[51] John W Tukey. We need both exploratory and confirmatory. *The American Statistician*, 34(1):23–25, 1980. 16

[52] Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013. 18

[53] Show-Jane Yen and Yue-Shi Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009. 24, 41

[54] Ivan Tomek. Two modifications of cnn. *IEEE Trans. Systems, Man and Cybernetics*, 6:769–772, 1976. 24

[55] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 24, 41

[56] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1322–1328. IEEE, 2008. 24

[57] Shihyen Chen, Bin Ma, and Kaizhong Zhang. On the similarity metric and the distance metric. *Theoretical Computer Science*, 410(24-25):2365–2376, 2009. 24

[58] Nicole Schweikardt. Short-entry on one-pass algorithms. *Encyclopedia of Database Systems*, pages 1948–1949, 2009. 28

[59] Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, volume 83. WoUongong, 1998. 32

[60] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008. 32

[61] Nikhil Bhat, V Farias, and C Moallemi. Optimal ab testing. Technical report, Working paper, 2015. 32

[62] MENG Minghui and ZHAO Chuanfeng. Application of support vector machines to a small-sample prediction. *Advances in Petroleum Exploration and Development*, 10(2):72–75, 2016. 41

[63] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016. 41

[64] Marwan Hassani, Daniel Töws, Alfredo Cuzzocrea, and Thomas Seidl. Bfspminer: an effective and efficient batch-free algorithm for mining sequential patterns over data streams. *International Journal of Data Science and Analytics*, pages 1–17, 2017. 46

Appendix A

Additional Prediction Results

In this appendix firstly the recall, MAP and $MAF1$ values when allowing for 1 or 5 predictions are given for the RFM groups not explored in detail when the predictions are done for the action immediately following. Afterwards a table is given in which the same metrics are given for the same RFM groups both when using OARA with a span of 3 and when using OARA with the additional dataset. For insights derived from these values please refer to Section 4.7.

A.1 Next Event Prediction

RFM Group	Event	Predictor	Recall@1	MAP@1	MAF1@1	Recall@5	MAP@5	MAF1@5
LLL	4	OARA	0.754	0.754	0.754	0.853	0.771	0.810
LLL	4	Grad-Boost	0.385	0.385	0.385	0.934	0.598	0.729
LLL	4	OCuLaR	0.026	0.032	0.029	0.918	0.466	0.618
LHL	4	OARA	0.262	0.311	0.284	0.689	0.379	0.489
LHL	4	Grad-Boost	0.19	0.19	0.19	0.71	0.369	0.486
LHL	4	OCuLaR	0.174	0.191	0.182	0.721	0.346	0.468
LHL	5	OARA	0.525	0.576	0.550	0.746	0.614	0.674
LHL	5	Grad-Boost	0.578	0.578	0.578	0.832	0.681	0.792
LHL	5	OCuLaR	0.083	0.099	0.090	0.677	0.287	0.403
LLH	4	OARA	0.354	0.409	0.379	0.538	0.454	0.492
LLH	4	Grad-Boost	0.570	0.570	0.570	0.801	0.658	0.723
LLH	4	OCuLaR	0.056	0.109	0.074	0.715	0.396	0.509
HHL	4	OARA	0.232	0.272	0.250	0.718	0.350	0.471
HHL	4	Grad-Boost	0.154	0.154	0.154	0.72	0.355	0.438
HHL	4	OCuLaR	0.136	0.166	0.149	0.716	0.365	0.484
HHL	5	OARA	0.379	0.411	0.395	0.782	0.476	0.592
HHL	5	Grad-Boost	0.383	0.383	0.383	0.770	0.513	0.616
HHL	5	OCuLaR	0.141	0.151	0.146	0.528	0.285	0.370
HLH	4	OARA	0.262	0.352	0.301	0.470	0.414	0.441
HLH	4	Grad-Boost	0.482	0.482	0.482	0.839	0.599	0.699
HLH	4	OCuLaR	0.262	0.352	0.301	0.470	0.414	0.441
LHH	4	OARA	0.131	0.229	0.167	0.439	0.291	0.350
LHH	4	Grad-Boost	0.148	0.148	0.148	0.630	0.307	0.413
LHH	4	OCuLaR	0.083	0.178	0.114	0.585	0.382	0.463
LHH	5	OARA	0.253	0.354	0.295	0.490	0.407	0.445
LHH	5	Grad-Boost	0.323	0.323	0.323	0.766	0.470	0.583
LHH	5	OCuLaR	0.080	0.146	0.103	0.464	0.313	0.374
LHH	6	OARA	0.410	0.410	0.410	0.575	0.446	0.499
LHH	6	Grad-Boost	0.647	0.647	0.647	0.867	0.727	0.791
LHH	6	OCuLaR	0.049	0.082	0.061	0.765	0.47	0.539

Table A.1: Metrics regarding predicting the next event using OARA on remaining RFM groups

A.2 Span of 3 - OARA

RFM Group	OG Pred Event	Recall@1	MAP@1	MAF1@1	Recall@5	MAP@5	MAF1@5
LLL	4	0.986	0.986	0.986	1.00	1.00	1.00
HLL	4	0.840	0.840	0.840	0.927	0.858	0.880
LHL	4	0.368	0.368	0.368	0.825	0.511	0.595
LLH	4	0.605	0.605	0.605	0.858	0.668	0.719
HHL	4	0.352	0.352	0.352	0.846	0.529	0.617
HLH	4	0.419	0.419	0.419	0.687	0.497	0.545
LHH	4	0.225	0.225	0.225	0.683	0.372	0.449

Table A.2: Metrics regarding predicting using OARA and a span of 3 on remaining RFM groups

A.3 Additional Dataset - OARA

RFM Group	Pred Event	Recall@1	MAP@1	MAF1@1	Recall@5	MAP@5	MAF1@5
LLL	4	0.822	0.822	0.822	0.922	0.866	0.893
LHL	4	0.388	0.444	0.415	0.722	0.589	0.649
LHL	5	0.520	0.532	0.525	0.876	0.589	0.648
LLH	4	0.408	0.500	0.449	0.625	0.618	0.621
HHL	4	0.243	0.264	0.253	0.75	0.454	0.566
HHL	5	0.355	0.387	0.370	0.849	0.572	0.684
HLH	4	0.269	0.362	0.308	0.504	0.532	0.518
LHH	4	0.146	0.255	0.191	0.466	0.443	0.452
LHH	5	0.224	0.354	0.275	0.565	0.488	0.524
LHH	6	0.516	0.516	0.516	0.736	0.646	0.691

Table A.3: Metrics regarding predicting using OARA and the additional data on remaining RFM groups

Appendix B

Paper based on the research

This appendix contains the paper which was based on this thesis that was submitted and accepted into the IEEE ICDM 2018 conference.

Effective Steering of Customer Journey via Context-Aware Recommendation

Submitted for blind revision
-
-
-

Submitted for blind revision
-
-
-

Submitted for blind revision
-
-
-

***Abstract*—Recently the analysis of customer journeys is a subject undergoing an intense study. The increase in understanding of customer behaviour serves as an important source of success to many organizations. Current research is however mostly focussed on visualizing these customer journeys to allow them to be more interpretable by humans. A deeper use of customer journey information in prediction and recommendation processes has not been achieved. This paper aims to take a step forward into that direction by introducing the Order Aware Recommendation Approach (OARA). The main scientific contributions showcased by this approach are (i) increasing performance on prediction and recommendation tasks by taking into account the explicit order of actions in the customer journey, (ii) showing how a visualization of a customer journey can play an important role during predictions and recommendations, and (iii) introducing a way of maximizing recommendations for any tailor-made Key Performance Indicator (KPI) instead of the accuracy-based metrics traditionally used for this task. An extensive experimental evaluation then highlights the potential of AORA against state-of-the-art approaches using a real dataset representing a customer journey of upgrading with multiple products.**

I. Introduction

In today's society, the interactions a customer has with an organization are quite plentiful thanks to the myriad of ways in which these customers are now able to interact with the organizations. These interactions can be seen as a sequence, where each time the customer achieves a certain goal with a specific interaction. Such a sequence of observed events which belongs to a single customer is referred to here as a customer journey. The analysis of such customer journeys can be a huge boon towards improving the organizations, as the key objective is to get an understanding of how the experiences of the customer can be enriched though what marketers call their decision-making process [1].

To properly interpret the customer journey data that organizations possess it is helpful to create a visualization of this information to get an idea of which steps are usually taken in the journeys. Such a representation is called a customer journey map. These artefacts often possess a non-linear structure while reflecting behavioural, emotional and cognitive drives [2]. A mapping in this paper is obtained by means of process mining. The result is known as a process model, of which an example is shown in Figure 1. The example shown here is from the website of a music festival. Firstly, a customer will have to register themselves. Upon completing the registration, they go on to either buy tickets or merchandise from a band. In case tickets are bought, it might occur that the customer wishes to also buy tickets from another band, which is indicated by the arrow to and from itself. A customer is able to end their journey after taking either of these actions, but it might also be the case that they still need to change part of their information, for example their payment credentials. This information can then be altered and afterwards everything is in order to deliver the tickets and/or merchandise, leading to the customer journey end. Note that in this example customers are only doing a single purchase, but it might also be the case that one wishes to model all purchases made by a single customer in which case the process model would be different and more complex.

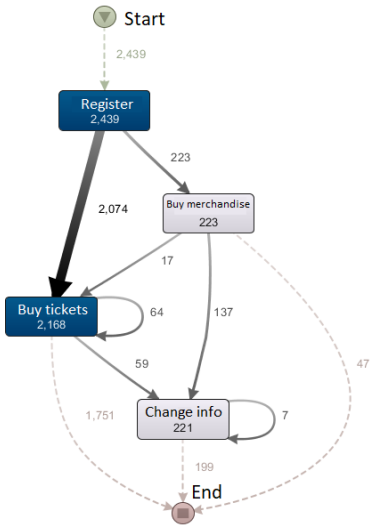


Fig. 1: Example process model of a customer journey.

Now that it has been made clear what customer journey information can be included in the process models, the goals are introduced. The approach proposed in this paper is called the Order Aware Recommendation Approach, shortened to OARA, and aims to improve upon the current state of the art in three areas. Firstly, the extraction of a customer journey map by means of process mining is a technique which hs been recently contributed in [3]. However, the approach proposed here aims to go beyond simply extracting a model. The

extracted model is now used by OARA to also do predictions and recommendations for future steps in the customer journeys in a tailor-made manner. This allows for the value of customer journey data to rise as a result, as one can now rely on machine learning techniques for these tasks which would otherwise involve a large amount of manual labour. Furthermore predictions and recommendations on the customer journeys can also be done by other predictor algorithms and recommendation systems since one still would like to obtain information on the future based on the past. OARA however aims to improve upon the existing methods as they do not take the explicit order information into account which is present in the customer journeys. This loss of contextual information can then lead to a decrease in quality compared to when this context is applied to the predictions and recommendations. There is also the issue that currently the evaluation of recommender systems is mainly focused around prediction accuracy, while other evaluation properties such as novelty are less explored [4]. This is a mismatch with reality as the goal organizations usually have when recommending the customer with an action is the maximization of a Key Performance Indicator (KPI). This is a value which measures how well an organization is performing on a specific key objective. To provide a solution to the current situation it is shown how one can take KPI maximization into account by using OARA.

To summarize, the contributions made by this paper are showing how it is possible to:

- Use a process model to do predictions and recommendations.
- Explicitly use the order of events during predictions and recommendations.
- Optimize recommendations for any chosen KPI.

The rest of this paper consists of the following sections: Firstly some related work for the rest of the paper is presented in Section II. In Section III the context and problems tackled in the paper are concretely defined, in Section IV it is explained in detail how OARA allows for the recommendations to be created. In Section V a real dataset is used for an evaluation of the quality of both the predictions and recommendations provided by OARA and to wrap everything in Section VI the conclusions and future research opportunities are given.

II. Related Work

In this section some related work from different areas is introduced. Firstly some more context is given regarding the concept of customer journeys, after which process mining is discussed. Following upon this stream data mining is covered and lastly the Recency, frequency and monetary value is introduced, which is later used for segregating the customer base.

A. Customer journeys

As described in [5], the term customer journey is quite widely used in scientific literature with yet no common understanding exists with regard to what a customer journey exactly entails. Descriptions used in the past include that a customer journey is the cumulation of repeated interactions between the service provider and customer [6], an ”engaging story” based on the interactions of a customer with the service [7], or a ”walk in the customer’s shoes” [8]. All descriptions have in common that a high importance is placed on the experience of the customer. The approach proposed in this paper aims at using the logged events to recreate this experienced journey and using the distinguishing qualities which lie inside them to achieve high quality predictions of customers future interactions and recommendations regarding the best possible interaction from both user and organization perspectives.

A combination of research between the research fields of process mining and customer journeys has occurred in the past [3], where the goals was to extract customer journey mappings. These are a visualization of the customer journey, and in that research the events which were relevant to the customer journey scenario were retained such that a process model could be created upon them. This process model was used both for further analysis tailored to process mining as well the creation of the customer journey mapping. The research in this paper also uses such a mapping towards a customer journey while taking care of two tasks in the proposed future research of this paper: finding techniques for clustering customer journeys and facilitating predictions on future behaviour in these journeys.

B. Process mining

Process mining is a research area which combines the domain of process modelling and analysis with the domain of data mining and machine learning. The goal of this combination is to discover, monitor and improve processes based on knowledge from data which is stored in the event log format regarding the process in question [9]. Event logs show the occurrence of events at a designated point in time, where the event is an action logged by an information system such as the sale of a product. This event is specified to have come from a specific process or instance, also known as case [10]. One such instance or case then encompasses all events belonging to a single customer which can be identified based on an ID.

To the approach described in this paper, process mining mainly plays a role in helping to determine which information and activities should be included during the predictions and recommendations on the customer journeys. Ideally, one is able to find an easy to understand model which shows the process from a high level, as exemplified in Figure 1. Here the overall process is short and intuitive, but there are also cases where the number of events per instance is very large while there are also connections between almost all of such events. In this case the model becomes entangled and as a result hard to interpret which can be counteracted by taking the most representative samples and segregating the customer base.

The notation for the customer journeys will be borrowed from the process mining domain, since as was shown by [3], a customer journey can be formed based on an event log. The examples on notation given here are based on Figure 1.

Firstly, a single event in the customer journey called for example $Register$ has the following combination of information: $Register = (c, a, t)$. c here stands for the case, which is a specific customer, a is the action performed, registering, and t the time at which the action was performed. A customer journey consists of multiple such events and is then denoted as $CJ = \langle Register, BuyTickets \rangle$, where $Register$ and $BuyTickets$ are events belonging to the same case ID c have consecutive timestamps. The entire collection of journeys is here equivalent to an event log and is denoted as $Log = \langle Register, BuyTickets \rangle, \langle Register, BuyMerchandise, ChangeInfo \rangle, \langle Register, BuyTickets, BuyTickets, ChangeInfo \rangle$. Note that based on the presence of a loop there is no exhaustive Log which covers all possible customer journeys and that journeys belonging to different customers might be interleaving depending on t.

C. Stream data mining

The environment described in this paper is one where the information is obtained by means of data streams, which can be characterized as continuous and typically non-constant [11]. There are two main issues which arise from such data streams. Firstly these streams produce massive, potentially infinite, amounts of data which can make it hard to use more time consuming operations on such data. Secondly the information in the data can change rapidly, which makes it important to facilitate an option for fast updates. This also does not suit the 'normal' data mining approaches either, as the multiple passes they require over the data are not possible in a streaming setting [12]. OARA aims to cover these issues by taking a collection of 'base' information on which it builds while having parts which can be updated with new information.

Several approaches have been proposed which are able to adapt to these circumstances. One of them is called OCuLaR [13], a recommender. The aim of of this approach was to generate recommendations which are easily interpretable by the customers based on data where there is implicit feedback, i.e. no information is supplied by the customers regarding their enjoyment on or motivations for choosing certain products. One thing to note about this approach is that it does not use any features, it only considers relationships based on the products customers bought, and as such does not explicitly use any context information during the recommendation. The approach proposed in this paper will on the other hand do this based on the hypothesis that there lies important knowledge in the context which can be used to amplify the predictive qualities.

D. Recency Frequency Monetary Value

The Recency, Frequency and Monetary values, often shortened to RFM, is a KPI based on how well a customer performs in the recency, frequency and monetary dimensions which has been introduced in [14]. Recency here means the time interval which has passed between the previously observed interaction of the customer and the present, Frequency involves how often a customer has interacted with the organization, possibly within a specified time period, and Monetary value is based on the cumulated amount of money the customer has spent at the organization. The RFM values are used in the case study of this paper to segregate the customer base, which was inspired by earlier successes such as reported in [15] and [16]. More specifically, the usage of the K-Means++ algorithm [17] in [16] to create the RFM groups is used. The notation for the groups here will be as follows: *XYZ*, where *X* signifies if the Recency was relatively high or low, and the same is signified by *Y* with regard to the Frequency and *Z* for the Monetary value. As such, for example one can have the *HLH* group where an event was recently observed, relatively few events were observed in total and the monetary value of the steps taken by the customer is relatively high due to the few purchases which were observed involving more expensive products.

III. Problem description and Setting

Firstly, the type of customer journey is described here to get an idea for what kind of environments the approach explained in this paper is applicable. A distinction can be made between journeys which have clear start and end points [18] and those which can be considered open-ended [19]. Respective examples are an appeal for tax returns which is either granted or rejected at the end and the purchasing behaviour of a customer at a convenience store. The emphasis lies on the latter type of customer journey in this paper, where one is not certain if the actor will remain engaged in the process or not. In this case there is no clear endpoint to a customer journey. To circumvent this, a solution is adapted from [20] where a customer journey is considered finished if there has been a significant period of inactivity. Significance is here determined by a time period being longer than the 85th percentile of the periods of inactivity between events.

Taking this limitation into account, the following main statistics apply to the data used during the case study: There are 35060 cases which contain a total of 141510 events made up out of 271 possible activities that lead to 9127 different variants. In the studied environment customers firstly buy a base product which allows for them to install upgrades and expansions in the future. These further purchases are entirely optional and solely conducted based on the interest the customer has in the product. A real life example of such a scenario is the purchase of a laptop, where the customer can additionally buy items such a mouse, carrying bag, or a new battery afterwards.

In the scenario of this paper a relatively large number of the customer journeys end early, with 38% buying only the bare necessity and finding out that the product doesn't fit them well enough. A consequence of this is that almost all of the 9127 variants are then contained in the remaining 62% of the journeys, which makes them rather heterogeneous and this will lead to some difficulties in their predictions, as will be showcased in Section V.

Given the above circumstances the aim is to strive towards the following two main goals:

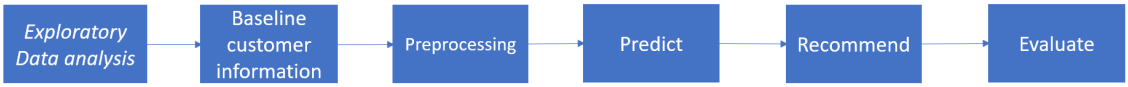


Fig. 2: A general overview of the components in the proposed approach.

1) Maximize metrics during the predictions of each of the events in the customer journeys. These predictions are done for future events in the customer journeys based on what has been observed in the past in them.
2) Maximize a customizable KPI during the recommendations. This KPI can be any value which is integral to the success of an organization in a specific area, such as sales or customer interaction. Examples of KPIs are the average purchase value, monthly sales bookings, and the customer engagement index.

Of particular interest to this paper is then determining how useful it can be to take into account the inherent order which is contained in the customer journeys while achieving these goals. The reason for this is that from an intuitive standpoint this additional information should be helpful for doing the correct predictions and recommendations. As such the main research question being answered by this paper is:

What influence does taking into account the inherent order in which events occur inside customer journeys have on both predictions and recommendations?

IV. OARA Overview

In Figure 2 a general overview can be observed which covers the main components of the system. The starting point is doing an optional exploratory data analysis to gain further insights into the dataset, for example by means of process mining or other visualization means. Afterwards the baseline customer information is determined that form the base for future predictions and recommendations. As was mentioned in the Section III, at this step one should take care that customer journeys have an appropriate time to finish in scenarios where there is no clear-cut end point to the customer journey as the only indicator of a finished customer journey in such a case is a prolonged period during which no events are monitored. Once the data to be used has been determined some preprocessing is needed for the baseline customer information to reach its maximal potential during the next two phases, the predictions and recommendations. The predictions are used during the recommendations in OARA and as such these activities cannot be conducted in parallel once the preprocessing has finished.

A. Preprocessing

Preprocessing part firstly involves the segregation of the set of customer journeys into smaller segments. The goal of this segregation is to allocate either specific types of customers which can be identified by domain knowledge, or customers which exhibit similar behaviour into their own groups. The reason for this is that then less behaviour needs to be kept in mind during the predictions and recommendations for a single group. As was mentioned in Section II, one way of segregating the customer used in this case study in this paper is based on the RFM values where customers who score similarly in each of the dimensions are grouped together.

Secondly during the preprocessing, the chosen KPI is calculated for each of the customer journeys to determine which scored better on it. During this stage, it is important that the chosen KPI is one suitable for the domain in which the scenario is deployed. Since this is a domain specific task, the only concrete advice which can be given in this regard is that the chosen KPI should be properly quantifiable for each of the customer journeys.

Another important aspect of the preprocessing of OARA is obtaining the customer journey mapping by creating a process model as mentioned in the introduction. There are many scientific tools available to do this, most notably ProM [21] and RapidMiner [22]. The main important point here is to strike a good balance between interpretability and complexity. The model should not be overly simplistic to the point where important parts of the customer journey are left out, while also making sure that the included events are common enough that they can be learned and predicted properly by a machine learning algorithm. The optional exploratory data analysis can help a great deal here, as this will aid in finding a proper balance since there is no ideal guideline to follow here. Upon having collected the process model, only those journeys which fit into the process model are used from the baseline customer information.

Apart from these tasks, the main interesting part of the preprocessing is the creation of representative customer journeys. The goal of these representative customer journeys is, as the name implies, to act as an artefact which represents most of the experiences encountered in the customer journeys between different customers. These are used during the recommendations, where they are used as a sanity check to make sure that the recommended action is both optimal and reasonable based on how well the observed actions match between the representative customer journey and the one which requires a recommendations. Two types of representative customer journeys are proposed in this paper, which are the subset-based representative customer journeys (SRCJ) and aggregated representative customer journeys (ARCJ).

1) Subset-based representative customer journey: The condition of the SRCJ is that there is a certain threshold of occurrence which a subset of distinct customer journey variants needs to surpass based on their weights. The weight is determined by how many journeys followed that exact same path. As such the condition $Threshold < \sum_{i=1}^{n} weight(i)$ needs to hold, where n is the number of allowed customer

journey variants. If this is passed, the variants can become the SRCJs. It may however be the case that the customer journeys differ wildly between the different customers. This would require the subset-based approach to then either have a very high n or to only represent a small margin of the customer base. In such a case, using SRCJs is a rather poor course of action.

2) Aggregated representative customer journey: The ARCJ acts as an alternative option to the SRCJs if large differences between the customer journeys are present. For the creation of the ARCJs all customer journeys are divided into groups based on how well they score according to the KPI which is to be maximized. Then inside each group $\sum_{j=1}^{m} \frac{\sum_{i=1}^{n} Action(j)}{n}$ is calculated. Here n is once again the number of customer journeys, m is the number of possible actions observed in the journeys and $Action$ is a boolean value based on if the action was present inside the i'th journey. As one can tell from the formula, the averages then form the ARCJs. As the usage of ARCJs is generally less precise than the usage of SRCJs, it is discouraged to use it if SRCJs are also a valid option.

As a final note here both the SRCJ and ARCJ can be updated based on new information smoothly to fit into a streaming setting. For the SRCJ one only needs to check if a different sequence has become more common than the current least common SRCJ, and at the ARCJ the averages can be changed based on a new journey which has a similar KPI value. If one desires this can also be configured to give preference to the newer customer journeys to make sure recent trends are taken into account during recommendations.

B. Predictions

Predictions are conducted with regard to the next event which occurs in the customer journey. Note that the predictions also include the option of predicting a customer journey to end. This is primarily interesting for journeys which do not have a set ending point, as in that case one predicts at which point the customer loses interest in continuing their journey. This is an avenue usually left unexplored for recommendation systems, where the main focus lies on monitoring the events actually logged by the system. Adding such knowledge of a customer losing interest can be useful by for example sending them a special offer to rekindle their interest.

Predictions should be conducted using an algorithm that allows for multiple options to be returned with a certain likelihood, as very often there is a vast range of options in the available paths which lie inside the customer journey. In these cases only providing a single option can lead to poor predictive qualities. In such cases it is valuable to take a larger number of predictions into account which all have a relatively high potential of being useful to the customer.

Based on the presented conditions the process of conducting the predictions using OARA is explained in further detail here. As the name indicates, the order in which the events have occurred inside the customer journey is taken into account here. This means that the past is not considered to be a bag of unordered events such as for example in the OCuLaR algorithm. This added structure improves distinguishability of the information used for predictions based on the assumption that customers who have followed the same trail in their customer journey have a high likelihood of taking similar actions in the future as well.

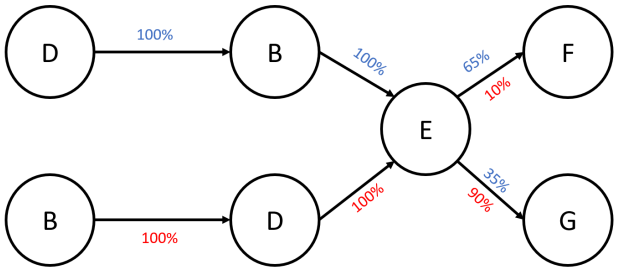


Fig. 3: Example of order influencing event probability.

An example of how taking the order into account can help is given in Figure 3. Here the order in which B and D are conducted has a high influence on what occurs after event E. If for example $\langle B, D, E\rangle$ is observed then the next event is F with a probability of 65%, while observing $\langle D, B, E\rangle$ lowers this to only 10%. Here the order clearly influences future choices, since without this order one would only observe that events B, D and E occurred and have a harder time determining whether F or G will follow.

Algorithm 1 ObtainPredictors

Input: Training customer journeys n
Output: Predictors $predictorsArray$

Initialize $sequencesArray, featuresArray, outcomesArray, predictorsArray$
for $i = 0$ to $len(n)$ **do**
  for $j = 0$ to $len(sequence(i)) - 1$ **do**
    $presequence =$ first j events of $sequence(i)$
    if $presequence$ not in $sequencesArray$ **then**
      Add $presequence$ to $sequenesArray$
    end if
    Obtain $features(presequence)$ and add to $featuresArray$
    Add $nextEvent(presequence)$ to $outcomesArray$
  end for
end for
for seq in $sequencesArray$ **do**
  Fit $predictor$ to $featuresArray(seq)$ and $outcomesArray(seq)$
  Add $predictor$ to $predictorsArray$
end for
return $predictorsArray$

Algorithm 2 OARA Prediction method

Input: Predictors P, Customer journeys n
Output: Predictions $predictionsArray$

```
1: Initialize sequencesArray, featuresArray, predictionsArray
2: for i = 0 to len(n) do
3:    Add sequence(i) to sequencesArray
4:    Obtain features(i) and add to featuresArray
5: end for
6: for j = 0 to len(sequencesArray) do
7:    Obtain prediction(j) based on predictor P[sequencesArray[j]] using featuresArray[j] and add prediction(j) to predictionsArray
8: end for
9: return predictionsArray
```

Algorithm 3 OARA Recommendation method

Input: Representative journeys RCJ, Predictions P, Prediction Sequences PS, Conditions C
Output: Recommendations $recommendationsArray$

```
1: Initialize distanceArray, recommendationsArray
2: for i = 0 to len(P) do
3:    Initialize distancesArray
4:    for j = 0 to len(RCJ) do
5:       Obtain distance(i)(j) between PS(i) and RCJ(j)
6:       Append distance(i)(j) to distancesArray
7:    end for
8:    Initialize foundRecc = False
9:    for k = len(RCJ) to 0 do
10:      if foundRecc == False then
11:         currentDist = distancesArray(i)(k)
12:         if C based on currentDist are met then
13:            Get recommendation based on P(i) and RJC(k) and add to recommendationsArray
14:            foundRecc = True
15:         end if
16:      end if
17:   end for
18: end for
19: return recommendationsArray
```

To properly do predictions for any sequence observed, OARA employs predictors for each of these sequences. As such, predictors need to be trained for all these sequences before proper predictions can be conducted for new customer journeys. This process is described in pseudocode in Algorithm 1. Here the sequence is the current path of a customer journey, e.g. $\langle A, B, C\rangle$. In that case there are two presequences to take into account, namely $\langle A\rangle$ and $\langle A, B\rangle$. For these presequences the features are then extracted based on the data available at those points of the journey, and the following event is stored as well. Once all customer journeys have been checked in this manner a predictor is trained on the features and outcomes of the sequences in the base customer data. Note that in cases where the journeys contain a large number of events which are heterogeneous that it can be useful to not take into the longer subsequences as there will be too little training information available for them to properly train the classifiers. Based on preliminary tests in these cases performance will increase if the subsequence is decreased in size.

Once these predictors are obtained the predictions using OARA can be conducted on new samples, which is described in Algorithm 2. Here the current sequence of the new customer journeys is obtained as well as their features, and then the prediction is conducted based on the pre-learned predictor which was tailored towards that sequence to give a prediction optimized for the order of events observed.

C. Recommendations

The recommendations are here aimed at maximizing a previously chosen KPI. This does however not mean that only the most profitable action is recommended for all customer journeys, as it is also taken into account how likely a customer is to take the recommended action. This is where the previously created representative customer journeys come into play. The representative journeys provide insights into which customer journeys led to higher KPI values and it is then possible to check how well a new customer journey aligns to the representative ones to get an idea of how reasonable a recommendation from that representative journey would be.

The general outline of how the recommendations using OARA are done is given in the pseudocode of Algorithm 3. The two most interesting points here are the $distance$ and conditions(C) parameters used in respectively Lines 5-6 and 12. The $distance$ between the representative journey and the new customer journey is measured based on how well the events in the new customer journey match up with the representative journey. One way of doing this is by first using one-hot-encoding for all types available at each of the observed events in the customer journey. After that the differences in value between the representative customer journey and the current customer journey can be calculated. It should be noted that this can be adapted to whatever preferred distance measure is most applicable.

The other important parameter is the conditions, which specify the constraints to which a recommendation needs to adhere. These are based on the relative distance of a customer journey to the representative journey. An example of such conditions can be found in Table II. Here there are 4 tiers, where the actual values of the percentiles can for example

Distance Tier	Distance%	Conditions
Best	0-15	Recommend 2 most likely events based on representative journey
Good	16-50	Recommend any of the top 2 predictions that match the representative journey
Decent	51-85	Recommend the top prediction if it matches the representative journey
Poor	86-100	Do not recommend based on this

TABLE I: Example of conditions on the recommendations

be based on the distances observed between the representative journeys and all journeys in the baseline customer information. This creates a baseline for the distances on which new journeys can be judged. The exact conditions are tunable based on the context in which OARA is employed. In the example if a new journey falls into the 'Best' distance tier for a representative customer journey then the top 2 most likely events in the representative one are recommended. In the 'Good' distance tier if any of the top 2 predictions for the new journey match with what occurs in the representative journey then they are recommended, and in the next tier the same holds for the topmost prediction. In case the journey is not similar to the representative one then the recommendation will not be based on it since the chance that they will follow something which aligns so poorly with their behaviour is negligible.

These conditions are tested in the order from the highest ranked customer journey to the lowest ranked one, to try and route the customer on a path that maximizes the KPI. Furthermore it is never recommended for a customer to stop their journey as removing contact with the customer is only of use to an organization based on very specific conditions. In the unlikely event that none of the conditions can be met the most often observed action from the highest scoring representative journey is recommended. This is done to still give some sort of advice which can lead to an advancement of the KPI.

V. Experimental Evaluation

As was previously mentioned, the evaluated customer journeys involved the initial purchase of a base product to which upgrades and expansions can be attached in the future. The customer journeys were then split into 8 groups based on if their RFM values were relatively high or low. In this section the *HLL* and *HHH* groups are considered. These were chosen as they contained the most customer journeys while also being most interesting from a business perspective. The *HLL* group covers the customers who only recently started their journey and still have to determine if they appreciate the product and as such holds a lot of potential value if their interest can be retained. Conversely the *HHH* group involves the 'best' customers which are currently still interested in the product and who already have purchased a relatively large number of products, which means that if relevant products can be recommended to them they are likely to also be interested in those products. For these groups process mining was used to extract customer journey maps, which helped filter out the journeys that are very hard to predict. The journey here consists of events which occur in a sequential order and those used for predictions are the 4th and 5th observed ones. Note that for the *HLL* group only the 4th is considered as the others do not exist in this group since the customer would then fall in a higher RFM group. Furthermore the 3 first events are all part of the initial setup of the product of the customer, and are therefore not predicted.

Outside of these initial segregations, OARA was configured here to use 10 ARCJs as there was too much variance for a reasonable number of SRCJs to properly represent the customer base. Based on preliminary tests the best performing algorithm to use as predictors for each sequence was the Support Vector Machine, which has been used for multi-label classification with success in the past [23]. The conditions were the same as listed in Table I.

The predictions and recommendations on these customer journeys have been conducted in a multitude of ways to facilitate an overview of how different approaches were able to tackle this dataset. The competitors which are compared are OARA, gradient boosting trees [24] and OCuLaR [13]. The reason gradient boosting trees were chosen to represent the traditional methods over any other well known machine learning algorithm is that preliminary results for this one were most promising. All scores have been obtained in a cross-validated manner while optimizing parameters for the methods based on the relevant metric.

The main metric of comparison used here is an altered version of the F1-score, which is normally built up from precision and recall but here mean average precision is used instead of normal precision. The reason for this is to prevent punishing additional predictions in case the correct prediction was already conducted, as average precision only updates when recall changes. To indicate the different F1-score it is called Mean Averaged F1(MAF1). The following formulas show exactly how it is built up:

$$Recall = \frac{\#CorrectPredictions}{\#Items} \tag{1}$$

$$Precision = \frac{\#CorrectPredictions}{\#Predictions} \tag{2}$$

$$AP = \sum_{i=1}^{n} Precision(i)\Delta Recall(i) \tag{3}$$

$$MAP = \sum_{i=1}^{n} AP(i)/n \tag{4}$$

$$MAF1 = 2 * \frac{MAP*Recall}{MAP+Recall} \tag{5}$$

A. Predicting the next event

The main objective of the predictions i determining the very next event in the customer journey. For this reason the MAF1 score has been obtained for the 5 top-most predictions for each of the 3 predictors, which can be found in Figures 4-6.

1) HLL Event 4: Based on the MAF1 scores in Figure 4 the predictions for this event are done mostly equally well for both OARA and Gradient-boosting, although OARA achieves a better initial prediction. As such, recall rises a bit faster using Gradient-Boosting, while OARA relies more on it's initial high precision. OCuLaR performs worse here mostly due to there being a relatively large number of people who stop their customer journey at this event, which is a bit troublesome for it to identify due to it missing the feature information which the other two competitors have access to.

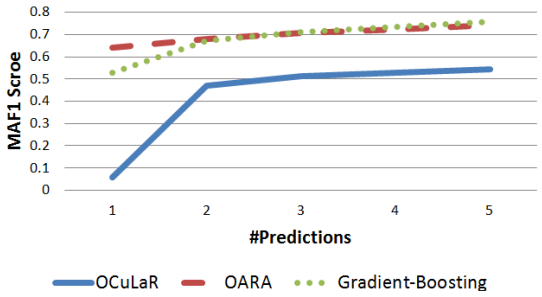


Fig. 4: MAF1 scores for the 4th event in group *HLL*.

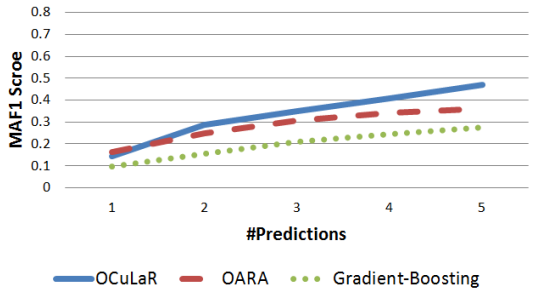


Fig. 5: MAF1 scores for the 4th event in group *HHH*.

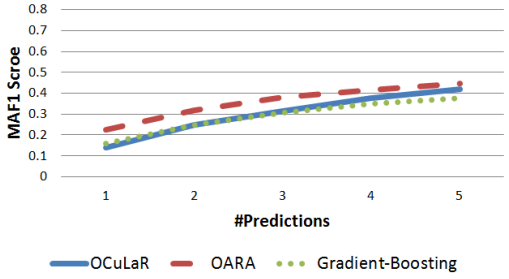


Fig. 6: MAF1 scores for the 5th event in group *HHH*.

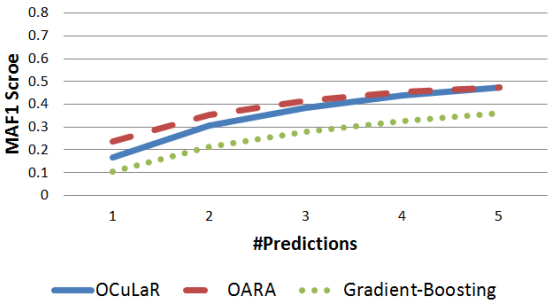


Fig. 7: MAF1 scores for the 4th event in group *HHH* with span of 3.

2) HHH Event 4: Based on the performance in Figure 5 all algorithms perform poorly here. This is caused by there being a large number of options to choose from which are all relatively uncommon and an insufficient amount of available information to distinguish between them. This shortage of information allows OCuLaR to outperform the other two methods due to it more effectively obtaining insights based on just the products bought if multiple predictions are allowed. OARA then, thanks to taking the order into account, still outperforms Gradient-Boosting, but is mostly less effective.

3) HHH Event 5: When comparing Figures 5 and 6 the prediction quality and conditions are about equal. However, this time OARA is able to outperform the other two methods instead of OCuLaR. The additional event has led to enough information becoming available that the combination of the order and features has become well-suited to do the predictions.

B. Using a span

A *span* here refers to a timespan during which it is allowed for a prediction to be valid. To clarify, if *span* = 3 then if the predicted action shows up either in the next event, the event after that, or the event following that then the prediction is considered to be correct. This can be useful when one is relatively sure that a group of actions will be conducted in the near future without the order being set in stone. An example of this is a user of an online music service who has already bought 3 albums of a single artist, where one can be relatively certain they will buy another album of that artist but not which one. Usage of a *span* for sequences of events is not unprecedented and has been used with success in the past in [25] which is adapted to the current situation.

The effect of using the *span* for the predictors here is exemplified in Figure 7, where the effect was most noticeable. A *span* of 3 was used here and when compared to the MAF1 scores obtained in Figure 5 the MAF1 scores here are higher due to the relaxed prediction conditions, as is to be expected. However, not all predictors profit equally from this and it allows for OARA to now outperform OCuLaR while with a *span* of 1 this is reversed. This shows that given a scenario where a span is reasonable using OARA can help improve the quality of the predictions.

C. Including additional context information

From an intuitive standpoint, it makes sense that to increase the predictive qualities it is helpful to include additional information. This enriches the customer journey by providing additional context to the observed events, much like taking the order into account did. To test this hypothesis, an additional dataset has been obtained and deployed in the use case. The comparison of the results with and without the added dataset can be found in Table II.

In this table the $MAF1$ score is given for the predictions of the events when allowing for 1 or 5 predictions without using a larger span. Based on the $MAF1$-scores the main improvements are found at the prediction of event 5 of the *HHH* group, while when allowing for multiple predictions the $MAF1$-scores of the predictions for the 4th event in both groups also seems to rise. The only time when there is little effect is during the prediction of the 6th event in the *HHH*

TABLE II: Comparing MAF1 scores based on the presence of context data

	HLL-4@1	HLL-4@5	HHH-4@1	HHH-4@5	HHH-5@1	HHH-5@5	HHH-6@1	HHH-6@5
No context data	0.642	0.741	0.161	0.360	0.223	0.447	0.388	0.582
With context data	0.652	0.783	0.146	0.415	0.291	0.479	0.382	0.583

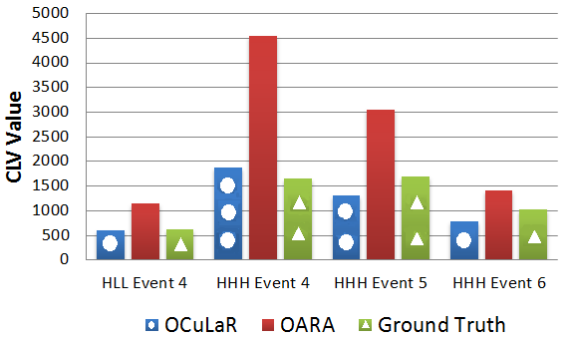


Fig. 8: Comparison of the CLV values on recommendations.

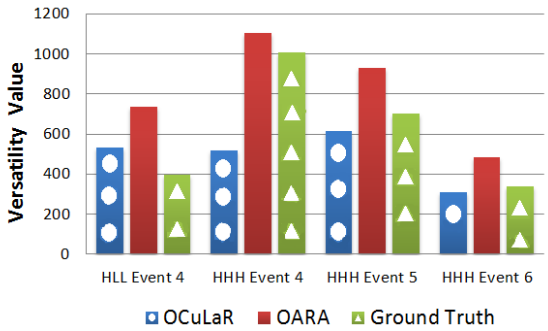


Fig. 9: Comparison of the versatility values on recommendations.

group. As such the context data used here is mainly of use during the earlier stages of the customer journey. Additional context information from a different source may prove useful to improve on the later predictions. This experiment has shown that providing additional context information can lead to an increase in metrics when using OARA.

D. Recommendations evaluation on KPIs

The evaluation of the recommendations for the next event in the customer journey need to be done in a novel way. The reason being that the evaluation of recommender systems is traditionally based on measures related to the precision of the recommendations and not the maximization of a KPI. To solve this the following metric is introduced which aims to capture how much value the recommender brings in terms of the KPI:

$$TotalKPI = \sum_{i=1}^{n} : KPI(recc(i)) \tag{6}$$

Here n is the number of customers which have been recommended a next step in the journey, and the KPI is calculated based on a specific recommender system. Note that this operates under the restriction that a KPI is used which can be calculated at any point in the customer journey. In other words, every step contributes a certain amount of KPI value. If this is adhered to then *TotalKPI* allows for an estimation of the recommendation's effect on the KPI. This however also relies on the assumption that the customers always follow the recommendation. It should be noted that this assumption is different from what one can expect to see in real life, and is mostly in place due to the lack of any prior research on how often recommendations are followed up on by customers. In case one wishes to be more realistic then one can for example assume only half of the recommended events are followed by the customers, while taking the KPI from the ground truth in the remaining cases.

$TotalKPI$ was calculated for OARA under the positive assumption that all recommendations are followed as well as OCuLaR and the ground truth to see if there is any positive effect. The first KPI used here was the Customer Lifetime Value(CLV), a KPI which aims to capture how valuable a customer is to an organization. Based on the specification in [26] the CLV is here non-contractual, i.e. customer can always stop buying products, and dynamic, which means each action has an effect on the KPI. The CLV is for this example solely based on the revenue per step in the customer journey.

In Figure 8 the CLV values have been calculated for 4 recommendations. The first two recommendations involving the *HHH* group have higher CLV values for all 3 recommendations methods due to there being more samples to increase the overall CLV. For each of the 4 recommendations OARA is outperforming the alternatives, which is caused by only OARA optimizing explicitly for the CLV value. There are also times when customers strictly following the OCuLaR recommendation would have a lower CLV value than the ground truth, although one should take into account that if the recommendation fit their tastes very well that this could positively affect their future purchasing behaviour. The same can however be said for OARA, which aims have a more direct influence by giving a recommendation that immediately increases the KPI. If one were to be more pessimistic and assume that only a subset of all recommendations by OARA are actually used, the advantage of OARA would decrease. It would however remain useful as the sum of CLV values may decrease yet it will not sink below the ground truth.

Another KPI which can be relevant is versatility in the types of products a customer purchases. This is then measured in the number of different types a customer purchases during their journey, and as such the highest ranked representative customer journeys are those which involve the purchase of many different types of products. The result when using this as

the KPI when calculating $TotalKPI$ can be seen in Figure 9. The result is similar to Figure 8 in the sense that OARA outperforms both OCuLaR and the ground truth, although this time with slimmer margins. Most notably here at the 4th event of group *HHH* the ground truth is very close to the OARA recommendation which is caused by this group of customers naturally buying many different items already, which lessens the need for and effect of a recommendation. As such recommendations here are mainly of use for customers which are conservative in their initial purchases.

VI. Conclusion and Future Work

The approach proposed in this paper allows for the predictions and recommendations on datasets which fit the concept of a customer journey, showcasing that one can go beyond merely visualizing the journey in a process model by utilizing the model for these tasks. Further scientific contributions which have followed from this research are an investigation of the positive effects of taking into account the order of events if they exist in a sequential manner. These are observable during the predictions both in situations where they need to be accurate in the immediate and slightly more remote future based on the selected *span*. It was also shown that OARA can be further enriched by effective use of an additional source of information. These predictions are then used in combination with the representative customer journeys during the recommendations to find a recommendation that both increases the KPI and is well suited based on the actions previously observed in the customer journey.

Interesting future work includes looking further into proper evaluation metrics in settings where recommendations are aimed at improving a general KPI. The main shortcoming currently lies in the assumption that the customers follow recommendations blindly, which was put into place due to a lack of prior research on how often customers actually follow the given recommendation. As such a case study of the effectiveness of recommendation could provide a lot of value to the assessment of recommender systems. Furthermore OARA has currently only been employed in a single scenario. Therefore deploying it in a different environment will likely lead to further insights on optimizations and generalizations in areas which were not significant in the scenario of this paper. It is preferable that this scenario includes structures where events can be conducted in parallel, after which a specific event follows. In theory OARA can handle such a sequence of events perfectly given the awareness of past behaviour in the customer journey, however no such patterns exist in the data of the current case study to verify this claim.

References

[1] Katherine N Lemon and Peter C Verhoef. Understanding customer experience throughout the customer journey. *Journal of Marketing*, 80(6):69–96, 2016.

[2] Julia Wolny and Nipawan Charoensuksai. Mapping customer journeys in multichannel decision-making. *Journal of Direct, Data and Digital Marketing Practice*, 15(4):317–326, 2014.

[3] Gaël Bernard and Periklis Andritsos. A process mining based model for customer journey mapping. In *Proceedings of the Forum and Doctoral Consortium Papers Presented at the 29th International Conference on Advanced Information Systems Engineering (CAiSE 2017)*, pages 49–56, 2017.

[4] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[5] Asbjørn Følstad and Knut Kvale. Customer journeys: a systematic literature review. *Journal of Service Theory and Practice*, 28(2):196–227, 2018.

[6] Anna Meroni and Daniela Sangiorgi. *Design for services*, pages 83–85. Routledge, 2016.

[7] Marc Stickdorn, Jakob Schneider, Kate Andrews, and Adam Lawrence. *This is service design thinking: Basics, tools, cases*, volume 1, pages 17–36. Wiley Hoboken, NJ, 2011.

[8] Stefan Holmlid and Shelley Evenson. Bringing service design to service sciences, management and engineering. In *Service science, management and engineering education for the 21st century*, pages 341–345. Springer, 2008.

[9] Wil Van Der Aalst, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, et al. Process mining manifesto. In *International Conference on Business Process Management*, pages 169–194. Springer, 2011.

[10] Boudewijn Frans van Dongen. Process mining and verification, 2007.

[11] Nauman Chaudhry, Kevin Shaw, and Mahdi Abdelguerfi. *Stream data management*, volume 30. Springer Science & Business Media, 2006.

[12] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[13] Reinhard Heckel, Michail Vlachos, Thomas Parnell, and Celestine Dünner. Scalable and interpretable product recommendations via overlapping co-clustering. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*, pages 1033–1044. IEEE, 2017.

[14] Jan Roelf Bult and Tom Wansbeek. Optimal selection for direct mail. *Marketing Science*, 14(4):378–394, 1995.

[15] Mahboubeh Khajvand, Kiyana Zolfaghar, Sarah Ashoori, and Somayeh Alizadeh. Estimating customer lifetime value based on RFM analysis of customer purchase behavior: Case study. *Procedia Computer Science*, 3:57–63, 2011.

[16] Derya Birant. Data mining using RFM analysis. In *Knowledge-oriented applications in data mining*, pages 91–108. InTech, 2011.

[17] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[18] Morris Foster, Susan Whittle, Stuart Smith, and Peter Hyde. Improving the service quality chain. *Managing Service Quality: An International Journal*, 1(1):41–46, 1991.

[19] Mirela Elena Nichita, Marcel Vulpoi, and Georgiana Toader. Knowledge management and customer relationship management for accounting services companies. *Chinese Business Review*, 12(6):435–442, 2013.

[20] Marwan Hassani, Sergio Siccha, Florian Richter, and Thomas Seidl. Efficient process discovery from event streams using sequential pattern mining. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 1366–1373. IEEE, 2015.

[21] Boudewijn F Van Dongen, Ana Karla A de Medeiros, HMW Verbeek, AJMM Weijters, and Wil MP Van Der Aalst. The ProM framework: A new era in process mining tool support. In *International conference on application and theory of petri nets*, pages 444–454. Springer, 2005.

[22] Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

[23] Glenn M Fung and Olvi L Mangasarian. Multicategory proximal support vector machine classifiers. *Machine learning*, 59(1-2):77–97, 2005.

[24] Ping Li, Qiang Wu, and Christopher J Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*, pages 897–904, 2008.

[25] Marwan Hassani, Daniel Töws, Alfredo Cuzzocrea, and Thomas Seidl. BFSPMiner: an effective and efficient batch-free algorithm for mining sequential patterns over data streams. *International Journal of Data Science and Analytics*, pages 1–17, 2017.

[26] Siddarth S Singh and Dipak Jain. Measuring customer lifetime value: models and analysis. *SSRN*, pages 10–17, 2013.