UPTEC STS 20014
Examensarbete 30 hp
Juni 2020

UPPSALA UNIVERSITET

Designing an Interactive tool for Cluster Analysis of Clickstream Data

Sara Collin
Ingrid Möllerberg

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Designing an Interactive tool for Cluster Analysis of Clickstream Data

Sara Collin, Ingrid Möllerberg

The purpose of this study was to develop an interactive tool that enables identification of different types of users of an application based on clickstream data. A complex hierarchical clustering algorithm tool called Recursive Hierarchical Clustering (RHC) was used. RHC provides a visualisation of user types as clusters, where each cluster has its own distinguishing action pattern, i.e., one or several consecutive actions made by the user in the application. A case study was conducted on the mobile application Plick, which is an application for selling and buying second hand clothes.
During the course of the project, the analysis and its result was discovered to be difficult to understand by the operators of the tool. The interactive tool had to be extended to visualise the complex analysis and its result in an intuitive way. A literature study of how humans interpret information, and how to present it to operators, was conducted and led to a redesign of the tool. More information was added to each cluster to enable further understanding of the clustering results. A clustering reconfiguration option was also created where operators of the tool got the possibility to interact with the analysis.
In the reconfiguration, the operator could change the input file of the cluster analysis and thus the end result. Usability tests showed that the extra added information about the clusters served as an amplification and a verification of the original results presented by RHC. In some cases the original result presented by RHC was used as a verification to user group identification made by the operator solely based on the extra added information. The usability tests showed that the complex analysis with its results could be understood and configured without considerable comprehension of the algorithm. Instead it seemed like it could be successfully used in order to identify user types with help of visual clues in the interface and default settings in the reconfiguration. The visualisation tool is shown to be successful in identifying and visualising user groups in an intuitive way.

Handledare: Elliot Rask
Ämnesgranskare: Mikael Laaksoharju
Examinator: Elísabet Andrésdóttir
ISSN: 1650-8319, UPTEC STS 20014
Tryckt av: UPPSALA

Wordlist and Abbreviations

- A **screen view** is one type of action a user can trigger. It is most often representing one screen of the application. Example: Looking at an ad.
- An **event** is one type of action a user can trigger. It is most often representing a press of a button in the application. Example: Giving a like to an ad.
- Screen views and events are often referred to as **actions**.
- **Clickstream** is a stream of actions occurring one after another in an application.
- **Personal Data** is all types of data that can be connected to a specific person. This includes, but is not limited to: Name, Social security number and email.
- **Pseudonymised information** is information that is anonymised to make it impossible to distinguish any personal data. Unlike anonymisation however, the real information exists somewhere, but very few people have access to it, and it is not meant to be used.
- Recursive Hierarchical Clustering (**RHC**), is a tool for clustering users of applications. The visualisation tool this thesis results in is based on RHC.
- **The visualisation tool/ The tool** is RHC with our changes in visualisation and functionality.
- A **user** is a user of the studied application.
- An **operator** is a user of the visualisation tool, or users of tools in general in the theory.

Populärvetenskaplig sammanfattning

Denna studie påvisade att det är möjligt att bygga ett verktyg som presenterar användardata från en mobil applikation så att en människa kan förstå och tolka in vilka användartyper som finns i applikationen. Den statistiska metoden hierarkisk klusteranalys användes för att identifiera olika användargrupper. Klusteranalyser grupperar datapunkter baserat på hur lika de är varandra. I denna studie grupperades användarna efter hur de rörde sig i en applikation. Det vill säga, det var de fönster och knappar användaren tryckt på som låg till grund för analysen.

En fallstudie utfördes på applikationen Plick. Plick är en marknadsplats där individer kan köpa och sälja kläder i second hand. Applikationen är byggd så att sociala interaktioner är en naturlig del av användandet, eftersom användare gillar, kommenterar, följer eller ger betyg till varandra. Då utbytet av produkter sker mellan användarna utan mellanhand är applikationen en så kallad peer-to-peer marknadsplats. Idag är det få i Sverige, där Plick är verksamma, som inte har tillgång till smartphones och den uppkoppling det innebär. En applikation som bygger på sociala interaktioner är ingenting utan dess användare. Att förstå användaren genom ett verktyg som detta är således intressant för fortsatt apputveckling. Det går genom verktyget att upptäcka användningssätt av applikationen som utvecklarna inte kände till, men eventuellt vill bygga vidare på.

Utvecklingen av verktyget kan delas upp i två faser. I den första fasen gjordes en intervju - och litteraturstudie av tidigare arbeten kring Plick för att samla bakgrundsinformation om applikationen, samt ytterligare en litteraturstudie för att finna en lämplig klusteralgoritm för analysen av användartyper. Sedan påbörjades utvecklingen av verktyget. Den andra fasen inleddes när verktyget implementerats framgångsrikt på användardatan och analysen givit resultat. Detta genom ett arbete med att göra resultaten och analysen förståelig för användaren av verktyget. En ytterligare litteraturstudie utfördes kring användarvänlighet och om hur människor uppfattar och tar till sig denna typ av information. Verktyget förbättrades utifrån de insamlade teorierna, och behövdes sedan utvärderas vilket gjordes genom användartester. Åtta utvecklare observerades när de försökte lösa givna uppgifter kring att hitta information i verktyget. Resultaten från testerna användes till att förstå verktygets styrkor och svagheter.

Det slutgiltiga verktyget kommer kunna användas för att förstå användarna i andra typer av applikationer. Undersökningen visar att det går att förstå användartyper när de är grupperade efter det rörelsemönster som blir när de har klickat i en applikation. Hierarkisk klustring har visat sig vara en lämplig metod för gruppering av användartyper baserat på användares interaktion med en applikation. Utvecklingen av presentation och visualisering av information har visat på att ett användarvänligt gränssnitt med inbyggda visuella ledtrådar kan hjälpa en användare förstå en komplex analys. De visuella ledtrådarna kan se ut på olika sätt men i detta verktyg består de av information kring vilka förutsättningar analysen bygger på, samt på extra information kring klustrena i sig. Det senare hjälper användaren tolka eller verifiera en tolkning av användarna i ett kluster.

Distribution of work

This thesis has been written by Sara Collin and Ingrid Möllerberg who, together, have worked on all areas covered in this thesis. Most of the code has been written separately, but in close collaboration, sitting next to each other discussing solutions. Pair programming has been used when needed, for example when encountering a difficult problem. Each author was given an area of responsibility. Ingrid had the responsibility of constructing the information box displaying the added information to each cluster. Sara had the responsibility of constructing the clustering reconfiguration. This, in combination with pair programming when needed, enabled developing an extensive tool in a relatively short amount of time.

The work of writing the thesis has been made in close collaboration. All texts have been reviewed by both parties until reaching agreement. This was done to the level of changing synonyms and prepositions of words.

Table of Contents

Part 1: Introduction

An introduction to the background and methodology

1. Introduction

The emergence of the mobile phone and cellular network has had a tremendous impact on our lives. With the rapid spread of wireless networks and mobile phones, we now have endless possibilities to connect and interact with others. This enables a multitude of different services and applications with the ability of connecting people in different ways and with different purposes. This has led to a rapidly expanding business area of mobile e-commerce. (Liu, 2014) E-commerce refers to selling and buying goods on the internet. Peer-to-peer is a specific type of e-commerce where individual customers deal directly with each other, without any third party involved. (Shopify, n.d.) These applications depend heavily on their users to provide the application with content. Without its users, the "shelves" would be left empty. These applications are driven by the engagement of its users, thus it is of value to understand user behaviour within the application. One way to understand the users is to identify the different types of user groups that dominate the application.

Data mining is a domain that has grown substantially in the last few years due to its ability to transform huge amounts of data into useful information and knowledge (Han et al., 2012). By analysing *clickstreams* one can get a better comprehension of how an application is used. A clickstream is a sequence of timestamped actions, such as button clicks or screen views, made by a user in an application. However, clickstreams contain a vast amount of data. The task of extracting useful information and knowledge from an immense amount of data is complicated. There is a jungle of possible algorithms and tools out there for collecting and analysing data. Algorithms are not universal and do not always produce meaningful results for all types of datasets. The studied dataset must therefore be matched with a suitable algorithm.

Even after succeeding with matching the dataset with an appropriate algorithm, the results must be presented in an intuitive and meaningful way that is possible to comprehend. Without comprehension of the results and how to interpret them, valid results will go to waste.

1.1 Purpose

The purpose of this study was to develop an intuitive interactive tool that supports identification and visualisation of user types based on clickstream data from an application. In order to fulfil the purpose of this study we chose to conduct a case study of the mobile application Plick, which is an application for selling and buying second hand clothes. The users were segmented by applying cluster analysis on their movements, clicks and interactions within this peer-to-peer mobile e-commerce application. The interactive tool was meant to visualise this complex analysis and its result in an intuitive way to enable a developer to gain a better understanding of how different types of users are using an application. The following research questions were used in order to accomplish the purpose of the study.

1.1.1 Research questions

- How can clustering algorithms be used in order to identify user group characteristics based on clickstream data within an e-commerce application?
- How can clickstream data be processed in order to perform an efficient and rewarding segmentation of users?
- How can a complex analysis be conceptualised in an interactive tool in order for an operator to use it in a correct and efficient way?

1.2 Disposition

The thesis consists of four parts. Part 1 includes three chapters. This first chapter presents the foundations of the study. In the second chapter the case studied in this thesis, Plick, and its features, users and data are described. The chapter also covers research of previous studies of Plick. In chapter three, a general introduction to the methodology used in the thesis is presented.

Part 2 consists of three chapters, covering the cluster analysis. It begins with a description of the theories used in the analysis, then the methods used in the cluster analysis is described. In the final chapter of Part 2, the different steps of performing the analysis are presented and discussed.

Part 3 equally consists of three chapters covering the construction of the interactive tool. It begins with a presentation of the theories used when creating the interactive tool. It is followed by further descriptions of the methods used in Part 3. Finally, there is a description and discussion of the construction of the interactive tool, which ends with a presentation and discussion of the usability test made on the final tool.

Part 4 includes conclusive findings from Part 2 and Part 3, and suggestions of future research that are outside the scope of this thesis.

1.3 Delimitations

In this project, a ready-made program that uses a clustering algorithm and visualises its results has been used. The primary alterations have been made in the visualisation. The clustering algorithm has mainly been left untouched. The thesis did not include an investigation of whether the algorithm could be further improved.

The assumed operators of the tool are application developers with a background knowledge of the application studied. Every design choice made has the intended operator in mind. The interactive tool is not meant to be used without any prior knowledge.

2. Background: The Plick system

In order to fulfil the purpose of this study, a case study was made on the mobile application Plick, which is an application for selling and buying second hand clothes developed by Swace Digital (Swace, n.d.). User clickstream data from the application was used as the object of study. In the following chapter the application, its users and data are further described.

2.1 Introduction to Plick

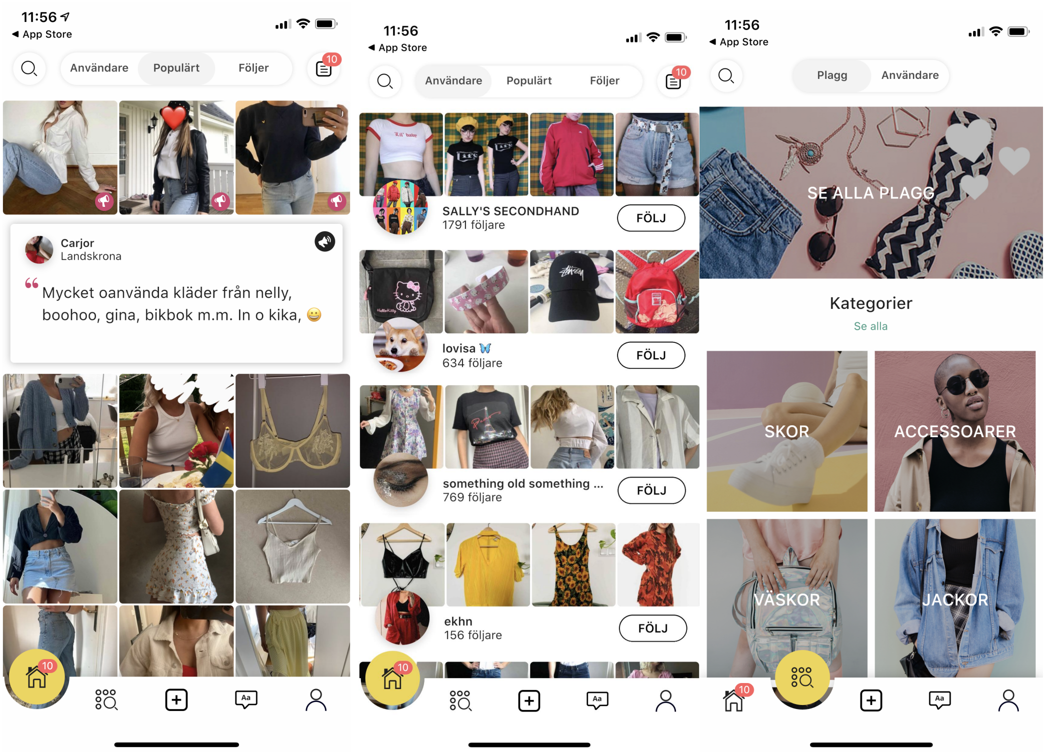
There are several platforms on the internet where individuals can exchange clothes and objects on a second hand market. The most well-known in Sweden are Blocket and Tradera (Blocket, n.d.) (Tradera, n.d.). There one can buy and sell most things, and each object has its own ad page. In contrast, Plick is a marketplace exclusively for second hand clothing and accessories. It is constructed in a similar manner to a social media platform with profiles, ads and a feed where relevant, interesting and popular profiles and ads are displayed. Plick is a peer-to-peer marketplace, which means that all communication and exchange of products goes through the users.

The biggest difference between this marketplace and its competitors is the social aspect of Plick. If a user is logged in to an account, they can like and follow other users. A user can build their brand and collect followers by engaging in social activities similar to social media applications such as commenting, liking and posting ads. The users of the application are often more active than users of other platforms and some users scroll the feed of items daily. (Heibert 2020, personal communication, 3 February)

Plick is available for both IOS and Android. The design and functionality of the application is the same for both platforms. The quota of users between the platforms is 88/12, with most users on IOS. There is also a web page but some major functions are only available through the application, such as posting ads. The application only exists on the Swedish market. (Heibert 2020, personal communication, 3 February)

2.2 System description

The application consists of several different pages with a menu bar at the bottom of the page (see Figure 2.1). The menu bar consists of five buttons. They are called feed, browse, create ad, communication and profile. When opening the application, the users are directed to the *feed* page (see Figure 2.1a). There they are first presented with the tab showing a feed with popular items. The popularity of the items on display are calculated with a formula. This formula is the result of an earlier study of Plicks user data, which is presented in chapter *2.5 Plick related work*. The images presented in this chapter are from the IOS app, version 4.0.20.



(a) Feed - popular items (b) Feed - users (c) Browse - garments

Figure 2.1: (a) Start page of Plick. Items arranged based on popularity. (b) Feed of items sorted by sellers. (c) Browsing page for viewing garments per category, or users.

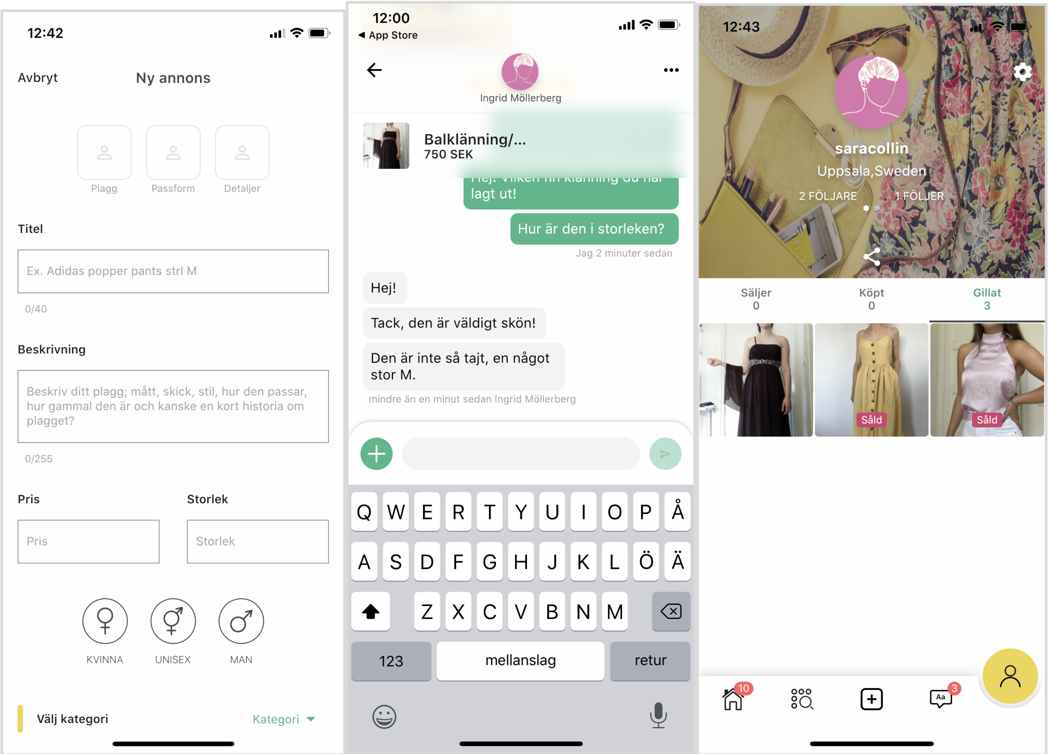
The user can apart from viewing popular ads also view items based on users (see Figure 2.1b). Accounts with many followers are more likely to be displayed higher up on this view.

In the browse page (see Figure 2.1c) there are pages for browsing items by category, or free search, where the search results are based on tags in the ads, or matches in the description of the ad.

When the user has found an item they are interested in, they can start a chat with the seller through a button on the ad's page (see Figure 2.2). The user can also like and comment on the ad. If the account is of specific interest to the user, for example if the account posts a lot of ads that the user finds appealing, the user can choose to follow the account.

Figure 2.2: One single ad. Here the user can choose to follow the account of the seller, like, comment or share the ad. It is also possible to start a conversation with the seller through the ad.

A user can upload an ad (see Figure 2.3a), converse with other users (see Figure 2.3b) and view its own profile. On the user profile, the user can see ads it has posted, items it has bought, and ads it has liked (see Figure 2.3c). The user can choose to write what city they live in. The city is displayed under the name of the profile.



(a) New ad (b) Conversation – buying (c) Profile – liked ads

Figure 2.3: (a) The screen view of creating a new ad (b) A conversation regarding the ad 'Balklänning' (c) Looking at your own profile and your liked ads.

2.3 Users

Since the launch of the application 2013, the typical user has changed. In the beginning the application attracted the male hipster culture in large cities who were searching for unusual second hand garments. Today, the ratio of men and women is strongly leaning towards women. The typical user is a young woman aged 15-30, living in a larger city in Sweden. (Heibert 2020, personal communication, 3 February)

To understand its users, Plick collects statistics of how much time and how often users spend time in the application. Users can belong to one of the categories low-, middle- or high activity users. A user is categorised as a low, middle or high activity user if they uses the application 1-2, 3-4 or 5-7 times a week. The activity levels are colour-coded by Plick in the statistics page of the application with low being purple, medium being yellow and high being green. The user's activity level affects how the feed is presented to the user. A high activity user is for example presented with a more changeable feed, whereas a low activity user is presented with more high-quality ads in order to keep their interest for the application. High-quality ads are ads that have gotten a lot of attention and have been proven popular among more frequent users. (Heibert 2020, personal communication, 3 February) (Alex 2020, personal communication, March) (Rask 2020, personal communication, March)

2.4 Data

The application is connected to Google's mobile and web application development platform Firebase that has functionality for analysing usage of apps. Firebase saves data from the front end of the application and this data is structured around what screens a user is looking at. Each screen view contains multiple data points such as an ID of the user performing the action, timestamps of the activity, device and application information as well as geoinformation. Firebase groups all actions that belong to the same session and assigns the group a session ID. The start of a session is defined by when the user moves the application to the foreground and the end is defined by when the application is moved to the background of the cell phone. The session also ends if the application has been untouched in the foreground for 30 minutes. (Support Google, n.d.) Firebase also contains information regarding, among other things, users' ads, likes, comments, account creation date, in-app-promotions and conversations. Data is additionally stored in a PostgreSQL database. In this study, only data from Firebase has been used.

The data from Firebase is stored in a format owned by Google called "big query". It can be downloaded in smaller chunks as JSON files. The dataset that has been used while developing the tool is 134.6 MB, and includes 535 543 user actions. How the data was collected from Firebase for the project is described in section *6.2 Tools used*.

An action saved in Firebases' database has 124 variables sorted in up to three layers of nested dictionaries. In this project, eight of the 124 variables describing the action have been used (see Figure 2.4). In appendix A the names of the data points used in this project, with an example of a data point, and its meaning is presented.


```
{"user_id":"100872","date":"20200310","timestamp":"1583829858194316","previous
_timestamp":"1583829855823004","event_name":"screen_view","screen":"feed/tab1"
,"previous_screen":"myprofile/tab0","session_id":"1583829847"}
```


Figure 2.4: The data points extracted from each action.

There are many different types of screen views and other events in the application. The most common screen view is looking at a single ad. Approxematly 53 % of all actions belong to this screen view. Occurrences of the most common screen views are presented in appendix B.

There is a lot of information in the databases but there is no existing way of understanding what the information tells about user activity and movements of the users. It is left to explore how users are using the application, which leads to the next section about earlier studies of the system.

2.5 Plick related work

In order to analyse which method could fit the Plick case, a study of previous work related to the Plick application was carried out. Two Plick related studies conducted in the past have been analysed and are more thoroughly explained in the following two sections.

2.5.1 Developing a Recommender System for Plick

The first study was conducted by Adam Elvander in 2015, a Master's student in the Programme in Sociotechnical Systems Engineering at Uppsala University. He wrote a Master's thesis about how to create a recommender system based on user data. The studied user data mainly consisted of clicks made by users on items in the application. The recommender system was meant to be implemented in the Plick application in order to personalise the ads feed, which at that time was chronological. Different recommendation algorithms were examined. Only three algorithms were considered suitable to implement and test. This was due to the structure of the user data from Plick where the number of items in the system exceeded the number of users and the items were greatly differentiated from each other. It was shown that a user-based collaborative filtering algorithm resulted in the most suitable recommendations. The algorithm dedicates a user to a neighborhood of users based on their similarities. Based on data from the neighborhood, a set of popular items within the neighborhood is calculated and presented to the user. The downsides of using a collaborative filtering algorithm in an e-commerce marketplace is firstly the "cold-start" issue. This means that there is not enough historical data to build the analysis upon. The other downside is the issue of data sparseness. It is highly unlikely that a user would interact with all items and the user-item rating matrix is therefore largely empty. In time, as users interact with more items, the user-item matrix will fill up. At last, there is an issue of the burying effect. Older items will be buried behind newly posted items. Even if the recommendation filtering might mitigate the burying effect, it will not solve it. Also, older items that lack view data will unlikely be introduced into the recommender system and thus they are unlikely to be found by users. (Elvander, 2015)

2.5.2 Plick Customer Segmentation

The second study was carried out by Andrew Aziz in 2017, a Master's student in Computer and Information Engineering at Uppsala University. The aim of the Master's thesis was to segment users into smaller subsets based on their clothing preferences and to implement a recommendation component in the Plick application based on the final segmentation. To segment the users based on clothing preferences, a cluster analysis was carried out based on user preference data from Plick. To establish user preference, three types of preference were measured. If a user viewed an item, it was counted as a user preference, and a "like" of an item indicated an even stronger preference of that item. At last, a started conversation with the seller indicated the strongest preference since the user showed behaviour of a potential buyer. Based on user preference data, a k-means cluster analysis was carried out. K-means is a clustering technique that groups data points based on their closeness to some predefined centroids. The technique is further described in section *4.2 Clustering Algorithm*. The cluster analysis could not detect any clear cluster separations. Thus, to get a better picture of the data, each user was assigned a favourite brand based on the amount of views on brand items and the users were then grouped based on their favourite brand. Because of the unclarity of the cluster analysis results, a visualisation of the results was created to get a better picture of the data and how it could be used to create better user segmentation. The result of the study showed that, in order to get a clearer customer segmentation, a more extensive analysis of how the data should be pre-processed and weighted should be carried out. It also suggested that other clustering algorithms should be examined since k-means clustering was the only algorithm used. Finally, the study stressed the importance of filtering the users and brands used in the clustering algorithm. (Aziz, 2017)

3. Methodology

In order to gain insight into how to create and visualise a complex data analysis in an intuitive way, a variety of methods have been used. The following section describes the different methods and tools used in order to understand the importance of a visualisation tool, how to pre-process the data and how to perform and analyse a cluster analysis.

3.1 Interviews

To understand how a visualisation tool of user engagement in the application can aid the application developers, interviews were carried out. Two developers of the app, the CEO of Plick AB and the author of one of the studies mentioned in *2.1 Related work,* were interviewed. To maximise the usability for the developers of Plick, it was necessary to establish what data presentation and features the visualisation tool should include. The interviews varied from unstructured to semi structured and from formal to informal conversations.

The interview with the developers included questions regarding information, objectives and values about the application. The interview with the CEO of Plick was even more focused on questions about the values Plick generates and the vision he has for the application. The author of the previous study of Plick was asked questions about his research process and lessons learned that could help our process. For example, questions about the structure of the data and the programs and programming language used were discussed. The informal conversations with the CEO of Plick continued throughout the project. During these, the progress of the tool was presented, followed by feedback and questions from the CEO. The main results of the formal and informal interviews are further described in chapter *6.1 Establishing the purpose of the tool*.

3.2 Performing the analysis

A literature study was conducted in order to find a suitable method for clustering users based on clickstream data. A clustering algorithm tool called Recursive Hierarchical Clustering (RHC), developed by Wang et al (Wang et al., 2016), was considered the most suitable to implement. In order to implement the tool, the user data from Plick had to be pre-processed into the format required by the algorithm. The clustering algorithm, the choice of using RHC and the pre-processing is further described in *Part 2: cluster analysis*.

3.3 Designing the interactive tool

The program developed by Wang et al (Wang et al., 2016) included a visualisation of the resulting clusters in addition to the clustering algorithm. This was used as a base for the presentation of the clustering result. RHC with our changes in visualisation and functionality is termed “the visualisation tool”, or simply “the tool” in this thesis. The original visualisation of the clusters is further described in chapter *7.4 Recursive*

Hierarchical Clustering. A literature study was made in the process of changing the tool in order to fulfil the purposes of this study. Theories about how to conceptualise complexity and how to help the operators comprehend presented information, were studied. To check whether the visualisation tool fulfilled its purposes, usability tests were made. The design of the interactive tool is further described in *Part 3: The interactive tool*.

3.4 Tools used

Some tools for accessing and pre-processing the data have been used. Plick is connected to Googles' tool Firebase. Firebase stores, amongst many other things, user data and tracks triggered events in the application as described in section 2.4 *Data*. This data was accessed by a tool provided by Google named BigQuery, which allows running SQL-commands in a browser. The resulting BigQuery tables were exported to Google Cloud Storage as csv and JSON files and downloaded locally. The files were read and processed in the scripting language Python. They were then translated into a text format used by the RHC. The final visualisation tool was a further development of the preexisting RHC. This tool uses JavaScript, CSS and HTML. Python was used as the primary language to execute the clustering algorithms. To visualise the clusters, a JavaScript library for visualisation named D3 was used. All work has been structured using Git.

3.5 Ethics

Morality within information technology can be a delicate subject and there are a multitude of studies available covering the subject. In this thesis, a definition of developers' moral responsibility phrased by Anton Vedder in *The handbook of information and computer ethics*, have been used. He was at the time of the publishing of the book an Associate Professor of Ethics and Law at Tilburg University. The handbook is composed by many professors in different areas, including information techniques and philosophy. The chapter named *Responsibilities for information on the internet* is used in this thesis. According to Vedder, the moral responsibility of developing programs and using personal data depends on three criteria. Firstly, for the developer to be morally responsible for an action, there must be a causal relationship between the developer and the consequence of the action. The causal relationship can be either direct or indirect. Secondly, the action and its consequences must be brought about intentionally by the developer. An action and its consequences is seen as intentional if the developer does not openly oppose the action or its consequences. Lastly, if the action is not morally indifferent, a consideration of the morality of the action and its consequences must be taken into consideration. (Vedder, 2008)

The Stanford Encyclopedia of Philosophy presents a discussion of privacy and information technology. The Encyclopedia is only updated by appointed persons elected by Stanford University Department of Philosophy, and all entries are peer-reviewed before publishing. This Encyclopedia is another framework to help navigate in the field of ethics in computer science. For instance, it discusses four situations where a developer has moral reasons to protect personal data in order to prevent harm to the user. Firstly, personal data should be protected because it could lead to access to users' accounts. If someone unauthorised gets access to a user's account it can cause harm to the user by making changes in the content or by acting in the name of the user. Secondly, since a human is more complex than any analysis a computer can mirror, data about the user can be used in an incorrect simplified way that does not reflect the real user. Thirdly, the reason a user might give approval for using personal data might not apply in another situation. The personal data shared by the user in one situation might be harmful for the user in the hands of some other party. Lastly, profiling of users based on personal data are discussed to have the risk of aiding in discrimination. A common example is insurance companies using profiling when deciding what insurance to offer an individual. (van den Hoven et al., 2014)

The visualisation tool can be used to understand how users use an application. In this project it is used in order to understand the usage of the application Plick. It is for example possible to understand what type of usage is common among high activity users. The intention for creating the tool is to help the developers of Plick create an application that is usable and enjoyable for the users. However, the tool could easily be implemented to fit other applications. This means the tool could be used for purposes and intentions different from those that have been investigated in this case study. These intentions might not always benefit the users that are analysed. For example, the tool could be used in order to identify user groups that are easier to manipulate with deceptive ads.

To make sure that the individual users of the application are anonymous, all personal data from Firebase and the data from PostgreSQL are anonymised. In PostgreSQL, names and birth dates of the users cannot be accessed, neither during the work with this thesis, nor in daily development. The email addresses are also given pseudonyms and the hash of the passwords are not shown. However, not even the anonymised version of this information is taken into Firebase, which means there is no way of finding out which user is connected to a specific user ID through the tool. There is of course a risk that someone with access to the original data can combine the anonymous information in order to identify specific users. This could happen within Plick or any other application the tool is used for. In the case of Plick, this would mean breaking laws of data security and thus seems unlikely. It is however important to state that the risk exists. (Vogit and von den Bussche, 2017)

The anonymisation of the data is enough for making sure the interests of the users are satisfied. The identification of user groups in the visualisation tool does not facilitate any improper use of personal data as described by the Stanford Encyclopedia of Philosophy. The visualisation does not make it easier to hijack any user's account in Plick or their email-service and thereby causing the user harm. Nor does the tool use the information in a different type of context such as if the information was connected with memberships of retailer stores that would use the information to direct advertisement. The tool does generalize users into categories of user types. As the rest of the thesis will describe, the algorithm of the clustering was chosen partly because it does not put any label of behaviour from the developers on the users in the segmentation process. The tool is grouping the users based on their clickstreams, and the result has to be analysed to understand what attributes the users have in common. This means no user is forced into a predefined mold of user types, and the complexity of the behaviour of the user behind the clickstream is preserved.

As can be seen, there might be a causal relationship between the developers of the visualisation tool and the effect the application of the tool might have on the users analysed. The implementation of the tool must be regarded as intentional. When developing the visualisation tool during this thesis, the intention is for the tool to increase the usability and enjoyment for the users in the application analysed. Although, unethical implementation by operators is not possible to prevent. According to Vedders' definition, this work has moral responsibility to what it produces. Therefore we, as developers, have made the efforts described in this section to not violate the different ways of breaching personal data stated by the Stanford Encyclopedia of Philosophy.

3.6 Sources of Error

An effort has been made to describe both the clustering algorithm and the visualisation of the clusters in RHC in general terms. The article and the code itself have been thoroughly studied in order to reach full understanding. Since RHC is quite extensive, only the most necessary details about the algorithm and visualisation will be presented. The full description of RHC is found in the article *Unsupervised Clickstream Clustering for User Behavior Analysis* written by Wang et al (2016).

Part 2: Cluster analysis

The design problem: complexity in analysis

4. Theory

The research question of this thesis is the question of segmenting users into user groups. Clustering, as has been mentioned previously, is a mathematical method for classifying objects based on some predefined similarity measure. This chapter will describe relevant algorithms available for clustering users and methods used to increase the quality of the result. These make up a complex, but necessary, process of calculation to reach the goal of segmenting users and understanding their aims and intentions.

4.1 Pre-Processing

Today, databases often include enormous volumes of data that are susceptible to inconsistent, missing or noisy data (Han et al., 2012). These large amounts of data are causing trouble for data mining methods such as clustering (Aggarwal and Reddy, 2013). In order to gain high-quality clustering results, high-quality data is necessary. Data quality is defined by several factors including its *accuracy*, *completeness*, *consistency*, *timeliness*, *believability* and *interpretability*. Inaccurate data might result from faults in the data collection tool or from inaccurate data entries in either computer or human errors. Computer errors can be a result of faulty instruments. Incomplete data can result from attributes not being considered important at the time of data collection or entry. Inconsistent data can derive from inconsistencies in naming the data or inconsistency in the format of the input fields. Timeliness does influence the quality of data. The times of the data entry might be asymmetrical, resulting in large variations in the data set. For example, there might be a latency in data entry that has to be considered when collecting data. Finally, believability and interpretability, which also influence the data quality, reflect how the data is trusted by the users and how easily the data is understood. (Han et al., 2012)

Pre-processing has the ability to increase the quality data and therefore the quality of the resulting clusters, as well as the time it takes to extract the clusters. Pre-processing includes for instance *data reduction* which includes *feature selection* where the data size is reduced by eliminating redundant features or by grouping features and *data transformation* where the data is transformed in order to improve the accuracy of the clustering method. (Han et al., 2012)

One might think that the more variables the dataset contains the better, but that is not always the case. Sometimes, an overflow of variables can be problematic. For example, two variables have four possible combinations. By increasing the dimension from two to three, the possible combinations increase from four to 27. This problem is often referred to as the “the curse of dimensionality”. As the number of variables increases, the possibility of sparseness in the dataset rises. A sparse dataset is a dataset with a lot of missing data points. The more sparse a dataset is, the more data is needed in order to find all possible variations. Also, with more variations, fewer items will be categorised in each group. A larger magnitude of variables also makes it more difficult to detect outliers. A balance between how to gain most insights without lowering the quality data must be made when choosing what variables to include in the analysis. (Altman and Krzywinski, 2018)

The most common pre-processing technique is *feature selection*. *Feature selection* is used in order to increase the quality of the underlying clustering by removing unnecessary and noisy features. *Feature selection* means that parts of the dataset is removed, or grouped, before analysis. The choice of what to remove can be based on different values but most often it is based on relevance. *Feature selection* is a technique often used in applications regarding pattern recognition. (Aggarwal and Reddy, 2013)

4.2 Clustering Algorithm

Clustering can be defined as partitioning a set of data points into a set of groups based on similarity within the group and dissimilarity between groups. Classification requires a costly collection and labelling of a large training dataset. Clustering on the other hand first partitions the dataset into groups based on similarities, and then assigns labels to a few of the subgroups. (Han et al., 2012) It builds its results on the data rather than "learning by examples" as Han et al. writes. Clustering is an unsupervised classification method since it does not require any predefined assumptions about the groups, as supervised classification methods do. It does not provide any explanations of the result either since it does not require any labelling of the clusters. (Han et al., 2012)

There are several different clustering algorithms. Two well-known techniques are k-means clustering which belongs to the family of partitional clustering techniques and hierarchical clustering techniques. (Bandyopadhyay and Saha, 2013) One difference between these two clustering techniques is that k-means provide no result of the clustering until the whole algorithm is done. A hierarchical clustering technique gives a result of the clustering after each iteration. (Bandyopadhyay and Saha, 2013)

K-means clustering is one of the most established and used clustering algorithms in the domain of data clustering. The method's popularity stems from its simplicity in practical implementations. (Aggarwal and Reddy, 2013) K-means assigns points to a cluster based on its closeness to several centroids, which are predefined both in their amount and attributes. This method is however sensitive to outliers in the data and the method produces clusters that often are diffuse and hard to read. (Aggarwal and Reddy, 2013)

Hierarchical clustering algorithms group data objects into a tree of clusters. There are two types of hierarchical methods: *agglomerative* and *divisive*. *Agglomerative* hierarchical clustering algorithms start by placing each object in its own cluster and then merges these individual clusters into larger ones based on similarity. The algorithm stops when all objects have been merged into one single cluster, or until some stopping condition has been satisfied. This is a bottom-up method since it starts with the individual objects and works its way up to a single cluster with all objects. Agglomerative clustering are the most common algorithms used in hierarchical clustering methods. *Divisive* clustering on the other hand, are top-down methods which starts by placing all objects in one cluster. It then divides the cluster into sub clusters until each object has its own cluster, or until some stopping criterion has been satisfied. (Han et al., 2012)

Hierarchical clustering can sometimes provide challenges when deciding on a suitable splitting or merging point for the clusters. A badly placed split or merge can lead to low-quality clusters. The choice of splitting and merging points is of great importance to produce high-quality clusters. Hierarchical clustering techniques do not require any predefined number of final clusters, as k-means do. (Han et al., 2012)

Hierarchical clustering processes are often represented in so-called dendrograms. It shows how objects are grouped step by step in a tree structure. (Han et al., 2012) Hierarchical clustering has the benefit of the dendrogram over partitional k-means clustering, as the dendrogram works as a powerful visualisation of the clusters. It is easy to follow along in a hierarchical clustering algorithm since it can be stopped and traced back at any point. (Aggarwal and Reddy, 2013)

4.3 Similarity measures

Hierarchical clustering algorithms partition users into groups based on similarities between clickstreams. Therefore, a way of measuring similarity or dissimilarity must be defined. Similarity measures depend on the data being studied, thereby the similarity measure must be chosen with the specific case in mind. (James et al., 2013) Dissimilarities between objects are stored in a similarity matrix (Han et al., 2012). The similarity matrix is the base for both clustering and classification (see Figure 4.1). It is through the matrix that characteristics of the clusters can be found. (Bandyopadhyay and Saha, 2013) Each row and column represents an object, where d(i,j) represents how similar objects i and j are. The more similar objects are, the closer to 0 d(i,j) becomes. (Han et al., 2012) In this thesis, d(1,2) represents the similarity of user 1 and 2:s clickstreams.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Figure 4.1: Similarity matrix (Figure 2.9 in Han et al., 2012).

Euclidean distance, which is the most common distance metric, uses a comparison of magnitude to measure distance between data points. This metric is not well suited for sparse datasets. In the domain of clickstream clustering, the distance between two clickstreams, i.e. the similarity, can instead be computed using a normalised polar distance, which is also called angular distance. This measurement is a good choice when there are sparse vectors in the data, such as in clickstreams, since it compares direction rather than magnitude. (Wang et al., 2016)

A clickstream is formalised as a sequence S = ($s_1 s_2 \ldots s_j \ldots s_n$). S_j is the j^{th} action and n is the total number of actions in the sequence. K consecutive actions in a sequence is called a k-gram. For example, (A B) is a 2-gram since it is two consecutive actions. T_k is defined as all possible k-grams, in a sequence S

$$T_k(S) \; = \; \{k\text{-}gram \mid k\text{-}gram \; = \; (s_j s_{j+1} \ldots s_{j+k-1}), j \; \in [1, n+1-k]\}. \quad (1)$$

For example, the sequence S = (A B A B) contains the 2-grams (A B) and (B A) and all possible 2-grams (T_2(S)) in S is {(A B) (B A) (A B)}. The distance between sequence 1 and 2 are calculated by first looking at common k-grams between sequence 1 and sequence 2

$$T_{1,2} \; = \; T_k(S_1) \cup \; T_k(S_2). \quad (2)$$

Within each sequence S (S = 1,2), the normalised frequency of k-grams is represented by a vector

$$[c_{S1}, c_{S2}, \ldots, c_{Sn}], \text{S} = 1{,}2 \quad (3)$$

where $c_{S1}$is the count of the first k-gram in sequence S and $n \; = \; |T_{1,2}|$. Thereafter, the distance between sequence 1 and 2, $D(S_1, S_2)$, can be computed as the normalised polar distance between the two vectors $[c_{11}, c_{12}, \ldots, c_{1n}]$ and $[c_{21}, c_{22}, \ldots, c_{2n}]$,

$$D(S_1, S_2) \; = \; \frac{1}{\pi} cos^{-1} \frac{\sum_{j=1}^{n} \quad c_{1j} \times c_{2j}}{\sqrt{\sum_{j=1}^{n} \quad (c_{ij})^2} \times \sqrt{\sum_{j=1}^{n} \quad (c_{2j})^2}}. \quad (4)$$

Similarities between two sequences are represented as a low value on $D(S_1, S_2)$, that ranges between 0 and 1. Similarities between sequences are placed in a similarity matrix as described in Figure 4.1, where each row and column represent a sequence. (Wang et al., 2016)

4.4 Iterative feature pruning

In our study we have applied the method for unsupervised clickstream clustering for user behaviour analysis explained by Wang et al (2016). They developed RHC, an unsupervised system, which was based on clickstream data and visualised dominant behaviour. The algorithm identified clusters of users based on similarities between their clickstreams. A divisive hierarchical clustering algorithm was used to partition the similarity graph of the users. To capture fine grained user behaviours, an iterative feature pruning was implemented which has the capability to capture smaller changes in behaviour amongst users. (Wang et al., 2016)

RHC starts by calculating a similarity graph of all users based on the full set of features given to the program to analyse. The top-level clusters are retrieved by partitioning the similarity graph, as described above. Then, the top-level clusters are pruned of their dominant features to capture the more fine-grained subclusters within. These are called lower level clusters. Wang et al. uses a classic measure called χ^2-score (chi-square) (Yang and Pedersen, 1997) to select the topmost prominent features in a cluster. The χ^2-score measures the features' ability to separate data in different instances. The higher the score, the better discriminative power the feature has. The score is unreliable when using small datasets, but clickstream datasets are usually large. (Yang and Pedersen, 1997) The remaining features are then used to compute a new similarity graph for the subcluster and is thereafter pruned again. The program makes use of polar distance in order to calculate the similarity graphs. Finding the key features of the parent clusters is one of the key steps in feature pruning. Iterative feature pruning is used in order to capture smaller differences between clusters within clusters. The algorithm stops when the clustering quality has reached a minimum threshold. The measure used for clustering quality is modularity. Modularity measures the density of edges inside the cluster and compares it to the density of edges outside the cluster. Modularity ranges from -1 to 1, where 1 indicates better clustering quality. Wang et al. used the modularity threshold of 0.01 for deciding when a partitioning of a cluster should stop. (Wang et al., 2016)

The quality of the clusters was examined and compared to other clustering methods such as the k-means algorithm. Wang et al. could see that their approach reached a higher accuracy in detecting user groups with similar behaviours than for example k-means. (Wang et al., 2016)

5. Method

RHC was used in the tool created in this thesis to cluster users based on their clickstreams. As mentioned in section *4.1 Pre-processing*, low-quality data results in low-quality data analyses. Therefore, the data had to be pre-processed to increase the quality of the dataset analysed and thus produce high-quality results. The usage of RHC to cluster the users, and the pre-processing of data is further described in the following sections.

5.1 Pre-processing of Data

Three main methods were used in order to pre-process the data. First, a study of the original data set was carried out. As described in section 2.4, the data is stored in Google's instrument Firebase. To understand how the data was stored and how to use it, the database was studied using Google's instrument BigQuery. There, we could inspect the structure of the database and its content. After the data had been exported from the database into the visualisation tool, a further study of some of the more important features was carried out. Histograms were used in order to gain a deeper understanding of the distribution of some features such as how often events and screen views occurred, and the time spent on each event or screen view.

Secondly, a technique called *feature selection* was used to increase the quality of the dataset. The number of features from the original dataset were reduced to the dataset used for the analysis. The result of the *feature selection* is further described in section *6.3.1 Data reduction.*

Lastly, the dataset was *transformed* into a format required by RHC which enabled the clustering analysis.

5.2 Performing the analysis

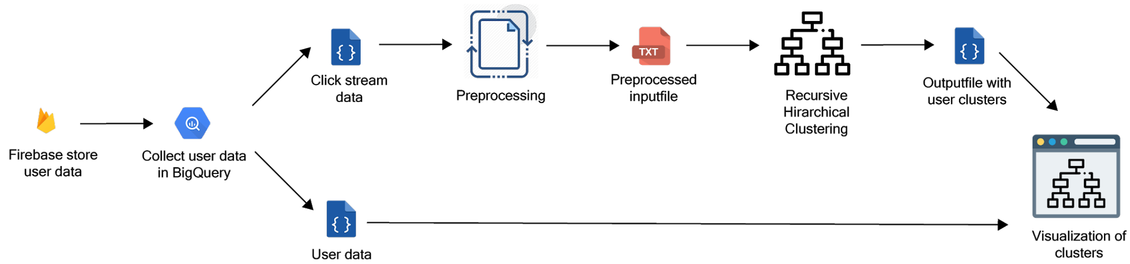
The clustering was performed by the RHC algorithm described in section *4.5 Iterative feature pruning.* In order to use RHC, the user data had to be transformed into a required format. This process is further described in section *6.3.2 Data transformation.* This transformation was done using a Python script.

5.3 Sources of error

When dealing with a vast amount of data, things can easily be overlooked. A study of the database was made in order to gain an understanding of the structure of the data and to support decisions that were made in the *feature selection*. It is still possible that some features have been wrongly included or excluded. *Feature selection* is nevertheless a crucial part in data pre-processing as a way to increase the quality of the dataset. Without the *feature selection*, the quality of the analysis could be lower.

6. Result and Discussion

Flowchart 6.1 models the flow of data through the tool. The application usage data is stored in the Firebase database. Through BigQuery SQL-queries, the information is saved in various JSON files. One of the files, “clickstream data”, consists of all actions made in the time interval defined by the SQL-query. This information is processed and formatted to a file readable by RHC. RHC calculates the clusters and saves them as JSON files. RHC reads the files in order to visualise the clusters. This chapter contains a presentation of the purpose of the tool, followed by a presentation of the pre-processing and the cluster analysis within RHC.



Flowchart 6.1: A model of the flow of data through the tool.

6.1 Establishing the purpose of the tool

During interviews with the CEO and developers of Plick, the intention of the application Plick was discussed in order to identify how the tool should be constructed to reach this goal.

Throughout the interviews, several desired features of the tool were discussed. Until now, the primary measure of activity in the application has been through measuring the activity levels of the users. The activity levels consist of three groups: high (using the application 6-7 times a week), medium (using the application 3-5 times a week) and low (using the application less than 3 times a week). The desire to further understand these groups, what they do and how users can be encouraged to move from a lower activity level to a higher, was expressed. The possibility of finding and observing the features of groups with as many high activity users as possible was discussed.

The Plick application was originally created in order to change buying and consuming behaviours and to encourage a more environmentally friendly second hand market. The main purpose of further development of the application is to create a strong community with a high activity level among the users. The intention is for the users to engage with the application and each other. Turnover of clothes or profit does therefore not seem to be the main purpose of the application.

6.2 Selection of RHC

In this project, RHC has been used as the primary tool for both clustering users based on clickstream data and to visualise the clusters. RHC has, as mentioned before, multiple major advantages over other methods studied, such as the k-means algorithm. The partition process in RHC leverages the iterative feature pruning in order to capture the natural hierarchies within the user clusters. The hierarchies can then be used in order to produce and visualise intuitive features that are distinctive to the user cluster. Hierarchical clustering algorithms have the ability to visualise the clustering in multiple steps. When using k-means for example, one can only see the final clusters. K-means also produces clusters that are more diffuse and harder to read, in contrast to hierarchical clusters that are more visually accessible. Another advantage of a hierarchical cluster algorithm is that the number of clusters does not have to be defined before running the algorithm. Neither does it require labelling of the clusters beforehand. This is good when trying to understand a dataset without bias.

As described earlier in section *4.4 Iterative feature pruning*, the methods used by RHC give a higher accuracy when identifying user groups with similar behaviours than other methods such as k-means. This supports the decision to use RHC to cluster users based on clickstream data.

6.3 Pre-processing

In order to increase the quality of the data analysed by the clustering algorithm, different steps of pre-processing were carried out. The following sections describe the process of studying the data in order to select among features, the process of *feature selection* in order to reduce the data and *data transformation*.

6.3.1 Data reduction

As described in *2.4 Data*, there are 124 variables saved per action in the database. This motivates *data reduction* to process only the needed data points for each action, making the data less sparse. A more compact dataset makes it easier to detect outliers. Therefore, as a first step in the pre-processing, only relevant variables were collected using SQL. This feature selection during the data collection phase is represented by the step called "Collect user data in BigQuery" in Flowchart 6.1. Feature selection is an important part of *data reduction* and is necessary in order to handle the curse of dimensionality.

The SQL-query filtered the data table in three ways:

1) For each action, the eight columns of information described in section *2.4 Data* were fetched.
2) Actions that belong to a user without a user ID were not included. This could for example be users using the application without being logged in.
3) Some events that were considered irrelevant for the analysis, were not included. Excluded events are presented in Table 6.1.

Table 6.1: Events that were excluded from the analysis because of their irrelevance.

Events	Reason for exclusion
app_exception	The analysis only intended to include actions that happens inside the application
app_update	The analysis only intended to include actions that happens inside the application
first_open	The analysis only intended to include actions that happens inside the application
force_touch_press	The analysis only intended to include actions that happens inside the application
login_sucess	Whether a user has successfully logged in is uninteresting since only users with active accounts were studied
login	Whether a user has logged in is uninteresting since only users with active accounts were studied
os_update	The analysis only intended to include actions that happens inside the application
session_start	It is not unique to any user group. All users with a session ID have started a session
user_engagement	User engagement is a duplicate of screen view
ad_not_sold	It was excluded since it at the time was considered irrelevant for the analysis. The removal of this event is, however, questionable. It could very well have been included.

Users who were not logged in were omitted because it is impossible to match what actions are done by the same user since they have no user ID. It is necessary for the analysis to group actions per user. Also, the users using the application without an account do not have the possibility of using all the functionalities of the application. For example, they cannot upload ads.

After collecting the data, the dataset was further reduced by grouping similar views. This process is represented by the step "Pre-processing" in Flowchart 6.1. Screen views were grouped since there are many different screen names. Some of them are in fact the same view when using the application, but the user can reach it from different places. The screen names of such screens were changed into the same name so that they for all purposes of the analysis are counted as the same screen. An example of a single screen view with multiple names is the news page in the application. This screen can be accessed from all four tabs: feed, browse, conversations and profile. A screen view name is in the database always preceded with the name of the parent tab. The screen view can consequently have the names "feed/news-page", "browse/news-page", "conversation/news-page" and "profile/news-page", even though it is the same screen view. After the pre-processing they were called "news". No events were grouped. Besides merging screen views as described above, some renaming was made to single screen view or event names simply to make them easier to understand.

6.3.2 Data transformation

When all information had been collected, the data had to be *transformed* into the format required by RHC. This format is represented by the text file "*pre-processed input file*" in Flowchart 6.1, the model of data flow through the visualisation tool. The required format is a text file with one line for each user's actions. The row should contain all the k-grams, also called an "action pattern", and their frequencies in the user's clickstream, as described in section 4.2 *Similarity measures* (see Figure 6.1). The example displays two action patterns made by a user with user id 1. The user has in the first action pattern looked at an ad for between 1-5 seconds, which is represented by the timebucket named "2" (time buckets are further described in section *6.3.3 Choosing time buckets)*. Then the user went to the feed of ads showing popular ads and looked at it for an undefined time, which is represented by the timebucket named "0". This first action pattern occurred four times for user 1 in the analysed dataset.

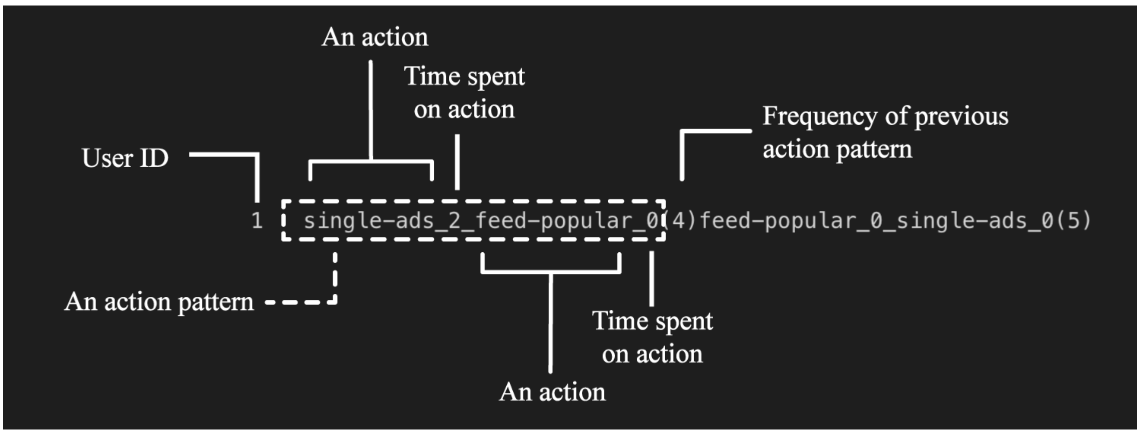


Figure 6.1: The input text file format required by RHC, with explanations.

To create this file, six steps of *data transformation* was executed:

1) *Data reduction:* Group screens to desired screen names.
2) *Data reduction:* Group events and screen views by user ID.
3) *Data transformation:* Replace old user ID's with new user ID's that fit RHC's format (from 1 to the total number of users) - with the original ID:s' saved in a text file.
4) Calculate the time of the event or screen view and add the corresponding time bucket to the action.
5) Concatenate the pre-decided number of actions in row and calculate how many times the sequence appears for each user.
6) *Data transformation:* Append the screen view/event sequence and their count in a text file, in the format that fits RHC. See Figure 6.1.

The data collected from Firebase is structured with information per action, in other words, it is structured per screen view or event, not per user session. Because of this, a control mechanism was built to construct clickstreams consisting only of actions made in the same activity session. One activity session is supposed to be one occasion of using the application. If the user puts their phone away with the application open for more than 30 minutes, the session ends. In case the user brings the application to the background in the mobile the session ends as well. An activity session is defined as actions having the same session ID. If a session ID is missing, actions are placed into the same session if they are made less than 30 minutes apart. Approximately 99.91% of the actions in the database were found to have a session ID. The actions without a session ID are cases of *incompleteness.*

6.3.3 Choosing time buckets

RHC can cluster clickstreams based on how much time a user has spent on each screen. Since time is a continuous value, it must be translated into discrete time buckets that the program can handle. After choosing what data to build the analysis on, a study of the actual data was conducted to decide the default parameters for time buckets required by RHC. The size and number of these time buckets were decided after an analysis of the actions performed by the users. All actions' time duration was plotted into histograms. The histograms show the frequencies of time duration. By studying the patterns in the histograms, a decision of where to put the boundaries of the suggested time buckets was made.

The results show that most times are close to 0 (see Figure 6.2 and 6.3). Looking at the area larger than 1 second in the histogram helps to define the rest of the interesting areas (see Figure 6.4).

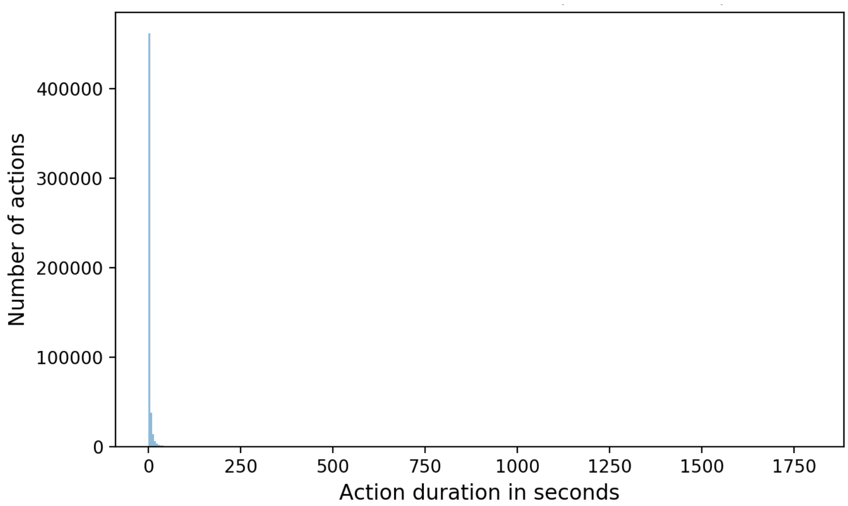


Figure 6.2: Histogram displaying time spent on event and screen views. Action times from 0 - 30 minutes

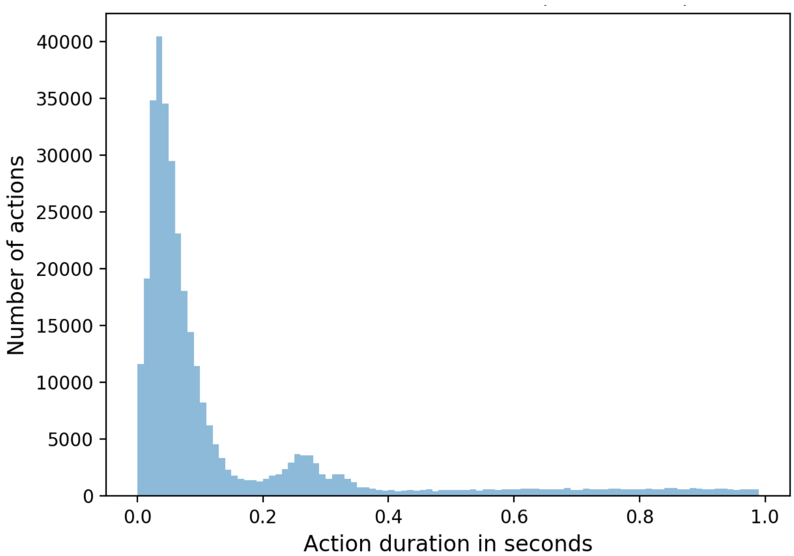


Figure 6.3: Histogram displaying time spent on event and screen views. Action times from 0 - 1 second.

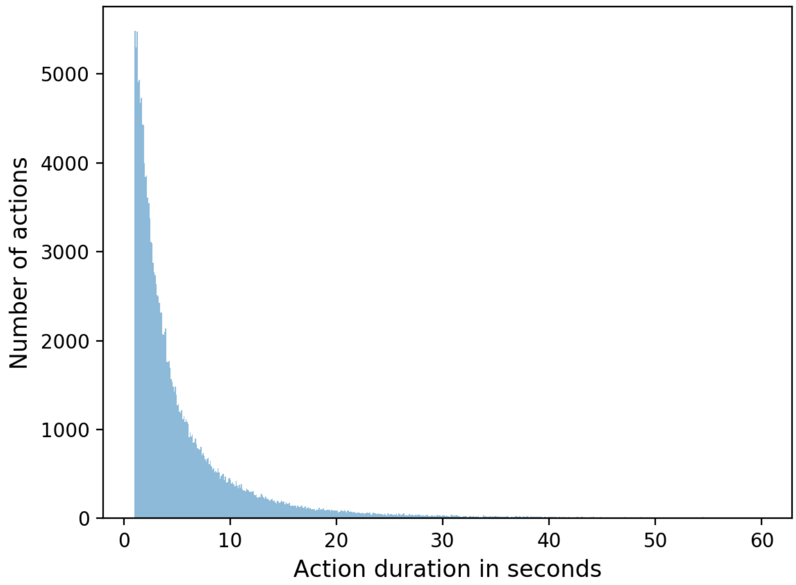


Figure 6.4: Histogram displaying time spent on event and screen views. Action times from 1 - 60 seconds

It was decided to split the time longer than one second into five parts. The lower limit of the largest bucket is chosen to match Firebases limit for sessions described in *2.4 Data*. In total there are seven buckets. The resulting time buckets are found in Table 6.2

Table 6.2: Time buckets and their corresponding time durations.

Bucket	Duration	Example
0	No time	All events and all screen views with no time duration.
1	<= 1	A screen view the user has to go through to reach another screen.
2	(1s-5s]	Looking at an ad. For example when going through many ads quite fast.
3	(5s-30s]	Scrolling the feed of filtered ads.
4	(30s-4min]	Conversing with another user.
5	(4min-30min]	Creating an ad.
6	> 30min	For example when the user has left their phone with the app open. It is defined as the end of the session.

The time buckets are represented by a discrete number, so that the first bucket is called 1, and the last is called 6. There is also a bucket 0, with 0 being assigned to the screen view in some special cases. Some screen views have no time duration because of missing data and some screen views have no time information at all. If a screen view has no relevant time information, its bucket is set to 0. Events have no time duration and are given bucket

0, and they represent a shift of state, such as following or unfollowing another user. The distribution of action times in the time buckets are presented in Figure 6.5.

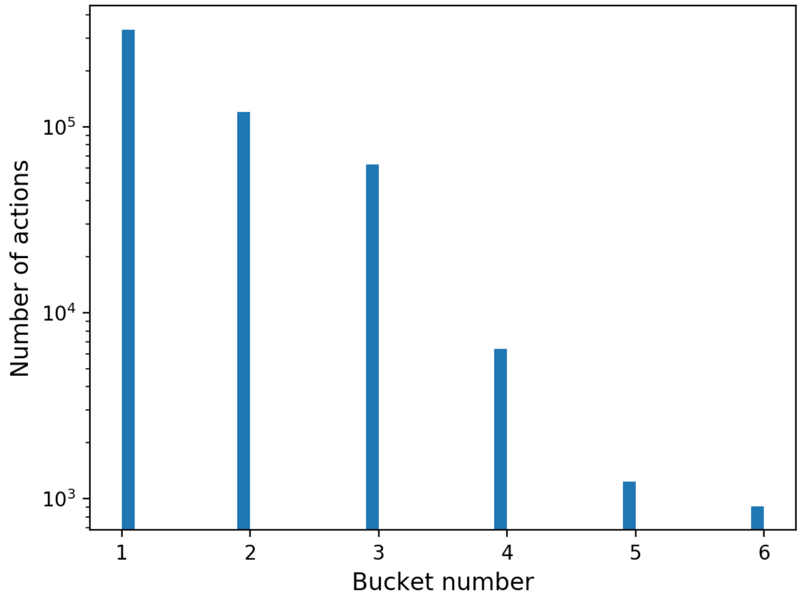


Figure 6.5: Histogram displaying distribution of action time in the respective time buckets in a logarithmic scale.

6.4 Performing the cluster analysis

When all pre-processing in the form of data reduction, feature selection and data transformation had been made, the analysis was ready to be performed. The previously described data transformation produced a file that could be run by the RHC algorithm. The file is presented as “pre-processed input file” in Flowchart 6.1.

RHC utilises a complex algorithm necessary for clustering users based on clickstream data. It creates a similarity matrix and uses it to find which action pattern, also called k-gram, that are most defining, and these users are put into the same cluster. A new calculation is made after removing the most defining action patterns of the cluster, calculating what remaining action patterns that define the new sub-clusters. This is done to a maximum level of four layers. The result is represented by “Output file with user cluster” in Flowchart 6.1. The pre-processing in combination with the clustering algorithm constitutes a complex process that can be hard to follow without extensive knowledge about the algorithm and the dataset analysed. An interface with a complex analysis is not of any use if it is not understood by its operator. The interface must therefore strive to present the analysis and its result in a way that the operator can comprehend and use. This leads to the following Part 3 that describes the process of designing an interface that is able to conceptualise the complexity of the analysis.

Part 3: The interactive tool

The design problem: conceptualising complexity

7. Theory

In part two, the complexity of the analysis was described. When dealing with complicated problems such as identifying similarities between clickstreams, a complex analysis can be a necessity. A complex analysis can capture small differentiations between clusters. But in order to capture those differentiations, the operator of the analysis must have an understanding for how it works in order to finetune the parameters. This is of great importance since the results of the complex analysis, such as described in the previous chapter, greatly depends on what data, and the quality of it, are put into the analysis. The interface, in which the analysis is performed, must therefore support the analysis operator in understanding the complexity and the implications of the parameter tuning. A complex analysis, and its results, is not of any use if it is not understood by the operator. The interface must therefore strive to conceptualise the complexity of the analysis.

This chapter presents theories of how humans interpret complex information and how to present it to the operators of the system. The chapter ends with a description of how RHC presents its information and how to perform usability testing on an interface.

7.1 Recognition-primed decision making

One of the main purposes with interface and data presentation design is to support decision making. An operator of a system can under normal operating conditions, that is without failures in the system, be called a decision maker. Depending on the perceived outcome of the situation, the amount of evidence needed in order to make a decision varies. In the last years, the research field of decision making has evolved a lot. Earlier, decision making was mainly considered an analytic process. The operator assessed every possible dimension of the problem and all alternatives were weighted and evaluated before making the final decision. The cognitive processes were considered much more significant for decision making than the perceptual processes. Recently a new approach to the research field of decision making has emerged where the perceptual processes have proven to be significant. This new approach emerged from studies of how operators make decisions in their natural work domain. One example of this is recognition-primed decisions (RPDs). (Bennett and Flach, 2011) The RPD model emphasise the idea that a person can make a decision by using their experience, without having to compare different options. Studies show that RPD:s are much more commonly used in real world settings than analytical decision making. Even in more complex and nonroutine situations. (Klein, 1993)

Recognition-primed decision making consists of several steps. First of all, the operator recognises the problem and categorises it into a group of similar problems that the operator has encountered before (Bennett and Flach, 2011). If the problem is not similar to anything the operator has encountered before, more information about the situation is sought out in order to identify familiar features (Klein, 1993). The operator then performs a situation assessment. The assessment includes establishing what goals that are desirable and possible to reach, looking for critical perceptual clues in the interface design, developing an idea of what will probably happen in the subsequent steps and finally identifying typical approaches that have been used in similar situations before. (Bennett and Flach, 2011) In low-complexity situations a decision can be made after the assessment. When dealing with more complex situations, the operator usually makes a mental run through, or simulation, after the assessment of all steps in the possible solution to identify if the action will fail or succeed. If the mental run through reveals major flaws in the strategy, the operator returns to assessing the situation, seeking out new information that could lead to an alternative action plan. (Klein, 1993)

Studies show that experts do not assess all possible solutions, as suggested by earlier studies, but rather evaluate a few good alternatives. In short, operators look for the first best alternative that could generate a solution. (Bennett and Flach, 2011) Herbert Simon is the inventor of the term *satisficing*, which is the process where a person tries to fulfil his or her goal and chooses a solution which is adequate. *Satisficing* is a portmanteau of the words satisfactory and suffice and indicates that the chosen solution does not have to be the optimal one, but one that is "good enough". There are many areas in the real world where an optimal solution is non-existing or takes too much time and resources to calculate. Simon describes that both optimisation and *satisficing* measures the alternatives by standards, but there are some differences between them. Optimisation strives to fulfil all possible utility functions for the task, whereas *satisficing* instead compares the outcome of the solution with the desired outcome of the operator. Simon means that satisfaction is recorded as positive if the outcome exceeds *aspirations*. If the aspired outcome is reached, the solution is *satisficing*. (Simon, 1996)

7.2 Conceptual models theory

Donald Norman is the inventor of six design principles that affects how discoverable a product is for an operator. How well these principles are implemented is crucial for how the operator uses the interface and can interact with it. The principles are *affordances*, *signifiers*, *constraints*, *mappings*, *feedback* and *conceptual models*. (Norman, 2013) The first five principles can be described as follows:

Affordance tells whether an object is signalling that it can be used as intended. A pitcher *affords* being held and holds some type of mass in it. This is *signified* by its handle that invites grabbing, but also the pipe that suggests something can be poured out of the item. One could call these *signifiers* clues for the intended purpose of the object. As Norman writes: "*Affordances determine what actions are possible. Signifiers communicate where the action should take place.*" (Norman, 2013) The pitcher affords being held because of the handle affords being held by someone with sufficient sized hands. A giant might not fit to grab the handle since they are the wrong size. In addition, an infant might not have the strength to lift it. The object needs both to communicate its affordance and to enable the actual possibilities. (Norman, 2013) The person to first define *affordance* was James Gibson but his definition of the term was slightly different. It was a wider concept of objects' *affordances* since it includes all possible actions of the object, regardless if it is perceivable by the user or not. (Gibson, 1986) In this thesis *affordance* refers to Norman's definition because it suits the domain of computer screens better.

Constraints prevent the operator from performing actions and ideally help the operator understand what actions cannot be performed. These constraints can be physical, such as trying to insert the wrong key into a keyhole. The inability to insert the key is a sign of constraint, and even if it is possible to insert it, it still will not be possible to rotate. *Mappings* indicate some connection between an action and a functionality. The scrolling when reading this document on a screen is a *mapping* between the mouse wheel, scrollbar or touch-movement you use to move the text. This *mapping* is an example where different *mappings* are seen as the most natural. By moving the scrollbar down, some people assume the page moves down, but others assume an imaginary box of looking at the text to move down, which means the text actually moves upward. *Feedback* is the process of communicating to the operator that an action is being handled and that some status changes with the action. (Norman, 2013)

Norman points out the sixth principle, *conceptual models*, specifically. Conceptual models can be described as derived from the other principles. The models are essential to how an operator understands how to use an object, for example a web-page. The conceptual model can be seen as an inner interpretation of how the product works. This interpretation is not always accurate, but the purpose is not to be an exact replica of the actual workflow. It is used to help the operator remember and understand the mappings and what they result in. (Norman, 2013)

7.3 Data representation design

When designing an interface or data presentation, one must take into consideration components that support the perceptual processes of the operator and reinforce the operators' *conceptual model* of the system. The interface should include visual clues of the limitations of the domain and the *affordance* and *constraints* of the tool. (Norman, 2013) (Bennett and Flach, 2011) The visual clues will support the decision maker in the studied domain to identify the current situation and to evaluate details in the current situation that could affect the possible actions and outcomes. The visual clues could also support the decision maker with options of how to make the decision. By introducing these visual clues to the operator in the interface, the operator can get an understanding of the system's current state and its possible and encouraged action options. This understanding can be achieved by just looking at the visual clues. (Bennett and Flach, 2011)

One data representation design strategy emerges from an aesthetic perspective. This strategy was developed by Edward Tufte, who emphasises intuitive judgements when making design decisions. (Bennet and Flach, 2011) According to Bennett and Flach (2011), his work is widely acclaimed. Tufte describes the complexity of displaying information that exists in a three-dimensional space on a two-dimensional surface, such as a screen. He means that escaping the flatland is essential for being able to envision the information. Some data is far too complex than a two-dimensional surface is able to reflect. Tufte presents several design principles that enable comprehension of complex data on a two-dimensional screen. (Tufte, 1990) The *data-ink ratio* is one of his design principles. The *data-ink ratio* refers to how much ink is needed to represent the data, as opposed to non-data elements in the representation. A graph or histogram where more ink is dedicated to data representation components rather than to non-data components is considered more effective. The second principle is to have a higher *data density*, that is number of data points per areal unit. Other principles are to eliminate irrelevant graphical structures such as decorations and containers and to use other aesthetics such as effective labels and proportions. Tufte has broadened these design principles to also include nonquantitative displays as well. He states that it is more important to effectively arrange the information than how much information there is. (Bennett and Flach, 2011) He means that details lead to more coherent structures. When data that is derived from complex and detailed information is properly arranged, it is simple to read both the wider picture and the specific details of the data. He says that "*to clarify, add detail".* Although, he also states that it is bad if the operator cannot understand the presentation, and clutter and confusion in the design presentation is a strategy for failure. (Tufte, 1990) Lastly, he also indicates that a successful design principle is to stratify various aspects of the data, working with layers and separation. This stratification will help the operator to facilitate and extract information from the data representation. (Bennett and Flach, 2011) Specifically contouring and ordering of data help the operator distinguish information. Using a grey colour for background to black information makes the reading of the data easier. (Tufte, 1990)

7.4 Recursive Hierarchical Clustering

RHC is a program free for use by a GNU license[1]. (Wang et al., 2016) (Zhang, 2020) It provides a visualisation interface where the produced behavioural clusters are visualised in an interactive way. RHC was meant to be used by service providers to better understand what types of behaviour groups that occur and are dominant in their system. RHC visualises what behaviours dominate the different user groups. The system needed to fulfil three requirements. Firstly, it had to scale and handle large scale and noisy clickstream data well. Secondly, it had to be able to capture previously unknown behaviours. Finally, the tool had to be intuitive and easy to interact with in order to help the operators to easily understand the user behaviours. A case study was carried out on two social network applications to evaluate the system. The tool was able to handle a large noisy dataset and capture and visualise expected and unexpected user behaviour in a simple and intuitive manner. (Wang et al., 2016)

[1]From 2007, this license states that changes can be made to the code. All released software should be public. https://www.gnu.org/licenses/quick-guide-gplv3.html

The visualisation interface of RHC consists of two main parts (see Figure 7.1). At the right side of the screen, there is a display with a description of the tool and settings for the visualisation (see Figure 7.2).

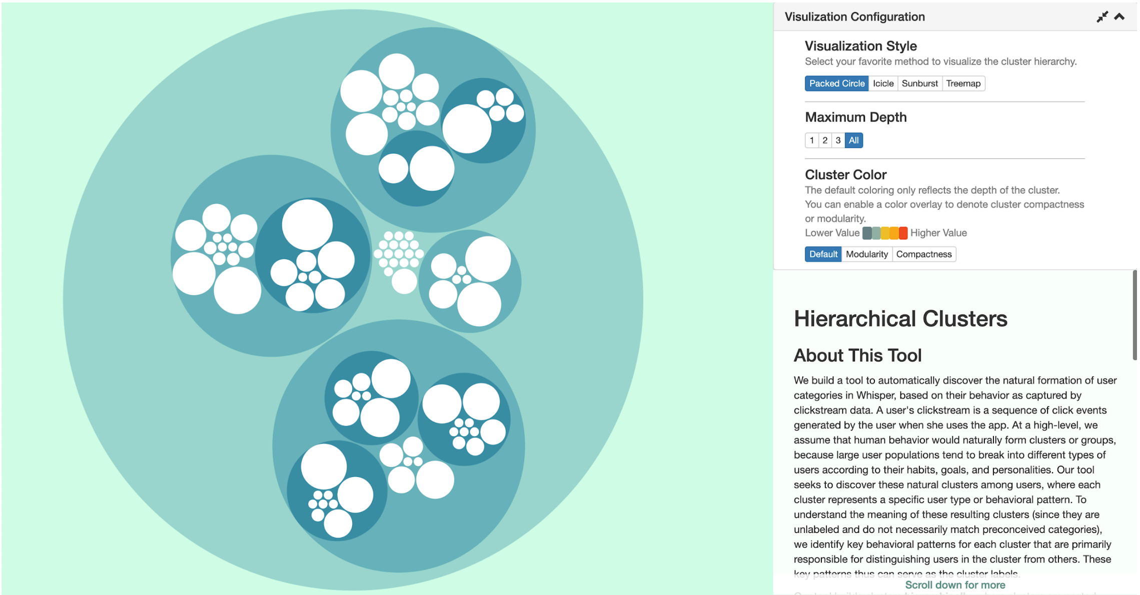


Figure 7.1: Print screen of the visualisation interface of RHC.

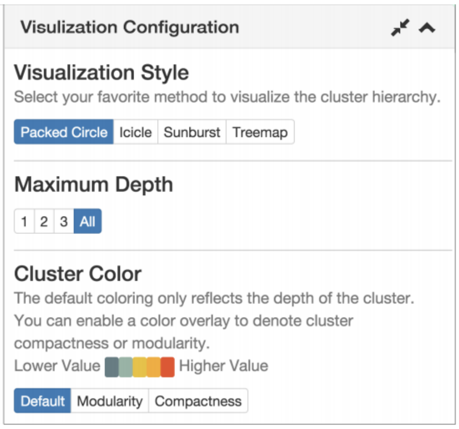
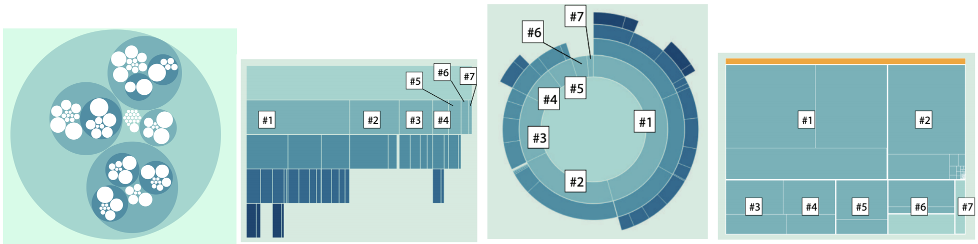


Figure 7.2: Print screen of the settings panel for the visualisation interface in RHC.

In the settings the operator can choose visualisation style. The cluster can be displayed as packed circles, hierarchy as an icicle, sunburst or as a treemap (see Figure 7.3). By default, the cluster hierarchy is displayed as packed circles. The size of the circle represents the cluster size, based on the numbers of users in it, and child clusters are placed inside of their parent cluster. The operator can also choose the maximum depth of the clusters and if the colour of the clusters should represent modularity or the compactness of the clusters.



(a) Packed circles (b) Icicle (c) Sunburst (d) Treemap

Figure 7.3: Example of hierarchical clusters displayed with different visualisation styles in RHC. (Wang et al., 2016)

At the left side of the visualisation interface the cluster hierarchy is displayed with the chosen visualisation style of the interface settings. The operator can interact with the clusters by clicking them. A pop-up window is then displayed with information about the cluster (see Figure 7.4). This is where information about behaviours in the groups are displayed. The pop-up window contains information about how many users the cluster includes, its action pattern, the frequency of the action pattern and a χ^2 score of the action pattern. The rank of the action patterns are based on the χ^2 score. It is also possible for the operator to add a description to the cluster as plain text. The action pattern is a series of actions and its time usage. The actions in the action pattern are chronologically ordered. The frequency of the action pattern describes how often the action pattern occurs in the cluster compared to users outside the cluster. The red bars show how frequently the action pattern occurs inside the cluster, while the green bars describe how frequently the action pattern occurs outside the cluster.

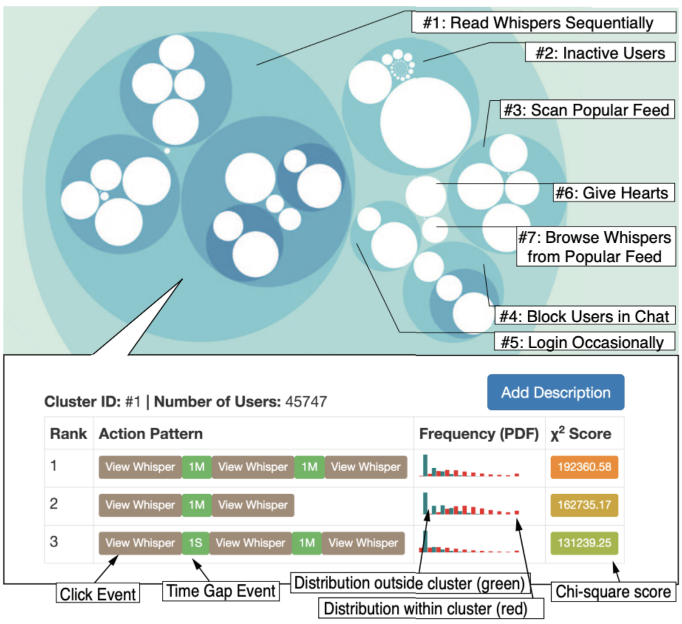


Figure 7.4: Example of hierarchical clustering displayed as packed circles in RHC. The pop-up window shows more detailed information about the cluster. (Wang et al., 2016)

7.5 Usability testing

According to the international organisation for standardisation, the definition of usability is "*the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*" (ISO, nd).

When a product has a built-in usability, the operator does not think about it. If the usability is inherent in the product, the operator does not have to bend to the will of the product. Instead the product works as the operator expects it to. That does not mean that the operator understands everything about the product, but the effort of learning how to use it is worth it because the rewards of using the product is greater than the effort of learning how to use it. A rewarding sensation can be ease of use, ease of learning the product, intuitiveness and having fun. (Barnum, 2010)

Usability testing is a systematic way of testing how potential real operators would interact with the product under controlled circumstances. In usability testing the operator gets to use the product on their own without major support. (Dumas and Loring, 2008) A developer of a product knows too much about it to be able to know if the product will be usable and understandable for an operator that does not know what the developer knows. Usability testing enables for observation of how operators actually use and understands the product and what they do not understand. (Barnum, 2010)

Usability testing of a system, service, or a product can be conducted remotely, meaning the participant and the moderator are not in the same physical place. The testing is then conducted through shared technology. Remote testing can be conducted synchronously or asynchronously. During asynchronous testing, the participant and the moderator work at their own pace separately and the actions of the participant are recorded and analysed afterwards. During synchronous testing, the communication and feedback between the participant and the moderator occurs in real time. Remote synchronous testing has some disadvantages such as that the moderator might miss out on what the participant is looking at. The moderator will moreover not be able to control the testing environment, which might lead to distractions for the participant such as children running around or traffic sounds in the background. Remote synchronous testing enables testing when it is not possible for the participant to attend at the same physical location. It is also time effective, since neither the moderator nor the participant has to travel anywhere. The participant might also feel more comfortable to participate in the usability testing when situated in a known environment. The participant can also be studied using the product in the environment the product would be used in real life situations. (Dumas and Loring, 2008)

8. Method

RHC was used as a base for the presentation of the clustering result. RHC was improved and tweaked to further fit Plicks' needs. Both appearance and functionality were changed. The original visualisation of the clusters is further described in section *7.4 Recursive Hierarchical Clustering*. Usability tests were made in order to understand whether an operator of the system can understand and interpret the result of the analysis and the available information. The purpose of the usability tests was to observe the participants' ability to find correct information, and their ability to interpret and use the information in a correct way. The following chapter describes and discusses the usability test conducted and its potential flaws.

8.1 Usability testing

The usability tests were conducted on April 21st and 22nd, in sessions of 45 minutes. A pilot study was conducted with one participant four days earlier in order to identify weaknesses or unclarities in the task formulations. On May 4th and 5th, follow-up questions were asked in sessions of 10 minutes with the participants. Eight people, in addition to the pilot participant, participated in the study. All respondents work as developers at Swace. Table 8.1 summarises the respondents previous experience of Plick and clustering. The intended operators of the tool are developers, therefore the participants in the usability tests correspond well with the intended operators. All participants were presented with a consent form and accepted it verbally. The participants were informed about anonymisation of their identity, the goal of the tests, and the possibility to opt-out from the study. Finally, they were asked for approval of being cited in the report.

Table 8.1: A summary of the usability test participants and their background experience with Plick and clustering.

Partici pant	Has worked with Plick	Knowledg e of Plicks user data	Knowledge of screen names and events in Plick.	Have experience with clustering	Express knowledge about hierarchical clustering
1	Some	Some	Fairly good	Have used k-means	No
2	No	No	No	Maybe from University courses	No
3	Some	Quite good	Knows the basics but knew it better before.	University courses, Mentor to master thesis with clustering.	No
4	Yes, built the app	Average	Fairly good	No	No
5	Yes, built the app	Very good	Very good	Some. Have done clustering.	No
6	No	No	No	No	No
7	Some	Some	No	Conceptual knowledge	No
8	Some	Not very good	Some	University courses	No

During the time of the testing many parts of society were affected by restrictions implemented because of the virus Covid -19. The public health organisation in Sweden recommended social distancing in order to stop the virus from spreading too fast. (Krisinformation, 2020) Therefore the testing was conducted remotely through Zoom, a video communication tool, with the participant and moderators present on separate computers (Zoom, 2020). All participants were familiar with the platform Zoom. The moderators, and a majority of the participants, had their web camera on throughout the testing. Meetings through Zoom have the functionality of sharing one screen to all in the meeting. The host can also give participants remote access to control the screen of the moderator. This was used in the testing so that the participants were able to control the tool through their own computers. The complicated task of sharing the code and installing it locally on the participant's computer could then be avoided. A small latency was experienced during remote control, but since the tool is not depending on very fast operator actions, the latency was acceptable for the tests. All parties were aware of the situation with restrictions in society, and the small latency was not mentioned by any participant.

Remote meetings had the advantage of the test subject being able to hide the image of the test leaders if they felt it disturbed them. It also made it possible for the test leaders to mute themselves and discuss how to handle various situations such as when to give more leading information and when to help a participant. One disadvantage of remote meetings is the possibility of the participants being distracted by factors in their surroundings that the moderators cannot control. To our knowledge, this only happened once during the tests when a participant received a message that they answered right away. This did however not affect the study to any larger extent.

After a short introduction of the project and a check of the participants background knowledge of Plick and clustering, the participants were asked to read through the information page that described the tool (see Appendix C). The participants spent an average of 2 minutes and 25 seconds on reading the information page. Thereafter, the participants were asked to get acquainted with the tool and the information presented about the clusters and the clustering reconfiguration settings. They spent between four and nine minutes examining the tool on their own.

Before the testing started, the participants were informed that the test-leaders would interact as little as possible with them during the tests. The participants were told not to double-click in the tool too much since that often triggered errors at the time of the test. In the case of other unexpected errors, the participants received instructions on how to handle them. After the introduction they were presented with a slideshow containing the tasks of the usability test. The participants could toggle between the tool tab and the tab with the tasks by themselves. The participants had access to the exact formulation of the task during the entire usability test. The test consisted of eight main tasks with subtasks. All tasks are presented further in Appendix D. After the usability test had been conducted, the participants were asked to say one good thing about the tool and at least one that could be improved. Open questions at the end of an interview enable the participants to mention things that they have been unable to express during the more directed questions. These concluding questions could capture insights otherwise missed.

Finally, the participants were asked to fill in a questionnaire anonymously. The questionnaire consisted of eight subjective statements, with a 4 point Likert scale (Likert, 1932) between 1, “I do not agree at all”, to 4, “I totally agree” (see Appendix E). After a reminder, seven out of eight participants filled in the questionnaire. Since there was no way of knowing who the last person was, it was decided not to send out a second round of reminders but to use the results of the seven answers.

In the follow-up meetings, three questions (see Appendix D) were asked about the understanding of the analysis within the tool such as: “What does an action pattern mean”. These meetings took place two weeks after the original usability tests. The purpose of the follow-up meetings was to get a comprehension of the participants’ understanding of the system. The need for follow-up meetings was discovered when analysing the results from the first usability tests.

8.2 Sources of error

Some of the tasks in the usability tests triggered errors more often than others. Other tasks were harder to conduct without previous knowledge of Plick and its screen names. In these cases, the test was interrupted by the test leaders and the participants were given some guidance in order to complete the task. This was not considered a hindrance for the usability testing since the tool is meant to be used by operators with background knowledge of the screen names. The task of grouping all screens that belong to the feed tab of the application was one task that needed instructions regarding the names of the screens. These instructions were given orally, and the ambition was to give the information in the same way, but slight variations occurred. The goal of the specific task was not to study the participants mapping of Plick pages, but simply to test if it was possible to find how to group pages. In summary, the average time for that task should be treated with caution and should not be analysed.

Despite some alterations made to the tool after discovering errors in the tool during the pilot test, a minor error occurred in the version that the participants used. In the graph showing how many ads a user has created, both created and sold ads were labelled with the same name. Two participants made the test with this error, and it was corrected to the later tests. The two first participants were informed of the error instead.

Some of the participants had seen the tool during its development when they were assisting with technical questions. This means they have had some previous knowledge of the tool besides the introduction during the test. This extra knowledge could have affected the conceptual model of the system.

The test was done with data from a quite small dataset of approximately 600 users. The choice to use this dataset was made with regards to the positive effects of being able to recluster quite fast. With 600 users the reclustering took a few minutes. This pause would not interrupt the test, or the participants willingness to lend their time to the test. When trying to make an analysis as good as possible, a larger dataset is required. The goal of the thesis was to enable analysis, not to make the analysis. Consequently, a small dataset was not considered a problem. Nevertheless, using a large dataset in the usability test would have enabled testing of more ecologically valid situations.

All participants in the usability test were engineers, as well as developers. It is likely that engineers search for information in a similar manner, but developers can come from a variety of backgrounds, education or disciplines. This could make the testing of the tool valid in the case study, but it could result in lack of valuable insights of how other types of developers search for and interpret information.

9. Result and Discussion

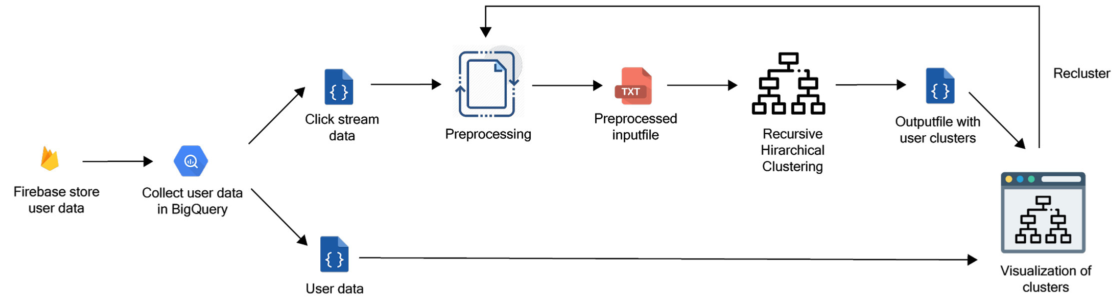
RHC, created by Wang et al., is explicitly created for service providers of systems. RHC visualises the clusters and their action pattern but does not provide any further explanation of the results or the algorithm. During the continuous informal conversations with the CEO of Plick, when the tool was presented, it was established that there was a need for further explanation of the results and analysis made by RHC. It was expressed that an additional understanding of the analysis and the result was desirable in order to better interpret the result. Substantial developments of RHC were therefore implemented for the operators to better understand the complex results of the analysis. Without improved comprehension of how the tool works or what the results imply, usage of the results or further analysis would be difficult. RHC with our changes in visualisation and functionality is termed “the visualisation tool”, or simply “the tool” in this thesis. The modifications of the tool belong to one of two main areas. Firstly, modifications were made to what information should be displayed in the tool, regarding the clusters, and how it was displayed. Secondly, the tool was further equipped with a possibility to modify the pre-processing of the data in order to recalculate the clusters and thereby extract more defined user groups. The following chapter presents the two types of modifications implemented in the tool.

9.1 Clustering reconfiguration

In order to bring the complex cluster analysis closer to the operator of the tool and make it more understandable, a clustering reconfiguration feature was implemented. The developer gets to interact with the settings of the clustering analysis by using the clustering reconfiguration feature. The analysis does not have only one correct outcome. The reconfiguration makes it possible for the developers to experiment with the analysis and this enables the operator to find other results that might be more suitable for their purpose. For instance finding a specific subgroup of users. The settings and the reclustering is also a way for the operator to experiment and confirm his or her conceptual model of the clustering.

Default parameters were offered in the interface in order to support the operator’s conceptual model of the system. These default settings, and the entire reconfiguration of the cluster analysis was meant to act as a visual clue to the limitations of the domain, as described by Bennett and Flach. The parameters chosen in the default option was decided after studying the available dataset. The visual clues were intended to support the operator of the visualisation tool in identifying the current situation and evaluating the details of the situation that could affect possible actions. The visual clues were also meant to give the operator a clue of how to reconfigure the clusters in a meaningful way. The clustering reconfiguration feature was designed in four steps that should support the operator in making a well-informed decision. In the first step, the operator is faced with the possibility to exclude unwanted screen views and events from the analysis. In the second step, the operator was able to change the grouping and the naming of the screen views and events.

In this step, the operator could exclude specific screen views and events. In the third step, the operator was able to adjust specific event and time settings. In the last step, the operator triggers the clustering reconfiguration. These steps are further described in the following chapters. The reclustering is represented in Flowchart 9.1 by “Recluster”.



Flowchart 9.1: A model of the flow of data through the tool including the functionality of reclustering.

9.1.1 Exclude screens and events

In the first step of the clustering reconfiguration, the operator was presented with the possibility to exclude screen views and events from the analysis (see Figure 9.1).

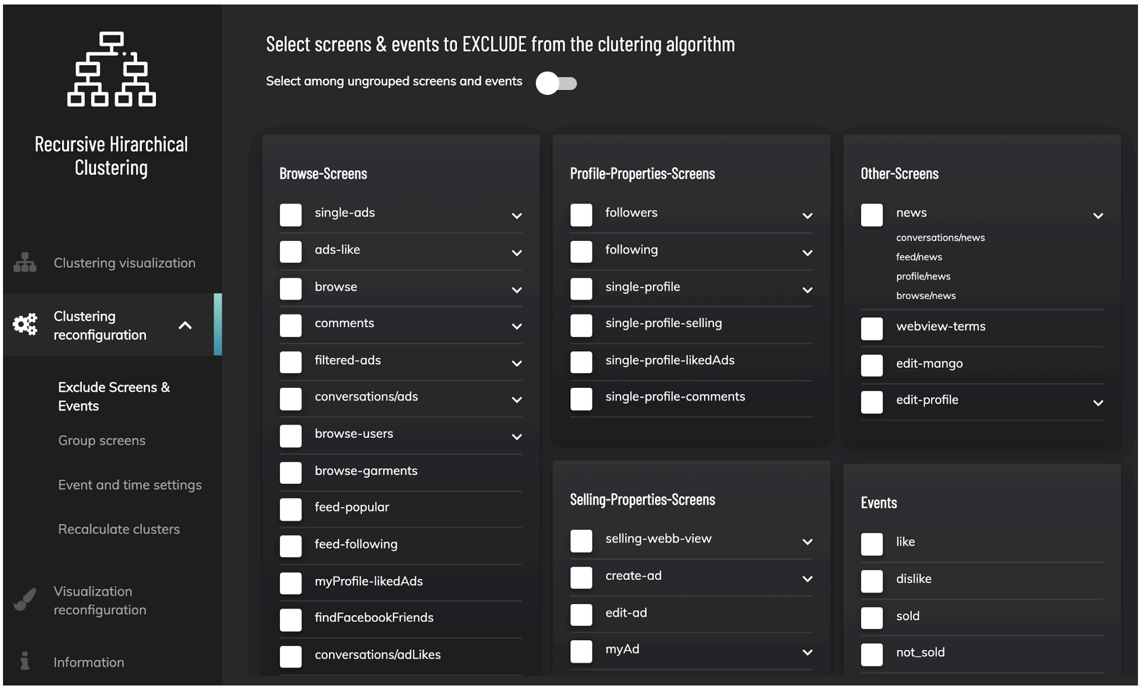


Figure 9.1: The first step in the clustering reconfiguration, where the operator can exclude screen views and events from the analysis.

This view represents the pre-processing step of the data described in section *6.3 Pre-processing*. It can be seen as a manual pruning step that enables the clustering algorithms to find other patterns that would not have a prominent place in the visualisation because of more dominant features. For example, a screen view that almost all users look at is the screen view “single-ads”, the view for looking at an ad. By excluding the “single-ad” view, more fine-grained differences between the users can be found and visualised. Excluding some screen views and events can in addition help against the curse of dimensionality. Each screen view and event can be seen as a variable in the analysis. By excluding some, the possible variations are decreased and thereby also the dimensionality of the analysis.

In the clustering reconfiguration settings, the operator can observe the pre-processing step of grouping similar screen views and events and see the predefined settings, as described in section *6.3.3 Data transformation* (see Figure 9.1, in “Other screens” under the group “news”). The operator can additionally choose to exclude among ungrouped screen views and events by toggling the “select among ungrouped screens and events”. By clicking the toggle, the operator is exposed to all screen views and events with their original names. This view should give the operator visual clues of how the pre-processing of groups is used in the analysis. It should likewise give the operator a clue of the limitations of the domain, meaning the screen views and events analysed by the tool.

9.1.2 Group screen and events

In the second step of the clustering reconfiguration, the operator can change the grouping of screen views and events (see Figure 9.2).

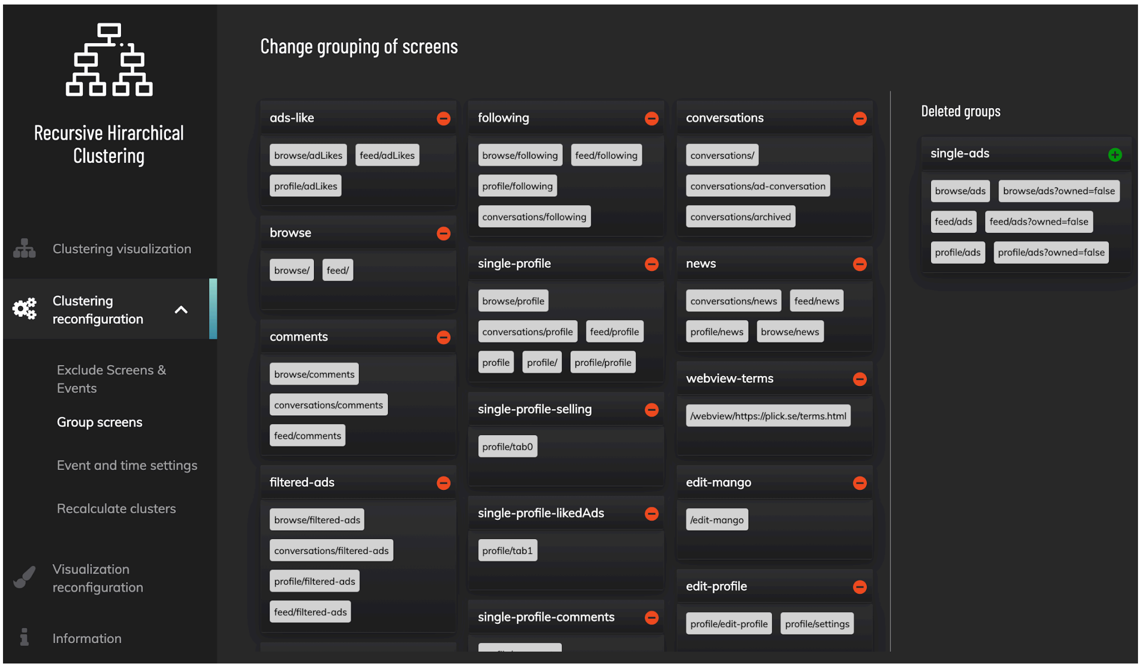


Figure 9.2: The second step in the clustering reconfiguration, where the user can change the grouping of screen views and events.

The operator can furthermore delete or take back deleted groups into the analysis. By dragging the desired screen view or event into either an existing or a new group, the operator can change the default grouping of screens. By offering default grouping, the operator can get an understanding of how grouping of screen views and events can be performed. It should enable the operator in their effort to recalculate the clusters in an efficient and correct way. The operators are able to group screen views and events without default grouping by toggling "select among all screens and events". It requires extensive knowledge about the screen names in the application.

By grouping the screens, the dimensionality of the analysis can be further compressed as well. By grouping two screens, the operator is replacing four dimensions with one. For example, by joining the screens A and B to C, the operator replaces the possibility of the action pattern A → B, B → A, A → A and B → B with the new action pattern $C = \{A \rightarrow B, B \rightarrow A, A \rightarrow A, B \rightarrow B\}$. Reducing the number of possible action patterns can lower the risk of curse of dimensionality.

9.1.3 Event and time settings

In the third step, the operator can change specific event and time settings in the analysis (see Figure 9.3).

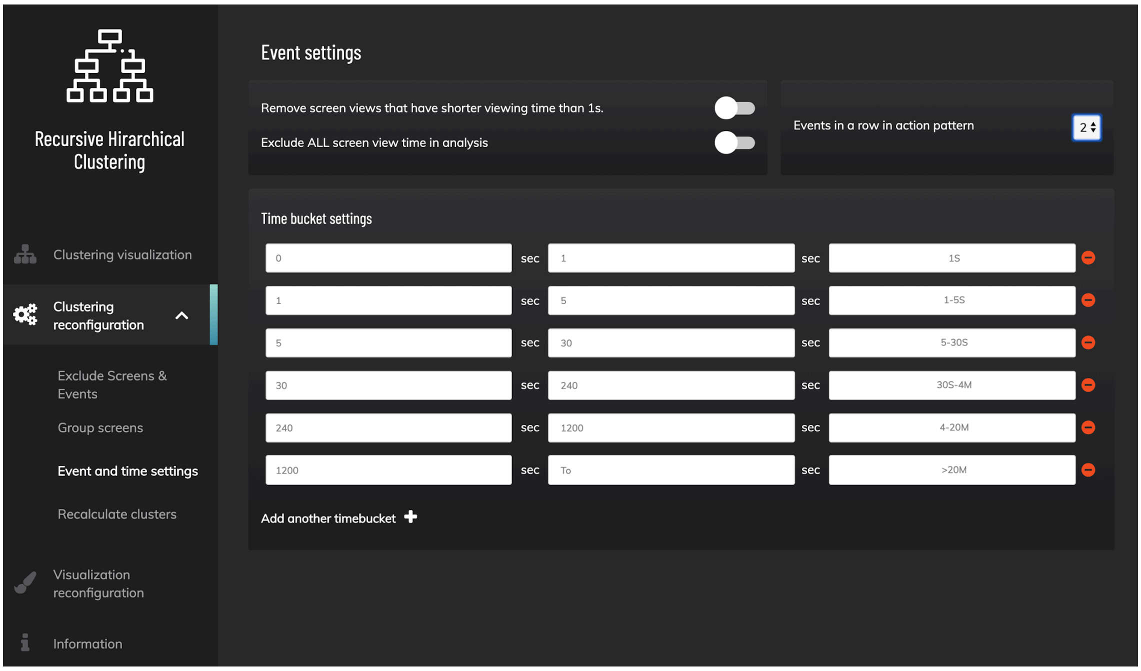


Figure 9.3: The third step in the clustering reconfiguration, where the operator can change event and time settings.

The operator can choose to exclude screen views that are shorter than one second. This was made possible because when a user opens a tab in the application, the intention might not be to look at the view displayed when opening the tab, but rather a view that the user only can reach by first clicking on the tab. By excluding screen views shorter than one second, it is possible to leave out those unintended screen views. One second long screen views was chosen as a signal of unintended views after a study of how long screen views usually are. This study is further described in section *6.3.3 Choosing time buckets.*

The operator can choose how many screen views and events in row that should be analysed in the clickstream. The screen views and events that are presented in the action pattern are chronologically ordered actions. By including more actions in the action pattern, a greater variation and more fine-grained differences between the clusters can be observed. For example, two actions, A and B, in a row in the clickstream results in four different variations: A → A, A → B, B → A or B → B. But three actions in a row can be combined in 27 different ways. By looking at fewer actions in a row, a reduction in dimensionality can be obtained. This might lead to avoidance of the curse of dimensionality. The default value in the analysis is set to two screen views or events in a row in order to get a balance between finding fine grained variations between users, but still trying to avoid the curse of dimensionality. However, the operator can choose to have screen views and events up to five in a row. It can be of interest to include more actions in the action pattern if the aim is to examine in what order screen views appear. The limit of five actions in row was chosen in a consideration of complexity in the results versus the risk of a curse of dimensionality. It is possible to reduce the complexity by removing several screen views or events instead of reducing how many actions to have in a row.

Finally, the operator can either modify or exclude the time variables from the analysis. Similar to the amount of screen views or events in a row, the amount of time variables included in the analysis increases the number of possible action patterns. By excluding all time variables, focus is put on what screens the user has visited, in what order, and removes the factor of how much time he or she spent on the different screens. By including time variables, visiting a screen for 10 seconds is not categorised the same as visiting the same screen for 4 seconds (when using the predefined time buckets).

9.1.4 Reclustering

In the final step the operator can recalculate the clusters (see Figure 9.4). Depending on the size of the dataset to be analysed, the recalculation might take some time. The operator is presented with a warning pop-up window, stating that the recalculation might take some time. The operator can then either abort or accept the recalculation. If accepted, a spinning wheel is shown to indicate that the process is running in the background. This is a common technique for letting the operator know states are changing, even if the page itself remains static. The loading symbol is an example of *feedback* to the operator that the program is working in the background. With careless dimensionality increase, the computing time can quickly increase from minutes to hours or even days. The feedforward could therefore be further improved by displaying to the operator an approximation of how long the calculations might take.

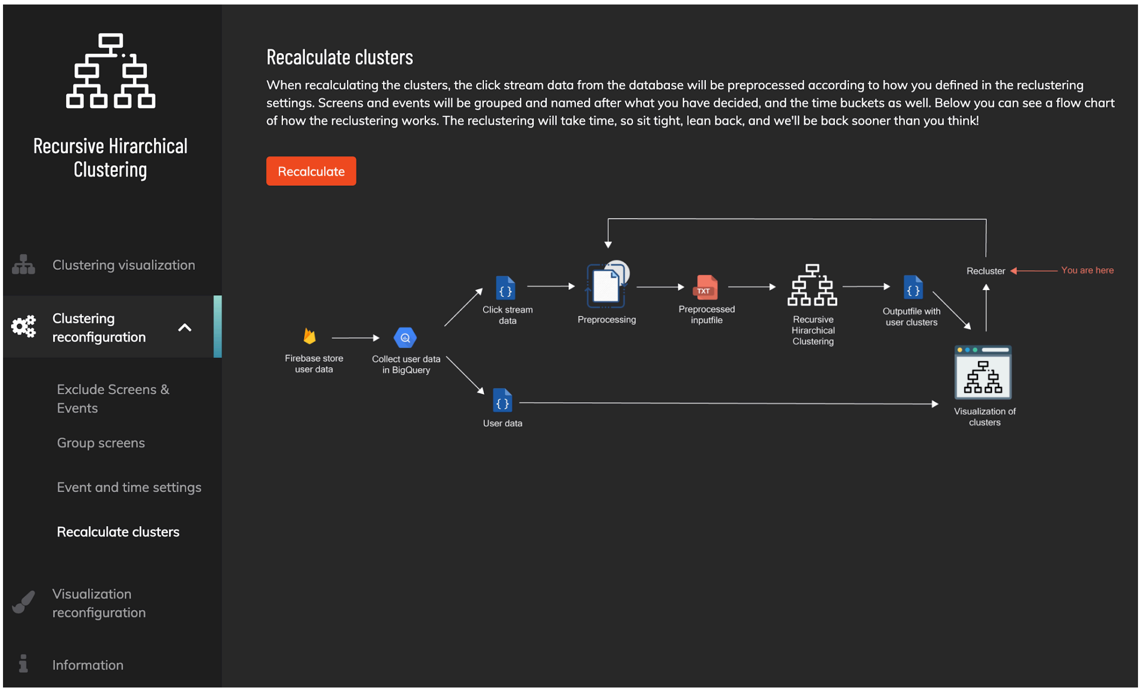


Figure 9.4: The last step in the clustering reconfiguration, where the operator gets an overview of the process of the analysis and can recalculate the clusters.

In this step the operator gets an overview of the flow of the tool. More precisely, it displays the relationship of the different steps in the tool, how the data is handled, where the pre-processing is carried out and what part of the process the reclustering affects.

9.2 Data visualisation

The visualisation of the clusters is inherited from RHC. Using a hierarchical cluster method, the clusters naturally split into smaller and smaller clusters. The four visualisation styles of the clusters presented earlier in section *7.4 Recursive hierarchical clustering* use layers to display the clusters in an intuitive way. The operator can decide what depth he or she wants to display and which visualisation style to use. Layering and separation is a good way of arranging information when the goal is to aid the operator in understanding the information. During the development of the tool, the result of the analysis was discovered difficult to understand by the operators. Extra information tabs were therefore added to each cluster. These are further described in the following section.

9.2.1 Information tabs

The purpose of the tool is for the operator to understand the users of the application. This means information is needed about the users. As seen in Figure 9.5, when the operator clicks on a cluster, a box with information about the cluster appears at the right of the screen. This box has five tabs, each displaying different information. They are: “Action Pattern”, “Seller Properties”, “App usage”, “User Properties”, “User ID:s”. The operator is given visual feedback of what cluster the information belongs to by a light blue border around the chosen cluster. Since there is no action pattern for the root cluster, containing all users, RHC does not allow the operator to choose that cluster. In the visualisation tool it is possible to click on the root cluster. The information box for the root cluster contains the same information as for the other cluster, except the information about action patterns.

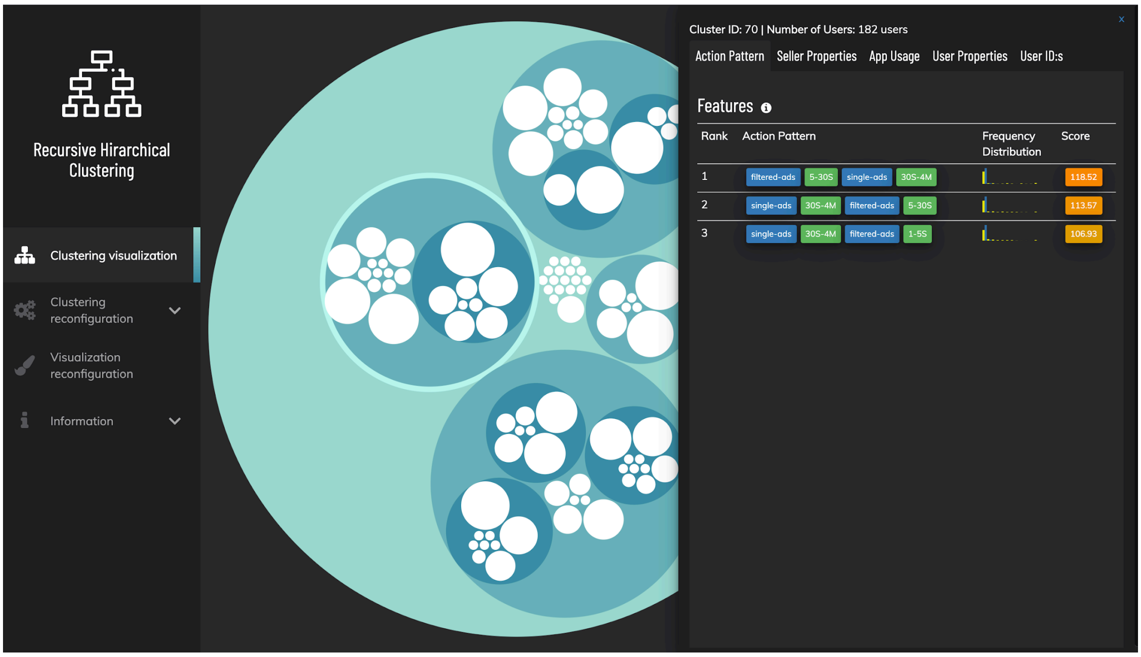


Figure 9.5: The box of information of the chosen cluster. The chosen cluster is highlighted with a boarder. The action pattern is displayed.

The action pattern view includes mostly the same information as in the original visualisation described in section *7.4 Recursive hierarchical clustering*. The only changes to this view include the colouring of the frequency distribution and an added information button where the operator can get an explanation of the features (see Figure 9.6). The original colours in the frequency distribution, green and red, is often a visual clue for something correct or wrong. Since green was corresponding to all users outside of the cluster, but often is the colour of success or a right choice, it could be misleading and easy to swap the colours in the analysis. Therefore, the colours were changed into two completely different colours that blend into the colour scheme of the tab better. These colours also work better for colour-blind operators and reduce the risk of being interpreted as either being more correct than the other. The yellow colour represents the users within the cluster and blue represents users outside the cluster.

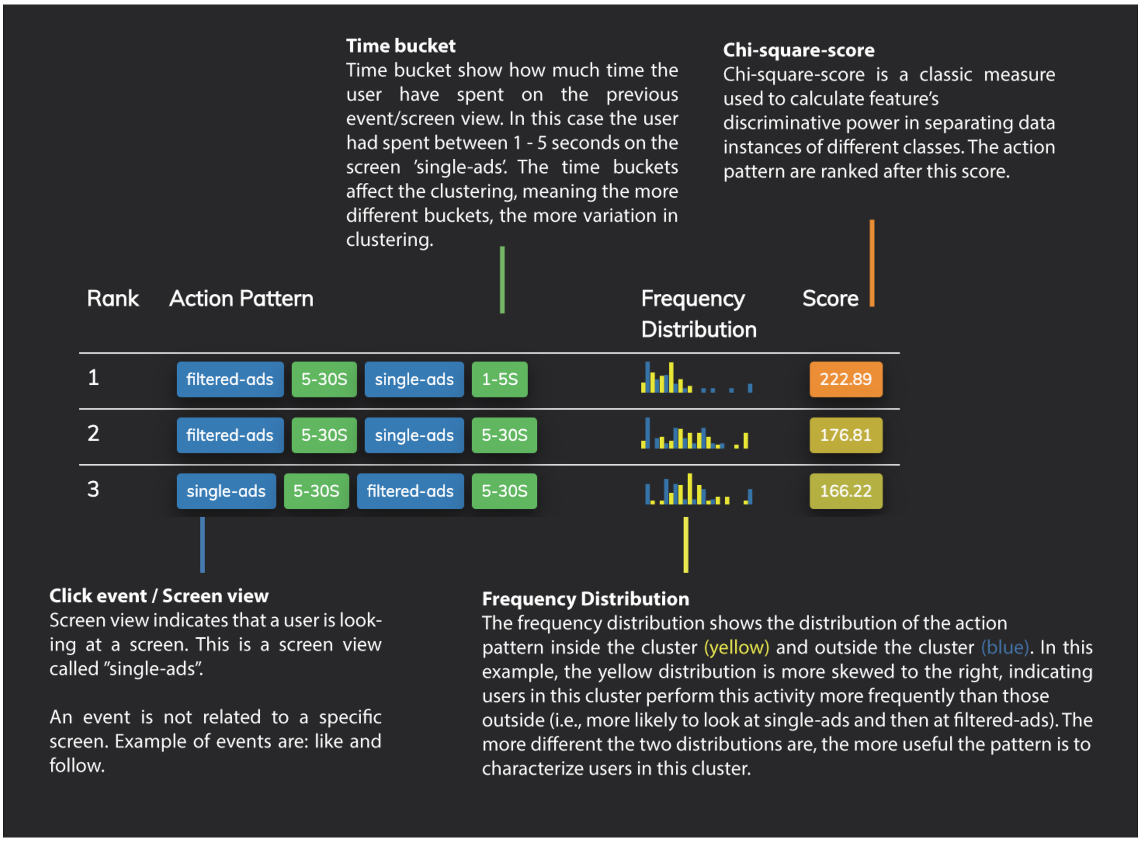


Figure 9.6: Pop-up with information about the features presented in the action pattern tab.

In all tabs various information is displayed. Most of the tabs are structured so that easily accessible information is presented as numbers at the top, and graphs or histograms of data that require a bit more mental activity are lined up below. The ambition was to implement Tufte's terms *data-ink-ratio* and *data density* in this presentation of information. Little emphasis is put on decorative visualisation that might clutter the screen. The extra visual clues that remain are for instance the borders around the graphs and numbers. Those visual refinements are intended to support the operator in distinguishing the data presentations from each other and are an example of clarification of the structure. Details in the graphs and histograms, such as a presentation of how many users that a bar represents, can be found when hovering over a bar or pie slice.

Graphs and histograms are an example of a well-balanced *data-ink-ratio* and therefore most information is presented in the form of graphs and histograms. The data presentation at the top will have the effect of starting on 0 and counting up to their correct values when the tab is displayed. This is intended to give feedback to the operator that the status of the tab has changed. The movement alerts the eye to look at the information, which is the purpose of the tool. The numbers have a larger font than its explaining text. The explaining text is also written in a more blending shade of grey. By using proportions, the operator can quickly focus on the data presented.

In the tab "Seller Properties", information about ads created by the users in the cluster is displayed (see Figure 9.7).

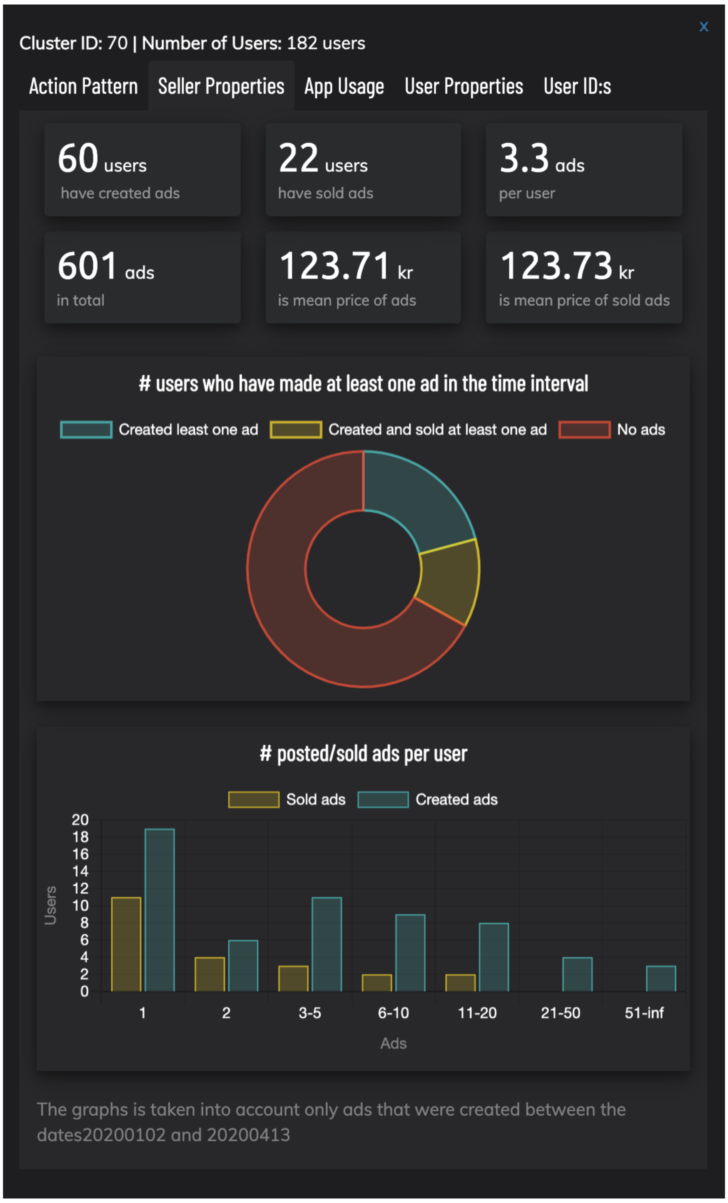


Figure 9.7: Seller Properties is the tab open.

The tab "App Usage" has information about how users interact within the application, and when they first made their Plick-account (see Figure 9.8). The metric "activity levels" that Plick are using are colour-coded into low: purple, medium: yellow and high: green. The same colours are used in the tool for the operator to recognise.

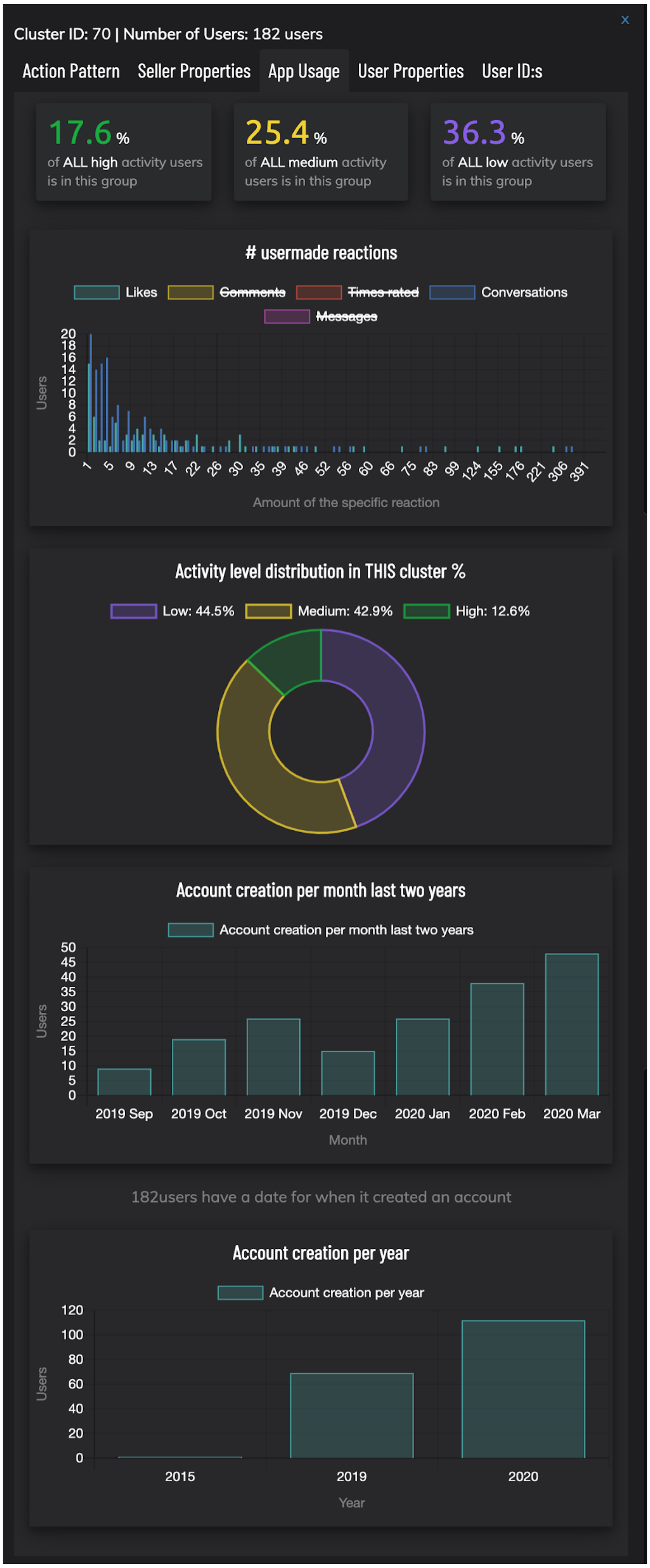
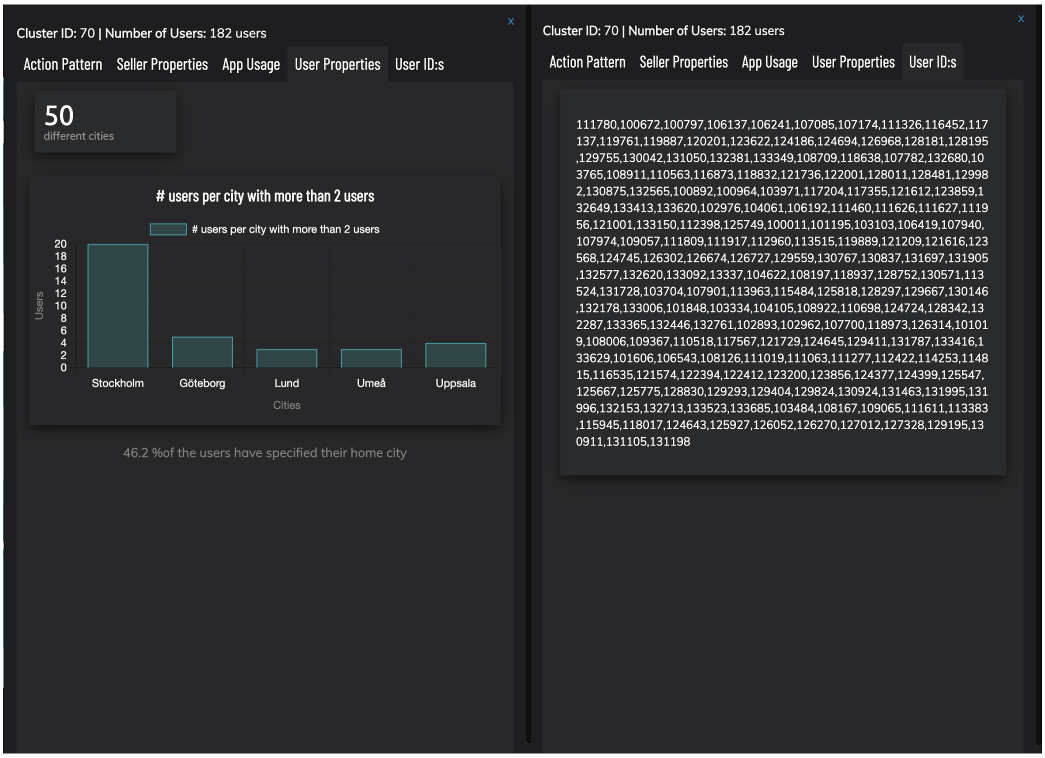


Figure 9.8: App Usage is the tab open.

In the histogram titled "# usermade reactions" one will find a few strikethrough labels (see Figure 9.8). This is intended to act as a *signifier* that the graphs and histograms *afford* the function to hide and show datasets. The choice to hide some data by default (Comments, Times rated and Messages) is intended to act as a demonstration of the functionality to the operator.

The tab of "User Properties" shows a histogram of users per city. In Figure 9.9a it can be seen that there are 50 different cities represented among 46.2% of the 70 users in the cluster. Out of these cities, only cities with more than two users are displayed. Cities are counted as the same city regardless of the letters being capitalised or lowercase. The use of spacing before or after the name is removed. Cities with å, ä and ö can be written with or without its special sign. The input of "Malmö" and "malmo" are considered the same city. However, if a user misspells the city or writes more than one place, it will not count as the same city. The tab "User ID:s", displays all the ID:s of the users in the chosen cluster (see Figure 9.9 b).



(a) User Properties (b) User ID:s

Figure 9.9: Tabs displaying user properties and user ID:s.

9.2.2 Calculating the information

As described in *6.3.3 Data transformation*, the ID:s of the users is transformed into a format required by RHC. To be able to fetch the added information about the users, the ID:s had to be translated back to their original numbers.

The code fetching the extra information is executed every time the operator clicks a cluster. The calculation of the extra information takes more time to complete with an increasing number of users in the analysis. This required the code to be effectively written. If not, the operator of the tool will experience loading time each time a new cluster is clicked. Loading time does not affect any conceptual model but it might decrease the satisfaction of using the tool.

Throughout the creation of the tool, the intention has been to create a *satisficing* inclusion and presentation of extra added information. By looking at how the statistics page of Plick is used today, it was found that the statistics page was used only for shorter time periods rather than for hours or days. With that in mind, the time using this tool should be as short as possible to lower the barriers of using the tool. The operator must be able to find satisfactory information in an efficient and effective way. As mentioned earlier, the reclustering takes time to perform but that function is not supposed to be used often during one user session of the tool. There is plenty of data available in Firebase. The choice of which information to include, and how to display it in graphs, histograms and static numbers are made in a way of *satisficing* the operator. It is reasonable to assume that the information can be further developed when the tool has become a more regular part of the development process.

9.3 Usability testing

In order to identify whether the tool achieves the goal of conceptualising complexity, usability tests were made with employees at Swace Digital. The result of the testing is found in Table 9.1, which contains information about the tasks, translated from Swedish to English, the success rate of the task and the average time spent on the task. The protocol for the usability test and its exact formulations of questions is found in Appendix C. Results from the questionnaire and the second meeting with the participants is discussed throughout this section.

Table 9.1: Summary of the result from the usability testing.

Task	Success rate	Average time
How many users are included in the analysis?	8/8	1:42 min
How large a portion of all users have a high activity level?	7/8	
Find the cluster (among the highest level of clusters below the root cluster) that seems to have most new users from 2020.	6/8	3:42 min
How large of a portion of the users in the previous cluster have posted more than five ads?	8/8	
Find the action pattern with the highest score.	7/8	1:42 min
How much time have the users spent on these screens or events?	8/8	
What screens are included in the group "browse-users"?	8/8	0:42 min
Reconfigure the clustering. It should include the following changes:		5:44 min
- Exclude the screens with the highest action pattern score.	8/8	
- Group all screens that belong to the feed tab. Name the group "Feed tabs"	6/8	
- Exclude all time variables from the analysis.	8/8	
- Change so that the action pattern includes three screens/events in a row.	8/8	
Among the new clusters, find the cluster with the highest mean value for sold ads (only among the highest level of clusters)	7/8	1:18 min

In total there were four participants that did not fail on any task. The rest of the failures could be categorised into three types of failures. The first category of failures included failures in identifying the correct cluster. When performing the task of finding new users and the cluster with the highest χ^2-score, two participants picked the wrong cluster. When talking about it with the participant during and after, it was clear they knew where the information was to be found. Most failures in this category occurred due to carelessness while searching for the answer among the clusters.

The second category of failures include failures because of misinterpretation of the task. In the first and last task in Table 9.1 there is a scored failure because the participant answered to a slightly differently interpreted task. One participant found the number of high active users instead of the share, and one user found the cluster with the highest mean for created ads instead of sold ads. In spite of these failures the users have a pretty good knowledge of how to find the needed information. However, they might need to use a little more time for comparison between clusters. Alternatively, the failures indicate that a further development of a functionality of comparing clusters could be of use in the tool. In an actual situation of using the tool, the participants will search for information based on tasks defined by themselves and what they know they can do with the tool. In such a situation, the operator would find the share instead of the amount if that was the executed task.

Finally, the third category included failure because the participant needed too much support in order to perform the task. The most troublesome task was to group screens belonging to the feed tab. Almost all participants got help finding the correct screen names, since only a few of the participants had background knowledge of Plicks' screen view names. Two participants asked for too much help, and the performance could therefore not be defined as successful. During the task, one participant dragged the screens into each other, which is not supposed to be possible. That is therefore an error of the interface design that should be fixed.

As can be seen in Table 9.1, nearly all tasks have a very high success rate. The participants had no trouble finding specific information about clusters, such as how many users that were included in the analysis. All participants understood that they had to click on the largest cluster in order to find information about all users. This signals a correct conceptual model of how hierarchical clusters are visualised. All participants were also able to find what screens belonged to the group "browser-users". It suggests that the participants have a conceptual model of how screens can be grouped in the analysis, and where to find those groups.

Some participants expressed confusion whether a screen could belong to several groups in the grouping part of the reclustring phase. This indicates a misunderstanding of the cluster analysis since this would make little sense in the clustering. However, the intended operator of the system will have had more introduction than the participants in the test, so the misunderstanding is not grave. Nevertheless, the same participants were able to find information in the tool and make interesting analyses of a given cluster.

Reconfiguring the clustering was the task that the participants spent most time on. It was not surprising since it was by far the hardest task to perform, and also the task with most subtasks. When the loading of new clusters was done and the participants could see a new presentation of clustering, a common reaction was one of interest and surprise, as if understanding the chosen settings in the recluster phase had real impact on the result.

The last task of the usability test, to analyse one cluster, was executed well by most testers and most analyses seemed accurate. Some participants expressed that the specific cluster, with a value of 4.5 created ads per user, was a cluster with high density of selling users. These types of conclusions are very hard to make when not familiar with Plick and its user-data. In a neighbouring cluster, the value for created ads per user were close to 15, which was a more distinctive cluster of sellers. If given more time to compare clusters, the participant would probably have realised that 4.5 created ads per user was not enough information to confirm a cluster of sellers but it is simply one indicator of sellers. Still, the participant seemed to know where to find the information and with more time they could have made a more well-informed conclusion. More knowledge did result in more developed and faster analyses of the user groups. Since the tool can be used and understood even by inexperienced operators, the tool can be seen as *satisficing* in clustering user groups based on clickstream data and visualising them in an intuitive way. The tool seems to give the operators enough visual clues for the operators to achieve a conceptual model of the analysis and its results.

Many participants described their experience of the tool in positive terms and expressed a potential usefulness of the tool. Even the participants struggling with some part of the tasks expressed that they thought they would have found useful information if they used the tool in their work. Even so, some concepts were discovered to be more complicated to understand by operators with less experience with Plick.

All in all, the tests showed that the tool can be used in a *satisficing* way by novice users, and very well by users with a little bit more experience. This indicates users with access to the tool on a daily basis could find value in the tool. The reason some participants failed was for most part not very severe. Some errors were due to misreading or misinterpretation of the task, but it is possible that the participant would have found the information if he or she had been looking for the correct value.

The questionnaire showed, similar to what the participants expressed during the test, that most participants found the tool easy and meaningful to use. Most of the participants answered that they believed they would use the tool. The questionnaire highlights that some participants did not feel totally comfortable using the tool. It was the only question that scored an answer of the least satisfactory value on the Likert scale. This might have been because some participants had previous experience neither of Plick nor of clustering, or because of the situation of being observed and recorded while testing a new tool. During the testing sessions and in the answers to the questionnaire, it could be observed that some knowledge about Plick was necessary in order to complete the tasks in a confident way. Extensive knowledge about Plick's data also made the analyses of the information superior. The participants with the highest confidence within the test group also chose the lowest values for needing more knowledge about Plick. The person finding the tool hardest to use expressed needing more knowledge both of Plick and clustering to use the tool. It is not very likely that operators of the tool have this extensive knowledge that only main developers of an application can have. Some of the participants in the usability tests might have been overqualified and does not reflect a situation that is likely to occur.

Most operators seemed to have conceptual models that helped them guess what their actions within the tool would lead to. Even though most operators did not evaluate themselves to have extensive background knowledge about clustering and did not believe an operator must have extensive knowledge of clustering, they could use the tool to gain insights. The visual clues of the tool seemed to be guiding the operators. The results from the questionnaire are further presented in Table 9.2. The mean value of the more positive questions are closer to 4 than the mean value of more negative questions are close to 1. This indicates a positive experience of using the tool.

Table 9.2: Summary of feedback from the questionnaire. 1 represents "I do not agree at all" and 4 represents "I totally agree".

Question	Mean value
I think I would use this tool	3
I think this tool was unnecessarily complex	1.5
I think this tool was easy to use	2.625
I think I would need more knowledge about Plick in order to use this tool	1.875
I think I would need more knowledge about clustering in order to use this tool	1.5
I think most users with a developer background would be able to learn how to use this tool	3.25
I thought that the tool was hard to use	1.5
I felt confident when using this tool	2.125

Later on, the participants were asked follow-up questions regarding their understanding of the analysis. The follow-up questions showed that the participants for the most part understood the meaning of the analysis, and what it was based on. Most participants were able to explain in a simplified way how the clustering algorithm operates. A common explanation of the algorithm was that the algorithm takes in a large amount of data that it groups based on pre-set parameters that the operator can define, and the information can be used in order to understand the users. When the participants were asked how much they understood while performing the first usability test, most responded that they understood what they saw, but not always what happened "in the background". It seems like the participants understood how to use the results they could see in the tool, even though some of them did not understand the underlying algorithm producing the results.

Part 4: Conclusion

10. Conclusion

In this thesis, it was examined how clustering algorithms could be used in order to identify user group characteristics based on clickstream data within an e-commerce application. Through literature studies of common algorithms used in clickstream data clustering, RHC was selected as a suitable method for the task. RHC has been shown to effectively cluster users based on clickstream data. The algorithm has the ability to capture fine-grained differences between users in an abundance of data.

The thesis also aimed to identify how clickstream data could be pre-processed in order to perform an efficient and rewarding segmentation of users. We could see that the data analysed had to be pre-processed for RHC to produce valid and meaningful results. By applying *feature selection*, the quality of the data set was increased, and the curse of dimensionality could be avoided to a great extent. The option of reconfiguring the cluster analysis enabled further pre-processing of the dataset. This required the operator to have an understanding of how including or excluding variables might affect the analysis. Usability tests showed that the default settings offered in the analysis acted as visual clues in order to support the operators' conceptual model of the analysis and the tool.

The thesis did also investigate how the complex analysis could be conceptualised in the tool in order for an operator to use it in a correct and efficient way. For the visualisation tool to be useful, the analysis and results must be understandable by its operator. Comprehension of the result could be supported in several different ways. Firstly, the natural visualisation of hierarchy was discovered to be a visual form that is easy for the operator to comprehend. Secondly, added information to each cluster was shown to act as an amplifier of the action pattern result created by RHC. Furthermore, it was shown that the added information could be used as the main source for analysing and identifying user groups. The operator can therefore validate the identification of user groups in both ways. Lastly, by using the functionality of reconfiguring the clusters, the operator got a first-hand experience of how the pre-processing could affect the analysis, and many operators expressed an "aha-moment" when the new clusters appeared after reclustering. The reclustering could thus reinforce the operators' conceptual models of the analysis, and consequently reinforce their understanding of the results.

All in all, we could see that the interactive tool seems, to a great extent, to be able to capture and visualise user groups based on clickstream data in an intuitive way. The usability study showed that the operators of the tool, despite a very short introduction to the tool, were able to both understand the clustering results and reconfigure the cluster analysis. The intended operators are supposed to get more time to learn to use the tool than the test participants, and they are also supposed to have more knowledge of the screen views and events analysed by the algorithm than the test participants had. The tool seems to succeed in conceptualising the complexity and make it understandable to the operator, at least to the extent that the operator can use the tool in a rewarding and *satisficing* way.

The work of conceptualising the results of the analysis and the analysis itself has been made with the ambition to achieve a *satisficing* result. It could be the case that the full potential of the tool is not reached because the operator stops at a *satisficing* result. The operator of the tool also analysed the results in a *satisficing* way. To reach a fuller understanding, more work is required of the operator. This might not be a problem since there are a multitude of possible valid outcomes from the analysis, but it can be of importance to remember. There simply are many correct ways of using the tool.

Hereafter, it would be of interest to include other clustering algorithms in the analysis and compare them to RHC. The RHC study highlights that other methods did not reach the same accuracy as their algorithm, but it might be case dependent, and could therefore be of interest to examine further. It would also be of interest to examine how the default parameters could be improved in order to enable an even better pre-process dataset. All default parameters were chosen with much consideration, but we believe the analysis could be additionally optimised by conducting studies focused mainly on the parameters. Finally, an additional study including more participants with various education and background knowledge could enable a deeper understanding of how to conceptualise complexity. The better an operator understands how the tool works and how to utilise the results, the better the operator can gain understanding of the users analysed.

References

Aggarwal, C.C., Reddy, C.K. (2013), Data Clustering: Algorithms and Applications. CRC Press LLC, Philadelphia, PA, United States.

Altman, N. & Krzywinski, M. (2018), "The curse(s) of dimensionality", Nature methods, vol. 15, no. 6, pp. 399-400.

Aziz, A. (2017), Customer Segmentation based on Behavioural Data in E-marketplace, Uppsala universitet, Institutionen för informationsteknologi.

Bandyopadhyay, S., Saha, S. (2013), Unsupervised classification: Similarity measures, classical and metaheuristic approaches, and applications. Springer, Berlin, Germany.

Barnum, C.M. (2010), Usability testing essentials: ready, set-- test. Morgan Kaufmann Publishers, Burlington, MA, United States.

Bennett, K.B., Flach, J.M. (2011), Display and Interface Design. CRC Press, Boca Raton, United States.

Blocket, n.d, Om Blocket. Available at: https://www.blocket.se/om (2020-05-04).

Dumas, J.S., Loring, B.A. (2008), Moderating Usability Tests, The Morgan Kaufmann series in interactive technologies. Morgan Kaufmann Publishers Inc.

Krisinformation, (2020). Folkhälsomyndigheten om distansarbete. Available at: https://www.krisinformation.se/nyheter/2020/mars/folkhalsomyndigheten-uppmanar-till-distansarbete (2020-04-23).

Elvander, A. (2015), Developing a Recommender System for a Mobile E-commerce Application. Uppsala universitet, Avdelningen för datalogi.

Gibson, J.J. (1986), The ecological approach to visual perception. Psychology Press, New York, pp 127-143.

Han, J., Kamber, M., Pei, J. (2012), Data Mining: Concepts and Techniques. Elsevier Science & Technology, Waltham, United States.

ISO, nd. 9241-11:2018(en) Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. Available at: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en (2020-05-22).

James, G., Witten, D., Hastie, T., Tibshirani, R. (Eds.). (2013), An introduction to statistical learning: with applications in R, Springer texts in statistics. Springer, New York, United States, pp. 373-413.

Klein, G. A. (1993). "A recognition-primed decision (RPD) model of rapid decision making" in: G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), Decision making in action: Models and methods, pp. 138–147. Ablex Publishing.

Likert, R. (1932), A technique for the measurement of attitudes. New York, United States.

Liu, Y.L. (2014), "Research on the New Trend in E-Commerce Industry: Mobile E-Commerce" in: Advanced Materials Research, vol. 926-930, pp. 2634-2637.

Norman, D.A. (2013), The design of everyday things, Revis and expand edn, Basic Books, New York, New York, United States.

Tradera, n.d. Om oss. Available at: /info/om-oss/ (2020-05-04).

Shopify, n.d. Ecommerce Definition - What is Ecommerce. Available at: https://www.shopify.com/encyclopedia/what-is-ecommerce (2020-05-06).

Simon, H. A. (1996), The sciences of the artificial, 3rd edn, MIT Press, Cambridge, Mass. Chapter 2 "Economic Rationality: Adaptive Artifice", pp 25-50.

Support Google, n.d. Session calculation. Available at: https://support.google.com/firebase/answer/9191807?hl=en (2020-05-04).

Swace, n.d. Plick. Available at: https://swace.se (2020-05-04).

Tufte, E.R. (1990), Envisioning information, Graphics Press, Cheshire.

Van den Hoven, J., Blaauw, M., Pieters, W., Warnier, M. (2014). Revised 2019. Privacy and Information Technology in: Stanford Encyclopedia of Philosophy. Available at: https://plato.stanford.edu/entries/it-privacy/ (2020-05-13)

Vedder, A. (2008), "Responsibilities for information on the internet" in: Himma, K.E., Tavani, H.T., The Handbook of Information and Computer Ethics. Publisher:

John Wiley & Sons, Incorporated, Hoboken, United States, pp. 339-360.

Voigt, P., von dem Bussche, A. (2017), Material Requirements in: The Eu General Data Protection Regulation: A Practical Guide, Springer, New York, United States, pp. 87-138.

Wang, G., Zhang, X., Tang, S., Zheng, H., Zhao, B.Y. (2016), Unsupervised Clickstream Clustering for User Behavior Analysis, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16. Presented at the 2016 CHI Conference, ACM Press, Santa Clara, California, United States, pp. 225–236.

Zoom. (2020), Video Conferencing, Web Conferencing, Webinars, Screen Sharing. Available at: https://zoom.us/ (2020-04-23).

Yang, Y., Pedersen J, O, Pedersen. (1997), A Comparative Study on Feature Selection in Text Categorization. no publisher.

Zhang, X. RecursiveHierarchicalClustering. (2020), GitHub repository. Available at: https://github.com/xychang/RecursiveHierarchicalClustering (2020-02)

Interviews

(Heibert, Jimmy; CEO Plick. 2020. Interview 3 February.)

(Alex, Christian; Developer Plick. 2020. Interview. March)

(Rask, Elliot; Developer Plick. 2020. Interview. March.)

Appendix

A: Data points for one action

Name in database	Example	Meaning
user_id	100872	A unique ID for the user
date	20200310	The date of the action
event_name (an event can either be a screen_view or something else)	screen_view	Screen view indicates that a user is looking at a screen
	like, follow	An event that's not related to a specific screen. For example a user liking an ad, or a user clicking on follow on another user. The event is triggered without changing what screen is displayed.
screen	feed/tab0	Screen tells which screen the user is looking at when the event_name is equal to screen_view. In the example the user is looking at the tab with popular items on the feed page. This view is the start page of the application
previous_screen	myprofile/tab0	Previous_screen tells what screen the user showed before the active screen. Here the user looked at its own profile, in the tab showing its ads.
session_id	1583829847	Firebase assigned a session_id to every action. This indicates which actions belong to the same session.
timestamp	1583829858194316	Unix time in nanoseconds
previous_timestamp	1583829855813004	Unix time in nanoseconds for the previous action with the same event_name. If the action is a screen_view then, the previous timestamp gives information about the previous shown screen. If the

	action is a like, then the previous timestamp indicates the previous triggered like-event. This number differs approximately +0,19 seconds from the actual previous timestamp.

B: Occurrence of different types of screen views

The names used to describe the screen view in the table have been simplified for easier understanding and screen views with multiple names have been renamed. Screens views with a percentage of less than 1% of the total usage is omitted from the table. No event had a percentage larger than 1%.

Screen view	Explanation of screen	Occurences	
		#	%
single-ad	A single ad, not mine	281953	52.6
single-profile-selling	Tab with items for sale on another users' profile page	85110	15.9
feed-popular	Start page of Plick. Items arranged based on popularity	33029	6.2
single-profile	A profile of someone else	26964	5.0
filtered-ads	An overview of ads after using a filtering function	24390	4.6
single-profile-likedAds	Tab with liked ads on another users' profile page	11265	2.1
myAd	A single ad, my ad	9735	1.8
myProfile-selling	Tab with items for sale on the users' own profile page	6235	1.2
myProfile-likedAds	Tab with ads the user has liked on the users' own profile page	6233	1.2
conversation-selling	An overview of all conversations started on my ads	5424	1.0
Other screens	43 different screen views, including like, news, create-ad and follow.	45223	8.4

C: Information page

Hierarchical Clusters

About This Tool

We build a tool to automatically discover the natural formation of user categories in Plick, based on their behaviour as captured by clickstream data. A user's clickstream is a sequence of click events generated by the user when he or she uses the app. At a high-level, we assume that human behaviour would naturally form clusters or groups, because large user populations tend to break into different types of users according to their habits, goals, and personalities. Our tool seeks to discover these natural clusters among users, where each cluster represents a specific user type or behavioural pattern. To understand the meaning of these resulting clusters (since they are unlabeled and do not necessarily match preconceived categories), we identify key behavioural patterns for each cluster that are primarily responsible for distinguishing users in the cluster from others. These key patterns thus can serve as the cluster labels.

Our tool builds clusters **hierarchically**, where clusters are nested, and bigger clusters represent the higher level categories of users. Smaller, sub-clusters represent more fine-grained user types. The **circle size** represents the number of users in a cluster.

How to Use this Tool

The clusters displayed are generated from Plick users' clickstreams. They represent the major user behavioural categories in Plick. You can browse different clusters to understand the detailed user behavioural patterns. .

By **double-clicking** on any cluster, you can **zoom** in on the cluster for more focused inspection. To zoom out, simply double click on the current cluster.

Clicking on a cluster will show window to the right with more detailed information about this cluster. Here, we explain how to read the information in the pop-up window. First, on the top of the window, it shows the **ClusterID** and the **Number of Users** in this cluster. Then, below that is a tab bar indicating different types of information. In the first tab, we show a list of **Action Patterns** that can characterize the behaviours of users in the cluster. Each row contains one Action Pattern. These Action Patterns are ranked by their distinguishing power in classifying users in this cluster (more important patterns on top). We only display the most important Action Patterns in the window and the rest are omitted because they are *significantly* weaker.

> The first column shows the **Rank** of the Action Pattern. A pattern with a higher ranking means this pattern is more important in classifying users in this cluster.
> The **Frequency Distribution** shows how frequently the Action Pattern does appear in users in this cluster versus outside the cluster.

You can see how users in this cluster are different from users outside of the cluster on this particular Action Pattern. The red bars show the pattern frequency distribution (PDF) for users within the selected cluster. As a baseline comparison, the green bars show the pattern frequency outside the cluster. In this example, the red distribution is more skewed to the right, indicating users in this cluster perform this activity more frequently than those outside (i.e., more likely to read whispers sequentially). The more different the two distributions are, the more useful the Pattern is to characterize users in this cluster.

> Finally, the **Score** column shows the Chi-Square score of the Action Pattern. We rely on this score to rank Action Patterns (a higher score means the pattern is more important). Socres are colored from higher to lower.

The other tabs of the information windos is **Seller Properties**, **App Usage**, **User Properties** and **User ID's** Under these tabs are various information about the users in the cluster presented. Such as what city they belong to, how long they have had their account or how much the users in the cluster comment or rate.

D: Test protocol

First interview

Fråga om det är okej att vi spelar in mötet.

Introduktion till vårt exjobb

Inledande frågor till testpersonerna

- Har du jobbat i någon form med Plick? Hurdå? (utvecklat eller dylikt)
- Hur bra koll har du på Plick och dess användardata?
- Hur bra känner du till Plicks olika screens, screennamn och events?
- Vad är din erfarenhet av klustring?
- Vet du vad hierarkisk klustring är?

Snabb presentation av verktyget

Scenarios

	Uppgift	**När är uppgiften klar**
1	Läs informationssidan, och säg till när du är klar.	När användaren säger att hen är klar.
2	Klicka runt i systemet i några minuter minuter och bekanta dig med de olika funktionerna. Bekanta dig med vilken information som presenteras. Säg till när du är klar, eller så säger vi till.	När det gått 2 minuter, eller när användaren säger att hen är klar.
3	Hur många användare är inkluderade i analysen, samt hur stor andel av alla användare är högaktiva användare?	När rotklustrets info-boxs' tab app usage är öppen. 19,8%
4	Hitta klustret som verkar ha flest nya användare 2020. På ett ungefär, hur stor andel i detta kluster har lagt upp fler än 5 annonser? (Endast högsta nivån av kluster)	När klustret med flest användare är öppet på seller properties-tabben.
5	Hitta det action pattern som har högst score. Hur lång tid har användaren spenderat på dessa	(kluster 47 övre höger har score 458,4 - single-ads och single-

	sidor eller event?	profile) (1-5 sek)
6	Vilka sidor ingår i gruppen “browse-users”.	browse/users, browse/tabs1, feed/tab0
7	Ändra klustringen. Den ska innebära detta: - Hitta det action patter som har högst score och exckludera dessa screens från klustringen. - Gruppera ihop alla screens som hör till feed tab (alltså tabbarna på “start/hem” sidan). Döp gruppen till “Feed tabs”. - Excludera alla tidsvariabler ur analysen. Ändra så att action pattern inkluderar 3 screens/events i rad. - Klustra om Säg till när du är klar.	När man klickat på recluster, och godkänt popup. (gruppera feed/tabs0, feed/tabs1, feed/tabs2)
8	Hitta klustret med högst medelvärde av pris för sålda annonser. (Endast högsta nivån av kluster)	
9	Utifrån vad du ser i informationsrutan i det här klustret (ett specifikt utvalt), vad tror du detta är för typ av användare? säg till när du är klar.	

Avslutande frågor:

Beskriv något

- Bra
- Som kan förbättras.

Second interview

Skulle du i egna ord kunna förklara hur klusteranalysen fungerar?

Vad betyder ett action pattern?

Förstod du vad du gjorde och vad som hände när du gjorde testet? Förklara gärna!

E: Questionnaire of Usability Tes

RHC: SUS formulär

Detta är ett formulär för att utvärdera interaktionsverktyget RHC på system användbarhetsskalan.
Tack för att du svarar!
*Obligatorisk

1. Jag tror att jag skulle använda det här verktyget *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	◯	◯	◯	◯	Jag håller med helt och hållet

2. Jag tyckte att verktyget var onödigt komplex *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	◯	◯	◯	◯	Jag håller med helt och hållet

3. Jag tyckte att verktyget var enkelt att använda *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	◯	◯	◯	◯	Jag håller med helt och hållet

4. Jag tror att jag skulle behöva mer kunskap om Plick för att kunna använda verktyget. *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	⬭	⬭	⬭	⬭	Jag håller med helt och hållet

5. Jag tror att jag skulle behöva mer kunskap om klustring för att kunna använda verktyget. *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	⬭	⬭	⬭	⬭	Jag håller med helt och hållet

6. Jag tror att de flesta med utvecklarbakgrund skulle kunna lära sig att använda verktyget fo *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	⬭	⬭	⬭	⬭	Jag håller med helt och hållet

7. Jag tyckte att verktyget var besvärlig att använda *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	⬭	⬭	⬭	⬭	Jag håller med helt och hållet

8. Jag kände mig självsäker i min användning av verktyget *

Markera endast en oval.

	1	2	3	4	
Jag håller verkligen inte med	⬭	⬭	⬭	⬭	Jag håller med helt och hållet

9. Har du några frågor/funderingar eller åsikter så får du gärna dela med dig av dem här! :)