Adaptive User-Centric Recommendations: A Holistic User Context-Aware Recommendation Algorithm

A thesis presented to the

Department of Computer Science
Faculty of Science
University of Zimbabwe

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

by

Tatenda Duncan Kavu

Supervisors:

Peter G. Raeth PhD
Kudakwashe Dube PhD
Gilford T. Hapanyengwi PhD

April 23, 2021

0.1 Abstract

Information filtering on the World Wide Web has grown widely since information overload has become a serious problem. Recommendation systems, the topic of this thesis, filter and apply the mass of available information to help users make personalized selections of products and services. However, algorithms of today lack dynamism, user-focus, and an element of surprise in recommendations. Literature recognizes that user context contributes to decisions made. Incorporating context in recommender systems (RSs), can result in better novelty and serendipity in RS. The literature also recognizes that context-aware recommendation systems (CARS) enable novelty, dynamism and serendipity. Yet, there still exists the problem of inadequate or limited user contexts in CARS, thus causing limited representation of user tastes. This thesis addresses the problem of inadequate contextual parameters that define a user in CARS. This thesis' research investigated the incorporation of adequate user-contextual parameters in CARS. Systematic literature review approach was used to establish the inadequacies in RS evaluation methods and to investigate the theory of user-contextual parameters in CARS. Ethnography and data mining, were the main methods applied to come up with an optimal set of user-contextual parameters which can be incorporated in CARS. Web mining, collaborative filtering and contextual modeling were applied to create a technique of estimating user preferences. Computational complexity, user and system-centric evaluation methods were the main evaluation techniques applied to evaluate the method that estimates user preferences. This thesis contributes, a User-centric Evaluation Conceptual(UEC) framework, Social, Cultural, Psychological and Economic (SCuPE) framework. Optimal set of user-contextual parameters to be incorporated in CARS, and a Holistic User Context-aware (HUC) recommendation algorithm that applies the SCuPE framework. The resulting algorithm showed improved dynamism, novelty, and serendipity with an average of 84% novelty and 85% serendipity when compared with existing CARS. The UEC framework developed in this research is a new approach to look at existing evaluation methods, their shortfalls and possible ways to improve the shortfalls. The SCuPE framework brings a better understanding of user contexts and factors that affect users' change of preferences. An optimal set of user-contextual parameters solves the problem of using

any user context in estimating user preferences. Finally, the HUC algorithm provides a better alternative when compared with the status quo, in implementing context-aware recommendation algorithms that offer dynamic, novel and serendipitous recommendations.

Preface and Declaration

The study described in this thesis was carried out in the Faculty of Science, Department of Computer Science, University of Zimbabwe, during the period 1 July 2015 to 13 March 2020. This thesis was completed under the supervision of Dr Peter Raeth, Dr Kudakwashe Dube and Dr Gilford Hapanyengwi. This study represents original work by the author and has not been submitted in any form to another University. Where use was made of the work of others it has been duly acknowledged in the text.

Copyright ©2021 Tatenda Duncan Kavu

Acknowledgements

I would like to give thanks and appreciation to my supervisors, Dr Kudakwashe Dube who trained me in doing high profile research in computer science I benefited a lot from his experience and dedication, he used his own funds to fly from New Zealand to Zimbabwe to support this research project and also in publications and many conferences that I presented. Dr Peter Raeth was such a pillar in this research project, his support and guidance was exceptional, he also used his own funds to fly all the way from the United States to support this project, and he contributed a lot in publications and conference presentations. Dr Gilford Hapanyengwi was quite supportive from the initial stage of the research project up to the end, he used a lot of his time and his resources to support this project. I would also want to thank specifically Dr Benny Nyambo, Mr Benard Mapako and Mr Taurai Rupere from the computer science department who led all the administrative processes which the research project went through. Finally, I would like to thank my Family, Colleagues and Friends for their unwavering support and prayers.

Dedication

I dedicate this work to the author and owner of wisdom and Knowledge, the almighty God Jesus Christ. I dedicate this work to my beloved wife Yolandar Tatenda Kavu and my three daughters Shalom Maranatha Kavu, Omega Samara Kavu and Tamiranashe Kavu for their tremendous support through out the study. I also dedicate this work to my parents Tennison Kavu and Alpha Kavu who have been encouraging me since the day I opened my eyes in this world.

True education means more than the perusal of a certain course of study. It means more than a preparation for the life that now is. It has to do with the whole being, and with the whole period of existence possible to man. It is the harmonious development of the physical, the mental, and the spiritual powers. It prepares the student for the joy of service in this world and for the higher joy of wider service in the world to come (Education pg 13.1, Ellen G White).

Contents

List of Figures

CHAPTER 1

Introduction

This chapter introduces the current state of knowledge, missing knowledge and similar work that has been done. It also sets the tone by defining the problem statement, showing the structure of the thesis and its contributions.

1.1 Background and Motivation

Recommender algorithms have recently become a driver of online businesses in many domains especially e-commerce and social networks and they are now very familiar to the general public on platforms like YouTube, Google, Facebook, Amazon and Online Reservation platforms [Yan+14], [Che+18], [CD17], [Lu+15], [YM17]. The amount of content on the World Wide Web has risen exponentially [Yan+16], [Var15], [Lu+15]. Recommender algorithms help users get information speedily and conveniently. This has proved to increase revenue in e-commerce and user satisfaction in other services provided online [JWK14], [Var15], [Pag+16].

It has come to the attention of researchers in recent years that research needs to be done in recommender algorithms to study the user data deposited on the web during interaction with web services [Zha13], [BAC17], [NT14]. Understanding a user's preference can be fundamental to help that user get satisfying information or services conveniently. ACM conference on recommender algorithms have existed since 2007, and this proves the fact that many researchers are working in this area [Var15], [Fat18]. However there are some challenges which include changing of user preferences [Agh18], [HCS18]. Many

Contextual Recommender Systems(CARS) incorporate few contexts which many a times tell a little or no story about users [Sun+16], [Har+17], [Ung+16]. There is a lack of adequate novel and serendipitous recommendations [RAL16], [APO18], [Har+17].

Recommender algorithms solve the problem of finding pertinent information or items on a system that is offering a service [Sun+16], [KFG15]. Research has demonstrated that users choose products/items based on usability, cost, consequences of selecting the item, experience with the product, feelings based on experience and social impact of the item [Kar13], [Kni+12], [Bao+16], [Yan+14]. This can be illustrated to a certain extent: for example on vehicle online markets. If a user wants to buy/search a car he/she is driven by certain brand experience, social status, usability of that car, financial consequences after choosing a car. All these factors demand attention when making recommendations [Eks+15].

Recommendation methods are heavily affected by the dynamics of user preferences and lack of interest by users to supply information. Researchers have pointed to this as an area for further investigation [Bee17], [IFO15]. What a user likes today is not likely to be the same tomorrow. This has been seen with online news providers, retail shops, tourist bookings sites and online reservation systems. To generate recommendations for users, recommenders use two main methods to gather user data: explicit and implicit [Bee17], [PC10]. Explicit methods gather user data from ratings, reviews and votes, whilst implicit methods use server-logs, search and browsing history. These methods are worth investigating, especially these days were online industry always finds ways to be continuously competitive.

Developers mainly test RS based on system-centric factors forgetting the most important user-centric factors which are important to the clientele [Lei14], [Lu+12], [Kni+12]. Lack of attention to user-centric factors leads to a loss of sales or a loss of service use. Thus, profit, business and jobs are lost. Ignoring user-centric factors is a deliberate move to ignore studying clientele and that result in designing algorithms which does not un-

derstand users thereby giving irrelevant recommendations [Kni+12]. There are user-centric factors such as buying decisions, user experience and user interactions that need to be evaluated effectively so that researchers produce competent algorithms.

Literature demonstrated that there is a trade off between algorithmic accuracy and user satisfaction and of late researchers have been mainly focusing on system-centric factors like algorithm accuracy [Lei14], [Pag+16], [KR12], [Kni+12]. Some work has given another perspective of shifting focus to user interactions, decision-making processes and user overall experience, which define the needs of a user [Tin16]. RS must be assessed based on improvement of user experience, interaction and decision making processes [KR12]. It is now widely acknowledged [Lei14], [Pag+16], [Kni+12] that it is the user-centric factors that determine if a recommender algorithm is going to increase user satisfaction or not. Generating recommendations is not enough since there is need to evaluate whether or not the needs of the user have been met [Her+04], [KR12]. Helping and providing good value to users should be the purpose of recommender algorithms. Hence, user-centric factors deserve to be one of the key foci in recommender algorithms [KR12], [Eks+15].

Recommender algorithm developers have always been looking to researchers to find ways to offer novel and serendipitous recommendations since users get used to the same recommendations. Some researchers have also found out that there are few theoretical and experimental studies on serendipity [LGS11]. Some work has been done by Saul Vargas on novelty and diversity [Var15] however the algorithm was not incorporating user-context and is highly likely to be affected by the dynamics of user preferences. Therefore some methods can be investigated that look into deeper aspects of user's preference models and how these can be simulated in computational environments to predict perpetual dynamic, novel and serendipitous recommendations.

1.2 Research Problem

Recommender algorithms lose relevance with time[LH14]. The evidence being low retention rates with time, lack of timely adaptive, user-specific, novel and serendipitous recommendations. This is also seen as timely inefficiency in recommender systems [JWK14], [Var15], [Pag+16]. CARS have been implemented to curb this challenge[Ver+12] however there still exists the problem of inadequate user contexts in CARS bringing about limited representation of users and their tastes [Ung+16], [Har+17]. This thesis addresses the problem of inadequate valuable user-contextual parameters that define a user in CARS. Failure to investigate this problem results in recommender algorithms recommending irrelevant items with time to the users [Agh18]. It is therefore imperative to investigate the problem, then offer a practical solution that can be implemented on different platforms such as e-commerce, real-estate and e-learning for testing purposes. The solution will bring a situation whereby users experience dynamic, novel and serendipitous recommendations on different platforms, thereby creating an environment of continuity in business.

Research Aim

The aim of this thesis is to investigate the incorporation of adequate user-contextual parameters in CARS to offer recommendations that adapts to dynamic user-contexts, guarantee novelty and serendipity.

Hypothesis

The incorporation of a comprehensive set of user-contextual parameters in CARS bring about an enhanced performance and improves dynamism, novelty and serendipity of algorithms in CARS.

Research Objectives

1. Establish the inadequacies in RS evaluation methods and user contexts.

2. Investigate and develop the theory of user-contextual parameters in CARS.

Table 1.1: Research objectives and corresponding research methods

Research Objectives	Research Methods
Objective 1: Establish the inadequacies in RS evaluation methods and user contexts	Systematic Literature Review
Objective 2: Investigate and develop the theory of user-contextual parameters in CARS.	Systematic Literature Review
Objective 3: Determine best user-contextual parameters which can be incorporated in CARS.	Ethnography Data mining
Objective 4: Use comprehensive set of user-contextual parameters to estimate user preferences and generate recommendations.	Web mining Action
Objective 5: Evaluate how the method in (4) performs with respect to dynamism, novelty and serendipity.	Quantitative Computational complexity

3. Determine best user-contextual parameters which can be incorporated in CARS.

4. Use comprehensive set of user-contextual parameters to estimate user preferences and generate recommendations.

5. Evaluate how the method in (4) performs with respect to dynamism, novelty and serendipity.

1.3 Research Methods

This section shows the summary of research methods which were applied to fulfill the objectives, the detail about the research methods is covered in chapter 3. Table 1.1 shows the research objectives and the corresponding research methods.

The first objective focus was to investigate RS evaluation metrics and methods. The intention was to investigate the short falls in present evaluation methods and metrics, since it made scientific sense to investigate the assessment methods which gave a green light to the products which were not performing as expected in the market. The investigation results created a road map to address the problems that were identified specifically in the design of RS. Systematic literature review was used to fulfill the objective. The second objective looked into the theory of user-contextual parameters in CARS, specifi-

cally the comprehensive parameters which define a user, since this was one of the major issues that was identified in the first objective. Systematic literature review and quantitative meta-analysis were applied to fulfill the objective. Objective 3 main focus was to identify an optimal set of user-contextual parameters which estimate better user preferences. Ethnography and data mining, were the main research methods applied on objective 3. Objective 4 focused on using the comprehensive parameters found from objective 2 to predict user preferences and generate recommendations. Web mining and action research approaches were applied to fulfill objective 4. The final objective focused on evaluating the outcome of objective 4 to assess if the method presented solve the research problem. To fulfill objective 5 quantitative and computational complexity were the main techniques applied.

1.4 Expected Contributions

This thesis is expected to contribute the following:

1. A framework to establish the inadequacies in RS evaluation methods.
2. A framework that defines a comprehensive set of user-contextual parameters.
3. An identification of a set of optimal user contextual parameters that best estimate the preferences of the users and hence led to best recommendations.
4. An algorithm to generate recommendations that consider contextual parameters to improve user preference adaption and user satisfaction.

1.5 Definitions of terms and concepts

User contextual parameters- define the context of a user during the time when he/she interacts with a recommendation system. Contextual parameters also includes the attributes of the user, such as: age, gender, location, mood, hometown, education background, time et.c). These parameters where incorporated in a context-aware recommendation algorithm using the contextual modelling approach, so as to solve the problem of inadequate user contexts in CARS.

CARS- represents recommendation systems(algorithms) which incorporate user-contexts either before, during or after generating recommendations. Context-Aware Recommender Systems (CARS) are known to produce highly relevant recommendations by exploiting context information that takes into account the user's preferences and situations and to emphasise the goal of being sensitive to changing user needs. This research project took the approach of using CARS so as to solve the research problem.

Evaluation- is the process of testing recommendation algorithms. In this research project, the HUC algorithm which was created was evaluated based on its ability to offer adaptive/dynamic, novel and serendipitous recommendations.

Holistic User-context- This is the comprehensive contextual information accessible to the recommendation algorithm and which defines a user profile/context. The SCuPE framework defines the nature of the comprehensiveness of a holistic user-context.

Adaptive recommendations- These are recommendations that cope with changing user contexts, a recommendation algorithm need to cope to the users' contexts so as to offer relevant novel recommendations.

1.6 Thesis Organization

The thesis is structured as shown on Figure 1, starting with chapter one which is introduction. Chapter two critiques the state of the art in recommender algorithms, it explains the current state in relation to user-context incorporation in recommender algorithms and how this affects user satisfaction. Chapter three explains the research methods which were applied to fulfill the research objectives. Chapter four explores the theoretical contributions on user-preferences and how recommender algorithms can harness this theory into computational models. Chapter five demonstrate a holistic user context-aware recommender algorithm. This chapter undertakes experiments in recommendations. Chapter six explains the contributions, conclusions, comparisons with related work, benefits, limitations, areas of improvement and future research directions.

It is also the chapter that concludes the whole thesis and explains the importance of all the contributions to recommender systems and how these change the world.

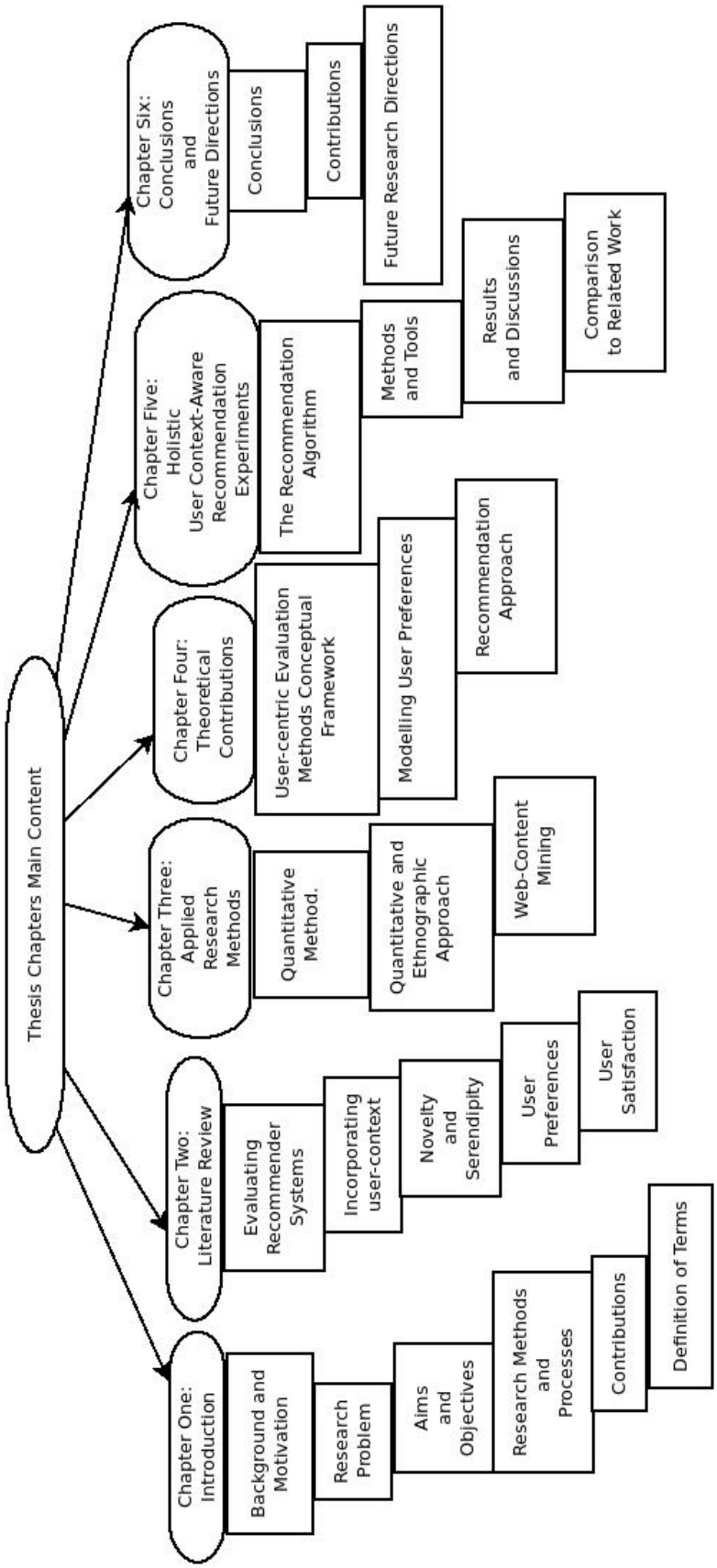

Figure 1: Thesis Structure

CHAPTER 2

State of the art in Recommendation Algorithms

This chapter critiques the current state of the art in recommender algorithms. The chapter explains why we need to focus more on a new design of recommender algorithms even though we have thousands of existing recommender algorithms. It exposes the gap and defends our view of the new approach to design recommender algorithms that meditate on individual preferences.

Since 2007, Association for Computing Machinery (ACM) has been running annual conferences on recommender algorithms and it is through these conferences that different methods of creating and evaluating recommender algorithms has extensively been discussed and analysed. It is also from these published works and many created algorithms that we got some insights about a paradigm shift that is happening in the design of a new type of recommender that takes contextual knowledge about a user in order to predict preferences.

2.1 Evaluation Methods for Recommender Algorithms

This section explores the shortfalls in RS evaluation methods. Recommender algorithms evaluation methods are classified in to two main classes which are user-centric and system-centric. Lately it has proved necessary to investigate and assess these two methods in order to pay attention to the details that make an RS effective and acceptable [Kni+12]. Literature proved that there is a strong trade-off between the accuracy of an algorithm

and the satisfaction of the user [Kni+12], [Lei14]. Researchers have not paid much attention to user-centric evaluation methods as they did to system-centric approaches [Lei14], [Cre+12], [KR12], [Kni+12]. It is then persuasive to look into user-centric factors such as user interactions, decision-making processes and overall experience which define user-centricity [Tin16].

The main aim of this section is to explore the literature and to show that existing evaluation methods lack the rigour to assess user factors which seem to be the prerequisite of user satisfaction. The section therefore formally provides the foundation for further investigation of techniques that can address current challenges. Terms to search for relevant literature were elicited from the research problem. All types of RS used in many domains such as: eLearning, news social media and e-commerce were included in the literature search and all other work without anything to do with evaluation of recommender algorithms were excluded. The results from the literature search demonstrated that there are many system-centric methods as compared to user-centric methods. The work that focused on user-centric evaluation methods missed the evaluation of the entire user-centric factors which are decision making, user experience and user interactions. The systematic literature view framework that we used discovered some articles which dealt with evaluation methods which assessed either one or two of the user-centric factors not all of them. The lack of focus on the user-centric factors result in design of inferior RSs, which does not necessarily solve the existing challenges.

When evaluating the efficiency of recommender algorithms, many methods dwell on predictive accuracy neglecting user satisfaction. From the general discussions in the literature it is quite clear that there are mainly two classes of evaluation metrics, notably: system-centric and user-centric evaluation metrics. System-centric metrics are dedicated to the performance of the recommender algorithm whilst user-centric methods measure user satisfaction [Lei14].

2.1.1 Classes of Evaluation Methods

Evaluation methods are classified into two major classes namely user-centric and system-centric as shown in Figure 2.

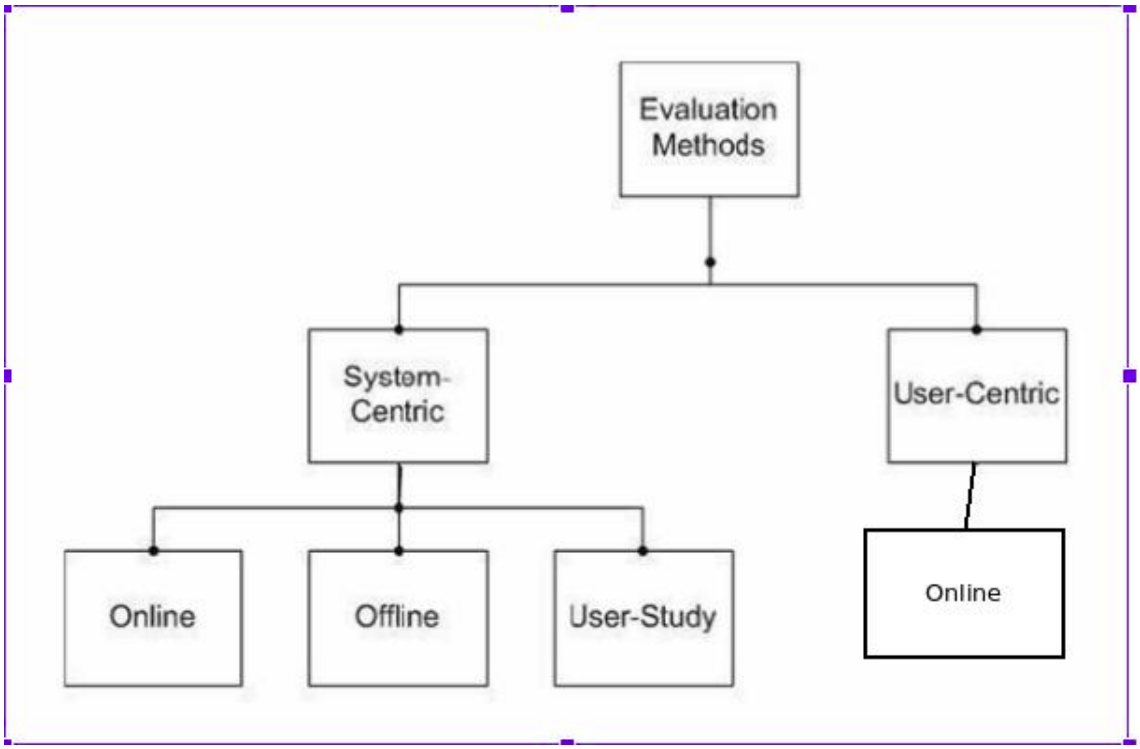

Figure 2: Major Classes of Evaluation Methods

2.1.1.1 System-Centric Evaluation Methods

System-centric evaluation methods mostly refer to an evaluation of a recommender algorithm against a pre-collected dataset of preferences. Users mostly do not interact with the system, user moves, explicit or implicit would have been collected beforehand and stored in an offline dataset for testing [Cre+12], [Her+04], [Lei14]. Researchers have argued that evaluating recommender algorithms has long focused on system-centric methods [Her+04], [Kni+12], [KR12]. There are three subclasses of system-centric evaluation namely: Online, Offline and User-study methods as shown in Figure 2. Online evaluation under system-centric evaluation mainly assess the robustness, precision, utility, coverage and scalability of the RS. Offline evaluation is mainly concerned with the accuracy and the performance of the algorithm when tested with an external data set and it is not necessarily concerned with the performance of the algorithm when users are interacting with it. User-study is the evaluation of how users feel about the performance of

Table 2.1: Classifying recommendations [SG11]

	Recommended	Not recommended
Selected/Clicked	True-Positive (tp)	False-Negative (fn)
Not selected/clicked	False-Positive (fp)	True-Negative (tn)

the RS. From our investigation we found that online and user-study evaluation methods has some elements of user-centric methods but they lack some fundamental elements or features which we explained below. All these three methods apply differently the metrics below.

Prediction Accuracy

This evaluation method focuses on how the algorithm missed the actual rating that it was predicting. This is by far the most common evaluation method. Under prediction accuracy Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) are the popular methods used to evaluate the accuracy of algorithms [Mub13]. There is also ranking prediction precision which looks at how many recommendations were selected in binary ratings, that is, either the item was selected (1) or not (0) [Mub13]. Ranking prediction accuracy can be calculated using two techniques: precision and recall using the items in Table 2.1.

True-Positive (tp): refers to the recommended items which were selected/clicked by the user.

False-Positive (fp): refers to the recommended items which were not selected/clicked by the user.

False-Negative (fn): Refers to non-recommended items which were selected/clicked by the user.

True-Negative (tn): refers to non-recommended items which were not selected/clicked by the user.

Precision measures the number of items in the recommended list selected as compared to the total number of items in the recommended list, it is also referred to as the hit rate.

$$\text{Precision} = \frac{tp}{(tp + fp)} \tag{2.1}$$

Recall measures how many items where selected by the user in relation to the total number of items available in the system (Vignesh and Kant, 2012).

$$\mathrm{Recall}(\mathrm{TruePositiveRate}) = \frac{tp}{(tp + fn)} \tag{2.2}$$

Precision and recall are inversely related and different user tasks require different trade-offs between them. [Eks+11]

Coverage

The term coverage embraces properties such as item space coverage and user space coverage. Item space is the ratio of items that the RS can pick for recommendations, whereas user space coverage is the ratio of users for which the RS can recommend items [SG11], [Lei14], [KM09], [KJN11].

Utility

Utility is a degree of use of the RS by users, so RSs can be compared on how they are being used by users. If the utility A exceeds the utility for B, then A is better than B [SMG09], [SG11]. It is mainly the value that is gained by the RS from each recommendation [SG11].

Robustness

This is the performance of the algorithm when given fake data [SG11]. Therefore during evaluation the system can be bombarded with fake data and check its response. Resistant algorithms are considered superior.

Scalability

On this metric we will be testing the performance of the RS by increasing number of users and the size of the data set. Many RS are seriously affected with this change and most of the times the algorithm need to sit on a powerful server with relevant computational power [SG11].

2.1.1.2 User-Centric Evaluation Methods

Refers to users interacting with a running recommender algorithm (what we call online evaluation) and receiving recommendations. Measurements are collected by asking the user (e.g., through interviews or surveys) or observing their behavior in real time, or analyze server event logs [Cre+12], [Lei14]. However the measures are comprehensive and the user's response to questionnaires does not resemble exactly what the user did during his/her interaction with the recommender algorithm. Evaluation which involves asking the user and tracking his/her interaction paths is the only extensive method of measuring users' satisfaction and quality decision making processes [Cre+12]. The main challenge is that conducting empirical tests involving real users is difficult, expensive, and resource demanding. Thus, user-centricity has been ignored and this has contributed to limited research in important user-centric factors such as user experience, user interactions and decision making processes [Lei14]. The following are some of the user-centric evaluation metrics which have been applied in literature.

Novelty

Novelty evaluates the capability of a recommender algorithm to recommend non-obvious items to a user [Her+04]. If a particular recommendation approach produces a recommendation list of size N and if R is the number of items in this list which are relevant to the user and do not appear in the popular items then the system is novel.

$$\text{Novelty} = \frac{R}{N} \tag{2.3}$$

Serendipity

Serendipity has an element of surprise it measures of how surprising is the recommended items to the user [SG11]. Serendipity can be detected for instance if a user is browsing a recommendation list of movies then he/she unexpectedly finds a movie of an actor that she likes. Users can be asked to identify if the recommender is really providing serendipitous recommendations using interviews or questionnaires.

Adaptability

RS operate in a setting where trends of user interests changes in time. Adaptation in RS measures how the RS adapt to users' personal tastes [SG11]. User's needs always change with time, and some changes of preferences are driven by changes in interface.

2.1.1.3 Framework for user-centric evaluation

Knijnenburg created a user-centric evaluation [Kni+12]. The framework uses strong psychological models and gathers information from users using questionnaires. It also evaluates online user-interaction. However we found that it didn't evaluate the entire factors of user-centric evaluation. Moreover we also found that questionnaires brings a challenge since some of the issues that affect user needs and preferences are not explainable in words by users. Therefore if the evaluation framework depends wholly on answers rendered by users the recommender algorithm will have a biased, incomplete, and sometimes incorrect understanding of users [SG11].
Böhmer et al came up with a usage centric evaluation framework called AppFunnel [Böh+13]. They considered a new approach for evaluating app recommender algorithms since evaluations need to go beyond ratings, click-through-rates or download statistics. The framework was mainly focusing on user-centric issues during the life span of applications. They tracked interactivity of a downloaded application from the time that it is downloaded, installed up to usage.

Pu came up with a comprehensive framework using popular usability models [PC10]. This framework was used to evaluate the perceived qualities of recommender algorithms. The framework was made of 13 constructs and a total of 60 question items that assess qualities of recommenders. The qualities involve usability, usefulness, interaction, satisfaction. The major drawback of this framework is similar to that of [Kni+12] were some of the issues that affect user needs and preferences are not explainable in a questionnaire.

2.1.2 Discussion on Evaluation Methods

Recommender algorithms evaluation has generally been associated with predictive accuracy, when an algorithm scores high in terms of accuracy it is regarded as as a good. However, this is no longer proving to be valuable because of the need to recommend what the

user needs [Sai+13]. Attention to user needs makes evaluating recommender algorithms quite difficult because users are quite diverse [CD10]. Much work has been done towards implementing recommender algorithms on different platforms [KA12], [Pog+13], [KC14], and [VP11]. A good number of researchers have come up with many evaluation metrics for recommender algorithms and each one of them was looking at different perspectives. In this section we discussed the different types of evaluation methods. We found that many of these are system-centric and, in practice, this affects revenue since user experience is the key to success or failure of recommender algorithms [Lei14]. In spite of knowing the importance of user-centric evaluations, these have not been employed within automated evaluation approaches as compared to system-centric metrics. The main problem with recommenders' evaluation is that current systems are limited to offer a comprehensive computational real-time user-centric evaluation and system-centric methods do not fully support user-centric methods.

Existing evaluation methods are aligned to either user-centric or system-centric evaluation methods. As the literature stands, no method can be considered comprehensive(encompassing both system-centric and user-centric) in its approach. User-centric evaluation frameworks consider methods such as use of questionnaires and psychometric surveys as their way of gathering information from users. System-centric evaluation methods use mathematical evaluation metrics such as prediction accuracy, robustness, scalability, utility, and coverage. These do not address users' concerns as they decide on accepting a recommender algorithm. We therefore found out that a holistic or comprehensive evaluation framework that combines user-centric and system-centric can be developed which must be able to use computational methods that make use of machine learning to assess both user-centric and system-centric factors, while a user is browsing. This brings a lot of advantages since it opens the possibility to assess prediction accuracy of the recommender algorithm as well as user satisfaction in real time.

2.2 Incorporation of Contextual User-specific Knowledge in Recommendation Algorithms

User's preferences or behavior can be influenced by the long term context or short term context of the user(the surrounding atmosphere). The context directly involves the user's way of life and its evolution, it also includes (financial status and background, residential places, time of the day, personal upcoming events, mood of the day) [Agg16]. Preferences of a user are heavily affected by these factors, for example the type of food a user can buy during the morning is different when it is in the evening or during lunch time. This contextual knowledge demands attention when constructing a recommender algorithm which can at least satisfy a user. Contextual knowledge is most of the time collected using implicit means, such as data mining and inferences. This section investigates the principle of incorporating user-specific contextual knowledge in recommender algorithms in order to satisfy the user. The following subsections discuss how this principle was implemented in related work.

2.2.1 Location as a Context

Recommending items to users based on their immediate location has becoming very prominent recently [Agg16]. This kind of recommendation is normally found in tourism recommender algorithms, whereby they incorporate a user's GPS location to recommend nearby food cafes, resort places, hotels and motels and other tourist attraction areas. There are algorithms which are dedicated to location based knowledge to recommend items to the user if the device he/she is using is GPS enabled.

Geographical location plays a role on personal or social preferences for example: type of food, clothing, choice of friends, recreational activities, choice of hotels and so on [Lev+12]. Research done by [Lev+12] on MovieLens proved that user ratings are inherently associated with location. Ratings entered by other users in the same location can be used in order to provide recommendations for a specific user in the same general location [AT08]. One of the main problems that often arises on location-dedicated recommenders is when the location in question has very few people in the system to be considered for

recommendations especially in CARS, it will be challenging to compute user neighbourhood [Agg16], therefore location must not be the only factor considered.

2.2.2 Time as a Context

Time can be incorporated in CARS since recommendations are timely in nature, and if a RS neglect time, it is most likely to offer irrelevant recommendations [Agg16]. A RS integrated on a restaurant website has to recommend a meal based on the time of the day. A RS working at a clothing website has to consider the time of the year, recommendations for summer wear should be different as compared with winter wear. Time can be classified differently, it can be put into categories of the day, year and seasons. It can also be classified into personal and national calendar events. Recently many algorithms that incorporate time as a context has been implemented [Kor09], however an algorithm that can incorporate all of the contextual knowledge has not been found yet.

2.2.3 Social Knowledge Incorporation

The social context is often important when recommending items to users [Kar13]. The choice of a user's friends can influence the user's preference. A person might choose to watch a different movie depending on his/her social context. Social issues also incorporate if the user is in either of these common social categories Upper, middle or lower income class. A user in upper class has a lifestyle that is altogether different from a user that is in the lower class. Social recommendations have been implemented that incorporate social knowledge in their recommendation strategies. Many of the recommender algorithms have been designed from the social network point of view. The notion used was if two users are friends on a certain social network, they are likely to like a certain product, this might be true to some extent if these users know each other, but if somebody has thousands of friends on social network that concept is likely to fail.

2.2.4 Ratings

Ratings in recommender algorithms is considered one of the primary factors that can display a user's taste, although recommender algorithms that use ratings face a serious challenge of data sparsity. It is sometimes difficult to get full knowledge about a user

through the expression of a rating. Sometimes the users do not rate anything or the ratings that can be used to compute recommendations might be irrelevant due to the time they were recorded. Nevertheless we found ratings to be a widely used factor incorporated by recommender algorithms to compute recommendations.

2.3 Novel and Serendipitous Recommendations

Recommended items are quite helpful to the user if they have not been seen in the past (novelty) [Agg16]. Unexpected recommendations are encouraged and this is serendipity. The difference between serendipity and novelty is that serendipity focuses on surprising users were novelty is about recommending unpopular items but relevant [Agg16]. Serendipity can cause a deviation in the user browsing patterns or changing user preferences just because the user has encountered something surprisingly but interesting. A serendipitous algorithm has long-term and strategic benefits in e-commerce for example because a user has the possibility of discovering entirely new areas of interest. Serendipitous recommendations bring an increase of customer conversion rate. Therefore we dedicate this section to investigate what has been done currently to curb the lack of novelty and serendipity in recommender algorithms.

2.3.1 Novelty

Many investigations done by [Var15] exposed a gap in terms of missing significant work done in novelty of RS. Saúl came up with a framework from different mathematical models to define novelty and diversity [Var15]. The framework was built on the notion that novelty and diversity are almost the same, however in some cases recommendations might be diverse yet not novel. Seen (discovered), chosen and relevance are the three essentials of the framework, when a user sees or discovers an item he/she can either choose or not that item based on its relevance. Therefore according to [VC13], the three are interrelated.

Novelty is then derived from the computation of this probabilistic model below:

$$p(\text{choose}) = p(\text{seen} \cap \text{rel}) \sim p(\text{seen})p(\text{rel}) \tag{2.4}$$

Therefore the probability that item i is novel given that it has been seen before is rep-

resented by the formula below:

$$p(i|seen) = \frac{p(seen|i)}{\Sigma_{j \in J}\, p(seen|j)} \tag{2.5}$$

Lee and Kyogu created a graph-based algorithm that provides novel recommendations. The algorithm picked positively rated items from the users' profiles to construct a highly-connected graph with items as nodes and positive correlations as edges [LH14]. The algorithm was able to offer relevant and novel recommendations.

Recent research seem to be have covered novelty significantly, however the average novelty computed in many algorithms still circulate around 77% [Zha13], [HZ11] and with the advent of CARS there is a room to improve the average novelty and serendipity of recommender algorithms.

2.3.2 Serendipity

Serendipity is finding something interesting by accident whilst not searching for it, it is a measure that indicates how the recommender algorithm can find unexpected and useful items for users [Sri14].

Upasna Bhandari bring a concept of serendipitous recommendations in mobile applications using a graph-based approach [Bha+13]. They considered the idea that if two apps are connected by a path with highly weighted edges then these apps are also similar. Therefore if there exists a path connecting two apps and the weights on each edge are all sufficiently large, apps along this path can be recommended to a user and are likely to be serendipitous. However, with this method, the recommender algorithm would experience a very low accuracy rate, although it came up with better diversity. This algorithm is also hard-wired to mobile-apps recommendations. Therefore it may be difficult to apply it to other domains that can be serviced by a recommender algorithm.

It is noticeable in the literature that all of the algorithms that have been designed for serendipitous recommendations use collaborative filtering on ratings. However, we have

not come across a robust serendipitous recommendation algorithm implemented in contextual knowledge-based recommender algorithms.

2.4 Change of User's Preferences and Satisfaction

Recommendations can improve user satisfaction on a website offering a service [Agg16]. If a user gets what he/she wanted without a hassle then the user is satisfied. It is more advantageous to cope with the user's satisfaction and also to divert the user's attention to some items. This is important in business to satisfy the user and make the business thrive as well. User preferences are sometimes determined from conversational systems [Agg16], where a user is asked questions then responds. It is also known that user context corresponds to user preferences [Kor09]. Meaning a user context such as current location or lifestyle influence to a greater extent the user's preferences. It is of paramount importance that a recommender algorithm can use this knowledge to compute recommendations.

Acquiring knowledge about users is a significant problem in RS [Var15], since personalized recommendations requires knowledge about the user. Some collaborative filtering approaches use ratings only but this approach can not be able to supply enough knowledge about users and their tastes. Implicit methods of collecting data gives great uncertainty of determining the actual user-preferences as well. Therefore, recommender algorithms work with incomplete knowledge about the user [Var15]. This brings more problems since user preferences are: not consistent, context-dependent and highly fluctuating [VC13]. User preferences change with time [Li14]. For example, the user's interest may be dominated by the recent events or life experiences and all these facts must be looked into when deriving recommendations.

2.5 Summary

From the analysis that we have done in this section, it is clear that user preferences changes, and they are diverse and complex to understand when taken all at once. However, recommender algorithms must be designed in such a way that they satisfy users. Therefore a modelling approach to incorporate contextual user-centric knowledge in recommender algorithms is imperative. From the discussion made in these sections: evaluation methods, contextual knowledge incorporation, novelty and serendipity, user's preferences and satisfaction we have been able to see that there is a recurring effort to sustain users' satisfaction. However there is also a recurring inadequacy of offering sustainable novel and serendipitous recommendations due to 1) methods that are being used to evaluate recommender algorithms 2) inadequate user contextual knowledge incorporation in algorithms. This insufficiency results in a limitation of understanding user's preferences thereby resulting in failure to deliver sustainable user satisfaction.

CHAPTER 3

Methodology

This chapter explores the methodology that was used to solve the research problem as well as to fulfill the given objectives. The methodology applied is a hybrid of many research methods with action, ethnography, systematic literature review and data mining being dominant research methods. Action research mainly focuses on attacking direct and specific problems in a local environment. This approach aims to intervene in the studied circumstances by improving the situation. Ethnography's goal is to study a community of people to understand how the members make sense of the social interactions. It allows for computer science researchers to have insight into human, social and organizational aspects of technology innovations and applications. Figure 3 demonstrates how these two methods were employed in this research project.

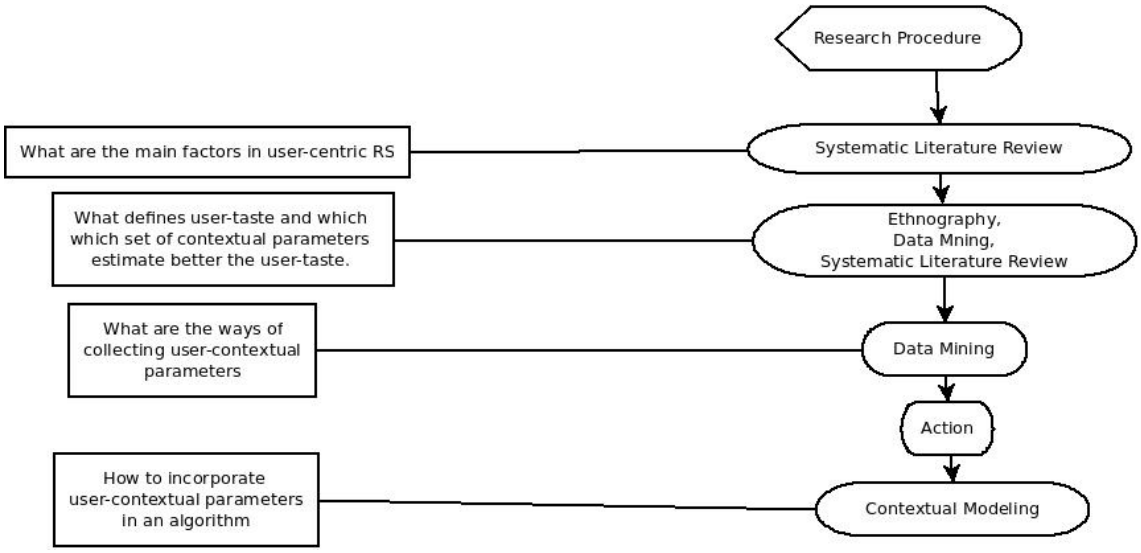

Figure 3: Research Methodology

In chapter one, there is a section of research objectives. Those research objectives were

tackled with appropriate research methods in Figure 3. Table 1.1 shows the research objectives and the corresponding research methods which were used to fulfill those objectives.

Figure 3 demonstrates the procedure that was taken in this research. In Figure 3 the first question is "what are the main factors in user-centric recommender systems". We found that a systematic literature review approach could answer that question.

3.1 Research Methods for Objective 1

An investigation of existing user-centric evaluation methods was done using a conceptual framework that applied that applied the systematic literature review approach. The main intention was to investigate inefficiencies in RS evaluation methods mainly targeting those aligned or related to user-centric factors. More so the strategy here was to investigate current evaluation methods/metrics for recommender systems in a bid to find out, why do current existing problems exist if recommender systems were rigorously evaluated. Moreover to find out what are the main factors in user-centric RS which can be employed in RS design to curb existing challenges. A User-centric Evaluation Conceptual framework(UEC) was designed to investigate current RS evaluation methods and expose loop holes. A systematic integrative literature review approach was applied in the framework, coupled with a thematic results analysis method to encompass all concepts regarding evaluation of recommender systems. Papers published from 2009 to 2018 were collected and those considered were those published in the peer-reviewed literature.

Search terms were derived from the questions and articles were gathered from these platforms: ACM, IEEE, Scopus, Science direct, MDPI, Google Scholar and dblp. The articles found were then characterised by the conceptual framework represented on Figure 4. The characterisation process determine the extent to which these articles contribute to the three fundamental principles or factors of user-centric evaluations (i.e. user interaction, decision making processes and user overall experience).

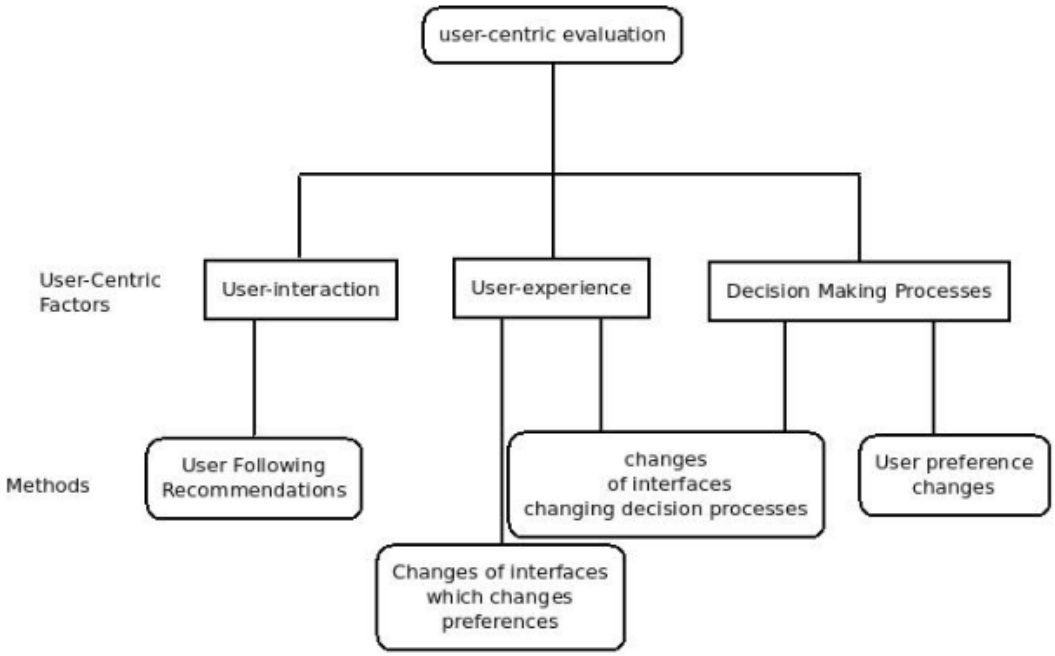

Figure 4: The User-centric Evaluation Conceptual Framework (UECF) for evaluation

3.2 Research Methods for Objective 2

From the investigation in objective 1, we identified that there was a lack of a better approach of representing a user profile which can then be used to derive optimal recommendations. We then ventured into context-aware recommendation systems (CARS) algorithms since there is a growing interest in CARS due to their ability to enable novelty, dynamism and serendipity in resulting recommendations. Firstly a systematic literature survey that aligns to some known frameworks like Kitchenham procedures was conducted, articles for CARS were collected from these sources :Scopus, ACM, IEEE, Science direct, MDPI, Google Scholar and dblp published from 2014 to 2019. A detailed investigation was done on the articles found, to find the main components of user contextual data. A characterisation process was done to define a framework that will define a user's context. The characterisation brought about a comprehensive set of contextual parameters into a framework called the SCuPE framework. SCuPE is an acronym for Social, Cultural, Psychological and Economic set of contextual parameters. It is the SCuPE framework that was used to predict user preferences and generate recommendations.

3.3 Research Methods for Objective 3

To identify the optimal set of user-contexts which estimate better user tastes three methods were employed. Systematic literature review method was used to find out the most dominant user-contexts used in literature and then characterise them. Articles for RS were collected from Scopus, ACM, IEEE, Science direct, MDPI, Google Scholar and dblp published from 2014 up to 2019. The contexts found from the articles were characterised based on the most superior classes of user-contexts. The second method used was ethnography which was used to collect characterised user information about user-contexts and their corresponding tastes. Using ethnography data was collected from voluntary participants from the ages of 18 to 55. The process was done in two stages and two data sets were collected from different people in Africa, each data set had 21 classified user-contexts and corresponding likes. Data mining was then used in implementing an apriori algorithm to identify he optimal set of user-contexts from a data set of user-context profiles with associated likes/tastes. Rules were generated and contexts which were found in rules with associated high support ratio and confidence values were extracted. Python libraries were employed to create graphs showing the relationship between extracted contexts and support ratio as well as confidence values. The optimal set of user-contexts were identified, and they can be used by RS developers in order to enhance recommendations.

3.4 Research Methods for Objective 4

The question that was also addressed in this section was, how could a recommender algorithm have access to a user content. This section expands on the basic methods that was used in this research to access user-content.

Proper user contextual knowledge can be found from two sources: either psychometric surveys or social media. Chao Yang demonstrates how psychometric surveys can give almost the same result as social media analysis when predicting users' preferences on different brands based on their personal traits [Yan+15]. Their research suggests that, using somebody's personal traits gathered from a psychometric survey, together with preferences of certain brands, it can be concluded that people with certain personal traits prefer

certain brands. It has become clear from contemporary research that people ignore surveys or they do not have time to answer surveys. In light of this development, social media has come as a rescuer since we can gather massive unstructured data, process it and get useful insights about individuals.

Chao Yang and his team gathered specific users' personal traits from both psychometric surveys and tweets from Twitter. The psychometric survey asked users to state their personal values and individual needs then pick their favourite brand. They found that after mining the same individual's account from twitter and analysing their sentiments they were able to predict the individual's brand preferences with an 86% accuracy rate. The main advantage of this research is that they proved it is possible to use personal traits gathered from social media to predict users' likes in different domains.

Social media through APIs give access to applications to get some information about their users, if the user gives permission to access information to an application. This information is rich if used by a recommender algorithm, and this is the approach which this research took. We have classified this information as general profile and activity as shown in Table 3.1. The general profile is somehow static and it does not change regularly and is not normally affected by time and events, moreover this category does not give more information about the movement of the user such as, change of tastes or general activities that the specific user do. The activity category is rich of user's whereabouts; change of location(for example a user can tag himself/herself on a particular location, or can notify friends on social media that he/she is flying to a certain location), soccer team support, movie likes, friends list update, currency etc. The activity class supplies contemporary knowledge about a user's context.

The information in the Table 3.1 can be accessed in the form of variables from social media.

The techniques and methods used to design the Holistic User Context-aware (HUC)

Table 3.1: Social Media Content

General Profile	Activity
1. user-work-history	1. interested-in
2. age-range	2. friends-religion-politics
3. gender	3. Friends-work-history
4. education	4. Relationship-status
5. political	5. Friends-relationship-details
6. Religion	6. User-likes
7. User-education-history	7. User-about-me
8. user-hometown	8. user-location
9. friends-hometown	9. friends-location
10. Languages	10. locale
11. quotes	11. Timezone
12. user-birthday	12. favorite-athletes
13. Friends-birthday	13. favorite-teams
	14. currency

algorithm include: contextual modelling, unsupervised learning, data mining, collaborative filtering and content based filtering. Social media content was used to generate recommendations. The algorithm that was created recommends friends or pals on Unipals. The SCuPE framework defined the comprehensive user-contextual parameters which would be built gradually when users interact with the social network. Some of the user-contextual parameters which define a user profile on Unipals and used in HUC was geolocation, nationality, time, friends(social relations), age, sex, picture-tags class, language, likes, hobbies and any other interests. The main task of the HUC algorithm on Unipals was to recommends friends to users.

3.4.1 Action Research Method

The method was used to create user models and the recommender algorithm. Unsupervised learning, data mining, contextual modelling, collaborative filtering and content

based filtering were the main techniques and methods that were applied in this research method.

Data Mining

Data mining techniques are applied to collect content on social media platforms. This collection of user content is mainly done through APIs, however in this research project the algorithm was directly integrated to the social network platform therefore data was collected from the logs and the database. Data from the logs was pre-processed to visualize how the users interacted with the platform. The database provided the user content which was used to derive the user-contexts.

Content-based Filtering

Content-based filtering was applied after data mining. When the content was collected it went through cleaning processing so that similarity computations could be done. Content-based filtering mainly focuses on creating recommendations based on the similarity of descriptive attributes of items [Agg16]. In this research the content was the user data that was extracted during data mining. After user-context was accessed, the contexts were filtered using the Jaccard similarity algorithm.

Unsupervised Learning and Collaborative Filtering

Unsupervised Learning is a machine learning technique that learns patterns from unlabelled data. In this research this approach was used to create clusters(neighbourhoods) of users with similar context, this approach works hand in glove with collaborative filtering. Contextual similarity was computed by the Jaccard similarity function. When a user logs in, a similarity computation is done so that the user is assigned to a neighbourhood or cluster, and the recommendations that the user will get will be derived from the neighbourhood by other processes. The Jaccard similarity metric, defined as the function J, was used to calculate the similarity between people's profiles or contexts. That is, the similarity between two people's profiles p1, p2 is the Jaccard metric between their two profiles(P)

$$J(p1, p2) = \frac{(|P(p1)INTERSECTIONP(p2)|}{|P(p1)UNIONP(p2)|}$$

Using Jaccard coefficient to Calculate Similarities between $User_i$ and $User_n$ profiles,

$$J(u_i, u_n) = \frac{S_i \cap S_n}{S_i \cup S_n} + \frac{C_i \cap C_n}{C_i \cup C_n} + \frac{P_i \cap P_n}{P_i \cup P_n} + \frac{E_i \cap E_n}{E_i \cup E_n}$$

Where S,C,P and E are the classified contexts from the SCuPE framework.

The techniques discussed so far did not stand alone. For these techniques to work efficiently there are supposed to employ some methods to incorporate contexts in the algorithm, and this can be done either through pre-filtering, contextual modeling or contextual post filtering [Har+17].

3.4.1.1 Contextual Modelling

Contextual modeling is the incorporation of contextual information directly in the recommendation model [Sun+16], [IL16]. In this research we use the contextual data to compute the neighbourhood of users as well as to compute recommendations. The Jaccard similarity algorithm was used to compute the similarity of user contexts. If there are many active users probably hundreds of thousands it will be resource intensive to find the Jaccard similarity among the logged in user with all active users. Therefore to solve this challenge the ID3 decision tree algorithm was used to classify profiles/contexts. Since user profiles comes in as sets, it is logical to classify user profiles. When a new profile comes in, the ID3 algorithm would put the profile in a specific class, a process of computing similarity would be done in each class to find profiles that are closely similar to the new profile, these similar profiles would then create neighbourhoods. We discovered during experiments that this approach reduces running time significantly.

3.5 Research Methods for Objective 5

Comparing the performance of two algorithms using the same data set has been proved difficult since RS has different architectures [Sai+13]. Therefore, RS evaluation demands a comprehensive approach so that at the end of the day the process of evaluation is fair to the RS. Offline evaluation is important, but user satisfaction evaluation is even more

important for the success of the RS especially when it is done online [Hay+02]. In online evaluation, the environment is more real, and users are not aware that they are evaluating a RS. Researchers have found that this technique reduce any bias when selecting users to test an RS [Agg16]. In this research project online evaluation was the major technique applied to evaluate the performance of the HUC algorithm. The algorithm was hosted on the Unipals platform and students interacted with the algorithm as they clicked recommendations which were recommended by the algorithm. The algorithm recommended friends therefore users were treated as items in this case. The user interaction was recorded in a .csv file. The online evaluation process also include computing some metrics such as: coverage, robustness, accuracy, novelty, and serendipity. User satisfaction was associated with the dynamism, novelty and serendipity of the algorithm. The time complexity was calculated and found that the average time complexity of the algorithm is $O(n^2)$.

3.6 Summary

This chapter highlights the research methods which were applied to fulfill each objective. The methods include: data mining, action research method and systematic literature review. It is also worthy to note that we applied the methods together, there was hardly a time where one method was solely used to address an objective. The researcher believed that these were the best methods and they were applied adequately.

CHAPTER 4

Incorporation of Comprehensive User Contextual Parameters in Recommendation Algorithms

4.1 Introduction

This chapter investigates the theory of user preferences and user contexts. Its main objective is to investigate the theory of user contexts in the design of a context-aware recommender algorithm. The objective covers the use of a framework for evaluating recommender systems evaluation methods and modelling user preferences. The theoretical conceptual framework proved to be a valuable tool for finding out what has been done, and what is missing. This framework also shows the status quo and highlights areas of exploration that can enhance the status quo in this domain of research. modelling user preferences helps this research to explore the theory behind user-preferences and user-contexts and how recommender algorithms can harness this theory into computational models. A concise understanding of the dynamics of user-preferences provides a platform for implementing robust algorithms that offer dynamic, novel and serendipitous recommendations. Thus, there is continuity of users' satisfaction within online businesses [Lei14].

The overall research problem states that there still exists the problem of inadequate user contexts in CARS bringing about limited representation of user tastes [Ung+16], [Har+17]. This problem is resulting in recommender algorithms losing relevance [LH14]

and efficiency with time [JWK14], [Var15], [Hid+16]. Failure to investigate this problem results in recommender algorithms recommending irrelevant items to users with time [Agh18]. In this chapter we started by investigating the RS evaluation methods. This investigation revealed with greater detail the actual problems in RS or CARS. The investigation was so helpful that it helped in terms of bringing out a concise research problem. This investigation was carried out using a conceptual framework.

The conceptual framework reveals the challenges with current methods being used to incorporate user-contexts in CARS. The chapter then elaborate the best theoretical means of solving these challenges using the theory of user-preference. The theory of user preference then leads to user preference modelling.

4.2 User-centric Evaluation Methods

Many evaluation methods for recommender systems have been implemented before [Lei14], [Lu+12], [Kni+12], [Kav+16], however algorithms which were identified as optimal according to the evaluation methods are facing challenges when measured against evolving user's expectations [Lei14], [Cre+12], [KR12], [Kni+12]. It is therefore necessary to come up with a conceptual framework that investigates the current evaluation methods and find out their strengths and loopholes that are causing recurrent problems. This conceptual framework also exposes areas which need attention and how these challenges can be addressed. The results of the investigation lead to new methods of modelling user-preferences so that algorithms can cope up with users' dynamic preferences and expectations.

Recommender algorithms are implemented to help users to get their services in time, because of serious information overload, especially on the web [Var15]. It has become a general norm that recommender algorithms should help users get to their services on time as well as try to convert visitors to clients [JWK14], [Var15], [Hid+16]. Therefore, when evaluating a recommender algorithm it must be tested along the user-centric factors, because it was implemented specifically for the user [Kni+12], [Tin16], [KR12], [JJD17].

This particular section discusses the fact that existing evaluation methods have not done enough to evaluate recommender algorithms relative to users' perspectives or expectations [Kav+16].

4.2.1 Challenges and Significance of User-Centric Perspectives in Evaluating Recommendation Algorithms

Adequacy of user-centric evaluation methods has always been a challenge due to the lack of standardization [Sax+09], since standardization brings the holistic view of the evaluation processes. If an evaluation standard is created by the research community then all RS including CARS would have one uniform benchmark of evaluation thereby eliminating doubt in deployed algorithms. Up to now, the research community has not standardized user-centric evaluation metrics for recommender systems. As a result, expected key performance areas for recommender algorithms are not all identified and assessed.

It is quite important to evaluate RSs on the basis of their impact on user experience and how they help users to meet timely goals [Kon+06]. It is now widely acknowledged [Lei14], [Cre+12], [Kni+12] that it is the user-centric factors that determine if a recommender algorithm is going to be adopted or not. Therefore displaying recommendations is not enough, since there is a need to evaluate whether or not the needs of the user have been met [Her+04], [Kon+06]. Helping users and providing good value to users should be the purpose of recommender algorithms. Hence, user-centric factors deserve to be one of the key foci in recommender algorithm evaluation methods [KR12], [Eks+11].

4.2.2 Characterizing User-Centric Evaluation

Researchers have proposed different approaches to user-centric recommender algorithms evaluation; some are categorized as methods or frameworks [Kni+12], [CD10], [Böh+13], [PC10]. This section introduces a characterization of user-centric evaluation methods. This is done to have a detailed investigation of existing evaluation methods. The characterization involves an investigation of the adequacy of existing evaluation methods in evaluating user-centric factors in recommender systems. It also presents that the main

purpose of recommender algorithms is to enhance user experience, interaction and decision making processes.

Existing RS evaluation has been focusing on system-centric factors, mainly statistical measures which include accuracy metrics ([Her+04], [MRK06], [Cre+12], [KWK11], [BS12], [Kni+12], [PC10]). It is clear in some early work that user-centric factors were already recognized as important ([SM95], [SS01]), however the challenges associated with user-centric evaluation [Nan13], [Eks+11], [KWK11], [BS12], [KR12], [Cre+12] bring about little commitment to research in user-centric factors as compared to system-centric evaluation [Cre+12]. Looking at progressive research work in RS evaluation, it is a fact that user-centric factors are widely accepted and their crucial role known in the RS research community [Her+04], [MRK06], [Eks+11], [Kni+12], [PC10].

From the analysis that has been found from literature, it is clear that user experience, user interaction and user decision making are the core targets of recommendation engines. When consumers are browsing e-commerce sites shop owners take note of consumers interests, overall experience and assess customer's moves before getting to a buying decision. A buying decision can be represented as an ultimate favored act on e-commerce sites. A user ultimately reading an online news article on a news platform is also engaging in the desired act on the news platform, meaning a recommender algorithm on such a platform will be recommending articles so that the desired act will be realised. Figure 5 presents the main characteristics of user-centricity. The figure shows that the user's interaction with an e-commerce site or a product recommender, influences his/her buying decision, and this interaction can further add to the user's experience. The user experience contributes to the user retention rate, which either increases or decreases user interaction. Experience and interaction both influence the buying decision.

4.2.2.1 User Interaction

Interaction is the observable behaviour of the user [Kni+12]. It is quite important to consider the impact of interfaces, user interaction, decision-making processes and user experiences when evaluating recommender algorithms [Tin16]. The interaction must be

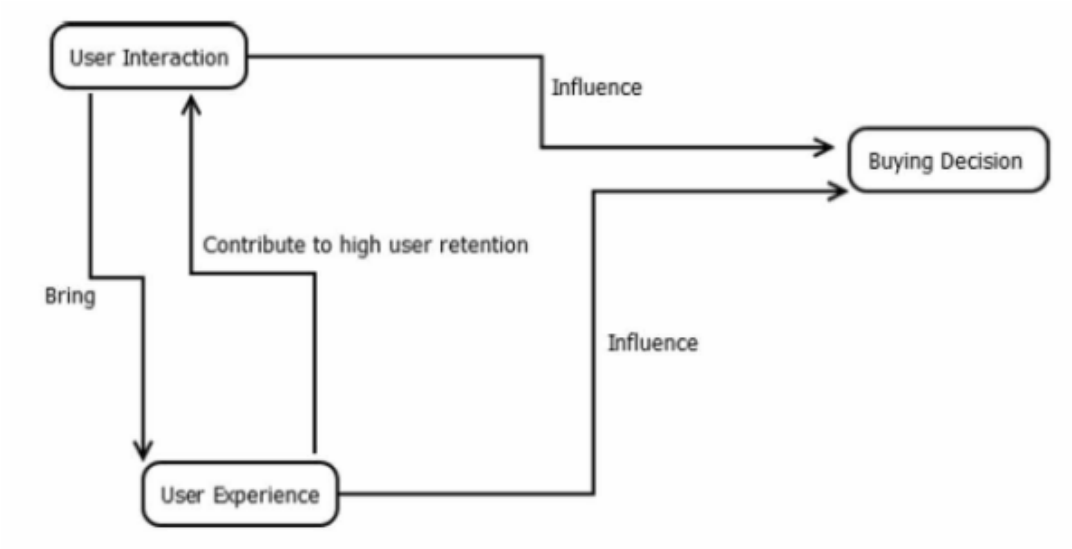

Figure 5: The core targets of recommendations.

easy, as this creates a form of trust with the system especially if the user feels to be in full control with privacy preserving mechanisms [PC10].

4.2.2.2 Decision Making Processes

Users' confidence to make decisions rely heavily on users' overall perception of a recommender algorithm and features like personalization support and purchase decisions [PC10]. Decision theories are focusing more on reducing cognitive effort rather than on improving the desired decision, which is the main issue in recommender algorithms [PC10]. Certain characteristics of RS such as interaction design and interface, cause changes in users' decision-making behaviours [Kni+12].

4.2.2.3 User Experience

User experience is that feeling which a user experience whilst interacting with a platform, it is either satisfying or not [Kni+12]. Usually it comes from the user's familiarity with a site, which results in high user retention, therefore in this project we assume that higher retention rate is strongly correlated with higher user experience. Experience is associated with choice satisfaction and the perceived effectiveness of the system [PC10]. Bohmer argue that there is a need to come up with a variety of techniques to evaluate user experience since they always evolve [Böh+13].

4.2.3 The User-centric Evaluation Conceptual (UEC) Framework

User-centric evaluations focus on how recommender algorithms consider user factors during recommendations. Such other factors include (mood, urgency, income and interest in a particular product). User-centric approaches have users interacting with running recommenders, while interactions are recorded and then analysed [Lei14], [Cre+12]. User-centric approaches are mostly sophisticated, difficult to comprehend and take much resources to be usable [Cre+12], [Kni+12], [Lei14], [BS12], [KR12]. From the investigation done in the previous sections we found out that the three main user-centric features when evaluating are: user interactions, decision-making processes and overall experience. These features can be evaluated to test if recommender engines support them. These three are exhibited in user browsing patterns behaviour collected in event logs or user preferences as collected via questionnaires and surveys. In order to measure these, five evaluation methods were derived and these methods fall under user-centric evaluation as shown on the UEC framework on Figure 4.

4.2.3.1 User following Recommendations

A user can either pick a recommendation or not. A recommender can influence a user to change his/her preferences by issuing serendipitous recommendations. Therefore if a recommendation can influence a change of user preference, that recommendation is to some extent user-centric. Evaluation methods need to assess if a RS influences change of preferences and track user's browsing patterns.

4.2.3.2 Changes of Interface which Affect User Preference

If a recommender system interface, or a site interface changes, user experiences changes since it is directly related to interface, and also user's browsing patterns are likely to change [AW12]. This change in browsing patterns can change user's preferences, because a user can meet other items just by diverting the path he/she was taking before. Therefore a recommendation engine should take note of interface changes since online platforms are always evolving.

4.2.3.3 Changes on Interface which Changes Decision Making Processes

If interfaces change, browsing patterns also change. This affects browsing patterns that lead to some decisions (such as buying decisions) thereby affecting revenue, since user experience influences buying decisions. Evaluation metrics should measure the impact of user interface changes on user decisions so that it becomes possible to evaluate the impact of interface on revenue.

4.2.3.4 Changes of Decision Making Processes

Users have patterns that they follow to reach a decision [Pog+13] and these patterns are subject to change due to preference changes influenced by different factors. This method should evaluate if a recommender engine considers deviations of user's patterns that lead to desired decisions during the testing process.

4.2.3.5 User Preference Changes

The idea is to predict user's needs as well as identify patterns of decision making processes. A recommender should know changes of a particular user's preferences, if a recommender algorithm is not aware of user preference changes it means it will provide irrelevant recommendations. Evaluation metrics should check if a recommender engine takes note of user preference changes during testing.

4.2.4 Application of UEC Framework: In search of User-centric Support in Existing Evaluation Methods

With reference to the structure of existing evaluation methods, it is quite compelling to investigate if user-centric and system-centric evaluation methods consider user interaction, decision making processes and user experience during evaluation. We found these giving enough justice to our method and results, so that at the end we have both perspectives for user-centric and system-centric evaluation methods. An integrative review approach was used, coupled with a thematic results analysis method to encompass all concepts regarding evaluation of recommender systems. Papers considered were those published in the peer-reviewed literature from 2009 to 2018.

Table 4.1: Number of articles found

Search Category	Components of the Category	Number of articles found(including duplicates)
User-centric evaluation	1. User-centric evaluation metrics 2. User-centric evaluation methods 3. User-centric evaluation 4. Frameworks 5. User-Factors evaluation	4
Evaluation	1. Evaluation metrics 2. Evaluation methods 3. Evaluation frameworks	11
System-centric evaluation	1. System-centric evaluation metrics 2. System-centric evaluation methods 3. System-centric evaluation frameworks	22

4.2.4.1 Results from the UEC Framework Application

Table 4.1 shows that system-centric evaluation of recommender systems search retains many articles as compared to any other search terms. User-centric evaluation search terms retain few numbers of articles, resembling that it is a growing research area. Many of the search terms were retaining articles from previous searches terms regardless of the source, thereby our articles add up to thirty eighty.

Figure 6 shows the evaluation metrics and their respective authors. From prediction accuracy up to scalability, results show that these metrics were covered in articles which focus on system centric evaluation. Results also show evaluation methods which were

not necessarily metrics since they are not standardized. Other evaluation methods were frameworks i.e. (combined methods and metrics). Moreover, we also found evaluation methods which were not standardized, most of these were aligned to evaluate the context of recommendations. These evaluations if given recommendations consider the context of users. Other methods to evaluate user satisfaction solely use questionnaires and surveys to gather information.

Results reveal thirty eight articles which we classified as i) user-centric ii) system-centric

Dimension(s)	Evaluation Category	Citation
Prediction Accuracy	Metric	(Mubarak & Imran, 2012; Shani & Gunawardana, 2011; Vignesh & Kant, 2012; Michael et al., 2011; Gunawardana & Shani, 2009)
Coverage	Metric	(Shani & Gunawardana, 2011; Leino, 2014; Su & Khoshgoftaar, 2009; Nasraousi et al., 2008)
Cold Start	Metric	(Shani & Gunawardana, 2011)
Confidence and Trust	Metric	(Shani & Gunawardana, 2011; Leino, 2014; Michael et al., 2011; Lin et al., 2013)
Novelty	Metric	(Herlocker et al., 2004; Chandrashekhar & Bhasker, 2011; Shani & Gunawardana, 2011)
Serendipity	Metric	(Shani & Gunawardana, 2011; Pasquale, Marcode, & Giovanni, 2011; Lops, 2014; Leino, 2014)
Diversity	Metric	(Shani & Gunawardana, 2011; Leino, 2014)
Utility	Metric	(Gunawardana & Shani, 2009; Shani & Gunawardana, 2011)
Adaptability	Metric	(Shani & Gunawardana, 2011)
Scalability	Metric	(Su & Khoshgoftaar, 2009; Shani & Gunawardana, 2011; Leino, 2014; Mibarak & Imran, 2012; Suneetha & Kumar, 2013; Michael et al., 2011)
Evaluation methods	Method	(Pessemier et al., 2015; Dabrowski & Acton, 2013)
Adaptability	Metric	(Shani & Gunawardana, 2011)
Scalability	Metric	(Su & Khoshgoftaar, 2009; Shani & Gunawardana, 2011; Leino, 2014; Mubarak & Imran, 2012; Suneetha & Kumar, 2013; Michael et al., 2011)
User following recommendations	Method	(Shani & Gunawardana, 2011)
Robustness	Metric	(Shani & Gunawardana, 2011; Gunawardana & Shani, 2009; Konstan & Riedl, 2012)
Evaluation frameworks	Framework	(Cioreclas, Irdeto Res., Doumen, 2010; Pu et al., 2012; Knijnenburg, Meesters, Marrow, & Bouwhuis, 2010; Funk et al., 2010; Surendren & Bhuvaneswari, 2014)
Other Metrics	Metrics	(Arana-Llames et al., 2014; Epifania & Cremonesi, 2012; Cremonesi et al., 2013; Pu et al., 2012; Said, Fields, Jain, & Albayrak, 2013)

Figure 6: Articles and their focus

iii) and both user and system-centric. Figure 7 below shows the total number of articles per category. We went further to determine the extent to which these articles contribute to the three fundamental principles of user-centric evaluations (i.e. user interaction, decision making processes and user overall experience). The graph shows that not much

has been done to explore the entire user-centric base, i.e. no work was found that evaluates recommender systems checking all the three fundamental factors of user-centricity. Different articles were found which cover either one or two but not all these three factors. The graph also shows that research needs to focus on evaluating recommender systems on how best they increase and sustain desired decisions or patterns that lead to decisions(such as buying decisions). **Note**: On Both we looked at articles that focus on both

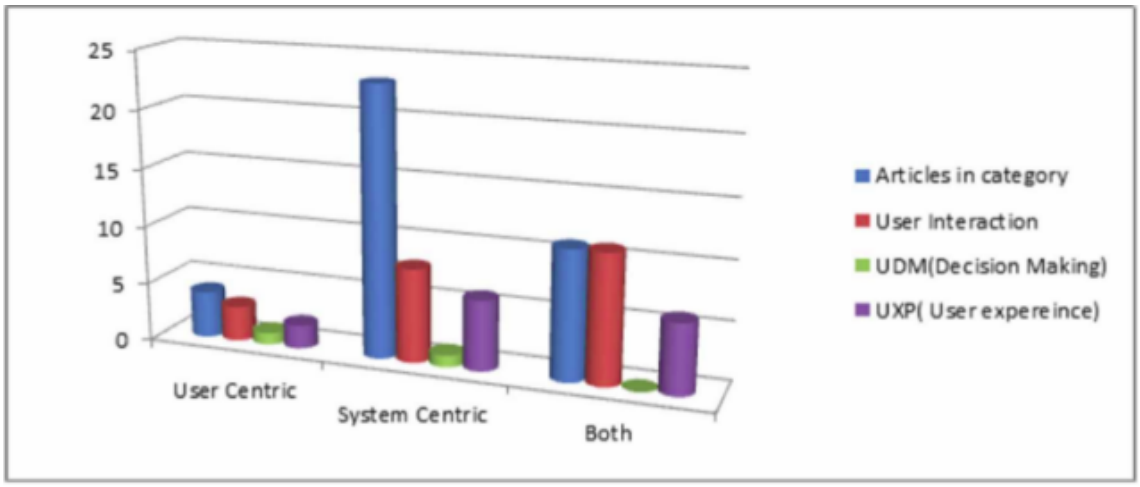

Figure 7: Articles found and their respective categories

system and user-centric evaluation. Therefore

$$UI + UXP \neq Totalnumberofarticles(thebluebar)$$

since an article can target both UI and UXP and others will only target one.

4.2.4.2 Evaluation and Analysis of Results

As suggested earlier, a user-centric evaluation method should be able to evaluate:

- User Interaction (UI): The user's interaction with the recommender system,
- User Decision-Making (UDM): Is the recommender helping a user to make a desired decision (or does the system enhance decision making processes) and
- User Experience (UXP): Is the overall user experience enhanced so that there is high user retention.

User Interactions

There is an assumption in recommendation systems that a user will always go through

a particular path when navigating an online platform [RV10]. However, this does not remain accurate with time since a user will take various approaches in their search for services, information, and products [Pog+13]. The are two trade offs in user interaction which is the accuracy of the algorithm and the user interface, if you focus on the accuracy the interface has to remain unchanged [Lei14]. Again if you care about the outlook of the interface, the back-end algorithm should remain unchanged as well [SMG09]. There is a need to develop algorithms that adapt automatically to user changes when the interface changes so that net profit is maintained.

Novelty is still a challenge to evaluate in existing systems [Var15]. This metric is regarded as important in RS. However, evaluating novelty using questionnaires and surveys poses a challenge since users do not always remember what they have seen or do in the past. Another challenge is that irrelevant recommendations may be considered new but worthless to the user. To detect relevant items for recommendations, it is important to know the exact items the user is looking for and this is still a challenge with the present state of the art [SG11]. Another user-interaction metric is serendipity. A means of getting information to measure serendipity has one challenge [Var15]: user feedback on what they did is rarely the same as what they actually did [Lei14]. Some consider the serendipity problem as a programming issue [LGS11]. Issues with serendipity has not yet found concrete solutions in RS [LGS11], [Lei14].

Decision Making Processes

A good recommender system should help a user make a decision (such as buying an item, finding a service, reading an article etc.) [Ras+12], [Agg16]. We discovered that there are still weaknesses with present evaluation methods when they test if recommender systems improve user decisions.

User Preference Changes

Preferences of users always change with time because of factors like dynamics on web structures driven by evolving user awareness, browser technologies, social factors, meth-

ods of communication, and operating system updates [Kar13]. Failure to manage preference changes causes loss of desired decisions on different systems [Kap+15]. The other challenge is that some data from user behaviors can be difficult to use to interpret user experience [Kni+12].

Changes of Patterns that Lead to Desired Decisions

Browsing patterns evolve so quickly that attempting to make predictions or guide consumers to make positive decisions, according to frequent patterns coming from the analysis of an access log, becomes challenging. Some work must be done to make recommendation processes adapt to frequent product changes, as well as customer needs and preference changes [Wac11]. There is lacking a method for evaluating if a recommender engine is aware of the number of potential clients that deviate from their earlier patterns that led to desired decisions due to change of preferences [Pog+13].

Interface Changes Influencing Change of Preferences

Much research is underway to investigate the impact of user context in RS, as well as exploring some factors that affect user decision-making processes and user experiences [Tin16]. Recommender interfaces have a big impact on decision support, satisfaction and overall user experience with the recommender system [Tin16], [Kni+12].

Changes of Interface Affecting Patterns that Lead to Desired Decisions

The research community has begun to see the importance of considering user-centric factors when dealing with success or failure of RS [Lei14], [AW12]. This is an area that has not been focused on by researchers. Yet, state of the art techniques can detect a change in net profit when there is an interface change [Lei14].

4.2.4.3 Implications of findings

Some major improvements in the present state of the state need to be done to satisfy users when interacting with recommender systems, as it is shown that prediction accuracy is not the major objective but user satisfaction [Lei14]. Recommender systems need to help users make better decisions, algorithms should serve the needs of users even at the ex-

pense of accuracy and precision [Lei14]. Other evaluation metrics such as site utility need a closer look since eliciting user perception of utility is a very difficult task because it is more psychological than computational. The reason why a user makes a certain decision is based not only on what is presented in a recommendation list but also on issues like income and purpose of the object to be picked [SMG09].

4.2.5 Conclusion

The main contributions of this section are: a detailed characterization of user-centric factors for evaluating recommender systems, a user-centric evaluation conceptual framework as well as it's application. The section also presents suggestions that can improve the status quo. This helps developers of recommendation engines to address the needs of the clients. The section highlights that there is a need to ask questions about satisfying users when they interact with recommenders. This entails that an evaluation needs to be done to assess if recommenders know in advance user's preference changes and deviations from their interests. The section concludes that there is a need to investigate user-centric evaluation methods that assess all the following three important factors: user decision making processes, user experience and user interaction.

4.3 User Preference Prediction

User modelling is basically done using two main approaches i.e content-based filtering and collaborative filtering [KWV16]. In these modelling approaches user information is gathered through implicit and explicit means. This section demonstrates the theory of user preferences, and how the understanding of this theory will be harnessed to create preference models that enable recommendations. The theory provides a deeper understanding of the user, so that novel and serendipitous recommendations becomes a reality.

4.3.1 The Theory of User Preferences

Kamor describe a model of user preferences in the form of three significant states: sensitization, boredom and recurrence [Kap+15]. They found that the dynamics of user preferences on items can be attributed to the dynamics in those states and people often choose products based on their current tastes. These tastes always evolve with time [FO17], caus-

ing the user to manually select different content [Kap+15]. When someone is experiencing a certain psychological situation or is in a certain mood that person can be interested or not interested in certain products or services [FO17] algorithms need to note these changes as well. Recommender algorithms seem to neglect such changes when recommending items to users, they solely use past user choices to infer their preferences [Kap+15]. Typical approaches are lacking the modelling of the evolution of preferences with time [Lu+15]. Human psychology studies reveal well known phenomena of boredom when a person is repeatedly exposed to the same stimuli [Kap+15]. Lack of modelling preference changes normally result in losing track of user boredom.

Karimi defines consumers as subjects that find information, evaluate it and make a choice, and the context of the choice is significant as well [Kar13]. A decision/choice made reveal the decision maker's values [Buc06], [FO17]. This can imply that if a user choose an item, another user who share the similar taste can pick it as well [Eir+18], [Che+18], [MZH18], [Yan+14]. This analogy can develop into an approach of developing recommendations algorithms. This analogy can be illustrated as follows: Given U_i who is a user with attributes a,b,c,d,.....z who is interested in $item_i$,and U_k, with the same attributes a,b,c,d,.....z, there is a greater chance U_k might be interested with $item_i$.

Understanding users/clients in RS is a fundamental concept that result in enhanced recommendations, if users are understood recommendations can be relevant and will be offered at the correct time. There are factors that affect decisions made or to be made by users such as financial status, psychological status (which is normally mood), culture and social background [FO17], [Zha+18], [Eir+18]. For a user to like an item, he/she is driven mostly by his/her context for example mood(psychological status). What does he/she want to use it for(cultural/social status), if it's a purchase does he have an experience with it, is it relevant (cultural/social status), does he have the money(economic status). Moreover does he enjoy having it(if it's an item to be purchased), reading(if it's a news article), all these are basically(economic, psychological , cultural and social) questions which are influencing this decision to be made by the user.

Constructing user profiles rely on implicit and explicit methods of gathering data. Implicit methods of gathering user data involve click-streams, purchases and selection of recommendations. Explicit methods rely on asking users questions, ratings, votings et.c. However for implicit methods purchasing a product does not fully mean liking a product. A user can buy a product out of ignorance and dislike it afterwards. Not clicking a product is ambiguous since that can mean dislike or did not see it. Actions are also very difficult to scale because of some ambiguity. Behavioral data can be a big challenge to interpret as well [Kni+12]. Explicit methods also have their challenges, ratings are unreliable and inaccurate to a reasonable degree since users are typically not interested and are reluctant to rate items [ERR16].

Ratings lack detailed information about the user's preferences, which can give recommenders confidence in their recommendations. Binary data as well is too rigid to express neutral feelings about any item. Such limited data does not provide more information about a user and that has implications to the recommendation strategies. Explicit data does not cope with users' preference changes since a recommender system can continue working with ratings which are no longer relevant to a particular user.

To the best of the researcher's knowledge of the challenges discussed above occur in recommender systems as a result of treating a user as a virtual black box. Yet recent research exposes the advantages of opening the black box in order to know the changes of user preferences [Kav+18], [Kar13]. Algorithms lack understanding of the holistic contextual parameters to employ during prediction in order to offer recommendations that satisfy users. As part of this work's investigation [KDR19], a holistic approach that considers user actions together with detailed information from the user can give better recommendations. In this way, recommendations would be more accurate and can cope with evolving preference changes.

4.3.2 User Contextual Knowledge

The contextual knowledge of a user is significantly important [FO17], [Zha+18], [Eir+18], since users look for certain items or services depending on their contexts, therefore recommender algorithms need to take context into consideration. Creating a recommendation model based on user contextual data is vital and that brings about the ability to present relevant recommendations to the user's current context [Li14]. The user's contextual data(features) might be time, location, social surrounding, psychological status et.c, and these are vital for novel and serendipitous recommendations [Eir+18].

A user can surprisingly find out that there is for example a new chinese restaurant in an area, by only visiting that area, and get that recommendation [KWV16]. A user can find that there are new fury types of jerseys under sale on a usual e-commerce site, just because it is winter and the user is recommended that jersey, this is a demonstration of novelty. Recommender engines that incorporate contextual knowledge tend to be dynamic, novel and serendipitous [KWV16].

Ratings do lack contextual information about users [Agg16]. They lack vital information such as user preference changes, meaning that a recommender algorithm using ratings can continue using ratings which has already expired [ERR16]. User contextual information such as location can be quite phenomenal when dealing with tourist resorts recommenders, incorporating location can give an advantage to algorithms deployed in such domains. User preferences change over time [Agg16], and it is vital when it comes to recommendations [Zha+18]. Time is a very important contextual parameter that can guide recommender algorithms [Eir+18]. Time can be divided and aggregated in various forms of seasons such as day periods(morning, day, evening), events(Christmas, independence day, wedding day, black Fridays etc). If an individual is tracked on the basis of time and other contexts, recommendations may be quite satisfying to that particular user. It is a weakness to many recommender algorithms that they focus on one context only, this weakens the strength of many algorithms [Zha+18].

4.3.3 Contextual Parameters in Literature

The objectives of this section is to:

- identify existing user-contextual parameters which are being incorporated in CARS in literature,
- find existing challenges in incorporating user-contextual parameters in CARS algorithms which reduce the effectiveness of the algorithms,
- identify the optimal subset of user-contextual parameters which are strongly associated with user likes.

4.3.3.1 Methodology and Tools

Firstly a systematic literature review was done as a way to fulfill the first objective. Systematic literature review was also used to find the existing challenges in the incorporation of user-contextual parameters in CARS. After finding existing contextual parameters in literature we went on and carry two surveys in different periods of time. The surveys targeted people from all walks of life. 32.4% of the respondents were female and 67.6% male, the age ranges from 15 to above 50 years, the respondent came from different locations around the world but 96.2% were in Africa. The surveys collected peoples' contexts (contextual parameters) and their corresponding likes or preferences. Meaning that a person would enter the values of each contextual parameter according to his/her context and the items that he/she is interested in, in that particular time. The two surveys were carried in a space of 3 months apart. The data collected from the two surveys created two different data sets. 270 people responded to the first survey and we created the first data set, 306 people responded to the second survey and we created the second data set.

To fulfill the third objective an apriori algorithm was used on the two data sets. The algorithm was used to find frequent sets of contexts and their relation to user likes/preferences, the source code can be found here or `https://github.com/kavuur/CARS-Research/blob/main/First_analysis.ipynb`. The apriori is an algorithm that generate association rules from data. In this research project the apriori created rules from the item sets found from the two data sets which consists of user contexts and associated likes.

To run the algorithm on ordinary PCs with the following specifications: 4GB RAM, Intel quad core i7 processor, took 30 minutes to generate results. The main purpose of using the apriori algorithm was to identify user-contextual parameters which correlate with user likes, through identifying user-contextual parameters which are frequently found in rules with highest support and confidence values. The apriori algorithm also shows the rules in the form of user contexts which estimate better user likes. The algorithm seeks to find out frequent contextual parameters tightly coupled with user tastes. This leads to identifying dominant parameters which has strongest correlation with user likes.

The Analysis was done using the apriori and other python libraries, the data set was put in a data structure then fed into the algorithm to generate association rules. The generated association rules were further analysed to find out the contextual parameters which were tightly coupled with user likes.

4.3.3.2 Theoretical Approach and Concepts

To address these research questions articles for RS were collected from these sources :Scopus, Science direct, MDPI, Google Scholar and dblp published from 2014 to 2019. These articles mentioned either of the following search terms:

- Context-aware
- User context
- User data
- User information

The search queries managed to return a total number of 21 articles duplicates included. The Table 4.2 shows how the articles were found.

An analysis was carried out on the articles found, to answer the first research question that investigates the contextual data that is being used in literature. The results are shown here.

4.3.3.3 Accessible User Contextual Information

Most of the user contextual data which is accessible and relevant in RS is shown in the table below. The table shows each contextual data and the number of articles which it was

Table 4.2: Search terms and articles found.

Search Term	Number of articles found
Context-aware	19
User context	3
User data	1
User information	0

used or highlighted. This table highlights the most important categorical information about users which can be used by RS developers.

4.3.3.4 User-Context Modelling in Literature

Raw context data is usually dynamic and sequential, it cannot be used directly in recommender systems thus, an abstraction process is needed to summarize raw context data into context concepts for it to be usable [KFG15]. The contextual information that is gathered by RS either explicitly , implicitly or using a machine learning approach is used within recommendation models [Sun+16], [Har+17]. The incorporation of contextual data in recommendation models is classified as pre-filtering ,contextual modelling and contextual post filtering.

Pre-filtering

In pre-filtering, user-contextual parameters/data will be used as a label to filter out the recommendations which fail to relate to the user's specified contextual data [Sun+16]. The labels are used to segment the data, the process of segmentation can involve looking for co-occurrences of contextual information [San+18]. From the co-occurrences relations are found then these contextual terms are clustered and processed together [Har+17]. In contextual pre-filtering, only the filtered items are considered for recommendations. One advantage of this approach is its ability to employ any of the many classical recommendation algorithms to recommend the filtered items [Har+17].

Contextual Modelling

The contextual modelling approach consists of using contextual information directly in the recommendation model in addition to the user and item data [Sun+16], [IL16]. There

Table 4.3: Literature associated with user-contexts in RS.

Search Term	No of articles	Citations
Emotions/ mood/ mental stress	5	[IL16], [KFG15], [APO18], [Har+17], [Lit+17]
Time/ seasonality	8	[Fat18], [RAL16], [Har+17], [Col+18], [Agh18], [BAC17]7
geographical location/ location/ user's mobility history/ city/ country, zipCode, vicinity/ distance to an available point of interest/ nationality/ language	11	[KFG15], [RAL16], [Ing+14], [Har+17], [Col+18], [Col+14], [Lit+17], [NT14], [BAC17], [HCS18], [CSS15]
age	4	[Har+17], [Col+18], [Lit+17], [NT14]
occupation/ social influence/ expertise/ role	3	[Har+17], [Lit+17], [NT14]
favorite genre/ ratings/ user-interest/ intent of Purchase/ intent/ personal preferences/ user behaviors	5	[RAL16], [Har+17], [Agh18], [Lit+17], [NT14]
day type/ Activity/ current situation/ current activity/	4	[IL16], [Har+17], [Lit+17], [NT14]
gender	4	[Har+17], [Col+18], [Agh18],[Lit+17]
Weather/ current weather/ environment	3	[Fat18], [Har+17], [Lit+17]
followers/ Friend/ Parent/ social/ has-Relative/ has-Friend/ hasCo-worker/ companion/ social relations/ accompanying persons	5	[Fat18], [Har+17], [Col+14], [Col+18], [Lit+17]
budget/ technology/ device	1	[Har+17]

are many techniques that use contextual modelling which include techniques like Support vector machines [Fat18]. However Tensor factorization seems to outperform all other techniques. In some Context-Aware Recommender Systems(CARS), user profiles are compared, similarity is computed and a set of recommendations are generated based on the similarity of user contextual profiles [APO18], [Har+17].

Contextual Post-filtering

In this approach user contextual data is used after the recommendation engine has been invoked, there is reordering of the recommendations items meaning the context is initially ignored [Sun+16], [Har+17]. Similar to contextual pre-filtering, contextual post-filtering also allows the use of any of the numerous traditional recommendation algorithms [Har+17].

Much of the research work apply the pre-filtered approach [Har+17]. However the post-filtering approach has significantly outperformed the pre-filtering approach in all experimental setup [Har+17]. Most previous research also find out that context leads to a decrease in the misclassification of users.

4.3.3.5 Handling Challenges in User-contextual Parameters

Research has been carried out to incorporate user-contextual data in RS, however many challenges have also been encountered especially in gathering data. Explicit methods of gathering data rely too much on users therefore they make RS become very much like search engines and lose the capability to recommend items [Bee17]. It is worth noting as well that users are typically not interested and are reluctant to rate or search items [ERR16].

With the increasing amount of user contextual data, it has become pretty complicated to compute similarity of users [Yan+16], [MZH18]. If the neighbourhood is based on only single context, it might suffer from the sparsity problem when there are no or few neighbors meeting constraints [NT14]]. This is also amplified sometimes when various unrelated contexts are used in the recommendation process [Ung+16]. Many CARS tend to

incorporate a small set of contextual parameters which do not necessarily represent user context or define the user [Ung+16]. This sometimes leads to incorporating one or two contexts which might not be enough to represent users preferences and not enough as well for tailoring personalized services to users [Har+17]. It is also worth noting that it is challenging to obtain all the contextual information about a user [Agh18].

The other challenge is determining when and how to incorporate contexts [Har+17]. There is also a lack of automatic methods to obtain contextual information for users, therefore the acquisition of contextual information is a research area that needs to be explored [Sun+16]. Users' preferences may change over time due to user mood change or context change, identification of such a change is challenging [Agh18]. Regardless of contextual changes, the recommendation may not match the user's personal preference and this recommendation will not be useful to the user based on the current context. One final challenge being faced by CARS is overspecialized recommendations [Lu+15]. Users are restricted to getting recommendations similar to items already defined in their profiles [IFO15].

4.3.4 The SCuPE Framework

Since each item in RS is expressed by several features, general conceptual domains of items must be identified [KFG15]. One of the challenges in CARS is that user contexts which are considered for incorporation sometimes are so few that they cannot represent users preferences, thereby causing the CARS to fail to provide tailor made personalized recommendations [Har+17]. Therefore it is imperative to define the holistic contextual data that can be used by RS to generate recommendations. We characterise the holistic or comprehensive set of contextual parameters into a framework which we referred to as the SCuPE framework. SCuPE is an acronym for *Social, Cultural, Psychological and Economic* contextual parameters. The structure of the SCuPE framework is elaborated in the next section.

Table 4.4: The SCuPE Framework: *Classification of user-contextual knowledge*

Contexts Features Category	Context features
Social	followers,friend, parent, hasRelative, hasFriend, hasCoworker , companion , social-relations, accompanying-persons, crowd,age,gender, favorite genre, user-interest, intent of purchase, personal preferences, user behavior
Cultural	Time,seasonality, geographical location, user's mobility history, city,country, nationality, language, weather, environment
Psychological	Emotions, mood, mental stress, day-type activity, current situation, current study
Economic	Occupation, social influence, expertise ,role, budget, technology,device used

4.3.4.1 Significant User Contextual Knowledge

There are quite a significant number of contextual features which we can call knowledge features that can be incorporated in an algorithm, and this can bring a whole new set of advantages to recommender algorithms. From Table 4.3 we can see that the prominent contextual data used in CARS needs to be classified or characterised in a framework for the benefit of CARS. It is quite noticeable that social media can give these contextual features for particular users on any platform(web or mobile) [Eir+18], [CD17], [ERR16], [Yan+14]. Social platforms like Facebook and Twitter have developed APIs which allow developers to register for permissions and access user data that they can use to compute recommendations. For this particular case developers using this approach can use these APIs to access the user's social network profiles as well as their sentiments if these users give them access to this information. The following sections will describe each of the knowledge features in the SCuPE framework. The classification of features is shown in Table 4.4.

Social Knowledge

A person's social profile tells a story about him/her [ERR16]. Tastes can be derived from a social profile. The best user social knowledge that can be tapped by recommender algorithms can be grouped into a set, and that can be defined as a social profile. The set of social profiles can be found on social networks, and researchers can tap this information from these data sources and be able to manipulate them to test different theories. The

social context of a user is also very important in RS. A user can watch a movie because he/she is going to watch it with a friend who also like that same movie, since users ask their friend's suggestions on items like movies or restaurants [Agg16], [Eir+18]. A user's age range or education can determine interests, all these are social contexts that can be considered by a recommender algorithm. Figure 8 depicts more contextual parameters which can be classified under social knowledge.

Cultural Knowledge

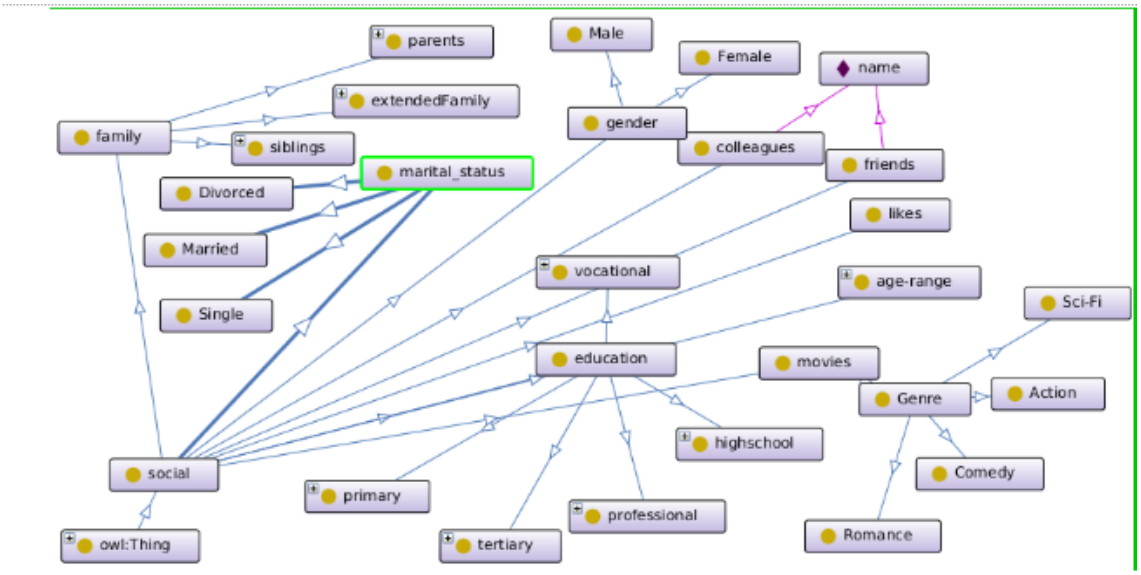

Figure 8: Social Profile

There is a thin line between social and cultural user profiles. Cultural contextual background of a user may have a great influence on his/her core values and expectations therefore it also drives interest and tastes. Cultural contexts drive aspects of life such as dressing, forms of entertainment, food, lifestyle, places to live, occupation, etc. So given a user's cultural profile an algorithm can predict tastes and preferences. Erin Meyer at Harvard Business school in 2017 explored culture on a study to find out, cultural similarities and differences among people in the same location and across the globe. The research shows that somebody's cultural values might belong to a certain sect of people even if these people do not live nearby. Therefore to know the cultural values of a user might be important, so that predicting the user's preference might be a bit easier. Figure 9 depicts more contextual parameters which can be classified under cultural knowledge.

Psychological Knowledge

Users' psychological status is very important to consider when recommending items [Kap+15],

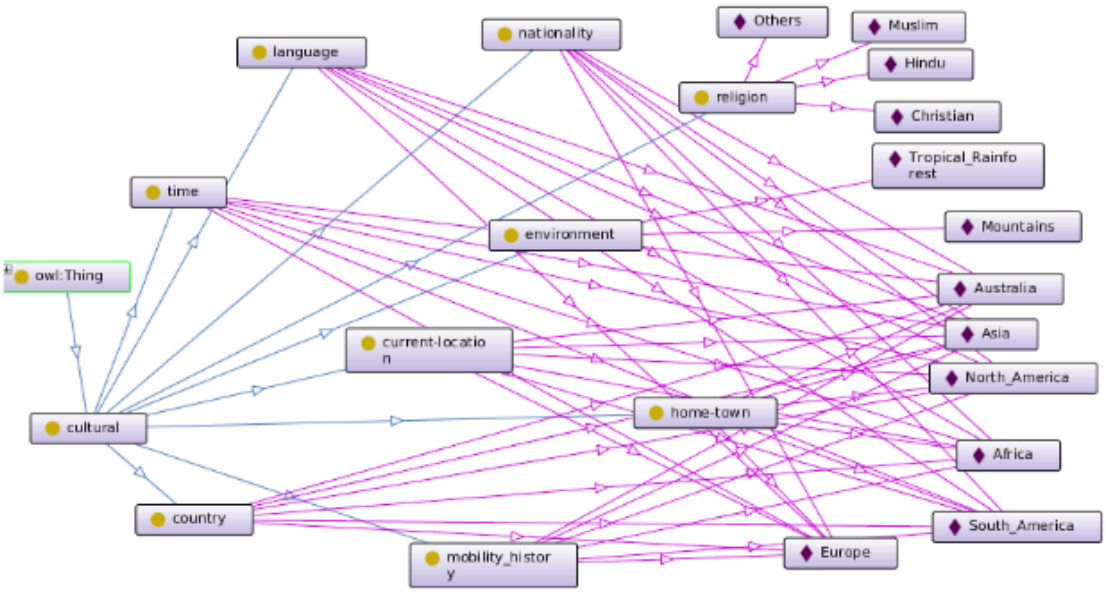

Figure 9: Cultural Profile

[KWV16]. When a user is celebrating a birthday or wedding he/she is likely to be in a certain status of mood, the user might be interested in cakes etc. If an algorithm takes this context into consideration it is likely to offer novel and serendipitous recommendations to the user [KWV16]. Figure 10 depicts more contextual parameters which can be classified under psychological knowledge.

Economic Knowledge

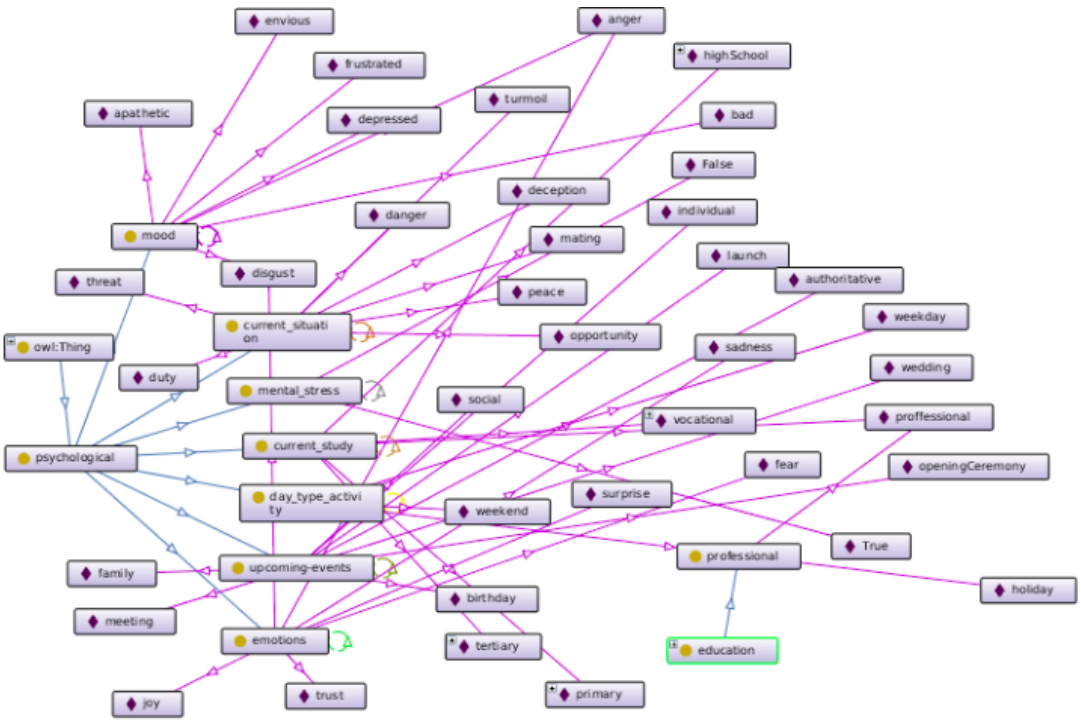

Figure 10: Psychological Profile

The economic status of users is also very crucial when recommending items, rich celebrities tend to like expensive items, best resorts places, parties etc. Therefore a recommender algorithm can detect a user's celebrity status and track followers(on a social network platform) as well as the economic status of the user. This helps to recommend novel and serendipitous recommendations which might be relevant to the user. Figure 11 depicts more contextual parameters which can be classified under economic profile.

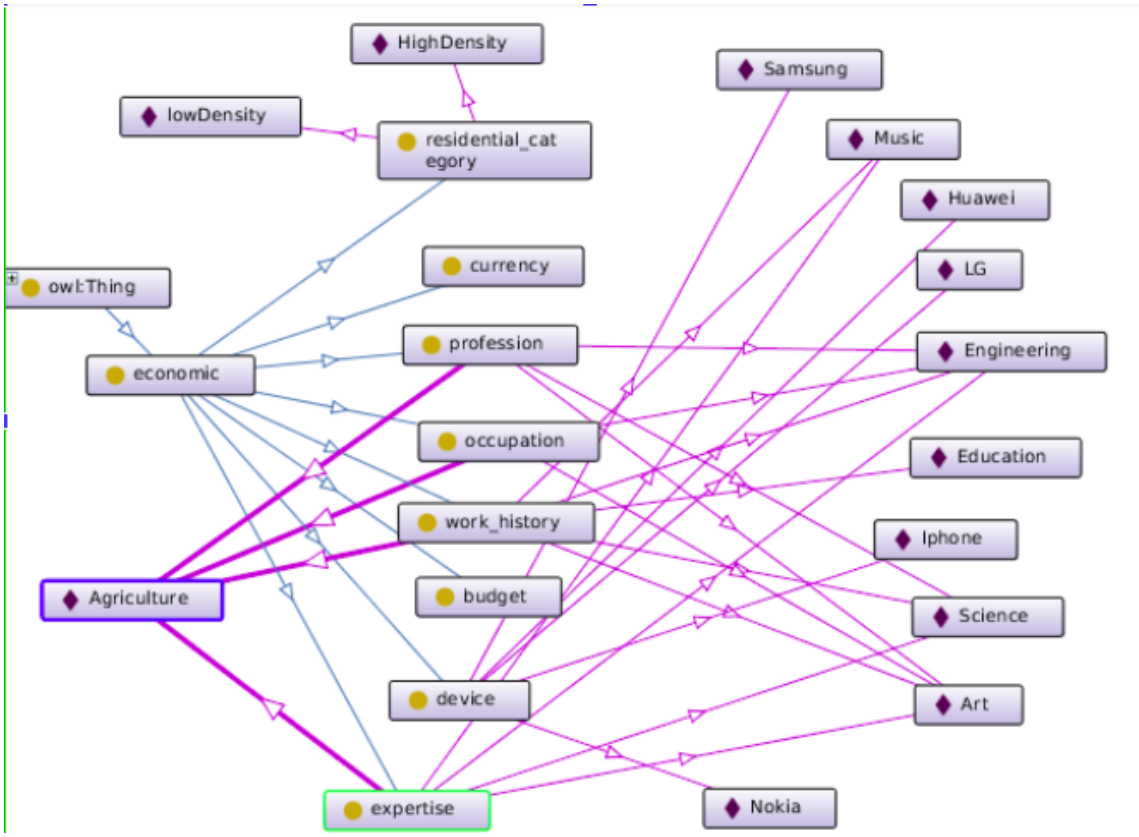

Figure 11: Economic Profile

4.3.5 Finding Optimal set of User-Contextual Parameters

A set of classified user-contextual profiles were collected from two surveys, together with their corresponding likes/preferences, the files can be found here and here. 270 users responded to the first survey and 306 responded to the second survey, they supplied their contextual information together with their corresponding likes. This was done as a way to analyse the relationship between user-contextual profiles and their likes. The set of user profiles collected are shown in Table 4.5 and 4.6. The data sets were then imported on the Jupyter notebook python platform where it was analysed using python libraries.

Table 4.5: User-contextual parameters

Contextual-Profile/Parameter	Common Content
Gender	Male/Female
Marital status	Married /Single / Divorced
House hold	Extended family/immediate family/single/alone
Education	Primary/ high school / professional/ tertiary/ vocational/ drop out
Age range	15-20/21-25/26-30
Country of origin	Zimbabwe/Zambia/USA/South Africa/Ukraine
Town/province	Harare/Gauteng/Bulawayo
Current location	Harare/Sydney/Bulawayo/Johannesburg
Language	English/Shona/Ndebele
Religion	Christianity/Hindu/Muslim
Mood	Apathetic/envious/frustrated/depressed
Current situation	threatening/ dangerous/ deceptive/peaceful
Have stress	Yes/No
Studying	Yes/No
Any event in the next two weeks	Yes/No
Emotional state	joyful/fearful/sorrowful
expertise	engineering/IT/Agriculture/Accounting
Income range(USD)	1-100/101-1000/1001-10 000

Table 4.6: Likes

Likes/Tastes	Common Content
Sport	Athletics / Soccer/ Rugby/Tennis / Swimming/ Cricket/ Others
Technology	Appliances/Industrial Machines/ Accessories/ Medical Technology/ Science/ Robotics/ Artificial Intelligence/ sensors/ Transport/ Energy/
News	hard news/ sport news/ editorial/ column/ feature news/ Lifestyle/ entertainment/ Profiles/ Investigative News/ Tech News/ Crime News/ Business News
Music	Gospel/ religious/ pop/ rock/ sungura/ jazz/ Rnb/ Dance-Hall/ Reggae/ Country/ House/ ElectronicMusic
Hobbies	Cooking/ Mountain climbing/ /fishing/ e-gaming/ sporting/ Dance Hall/ singing/ watching TV
Food	All types of beans/ vegetables, fruit, herbs, spices/ Ground nuts and seeds/ Bread/ milk/ poultry eggs/ meat/ grains/ aquatic foods

The likes were also classified as well as shown in Table 4.6. The purpose was to find the relationship between the likes (sport/technology/music/news etc) of the users and their contexts.

4.3.5.1 Relationship between user profiles and user preferences from association rules

The apriori algorithm produces some association rules that shows the relationship between user-contextual parameters and user-preferences. These are some of the association rules and corresponding likes that were retrieved from the output of the apriori algorithm.

1. **Like(s):** Dairy products;Eggs;Meat;Cereals;Seafood

   Association rules found: 7

```
RelationRecord(
        items=frozenset({'Male', 'Married', '101-1 000'}),
        support=1.0,
        ordered-statistics=[
                OrderedStatistic(
                        items-base=frozenset({'Male', '101-1 000'}),
                        items-add=frozenset({'Married'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'Married', '101-1 000'}),
                        items-add=frozenset({'Male'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'Married', 'Male'}),
                        items-add=frozenset({'101-1 000'}),
                        confidence=1.0,
                        lift=1.0
                        )
        ]
) \\

```

Code block 1: *Sample association rule: User profile vs food preferences*

This rule show that married males who get an income in the range of US$101-1000 per month preferred dairy products, eggs, meat, cereals and sea food with confidence value of 1.0 and lift value of 1.0.

2. Like(s): Soccer;Rugby

   Association rules found: 7

```
RelationRecord(
        items=frozenset({'1001-10 000', 'Immediate Family', 'Married'}),
        support=1.0,
        ordered-statistics=[
                OrderedStatistic(
                        items-base=frozenset({'1001-10 000', 'Immediate Family'}),
                        items-add=frozenset({'Married'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'1001-10 000', 'Married'}),
                        items-add=frozenset({'Immediate Family'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'Immediate Family', 'Married'}),
                        items-add=frozenset({'1001-10 000'}),
                        confidence=1.0,
                        lift=1.0
                )
        ]
)\\

```

Code block 2: *Sample Association Rule: User profile vs sport preferences*

This rule shows that married people who have an income in the range of US$1001-10000 per month, and who stay with their immediate families like/prefer Soccer and Rugby, and this had confidence=1.0 and lift=1.0.

3. Like(s): Information Technology;Sensors;Entertainment and Media

   Association rules found: 15

```
RelationRecord(
        items=frozenset({'Tertiary', 'Immediate Family', 'Married', 'Male'}),
        support=1.0,
        ordered-statistics=[
                OrderedStatistic(
                        items-base=frozenset({'Immediate Family', 'Married', 'Male'}), items-add=frozenset({'Tertiary'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'Tertiary', 'Immediate Family', 'Male'}), items-add=frozenset({'Married'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'Tertiary', 'Immediate Family', 'Married'}), items-add=frozenset({'Male'}),
                        confidence=1.0,
                        lift=1.0
                ),
                OrderedStatistic(
                        items-base=frozenset({'Tertiary', 'Married', 'Male'}), items-add=frozenset({'Immediate Family'}),
                        confidence=1.0,
                        lift=1.0
                )
        ]
)\\

```

Code block 3: *Sample Association Rule: User profile vs technology preferences*

This rule also showed that married males who have tertiary education, who stay with immediate families preferred Information Technology; Sensors; Entertainment and Media with confidence=1.0 and lift=1.0. From these few rules it can be seen that [Gender, marital-status, family setup and income] mainly has a fair relationship with users' tastes or likes.

4.3.6 Analysis on First Data set

The list of rules generated from apriori algorithm were saved in a text file. This text file was analysed by python libraries, the source code can be accessed on GitHub. Graphs were generated which demonstrate the relationship between user-contextual parame-

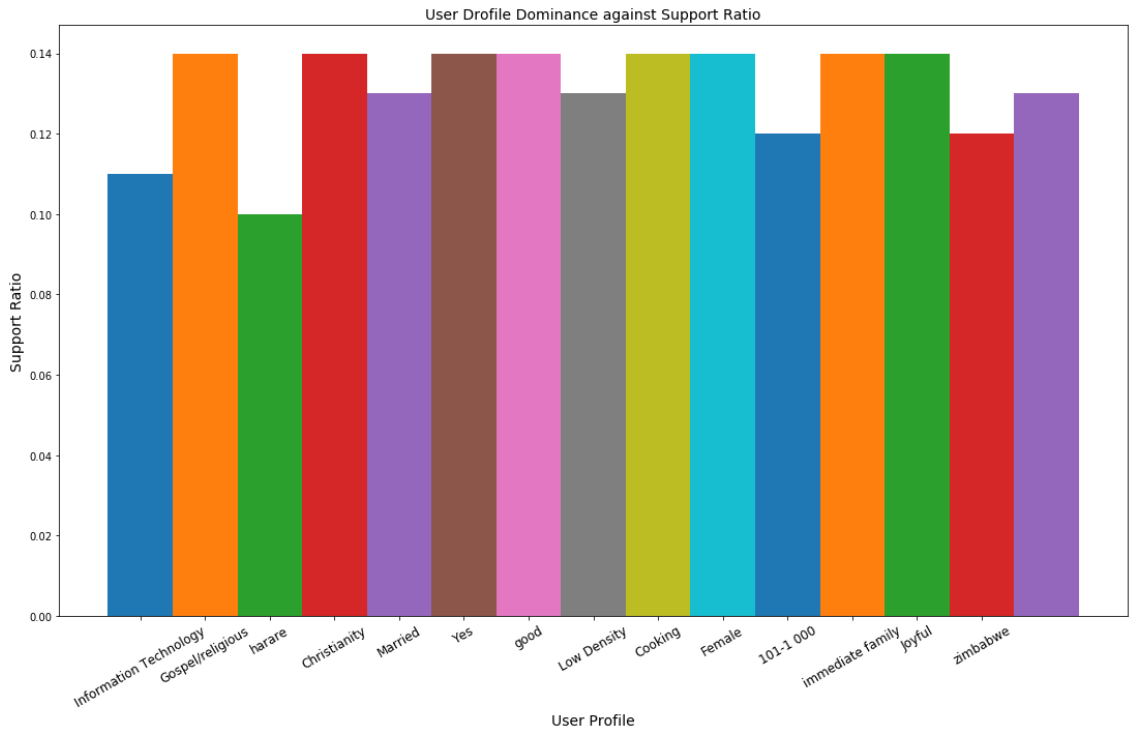

Figure 12: User context in relation to Support ratio

ters with support ratio as well as confidence values. Looking at Figure 12, the results demonstrate that there are user contexts which are more associated with people's likes and these can have more predictive power to estimate user tastes or preferences. in Figure 12, those contexts have the highest support ratio value(0.14). In this case we found that user-contextual parameters in this set [Hobbies, Mood, Emotional state, Family setup, Religion and Gender] are strongly associated with user likes or they are tightly coupled with likes. This work does not claim that this list of user contexts is exhaustive. There maybe other user contexts which might be very significant in terms of predicting user tastes. However this work found that this list might be of help to researchers and developers of CARS, when they choose user contexts to incorporate in recommendation algorithms.

Figure 13 shows a bar graph that demonstrates the relationship between user contexts and their corresponding confidence values derived from the association rules. From the association rules, each context in Figure 13 was found in any of the 160 rules that were generated. These 160 rules had their associated confidence values. Therefore user-contexts like user expertise where found in rules with confidence value less than 1.0. However there are user contexts like Hobbies, Mood, Marital status, Family setup, Home area,

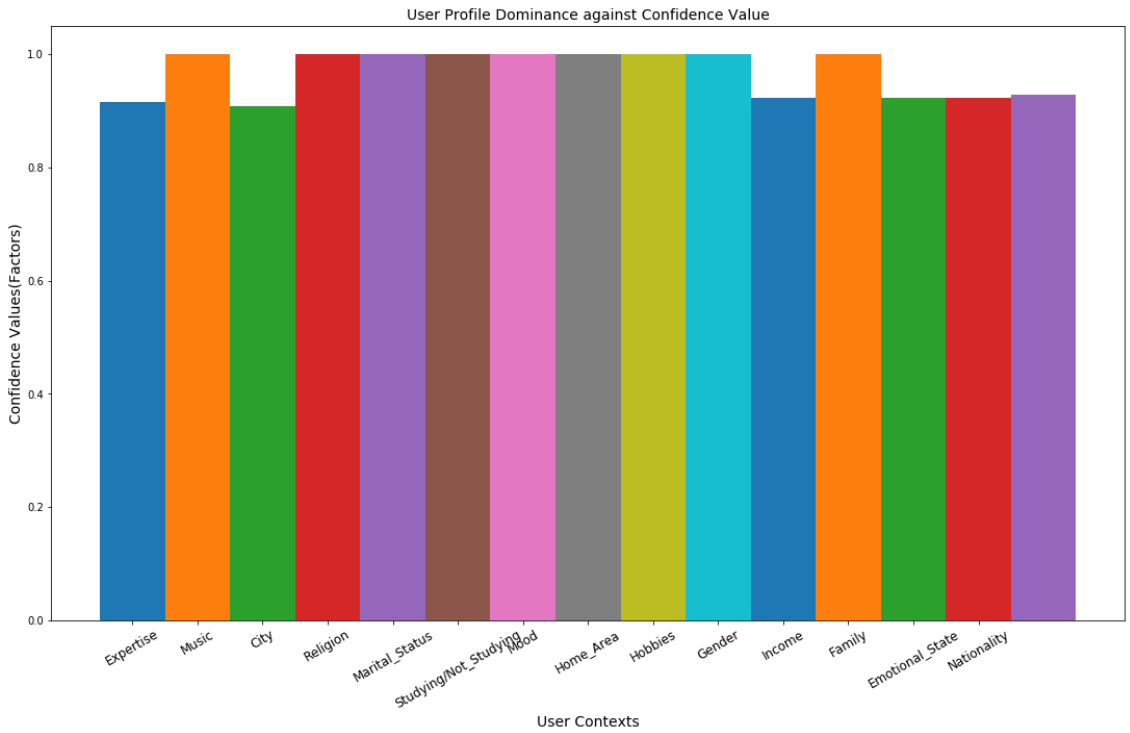

Figure 13: User contexts in relation to Confidence value.

Religion and Gender. These were mostly found in rules with a confidence factor of 1.0, meaning these are quite dominant in terms of association with user's preferences or tastes. Comparing Figures 12 and 13, we can see that these user-contexts [Hobbies, Mood, Family setup, Religion and Gender] significantly dominate as contextual features that estimate user choices. Thus, CARS can incorporate these user-contexts during recommendations so that users' may be satisfied by recommendations given by algorithms.

4.3.7 Analysis on Second Data set

A second experiment was done with another data set with 306 people's profiles together with their likes. Using python libraries, graphs were generated which demonstrate the relationship between user-contextual parameters with support ratio as well as confidence values. The source code can be accessed here. Looking at Figure 14, the results demonstrate that there are user contexts which have more predictive power to estimate user tastes or preferences. On Figure 14, those contexts have the highest support ratio value(0.14). In this case we found that the user-contextual parameters with highest predictive power are [mood, cooking, gender, family setup, religion and type of music]. The apriori algorithm using the support ratio alone identified this list of parameters as the most domi-

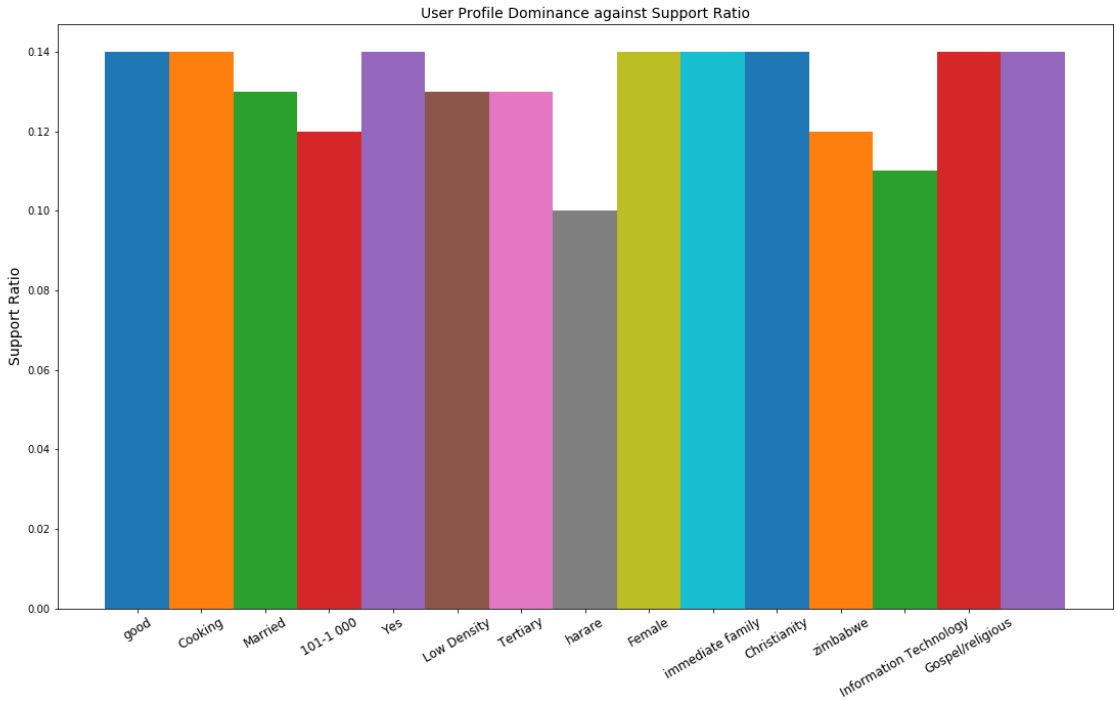

Figure 14: User context in relation to Support ratio on test set

nant to be associated with user likes/tastes.

Figure 15 shows a bar graph that demonstrates the relationship between user contexts and their corresponding confidence values derived from the association rules in the test set. Figure 15 shows that this list of parameters [mood, hobby, marital status, residential location, gender, family setup, religion, music] were identified to be much associated with user likes. Comparing Figures 14 and 15, one would see that these contexts [Hobbies, Mood, Family setup, Religion, type of Music and Gender] significantly dominate as contextual features that resonate with user choices. Looking at the first analysis with a first data set of 270 user profiles and the second test of the algorithm with 306 user profiles it seems that [Hobbies, Mood, Family setup, Religion and Gender] came up as the optimal set of user-contextual parameters which are tightly coupled with user tastes. Thus, CARS can incorporate these user-contexts during recommendations so that algorithms can predict novel and serendipitous recommendations which are tightly coupled with user tastes.

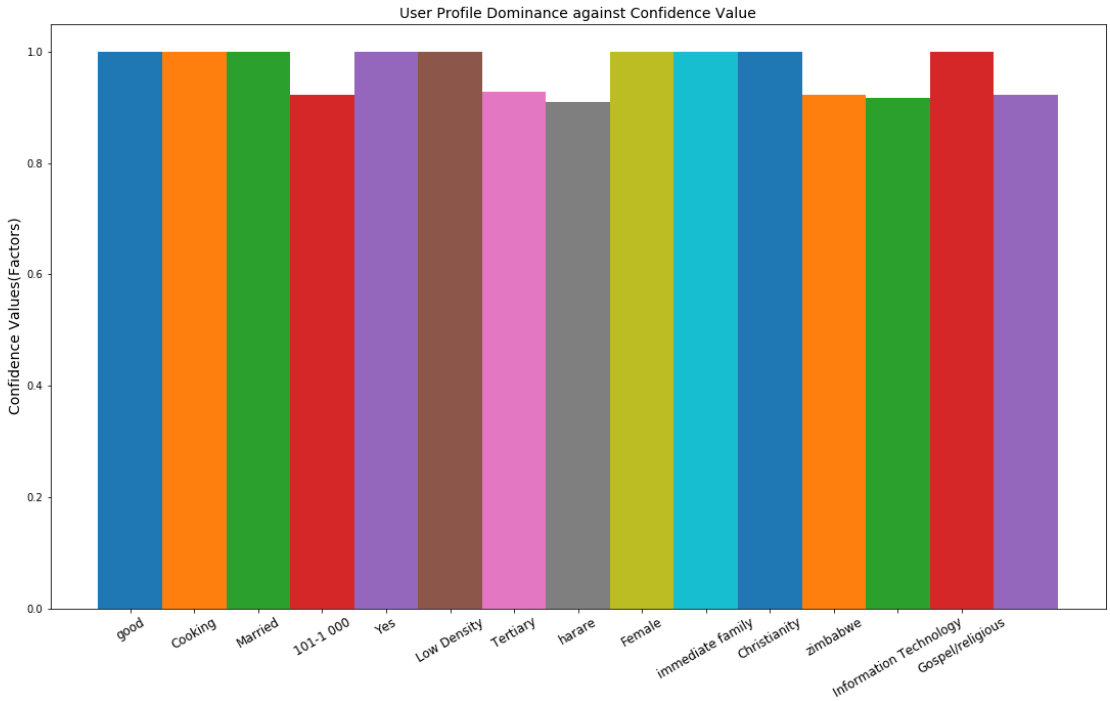

Figure 15: User contexts in Test set relating to Confidence value.

Table 4.7: Dominant contextual parameters and how they relate to literature

Dominant Contextual Parameters found in this paper	Citations
Hobbies/current-activity/day-type	[IL16], [Har+17], [Lit+17], [NT14]
Mood/emotions	[IL16], [KFG15], [APO18], [Har+17], [Lit+17]
Family-setup/relations/companions	[Fat18], [Har+17], [Col+15], [Col+18], [Lit+17]
Religion	Nothing was found
Gender	[Har+17], [Col+18], [Agh18], [Lit+17]

4.3.7.1 Dominant contextual parameters and how they relate to literature

Tables 4.7 and 4.8 were derived from the literature review spreadsheet. The purpose was to demonstrate how the dominant contextual parameters found in this research project compare with literature. In other terms, we wanted to find out if they were used in literature and how many articles mentioned or used them.

Tables 4.7 and 4.8 shows what was found in this research as compared with literature. As shown in Tables 4.7 and 4.8 user location, companions/friends and time were the most used contextual parameters in literature. However from this research user mood/emotions was found as the best to estimate user preferences. From Table 4.7 it is clear that

Table 4.8: Frequently used Contextual Parameters found in literature

Frequently used Contextual Parameters found in literature	Citations
location	[KFG15], [RAL16], [Ing+14], [Har+17], [Col+15], [Col+18], [Lit+17], [NT14], [BAC17], [HCS18], [CSS15]
time	[Fat18], [RAL16], [Har+17], [Col+15], [Agh18], [Lit+17], [BAC17]
likes	[RAL16], [Har+17], [Agh18], [Lit+17], [NT14]
age	[Har+17], [Col+18], [Lit+17], [NT14]
companions	[Fat18], [Har+17], [Col+15], [Col+18], [Lit+17]

other four contextual parameters found in this research project were also used in the literature although the numbers were quite few. This might be the case that researchers did not get their value yet in estimating users' preferences, yet this research project prove that they are important as well. There was no paper that was found during the literature review that used religion as a contextual parameter to predict user tastes. However this research project found that religion is one of the key contextual parameters that can estimate or predict people's tastes. This research project used two data sets of 270 and 306 users respectively and it might be possible that using a data set of a billion users, results might differ. However the researchers believe that this method is the best to identify valuable user-contextual parameters that can be used by CARS developers even with a data set of a billion users. The implementation of a context-aware recommender algorithm using these 5 contextual parameters was reserved for future work.

In summary, the results presented in this section show that rules which include user mood, hobbies, gender, family setup and religion always had a confidence of 1.0, meaning that the frequent sets of those contexts were always found having the same interests. Therefore, the research project brings about a very important concept that can be used by developers to pick the best user contexts which bring value to CARS.

4.4 The Recommendation Approach

The previous sections have explained the significance of different user contextual knowledge, this section explores the processes of accessing contextual knowledge, creating user models and predicting user preferences from contextual knowledge/parameters.

4.4.1 Accessing User Contextual Knowledge

Proper user contextual knowledge can be found from two sources: either psychometric surveys or social media. Chao Yang demonstrates how psychometric surveys can give almost the same result as social media analysis when predicting users' preferences on different brands based on their personal traits [Yan+15]. Their research concluded that people with certain personal traits prefer certain brands. Chao Yang and his team gathered specific user's personal traits from both psychometric surveys and tweets from Twitter. The psychometric survey asked users to state their personal values and individual needs then pick their favourite brand. They found that after mining the same individual's account from twitter and analysing their sentiments they were able to predict the individual's brand preferences with an 86% accuracy rate. The main advantage of this research is that they proved it is possible to use personal traits gathered from social media to predict a people's likes.

The format of the user information can come in two forms, i. Bag of words from sentiments ii. Keywords from social media profiles and activities. In this research we used keywords derived from social media profiles and activities but we found it interesting that the other route for sentiment analysis can be followed using the principles of content-based-filtering approach.

Social media through APIs give access to applications to get some information about their users, if the user gives permission to access information to an application. This information is rich in knowledge if used by a recommender algorithm, and this is the approach which this research took. We have classified this information as general profile and activity as shown as in Table 3.1. The general profile is somehow static and it doesn't change regularly and is not normally affected by time and events, moreover this category does not give more information about the movement of the user such as, change of tastes or general activities that the specified user does. The activity category is rich in the user's whereabouts. Change of location(for example a user can tag himself/herself on a particular location, or can notify friends on social media that he/she is flying to a certain

location), soccer team support, movie likes, friends list update, currency etc. The activity class supplies contemporary knowledge about a user's context. The information in the table 3.1 can be accessed in the form of variables from social media.

4.4.2 Modelling User-Preferences for Recommendations

RS users are naturally found in different contexts [Kni+12], [Agg16]. A user's location or place of stay can bear significant influence on his/her preferences in terms of things like dress, food, technology, mode of transport, recreation and so on [Lev+12]. The user's comprehensive contextual parameters can define that user to some extent. Contextual parameters like mood can be affected by different situations, circumstances and events like (birthdays, graduation day or funeral time). These parameters also include one's economic status i.e(profession, social status, wealth etc.). When an algorithm incorporate comprehensive contextual parameters, that algorithm could at least offer reasonable novel recommendations.

It is this knowledge that can help an algorithm to predict novel and serendipitous recommendations to the user without losing relevance. This approach creates dynamic recommendations since when the user changes context his/her neighbourhood changes depending on the category of the profile that has changed(either social, cultural, psychological or economic). Therefore if the category changes the user starts to get new recommendations from the new neighbourhood and it's highly likely that some of these recommendations are novel and serendipitous.

The structure of the data that we are using demand the use of Jaccard similarity as the best function to determine similarities of profiles so that neighbourhoods can be computed. Computing the Jaccard similarity metric between two people's profiles p1, p2 is as follows:

$$J(p1, p2) = \frac{(|P(p1)INTERSECTIONP(p2)|}{|P(p1)UNIONP(p2)|}[Lev + 12]$$

Using Jaccard coefficient to Calculate Similarities between $User_i$ and $User_n$ profiles,

$$J(u_i, u_n) = \frac{S_i \cap S_n}{S_i \cup S_n} + \frac{C_i \cap C_n}{C_i \cup C_n} + \frac{P_i \cap P_n}{P_i \cup P_n} + \frac{E_i \cap E_n}{E_i \cup E_n}$$

Graham Jenson used the Jaccard similarity metric, as a function J, to calculate the similarity between people using their histories, (History : meaning what people have done on certain items). The similarity between two people p1, p2 is the Jaccard metric between their two histories

$$J(p1, p2) = \frac{(|H(p1)INTERSECTIONH(p2)|}{|H(p1)UNIONH(p2)|}$$

Given that Tanaka liked the hobbit and hated the x-men, where Panashe only hated the x-men.

- H(Panashe) = <hate, x-men>
- H(Tanaka) = <like, hobbit>, <hate, x-men>
- H(Tanaka) INTERSECTION H(Panashe) = <hate, x-men> with cardinality 1
- H(Tanaka) UNION H(Panashe) = <like, hobbit>, <hate, x-men> with cardinality 2
- The similarity between Tanaka and Panashe is therefore

$$J(Tanaka, Panashe) = \frac{1}{2}$$

4.4.2.1 Similarity of User-Profiles

The general idea behind the modelling approach is that profiles with the same contextual parameters in these classes i.e same social, cultural, psychological and economic context are likely to be interested in the same things. Similarity of profiles can be computed using any of these metrics namely: Jaccard, Pearson correlation function and euclidean distance. Table 4.9 illustrated the difference and similarity of these functions.

The Jaccard function was chosen because of the nature of the data that was being used in this research. It is statistic function that is used to compare the similarity of sets. It is advantageous because it always work with finite sets. The Jaccard index ranges from 0 and 1.0, 0 meaning profiles are not similar and 1 meaning they are very similar. It was introduced by Professor Jaccard [MT15]. Many researchers of RS have used it in their diverse

Table 4.9: Similarity Measure Functions [Les14],[KG13]

Jaccard	Euclidean Distance	Cosine	Pearson Correlation
Similarity of finite sets	Distance between data points	Distance measure function	Linear correlation
Can work with strings	Work with vectors of real numbers	Work with vectors with integers	Linear correlational relationships
Easy with social media profiles	Computationally expensive with social media data	Weak in clustering algorithms	Weak with complex relationships among data points
Best for content-based recommendations	Good for collaborative filtering algorithms	Good for collaborative filtering algorithms	Good for collaborative filtering algorithms

work to calculate the similarity of different types of content especially in content-based RS [Lev+12], [MT15], [MTO07], [SPA15], [KC14], [SNC16], [Les14], [MT15].

4.4.2.2 Graphical Representation of the Computational Theory

Calculating similarity can be represented perfectly in a graphical setup. The distance between vertices (which represent user profiles) resembles the similarity of profiles. A graph is defined as an ordered triple (V(G), E(G), ψG. Where V(G) is a nonempty set of vertices, E(G) is a set of edges and ψG which is associated with each edge of G as an unordered pair of (not necessarily distinct) vertices of G. ψG is therefore the similarity coefficient value that depicts the similarity between two vertices. If e is an edge and u and v are vertices such that ψG(e) =uv=0.9, then e is said to be joining u and v; 0.9 shows the similarity value of the two vertices and depict that u and v are very similar.

Given a scenario as follows where:

G=(V(G), E(G), ψG)

V(G)=user1, user2, user3, user4, user5, user6, user7 ,user8

E(G)= e1, e2, e3, e4, e5,e6,e7

Suppose at some unique time t ψG is depicted as follows:

ψG(e1)= user1 user2=0.8,

ψG(e2)= user1 user3=0.7,

ψG(e3)= user4 user2=0.1,

ψG(e4)= user5 user6=0.5,

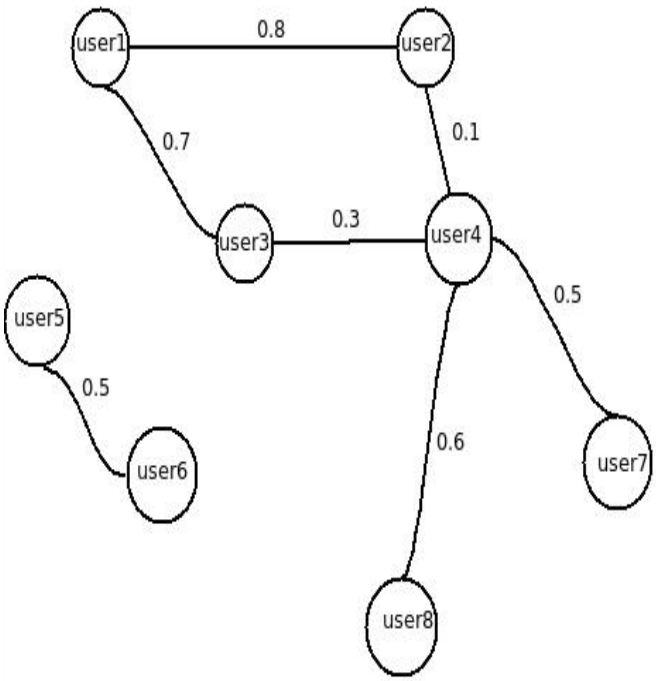

Figure 16: Similarity of user profiles.

ψG(e5)= user3 user4=0.3,

ψG(e6)= user4 user7=0.5,

ψG(e7)= user4 user8=0.6

If the distance between two vertices ranges from 0.5-1, then it means that the profiles are similar. Since the Jaccard values range from 0-1, the mean value in the range is 0.5, therefore if the Jaccard value is from 0-0.4 it means the profiles are not similar and if the Jaccard value ranges from 0.5-1.0 it means the profiles are similar. The other assumption is that similar profiles have a high probability of similar preferences. Therefore, from **Figure 16**, user 4 will be recommended items liked by users 7 and 8. Looking at the graph, we can conclude that user 8 and user 7 are likely to be similar since they are both similar to user 4. Therefore according to the HUC algorithm, both are likely to be recommended items liked by user 4. If it is on a social network user 7 will be recommended as a friend to user 8.

4.4.2.3 Efficiency in Computing Recommendations

Novel and serendipitous recommendations are normally experienced when dealing with active profiles especially during neighbourhoods computation. When the number of users grows exponentially, computing the Jaccard similarity between all active users becomes resource intensive. This will affect negatively the performance of the algorithm. Therefore the number of active users considered during neighbourhood computation should be reduced. Decision tree algorithm was found effective in this business since the data come as sets, it will be easy to make dynamic classification of profiles using the decision tree algorithm. The way it works is that when a new profile comes in, the profile will be classified in a specific class based on the parameters in the profile, within the class another process is evoked to compute neighbourhoods or (sets of very similar profiles) using the Jaccard function. This whole process bring about the much needed dynamism in RSs since classes and neighbours of a profile changes as the profile is updated by activities and context changes.

4.4.2.4 Classification using the ID_3 decision tree

The ID_3 version of a decision tree start by calculating entropy, which is a statistical metric that measures the impurity of the data set. Given a set S of user profiles, which contains two classes: Positive(meaning profile did the anticipated action) and Negative(meaning did not do the anticipated action), entropy with respect to this Boolean classification is:

$$Entropy(S) = -p(positive)log2p(positive)^{-}p(negative)log2p(negative)$$

Where ppositive is the probability of positive examples in S and pnegative is the probability of negative examples in S.

Information gain is the measure of the expected reduction in entropy. It decides which among the attributes of the concerned user's profile (the user that we want to classify) goes into a decision node(which attribute can be used to split the set). To minimize the decision tree depth, the attribute with the most entropy reduction is the best choice. The subset returned by a splitting decision must have a size of greater or equal to 50 profiles,

when the size get to less than 50, the splitting process stops and the user of concern is then classified.

$$Gain(n) = Entropy(n)^{-}([weightedaverage] * entropy) \tag{4.1}$$

$$Gain(S, A) = Entropy(S)Oentropy - \sum_{x \in values} (A)|S_v|/|S| * Entropy(S_v) \tag{4.2}$$

$$S = Each\ value\ v\ of\ all\ possible\ values\ of\ attribute\ A \tag{4.3}$$

$$S_v = Subset\ of\ S\ for\ which\ attribute\ A\ has\ value\ v \tag{4.4}$$

$$|S_v| = Number\ of\ elements\ in\ S_v \tag{4.5}$$

$$|S| = Number\ of\ elements\ in\ S \tag{4.6}$$

Given that on an e-commerce platform at a certain moment, a profile was about to be classified into a certain class which shares the same set of user-contextual parameters using the ID_3 decision tree. A set of profiles together with their transactional history (whether they have bought or not on the e-commerce platform) was retrieved from the platform as shown in Table 4.10. These very short profiles will be used to demonstrate how the ID_3 decision tree can compute the class of $user_i$. In column action, 1 represents that the user bought something, 0 represents that the user did not buy anything.

The attributes maybe{age-range,gender,location,hometown,time period,profession}, and they can have the following values:

age-range={10-15,15-23,23-32,33-45}

gender=F,M

location={Kigali, Pretoria, Joburg, Mogadishu, Lagos}

Table 4.10: A short blueprint of user profiles

user	age-range	gender	timeperiod	action
1	20-25	F	morning	1
2	25-30	F	afternoon	1
3	20-25	M	evening	0
4	30-35	M	morning	1
5	35-40	F	afternoon	0
6	35-40	M	afternoon	1
7	35-40	F	morning	1
8	20-25	F	afternoon	1
$user_i$	20-25	M	morning	1

hometown={Kigali, Butare, Harare, Luanda, Newyork}

time period={morning, afternoon, evening}

profession={IT, film, academic, indigenous, medical, education}

We need to find which attribute will be the first decision node in the decision tree.

Entropy(S) = — (7/9) Log2 (7/9) — (2/9) Log2 (2/9) = 0.764

Gain(S,gender) = Entropy(S) — (5/9)* Entropy(SF) —(4/9)*Entropy(SM)

= 0.764- (5/9)* 0.7219 — (4/9)* 0.811

=0.723

Entropy(SF)= — (4/5)*log2(4/5) — (1/5)*log2(1/5) = 0.7219

Entropy(SM)= — (3/4)*log2(3/4) — (1/4)*log2(1/4) = 0.811

Gain(S, time period)= 0.612

Looking at the scenario above, information gain of other attributes is less than 0.723, therefore gender will be used as the first decision node. $User_i$ end up in the class of males who made their transactions in the morning. Classification will continue until the class size is reduced to less or equal to 50 then the class of $user_i$ will be returned to other modules of the HUC algorithm for further computation. As shown in figure 17 $user_i$ will be in the same class as $user_4$. Therefore, a similarity function will be called to compute the similarity between $user_i$ and $user_4$.

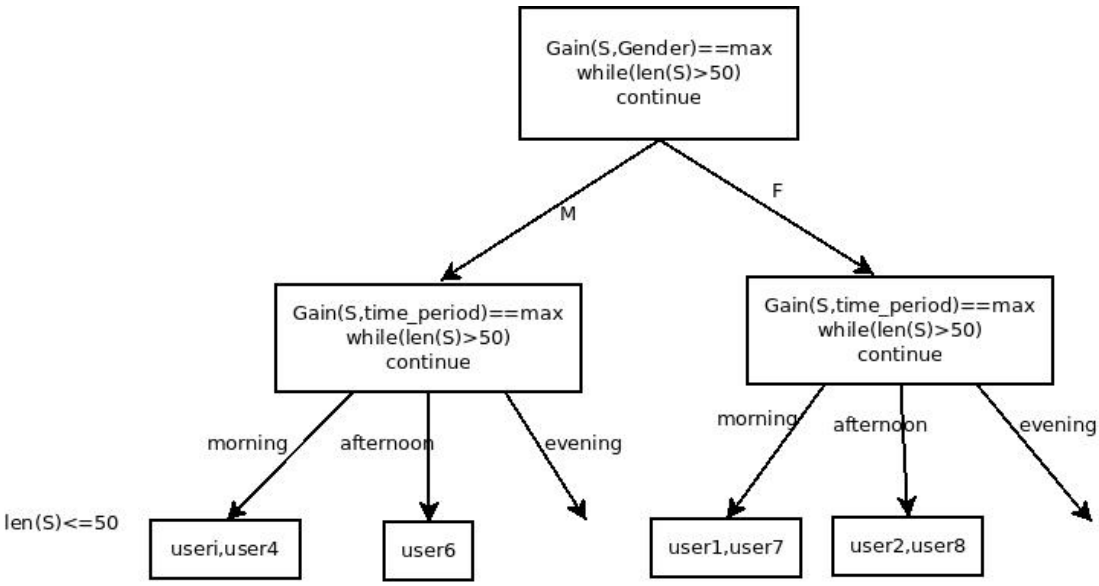

Figure 17: A snippet of the decision tree

When a profile's contextual parameters changes (the change might be in any of these: social, cultural, psychological or economic context) the ID_3 model is executed again and the class of that profile changes resulting in a new neighbourhood (since similarity of profiles is always computed due to profile updates). Predicted preferences will be computed from the current neighbourhood. Therefore the algorithm will not be stuck with past and irrelevant preferences. Novelty and serendipity will come as a by product of the dynamism occurring in classes and neighbourhoods. The first preference of items to be recommended comes from recent acted items within the neighbourhood.

4.5 Summary

This chapter explores all theoretical contributions that have been made in this research project, from user-centric conceptual framework, to the theory of preferences, and finally to the modelling of user preferences. It is a report of a new method of looking at the recommendation problem using the concepts which have been discussed in CARS research and RS research in general. The major problem addressed in this chapter is the inadequacy of user-contextual parameters used by CARS, to foster the actual preferences for users in a bid to offer dynamic and novel recommendations [Ung+16], [Har+17]. The theory highlighted that this problem had caused algorithms to lose relevance with time [LH14]. The evidence being low retentions rates with time, lack of timely adaptive, user-

specific, novel and serendipitous recommendations. This is also seen as timely inefficiency in recommender systems [JWK14], [Var15], [Hid+16].

CHAPTER 5

Evaluation of Holistic User Context-Aware algorithm: A University Student Social Network

5.1 Introduction

This chapter shows the application of the theoretical aspects discussed in chapter four. It demonstrates how the theoretical issues discussed in chapter four were applied in a Holistic User Context-Aware (HUC) Recommender algorithm, and it's evaluation on a live student social network. The experiments gives a demonstration of how comprehensive user-contextual parameters can solve the research problem. CARS have proved to have the capacity to provide novel and relevant recommendations according to changing user needs [AT08], [Lit+17], [APO18]. Research in recommendation systems (RS) has turned focus to CARS since the process of incorporating user-contextual parameters in RS is an effective approach in terms of bringing about relevant recommendations [APO18], [Agh18], [Sun+16], [CSS15]. CARS have created many advantages in RS, even if users have a limited number of initial ratings, personalized recommendations are realised [RAL16], [BAC17].

This chapter discusses the incorporation of comprehensive user-contextual parameters into a recommender algorithm using the contextual modeling approach and its demonstration in a practical situation. It shows:

- The transition from the theory discussed in chapter four to computational aspects

by conducting an experiment with the Holistic Context-Aware algorithm (HUC),

- A deployment of the algorithm to a working live platform,
- An evaluation of the algorithm in view of solving the research problem and,
- A discussion of the results found, and their significance.

5.2 The HUC Recommender Algorithm

The HUC is an algorithm that generates product recommendations to the user and it allows developers to add plugins for domain specific tasks. The general description of the algorithm is that when a user is new on a platform and has not actioned any item, the algorithm will look for this new user's profile (social, cultural, psychological and economic), then find others who have similar profiles. After finding users with similar profiles, items actioned by these similar profiles are retrieved and association-rule mining is performed to find frequent items that are actioned mostly by these similar profiles. After this stage, the new user is recommended frequently actioned items by similar profiles. If the user is not new meaning the user has some history of actioned items, this user will go through the same process as the process above, the difference being that similar profiles will be only derived from users who have actioned the same items as this particular user.

5.2.1 Analysis of Terminology

P = <s, c, k, e> is a tuple that represents the profile for a user where the order of the elements in the tuple is important. The order of items within s, c, k and e is important as well. Within $s = <s_1, \ldots, s_7>$, s_i is a social attribute (relationship-status, age-range, gender, education, likes, political-affiliation, social status). $c = <c_1, \ldots, c_5>$, c_j is a cultural attribute (religion, current-location, home-town, timezone/time-period, language). $k = <k_1, \ldots, k_4>$, k_x is a psychological attribute (birthday, friends-birthday, movies, upcoming-events). $e = <e_1, \ldots, e_5>$, e_y is an economic attribute (currency, work-history, profession, residential-category).

I = {x: x is an item/product},

I_p ={x: $x \in I$ AND x is recommended to some user profile P },

$I_p \subset I$,

U = {P: P is a user profile },

O = {P_t: $P_t \in U$ AND $t \leq T$,

T is particular time point, t is any arbitrary time point when P is created },

R = {P: $P \in O$ AND $\exists x \in I$ AND isActive(P, x) },

M = {Q: $\forall Q \in R$, some $P \in U$, JaccardSimilarity(P, Q) $\geq similaritythreshold(0.5)$},

$R \subseteq U$,

$M \subseteq R$.

If we want to test whether or not a profile P is active on a product x we use the Boolean function isActive(P, x).

For a user to get recommendations, the user's profile (<s, c, k, e>) is gathered first together with other profiles with the recent actions on the platform. The user's profile and other profiles with recent actions are used by the *computeSimilarProfiles* process in Figure 18 to return profiles that are similar to the user in question. The collection of similar profiles is often a very big dataset. Therefore, there is a process within the *computeSimilarProfiles* that implements a decision tree to trim the dataset and return profiles that are in the same set with the concerned user. The similar profiles are then used to compute *frequentItemsActionedbySimilarProfiles* this process implement the apriori algorithm. This algorithm returns items which are commonly preferred by people who have similar profiles to that of the user in question. The items might be news articles, degree programmes, hotels, books etc. These items are then recommended to the user. All other algorithms defined in this section support these main processes. The history of recommendations will be logged in a file *listofallRecommendations*, this was done to analyse the performance of the algorithm.

ComputeRecommendations computes the recommendations, it starts by passing existing profiles/users to ComputeSimilarProfile. ComputeSimilarProfile begin by extracting active profiles from the given list of profiles, then it computes and extract similar profiles to the logged in profile P from the active profiles, finally it will retain a list of

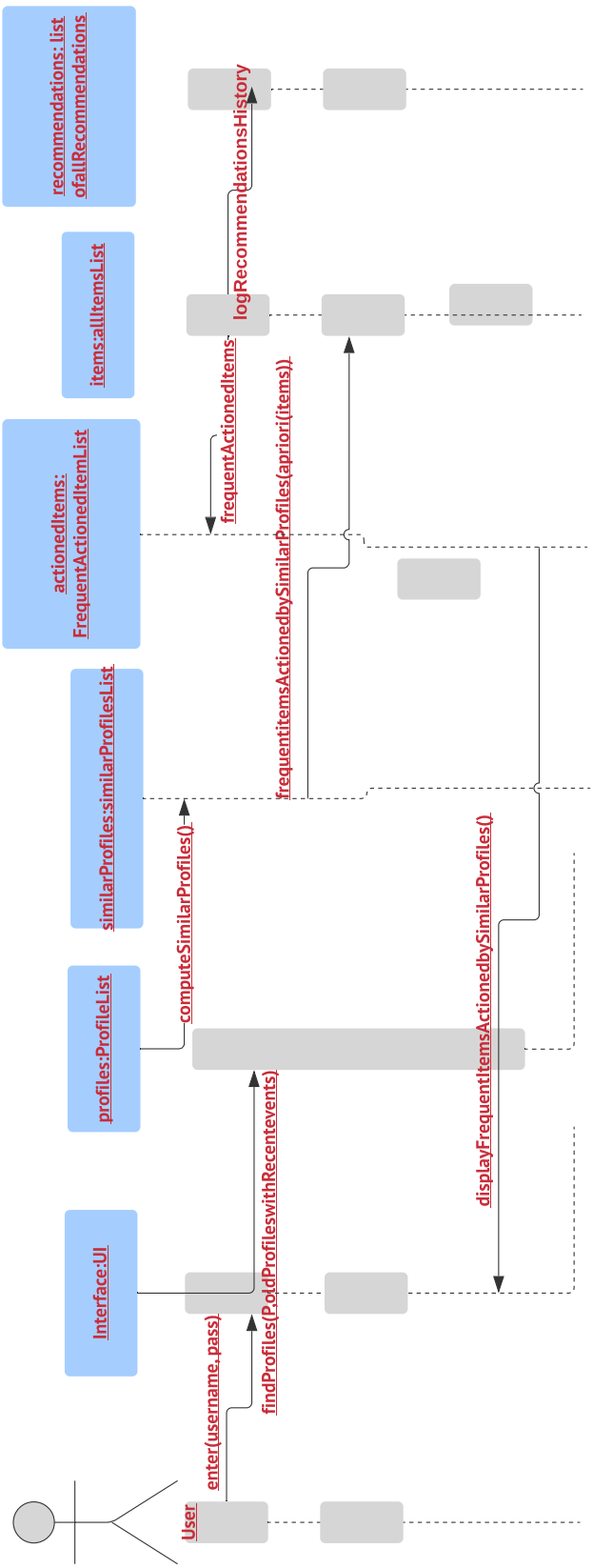

Figure 18: Sequence of events in the recommendation process.

similar profiles. ComputeRecommendations then computes frequent items actioned by similar profiles by calling ComputeItemsActionedbySimilarProfile. These frequent sets

of items are the ones recommended to the target profile. The time complexity for algorithm 4 (ComputeRecommendations) algorithm is mainly determined by the time complexity of ComputeSimilarProfile and ComputeItemsActionedbySimilarProfile which is $O(n^2)$.

```
ComputeRecommendations
INPUT:   I,P,O
OUTPUT:  Ir(items to recommend to P)
Begin
      M=(ComputeSimilarProfile(P,O)
      Items_on_demand = ComputeItemsActionedbySimilarProfile(M,I,P)
      Ir  = Items_on_demand
return Ir
End
```

Code block 4: *Compute Recommendations*

Algorithm 5 computes the similarity of user profiles. Given a list of profiles, it first picks profiles of active users and then computes the similarity between these profiles and the profile of the targeted user. Jaccard Similarity algorithm performs the underlying work of computing similarity. The time complexity of ComputeSimilarProfile is as follows:

Suppose: T(n)= Time function of an algorithm or the time needed to compute the algorithm

T(computeProfileClassSet(P,O))=nlogn

T(JaccardSimilarity(m, u))= n^2, where n is the size of m or u.

T(R)=T(isActive)=1

T(M)=T(JaccardSimilarity(m, u))= n^2

T(ComputeSimilarProfile)=T(M)+T(R)+T(computeProfileClassSet(P,O))

$=n^2+n^2+$nlogn

O(ComputeSimilarProfile) $\cong O(n^2)$

```
ComputeSimilarProfile
INPUT:  P,O
OUTPUT: M
Begin
      C = computeProfileClassSet(P,O)
      R = {r: r in C and isActive(r,I)}
      k = 0.5; (Jaccard coefficient threshold for highly similar profiles)
      M = {m: m in R AND JaccardSimilarity(m, P) >= k, for all m}
      return M
End
```

Code block 5: *Compute Similar Profile*

Algorithm 6 make use of the association rule mining (apriori algorithm) to compute frequent item sets. The main purpose of this algorithm is to come up with a set of items which are frequently actioned (accessed or selected) by the set of similar profiles. The actions might be (searching, clicking, rating etc). It's time complexity is as follows: Suppose:n= number of unique items in set I to be used to generate freqSet

T(freqSet)= T(Apriori)=n // time to compute the set freqSet is equal to the time to compute the apriori algorithm

T(T)=T(isActive)=1 // time to compute the set T is equal to the time to compute the function isActive

T(It)=T(isActive)=1

T(ComputeItemsActionedbySimilarProfile)= T(It)+ T(T)+T(freqSet)

=1+1+n O(ComputeItemsActionedbySimilarProfile) $\cong O(n)$

```
ComputeItemsActionedbySimilarProfile
INPUT:  M,P,I
OUTPUT: Ir(set of items actioned by M)
Begin
    freqSet = Apriori(M,I,P)
    Im = {X:X in I AND isActive(M,X),for all X}
    if(|freqSet|<=2):# the set of items frequently actioned by Similar
    profiles is too small
          Ir = {i:i in (Im U freqSet)}
    else:
          Ir = {i:i in freqSet}
return Ir
End
```

Code block 6: *Compute Items Actioned by Similar Profile*

Algorithm 7 reducing the list of profiles to be used during similarity computation. Its main purpose is to reduce the computational time of the ComputeSimilarProfile function by classifying the target profile in a certain class of profiles based on the contextual parameters. This process avoids computing similarity between the target profile with all

active profiles. This algorithm make use of the ID_3 decision tree algorithm. ComputeProfileClassSet divides a big list of active profiles to classes of at most 50 profiles each. ComputeSimilarProfile will then compute similar profiles from these small classes. The time complexity of this algorithm is shown: Suppose: n is the number of nodes in a decision tree

T(O)=1

T(Vo)= 1

T(ga[i])=T(Entropy)= nlogn

T(Go)= T(Entropy)= nlogn

T(Od)= T(Entropy)= nlogn

T(computeProfileClassSet)=T(O)+T(Vo)+T(ga[i])+T(Go)+T(Od)

= 1+1+nlogn+nlogn+nlogn

O(computeProfileClassSet)$\cong O(n\log n)$

```
computeProfileClassSet
INPUT:  P,O
OUTPUT: Od(set of profiles in same class as Pu )
Begin
     T = {x:x is a user attribute}
     O = {x: x = <v1,...,vn>, vi in Dom(ai), ai in T, i=1,2,...,n, n=len(T)}
     V[O]     {x:x in DomO(ai), ai in T and there exists va[i]:x = va[i] in tuple, some tuple in O } # all distinct values of ai in O
     ga[i] = Entropy(O) - sum(V[o](|Ox|/|Oa[i]|.Entropy(Ox))
     Go = {<x,y>: y=ga[i] i=1,2,...,n, n=|T|, there exists ai:ai= x in T }
     Od = {P:P in O, and for all Pa[i]  in P there exists ai : ga[i]=maxy(G0)}
     if(|Od|<=50):
          return Od
End
```

Code block 7: *Compute Profile ClassSet*

isActive which is the fifth algorithm just check if a particular profile is active on a platform or not. The time complexity for isActive algorithm is only O(1), since there is only one operation that check if a condition is True or False.

```
isActive
INPUT:  P,i
OUTPUT: bool(True or False)
Begin
    isActive = False
    ACTIONS = {click, search, rate, bought }
    A = {x:x is recent action on i by P AND x in ACTIONS }
    if(A not = {}):
        isActive = True
   return isActive
End
```

Code block 8: *isActive*

Jaccard Similarity function takes two sets of items and return a value from a range of 0-1, 1 meaning the sets are very similar 0 meaning they are not similar. Its known time complexity is as follows: T(JaccardSimilarity)=O(n^2), where n= number of items in sets involved in the computation.

```
JaccardSimilarity
INPUT: Pi,Pj
OUTPUT:k(similarity value)
Begin
    k =[(Pi or  Pj)/ (Pi or Pj)]
return k
End
```

Code block 9: *JaccardSimilarity*

The Apriori algorithm implements the association rule mining technique. It compute the frequent set using a threshold value. Its time complexity is O(n) , where n= number of unique items in set Im or in the set of all transactions.

```
Apriori
INPUT:  M,I,P
OUTPUT: F(set of items frequently actioned by M)
Begin
          k = 0.25;(support threshold)
      Im = {X:X in I AND isActive(M,X),for all X}
      FrequentSets = {X:Support(X,len(Im))>=k, X in Im}
      F={i: i in X and X in FrequentSets, for all X}
      return F
End
```

Code block 10: *Apriori*

The given time complexity for the entropy algorithm is O(nlogn), where n is the number of nodes in the decision tree.

```
Entropy
INPUT:  O
OUTPUT: Entropy Value
Begin
        Entropy = -(|O+|/|O|) Log2 (|O+|/|O|) - (|O-|/|O|) Log2 (|O-|/|O|)
      # O+ number of profiles in O with positive actions
        return Entropy
End
```

Code block 11: *Entropy*

```
Support
INPUT:  X(set of items),len(Im)(number of items transacted)
OUTPUT: Sratio(support ratio of X)
Begin
      count=0
      for all i in Im
          if(i==X)
             count=count+1
       Sratio= (count/len(Im))* 100
       return Sratio
End
```

Code block 12: *Support*

5.3 Methods and Tools

The HUC algorithm was evaluated using experimental online evaluation techniques since it was hosted on Unipals a growing social network site for Zimbabwean universities students. This site is mainly is used by university employees and students and when the evaluation process was done the users of the social site were mainly students and staff members from different universities across Africa. By the time it was tested in December 2018, the platform had 1486 users and the number is still going up. The 1486 users was a good sample size for a reasonable online evaluation.

Aim and Objectives of the experiment: To find out how the algorithm cope with the dynamics of user likes and expectations and to what extent does it offer novel and serendipitous recommendations.

Background: The algorithm can be integrated with a web/mobile application that provides recommendation services to its clientele. When a client is interacting with an application, the client's data and segmented history is sent to the algorithm so that it will compute recommendations. After computation, the algorithm returns items for recommendations which would then be presented to the client's interface. This procedure is shown in Figure 19.

Procedure: In this experiment the HUC was integrated with Unipals. The source code of the algorithm is found here github. The algorithm recommends friends. According to the structure of the algorithm, it recommends items which are frequetly being accessed by people of similar contextual parameters. Therefore in the context of Unipals it recom-

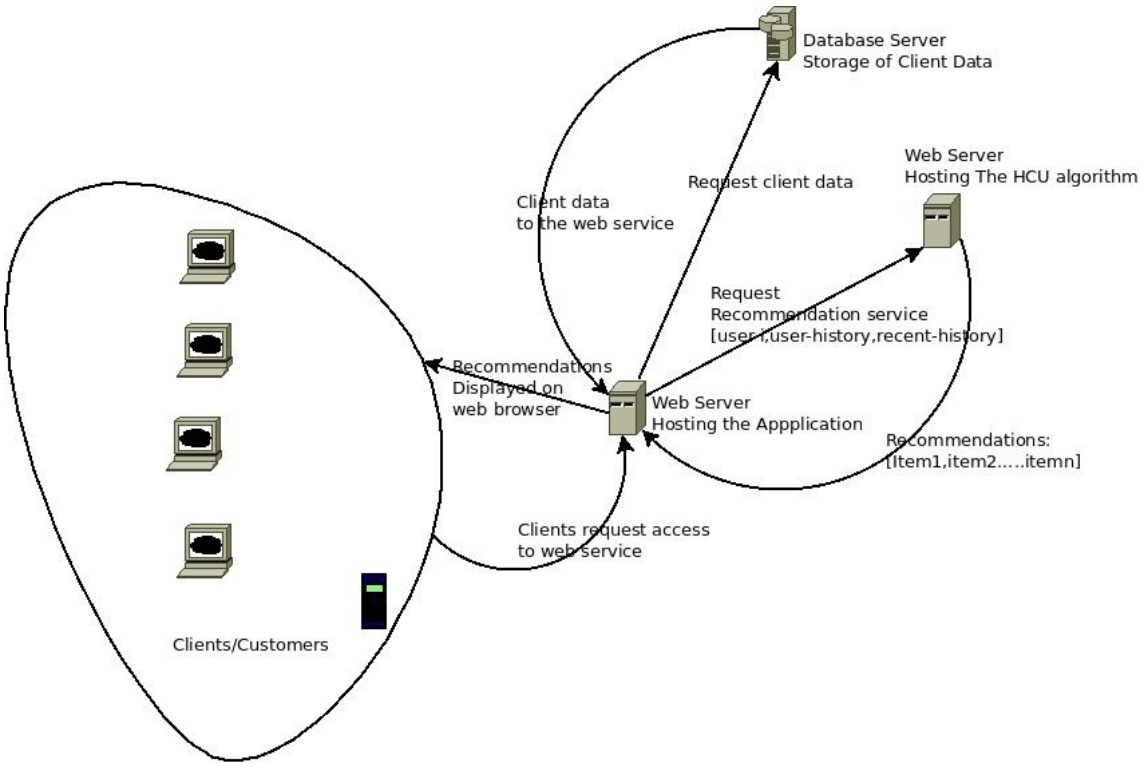

Figure 19: Experimental Setup: Clients Interacting with the HUC algorithm

mends people of the same contextual parameters and pals who were frequently getting in touch with people of the same contextual parameters. The users' profiles which are mainly composed of social, cultural, psychological, economic context are implicitly built as the user interacts with the platform. The profiles which were used by the algorithm consists of location, countryOfOrigin, time, friends, age, gender, picture-tags, language, likes, hobbies and interests. This list of comprehensive user-contextual parameters increases the confidence of the algorithm to offer novel recommendations. The algorithm computes and then recommends friends to the subject/target user. The summarized procedure of the whole process is depicted in Figure 19. Since this was a live online experiment, the users were not aware that the algorithm was under test. The algorithm captured how the target user responded to recommendations.

Dataset Design and Preparation: The interaction of users with the site and the interaction of users with the algorithm was logged and saved in a .csv file which can be accessed on log file. For security reasons, the user profile data is not given for analysis. The information which was captured during the transactions is shown on Table 5.1.

Table 5.1: Transactional Data

Data Captured	Data Type	Reason
Recommendation ID	Int	To uniquely identify each recommendation
Subject user	Object	To identify the user, receiving recommendations
Recommended time	time stamp	To figure out the time which the subject user received a recommendation
Clicked user	Object	To find out which user was clicked during the recommendation
Clicked time	time stamp	To find out what time was the clicked user clicked.

The recommendation ID in Table 5.1 represents the recommendations that have been offered to a target user. The subject user is the target user that is receiving recommendations and the clicked user is the user that was clicked by the subject user. When a target user logs in, he/she is recommended a friend, if the user clicks the recommended user, the recommended user becomes the clicked user. The recommended time is the time when the recommendation was made and the clicked time is the time when the clicked user was clicked.

Online Evaluation Experiment: The advantage of online evaluation, is that you are dealing with real users. This approach is less susceptible to bias during the user sampling process [Agg16]. From this background, online evaluation was applied as the sole approach to assess the performance of the HUC algorithm. The interaction of the users with the algorithm was recorded. The live process can be viewed here. This dataset has the recent and old recorded transactions. The online evaluation involve metrics such as: precision, coverage, diversity, novelty and serendipity.

5.4 Results and Discussion

Figure 20 shows my homepage just after logging in. The algorithm gave recommendations before I even made friends on the platform, showing how it solved the cold start problem although that was not one of the major objective of the research project, since recommendations were derived from the user's profile. Previous research has shown that

the cold-start problem was a serious problem in RS for sometime. The HUC algorithm is generic and in this case it was tested on a social site, therefore recommended users/friends were treated as items. If a new user(a user who has no interactive history) login, that user will receive recommendations of users with similar contextual profile and friends of users with same contextual profile. If a user who has some history login, the user will receive recommendations of users in the same context, friends of users with the same context and friends of people within the same context whom have common friends with the logged in user.

When the target/logged in user clicks one of the recommended friends, the algorithm computes again new novel recommendations. User interactions were captured and exported as csv dataset for analysis, python libraries were used for analysis and visualization. From the data gathered for analysis, (theuser) is the one that received recommendations and clicked (the user which was clicked by the theUser). From the dataset more than a thousand recommendations where generated and the first 700 recommendations generated a lot of responses from the users. 1115 recommendations were offered to 250 users, which entails that there were 250 active users on the platform, and these 250 active users received recommendations at least once from the algorithm.

Figure 21 shows the record of the number of active users and the time periods which they

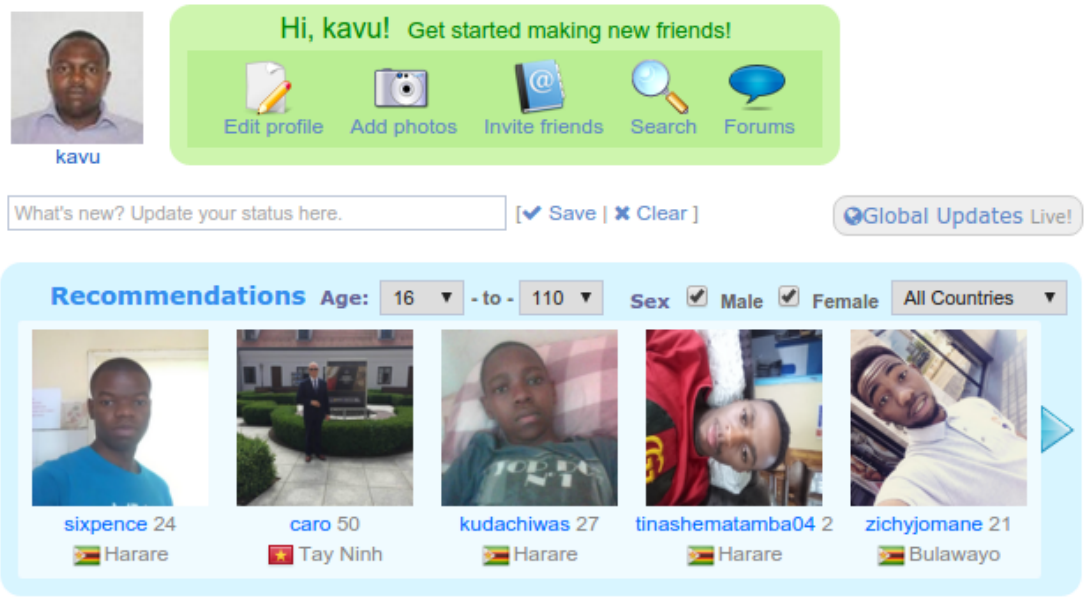

Figure 20: A snippet of recommendations made when a user logs in

responded to recommendations. Some records show that one active user responds to more than one recommendations, meaning that when the user logged in different times he/she selected/picked recommendations displaying the dynamism that was exhibited by the algorithm. The graph demonstrates that there was much activity in the morning as compared to other time sessions, this might because people normally go on social media before and after the busy day.

Time sessions were encoded as 0.0-1.0(midnight), 1.0-2.0(afternoon), 2.0-3.0(evening)

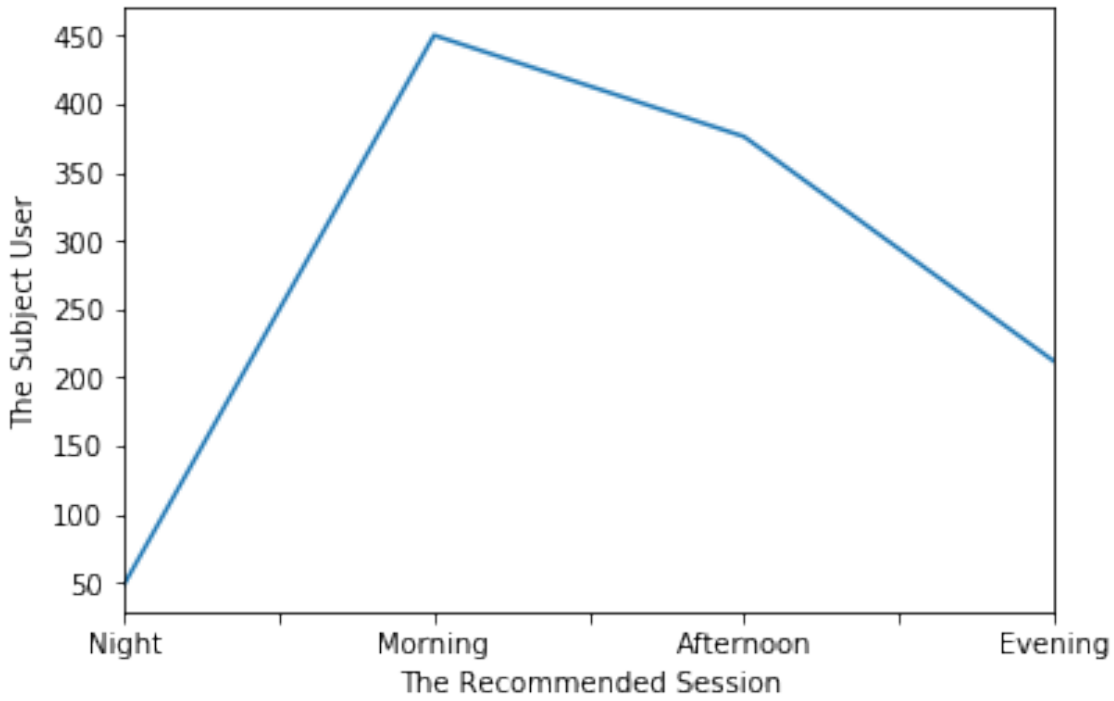

Figure 21: The relationship between subject users and recommendation time

and 3.0-4.0(morning) as shown in Figure 22. A lot of activity happened in the morning as compared to all other times. Figure 22 also shows that the first 100 target users responded a lot to their recommendations. This demonstrates that the algorithm was not affected by the cold start problem(which is a failure to draw any inferences for new users due to insufficient information of those users) since the first users who created accounts on the social network were recommended to each other and these recommendations where of interest to them. The user profiles of the users which were used in this experiment where not availed for analysis due to privacy policy of the social network.

Table 5.2 shows the first 20 records of user transactions, the-user column is a unique identification of active users which received recommendations. The column clicked rep-

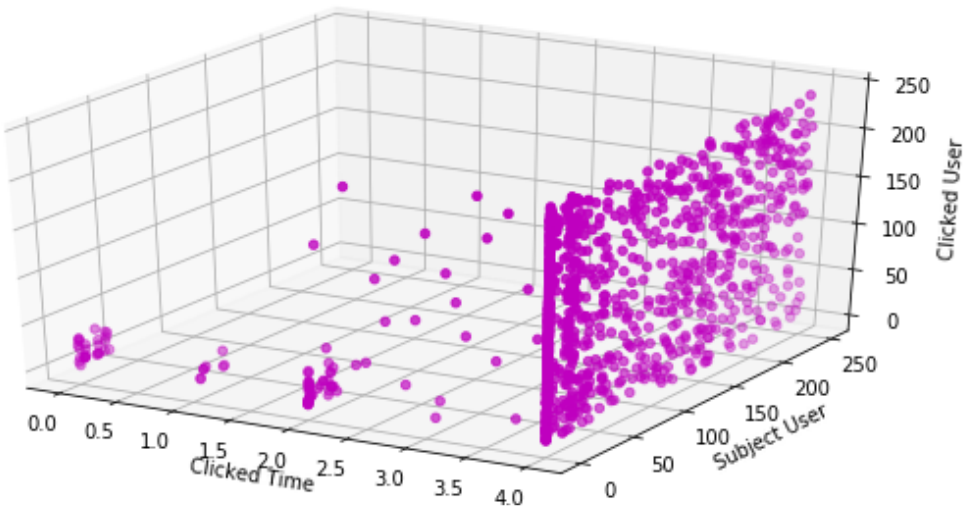

Figure 22: The relationship of subject user,clicked user and time

Table 5.2: A Snippet of recorded transactions.

DateRecommended	DateClicked	TimeRecommended	Session	TheUser	Clicked
2018-04-20 07:57:00	2018-04-20 07:57:00	7:57:0	Morning	0	0.0
2018-04-20 09:05:00	2018-04-20 09:05:00	9:5:0	Morning	0	0.0
2018-04-20 09:14:00	2018-04-20 09:15:00	9:14:0	Morning	0	1.0
2018-04-22 11:42:00	2018-04-22 11:42:00	11:42:0	Morning	1	5.0
2018-04-22 11:42:00	2018-04-22 11:42:00	11:42:0	morning	1	6.0
2018-04-23 08:17:00	2018-04-23 08:17:00	8:17:0	morning	2	7.0
2018-04-23 11:16:00	2018-04-23 11:17:00	11:16:0	morning	3	8.0
2018-04-23 14:29:00	2018-04-23 14:30:00	14:29:0	afternoon	5	9.0
2018-04-24 13:20:00	2018-04-24 13:20:00	13:20:0	afternoon	6	11.0
2018-04-23 17:13:00	2018-04-23 17:13:00	17:13:0	afternoon	1	11.0
2018-04-24 14:07:00	2018-04-24 14:08:00	14:7:0	afternoon	1	3

resents the users which were clicked by the-user, and ID numbers for the-user is different from clicked. If we look at the-user(1), we can see that the user is quite active, the-userID1 clicked clickedIDs 6,11,and 3 at different times, that is (11:42,17:13,14:08) respectively. These user's where clicked at different dates and time, which shows that the algorithm was a bit dynamic and changes its recommendations with the time, and the user preferred the recommendations.

True Positive (tp): refers to the recommended users which the target user clicked. False Positive (fp): refers to the recommended users which the target user did not click.

5.4.1 Precision

Since users log in sessions, it is important to figure out if unique recommendations per session would give valuable information. Suppose that each session took 10 minutes before a user log out. The total number of recommendations generated per session by the algorithm on the recorded transactions were 646. From the 646 recommendations per session, 77 active profiles responded to the recommendations per each unique session. Therefore the accuracy rate or precision is given below:

$$\frac{tp}{tp + fp} * 100 = (77/(646)) * 100 = 11.920\% \tag{5.1}$$

5.4.2 Diversity

Diversity refers to the uniqueness of the recommended items. From the 646 generated unique sessional recommendations, 251 of the 646 where unique recommendations(users which were never been recommended before). Which means the diversity rate of the algorithm was (251/646)*100= 38.9 %. This then shows many of the registered users where not active on the platform, therefore the algorithm was left with no option but to recommend the remaining active users which were not that unique.

5.4.3 Coverage

Coverage looks at the total number of items in the database and out of the total, how many were considered during recommendations. From the time when the transactions where recorded the social network had 1486 users and out of these 1115 where picked for recommendations. Therefore coverage rate becomes:

$$\frac{no - users - involved}{total - users} = (1115/1486) * 100 = 75.03\% \tag{5.2}$$

5.4.4 Novelty

Novelty determines how unknown or unpopular is a recommended item to the target user [Zha13]. That is, given $L \subseteq I_R$, where IR is the recommended list , L is the set of items in IR that the user(u) likes, L can be partitioned as $L=L_k \cup L_u$ into those items,L_k is already known items to the user and L_u is unknown items to the user. In this case a user

(item) was considered unknown if the user was never recommended to the logged in user before. Then the novelty per user is :

$$\text{Novelty}(R) = \frac{|Lu|}{|L|} \tag{5.3}$$

Where R is the set of all active users who received recommendations The average novelty of the algorithm is :

$$\sum_{u \in R} (|L_u|/|L|)/|R| = 0.85561x100 = 85.561\% \tag{5.4}$$

This shows that the novelty of the algorithm was quite high, from the transactions which were recorded

5.4.5 Timely Serendipity

A recommendation is serendipitous if it is novel and relevant(Saúl Vargas Sandoval, 2016). Serendipity looks at how surprising the successful or relevant recommendations are.

Given that:

1. Average number of recommendations: R at time t_1 and time t_2

2. Average number of obvious(known) items on time t_1 and t_2: q

3. Number of non-obvious items on time t_1 and t_2 = R-q

S_{user_i} :

count=0.0

$\forall i \text{ in } [R - q]$

if[R-q] are useful(if clicked) at t_1 and t_2

$\text{count} = \text{count} + 1$

$S_{user_i} = \text{count}/|\text{totalRecommendations}|$

$$\text{AverageSerendipity} = \sum^{n_i} (S_{user_i}/|\text{total}_{\text{recommendations}}|)x100 = 84.395\% \tag{5.5}$$

The average serendipity in percentage was 84.395 %

Before the HUC algorithm was integrated on Unipals, the social site had 20 users in May 2018, but when this algorithm was integrated the number increased sharply to 1486 from May to July 2018, now it has 1929 and it is still growing. Users' familiarity with a prospective friend was tracked from users' history, if a user has been recommended a friend before that friend is labeled familiar and this was done implicitly. However we did not consider other ways which the user might have been familiar with the friend outside the social site.

5.5 Comparison to Related Work

CARS works with different structure of data and therefore makes it difficult to perform offline comparisons with different algorithms. In this research project we took the path of online evaluation and therefore we compared the HUC algorithm with some of the RS and CARS algorithms that have been designed before. It seems many researchers are reluctant to give numeric performance of their algorithms especially in terms of novelty and serendipity. We found that the algorithm was quite comparable with existing algorithms, surprisingly the algorithms which were not CARS performed very well as compared with published metrics for some existing CARS. We found that for novelty algorithms designed by [Var15] had an average of 75.48%, [Zha13] and [HZ11] had an average of 77%, [Var16] had an average of 77%, [AT08] which is CARS had average novelty of 73%. HUC algorithm proved to be quite novel with an average novelty of 84%.

In terms of serendipity [Lu+15]'s algorithm had average serendipity of 28.67%, [Zha+15] had an average of 73%, [KB11] had an average serendipity 83% and [Zha+18] 69%. The HUC had an average of 84%, therefore it was quite comparable with the state of the art. The algorithm recommendations was also changing with time sessions, meaning it was quite dynamic in it's recommendation strategy. The accuracy rate was very low, since there seems to be a trade-off between serendipity and accuracy [KB11]. The hypothesis was supported that, if an RS incorporates comprehensive user-contextual parameters, mainly in

the form of social, cultural, psychological and economic profile, there is a greater chance of offering dynamic, novel and serendipitous recommendations.

This chapter demonstrated an experiment that was performed to prove the application of the theoretical contributions in chapter four. The chapter shows how the theoretical contributions contributed to a holistic context-aware recommender algorithm, the deployment of the algorithm on a social site and its evaluation. The algorithm proved to be quite novel and serendipitous, given that the average novelty and serendipity to each user was ranging at 84%. The diversity of the algorithm was a bit low with an average of 38% due to the number of active users on the platform, since the algorithm only consider active users to generate neighbourhood. The social site was not performing well on poor mobile networks, and this resulted in users visiting the platform only when using public networks or WIFI, this reduced the number of active users and affected the performance of the algorithm to a certain extent. However the coverage rate of the algorithm was at an average of 75% and this entails that the number of available users were mostly considered when constructing recommendations. Finally the algorithm demonstrated that given timely detailed profile of users(social, cultural, psychological and economic context) a CARS algorithm can offer dynamic, novel and serendipitous recommendations.

5.6 Summary

We have seen from literature that existing context-aware recommendation algorithms/systems lack a full understanding of the user in so much that they are inadequately capable of coping up with the dynamics of the user preferences. However existing work again points to the importance of user-context when estimating preferences. It has been proved from existing work as shown in this thesis that considering user contexts when offering recommendations improve significantly the quality, novelty and serendipity of the recommendations given. Therefore this research project explored the characterisation of the user context and found that the social, cultural, psychological and economic user-context is the best source of user-context which can be used to estimate preferences. An algorithm was implemented to test the impact of this theory and it shows a significant

improvement on dynamism, novelty and serendipity of recommendations given. The algorithm was evaluated using online evaluation methods so that we can be able to judge the novelty and the serendipity of the algorithm, and it performed better as compared with existing context-aware recommender algorithms.

CHAPTER 6

Conclusions and Future Directions

This chapter provides a discussion of the overall research objectives and how they were fulfilled, and also a highlight of major contributions and future research directions. The contributions are also compared with those found in the literature. During the research project some gaps were discovered and these are given as research directions for researchers.

6.1 Assessment of Thesis Objectives

Chapter one highlighted the research objectives as 1) Establish the inadequacies in RS evaluation methods and user contexts, 2) Investigate and develop the theory of user-contextual parameters in CARS, 3) Determine best user-contextual parameters which can be incorporated in CARS, 4) Use comprehensive set of user-contextual parameters to estimate user preferences and generate recommendations and 5) Evaluate how the method in (4) performs with respect to dynamism, novelty and serendipity.

6.1.1 Objective 1: Establish the inadequacies in RS evaluation methods and user contexts

On the first objective we came up with a conceptual evaluation framework(UEC). That framework was used to explore and assess existing evaluation methods. We found that some major improvements in the present state of the art needs to be done to satisfy users when interacting with recommender systems, as it is shown that prediction accuracy is not the major objective but also user satisfaction [KR12], [Kni+12], [Lei14]. Recommender systems need to help users make better decisions, algorithms should serve the needs of users even at the expense of accuracy and precision [Cre+12], [Lei14], [Kav+16]. Other

evaluation metrics such as site utility need a closer look since eliciting user perception of utility is a very difficult task because it is more psychological than computational [SG11]. The reason why a user buys depends not only on what is presented on a recommendation list but on issues like income and the purpose of the object to be bought [SMG09].

6.1.2 Objective 2: Investigate and develop the theory of user-contextual parameters in CARS

Under this objective the theory of user preferences and user context was extensively investigated and discussed. Some work in CARS showed that some contextual parameters like time and location has already been implemented in some CARS. However we discovered that most of the contextual parameters used do not fully represent a user or define a user fully. Therefore we came up with a framework (SCuPE) which define a comprehensive set of user-contextual parameters which can represent the context of the user. The SCuPE framework represent parameters which can be classified into social, cultural, psychological and economic classes. The SCuPE framework was used to derive recommendations for users.

6.1.3 Objective 3 : Determine best user-contextual parameters which can be incorporated in CARS

On this objective, an approach to characterise user-contexts was used in a bid to find out the optimal set of user-contexts which can be employed by recommendation algorithms to estimate user preferences. An apriori algorithm was employed to find out dominant contexts from two data sets of user-contextual profiles with associated likes. Rules were generated and contexts which were found in rules with associated high support ratio and confidence values were extracted. An analysis was done on the rules generated to draw graphs which show the relationship between extracted contexts against support ratios and confidence values. From the analysis done among all 21 user-contextual parameters, Hobbies, Mood, Family setup, Religion and Gender were found to be the most dominant user-contextual parameters. These contextual parameters can be incorporated in recommender systems that offer services in areas like social networks, music, travel and tourism, e-commerce, movies and food services.

Tables 4.8 and 4.7 shows a summary of the user-contextual parameters which were found in this research compared to the literature. The purpose of the table is to demonstrate how the dominant contextual parameters found in this research compare with those in literature. In other terms, we wanted to find out if the parameters found during this research were used in the literature and how many papers mentioned or used them.

As shown in Tables 4.8 and 4.7, user location, companions/friends and time were the most used contextual parameters in literature. However from this research user-contextual parameters which include hobbies, mood, family setup, religion and gender were found to be the most dominant or best user-contextual parameters. There was no paper that was found during the literature review that used religion as a contextual parameter to predict user tastes. However, this research found that religion is one of the key contextual parameters that can estimate or predict people's tastes. This research used data sets with a maximum of 306 users and it might be possible that having a data set of billion users may yield different results. However the researchers believe that this method is the best to determine valuable user-contextual parameters that can be used by CARS developers even with a data set of billion users. The implementation of a context-aware recommender algorithm using only these 5 contextual parameters was reserved for future work. In summary, the results presented in this section show that rules which include user mood, hobbies, gender, family setup and religion always had a confidence of 1.0, meaning that the frequent sets of those contexts were always found having the same interests. Therefore, this research brings about a very important concept that can be used by developers to pick the best user contexts which bring value to RS. These contextual parameters can be incorporated in RS that offer services in areas like (social networks, music, travel and tourism, e-commerce, movies and food services).

6.1.4 Objective 4: Use comprehensive set of user-contextual parameters to estimate user preferences and generate recommendations

An approach that make use of comprehensive user-contextual parameters was investigated to tackle three issues 1) to cope to evolving user preferences, 2) user-focused rec-

ommendations and 3) novelty and serendipity in recommendations. The significance of the approach was evaluated in a Holistic User-Context aware algorithm. The algorithm takes a hybrid approach of collecting comprehensive user-contextual parameters (which can be accessed from social media) then use a collaborative filtering approach to generate recommendations. The algorithm was integrated with a social network for Zimbabwe university students Unipals for evaluation. The comprehensive user-contextual parameters mainly include the social, cultural, economic and psychological context of a user. The algorithm had a time complexity of $O(n^2)$, where n refers to the number of user profiles considered for neighbourhood computation.

6.1.5 Objective 5: Evaluate how the method in objective (4) performs with respect to dynamism, novelty and serendipity

From the algorithm evaluation, the algorithm demonstrated a reasonable level of novelty and serendipity, since the average novelty and serendipity per each profile was 84%. The novelty of state of the art algorithms were as follows, [Var15] had an average of 75.48%, [Zha13] and [HZ11] had an average of 77%, [Var16] had an average of 77%, [AT08] which is CARS had average novelty of 73%. The serendipity of state of the art algorithms were as follows, [Lu+15]'s algorithm had average serendipity of 28.67%, [Zha+15] had an average of 73%, [KB11] had an average serendipity 83% and [Zha+18] 69%. The algorithm only considers active users to generate neighbourhoods based on user profiles. The diversity of 38% was comparable to Hurley and Zhang [HZ11], [Zha+11], which had a diversity range of 38–44%; and Zanitti et al [ZKS18] which ranges from 25–52%. The regular users of Unipals mainly visit the platform when using public hotspots or and WIFI, which affected the overall number of active users. The coverage rate of the algorithm was 75%, and this means that all available active users were mostly considered when recommendations were computed. Finally, the research gave a lot of insights about the significance of considering comprehensive user contexts in CARS, since this will make RS realize dynamic, relevant and novel recommendations.

The results were quite promising for a social platform and due to lack of availability of working platforms in other domains the algorithm was only tested on a social network.

Table 6.1: Published Contributions

Contribution	Paper(s)
UEC Framework for Evaluation	"A Characterisation and Framework for User-Centric Factors in Evaluation Methods for Recommender Systems". Published in: International Journal of ICT Research in Africa and the Middle East 6.1 (2016)
SCuPE Framework for User Contextual Modelling	"A Novel, Serendipitous and Dynamic User-Centric Recommender Algorithm". Published in: EAI International Conference for Research, Innovation and Development for Africa(2018),"Holistic User ContextAware Recommender Algorithm". Published in: Mathematical Problems in Engineering (2019).
HUC Algorithm	"Holistic User ContextAware Recommender Algorithm". Published in: Mathematical Problems in Engineering (2019)
Optimal set of Contextual Parameters	"Characterisation of User-Contexts for Context-Aware Recommendation Systems(2020)". Submitted at : User Modeling and User-Adapted Interaction Springer Journal

We reserve the deployment of the algorithm in other domains for future work.

6.2 Review of Contributions

Table 6.1 highlight published papers which came from the contributions made in this thesis. The contributions are explained in the following sections.

6.2.1 UEC Framework for Evaluating RS Evaluation Methods

Many RS evaluation methods have been discussed in literature [Lei14], [Lu+12], [Kni+12], [KDR19], [Cre+12], [KR12]. However, in this research project we identified that critical factors about user satisfaction(user-centric factors) were not rigorously evaluated and these factors include user-decision making processes, user experience and user interactions and this implied weak RS [WN12], [BS12], [KR12], [Cre+12]. An investigation focusing on the user-centric evaluation methods was carried out in this research project. The investigation shew some challenges when evaluating user satisfaction [JWK14], [Var15], [Hid+16], [Tin16]. This gave the researcher the basis to characterize user-centric evaluation factors and then propose a User-centric Evaluation Conceptual(UEC) framework

which act as a diagnosing tool to show the detailed missing factors in user-centric evaluation methods. An integrative review approach was used to formulate both the characterization and the conceptual framework for investigation. The results showed the necessity to create a comprehensive evaluation framework that looks into both system-centric and user-centric evaluation concepts and to bring computational means in user-centric evaluation methods. The investigation concluded that RS evaluation was missing some important user-centric concepts or factors such as: user interaction, user experience and user decisions making processes. An analysis of these factors during RS evaluation has the capacity of bringing a different type of RS that enhance user experience and increase revenue in the long run.

6.2.2 SCuPE Framework for User Contextual Modelling

One of the contributions that was made in this research project was classification of user-contextual parameters into four classes: social, cultural, psychological and economic. All the contextual parameters can be characterised as SCuPE Framework. SCuPE is an acronym for social, cultural, psychological and economic user contexts. A person's social profile tells a story about him/her [ERR16]. Social profile data can be found on social networks, and researchers can tap these sources and manipulate them to test different theories. A user can watch a movie because he/she is going to watch it with a friend who also likes that same movie, since users ask their friend's suggestions on items like movies or restaurants [Agg16], [Eir+18]. There is a thin line between social and cultural user profiles. Cultural contexts were derived from aspects of life such as dressing, forms of entertainment, food, lifestyle, places to live, occupation, etc. So, given a user's cultural profile, an algorithm can predict tastes and preferences.

Users' psychological status which incorporate mood is very important to consider when recommending items [Kap+15], [KWV16]. When a user is celebrating a birthday or wedding he/she is likely to be in a certain mood, the user might be interested in cakes etc. If an algorithm takes this context in consideration it is likely to offer novel and serendipitous recommendations to the user [KWV16]. The economic status of users is also very crucial when recommending items, rich celebrities tend to like certain items, best resorts places,

parties etc. When users are in the same context that is social, cultural, psychological and economic context, we found out that those users have a high chance of having the same preferences. Part of the future work is to determine among the four classes the class that best estimates a user's tastes or preferences.

6.2.3 Optimal set of User Contextual Parameters in CARS

CARS enhance the quality of recommendations by making use of context data that define or tell a story about a user [AT08], [Lit+17], [APO18]. Research in RS has recently turned attention to CARS for solutions because of their capacity to create more accurate and relevant recommendations [APO18], [Agh18], [Sun+16], [CSS15]. CARS has created many advantages in RS, even if users have a limited number of initial ratings, personalized recommendations are realised [RAL16], [BAC17].

Generally, if an RS is based on a single context, it might suffer from the data-sparsity problem when there are no or few users who are in the same context [NT14], [Ung+16]. The presently incorporated contexts have not been investigated if they improve the prediction capacity of the CARS. In this research a characterisation was done with the intention of determining user-contextual parameters which bring value to recommendations. An apriori algorithm was employed to find out dominant contexts from a data sets of user-contextual parameters with associated likes. Rules were generated and contexts which were found in rules with associated high support ratio and confidence values were extracted. A data science approach was used to analyse the data sets and draw graphs which show the relationship between extracted contexts against support ratio and confidence values. From the analysis done among all 21 user-contextual parameters (Hobbies, Mood, Family Setup, Religion and Gender) were found to be the most dominant user-contextual parameters. These contextual parameters can be incorporated in RSs that offer services in areas like (social networks, music, travel and tourism, e-commerce, movies and food services).

6.2.4 HUC Algorithm for Adaptive User-centric Recommendations

The main principle behind the modelling approach was that users with the same set of user-contextual parameters have the same taste. The similarity of profiles was computed by the Jaccard similarity function. The main contribution was a recommender algorithm which used the contextual modelling approach in CARS to incorporate characterised user-contextual parameters. At the end this contributed to enhanced dynamic, novel and serendipitous recommendations. During the experimental evaluation it was also confirmed that there is a trade-off between serendipity and accuracy rate [KB11].

6.3 Future Research Directions

This thesis advances research in CARS specifically in assessing RS and incorporation of comprehensive or holistic user-contextual parameters in CARS. During the process of working on these contributions, some research directions were also found which can further enlighten RS researchers, stakeholders and developers.

6.3.1 Holistic Evaluation for CARS

Different methods of evaluating CARS has been applied and investigated however there is a lack of standardization in the evaluation of RS especially CARS. Some methods work better on offline evaluation some do better with online evaluation. Some are user-centric and some are system-centric, there is a need to have a holistic evaluation standard or framework that can be used to evaluate all types of CARS. This framework should be able to cater for online and offline evaluation. On the same note, there is a serious lack of standardised offline evaluation data sets in CARS. A researcher can come up with a CARS which incorporate different types of contextual parameters but the data sets available do not provide the required features to evaluate the provided algorithm. Therefore this creates a speculation environment whereby the algorithm gets a credit which it does not deserve because there is no standard procedure for evaluating algorithms in CARS. The evaluation standard that we recommend should have both theoretical and computational means to evaluate CARS looking at all evaluation metrics.

6.3.2 Valuable user-contextual parameters incorporation in CARS

Research that was carried out in this project found that user-contextual parameters[Hobbies, Mood, Family setup, Religion and Gender] were the best parameters to estimate user tastes. Developers can develop algorithms that incorporate these user-contextual parameters. However one of the findings in this research was that, user-contextual parameters are found in four contexts which are; social, cultural, psychological and economic contexts. Meaning that the best user-contextual parameters are found in these four contexts. Research needs to be done to find out among these four classes which one can better estimate user preferences in services like (social networks, music, travel and tourism, e-commerce, movies and food services). This can help developers to optimise the effectiveness of recommender algorithms in different domains.

6.3.3 Commercialization of an Application Programming Interface(API)

In terms of commercialization, one of the future work is developing an API that implement the HUC algorithm. Businesses which run on web or mobile platforms can integrate with the API, so that they get recommendation services at a fee. The API will get data for recommendations from the platform which it gives services to and the engine computes recommendations and give an output of recommendations to the platform, then the platform will display recommendations to its clientele. User activities on the platform would always be fed to the HUC algorithm through the API. This is done so that the algorithm will work with recent user actions and will compute recommendations based on fresh and relevant user preferences.

6.4 Summary

This thesis demonstrate a piece of work that solve current problems in recommendation systems. There have been problems in online businesses on how the integrated recommendation systems can continue recommending timely, relevant and novel recommendations, since research proved that recommendations increase online business by 35%. This research project was implemented to solve these recurrent problems. Firstly a thorough investigation was done by designing and applying a User-centric Evaluation Con-

ceptual (UEC) framework, in a bid to understand the depth of the problems. During the investigation some pertinent issues to do with the surroundings or the context of the user arises. Then the research took the direction of investigating the theory of user preferences and user contexts, the SCuPE framework was then created as a result of the investigation. It was imperative that the research project need to come up with a method to generate recommendations based on the comprehensive set of user-contextual parameters defined by the SCuPE framework. The HUC algorithm was then designed and deployed on a social network, it was evaluated to judge if it solve the problems. The final result showed and proved that the method employed in the HUC algorithm, improve significantly the situation in context-aware recommendation systems(CARS). If it is implemented in many deployed recommendation algorithms the recurrent problems can be significantly reduced. The thesis also demonstrates how the research objectives were fulfilled, it goes on to show some of the future research directions that can be taken by other researchers. The final conclusion is that all the intended tasks to solve the research problem were done, the limitations of the contributions were discussed. The research community can take and use this work as a valuable piece of knowledge in the body of knowledge in recommendation systems.

APPENDIX A

Publications Associated With Thesis

1. Kavu T D, Dube K, Raeth P G and Hapanyengwi G T (2016). *"A Characterisation and Framework for User-Centric Factors in Evaluation Methods for Recommender Systems"*. International Journal of ICT Research in Africa and the Middle East 6(1), 1–16. DOI : 10.4018/IJICTRAME.2017010101

2. Kavu T D, Raeth P G, Dube K, and Hapanyengwi G T (2018). *"A Novel, Serendipitous and Dynamic User-Centric Recommender Algorithm"*. *EAI International Conference for Research, Innovation and Development for Africa*. https://dl.acm.org/doi/abs/10.4108/eai.20-6-2017.2270015

3. Kavu T D, Dube K, Raeth P G and Hapanyengwi G T (2019). *"Holistic User ContextAware Recommender Algorithm"*. Mathematical Problems in Engineering (2019). https://doi.org/10.1155/2019/3965845

4. Kavu T D, Dube K, Raeth P G and Hapanyengwi G T (2020). *"Characterisation of User-Contexts for Context-Aware Recommendation Systems(2020)"* : User Modeling and User-Adapted Interaction Springer Journal (Submitted)

References

[Agg16] Charu C Aggarwal. *Recommender Systems*. 1st. London: Springer International Publishing Switzerland, 2016, p. 518 (cited on pages 17–19, 21, 29, 31, 42, 47, 55, 68, 87, 101).

[Agh18] Mehdi Hosseinzadeh Aghdam. "Context-aware recommender systems using hierarchical hidden Markov model". In: *Physica A: Statistical Mechanics and its Applications* 518 (2018), pp. 89–98. DOI: https://doi.org/10.1016/j.physa.2018.11.037 (cited on pages 1, 4, 33, 51, 53, 65, 66, 77, 102).

[APO18] Ivana Andjelkovic, Denis Parra, and John O'Donovan. "Moodplay: Interactive music recommendation based on Artists' mood similarity". In: *International Journal of Human Computer Studies* 121.April 2018 (2018), pp. 142–159. DOI: 10.1016/j.ijhcs.2018.04.004 (cited on pages 2, 51, 52, 65, 77, 102).

[AT08] Gediminas Adomavicius and Alexander Tuzhilin. "Context-aware recoAdomavicius, G., & Tuzhilin, A. (2008). Context-aware recommender systems. Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08, 335. doi:10.1145/1454008.1454068mmender systems". In: *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08* (2008), p. 335. DOI: 10.1145/1454008.1454068 (cited on pages 17, 77, 93, 99, 102).

[AW12] Wil van der Aalst and Wil. "Process Mining". In: *ACM Transactions on Management Information Systems* 3.2 (July 2012), pp. 1–17. DOI: 10.1145/2229156.2229157 (cited on pages 37, 43).

[BAC17] Zahra Bahramian, Rahim Ali Abbaspour, and Christophe Claramunt. "A Cold Start Context-Aware Recommender System for Tour Planning Using Artificial Neural Network and Case Based Reasoning". In: *Mobile Information Systems* 2017 (2017). DOI: 10.1155/2017/9364903 (cited on pages 1, 51, 66, 77, 102).

[Bao+16] Jie Bao et al. "Geo-social media data analytic for user modeling and location-based services". In: *SIGSPATIAL Special* 7.3 (2016), pp. 11–18. DOI: 10.1145/2876480.2876484 (cited on page 2).

[Bee17] Joeran Beel. "It ' s Time to Consider ' Time ' when Evaluating Recommender-System It ' s Time to Consider ' Time ' w hen Evaluating Recommender- System Algorithms [Proposal]". In: October (2017). DOI: 10.13140/RG.2.2.33885.10729 (cited on pages 2, 52).

[Bha+13] Upasna Bhandari et al. "Serendipitous recommendation for mobile apps using item-item similarity graph". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8281 LNCS (2013), pp. 440–451. DOI: 10.1007/978-3-642-45068-6_38 (cited on page 20).

[Böh+13] A. E. Böhmer et al. "Lack of coupling between superconductivity and orthorhombic distortion in stoichiometric single-crystalline FeSe". In: *Physical Review B - Condensed Matter and Materials Physics* 87.18 (2013), pp. 1–4. DOI: 10.1103/PhysRevB.87.180505. arXiv: 1303.2026 (cited on pages 15, 34, 36).

[BS12] Jürgen Buder and Christina Schwind. "Learning with personalized recommender systems: A psychological view". In: *Computers in Human Behavior* 28.1 (Jan. 2012), pp. 207–216. DOI: 10.1016/J.CHB.2011.09.002 (cited on pages 35, 37, 100).

[Buc06] Leigh Buchanan. *Leigh Buchanan (2006), A brief History of Decision Making, HBR, January 2006.pdf*. 2006 (cited on page 45).

[CD10] Calin Ciordas and Jeroen Doumen. "An evaluation framework for content recommender systems - The industry perspective". In: *Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2010*. 2010, pp. 273–277. DOI: 10.1109/WI-IAT.2010.279 (cited on pages 16, 34).

[CD17] Mattia G. Campana and Franca Delmastro. "Recommender Systems for Online and Mobile Social Networks: A survey". In: *Online Social Networks and Media* 3-4 (2017), pp. 75–97. DOI: 10.1016/S1386-9477(99)00237-4 (cited on pages 1, 54).

[Che+18] Rui Chen et al. "A Survey of Collaborative Filtering-Based Recommender Systems: from Traditional Methods to Hybrid Methods Based on Social Networks". In: *IEEE Access* 6 (2018), pp. 64301–64320. DOI: 10.1109/ACCESS.2018.2877208 (cited on pages 1, 45).

[Col+14] Luis Omar Colombo-Mendoza et al. "MobiCloUP!: A PaaS for cloud services-based mobile applications". In: *Automated Software Engineering* 21.3 (2014), pp. 391–437. DOI: 10.1007/s10515-014-0143-5 (cited on page 51).

[Col+15] Luis Omar Colombo-Mendoza et al. "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes". In: *Expert Systems with Applications* 42.3 (Feb. 2015), pp. 1202–1222. DOI: 10.1016/j.eswa.2014.09.016 (cited on pages 65, 66).

[Col+18] Luis Omar Colombo-Mendoza et al. "Towards a knowledge-based probabilistic and context-aware social recommender system". In: *Journal of Information Science* 44.4 (2018), pp. 464–490. DOI: 10.1177/0165551517698787 (cited on pages 51, 65, 66).

[Cre+12] Paolo Cremonesi et al. "Decision-making in recommender systems: The role of user's goals and bounded resources". In: *CEUR Workshop Proceedings* 893 (2012), pp. 1–7 (cited on pages 10, 11, 14, 33–35, 37, 96, 100).

[CSS15] Zohreh Dehghani Champiri, Seyed Reza Shahamiri, and Siti Salwah Binti Salim. *A systematic review of scholar context-aware recommender systems*. Feb. 2015. DOI: 10.1016/j.eswa.2014.09.017 (cited on pages 51, 66, 77, 102).

[Eir+18] Magdalini Eirinaki et al. "Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions". In: *Future Generation Computer Systems* 78 (2018), pp. 413–418. DOI: 10.1016/j.future.2017.09.015 (cited on pages 45, 47, 54, 55, 101).

[Eks+11] Michael D. Ekstrand et al. "Rethinking the recommender research ecosystem". In: *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11*. New York, New York, USA: ACM Press, 2011, p. 133. DOI: 10.1145/2043932.2043958 (cited on pages 13, 34, 35).

[Eks+15] Michael D. Ekstrand et al. "Letting users choose recommender algorithms: An experimental study". In: *RecSys 2015 - Proceedings of the 9th ACM Conference on Recommender Systems* (2015), pp. 11–18. DOI: 10.1145/2792838.2800195 (cited on pages 2, 3).

[ERR16] Mehdi Elahi, Francesco Ricci, and Neil Rubens. "A survey of active learning in collaborative filtering recommender systems". In: *Computer Science Review* 20 (2016), pp. 29–50. DOI: 10.1016/j.cosrev.2016.05.002 (cited on pages 46, 47, 52, 54, 101).

[Fat18] Ismail Kassou Fatima Zahra Lahlou, Houda Benbrahim. "Context Aware Recommender System Algorithms : State of the Art and Focus on Factorization Based Methods". In: *Computer Science Review* 10 (2018) (cited on pages 1, 51, 52, 65, 66).

[FO17] Evgeny Frolov and Ivan Oseledets. "Tensor methods and recommender systems". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7.3 (2017). DOI: 10.1002/widm.1201. arXiv: 1603.06038 (cited on pages 44, 45, 47).

[Har+17] Khalid Haruna et al. "Context-Aware Recommender System: A Review of Recent Developmental Process and Future Research Direction". In: *Applied Sciences* 7.12 (2017), p. 1211. DOI: 10.3390/app7121211 (cited on pages 2, 4, 30, 32, 50–53, 65, 66, 75).

[Hay+02] C. Hayes et al. "An on-line evaluation framework for recommender systems". In: *Recommendation and Personalization in eCommerce* (2002), p. 50. DOI: 10.1.1.11.5630 (cited on page 31).

[HCS18] Daniel Horowitz, David Contreras, and Maria Salamó. "EventAware: A mobile recommender system for events". In: *Pattern Recognition Letters* 105 (2018), pp. 121–134. DOI: 10.1016/j.patrec.2017.07.003 (cited on pages 1, 51, 66).

[Her+04] Jonathan L. Herlocker et al. "Evaluating collaborative filtering recommender systems". In: *ACM Transactions on Information Systems* 22.1 (Jan. 2004), pp. 5–53. DOI: 10.1145/963770.963772 (cited on pages 3, 11, 14, 34, 35).

[Hid+16] Balázs Hidasi et al. "Parallel recurrent neural network architectures for feature-rich session-based recommendations". In: *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems* (2016), pp. 241–248. DOI: 10.1145/2959100.2959167 (cited on pages 33, 76, 100).

[HZ11] Neil Hurley and Mi Zhang. "Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation". In: *ACM Transactions on Internet Technology* 10.4 (2011), pp. 1–30. DOI: 10.1145/1944339.1944341 (cited on pages 20, 93, 99).

[IFO15] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh. "Recommendation systems: Principles, methods and evaluation". In: *Egyptian Informatics Journal* 16.3 (2015), pp. 261–273. DOI: 10.1016/j.eij.2015.06.005 (cited on pages 2, 53).

[IL16] Catherine Inibhunu and Scott Langevin. "Adaptive visualization of complex networks with focalpoint: A context aware level of details recommender system". In: *Proceedings of the Human Factors and Ergonomics Society* (2016), pp. 233–237. DOI: 10.1177/1541931213601052 (cited on pages 30, 50, 51, 65).

[Ing+14] Jóhannes Ingi Árnason et al. "Volvo intelligent news: A context aware multi modal proactive recommender system for in-vehicle use". In: *Pervasive and Mobile Computing* 14 (2014), pp. 95–111. DOI: 10.1016/j.pmcj.2014.06.005 (cited on pages 51, 66).

[JJD17] Michael Jugovac, Dietmar Jannach, and TU Dortmund. "Interacting with recommenders—overview and research". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7.3 (2017), p. 10 (cited on page 33).

[JWK14] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. "Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback". In: *ACM Transactions on Interactive Intelligent Systems* 4.2 (2014). DOI: 10.1145/2512208 (cited on pages 1, 4, 33, 76, 100).

[KA12] K. S. Kuppusamy and G. Aghila. "Semantic snippet construction for search engine results based on segment evaluation". In: *arXiv preprint arXiv:1202.2614* March 2014 (2012) (cited on page 16).

[Kap+15] Komal Kapoor et al. "" I like to explore sometimes ": Adapting to Dynamic User Novelty Preferences". In: (2015), pp. 19–26 (cited on pages 43–45, 55, 101).

[Kar13] Sahar Karimi. "A purchase decision-making process model of online consumers and its influential factor a cross sector analysis". In: *Theses PhD* (2013), pp. 1–326 (cited on pages 2, 18, 43, 45, 46).

[Kav+16] Tatenda D. Kavu et al. "A Characterisation and Framework for User-Centric Factors in Evaluation Methods for Recommender Systems". In: *International Journal of ICT Research in Africa and the Middle East* 6.1 (2016), pp. 1–16. DOI: 10.4018/ijictrame.2017010101 (cited on pages 33, 34, 96).

[Kav+18] Tatenda Kavu et al. "A Novel, Serendipitous and Dynamic User-Centric Recommender Algorithm". In: (2018). DOI: 10.4108/eai.20-6-2017.2270015 (cited on page 46).

[KB11] M Kaminskas and D Bridge. "Measuring Surprise in Recommender Systems". In: *Ir.Ii.Uam.Es* 69 (2011), pp. 107–144. DOI: 10.1007/978-0-387-85820-3_4 (cited on pages 93, 99, 103).

[KC14] P Kumar and Chalotra S. "An Efficient Recommender System using Hierarchical Clustering Algorithm". In: *Internationa Jurnal of Computer Science Trends and Technology (IJCST)* 2.4 (2014), pp. 1–6 (cited on pages 16, 70).

[KDR19] Tatenda D. Kavu, Kudakwashe Dube, and Peter G. Raeth. "Holistic User Context-Aware Recommender Algorithm". In: *Mathematical Problems in Engineering* 2019 (2019). DOI: 10.1155/2019/3965845 (cited on pages 46, 100).

[KFG15] Fateme Keikha, Mohammad Fathian, and Mohammad Reza Gholamian. "TBCA: A hybrid method based on trust and context-aware for recommender system in social networks". In: *Management Science Letters* 5.5 (2015), pp. 471–480. DOI: 10.5267/j.msl.2015.3.007 (cited on pages 2, 50, 51, 53, 65, 66).

[KG13] Wolfgang Kastner and Automation Systems Group. "Analysis of Similarity Measures in Times Series Clustering for the Discovery of Building Energy Patterns". In: (2013), pp. 579–597. DOI: 10.3390/en6020579 (cited on page 70).

[KJN11] Mohamed Koutheaïr Khribi, Mohamed Jemni, and Olfa Nasraoui. "Automatic personalization in E-learning based on recommendation systems: An overview". In: *Intelligent and Adaptive Learning Systems: Technology Enhanced Support for Learners and Teachers* May 2014 (2011), pp. 19–33. DOI: 10.4018/978-1-60960-842-2.ch002 (cited on page 13).

[KM09] Xiaoyuan Su Khoshgoftaar and Taghi M. "No Title". In: *Hindawi Publishing Corporation Advances in Artificial Intelligence* (2009) (cited on page 13).

[Kni+12] Bart P. Knijnenburg et al. "Explaining the user experience of recommender systems". In: *User Modeling and User-Adapted Interaction* 22.4-5 (2012), pp. 441–504. DOI: 10.1007/s11257-011-9118-4 (cited on pages 2, 3, 9–11, 15, 33–37, 43, 46, 68, 96, 100).

[Kon+06] Joseph A. Konstan et al. "Lessons on applying automated recommender systems to information-seeking tasks". In: *Proceedings of the National Conference on Artificial Intelligence* 2 (2006), pp. 1630–1633 (cited on page 34).

[Kor09] Yehuda Koren. "Collaborative Filtering with Temporal Dynamics". In: (2009) (cited on pages 18, 21).

[KR12] Joseph A. Konstan and John Riedl. "Recommender systems: From algorithms to user experience". In: *User Modeling and User-Adapted Interaction* 22.1-2 (2012), pp. 101–123. DOI: 10.1007/s11257-011-9112-x (cited on pages 3, 10, 11, 33–35, 37, 96, 100).

[KWK11] Bart P. Knijnenburg, Martijn C. Willemsen, and Alfred Kobsa. "A pragmatic procedure to support the user-centric evaluation of recommender systems". In: *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems* (2011), pp. 321–324. DOI: 10.1145/2043932.2043993 (cited on page 35).

[KWV16] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. "A survey of serendipity in recommender systems". In: *Knowledge-Based Systems* 111 (2016), pp. 180–192. DOI: 10.1016/j.knosys.2016.08.014 (cited on pages 44, 47, 56, 101).

[Lei14] Juha Leino. *User Factors in Recommender Systems: Case Studies in e-Commerce , News Recommending , and e-Learning*. 17. 2014 (cited on pages 2, 3, 10, 11, 13, 14, 16, 32–34, 37, 42–44, 96, 100).

[Les14] Jure Leskovec. "Mining of Massive Data Sets - Ullman". In: (2014) (cited on page 70).

[Lev+12] Justin J Levandoski et al. "LARS: A Location-Aware Recommender System". In: *2012 IEEE 28th International Conference on Data Engineering* 1 (2012), pp. 450–461. DOI: 10.1109/ICDE.2012.54 (cited on pages 17, 68, 70).

[LGS11] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. "Content-based Recommender Systems: State of the Art and Trends". In: *Recommender Systems Handbook*. Boston, MA: Springer US, 2011, pp. 73–105. DOI: 10.1007/978-0-387-85820-3_3 (cited on pages 3, 42).

[LH14] Dokyun Lee and Kartik Hosanagar. "Impact of recommender systems on sales volume and diversity". In: *35th International Conference on Information Systems "Building a Better World Through Information Systems", ICIS 2014* (2014), pp. 1–15 (cited on pages 4, 20, 32, 75).

[Li14] Lei Li. "Next Generation of Recommender Systems : Algorithms and Applications". In: (2014) (cited on pages 21, 47).

[Lit+17] Marin Litoiu et al. "What can control theory teach us about assurances in self-adaptive software systems?" In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9640 LNCS. Springer Verlag, 2017, pp. 90–134. DOI: 10.1007/978-3-319-74183-3_4 (cited on pages 51, 65, 66, 77, 102).

[Lu+12] Qiuxia Lu et al. "Serendipitous personalized ranking for top-N recommendation". In: *Proceedings - 2012 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2012* (2012), pp. 258–265. DOI: 10.1109/WI-IAT.2012.135 (cited on pages 2, 33, 100).

[Lu+15] Jie Lu et al. "Recommender system application developments: A survey". In: *Decision Support Systems* 74 (2015), pp. 12–32. DOI: 10.1016/j.dss.2015.03.008 (cited on pages 1, 45, 53, 93, 99).

[MRK06] Sean M. McNee, John Riedl, and Joseph A. Konstan. "Being accurate is not enough". In: *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06*. New York, New York, USA: ACM Press, 2006, p. 1097. DOI: 10.1145/1125451.1125659 (cited on page 35).

[MT15] Suchismit Mahapatra and Alwin Tareen. "A Cold Start Recommendation System Using Item Correlation and User Similarity". In: (2015) (cited on pages 69, 70).

[MTO07] Marta Millan, Maria Trujillo, and Edward Ortiz. "A Collaborative Recommender System Based on Asymmetric User Similarity". In: *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*. IDEAL'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 663–672 (cited on page 70).

[Mub13] Imran Saleemi Mubarak Shah, Yang Yang. "IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 1 Discovering Motion Primitives for Unsupervised Grouping and One-shot Learning of Human Actions, Gestures, and Expressions". In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* (2013) (cited on page 12).

[MZH18] Ruihui Mu, Xiaoqin Zeng, and Lixin Han. "A Survey of Recommender Systems Based on Deep Learning". In: *IEEE Access* (2018), pp. 69009–69022. DOI: 10.1109/ACCESS.2018.2880197 (cited on pages 45, 52).

[Nan13] Izak Benbasat Nanda Kumar. "Consumer of Recommendations Reviews". In: 17.4 (2013), pp. 425–439. DOI: 10.1287/isre.l060.0107 (cited on page 35).

[NT14] Thuy Ngoc Nguyen Nguyen and An Te. "No Title". In: *EAI Endorsed Transactions on Context-aware Systems and Applications* 1.1 (2014) (cited on pages 1, 51, 52, 65, 66, 102).

[Pag+16] Roberto Pagano et al. "The Contextual Turn". In: (2016), pp. 249–252. DOI: 10.1145/2959100.2959136 (cited on pages 1, 3, 4).

[PC10] Pearl Pu and Li Chen. "A user-centric evaluation framework of recommender systems". In: *CEUR Workshop Proceedings* 612.JANUARY 2011 (2010), pp. 14–21. DOI: 10.1145/2043932.2043962 (cited on pages 2, 15, 34–36).

[Pog+13] Nicolas Poggi et al. "Business Process Mining from E-Commerce Web Logs". In: Springer, Berlin, Heidelberg, 2013, pp. 65–80. DOI: 10.1007/978-3-642-40176-3_7 (cited on pages 16, 38, 42, 43).

[RAL16] Jalal Rezaeenour, Corresponding Auther, and Fatemeh Sadat Lesani. "A Context Aware Recommender System for Mobile Phone Selection Using Combination of ". In: 32.4 (2016), pp. 1203–1228 (cited on pages 2, 51, 66, 77, 102).

[Ras+12] Aishwarya Rastogi et al. "Web Mining: a Comparative Study". In: *International Journal Of Computational Engineering Research* 2.2 (2012), pp. 2250–3005 (cited on page 42).

[RV10] Pablo E. Roman and Juan D. Velasquez. "Stochastic Simulation of Web Users". In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, Aug. 2010, pp. 30–33. DOI: 10.1109/WI-IAT.2010.60 (cited on page 42).

[Sai+13] Alan Said et al. "Information Retrieval and User-Centric Recommender System Evaluation". In: *In Proceedings of the 21st Conference on User Modeling, Adaptation and Personalization (UMAP 2013)* (2013), pp. 5–8 (cited on pages 16, 30).

[San+18] Fabiano Fernandes dos Santos et al. "Latent association rule cluster based model to extract topics for classification and recommendation applications". In: *Expert Systems with Applications* 112 (2018), pp. 34–60. DOI: 10.1016/j.eswa.2018.06.021 (cited on page 50).

[Sax+09] Abhinav Saxena et al. "Evaluating algorithm performance metrics tailored for prognostics". In: *2009 IEEE Aerospace conference*. IEEE, Mar. 2009, pp. 1–13. DOI: 10.1109/AERO.2009.4839666 (cited on page 34).

[SG11] Guy Shani and Asela Gunawardana. "Evaluating Recommendation Systems". In: *Recommender Systems Handbook* (2011), pp. 257–297. DOI: 10.1007/978-0-387-85820-3_8 (cited on pages 12–15, 42, 97).

[SM95] Upendra Shardanand and Pattie Maes. "Social information filtering". In: (1995), pp. 210–217. DOI: 10.1145/223904.223931 (cited on page 35).

[SMG09] Guy Shani, Christopher Meek, and Asela Gunawardana. "Hierarchical Probabilistic Segmentation of Discrete Events". In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE, Dec. 2009, pp. 974–979. DOI: 10.1109/ICDM.2009.87 (cited on pages 13, 42, 44, 97).

[SNC16] Sridhar Dilip Sondur, Shantharam Nayak, and Amit P Chigadani. "Similarity Measures for Recommender Systems: A Comparative Study". In: *International Journal for Scientific Research and Development* 2.3 (2016), pp. 76–80 (cited on page 70).

[SPA15] Ranveer Singh, Bidyut Kr. Patra, and Bibhas Adhikari. "A Complex Network Approach for Collaborative Recommendation". In: (2015), pp. 1–22. arXiv: 1510.00585 (cited on page 70).

[Sri14] Suboojitha Sridharan. "Introducing serendipity in recommender systems through collaborative methods". In: *ProQuest Dissertations and Theses* (2014), p. 66 (cited on page 20).

[SS01] K. Swearingen and R. Sinha. "Beyond Algorithms : An HCI Perspective on Recommender Systems". In: *ACM SIGIR 2001 Workshop on Recommender Systems (2001)* (2001), pp. 1–11. DOI: 10.1.1.23.9764 (cited on page 35).

[Sun+16] Camila Vaccari Sundermann et al. "Privileged contextual information for context-aware recommender systems". In: *Expert Systems with Applications* 57 (2016), pp. 139–158. DOI: 10.1016/j.eswa.2016.03.036 (cited on pages 2, 30, 50, 52, 53, 77, 102).

[Tin16] Nava Tintarev. "Joint Workshop on Interfaces and Human Decision Making for Recommender Systems". In: (2016) (cited on pages 3, 10, 33, 35, 43, 100).

[Ung+16] Moshe Unger et al. "Towards latent context-aware recommendation systems". In: *Knowledge-Based Systems* 104 (2016), pp. 165–178. DOI: 10.1016/j.knosys.2016.04.020 (cited on pages 2, 4, 32, 52, 53, 75, 102).

[Var15] Sa´ ul Vargas. "NOVELTY AND DIVERSITY EVALUATION AND ENHANCEMENT IN RECOMMENDER SYSTEMS". PhD thesis. 2015 (cited on pages 1, 3, 4, 19, 21, 33, 42, 76, 93, 99, 100).

[Var16] Sa´ul Vargas. "New Approaches to Diversity and Novelty in Recommender Systems". In: (2016), pp. 8–13 (cited on pages 93, 99).

[VC13] Saúl Vargas and Pablo Castells. "Exploiting the Diversity of User Preferences for Recommendation". In: *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval* (2013), pp. 129–136 (cited on pages 19, 21).

[Ver+12] Katrien Verbert et al. "Context-aware recommender systems for learning: A survey and future challenges". In: *IEEE Transactions on Learning Technologies* 5.4 (2012), pp. 318–335. DOI: 10.1109/TLT.2012.11 (cited on page 4).

[VP11] J Vellingiri and S.Chenthur Pandian. "A Survey on Web Usage Mining". In: *Global Journal of Computer Science and Technology* 11.4 (2011), pp. 67–72 (cited on page 16).

[Wac11] Elmar P. Wach. "Automated ontology evolution as a basis for adaptive interactive systems". In: *CEUR Workshop Proceedings* 747 (2011) (cited on page 43).

[WN12] Bart P. Knijnenburg · Martijn C. Willemsen and Zeno Gantner · Hakan Soncu · Chris Newell. "Explaining the user experience of recommender systems". In: *User Model User-Adap Inter: Springer* 22 (2012), pp. 441–504 (cited on page 100).

[Yan+14] Xiwang Yang et al. "A survey of collaborative filtering based social recommender systems". In: *Computer Communications* 41 (2014), pp. 1–10. DOI: 10.1016/j.comcom.2013.06.009 (cited on pages 1, 2, 45, 54).

[Yan+15] Chao Yang et al. "Using Personal Traits For Brand Preference Prediction". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), pp. 86–96 (cited on pages 26, 67).

[Yan+16] Zhe Yang et al. "A survey of collaborative filtering-based recommender systems for mobile internet applications". In: *IEEE Access* 4 (2016), pp. 3273–3287. DOI: 10.1109/ACCESS.2016.2573314 (cited on pages 1, 52).

[YM17] Raciel Yera and Luis Martínez. "Fuzzy Tools in Recommender Systems: A Survey". In: *International Journal of Computational Intelligence Systems* 10.1 (2017), p. 776. DOI: 10.2991/ijcis.2017.10.1.52 (cited on page 1).

[Zha+11] Lei Zhang et al. "Combining lexicon-based and learning-based methods for twitter sentiment analysis". In: *HP Laboratories Technical Report* 89 (2011) (cited on page 99).

[Zha+15] Heng Ru Zhang et al. "A hybrid recommender system based on user-recommender interaction". In: *Mathematical Problems in Engineering* 2015.1 (2015). DOI: 10.1155/2015/145636 (cited on pages 93, 99).

[Zha+18] Rong Zhang et al. "How does serendipity affect diversity in recommender systems ? A serendipity-oriented greedy algorithm". In: *Expert Systems with Applications*. IDEAL'07 1.4 (Mar. 2018). Ed. by Francesco Ricci ·Lior Rokach ·Bracha Shapira · and Paul B Kantor, pp. 217–250. DOI: 10.1007/s00607-018-0687-5. arXiv: 1802.08452 (cited on pages 45, 47, 93, 99).

[Zha13] Liang Zhang. "The definition of novelty in recommendation system". In: *Journal of Engineering Science and Technology Review* 6.3 (2013), pp. 141–145 (cited on pages 1, 20, 91, 93, 99).

[ZKS18] Michele Zanitti, Sokol Kosta, and Jannick Sørensen. "A User-Centric Diversity by Design Recommender System for the Movie Application Domain". In: (2018), pp. 1381–1389. DOI: 10.1145/3184558.3191580 (cited on page 99).