

Academia Recommender System

Android app

*Report submitted in fulfillment of the requirements
for the Exploratory Project of*

Second Year B.Tech.

By

Kumar Shivam Ranjan (18075031)
Asht Bhuja Prasad Rastogi (18075013)

Under the guidance of
Dr. Sukomal Pal



Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI Varanasi 221005, India
May 2020

Dedicated to:

My parents, professor, and everyone who helped and motivated me a lot in the successful completion of this report.

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.
2. The work has not been submitted for any project.
3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving the required details in the references.

Place: IIT (BHU) Varanasi

Date:- 05-06-2018

1.Kumar Shivam Ranjan

2.Ashta bhuja Prasad Rastogi

(B. Tech Students)

Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

This is to certify that the work contained in this report entitled

“Academia Recommender System”

being submitted by

Kumar Shivam Ranjan (18075031)

Asht Bhuj Prasad Rastogi (18075013)

carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi is a bona fide work of our supervision

Dr. Sukomal Pal

Place: IIT (BHU) Varanasi

Date: 05-06-2018

Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

I would like to express my sincere gratitude to our respected **Dr.Sukomal Pal**, who was my mentor on an exploratory project. He helped us a lot at every stage during my project to clear my doubts. I also wanted to express my sincere gratitude to tutors(Ph.D. scholar) **Mr. Trivikram** sir who helped me to find related materials to study, getting ideas about our topic, and in building our model.

Abstract

There has always been a vast amount of data available on the internet.

The users every second need an immense variety of data and information and the handiest device to showcase the result is iOS or Android app.

There has been an information overload and every consumer using either an android app or web app has to go through tedious searching and filtering of information to get to his/her preferred results.

Why is the Android/iOS app handier for information retrieval?

Because it requires minimal efforts to get to our results and it is within the reach of almost everyone. So, we just have to design a system which will fetch out the most obvious results based on the user's preferences or most recent searches.

The main objective of this report and the project is to present the most desirable results to the user with the minimum overload of information.

That is, the person doesn't have to spend too much time looking for the searches which are most suitable for him/her. Our App will do the same.

This problem was divided into sub-problems, namely:-

- 1) Collecting and storing application metadata
- 2) Collecting and storing user consumption of searches
- 3) Filtering out less probable results and presenting the most desirable ones to the user.

So, we build a recommender system integrated into an android app that will filter out the most preferred journals, articles, or reports for specific users. The user doesn't have to search for his/her queries every time. The app will show suggestions of the most preferred journals for him/her.

Contents:

List of Figures.	viii
1. Introduction.	viii
1.1 Overview	ix
1.2 Motivation of the Research Work	xi
1.3 Organisation of the Report	xii
2 Project Work	xiii
3 Conclusions and Discussion.	xxii
4 Bibliography.	xxiii

List of figures:

1. Activities GUI
2. Android app recommendation process sequence
3. Recommendation process in mobile environments
4. Different types of Recommender systems algorithms.

Chapter 1

Introduction

1.1 Overview

The main goal of this project is to build an android system that helps users in finding articles, journals, newspaper reports, etc of the most preferred choice for him/her. Initially, the system doesn't know the user's preferences. The system collects application consumption data including his/her most recent searches to base future studies on. This system has 2 recommender systems and other filtering solutions to the backend of the android API.

The two recommender system used in our project are-

- 1) BARC recommender system
- 2) Venue recommender system

The aim of the initial study for this project was to explore the android world and learn how to design attractive GUI and how to move from one activity to another passing data if required between them.

After that, It was to understand and explore the field of recommender systems in general and how they work and how they can work as a backend to the android app. Our Android application has a launcher activity as a Login page which lets users authenticate themselves. If a user doesn't already have an account, he/she can go to the Registration page to enter the dashboard. Then follow various activities that are user-friendly and help users to see their preferred choice without even manually searching for them.

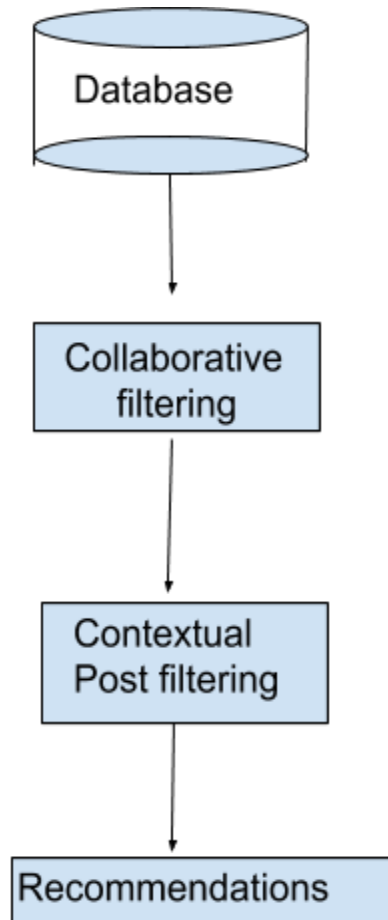


Fig 1.1 Android app Recommendation Process

Our android system maintains a database where the authenticated users have interacted with items(journals, articles, etc) by reading them. This information or data is crucial for the recommendation system because there must be some sort of interaction between users and items for the recommender system to work. The recommender system uses this piece of data to recommend stuff. The recommender system then applies a collaborative filtering algorithm to the data and as a result, comes up with a list of top K recommendations. The user will input:

- Title
- Abstract

Further, the contextual information is sent from the mobile device to the server so the irrelevant recommendations according to the current context are not recommended. The recommendations not fit for current context such as location company, are simply removed [1].

1.2 Motivation of Research Work

The field of e-commerce has widely been affected by recommender system projects in recent years. In fact, these e-commerce business giants mainly work on their recommender system protocol. If their recommender system protocol is good enough, they can influence the international market to a great extent. E-commerce giants like Netflix , Youtube,Amazon Prime, etc have recommender systems in their backend. They use very popular recommendation libraries including Apache and Mahout.

In order to recommend items to users, different systems use different algorithms. It depends upon the system's design and ease to handle algorithms. We have used two recommender systems in our project. That itself is enough to show their great advancements, popularity, and scope for research work in this area.

It aims to solve search-related problems in a way the information overload problem found in the World Wide Web is reduced to a drastic extent.

The recommender system adapts to each user in a personalized or a specific way to give suggestions, which comes out as a result of previous preferences and current interests. The provided recommendations are suggestions of items or services that could be of potential interest to the user.

The recommendations are of various kinds. For example, books, movies, songs, shopping-related stuff, etc. The recommendation of "People you may know" in social networks like Facebook also uses the fancy recommender system.

The recommender system adapts to each user in a personalized way to provide recommendations, which is usually done through previous preferences, current interests, or a combination of methods. The provided recommendations are suggestions of items or services that could be of potential interest to the user.

In our android app, we have a recommender system that recommends the journal,articles,etc which is most suitable for the user, that's why the name Academia recommender system.

1.3 Organisation of the Report

- App Development
- Device Compatibilities
- GUI components and various activities
- Recommender system

Chapter 2

Project Work

2.1.App Development

We built an android app that recommends users their preferred choice of articles, journals, etc. The app was built using a software solution for Android development (Android Development Tools (ADT) Bundle) which integrates a collection of various programs.

- Eclipse is an integrated environment for the development of software projects with Multi-Language support.
- ADT plug-in: It is the toolset for Eclipse designed to allow the development of Android Apps.
- Android Software Development Kit (SDK): provides the API libraries and developer tools required to build apps for android.
- Android Native Development Kit (NDK): is the collection of tools that allows implementing apps using native-code languages such as C and C++.
- The code was written using two languages, namely, kotlin and Python. The former was employed for the development of the graphical user interface and app basic controls. Python was reserved for computationally expensive tasks such as collecting meta-data, running trains, and testing on data and developing an algorithm that will eventually recommend users their preferred choices.

2.2 Device compatibility

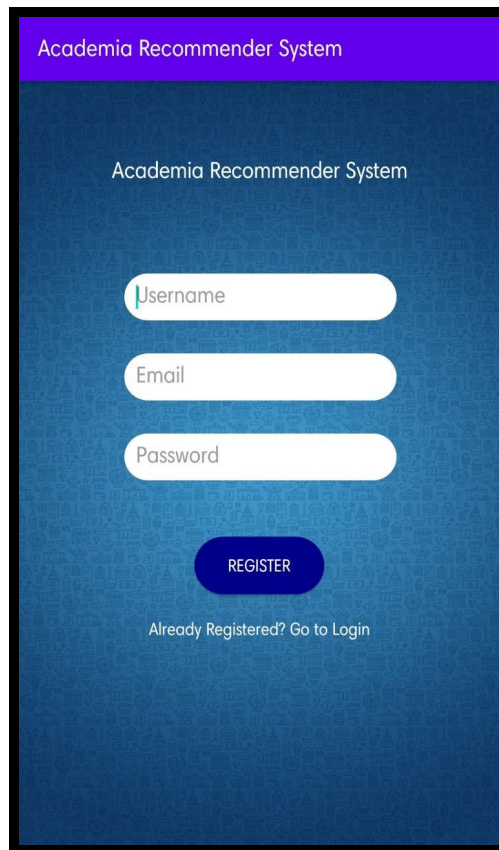
The plugins used in our android app are:

- Com.android.application
- Kotlin-android
- Kotlin-android-extensions
- com.google.gms.google-services

- 1) Compiled SDK version : Version 29
- 2) buildToolsVersion "29.0.3"
- 3) minSdkVersion 16
- 4) targetSdkVersion 29
- 5) Ext.kotlin_version 1.3.72

2.3 GUI components and various activities of the project

We first began by creating a launcher activity which is our app's Register page. It is very necessary to authenticate users in order to provide them the required services. Our app doesn't allow anonymous users to use the services of the recommender system. Users will have to explicitly provide their username and email and set a password in order to register for the app. Here is how the UI looks.



Academia Recommender System

Academia Recommender System

Username

Email

Password

REGISTER

Already Registered? Go to Login

Fig1. Register Activity (a)

The image shows a mobile application interface for the 'Academia Recommender System'. At the top is a purple header bar with the text 'Academia Recommender System'. Below this, on a dark blue background with a subtle geometric pattern, is the title 'Academia Recommender System' in white. There are three white rounded rectangular input fields: the first contains 'alice102', the second contains 'alice102@gmail.com', and the third contains three dots '...'. Below these fields is a dark blue rounded rectangular button with the word 'REGISTER' in white. Under the button is the text 'Already Registered? Go to Login'. At the bottom, a grey rounded rectangular box contains an error message: 'Failed to create user: The given password is invalid. [Password should be at least 6 characters]'.

Fig2. Register Activity (b)

Not to mention that this activity is enriched with various types of validations for users like the length of the password and any of the username password or email cannot be empty

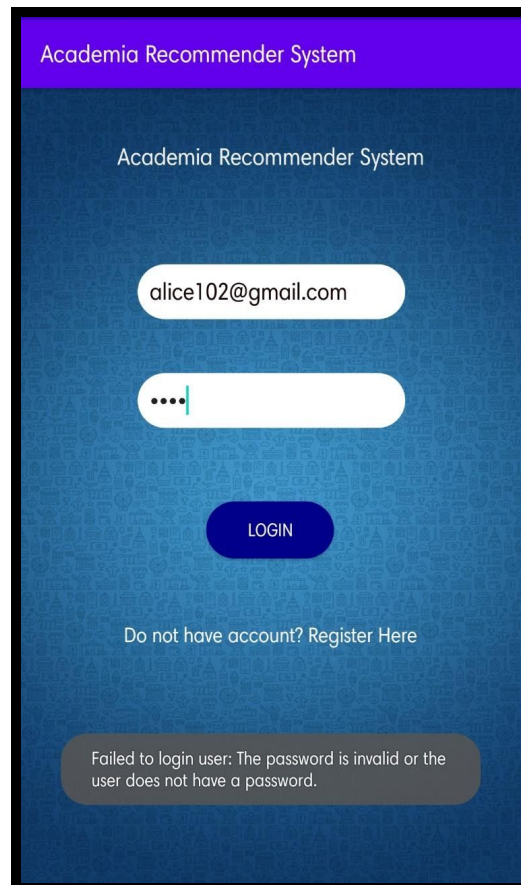


Fig3 Login Activity

We then proceed to build the login activity which lets the authenticated users enter our app and use the services of the Recommender system API.

For building the Registration UI, we incorporated Google's firebase authentication as a backend to store all the user's information.

Creator of the app needs to set up their own [Firebase](#) authenticator + database and download the `google-services.json` file, then add it to the **App/app** folder.

The dependencies are already set up in our Academia Recommender system App

```
dependencies {  
    implementation fileTree(dir: "libs", include: ["*.jar"])  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.1.0'  
    implementation 'androidx.appcompat:appcompat:1.1.0'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'  
    implementation 'com.google.firebase:firebase-analytics:17.2.2'  
    implementation 'com.google.firebase:firebase-auth-ktx:19.3.1'  
}
```

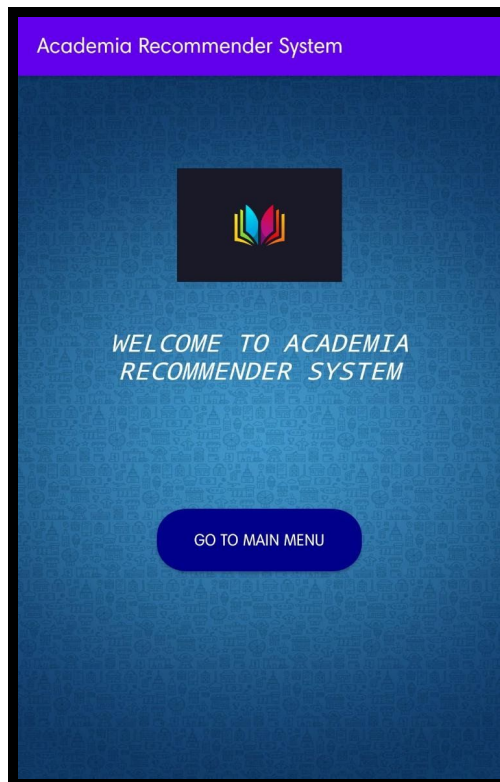


Fig4 Main Activity

This is the first page the user sees after logging in. Attached to it is the dashboard activity that provides some filtered options to the users.

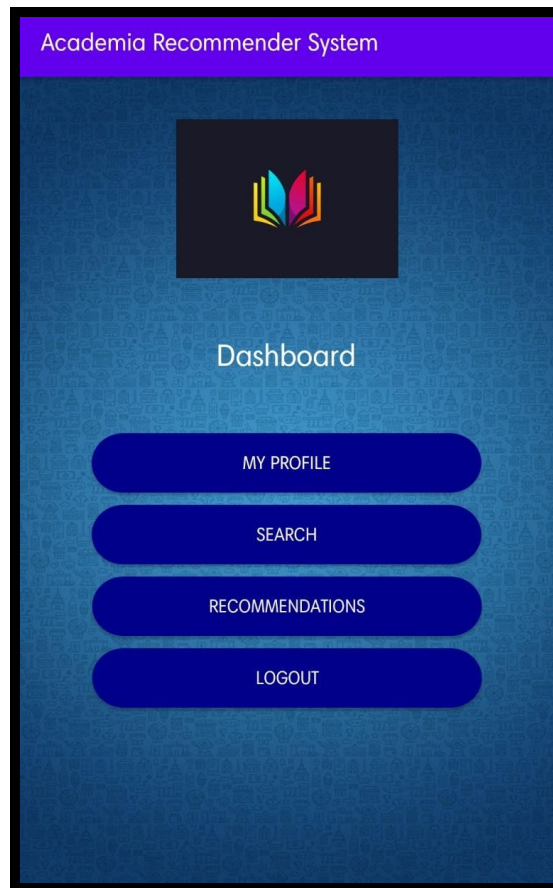


Fig5 Dashboard Activity

The Dashboard Activity provides the user's main user-friendly GUI to users looking for the most preferred journals for them.

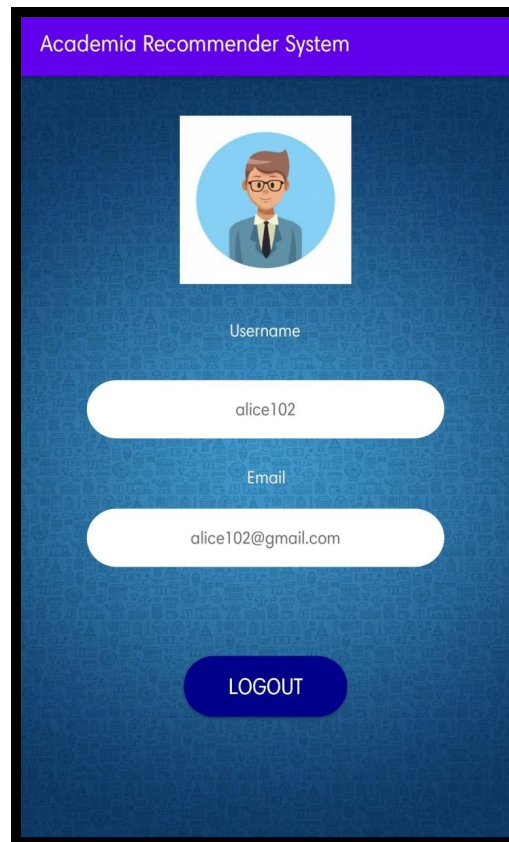


Fig6 Profile Activity

If the user wants to check out his/her profile, he/she can do it anytime while he/she is logged in by clicking the My Profile button in the dashboard activity. This ensures to some extent of transparency in our android app.

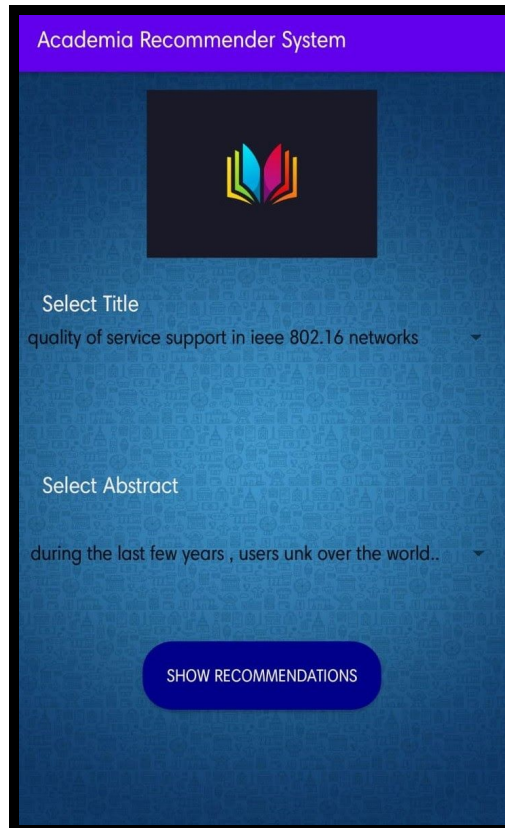


Fig7 Recommendation Activity(a)

For the recommendations, we have to provide the values for the fields TITLE and ABSTRACT. It will be through a dropdown option of androidx library.

The androidx library provides various GUI components. Here, we have used the spinner component to show the drop-down list. Spinner component along with the ArrayAdapter class of androidx stores the default data for users. Users can select one of them.

The values will be stored for a processing purpose but before that when the user selects an option, there will be a Toast showing which option the user has selected.

It will look something like this

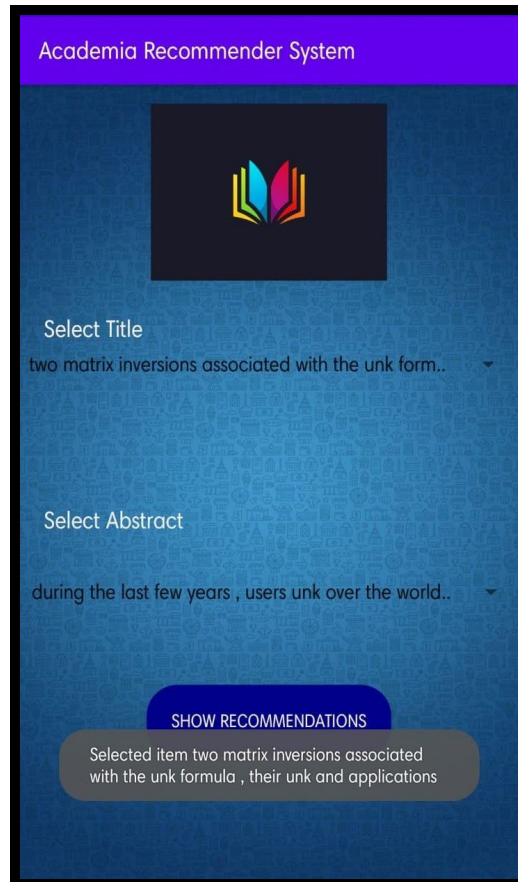


Fig7 Recommendation Activity(b)

2.4 Integration of app with recommender systems

Academia Recommender System	
TOP 15 Results	
IEEE Communications Magazine	
IEICE - Transactions on Information and Systems	
Pattern Recognition Letters	
Computer Networks: The International Journal of Computer and Telecommunications Networking	
Mobile Networks and Applications	
Computer Communications	
Journal of Computing Sciences in Colleges	
IEICE - Transactions on Information and Systems	
Computer Networks: The International Journal of Computer and Telecommunications Networking	
IEEE Internet Computing	
MILCOM'06 Proceedings of the 2006 IEEE conference on Military communications	
GLOBECOM'09 Proceedings of the 28th IEEE conference on Global telecommunications	
IEEE Transactions on Mobile Computing	
Computer Networks: The International Journal of Computer and Telecommunications Networking	
IEEE Transactions on Robotics	

Fig 8 Result Activity

The Recommender system will filter out 15 best results according to the input of the user. Our android app needs to be integrated with the recommender systems in order to fetch results to the users. We need to have an API ready with the specific recommender algorithms implemented on it. (the engine). We can access this API from our android app. We can do this by sending and receiving HTTP requests to the endpoint.

There are two ways to get access to the implemented recommender engine by using HTTP requests:

1. We can build the recommender system ourselves and thereby implement and host our engine.
2. We can use the Abracadabra Recommender API. we only need to send HTTP calls to the API. This will train our models and as a result of that, will fetch us the desired results.

The GUI input of an android app needs to be converted in a JSON file. Upon that JSON file, we run our recommender system model and save the results in the JSON file as well. The result JSON file needs to be converted to android GUI to further display the results to the user.

2.5 Recommender system

Recommender systems are a great tool in today's era where such a vast information overload is there and it's increasing every single day. It provides the most preferred and accurate suggestions to users. They tackle the problem of information overload in the most efficient way.

Many e-commerce giants' revenue is largely affected due to the quality of their recommender systems. They interact with users largely by their recommender system programs. These e-commerce giants' are Amazon, Netflix, Flipkart, etc.

This prediction is computed using a number of predictive models that have a common characteristic. For example, they make use of ratings provided by a user's previously viewed or purchased items in case of a movie or item recommender in Amazon. Recommender systems differ from each other on the basis of techniques of algorithms they use for recommendations.

Recommender systems, depending on the method they employ, can be classified as follows[5].

1. **Content-based Recommendations:** This technique uses a method that first precomputes the item's description and characteristics. For a movie recommender system, this method first precomputes the feature vector for each movie which shows up to what extent a movie is romantic, action, drama, etc. A number of features equals a number of genres like romance, action, drama, etc. Formally, the algorithms used in this recommender system recommends those items that are similar to the items, which have been previously viewed by the user[5]. However, this algorithm will fail if the user is new, i.e. the user hasn't viewed any items yet.
2. **Collaborative filtering:** This technique first collects a user's preferences. They collect the data regarding the user's feedback before recommending items to them. More formally, this algorithm recommends those items which have been approved by users who have similar taste. The movies with the highest predicted ratings and which the user has not yet seen or rated yet would then be recommended to the users. This algorithm suffers from a new item problem that causes new items to be ignored. They are not recommended until they receive some ratings from the user.
3. **Knowledge-based.** These recommenders use techniques that help the system to suggest items to users using two things:

- Inferences about the preferences of users
- Utilizing specific domain knowledge.

4. Hybrid: Both the above-mentioned algorithms have some drawbacks. One suffers from a new item problem and the other from a new user. So, Hybrid techniques merge the functionalities of both the problems to predict stuff for the user.

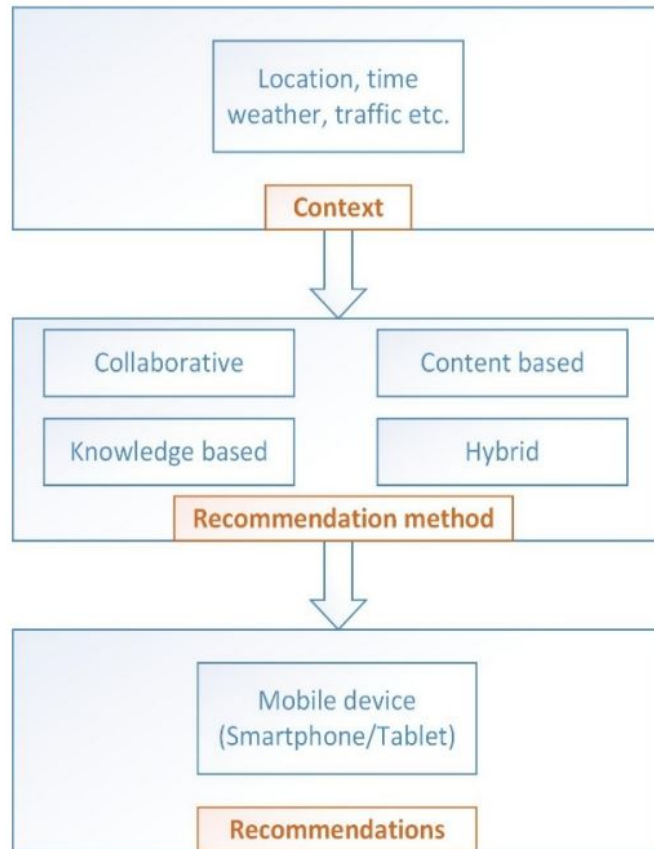


Fig. Recommendation process in mobile environments

3. Conclusion and Discussion

The main work and aim of this project were to build a mobile recommender system and conclude that there still exists a wide gap between recommender systems and mobile computations whether it is android or iOS which needs to be filled to provide much more convenient way to users to filter information and to avoid information overload. We first built an android GUI which was user friendly and helped users to fetch as much information as possible for the recommender system to use that information to predict things.

Android GUI worked as a way to collect information from the users. The connected backend recommender system then applies various algorithms like content-based filtering or collaborative filtering to suggest users the most preferred choices. One future research direction in this field is the delivery of a complete framework for mobile recommender systems which aims at bridging the gap between android /iOs computing and recommender systems.

However, the mobile recommender systems do suffer from some serious drawbacks. One of the most important ones is the user's privacy.

Sometimes, excessive personalization or continuous monitoring of their actions can cause a feeling of concern or discomfort for the users even though personalization of the application might seem useful to the users.

It is still possible to thoroughly gather the contextual information and further recommendations can take into consideration other contexts like users' battery level, mobile compatibility, or the users' network signal.

Bibliography:

- [1] (Kato et al., 2012; Kido, Yanagisawa, & Satoh, 2005; Polatidis, Georgiadis, Pimenidis, & Stiakakis, 2017).
- [2] (Košir, A., Odic, A., Kunaver, M., Tkalcic, M., & Tasic, 2011)
- [3] Ricci F., Rokach L., Shapira B., Kantor P. B., Recommender Systems Handbook, Springer-Verlag New York, 1st edition, 2010.
- [4] Ricci F., Mobile Recommender Systems, Faculty of Computer Science, Free University of Bolzano, Italy, 11.1.2010.
- [5] (Bobadilla et al., 2013; Jannach et al., 2010; Shi, Larson, & Hanjalic, 2014; Su & Khoshgoftaar, 2009):