Abderrahmane Mira Bejaia University
Faculty of Exact Sciences
Computer Science Department

Thesis to obtain the Master of Computer Science
Degree in software engineering

MORES : A Movie Recommendation System

Thesis of Mrs :

ADJALI Naziha Fatma
AKROUCHE Wassila

Examination Committee :

Chairperson :	Prof. BOUKERRAM Samira
Member of the Committee :	Prof. GHANEM Souhila
Supervisor :	Prof. EL BOUHISSI Houda (Mrs.BRAHAMI)

September 2020

Abstract

With the increasing amount of data content produced daily, it becomes very difficult for users to find the resources suitable to their needs. Recommendation systems are proposed to solve this problem and are capable of providing personalized recommendations or guiding the user to interesting or useful resources within a large data space. Recently, Recommender systems are getting importance due to their significance in making decisions and providing detailed information about the required product or a service. In this paper, we conduct a systematic review for recommendation models, and discuss the challenges and open issues. Furthermore, we propose a new recommendation system ontology-based in which machine-learning algorithms are used to achieve user needs identification and provide precise recommendations.

Résumé

De nos jours, le volume de données produit quotidiennement ne cesse d'accroitre, il devient par conséquent très difficile pour les utilisateurs de trouver les ressources adaptées à leurs besoins. Des systèmes de recommandation sont proposés pour résoudre ce problème et sont capables de fournir des recommandations personnalisées ou de guider l'utilisateur vers des contenus susceptibles de l'intéresser. Dans ce mémoire, nous effectuons une revue systématique des modèles de recommandation et discutons des avantages et des défis à relever dans le domaine de la recommandation. De plus, nous proposons un nouveau système de recommandation basé sur une ontologie dans lequel des algorithmes d'apprentissage automatique sont utilisés pour identifier les besoins des utilisateurs et fournir des recommandations précises.

Table of contents

List of Figures

List of Tables

Nomenclature

AI Artificial Intelligence, page 20

CART Classification and Regression Tree, page 25

CBF Content-Based Filtering, page 2

CF Collaborative Filtering, page 2

CHAID Chi-squared Automatic Interaction Detection, page 25

DF Demographic Filtering, page 11

HCI Human-Computer Interaction, page 7

HF Hybrid Based, page 11

ID3 Iterative Dichotomiser 3, page 25

IR Information Retrieval, page 7

IRS Intelligent Recommender System, page 29

KBF Knowledge Based Filtering, page 11

KNN K Nearest Neighbors, page 27

ML Machine Learning, page 7

OPCR Ontology-based Personalized Course Recommendation, page 26

RS Recommendation Systems, page 2

UML Unified Modeling Language, page 37

Acknowledgements

We are very pleased to provide this written acknowledgement of our gratitude to all those who have given us their support and trust throughout our studies.

Firstly, we are very grateful to our director of research, Professor ELBOUHISSI Houda, who guided us throughout the realization of this thesis. We thank you for your great availability, your listening and your patience. Your insightful feedback pushed us to sharpen our thinking and brought our work to a higher level. Thank you for everything !

We would like to thank the members of the examination committe, the honor they had done us by accepting to be the corrector of our dissertation. We thank you for agreeing to give your precious time to evaluate and criticize our work.

Finally, we thank our families, and especially our parents for their encouragement and support, which has allowed us to never give up.

To our dear parents for their sacrifice and support throughout our lives,

To our dear brothers and sisters who we wish success and happiness,

To all the people who have supported us during this project,

And to all those whose names have not been mentioned but who are no less dear.

THANK YOU

Chapitre I

Introduction

> *Artificial Intelligence, deep learning, machine learning –whatever you're doing if you don't understand it– learn it. Because otherwise you're going to be a dinosaur within 3 years.*
>
> Mark Cuban, *Upfront Summit* , 2017

This chapter discusses the motivations that led us to the choice of this theme and to the resolution of the problems related to it, as well as the main objectives that this research tends to achieve. Following this, the contributions of this research work are highlighted. Finally, the organization of the thesis is explained.

I.1 Motivations

With the increasing number of people contributing on the Internet consciously or inadvertently, a huge set of data is available giving insights into personal tastes, marketing and human behaviour. The ability to collect all of these information and the computational power to process and interpret it led to a better understanding of user needs.

As a matter of fact, a conventional information system is not able to provide a good user experience since only a few of the items are within the interest of the user. This is where machine learning and statistical methods become more important. These methods are used widely in interpreting and arranging the large amount of data that is displayed all over the web [1].

In addition, these data suffer from heterogeneity. As a result, the ability to target relevant information to the user remains at the heart of a significant amount of research. Customization is a suitable solution to this problem. Existing customization tools and adaptation services include the recommendation systems.

Recommendation Systems (RS) have been in full expansion since the early 1990s. They offer to users the possibility of choosing better from the different content available in information systems. The most representative example is that of the Amazon.com site. It offers products to users with the famous phrase : "Customers who bought this item also bought. . . ".

Due to their usefulness, recommendation systems have gained several areas : e-commerce sites, scientific publications, press and platforms movie rentals. Their issues have therefore become considerable, not only from the economical point of view for commercial systems, but also, more generally, from the point of view of user satisfaction. Indeed, the user represents the heart of such systems. Understanding and anticipating their expectations and needs is essential to obtain their satisfaction.

To generate recommendations, a number of approaches have been identified, where the most used are Collaborative Filtering (CF) and Content-Based Filtering (CBF) . Both techniques have their strengths and weaknesses, where hybridization between them has been quickly adopted to take advantage of their benefits [2].

One of the major problems with recommendation systems is the problem of the stability of these systems in relation to the dynamic profile of the user (Stability vs. Dynamicity Problem) [2]. This problem comes back to the fact that if the user is interested in several different items at the same time, as he can alternate his preferences over time, and if his profile is created in the system, it becomes difficult to change his preferences and take into consideration their different choices and preferences. This limits the ability of recommendation systems to follow the evolution of the user's profile and to adapt to their different choices and preferences, and subsequently to recommend items that do not correspond to their different choices and interests, which leads to a lack of diversity in the recommendation lists. In addition, recommendation systems, especially those using Collaborative Filtering suffer from the problem of scalability [2], when adding a new user or a new item, and also suffer from the problem of sparse data [2], due to the large user-item matrices containing scattered evaluations.

The objective of the Collaborative Filtering method is to generate suggestions of items to users using the preferences of their neighbors, based on methods and metrics to group users and find all the relevant neighborhoods through the use of similarity measures, such as the Pearson Correlation measure and the Cosine Correlation measure. But the classical similarity measures calculate the similarity between the active user and the set of users in the system without considering their ambiguous and dispersed preferences, which requires enormous computation times and leads to having an irrelevant set of neighbors. A natural idea is to select the nearest neighbors from similar groups using effective similarity measures to reduce calculations.

Another challenge that recommendation systems suffer from is the so-called cold start problem. The cold-start problem typically happens when the system does not have any form of data on new users and on new items. It is difficult to recommend items to a user we have no information about his tastes as it is difficult to recommend an item that is not rated by anyone to other users.

However, in recent years, new qualities of a good recommendation system have been presented in the literature, in addition to the performance of predictions. An effective recommendation system must offer new and diverse items to users, which meet their different interests and preferences, which requires the development of new ideas and techniques to formulate recommendations of interest. Thus, the interesting recommendations should contain various and relevant items taking into consideration the performance of the system.

All these facts provided the motivation to propose a novel approach in building a recommendation system to overcome the problems of information overload, data sparsity and cold start.

I.2 Aims and objectives

The purpose of this thesis is to present a new recommendation system that considers the points highlighted above in order to deal with them in the best possible way.

The main objectives are listed in the following :

1. In order to have a global overview on the different recommendation systems that have been built and the tools that have been used, the study of the state of the art of recommender systems is our starting point. We did not focused on one domain where recommender systems are used but gave a more generalist research to assimilate the concept and its difficulties.

2. The importance of having a good user profile in the system is relevant for the process of recommendation. We focus on the building of a good user profile in order to fix the problem of cold start and provide a better user experience.
3. The hybridization : Combining several approaches to recommending to improve the system performances.
4. The use of semantic web for a better formalisation of data and a better personalisation of the recommendations.

I.3 Major contribution

In recommendation, it may be interesting to place users (or items) in a specific context in order to extract more information and thus obtain a better prediction of preferences. The objective of this thesis is to propose a new approach of movie recommendation by exploiting semantic web technologies to overcome the problems cited before (cold start, sparsity, information overload). Our major contribution consist in generating a profile ontology for the user and match it with a domain ontology in order to enrich his information saved in the database and by that enhance the recommendation process. The domain ontology also contributes in improving the diversity of the recommendations.

I.4 Thesis organisation

The research work is organised as follows :

- Chapter 2 gives an overview of recommendation systems. This overview presents the history of recommendation systems followed by some basic concepts and notation related to the context of recommendation and its definition. Then, the presentation of several well-known classification logics in this field and recommendation techniques provides valuable insights into the functioning of these systems. After that, we briefly summarize the steps to follow in order to build a simple recommendation system. Chapter 2 ends with citing the challenges encountered by recommendation systems.

- Chapter 3 discusses background details and related work and research on recommendation systems and the aspect of ontology. It also highlights different recommendation algorithms and the main challenges faced by general recommender systems. Attention is mainly focused on three aspects : machine learning based recommender systems, ontology based and hybrid based systems. The strengths and weaknesses of each system are being pointed out and analysis is done in order to identify the challenges that need to be solved in this area.

- Chapter 4 presents our solution to meet the needs presented in Chapter 3. We describe the system architecture and the implementation of the different concepts. We also define the tools used in order to build the system
- Chapter 5 presents the overall validation of our system, as well as the results of tests and comparisons of the different types of algorithms used. All this after having presented the technological aspects surrounding the implementation of our system as well as the description of the different interfaces of the application.
- The last chapter concludes this thesis and presents some future perspectives.

Finally, the thesis includes the bibliographic references used for its elaboration.

Chapitre II

Recommendation Systems

II.1 Introduction

With the advent of the Internet, we are now witnessing an information overload, which makes the selection of the most interesting information according to the interests of each user a very difficult task, therefore recommendation systems have emerged to remedy this problem.

Moreover, with the rise of YouTube, Amazon, Netflix and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

The recommendation systems are software components whose purpose is to provide users with information that corresponds to their interests and this by analyzing their interactions with their information space.

These recommendation systems make predictions for users with the objective of presenting them with only those elements by which they will be attracted. Thus, with recommendation systems, the Internet is no longer neutral, it is now adapted to everyone.

In this chapter, we start by establishing an overview of recommendation systems' history. Then, we define the most important concepts related to the context and continue with some RS definitions. After that, we give different types of classification that describes the evolution of these systems. Then, we enumerate the recommendation techniques and present the different steps for building a recommendation system. Finally, we conclude the challenges that recommendation systems face.

II.2 History

The purpose of recommendation systems is to provide active users with recommendations of items that are potentially of their interest. These recommendations may concern an article to read, a book to order, a movie to watch, etc.
The roots of recommendation systems can be traced in the extensive work in the cognitive sciences, the theory of approximation, the literature search, the theory of foresight, and also have links to the science of management and marketing in the modelling of consumer choice [3].
The field of research on recommendation systems emerged in the early 1990s and is reduced to collaborative filtering systems. Since then, particularly with the integration of social networks, artificial learning and big data, this field has been in constant evolution. Among the pioneering systems in this field, we cite the first Tapestry introduced by Goldberg for the recommendation of newsgroup messages. Two years later, researchers at Grouplens presented their first RS for the recommendation of Usenet articles in parallel with the Ringo system [4] for the recommendation of music.
The first hybrid recommendation system is created in 1997 by Balabanovic and Shoham [5], which combines content-based and collaborative filtering.
In recent years, recommendation systems have become a topic of growing interest in the fields of Human-Computer Interaction (HCI) , Machine Learning (ML) and Information Retrieval (IR) , and have become an essential component of most e-commerce sites.

II.3 Basic concepts and notations

In this section, we define some concepts related to recommendation systems.

II.3.1 User and Items entities

In every recommendation system, there are two important entities : users and Items.

- **Users :** Is a person who accesses the system and registers by entering his personal informations (interests, age. . .). The set of users in the system is represented by U, where a user is u∈U.
- **Items :** In RS, an item is the entity that represents any item that constitutes a recommendation list and that corresponds to the user's needs, including any product that may be sold (books, products, etc. in e-commerce sites such as Amazon.com), seen (movies in online TV sites such as Netflix), listened to (music) or read (such as information in online newspapers, magazines in digital libraries), as well as vacation destinations, restaurants, etc.
  Note that an item can also be an individual or a set of individuals suggested to the user in social networks. The set of items available in the system is represented by I, where i∈I.

II.3.2 Rating concept

A rating is a numerical value in any scale (the most used is [1-5] or binary (like/do not like, good/bad, etc.) which represent the preference of the user for a given item. A common approach to building such a user preference model is through eliciting feedback from the user, either explicitly or implicitly. For explicit feedback, a score can be assigned directly by a user to an item by giving a numerical or binary value through the interface of the system. On the other hand, implicit feedback is generated by the RS itself, through inferences it makes about the user's behavior [6].

II.3.3 Community concept

A community or population is a set of similar users who share the same preferences and tastes. They are grouped together based on a given criterion (similarity criterion). Several criteria can be used in order to group users in the same community, we cite the evaluation they gave to items, their shared interests, their ages and their demographics data.
According to each of these criteria, the communities created by the system vary and the positions of users in these communities change. Therefore, each user can belong to as many communities as there are criteria used for their training.

II.3.4 Profile notion

Generally speaking, the profile of an object is a set of characteristics that allow it to be identified or represented. Two types of profiles can be used in recommendation systems, corresponding to the two entities used in these systems : the user and the item.

1. **User profile :** it is a description of the user's characteristics, which may be his or her interests, demographics, or preferences expressed in the form of evaluations, etc. Several approaches for acquiring information about the user in order to build his profile exist, and can be grouped into manual approaches and automatic or semi-automatic approaches [2].

   Manual approaches are based on the user's intervention, while automatic approaches automatically deduce the user's profile. Among the automatic or semi-automatic approaches, we can distinguish between profiling [7] and stereotype approaches [8]. Profiling consists of examining, analyzing and recording the actions and succession of actions of a user during the various search sessions and interaction with the system to determine his profile. In contrast, the stereotyping approach is based on the identification of user groups and the determination of the criteria for each group. A stereotype of a user in a recommendation system consists of a vector of items and their ratings that increase continuously as the user interacts with the system over the time [2].

2. **The item profile :** it corresponds to the description of the item with a set of characteristics, also called attributes or properties, for example in a film recommendation system, items (films) are represented by their Ids, title, genre, director, year of production, main actors. While in a document recommendation system, attributes are keywords that describe the semantic content of the document.

II.3.5 User-Item matrix

It consists of a table where each row represents a user, each column represents a specific item, and each entry represents the rating given by the user to the particular item.
Figure II.1 shows an example of user-item ratings matrix in a movie RS where users express their preferences to the items (movies) by using a five points rating scale. The items with a question mark (unknown rating) are unseen for the corresponding user. However, users only rate a small number of items which causes sparsity inside the matrix [9].

FIGURE II.1 – User-item matrix in a movie recommendation scenario [9].

II.3.6 Prediction

Prediction is the calculation of the probable score that the user will give to an item that he has not seen or evaluated.
In general, evaluation matrices have only a few cells with values while the others have unknown values and in most cases they have a "0" inside, resulting in hollow matrices. Therefore, the density of these matrices will not be sufficient to generate accurate recommendations. Then, methods for predicting missing assessments are used to increase the density of the user-item matrix in order to make more powerful and relevant recommendations.
The prediction calculation is based on the use of scores given by the user's neighbors (user-based prediction) or assigned to items neighboring the test item evaluated by the active user (item-based prediction), or given by a model (model-based prediction). Then, the items with the highest prediction values will be recommended to the user.

II.3.7 Recommendation

The recommendation is the action of calculating a list of items (Top-N items) that the user will like the most. Recommendation lists are calculated by assigning scores to items based on their popularity or preferences [10], for example. Unlike prediction, the calculation of recommendations is not strictly based on ratings.

II.4 Definition

The first definition of a Recommender System (RS) was given at Resnick and Varian's seminal article in 1997, where they described it as follows [11] :

> "In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases, the primary transformation is in the aggregation ; In others the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations."

Later, researchers have expanded the definition to [12] :

> "Any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options."

In a general way, RS is a software tool and an intelligent system that provides the user with suggestions on items or products that meet his needs or are simply likely to interest him. For example, what movie to see, what book to buy or even what music to listen, these suggestions are based on the individual's tastes by analyzing the browsing history, opinions, comments and ratings given to products and the behavior of other users [13].

There are many benefits in using these systems in various applications on the Web. Divers companies have adopted RS in their e-commerce and have proved its efficiency. Researchers in the field, stipulate that using recommendations increase the number of sales and by that increase the revenue of the company. Another advantage is the client retention : RS facilitate and guide the client e-commerce activities, which makes him, bound to visit this site again.

II.5 Classification

The authors of Tapestry [14] were the first to use the term "collaborative filtering". Five years later, Resnick and Varian published their paper [11] called "Recommender systems" in which they argue that collaborative filtering is not the only approach to Recommendation. Since then, multiple syntheses have followed to describe the evolution of the recommendation field, in which the authors have proposed classifications of recommendation approaches. These classifications are generally similar.

Recommendation techniques can be classified in different ways. Sometimes several terms are used to refer to the same method or approach.
The most used classification is the classical classification according to two approaches [3] :

- Content-Based Filtering (CBF)
- Collaborative Filtering (CF)

Then, Robin Burke [15] proposes to consider three other approaches that qualify as special cases of classical approaches :

- Population-based recommendation
- Knowledge-based recommendation
- Utility-based recommendation

The classification of Rao and Talwar [16] : this is a classification according to the source of information used.
Another classification which is The classification of Su et al [17] it is used in collaborative systems. In which the authors propose a sub-classification that includes hybrid techniques and classify them in collaborative methods. They classify collaborative filtering in three categories :

- Memory-based CF approaches : for K-nearest neighbors ;
- Model-based CF approaches including a variety of techniques such as : Clustering, Bayesian networks, matrix factoring, Markov decision processes ;
- Hybrid CF that combines a CF recommendation technique with one or more other methods.

II.6 Recommendation techniques

Recommendation systems have been studied in various fields such as the web, e-commerce and many others. Here, we discuss different approaches for proposing recommendations to a user. There are four main approaches : Content-Based Filtering (CBF), Collaborative Filtering (CF), Knowledge Based Filtering (KBF) and Hybrid Filtering (HF) . In this section, we will overview these main techniques and widen our research by discussing a fifth technique namely Demographic Filtering (DF) .

II.6.1 Collaborative Filtering

Collaborative Filtering technique is considered as the most basic and the easiest method to find recommendations and make predictions [18].
Recommendation techniques based on collaborative filtering collect and analyze user comments, ratings and preferences and exploit the similarities in ratings among multiple users and the similarities between items to make appropriate recommendations [19, 20].
Figure II.2 illustrates CF technique.

FIGURE II.2 – Collaborative Filtering technique.

CF technique is commonly categorized into two types :

a. Model Based Collaborative Filtering.

b. Memory Based Collaborative Filtering.

a.Model Based Collaborative Filtering :

In model based methods, machine learning and data mining methods are used in the context of predictive models. These methods inspect the user-item matrix to identify relation among the items in order to distinguish the list of recommendation [21].

b.Memory Based Collaborative Filtering :

Also referred to as *neighborhood-based collaborative filtering algorithms*. In this type, the ratings of user-item combinations are predicted on the basis of their neighborhoods which can be defined in one of the two ways :

b.1. User-based CF :

In this case, recommendations are given to the users based on the consideration of the cluster of other people with same preferences. For example, playing a Michael Jackson song on YouTube make you join a cluster of people who also like the artist. Then the YouTube recommendation system shows you other videos chosen by user in your cluster.

b.2. Item-based CF :

Here, the algorithm analyzes product association taken from user ratings. In order to know if a client A would be interested by an item B we first determine a set S of items that are most similar to B. The ratings on the set of item S made by the client A are used to predict whether he would be interested by target item B [21].

II.6.1.1 Advantages

- Memory-Based CF are simple to implement. It makes implementation of the recommendation system easier.
- Memory-Based CF allows us to add new data easily and in incremental manner.
- The combination of the two previous models (Memory-Based and Model-Based) improve prediction performance [22].

II.6.1.2 Disadvantages

- CF requires an enormous quantity of existing data on which user can make exact recommendation. It is very difficult in this case to make predictions to a new user whose preferences are unknown [21].
- In practice, number of items that are sold on e-commerce site is enormous. Only a few of them are rated by users. Therefore, it is hard task for a CF recommender system to make prediction in this context [22].

II.6.2 Content-Based Filtering

Content-Based Filtering (CBF), as shown in figure II.3, focuses on item descriptions and user preference profiles. Basically, the algorithms of a CBF system rely on matching user data (age, gender and item rating list) with similar items to determine which recommendation is most appropriate for a particular user [20].

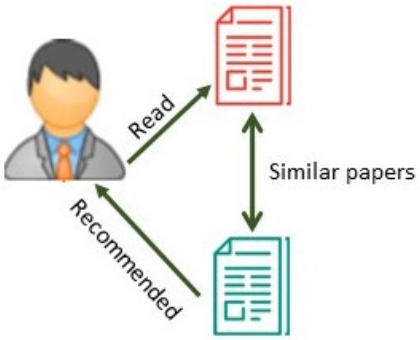

FIGURE II.3 – Content-Based Filtering technique.

II.6.2.1 Advantages

- The possibility for users to build their own profile through exclusive ratings. In other words, CBF provides user independency.
- CBF recommender system gives explanation on how the recommender works (transparency).
- CBF is advantageous for new items. This is because the model is able to link this new item with other items with same attributes that might have been rated by the active user. Therefore, recommendation can be made even if there is no history of rating for that item [21].

II.6.2.2 Disadvantages

- CBF reduces the diversity of the recommendation item : if a user has never consumed an item with a particular set of keywords, it will never be recommended to him.
- CBF is not effective in providing recommendations for new user. This is because the training model need the history of ratings of the user. It is necessary to have a huge number of ratings for the target user to make right predictions for him.

II.6.3 Knowledge-Based Filtering

Knowledge-Based Filtering (KBF) recommends articles to users based on knowledge of users, articles and/or their relationships.
In general, KB recommendations retain a functional knowledge base that describes how a specific item meets the different needs of a user, which can be achieved based on inferences about the relationship between a user's needs and a possible recommendation [22].
KBF are very useful in the context of items that are not purchased very often, such as automobiles, real estate or luxury goods. In such cases, ratings may not be available as needed for the recommendation process.
For example, real estate may have several particularities such as the number of rooms, the existence of a garden or not, the surfaces and the prices. User interest may be regulated by a very specific combination of these particularities. In this context, it is hard to associate ratings with the multiple possible combination of these options cited before.
This is where Knowledge Based Filtering RS can be addressed. With the use of knowledge bases which contain rules and similarity functions for the retrieval process, we can explicitly specify what the user want which is different from the CF and CBF techniques as shown in table II.1 above [21].

Approach	Conceptual goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings+ community ratings
Content-Based	Give me recommendations based on the content (attributes) based I have favoured in my past ratings and actions.	User ratings+ Item attributes
Knowledge-Based	Give me recommendations based on my explicit specification based of the kind of content (attributes) I want.	User specification+ Item attributes+ knowledge domain

TABLE II.1 – Conceptual goals of various Recommender Systems [21].

II.6.4 Demographic Filtering

The demographic recommendation technique is based on the user's demographic profile. It takes into account user data such as age, gender, professional status and language spoken by the user, home ownership and even place of residence. The recommendation is made taking into consideration the user's demographic similarities [22].

II.6.5 Hybrid Filtering

It is noticeable that the previous filtering recommender systems use different types of input : CF rely on community ratings, CBF on textual descriptions, KBF systems depends on interactions with the user in the context of knowledge bases and demographic filtering uses demographic similarities between users.
Each of these techniques have its strengths and its weaknesses. In a situation where a vast variety of inputs is obtainable and usable, the opportunity of hybridization is possible. Hybrid Filtering (HF) is the combination of multiple recommendation systems techniques and thereby multiple types of machine learning algorithms to create a more powerful model.

II.7 Building a Recommender System

In order to build a Recommender System efficiently, we have to pass through different steps. These steps are reported in the figure II.4 and are described in this section.

II.7.1 Framing the problem

It is determining what problem the recommender system is designated to solve. Framing consists of establishing the context in which the recommender will be applied, we must also consider for what kind of target a recommender is intended. The problem faced must be assessed in order to understand it in detail to define exactly which aspects the recommender might support.
Although, we have presented this step as the first step in building recommender step, it may be crossed with activities from other steps [23].

In other word the framing problem answer to the following questions :

- Who will be the *user* of the recommender ?
- What *problem* is solved by the recommender ?
- Which *solution* is offered by the recommender ?

II.7.2 Determining the inputs

The inputs available depend on the context in which the recommender system will be used. It may be discussion forums, historical information, documentation libraries . . . etc. Once the input is determined, these data has to be collected and transformed into a format that is processable by a machine. We can cite three steps in preparing the inputs :

1 **Collection :**
It means extracting the input data from the input data.

2 **Clean-up :**
The retrieved data can be incomplete, erroneous or duplicated, thus, it needs to be cleaned before it is used in further steps. If a data is identified as incorrect, it

should be corrected or discarded. In addition, the duplicated data must be checked and cleaned [23].

3 Preprocessing :

Preprocessing is transforming the data into a format that is processable by the machine [23]. Data preprocessing includes several operations. Each of them aims to help in building better predictive models. These operations depend on the model we want to generate and the data we manipulate. In general, it involves in data cleansing, data transformation and data reduction [24] :

3.a Data cleansing : As mentioned before, it consists of filling in missing values, correcting inconsistent data and resolving redundancy.

3.b Data transformation : Data transformation is putting the data in better perspective. It can be achieved by aggregation, generalization, normalization and attribute/feature construction.

3.c Data reduction : Data reduction consists in obtaining a reduced dataset that is much smaller in volume but produce the same analytical results.

II.7.3 Building the recommendations

The inputs are determined and the data are cleaned and preprocessed, the recommender system can be built by choosing one or more mechanism for taking the inputs and transforming it into a set of recommendations using one of the techniques cited below in the previous section [23].

II.7.4 Delivering the recommender system

There are many ways to deliver a recommendation, careful design is necessary, information should be presented at a suitable point in time.

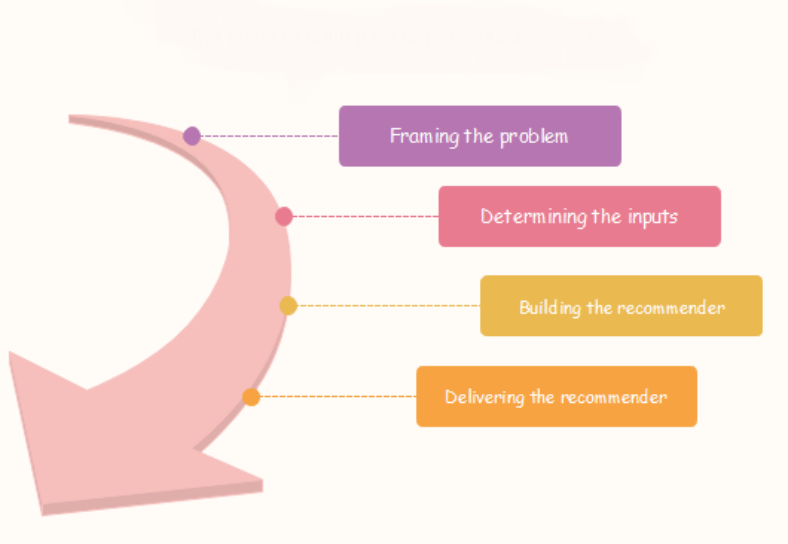

FIGURE II.4 – Steps for building Recommender Systems.

II.8 Challenges

Despite the success and the efficiency recommendation systems have shown, their considerable use has revealed real challenges. This section will go through some of the main challenges RS encounter namely : data sparsity, cold start problem, scalability, the problem of overspecialization, synonymy, privacy and the gray sheep problem.

II.8.1 Data sparsity

The problem of sparsity is one of the major challenges of the recommender system in the collaborative filtering approach. In practice, very large item sets are evaluated. Thus, even active users may have evaluated only a low percentage of the items. Therefore, the user-item interaction matrix is extremely dispersed and the RS will be unable to make any item recommendations [25].
There are two particularity under data sparsity :

- *Reduced coverage :* It is the percentage of items that the systems could provide recommendation for. The recommendations in this context fail when the number of ratings are very small in comparison with the number of items in the system.
- *Neighbor transitivity :* It is the difficulty of linking users that are positively correlated because of the sparse databases.

II.8.2 Cold start problem

The cold start problem appears when you have a new user or have just entered a new item into the system. In the first case, it is difficult to provide recommendations to a user whose preferences are unknown. In the second case, we can't recommend an article that has no rating [19].

II.8.3 Scalability

Scalability is the ability of a system to process an increasing amount of information efficiently. The explosion of data from recommendation systems generated by the enormous growth of internet information is a challenge in the face of a continuously growing demand for information [19].

II.8.4 Over Specialization Problem

The recommendation offered to the user is based on those already known or defined by his profile. This creates a lack of diversity so the user will not have any novelty. This problem is faced in Content-Based approach and is relatively small in CF recommenders where unexpected and novel items may get recommended [19].

II.8.5 Synonymy

Similar items can have different names but still have the same meanings. In this case, the recommender will have difficulties to identify whether the terms are similar or not. To reduce this problem, different techniques including ontologies can be used [26].

II.8.6 Privacy

Users are hesitant about feeding data to recommender systems. Therefore, a RS should build trust among their users by including randomized perturbation techniques that allows users to publish their private data without exposing their identities, and using Semantic Web technologies especially ontologies in combination with NLP techniques to mitigate the unwanted exposure of information [26].

II.8.7 Grey sheep and black sheep :

Grey sheep is when the opinion of a user does not match with any group, and therefore, is unable to get benefit from any recommendations. Black sheep are those users who have no or very few people who they correlate with. Recommendations are very difficult to make for this category [25].

II.9 Conclusion

Recommender systems are becoming essential in many industries and hence, have received always more attention in the recent years.
In this chapter, we have introduced basic notions required for a better understanding of recommendation systems. We followed that by defining what RSs are and the different classification that exists in the literature. We presented the main techniques of recommendation and the steps to follow in order to build these systems. We concluded by giving some of the challenges faced in this field.

In the following chapter, we are going to expose our related work that permitted us to build our approach and we precede that by giving an overview of two major domains related to recommendation systems namely machine learning and ontologies.

Chapitre III

Related work

III.1 Introduction

Recommender systems use algorithms to provide users with product or service recommendations. Recently, these systems have been using machine learning algorithms from the field of Artificial Intelligence (AI) . Machine Learning (ML) uses computers to simulate human learning and allows computers to identify and acquire knowledge from the real world, and improve performance on some tasks based on this new knowledge. However, choosing a suitable machine learning algorithm for a recommender system is difficult because of the number of algorithms described in the literature.
In order to overcome the issue of formulation and poor vocabulary in the recommendation, we also wanted to introduce another domain that helps dealing with this issue : ontology and semantic web. The semantic web offers the potential to help by making research queries more intelligent with the help of a new concept namely ontology.
In this chapter, we will first introduce the different concepts of ontology and machine learning and their relationship with the RS field. Next we will go through our literature review in which we describe previous works in the field of RS and give an analysis of the global achievement that helped us identify our approach.

III.2 Ontologies

III.2.1 Definition

Several definitions of the term ontology have been proposed according to currents and communities of thought. The word ontology has a long history in philosophy, in which it refers to the subject of existence. We have chosen to give definitions of the term ontology from a knowledge engineering perspective. The main ones are summarized in this section.

Ontology was introduced by Grüber and his team at Standford in 1993. His definition is the most cited in papers and researches works. Grüber stipulate that [27] : "An ontology is a **formal** and **explicit** specification of a **shared conceptualization**."

- **Formal :** refers to the fact that an ontology has to be machine-readable, i.e. that the latter must be able to interpret the semantics of the information provided ;
- **Explicit :** signifies that the type of concepts used and the constraints on their use must be explicitly defined ;
- **Conceptualization :** refers to an abstract model of certain phenomena in the world that identifies appropriate concepts of this phenomenon ;
- **Shared :** indicates that the ontology supports consensual knowledge, and is not restricted to some individuals but accepted by a group.

That is, an ontology is a description of the concepts and relationships that can exist for an agent (Human and machines).
Among the numerous other definitions, we cite :

1. Guarino [28] : an ontology is a shared vocabulary. It is a characterization of the "agreed" meaning of this vocabulary ;
2. The W3C : an ontology defines the terms used to describe and represent a field of knowledge.

III.2.2 Typology

According to their use, four categories of ontologies are classically distinguished [29] : generic ontologies, domain ontologies, task and application ontologies.
Generic ontologies : also called top level ontologies or high level models. They describe general concepts regardless of a particular domain or problem. Concepts can be time, space or events.
Domain ontologies : specify general concepts on a particular domain. The vocabulary is generally related to a domain knowledge like healthcare or law. The different concepts of domain ontologies are often considered as a specialization of generic ontologies. Domain ontologies are composed of :

1. A description of the vocabulary of the domain ;
2. A typology ;
3. A set of relations such as class/super-class.

Task Ontologies : describe the vocabulary of terms needed to perform generic tasks or activities (e.g., diagnosis) by specializing the concepts provided by the top level ontology.
Application ontologies : application ontologies describe the structure of knowledge necessary for the realization of a particular task.

III.2.3 Domain of application

III.2.3.1 Information systems

The main purpose of using ontologies in Information Systems (IS) is to reduce the conceptual confusion in the system and to lean towards a shared comprehension for the purpose of knowledge interoperability and reuse. It is used to :

- Describe and process multimedia resources ;
- Pilot automatic natural language processing ;
- Allow the integration of heterogeneous information resources.

Therefore, ontologies can be found in e-commerce sector, digital libraries, biology...etc.

III.2.3.2 Semantic web

The Semantic Web is an emerging research area which aims to overcome the challenge of allowing humans and computers to cooperate in the same way humans cooperate with each other by providing metadata that describes the information. These metadata are provided by the core component of the SW : ontologies. The purpose is to improve the organization, management and operation of the understanding of electronic information. Ontologies serve as a standardized vocabulary for the knowledge sharing. The World Wide Web Consortium (W3C) supports activities related to the SW through the Web Ontology Working Group.

III.2.4 Semantic similarity measure

Similarity measure is the process of assigning a numerical value reflecting the degree of resemblance between two ontology concepts. For example, in the field of cinema, concepts such as film, actor/actress, and genre are often employed. Then, relationships must be established between these concepts.
There are two types of relationships : taxonomic relationships and semantic relationships. The first type structures the hierarchy of ontological concepts by establishing links of specificity or genericity between them, e.g. Action Film and Film. The second type represents a semantic relationship between two concepts, e.g. the relationship “is realized by “between film and filmmaker. Semantic similarity measure has been widely used in natural language processing, information retrieval, word sense disambiguation and recommender systems [30].
In recent years, the measures based on WordNet have attracted great concern. WordNet is a lexical database developed and maintained by Princeton University since 1985. WordNet Nouns, verbs, adverbs and adjectives are organized by a variety of semantic relations into synonym sets (synsets), which represent one concept. Examples of semantic relations used by WordNet are synonymy, autonomy, hyponymy, similar, domain and cause and so on. Some relations are used for word form relation and others for semantic relation.

These relations will be associated with to form a hierarchy structure, which makes it a useful tool for computational linguistics and natural language processing [31].

Semantic similarity measures might be used for performing tasks such as term disambiguation, and for checking ontologies for consistency or coherency. Many measures have been proposed. We can classify the measures into four classes : path length based measures, information content based measures, feature based measures, and hybrid measures [31] :

- **Path length based measures :** The main idea of path-based measures is that the similarity between two concepts is a function of the length of the path linking the concepts and the position of the concepts in the taxonomy.
- **Information content based measures :** It assumed that each concept includes much information in WordNet. Similarity measures are based on the Information content of each concept. The more common information two concepts share, the more similar the concepts are.
- **Feature based measures :** Different from all the above presented measures, feature-based measure is independent on the taxonomy of the concepts, and attempts to exploit the properties of the ontology to obtain the similarity values. It is based on the assumption that each concept is described by a set of words indicating its properties or features, such as their definitions or “glosses” in WordNet. The more common characteristics two concepts have and the less non-common characteristics they have, the more similar the concepts are.
- **Hybrid measures :** The hybrid measures combine the ideas above presented. In practice many measures not only combine the ideas above, but also combine the relations, such as is-a, part-of and so on.

III.3 Machine learning

III.3.1 Definition

According to Arthur Samuel (1959) , the inventor of machine learning, Machine learning consists in letting the computer learn which calculation to perform, rather than giving it the calculation [32].

> "Machine Learning is the science of getting computers to learn without being explicitly programmed."

In the book (1997) entitled "Machine Learning", Professor Tom Mitchell et al define machine learning in these terms :

> "Machine Learning is the study of computer algorithms that improve automatically through experience."

Machine learning can be defined as an artificial intelligence technology that allows machines to learn without being specifically programmed for this purpose.

In other words, to develop algorithms capable of accumulating knowledge without having been explicitly programmed for.
Machine Learning is explicitly related to Big Data, as computers need streams of data to be analysed and trained on in order to learn and develop. Therefore, the Learning Machine, which is essentially derived from the Big Data, needs the Big Data to function. Machine Learning and Big Data are therefore interdependent [33]. Machine learning is linked to several disciplines :

- **Statistics :** for model inference from data.
- **Probabilities :** for modeling the random aspect inherent in the data and the learning problem.
- **Artificial intelligence :** to study the simple tasks of pattern recognition that humans do.
- **Optimization :** to optimize a performance criterion either for estimate the parameters of a model, or to determine the best way to estimate the parameters of a model decision to be taken, given an instance of a problem.
- **Computer science :** since it is about programming algorithms.

III.3.2 Machine learning types

Machine learning is generally classified into three broad categories algorithms [33] :

Supervised Learning : In supervised learning, the computer is provided with examples of inputs that are labeled with the associated output values. The purpose of this method is to learn a general rule that matches inputs to outputs by comparing the actual outputs with the "learned" outputs and thus predict the label values of a new input. Types of supervised learning are :

- Classification
- Regression

Unsupervised Learning : In unsupervised learning, the data is unlabeled and non-categorized so that the learning algorithm finds the structure of its data by itself without prior training. Unsupervised learning can be further subdivided into :

- Clustering
- Association

Reinforcement learning : A reinforcement learning algorithm or agent learns by interacting with its dynamic environment in order to achieve a specific goal. The agent receives bonuses for correct execution and penalties for poor execution. The agent learns without the intervention of a human based on observation.

III.3.3 Machine learning in recommendation systems

Recommendation systems are one of the most successful and widespread applications of machine learning technologies in companies because the recommendation of new articles or products can be processed by machine learning algorithms.
Machine learning algorithms in recommendation systems are generally classified into three categories (chapter 2) : content-based, collaborative and hybrid filtering methods.

III.3.4 Machine learning algorithms in RS

III.3.4.1 Decision trees

The decision tree is an algorithm that is based on a graph model and as its name implies it uses a decision model in the form of a tree. It is directed with a node called "root" followed by internal nodes or test nodes and nodes that have no descendant called leaves, terminal nodes or decision nodes. It uses a hierarchical representation of the data structure in the form of decision sequences for the prediction of an outcome or class. Their advantage is that they can be calculated automatically from databases by supervised learning algorithms.
These algorithms automatically select discriminant variables from unstructured and potentially large data. This allows the extraction of logical rules that did not initially appear in the raw data. Decision trees are often fast and accurate and are a great favorite in machine learning [34].
The most popular decision tree algorithms are :

- Classification and Regression Tree (CART)
- Iterative Dichotomiser 3 (ID3)
- Chi-squared Automatic Interaction Detection (CHAID)
- Decision Stump
- Conditional Decision Trees

III.3.4.2 Association rules

An association rule can be defined as a truth table that results from the combination of two or more characteristics. Association rules are used to derive relationships between unrelated data in a database, i.e. they are used to find relationships between objects that are frequently used together. These rules can discover important and commercially useful associations in large multidimensional data sets that can be exploited by an organization [35, 36].
The most popular association rule learning algorithms are :

- Apriori algorithm
- Eclat algorithm

III.3.4.3 Clustering

Clustering consists in the division of the population or data point into a certain number of groups so that the data points of the same groups are more similar to the other data points of the same group and different from the data points of the other groups.i.e. to put similar objects in the same cluster and different objects in different clusters [37].
The most popular clustering algorithms are :

- K-Means
- K-Medians
- Hierarchical Clustering

III.4 Literature review

This section is dedicated to provide a summary of the most important research papers related to RSs proposals in different areas, such as e-commerce, e-learning, and social events. The proposals are classified through a range of research areas including ontology-based, ML-based and hybrid approaches that uses ontologies as a semantic model and ML algorithms for prediction.
Table III.1 summarizes the major commonalities and differences between these approaches.

III.4.1 Ontology-based RS

Different ontology-based recommendation approaches have been developed using a variety of methods.

Zehra et al. [38], propose an approach for developing a recommendation system using ontology-based sentiment analysis to provide schools that matches the active user's preferences. This work uses the active user comments on their Facebook feed about a certain school to extract features that allow the creation of an ontology-based recommender, based on the polarity of the comments. For elucidating the knowledge domain, school ontology is manually designed based on a set of extracted post/comment data. Moreover, the recommendation process can recommend schools based on certain branches chosen by the active user. Despite the differences between this work and our proposal, this paper allows to perceive that ontology-based recommender can improve the quality of the recommendation process.

Nilashi et al. [39] present a hybrid RS, which combines collaborative filtering with ontology and dimensionality reduction techniques in order to improve the sparsity and time complexity of the collaborative filtering approach. The experimental evaluation shows the benefits of building a hybrid RS when compared to the traditional systems. The approach focuses more on the ratings given to movies. In addition, it does not give attention to the diversity of the movie's recommendation, which is one of our main contributions. In the

CF part, we also use a dimensionality reduction technique, Singular Value Decomposition (SVD), to find the most similar items and users in each cluster of items and users, which can significantly improve the scalability of the recommendation method. The authors claimed that the experimental results showed that the proposed method is effective in improving the sparsity and scalability problems in CF.

Ibrahim et al. in [40] proposed an ontology-based hybrid approach in order to recommend customized courses in a framework named Ontology based Personalized Course Recommendation (OPCR) framework. The proposed method can enable students to gain a comprehensive knowledge of courses based on their relevance, using dynamic ontology mapping to link course profiles and student profiles with job profiles. The recommended courses must fit student's personal needs by integrating all available information regarding the courses and supporting students to choose courses based on their career objectives. The OPCR framework is a key element for the created HRS. The designed HRS is available online for learners and researchers. The approach is flexible and can be adapted to different domains.

Feng et al. [41] propose an approach to improve the collaborative filtering recommender for the purpose to improve the accuracy and quality of the RS, mainly when the data sparsity issue occurs. The proposed approach consists in the use of three impact factors in order to find the similarity between users. The impact factors that are included in the similarity calculation between users are not only the co-rated items but also all data available between the two users. Experiments were performed to validate the efficiency of the proposed algorithm. Results show that the proposed method can effectively improve the preferences of the recommender system and it is suitable for the sparsity data.

Sheridan et al. in [42] present an ontology-based RS that integrates the knowledge represented in a large ontology of literary themes to produce fiction content recommendations. The authors propose an ontology-based method for computing similarities between items and its integration with the classical Item-KNN (K-Nearest Neighbors) algorithm. As a study case, they evaluated the proposed method against other approaches by performing the classical rating prediction task on a collection of Star Trek television series episodes in an item cold-start scenario. The authors claimed that their proposal retuned better accuracy.

Another ontology-based RS was proposed by Ayundhita et al. [43]. The proposal aims to recommend laptops to users who are not aware of low-level specifications. The proposed approach uses the ontology to map functional requirements with low-level specifications. The system asks functional requirements to the user according to their preferences and then the ontology maps those requirements with low-level specifications to find laptops that match the active user's preferences. This approach requires users to introduce the functional requirements of the intended laptop. However, when the active user does not know enough functional requirements, the RS does not work. According the experimental

evaluation, the proposal achieves an accuracy of 84, 6% compared to general RSs that achieve a low accuracy of 61.5%. However, the experimental evaluation was performed with only 39 users, which is not enough to conclude that the proposal is accurate.

III.4.2 ML-based RS

Integrating ML algorithms in RS certainly will improve the recommendation accuracy. In this section, we present the main RSs that use ML algorithms.

Verma et al [44] introduced another approach and proposed a recommender system based on hybrid filtering. The study is about numerical data in forms of ranks or ratings for different product and services. These data first filter/transform as per requirement. They analyzed different size files and came with the conclusion that their model was working perfectly and that size was not influencing the execution time. However, the model proposed is not handling text data.

[45] proposed a generic architecture for big data healthcare analytic by using open sources, including Hadoop, Apache Storm, Kafka and NoSQL Cassandra. The combination of high throughput publish subscribe messaging for streams, distributed real-time computing, and distributed storage system can effectively analyze a huge amount of healthcare data coming with a rapid rate.

Another interesting approach related to education big data is proposed by Dwivedi and Roshni in [46]. The RS uses recommendation techniques based on collaborative filtering to recommend electives to students. The recommendation is based on articles from the Mahout ML library over Hadoop to generate a set of recommendations. Schools, colleges or universities to suggest alternative electives to students can use the results of this study.

The authors in [47] proposed a prediction and RS in the context of diabetes. Healthcare RSs are important as people use social network to knowledge their health condition. They used the hybrid filtering approach to provide personalized healthcare recommendation. Data from various sources combined with powerful learning algorithms led to meaningful insights. Prediction here represents the disease risk diagnosis for future cases based on active patients. On the other hand, reliability and security of social health information must be considered.

Al-badi et al [48] explored the benefits of applying big data analytics in healthcare. The research is based on existing literature reviews and secondary data. Moreover, their experiment conducted to investigate the potential benefits using a real dataset and it showed very promising results. This research show that adopting the analytics in healthcare is essential but the highlighted limitations and challenges must be well addressed and resolved.

Sullivan and Ratnaparkhi [49] came with a novel approach in building a recommender system to new clients about food with a set of dishes which have been classified into categories (good, average, bad) based on previous reviews. He followed the KDD methodology, which is a set of steps that simplify the implementation of any project, which aims at generating knowledge from a given dataset. The result was that the prediction model gave a good accuracy and could be trained to much larger datasets.This research provides a good implementation for RS in other domain. However, it has the ability to work only with one world long so composed dishes are not taken into consideration.

Fernandez-Garcia et al in [50] created a RS with the use of ML algorithms to predict and recommend to developers the best cross-device component-based interfaces. Their work addresses the problem of creating a useful RS that would be able to forecast the use of components in cross-device component-based applications with multiple forms of interactions. The proposed system intends to create a RS that can help users discover the components most suitable for them, thereby improving their user experience of the applications. The authors conduct series of experiments that create recommendation models applying several ML algorithms to the optimized dataset to determine which recommender model obtains a higher accuracy.

Ramzan et al. [51] proposed a novel CF recommendation approach in which opinion-based sentiment analysis is used to achieve hotel feature matrix by polarity identification. The proposed approach combines lexical analysis, syntax analysis, and semantic analysis to understand sentiment towards hotel features and the profiling of guest type (solo, family, couple etc.). The proposed hotels RS based is based on the hotel features and guest type for personalized recommendation. The system makes use of fuzzy rules to determine the hotel class depending upon the guest type. The developed system not only has the ability to handle heterogeneous data using big data Hadoop platform but it also recommends hotel class based on guest type using fuzzy rules. Different experiments are performed over the real-world datasets obtained from two hotel websites. The system takes 2.65 milliseconds to generate high-quality recommendations by reducing the system execution time.

Serrano [52] analyzes in his article the product rank relevance provided by different commercial big data recommender systems and proposed an Intelligent Recommender System (IRS) based on the random neural network that acts as an interface between the customer and the different recommender systems that adapts to the perceived user relevance. IRS gets a request from the customer and obtains the products from the recommender system data set. Serrano with his novel approach shows that using neural network in recommender system is an innovative method.

III.4.3 Hybrid RS

Bahramian et al. in [53] proposed a travel content-based RS that uses ontology information to calculate the degree of similarity between user's preferences and point of interest to provide personalized recommendations. The proposed recommendation process has three steps including ontology-based content analyzer, ontology-based profile learner and ontology-based filtering component. The system generalizes user preferences through ML techniques. The proposed system overcomes sparse data problem of the traditional content-based recommender using Spreading Activation technique to learn the user profile dynamically.

Obeid et al. [54] propose a hybrid RS to recommend universities to the students. The proposal combines ontologies with ML techniques to perform the recommendation. The authors use ontologies to represent domain knowledge about the universities and students. Moreover, the proposal not only focuses on the students' grades, but also on their skills and interests, which is an innovative idea. However, an experimental evaluation is needed to perform in order to confirm the effectiveness of the hybrid recommendation system.

Table III.1 – Summary of the literature review.

Category	Approach	Dataset	Output	Used technique	Advantages	Disadvantages
Ontology based recommender systems	Zehra et al., 2017	User comments from Facebook	Schools RS	Ontology matching Sentiment analysis SPARQL	Recommendations are made using the previously calculated sentiment score	Subjectivity classification
	Nilashi et al., 2018	MovieLens EachMovie	Movies RS	CF technique Ontology matching Dimensionality reduction	Improve the sparsity and time complexity	Need to fill data before producing recommendation

Continued on next page

TABLE III.1 – *Continued from previous page*

Category	Approach	Dataset	Output	Used technique	Advantages	Disadvantages
	Ibrahim et al., 2018	UCAS (Universities and Colleges Admissions Service)	Personalized courses RS	Ontology mapping CB technique	Addresses cold start problem	Experimental evaluation not enough
	Feng et al., 2018	Movielens Film Trust Ciao Epinions	Movies RS	CF technique Similarity Ontology matching	Addresses data sparsity	Does not improve accuracy in datasets that do not suffer from sparsity
	Sheridan et al., 2019	Star Trek Television Franchise	RS fiction content	Ontology matching CB technique	Addresses cold start problem	Experimental evaluation not enough
	Ayundhita et al., 2019	E-commerce applications	Laptops RS	Ontology matching	Good accuracy (84,6	Experimental evaluation not enough (only 39 users)
ML based recommender systems	Verma et al, 2015	MovieLens	RS about movie ratings	-Hybrid filtering -Machine learning	-Tested with different size file	-Not handling text data (based only on numerical data)
	Ta et al, 2016	Healthcare big data	Generic big data stream computing in healthcare	Collaborative filtering Machine learning	-Handles different formats of data	-Low efficiency

Continued on next page

TABLE III.1 – *Continued from previous page*

Category	Approach	Dataset	Output	Used technique	Advantages	Disadvantages
	Dwivedi and Roshni, 2017	Education big data	Recommender system for courses	Collaborative filtering Machine learning	-Useful for training -Improve the education system quality -Improve the student and teacher performance	-Need to enhance the accuracy of the recommendation
	Archena and Anita , 2017	Diabetes data collected from hospital	prediction and recommendation	Hybrid filtering Machine learning	-Minimize costs -Warn patient of health risks	-Low security and reliability
	Al-badi et al 2018	-UC Irvine machine learning repository	Predicting kidney disease	-Machine learning	-Efficient healthcare services -Cost effective -Better performance	-Data collection obstacles -Management barriers
	Sullivan and Ratnaparkhi, 2018	-Yelp (restaurant category)	RS of best dishes at a restaurant to a new customer	-NLP -Machine learning	-Larger dataset can be used with this model	- The system cannot handle compound word
	Fernandez-Garcia et al, 2018	Component-based web application	RS suggesting the most suitable component based interfaces	-ML algorithms	-Accuracy >= 80	-Difficulty to gather data

Continued on next page

TABLE III.1 – *Continued from previous page*

Category	Approach	Dataset	Output	Used technique	Advantages	Disadvantages
	Ramzan et al., 2019	Hotel websites	Hotel RS	ML algorithms CF technique Sentiment analysis	Reduced system execution time	Non-credible information, sometimes missing
	Serrano, 2019	MovieLens Trip Advisor Amazon	Intelligent Recommender System	-Random Neural Network -Machine learning	-Direct connection between customers and products in a reduced time -Rearrange the products until the satisfaction of the customers	-Difficulty to obtain Data
Hybrid based recommender systems	Bahramian and Abbaspour, 2015	Tourism domain information	Travel RS	Ontology Matching ML algorithms CB technique	Mitigates the sparse data	Non-credible information, sometimes missing
	Obeid et al., 2018	Education data	RS of the appropriate university	Ontology Matching ML algorithms	Efficient learning services	Experimental evaluation is not performed

III.5 Analysis and comparison

Table III.1 summarizes the main features of the approaches cited above. The table contains seven columns that indicate a comparison criterion as follows :

- The column **"Category"** defines the category of the proposed approach (Ontology-based, ML-based or hybrid).
- The column **"Approach"** designates the underlying approach.
- The column **"Dataset"** indicates the data source used to generate the recommendations.
- The column **"Output"** indicates in which domain the RS is used.
- The column **"Used techniques"** specifies what methods are used for recommendation.
- The column **"Advantages"** introduces the main advantages of the approach.
- The column **"Disadvantages"** introduces the main disadvantages of the approach.

The above-mentioned systems uses different kind of data sources for the recommendation. The application domain of Movies is the one mostly used, one reason for this result is the ease of access to data in the movie domain. The University of Minnesota maintains a dataset with several movie ratings named MovieLens which is widely used.

Most of the approaches using CB technique recommend items that are similar in content to the item that the user liked in the past. However, this technique is efficient only if the item can be represented as a set of features. In addition, these approaches suffer from plasticity (the ability to change the user's preferences).
The CF technique matches users who shared same preferences using the ratings for items in particular domain. The majority of the approaches are based on the CF technique.
For hybrid recommendation, it integrates two or more recommendation techniques to limit the weaknesses of individual ones. However, the use of RSs has exposed many challenges : data sparsity, cold start problems, fraud and privacy when it comes to some areas like health domain.

Those who try to improve CF approaches only take into concern the ratings given to the products by the users, which mean that they do not include the knowledge about the active user in the recommendation process, active user's neighbors, products nor relationships between them.

Some papers have particular methods to elicit user interests and preferences ; they are focused on improving the quality of the recommendation process by using ontologies as a semantic model, which requires effort and knowledge from the active user. A RS based on an ontology can solve the cold start problem due to an initial lack of ratings for new users. However, users are not always familiar with the domain. Therefore, in order to improve the recommendation process, we try in our work, to avoid knowledge about the active user by using only the users' ratings given to the products and we create the user profile according to the ontology model specified by the domain expert.

Even if the majority of the proposed methods have been widely studied and used for recommending accurate and reliable items to users, they have not been well combined with the optimization to enhance further the personalized search or recommendation.
Most of the cited papers involved ML algorithms. ML algorithms are used in order to build supervised or unsupervised systems, which are applicable in different domains. Integrating ML algorithms in recommender system certainly will improve the recommendation accuracy. Clustering approach is widely used in generating the cluster, as it is a powerful unsupervised learning method to evaluate correctly the large amount of data created by applications. However, recommendation will be better only if the formed clusters are good.

The analysis of the literature review highlights that it is vital to use ML algorithms to create RSs for suggesting the most suitable product for users and integrate contextual information into theses RSs to provide improved recommendations. For this purpose, we would be able to choose the best ML algorithm that can handle RSs characteristics. In addition, the combination of ontology-based RS with ML algorithms is a promising approach for improving recommendation accuracy.
Nevertheless, to the best of our knowledge, little corresponding research has been conducted based on ontologies and ML algorithms, generally, it is noticed that this area of research did not receive sufficient attention.

Therefore, to cope with the issues cited above, we propose a new RS ontology-based in which ML algorithms are used to achieve user needs identification and provide precise and efficient recommendations.
By inspiring from hybrid approaches (ontology-based and ML-based), our method's focus is on prediction process, solving the cold start problem. Our work presents a different approach that combines knowledge-based recommenders with a collaborative filtering approach to provide more diversity recommendations.
Furthermore, the backup of the recommendations history is a crucial concept in our proposal, as it allows providing rapid responses for similar preferences. This is why our system is based on traceability, which consists in saving the recommendations history via two repositories, namely the user Profiles repository (PR) and the Recommendations Repository (RR).

Moreover, our proposal considers the approach scalability. One of our main goals is to provide recommendations of products that not only receive high scores from the respective neighbors but also from all users that have purchased them.

III.6 Conclusion

In this chapter, we have gone through our literature review and related work in the field of recommendation systems. We introduced two important domains for building effective recommendation models : ontology and machine learning. We defined what machine learning is, enumerated its types and enumerated some of its algorithms used to build RS. The same work was done for ontologies where we defined the concept and the benefit of using it in RS.
In the next chapter, we will introduce our approach in building our recommendation systems by describing the different steps followed.

Chapitre IV

Our proposal

IV.1 Introduction

In this chapter, we present in detaile the main components and characteristics of the ontology-based RS we are developing. We give descriptions of our proposed RS that uses ontologies as a semantic knowledge model and ML algorithms and aims to improve the recommendation process according to user preferences.

IV.2 Approach

In this section, we present our approach to build a recommendation system based on ontologies and ML algorithms.
The system architecture is depicted in figure IV.1 and involves five steps : (1) profile creation where we build a profile model according to user preferences, (2) profile enrichment, in this step the builder profile is annotated with a domain ontology to resolve interoperability issue and provide a profile model readable, (3) search for a similar profile, whose goal is to search for before recommendation in order to reduce work space, (4) clustering allows to group together same profiles, and finally (5) recommendation process that involves the recommendation as a whole.

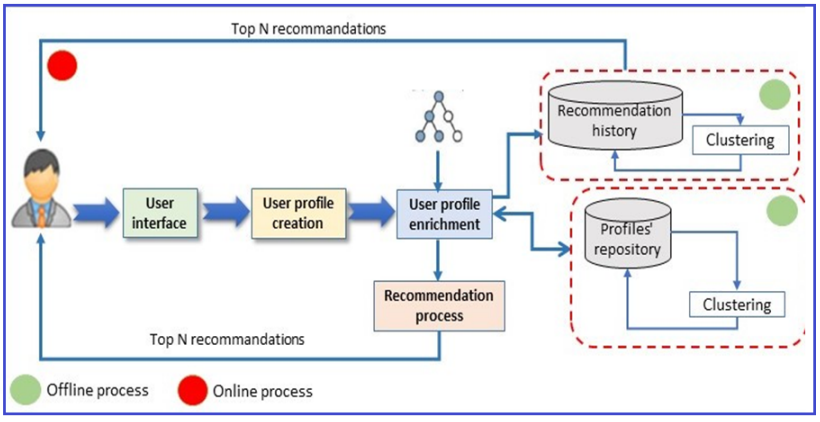

FIGURE IV.1 – System architecture.

The three important aspects that we highlighted in our proposal during the recommendation process are :

- Formulation of user requirements
- Search for profiles' similarities to provide faster recommendation.
- Improvement of the recommendation process using the ML algorithms

The recommendation process as a whole is modeled in a sequence diagram expressed in UML (Unified Modeling Language) standard and presented in figure IV.2. The user sends the request to the system, which discovers the suitable recommendation. The responses are ranked before they are presented to the final user. At the end, the top N recommendations are returned to the user.

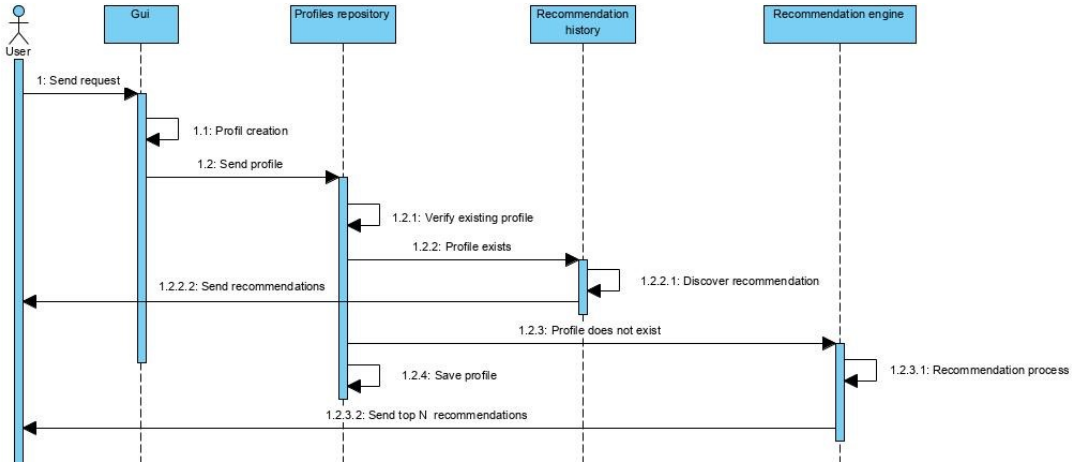

FIGURE IV.2 – High level interaction UML sequence diagram.

For this purpose, we have used forms where the user enters his relatively preferences which will be used later to create a profile for him. This profile will be enriched with domain ontology. Once the profile is enriched, we will look for it in the profile database. If the profile exists that means that another user with similar preferences is already registered in the database, so, we recommend the appropriate products directly from the recommendation database.
If not, we add the profile to the profile database and send the data to the recommendation engine for recommendation processing. Finally, the result is registered in the recommendation database. Integrating ML algorithms with ontology-based recommender systems lead to many remarkable revolutions in improvement of RS processes [55]. Moreover, the combination of ontology-based RS with ML algorithms is a promising approach for improving recommendation accuracy and efficiency.

The different steps of the new approach to the recommendation will be detailed in the following subsections :

IV.2.1 Step 1 : creating the profile

This step is responsible for creating the user profile based on the information entered. This is achieved via an HTML web page integrated into the portal.

This choice offers a user-friendly interface and allows the user to specify his preferences as input values. And for simplicity and fluidity, we will use HTML forms which are the most popular interface for communicating with people for data entry and display on the web.
The process of creating a profile using an HTML input form is described in figure IV.3 and goes through three basic phases :

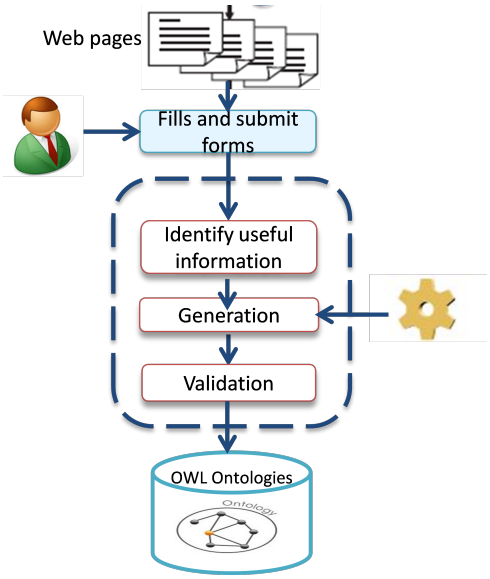

FIGURE IV.3 – Profile creation steps.

1. **Phase 1 : Identifying useful information**
   This step allows the acquisition of the necessary information to generate the relevant ontology (concepts, attributes, relations and axioms) from HTML forms. The HTML form is designed with the HTML code and is integrated in the <FORM> tag. Each HTML form contains a set of form fields that consists of various parameters as needed, for example text boxes, drop-down menus, radio buttons, check boxes, including banners, advertisements, diagrams, etc.
   Once the user has completed all the required information available in the forms, the user submits the form to the web server for further processing ; the relevant information is entered during the request runtime.
2. **Phase 2 : Generation**
   This stage focuses on the production of the profile ontology. The method we propose to build profile ontology from an HTML form data file includes different processes and at this phase, we use the mapping rules stated in [56].
   Ontologies are widely used and have proven their usefulness in many fields such as : knowledge engineering, artificial intelligence, information retrieval, e-commerce and are at the heart of the semantic web. Our proposal is motivated by the fact

that ontologies are an efficient way to manage and share knowledge of a particular domain among people and/or systems.

3. **Phase 3 : Validation**
   An automated result validation phase is necessary due to the erroneous concepts and relationships that may be introduced by the previous steps. The purpose of this phase is to validate the produced ontologies for correctness.

IV.2.2 Step 2 : Profile enrichment

This step consists of a matching process between a domain ontology and the ontology produced in the first step.
This matching is performed using a semantic similarity computation algorithm. In our work we have selected the wu-Palmer algorithm [56] which is revealed to be simple to compute, in addition to the performances it presents while remaining as expressive.
Wu and Palmer's measure of similarity [56] is predicated on the subsequent principle : Given an ontology formed by a group of nodes and a root node R . Let X and Y be two elements of the ontology whose similarity is calculated. The principle of similarity calculation relies on the gaps (N1 and N2) separating the nodes X and Y from the basis node R and also on the distance separating the subsumant concept (CS) of X and Y from the node R. The similarity measure of Wu and Palmer is defined by the subsequent expression :

$$SimWP = \frac{2N}{N_1 + N_2}$$

IV.2.3 Step 3 : Search for similar profiles

During this step, we compare the profile created in the previous step with all the profiles stored in the database. We suppose that the preferences could be the same for the other users of the application, which reduces the execution time and makes the recommendation easier and faster. A pair-wise comparison is made between this profile and the existing profiles.

IV.2.4 Step 4 : clustering profiles

Clustering involves the task of grouping data points into homogeneous classes or clusters. So that items in the same cluster are as similar as possible and items in different classes are as dissimilar as possible. In this step, we use standard Kmeans algorithm that is implemented within the profiles database [57]. The reason to choose K-means is that it is efficient and scalable for processing large volume of ontologies.
K-means clustering is an unsupervised machine learning algorithm. It is unsupervised because we don't give it any examples of what good I/O pairs would look like. It is also a parametric algorithm because it takes one parameter, k to work. The parameter k is used to tell the algorithm how many groups it needs to find. It works by finding the k points, called centroids, that satisfies the theorem that the sum of the distances between

all elements and their assigned centers of gravity is as small as possible. The k-means algorithm goes through the following steps :

1. Select k locations as cluster centers.
2. Loop through the following :
   (a) For each data point in the cluster, find the center of gravity with the shortest distance.
   (b) When all points are assigned, calculates the sum of all distances between the element and its center of gravity.
   (c) If the distance is not less than the previous run, returns the clusters.
   (d) Moves each centroid to the center of the assigned cluster.

IV.2.5 Step 5 : Recommendations process

In this part we are going to describe the recommendation process that we have built. First we introduce the architecture of the recommendation process and then review the different machine learning techniques for recommending.
The figure IV.4 below shows the recommendation architecture and come up with an illustration of which applications will build our recommender system .

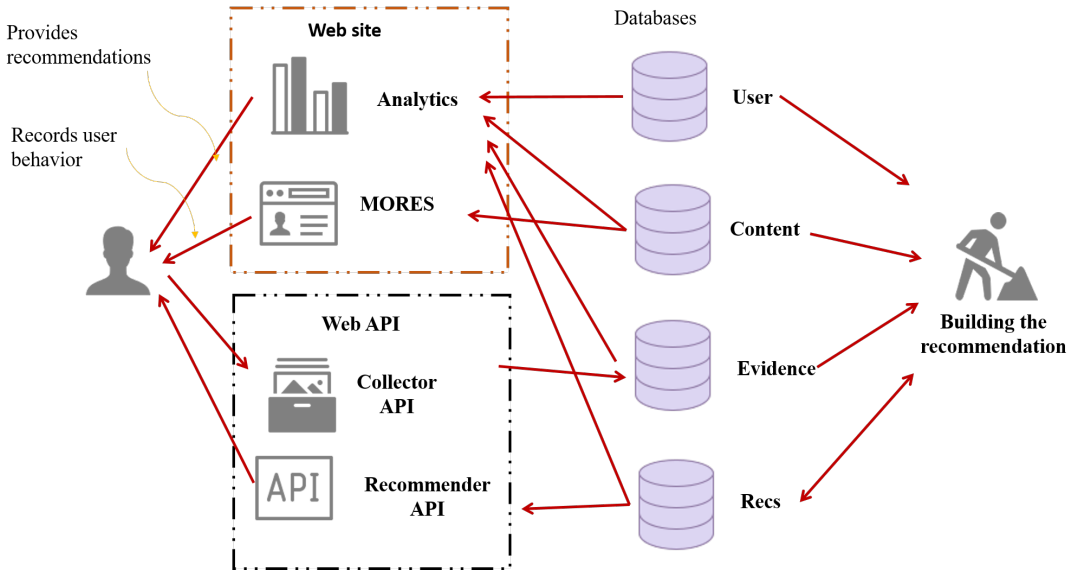

FIGURE IV.4 – MORES architecture.

Before describing the different components of this architecture in details we will first give a quick overview of the structure presented in the previous figure :

- **MORES** This is where the client logic (HTML, CSS, and JavaScript) is placed along with the Python code responsible for retrieving the movie data. This is the main part of the site.
- **Analytics** It is where everything can be monitored. This part will use data from all the databases to give an analytic chart.
- **Collector** This handles the tracking of the user behaviour and stores it in the evidence database.

- **Recs** It will deliver the recommendations to the MORES site. Recs represents the heart of the system.
- **Building the Recommendations** This is where all the machine learning and recommendations algorithms are. The builder pre-calculates recommendations, to provide them to the user.

IV.2.6 The collector

The collector's function is to collect data on MORES site. Data means events from anything that interacts with the user (a click, mouse hover ...etc.). The server side of the collector is built using a Django web API. When the server receives a notification that an event has occurred, its job is to serialize it and save it in the log database. The client side of the collector consists of a simple JavaScript function that posts data to the evidence collector on the server. This function is called "a snitch". The role of the collector is to record implicit ratings. Because of the social influences, explicit ratings can be easily biased, therefore it is important to collect all the action the user can make in the application.

The figure IV.5 presents the different steps that happen when a user clicks on the watch button of a film.

1. User clicks on the "watch" button.
2. The onClick events activates the JavaScript function.
3. The web server receives an HTTP request.
4. A lookup in the URL delegates it to the collector app.
5. The collector matches the log to view. This view creates a log object.
6. The Django ORM system saves the log object in the database.

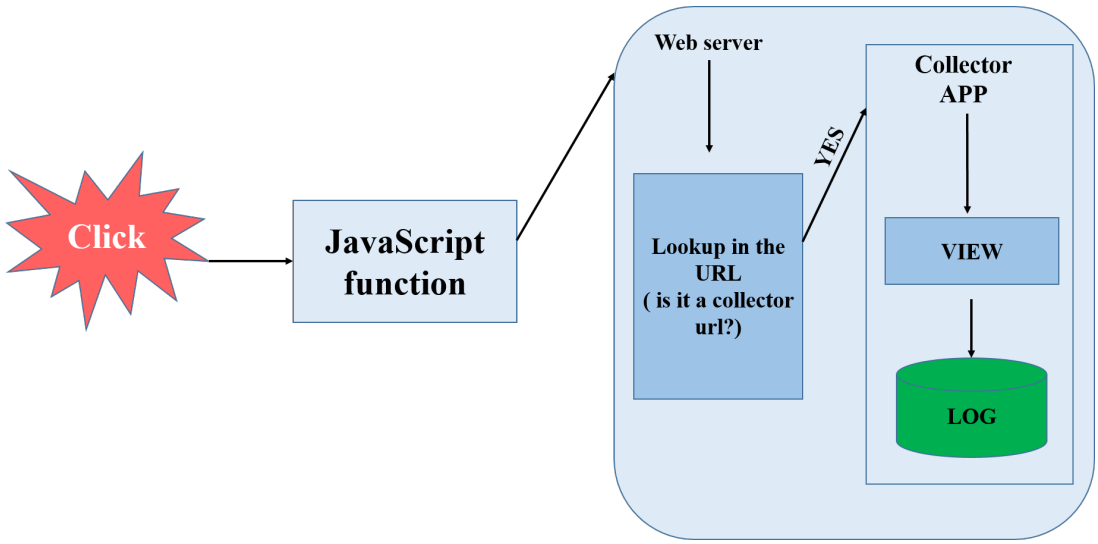

FIGURE IV.5 – Collector's behavior when the watch button is clicked.

IV.2.7 The analytics

The analytics page of MORES shows information about the data. Having a visualisation about our application makes it more instructional and educative. Therefore, we have implemented a dashboard to MORES for the purpose of tracking the stream of events and customer behaviour. Adding into that, each user has his own analytics page.
To build this part of MORES we will introduce 3 major concepts : Clustering, Similarity and implicit ratings.

IV.2.7.1 Clustering

In order to avoid comparing each new person arriving on the site with all users of the system and to obtain quick lists of similar users, it is best to divide the data set into smaller groups, so that the calculation of similarity will be done in groups with fewer users. This is why the most appropriate solution is the use of a clustering algorithm. So, we have chosen **k-means clustering**, which is one of the most popular segmentation algorithms used.

This method of clustering is implemented as follows :

1. Retrieves all user_ids from the leaderboard
2. Retrieves all content_ids from the leaderboard
3. Creates an instance of the k-means clustering algorithm
4. Make the cluster
5. Save the cluster.

IV.2.7.2 Similarity

The similarity features allow us to measure how similar two users are, using the ratings they have given to the content. In general the similarity can be defined as follows : Given two elements,i_1 and i_2, the similarity between them is given by the function sim (i_1,i_2). The return values of this function will increase as the items are similar. We can say that the similarity between the same item is Sim $(i_1,i_1) = 1$, and two items which have nothing in common will be Sim (i_1, nothing in common with i_1) = 0. In our system, similarity is considered part of the recommender system, so we added a similar_users method to the recommender API. This method requires a user_id and a type. The type allows it to be easily extended with other types of similarity calculations. This method also uses the Pearson method as well as the jaccard method.

1. **Jaccard similarity**
   Also called Jaccard index or Jaccard similarity coefficient. It is a measure of distance that indicates how close two sets are to each other. To calculate Jaccard similarity for two items you need :
   (a) Calculate the number of users who watched both items
   (b) Calculate the number of users who watched one (or both)

(c) Then divide (a) by (b)
Written more formally :

$$\underset{Jaccard}{Similarity}(i,j) = \frac{\text{users who watched both items}}{\text{users who watched either i or j}}$$

Where i represents item 1 and j represents item 2.

2. **Pearson similarity**
The algorithm calculates how much the two lines correlate between two users. The formula return a value between -1 and 1, where :
- 1 indicates a strong positive relationship.
- -1 indicates a strong negative relationship.
- A result of zero indicates no relationship at all.

The algorithm goes through the following steps :

(a) Obtain all evaluations from current users

(b) On the basis of these " ratings ", we find all the users who have also rated one or more of these films

(c) Retrieve the ratings of all users who have ratings that overlap with those of the user

(d) Extract all users

(e) Iterated through the method all users (Pearson or jaccard)

(f) Add a user to the list of similar users

IV.2.7.3 Implicit ratings

Implicit ratings are deduced from monitoring user's behaviour in order to ease information overload and help users with efficient recommendations. When a user watch a film or click multiple times on that film to read its description, we can deduce that he is probably interested so we must recommend it for him. This is implicit ratings. MORES does not allow users to rate films so using implicit ratings is a must.
In order to calculate implicit ratings we follow 3 major steps : retrieving data, calculating implicit ratings and viewing the results.

1. **Retrieving data**
Before calculating implicit ratings we must retrieve the log data that contains all the interactions of a specific user u with the content. The purpose of retrieving the data is to know how often a user interacts with a specific item and what kind of interaction it is. Each event determine how satisfied the user is for that content. In our case we have five events :

(a) Click to see the details about the film.

(b) Click on "View more details" to see more about the film description.

(c) Click on "save later".

(d) Click on a specific "Genre" in the list.

(e) Click on the “watch” button.

2. **Calculating implicit ratings**

   Knowing the list of events, we can deduce the degree of user satisfaction according to the importance of each recorded event. Basically, a user who clicks on the “ watch “ button or clicks several times on the “ view more details” button is much more interested in the content than the one who only pops over the detail (Table IV.1).

   To calculate implicit ratings, we attribute for each of these events a weight based on a logic where a higher weight signifies the great interest of the user about the content.

   The following table describes this logic :

Action	User interest	Weight
Click watch	Very interested	100
Click more details	Very positive	80
Save for later	Positive	50
Details (pop over the poster)	Not sure	< 50

TABLE IV.1 – MORES events and their implicit ratings weights.

   This logic leads to a list of equation that look like an optimization problem. We deduce the formula of calculating implicit ratings as follow :

   The result will be normalized to fit in a scale of 1 to 10 to be at the same scale of the MovieTweetings ratings.

3. **Viewing the results**

   The last step is visualising the results. The figure IV.6 shows a snippet of the collector_log table in the database. It represents some of the events recorded that user 37403409281 has made.

Data Output | Explain | Messages | Notifications

	id [PK] integer	created timestamp with time zone	user_id character varying (16)	content_id character varying (16)	event character varying (200)	session_id character varying (128
1	400571	2020-08-24 22:49:52.676214+00	37403409281	10230436	save_for_later	23b1493f-e64e-11e
2	400570	2020-08-24 22:49:52.674211+00	37403409281	10230436	details	23b1493f-e64e-11e
3	400569	2020-08-24 22:49:51.43034+00	37403409281	10230436	details	23b1493f-e64e-11e
4	400568	2020-08-24 22:39:10.992108+00	37403409281	10307440	details	23b1493f-e64e-11e
5	400567	2020-08-24 22:39:10.990109+00	37403409281	10307440	save_for_later	23b1493f-e64e-11e
6	400566	2020-08-24 22:39:08.195438+00	37403409281	10307440	details	23b1493f-e64e-11e
7	400565	2020-08-24 22:36:13.923248+00	37403409281	10003008	details	23b1493f-e64e-11e
8	400564	2020-08-24 22:36:13.921245+00	37403409281	10003008	save_for_later	23b1493f-e64e-11e
9	400563	2020-08-24 22:36:12.502953+00	37403409281	10003008	details	23b1493f-e64e-11e
10	400562	2020-08-24 22:14:23.6373+00	37403409281	10198072	details	23b1493f-e64e-11e
11	400561	2020-08-24 22:11:17.726502+00	37403409281	10149080	details	23b1493f-e64e-11e
12	400560	2020-08-24 22:11:17.724504+00	37403409281	10149080	save_for_later	23b1493f-e64e-11e
13	400559	2020-08-24 22:11:17.541788+00	37403409281	10149080	details	23b1493f-e64e-11e
14	400558	2020-08-24 22:11:17.539789+00	37403409281	10149080	save_for_later	23b1493f-e64e-11e
15	400557	2020-08-24 22:11:16.390845+00	37403409281	10149080	details	23b1493f-e64e-11e

FIGURE IV.6 – Snippet of the collector_log table.

   In the screenshot in figure IV.7 you will see the corresponding ratings in the site. The movie who receives a ratings of 5/10 has multiple “details” entries and the user

also clicked on " save for later" which signifies that he is interested on that content while the movie with a rating of 2.5/10 has only on entry in collector_log which is only a pop over on details once.

FIGURE IV.7 – Implicit ratings of a specific user.

IV.2.8 The recommendation builder

As mentioned before, the builder is responsible of all the recommendation algorithms. It pre-calculates the recommendation using the different machine learning python libraries. We have implemented multiple algorithms in order to have a global overview of the different options possible in the RS field : association rules, collaborative filtering, content-based filtering and we combine all of these to obtain a hybrid filtering.

IV.2.8.1 Association rules

The idea of association rules is that an item, a product or an article is used as an input to find other relevant content. Recommendations are made based on items that are bought (or watched in our case) together.
An association rules consists of an antecedent and a consequent where both represent a list of items (figure IV.8).

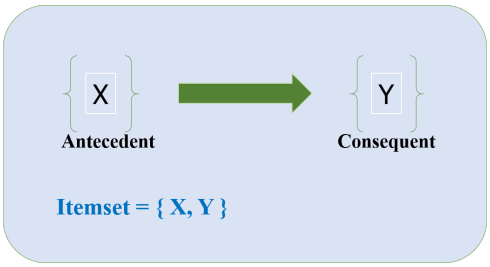

FIGURE IV.8 – Association rules components.

The strength of an association rule can be measured using two metrics : support and confidence.

- **Support**
  Support gives how often a rule is applicable to a given a dataset.

$$Support(X \rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Total number of transactions}}$$

- **Confidence** It determines how frequently item Y appears in transactions that contain X.

$$Confidence(X \rightarrow Y) = \frac{\text{Transactions containing both X and Y}}{\text{Transactions containing X}}$$

In MORES, transactions are the "watch" events happening for the same session (userID). We elaborate association rules by following the steps depicted in the figure IV.9 :

1. Retrieve "watch" events from the database (the collector log table).
2. Build the transactions : we group the events by transaction.
3. Calculate association rules using the previous metrics.
4. Save association rules in the database to display them on MORES.

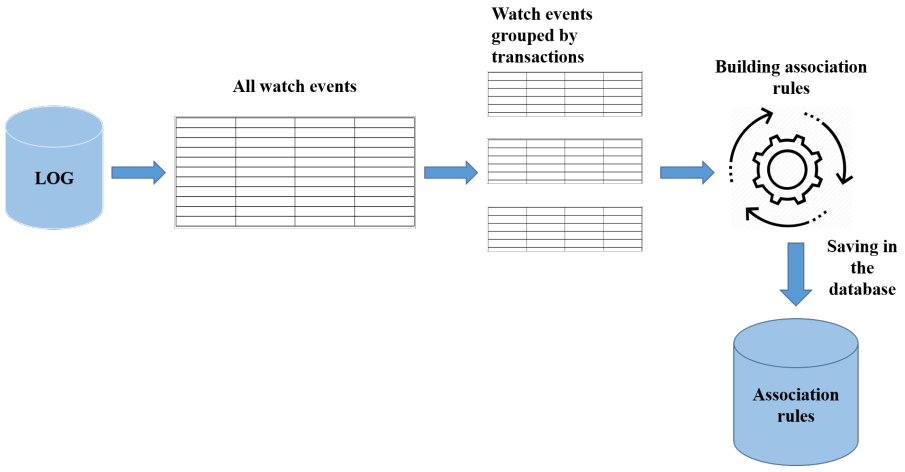

FIGURE IV.9 – Association rules steps.

IV.2.8.1.1 Cold start problem and association rules

The figure IV.10 show how association rules are implemented to handle cold start. When a new user enters the system we don't know much about, it becomes difficult to recommend content for him. This is the cold start problem. To address this problem we use association rules by creating a new method that queries the database to retrieve the content the user has interacted with. Then, we create a dictionary after applying association rules on the content found previously. The result is ordered by average confidence and displayed to the user.

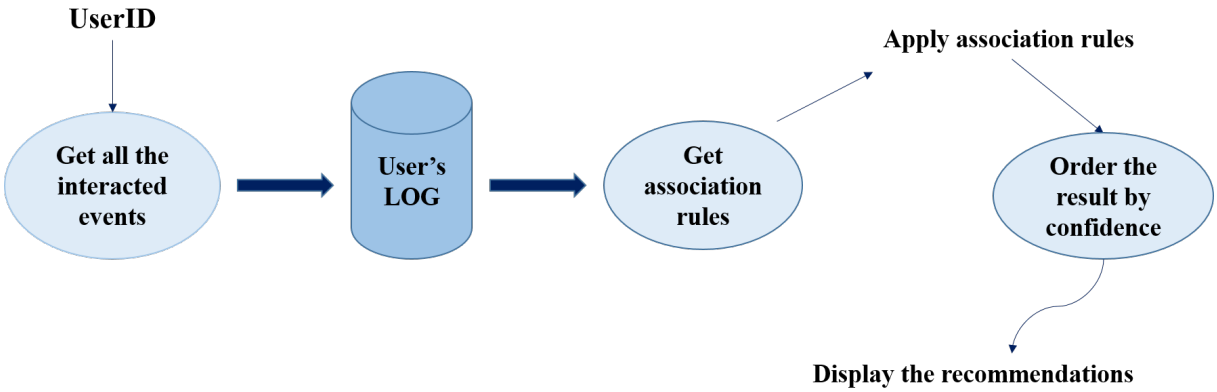

FIGURE IV.10 – Handling cold start with association rules.

IV.2.8.2 Collaborative filtering

Collaborative filtering recommends a list of items for a user based on the similarity between him and other users or other items. In our system, we have implemented neighbourhood based collaborative filtering (see figure IV.11). As mentioned in chapter two, neighbourhood filtering can be done in two ways : item based or user based filtering. For MORES we have chosen item based method for the reasons above :

- The number of users are greater than the number of ratings in our dataset
- Item-based predictions are more accurate than user based
- Item's neighbourhood changes much slower than user based therefore it is much easier to manage

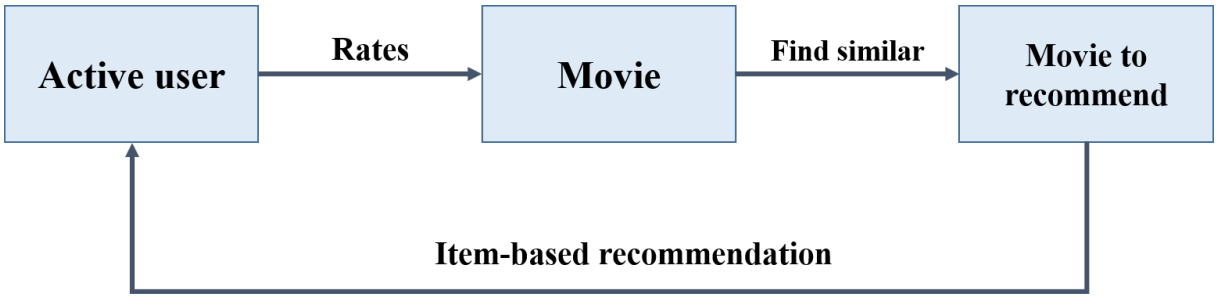

FIGURE IV.11 – Item based recommendation.

To build this recommendation method we have gone through three steps : calculate similarity between items that matches the user's tastes, select the neighbourhood for these items and finally calculate predicted rating for recommendation.

1. **Calculating similarities**
   In order to calculate similarity between items we have chosen to use the cosine similarity. This function provides a matrix of similarities were it takes a movie and comes up with a list similar movies.
2. **Selecting the neighbourhood**
   In this step, we are simply looking to the set of movies that are similar to the active content (the movie rated by the user). The small distance between the similar movies is called neighbourhood.

To do so we use the threshold method. By specifying a constant (the threshold constant), we only take movies of the neighbourhood where the similarity is above that constant. Deciding the value of the threshold is done after multiple tests for our system it is equal to 0.5.

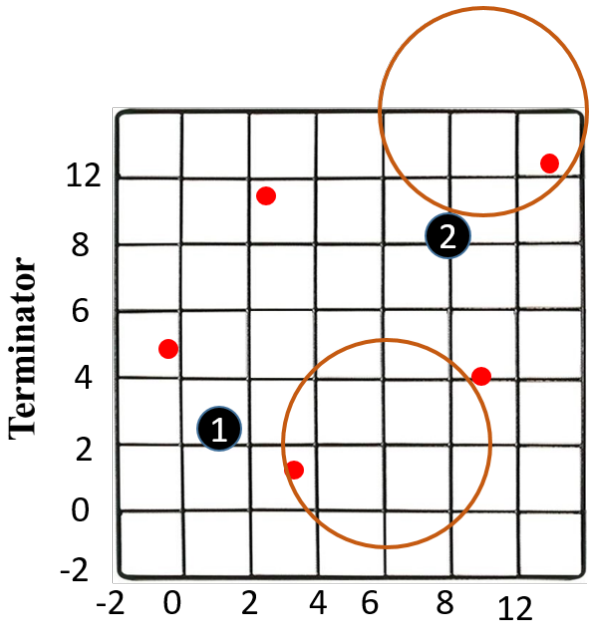

FIGURE IV.12 – Similarity threshold neighbourhood.

The figure IV.12 shows how the threshold method works. Around the active points, a circle is drawn (for the example) and every movie that's inside forms a neighbour. It represents the distance between the current movies rated by the user and its neighbours based on the threshold constant.

3. **Calculating predictions**

   In order to calculate recommendations we have used the prediction algorithm called regression. For each similarity the aim is to predict a rating for the target movie :

   - Find movies similar to the movie rated by the user (we use the movies that are in the neighbourhood).
   - Create a weighted average for each of the items : By summing the product of each similarity by the user's rating, and divide it by the sum of the similarities. This creates a weighted average to make predictions.

$$pred = \frac{\sum(\text{each similarity} \times \text{user's rating})}{\sum(\text{all the similarities})}$$

   - Make the recommendations : With the weighted average calculated we obtain the predicted ratings for the movies that are similar to the movie that the active user has already rated. With these predictions we can recommend movies for the user.

IV.2.8.3 Hybrid filtering

To take advantage of the strenghths of the two recommendation types implemented, we have implemented a hybrid filtering recommendation by mixing association rules and collaborative filtering techniques. This results to better recommendations.

IV.3 Conclusion

In this chapter, we have reviewed our approach of building our ontology-based RS. The system architecture, the different phases and steps of implementation of the approach were presented in this chapter.
The next chapter will define the experimentation environement and present different screenshots of our system.

Chapitre V

Experimentation

V.1 Introduction

This chapter provides implementation details. We first review the design and specification of MORES and the hardware and software environment used in the development of our system. Afterwards, we will use the architecture of our system presented in the previous chapter as a basis for a case study. The purpose of this chapter is to unfold the main aspects of our architecture, in order to show the feasibility and the highlighting of our ideas.

V.2 Design and specification

Our system is composed of a main page that is shown to all the visitors. Each movie has its own page in order to read their descriptions and/or watch them. Finally, in order to show how the different algorithms are implemented we added an analytics page which will show the statistics and graphics of our system.

- The main page of the site should show visitors the following :
  - An area with some movies
  - An overview of each film, without leaving the page
  - Recommendations as personal as possible
  - A menu containing a list of genres
- The page of each movie will contain :
  - Movie poster
  - Movie description
  - Movie rating
- Each genre should have :
  - The same page as the main page
  - Recommendations specific to the category
- The analytics page contains the following :
  - Statistics and graphics about the site

V.3 Experimentation environment

V.3.1 Hardware environment

All the experiments were carried out with an Intel Core i3 CPU with a frequency of 2.10 GHz and 12 GB of memory running under the Windows 10 platform.

V.3.2 Software environment

We have implemented our web application using the Django framework with HTML and Bootstrap. The machine learning algorithms used for calculating the recommendations and the analysis part (the chart and graphics) were written in python. Our database setup uses PostGreSQL. We have also used the MovieTweetings dataset and the poster images provided by the themoviedb.org API.

V.3.2.1 Python

IBM's machine learning department considers that Python is the most popular language for ML and based it on a trend search results on indeed.com. We cite here the main reason of its popularity in the machine learning field :

- Python offers great choice of libraries. A library is a module or a group of modules published by different sources like PyPi which include a pre-written piece of code that allows users to reach some functionality or perform different actions.
  ML requires continuous data processing, and Python's libraries let you access, handle and transform data [58].
  These are some of the libraries used in our implementation :
  - Scikit-learn for handling basic ML algorithms like clustering, linear and logistic regressions, regression, classification, and others.
  - Pandas for high-level data structures and analysis. It allows merging and filtering of data, as well as gathering it from other external sources like Excel, for instance.
  - NLTK for working with computational linguistics, natural language recognition, and processing.
- Python programming language resembles the everyday English language, and that makes the process of learning easier. Its simple syntax allows you to comfortably work with complex systems, ensuring clear relations between the system elements.
- Python is very flexible, programmers can combine python and other languages to reach their goals. There's also no need to recompile the source code, developers can implement any changes and quickly see the results.

V.3.2.2 Django

Django is a high level Python framework, allowing rapid development of secure and maintainable websites. It is a **built-in template** language that facilitates the process of building applications. It is free,**open source**, has an **active community**, good documentation, and several options for free. Django is also **scalable** : it can handle traffic and mobile app API usage of more than 400 million+ users helping maximize scalability and minimize web hosting costs [59]. Another advantage in using Django is because it is based on the **MVT (Model-View-Template) architecture**. MVT is a software design pattern for developing a web application. It has the following three parts [59] :

-Model : The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySql, Postgres).

-View : The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript files.

-Template : A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

The MVT architecture brings **modularity** to Django. Django also hides your website's source code. The framework has protection against XSS and CSRF attacks, SQL injections, clickjacking, etc. Django notifies of a number of common **security** mistakes better than PHP. The figure V.1 resumes the different advantages cited before.

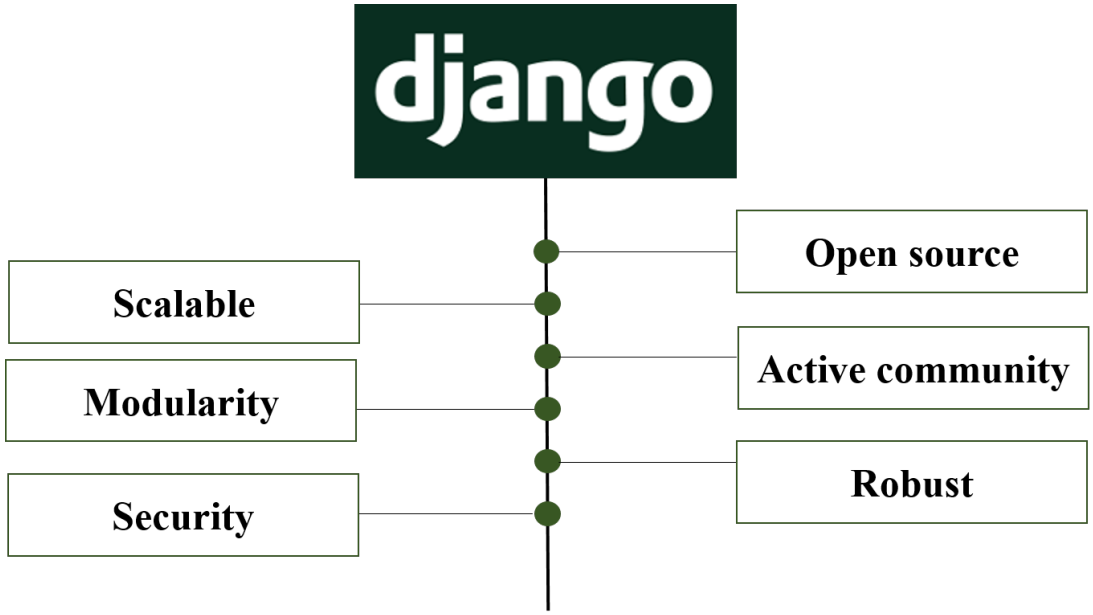

FIGURE V.1 – Django advantages.

V.3.2.3 PostGreSQL

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads [60]. It is on top of the game when it comes to CSV support. It provides different commands like 'copy to' and 'copy from' which help in the fast processing of data. Postgres can be used on Linux, BSD, Solaris and also Windows.

V.3.3 Test basis

V.3.3.1 MovieTweetings

MovieTweetings is a dataset consisting of ratings on movies that were contained in well-structured tweets on Twitter. It is always up to date and includes the most recent movies. This dataset is the result of research conducted by Simon Dooms (Ghent University, Belgium) and has been presented on the CrowdRec 2013 workshop which is co-located with the ACM RecSys 2013 conference [61]. The dataset consists of three files : **users.dat**, **items.dat** and **ratings.dat**. The table V.1 lists the number of each component of the MovieTweetings dataset that we have used in our implementation.

- **users.dat** : Contains the mapping of the users ids on their true Twitter id in the following format : userid : :twitter_id. For example : 1 : :177651718.
- **items.dat** : Contains the items (i.e., movies) that were rated in the tweets, together with their genre metadata in the following format : movie_id : :movie_title (movie_-year) : :genre|genre|genre. For example : 0110912 : :Pulp Fiction (1994) : :Crime|Thriller. The file is UTF-8 encoded to deal with the many foreign movie titles contained in tweets.
- **ratings.dat** : In this file the extracted ratings are stored in the following format : user_id : :movie_id : :rating : :rating_timestamp.
  For example : 14927 : :0110912 : :9 : :1375657563.
  The ratings contained in the tweets are scaled from 0 to 10. To prevent information loss we have chosen to not down-scale this rating value, so all rating values of this dataset are contained in the interval [0, 10].

Metrics	Value
Total number of ratings	883,182
Number of unique users	68,822
Number of unique items	36,193

TABLE V.1 – MovieTweetings dataset stats.

V.3.3.2 TheMovieDB API (TMDB API)

The API provides a fast, consistent and reliable way to get third party data. It is free to use. The posters of all the movies in our application is provided from the TMDB API. Thus, MORES uses the combination of several datasets and web services available for free on the web (see figure V.2), such as MovieTweetings dataset and the IMDB API for posters and movies descriptions.

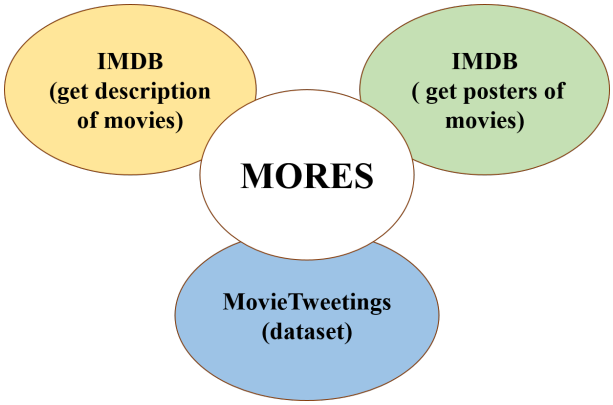

FIGURE V.2 – Representation of MORES interaction in the web.

V.4 Implementation

This section describe the different part implemented of the recommendation system. We will present how the recommendation process phase is implemented. The ontological part due to its complexity is still in the development phase and will not be included in the following.
This section will also present the different interfaces of MORES : the home page, movie details page, analytics page and clustering results.

V.4.1 Home page

This home page (see figure V.3) allows users to view all the movies on the site. It contains a search bar at the top of the page, a gender list on the left side that filters the movies by their genre, a "top 10 movies" list that represents our non-customized recommendation and a personalized recommendation section.

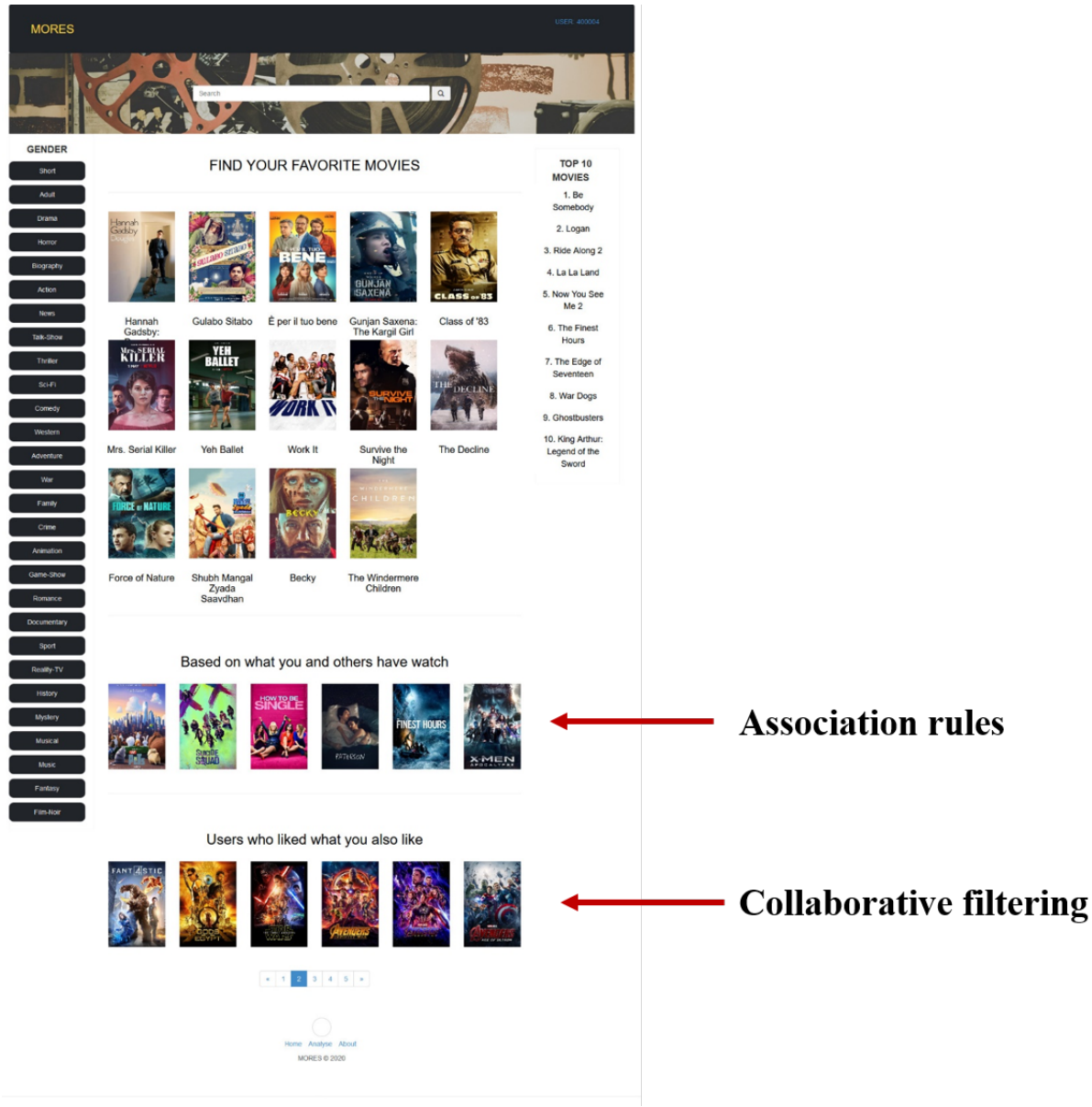

FIGURE V.3 – Home page of MORES.

V.4.2 Movie details page

The figure V.4 presents information and details about the film such as : date of production, genre and rating ...etc.

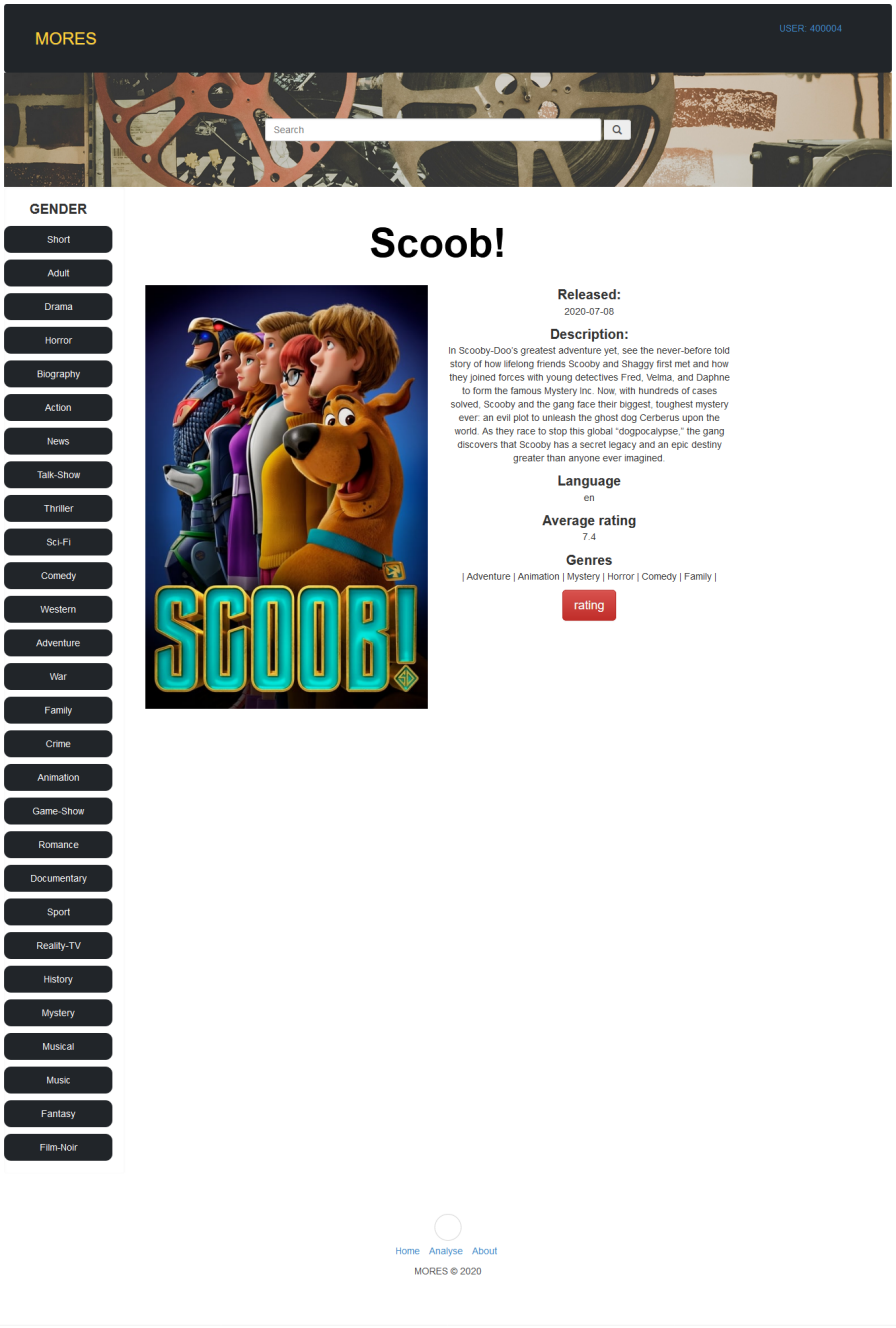

FIGURE V.4 – Movie details example.

V.4.3 Analytical page

The analytical page of MORES gives an overview of the different interaction of the active user with the system and also tells us in which cluster the user belongs (see figure V.5).

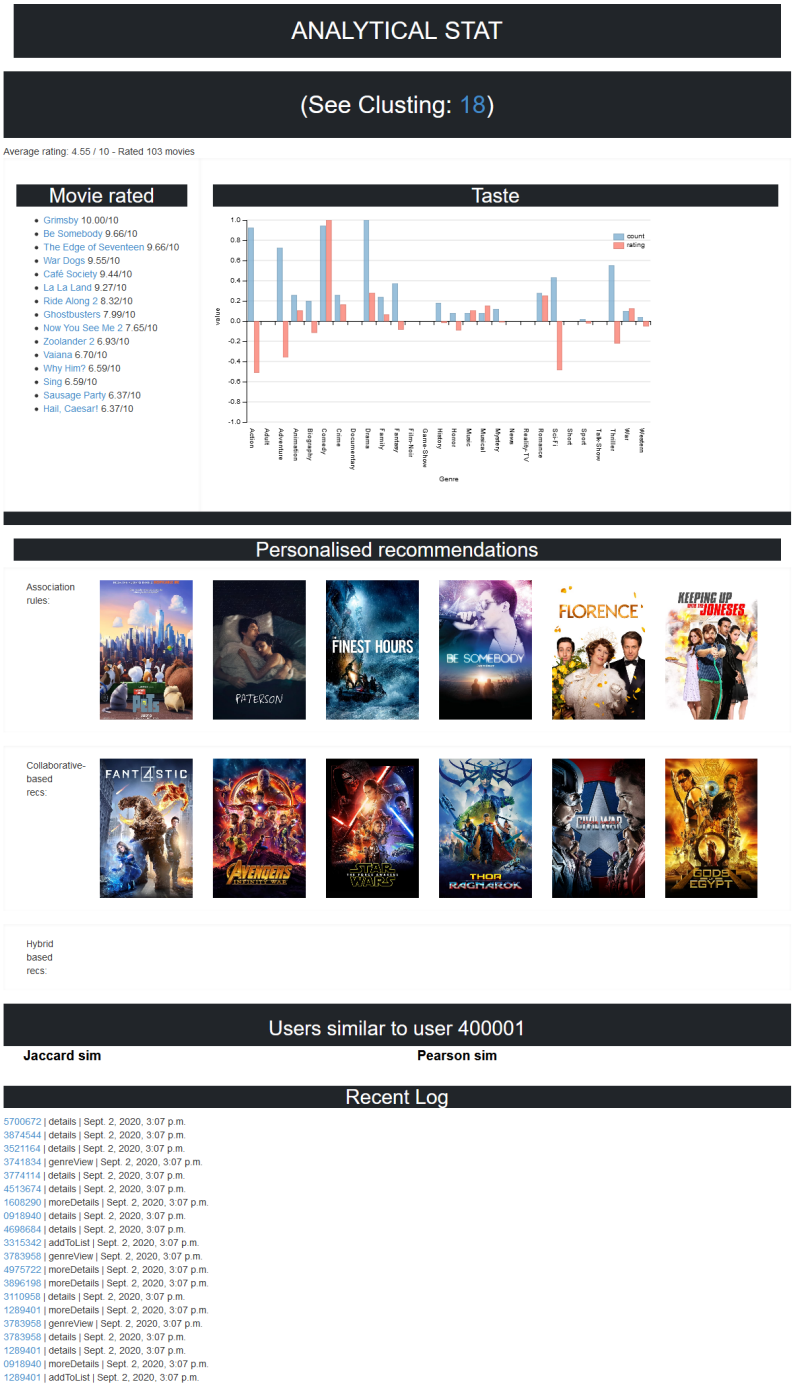

FIGURE V.5 – Analytical page of MORES.

The first row is depicted in the figure V.6 .It represents the results of the implicit ratings algorithms where the movie rated list represent the list of the movies that the user has rated and on the right side a graph that represent the user's taste within the different genre of movies available on MORES.

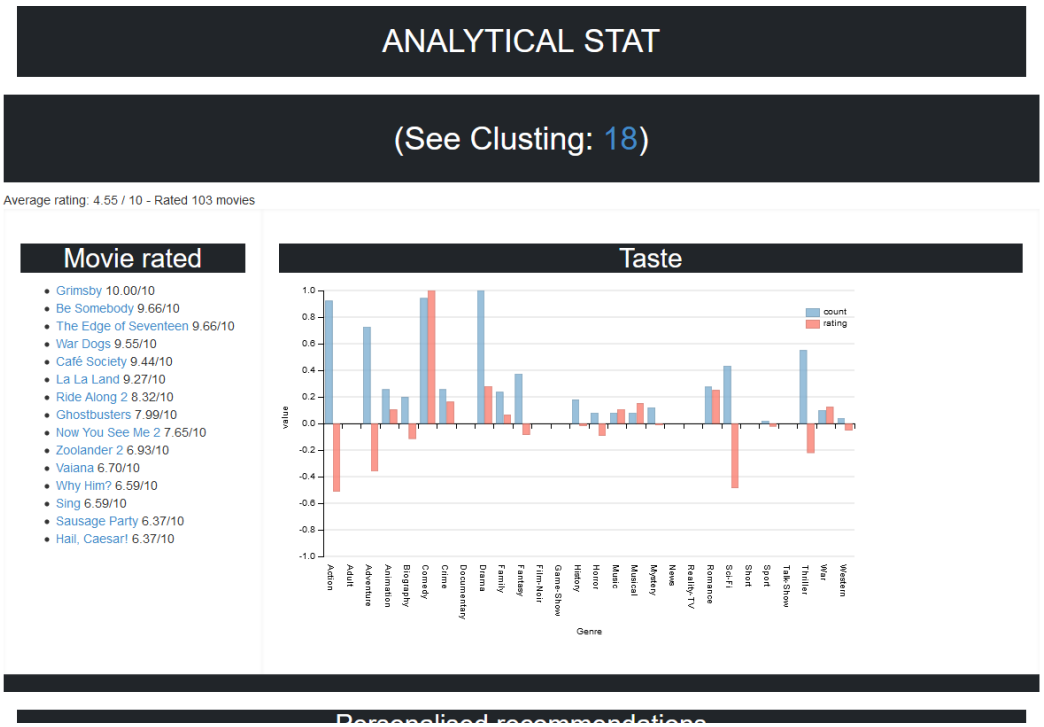

FIGURE V.6 – Analytical statistics : implicit ratings representation.

Figure V.7 show the second row where we can see the different recommendations in action (association rules, collaborative filtering and hybrid filtering).

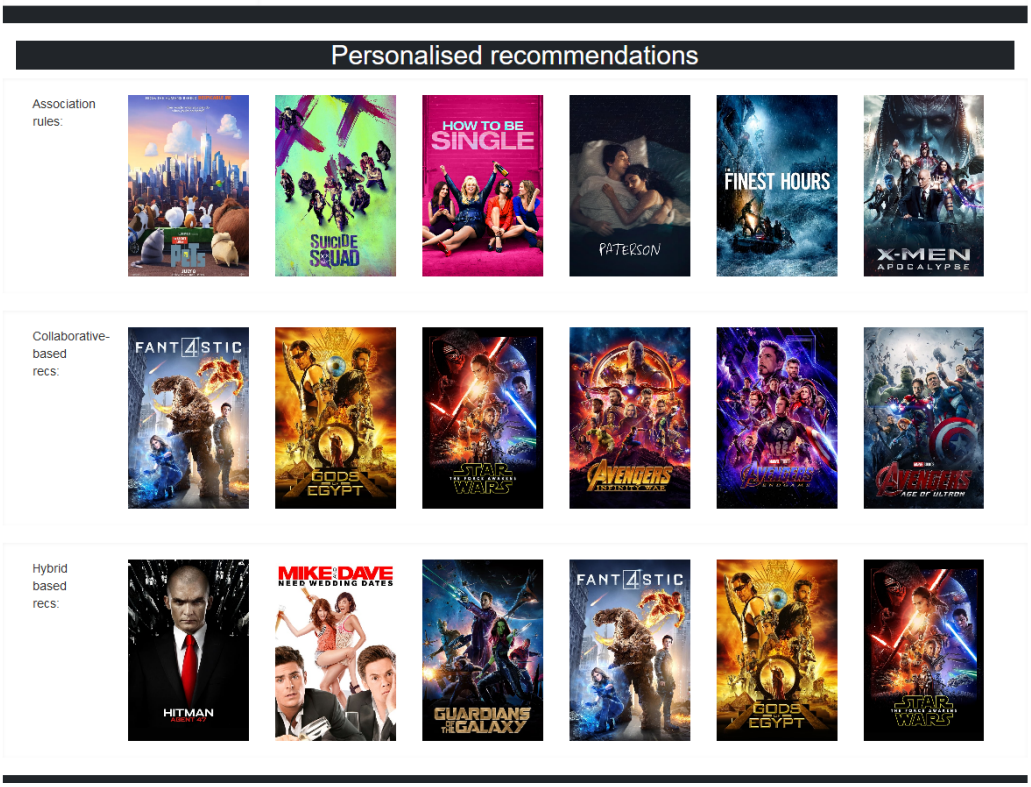

FIGURE V.7 – The different personnalised recommendations.

V.5 Conclusion

In this chapter, we have overviewed the design and specification of our system. Experimentation environment were then enumarated through hardware and software environment. We also described the dataset used for building MORES and finished by presenting the different interfaces of the site.

Chapitre VI

Conclusion

The problem addressed in this thesis is centered on the problem of the movies recommendation with all the information related to it. The research field of recommendation systems is rich and multidisciplinary. It includes several areas such as : search and filtering, data mining, personalization, social networking, word processing and user interaction. In addition, current research in recommendation systems is having a strong impact in the industry, resulting in many practical applications. We conclude that the problem of movie recommendation and recommendation systems as a whole is still far from being solved and is attracting a lot of interest in both academia and industry.

More specifically, in Chapter 2, we reviewed three classes of recommendation systems : systems based on collaborative filtering, systems based on content-based filtering, and finally hybrid systems. We found that these solutions are not completely efficient and need to be improved with other techniques in order to fully achieve the best recommendations to the user.

In order to understand the different challenges that recommendation systems face we have studied and analyzed different systems with various conceptions and approaches and resumed our research in the related work of the Chapter 3. This chapter also introduced the concept of ontology which many papers used to build a better recommendation systems and improve word processing most of the time. The importance of using web semantic and web services in general has proven its efficiency.

All this research work has inspired us to design MORES, our solution for movies recommendation. By making our own improvements, we were able to present a novel hybrid recommender system using ontology and machine learning. Ontology is used as a semantic knowledge model in order to create a user profile that reflect exactly the user preferences in our system. For the recommendation process, we passed in review the different machine learning algorithms to build the most efficient recommendations we could. Because all the users does not always obviously interact with the system, we choose to implement implicit ratings algorithms that performs recommendations according to the different im-

plicit interactions of the user with the system : clicking on a button, overviewing a film poster..etc. Implicit ratings could be built because we implemented a collector that tracked every user interaction and stored it in the database for machine learning purposes. On the other hand, we calculated similarities between users with Jaccard and Pearson similarity and put the users with the same similarity in a cluster by using the K-means clustering. These techniques allows us to recommend movies by matching preferences of different users. We have also implemented collaborative filtering recommendation with the technique named the neighbourhood filtering. Neighbourhood filtering has allowed us to calculate similarity between items and make prediction on the rating that the user will give based on how similar two movies where in the neighbouhood and then recommend movies with the highest score of prediction. Another machine learning techniques used in MORES was the association rules. Association rules make recommendations based on events happening for the same user. To fully take advantages of these different technique, we have finally implemented a hybrid recommendation where we mixed association rules with collaborative filtering to have more accurate recommendations.

In this thesis, we presented the methods of recommendation for a single user. Another type of recommender also exists (based on groups). These methods focus on providing recommendations to a group of users, trying to maximize the overall satisfaction of the group.
Despite our efforts we were not able to build the ontological part explained in our approach. Thus, our first perspective as futur work is to complete our implementation for the processing of user profiles. Secondly, we would like to add a content based filtering in our system run it separately and then add it into the hybrid filtering already implemented to have an even better recommendation.

Bibliography

[1] Toby Segaran. *Programming collective intelligence*. O'Reilly Media, Inc., 2007, p. 360. isbn : 9788578110796. url : https://www.amazon.com/Programming-Collective-Intelligence-Building-Applications-ebook/dp/B00F8QDZWG.

[2] Robin Burke. "Hybrid recommender systems : Survey and experiments". In : *User Modelling and User-Adapted Interaction* 12.4 (2002), p. 331-370. issn : 09241868. doi : 10.1023/A:1021240730564.

[3] Gediminas Adomavicius et Alexander Tuzhilin. *Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions*. 2005. doi : 10.1109/TKDE.2005.99.

[4] Upendra Shardanand et Pattie Maes. "Social information filtering : algorithms for automating 'word of mouth'". In : *Conference on Human Factors in Computing Systems - Proceedings*. T. 1. 1995, p. 210-217.

[5] Marko Balabanović et Yoav Shoham. "Content-Based, Collaborative Recommendation". In : *Communications of the ACM* 40.3 (1997), p. 66-72. issn : 00010782. doi : 10.1145/245108.245124.

[6] Gawesh Jawaheer, Martin Szomszor et Patty Kostkova. "Comparison of implicit and explicit feedback from an online music recommendation service". In : *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2010, Held at the 4th ACM Conference on Recommender Systems, RecSys 2010* (2010), p. 47-51. doi : 10.1145/1869446.1869453.

[7] Yoon Ho Cho, Jae Kyeong Kim et Soung Hie Kim. "A personalized recommender system based on web usage mining and decision tree induction". In : *Expert Systems with Applications* 23.3 (2002), p. 329-342. issn : 09574174. doi : 10.1016/S0957-4174(02)00052-0.

[8] Bracha Shapira, Peretz Shoval et Uri Hanani. "Stereotypes in information filtering systems". In : *Information Processing and Management* 33.3 (1997), p. 273-287. issn : 03064573. doi : 10.1016/S0306-4573(97)00003-4.

[9] Tommaso Di Noia et Vito Claudio Ostuni. “Recommender systems and linked open data”. In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2015. isbn : 9783319217673. doi : 10.1007/978-3-319-21768-0_4.

[10] Weng Li-Tung et al. “Exploiting item taxonomy for solving cold-start problem in recommendation making”. In : *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*. 2008. isbn : 9780769534404. doi : 10.1109/ICTAI.2008.97.

[11] Paul Resnick et Hal R. Varian. “Recommender Systems”. In : *Communications of the ACM* 40.3 (1997), p. 56-58. issn : 00010782. doi : 10.1145/245108.245121.

[12] Robin Burke, Alexander Felfernig et Mehmet H. Göker. “Recommender systems : An overview”. In : *AI Magazine* (2011), p. 13-18. issn : 07384602. doi : 10.1609/aimag.v32i3.2361.

[13] Francesco Ricci, Lior Rokach et Bracha Shapira. “Introduction to Recommender Systems Handbook”. In : *Recommender Systems Handbook*. Springer US, 2011, p. 1-35. doi : 10.1007/978-0-387-85820-3_1.

[14] David Goldberg et al. “Using collaborative filtering to Weave an Information tapestry”. In : *Communications of the ACM* 35.12 (1992), p. 61-70. issn : 15577317. doi : 10.1145/138859.138867.

[15] Robin Burke. “Hybrid web recommender systems”. In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. T. 4321 LNCS. 2007, p. 377-408. isbn : 3540720782. doi : 10.1007/978-3-540-72079-9_12.

[16] K. Nageswara Rao. “Application Domain and Functional Classification of Recommender Systems—A Survey”. In : *DESIDOC Journal of Library & Information Technology* 28.3 (2008), p. 17-35. issn : 09740643. doi : 10.14429/djlit.28.3.174.

[17] Xiaoyuan Su et Taghi M. Khoshgoftaar. “A Survey of Collaborative Filtering Techniques”. In : *Advances in Artificial Intelligence* (2009), p. 1-19. issn : 1687-7470. doi : 10.1155/2009/421425.

[18] Achin Jain et Vanita Jain. “A Literature Survey on Recommendation System Based on Sentimental Analysis”. In : *Advanced Computational Intelligence : An International Journal (ACII)* 3.1 (jan. 2016), p. 25-36. doi : 10.5121/acii.2016.3103.

[19] Lalita Sharma et Anju Gera. “A Survey of Recommendation System : Research Challenges”. In : *International Journal of Engineering Trends and Technology* 4.5 (2013), p. 1989-1992. issn : 2231-5381.

[20] Marwa Hussien Mohamed, Mohamed Helmy Khafagy et Mohamed Hasan Ibrahim. "Recommender Systems Challenges and Solutions Survey". In : *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering, ITCE 2019*. Institute of Electrical et Electronics Engineers Inc., fév. 2019, p. 149-155. isbn : 9781538652602. doi : 10.1109/ITCE.2019.8646645.

[21] Charu C. Aggarwal. *Recommender Systems The Textbook*. 2016, p. 8-21. isbn : 9783319296579. doi : 10.1007/978-3-319-29659-3_3.

[22] Poonam B.Thorat, R. M. Goudar et Sunita Barve. "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System". In : *International Journal of Computer Applications* 110.4 (2015), p. 31-36. doi : 10.5120/19308-0760.

[23] Sebastian Proksch, Veronika Bauer et Gail C. Murphy. "How to build a recommendation system for software engineering". In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. T. 8987. Springer Verlag, 2015, p. 1-42. isbn : 9783319284057. doi : 10.1007/978-3-319-28406-4_1.

[24] Jiawei Han, Micheline Kamber et Jian Pei. *Data Mining : Concepts and Techniques*. Elsevier Inc., 2012. isbn : 9780123814791. doi : 10.1016/C2009-0-61819-5.

[25] Tranos Zuva et al. "A Survey of Recommender Systems Techniques , Challenges and Evaluation Metrics". In : *International Journal of Emerging Technology and Advanced Engineering* 2.11 (2012), p. 382-386.

[26] Shah Khusro, Zafar Ali et Irfan Ullah. "Recommender systems : Issues, challenges, and research opportunities". In : *Lecture Notes in Electrical Engineering*. T. 376. Springer Verlag, 2016, p. 1179-1189. isbn : 9789811005565. doi : 10.1007/978-981-10-0557-2_112.

[27] Thomas R. Gruber. "A translation approach to portable ontology specifications". In : *Knowledge Acquisition* 5.2 (1993), p. 199-220. issn : 10428143. doi : 10.1006/knac.1993.1008.

[28] N Guarino. "Formal ontology in information systems". In : *Formal Ontology in Information Systems. IOS Press*. 1998. url : https://www.google.com/books?hl=pt-PT%7B%5C&%7Dlr=%7B%5C&%7Did=Wf5p3%7B%5C_%7DfUxacC%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PR5%7B%5C&%7Ddq=lexical+semantics+and+formal+ontologies+pustejovsky%7B%5C&%7Dots=nmTB%7B%5C_%7DWsyGK%7B%5C&%7Dsig=QlNEyVPYY9PAIk-HaIlc5KD9twE.

[29] G. Van Heijst, A. Th Schreiber et B. J. Wielinga. "Using explicit ontologies in KBS development". In : *International Journal of Human Computer Studies* (1997). issn : 10715819. doi : 10.1006/ijhc.1996.0090.

[30] Yu Du et al. "Apports des ontologies aux systèmes de recommandation : état de l'art et perspectives". In : (2019), p. 15.

[31] Lingling Meng, Runqing Huang et Junzhong Gu. “A Review of Semantic Similarity Measures in WordNet”. In : *International Journal of Hybrid Information Technology* 6.1 (2013), p. 1-12. issn : 1738-9968.

[32] Guillaume Saint cirgue. *Apprendre le machine learning en une semaine*. 2019, p. 100.

[33] Mohssen Mohammed, Muhammad Badruddin Khan et E. B. M. Bashier. *Machine Learning : Algorithms and Applications*. T. 7. 13. Dordrecht : Springer Netherlands, 2016, p. 2-11. isbn : 9781498705387. doi : 10.1007/978-94-017-2221-6_5. url : http://link.springer.com/10.1007/978-94-017-2221-6%7B%5C_%7D5.

[34] Lior Rokach et Oded Maimon. “Top-down induction of decision trees classifiers - A survey”. In : *IEEE Transactions on Systems, Man and Cybernetics Part C : Applications and Reviews* 35.4 (2005), p. 476-487. issn : 10946977. doi : 10.1109/TSMCC.2004.843247.

[35] Petr Berka et Jan Rauch. “Machine Learning and Association Rules”. In : *19th International Conference On Computational Statistics COMPSTAT 2010* 2010 (2010), p. 1-29. url : https://www.rocq.inria.fr/axis/COMPSTAT2010/TU%7B%5C_%7DBerka-Rauch%7B%5C_%7Dpaper.pdf.

[36] Trupti A. Kumbhare et Santosh V. Chobe. “An Overview of Association Rule Mining Algorithms”. In : *International Journal of Computer Science and Information Technologies* 5.1 (2014), p. 927-930.

[37] Pierre Gancarski, Antoine Cornuéjols et Younès Bennani. “Clustering collaboratif : Principes et mise en oeuvre”. In : *BDA (Gestion de Données — Principes, Technologies et Applications)*. 2017, p. 13.

[38] Samreen Zehra et al. “Ontology-based sentiment analysis model for recommendation systems”. In : *IC3K 2017 - Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. T. 2. 2017, p. 155-160. isbn : 9789897582721. doi : 10.5220/0006491101550160.

[39] Mehrbakhsh Nilashi, Othman Ibrahim et Karamollah Bagherifard. “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques”. In : *Expert Systems with Applications* 92 (2018), p. 507-520. issn : 09574174. doi : 10.1016/j.eswa.2017.09.058.

[40] Mohammed E. Ibrahim et al. “Ontology-Based Personalized Course Recommendation Framework”. In : *IEEE Access* 7 (2019), p. 5180-5199. issn : 21693536. doi : 10.1109/ACCESS.2018.2889635.

[41] Junmei Feng et al. “An improved collaborative filtering method based on similarity”. In : *PLoS ONE* 13.9 (2018). issn : 19326203. doi : 10.1371/journal.pone.0204003.

[42] Paul Sheridan et al. “An ontology-based recommender system with an application to the Star Trek television Franchise”. In : *Future Internet* 11.9 (2019). issn : 19995903. doi : 10.3390/fi11090182. arXiv : 1808.00103.

[43] M. S. Ayundhita, Z. K.A. Baizal et Y. Sibaroni. “Ontology-based conversational recommender system for recommending laptop”. In : *Journal of Physics : Conference Series*. T. 1192. 1. 2019. doi : 10.1088/1742-6596/1192/1/012020.

[44] Jai Prakash Verma, Bankim Patel et Atul Patel. “Big data analysis : Recommendation system with hadoop framework”. In : *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015* (2015), p. 92-97. doi : 10.1109/CICT.2015.86.

[45] Van Dai Ta, Chuan Ming Liu et Goodwill Wandile Nkabinde. “Big data stream computing in healthcare real-time analytics”. In : *Proceedings of 2016 IEEE International Conference on Cloud Computing and Big Data Analysis, ICCCBDA 2016* (2016), p. 37-42. doi : 10.1109/ICCCBDA.2016.7529531.

[46] Surabhi Dwivedi et V. S.Kumari Roshni. “Recommender system for big data in education”. In : *Proceedings - 2017 5th National Conference on E-Learning and E-Learning Technologies, ELELTECH 2017* 2 (2017). doi : 10.1109/ELELTECH.2017.8074993.

[47] Archenaa J et E.A.Mary Anita. “Health Recommender System using Big data analytics”. In : *Management Science and Business Intelligence* 13.4 (2017), p. 17-24. issn : 2070-3740. doi : 10.5281/zenodo.

[48] Ali Al-badi, Ali Tarhini et Salma Al Mayahi. *Exploring the Potential Benefits of Big Data Analytics in Providing Smart Healthcare Salma*. T. 200. Illsley 2014. Springer International Publishing, 2018, p. 16-37. isbn : 978-3-319-95449-3. doi : 10.1007/978-3-319-95450-9. url : http://link.springer.com/10.1007/978-3-319-95450-9.

[49] Dympna O Sullivan et Kedar Ratnaparkhi. “Recommender system for food in a restaurant based on Natural Language Processing and Machine Learning Kedar Ratnaparkhi Supervisor :” thèse de doct. National College of Ireland, 2018, p. 20.

[50] Antonio Jesús Fernández-García et al. “A recommender system for component-based applications using machine learning techniques”. In : *Knowledge-Based Systems* 164 (2019), p. 68-84. issn : 09507051. doi : 10.1016/j.knosys.2018.10.019.

[51] Bushra Ramzan et al. “An Intelligent Data Analysis for Recommendation Systems Using Machine Learning”. In : *Scientific Programming* 2019 (2019). issn : 10589244. doi : 10.1155/2019/5941096.

[52] Will Serrano. “Intelligent Recommender System for Big Data Applications Based on the Random Neural Network”. In : *Big Data and Cognitive Computing* 3.1 (2019), p. 15. issn : 2504-2289. doi : 10.3390/bdcc3010015.

[53] Z. Bahramiana et R. Ali Abbaspoura. “An ontology-based tourism recommender system based on Spreading Activation model”. In : *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. T. 40. 1W5. 2015, p. 83-90. doi : 10.5194/isprsarchives-XL-1-W5-83-2015.

[54] Charbel Obeid et al. “Ontology-based Recommender System in Higher Education”. In : 2018, p. 1031-1034. isbn : 9781450356404. doi : 10.1145/3184558.3191533.

[55] Petar Ristoski et Heiko Paulheim. “Semantic Web in data mining and knowledge discovery : A comprehensive survey”. In : *Journal of Web Semantics* 36 (2016), p. 1-22. issn : 15708268. doi : 10.1016/j.websem.2016.01.001.

[56] Houda El Bouhissi, Mimoun Malki et Mohamed Amine Sidi Ali. “From user’s goal to semantic web services discovery : Approach based on traceability”. In : *International Journal of Information Technology and Web Engineering* 9.3 (2014), p. 15-39. issn : 15541053. doi : 10.4018/ijitwe.2014070102.

[57] Aristidis Likas, Nikos Vlassis et Jakob J. Verbeek. “The global k-means clustering algorithm”. In : *Pattern Recognition* 36.2 (fév. 2003), p. 451-461. issn : 00313203. doi : 10.1016/S0031-3203(02)00060-2.

[58] Andrew Luashchuk. *Why I Think Python is Perfect for Machine Learning and Artificial Intelligence*. https://towardsdatascience.com/8-reasons-why-python-is-good-for-artificial-intelligence-and-machine-learning-4a23f6bed2e6. 2019.

[59] Arora Naveen. *Django Project MVT Structure*. https://www.geeksforgeeks.org/django-project-mvt-structure/. 2020.

[60] *What is PostgreSQL ?* https://www.postgresql.org/about/.

[61] Simon Dooms, Toon De Pessemier et Luc Martens. “MovieTweetings : a Movie Rating Dataset Collected From Twitter”. In : *Workshop on Crowdsourcing and human computation for recommender systems, CrowdRec at RecSys* 2013.January (2013), p. 43-44.

INTERNATIONAL CONGRESS ON

ROBOTICS & ARTIFICIAL INTELLIGENCE

Phronesis RESEARCHERS WORLD

ICRAI-2020 JUNE 22-23, 2020 | OSAKA, JAPAN

Date: February 25, 2020

Letter of Invitation

Dear Wassila AKROUCHE,
Faculty of Exact Sciences, University of Bejaia, Bejaia, 00006, Algeria.

Greetings from Phronesis LLC!!!

We cordially invite you to attend the "**International Congress on Robotics and Artificial Intelligence**" to be held during **June 22-23, 2020 at Osaka, Japan**. In this regard, on behalf of the **Phronesis LLC**, we are delighted to inform you that your research work has been accepted to present an **Invited Oral Presentation** on "**Improve Recommender systems in the era of Big Data: Deep learning based approach**" under the session **Machine Learning Methods** at **ICRAI - 2020**.

The **International Congress on Robotics and Artificial Intelligence** is being conducted at Osaka, Japan. ICRAI-2020 is a scientific congregation which brings together researchers, scientists, and key decision makers, industry professionals in the same physical space for a brief yet intense period of discussion, collaboration, and addressing related problems in research. We believe this conference will be a highly rewarding educational and networking experience for all. Additionally, we encourage you to take this opportunity to explore the many facets of Osaka and to experience the unique Japan culture.

We look forward to seeing you in **Osaka, Japan!!**

For more details about **ICRAI-2020**, have a glance PS: https://phronesisonline.com/robotics-artificial-intelligence-conference/

Regards,

PHRONESIS LLC USA

Arizona Grey, Phronesis LLC, 5 Great Valley Pkwy, STE 235, Malvern PA 9355, USA

Organizing Committee Members:

Dr. Cheng Siong Chin Newcastle University in Singapore, Singapore	**Dr. Fairouz Kamareddine** Heriot-Watt University, United Kingdom

Note:** This invitation is only to attend **ICRAI-2020** which is during **June 22-23, 2020** at **Osaka, Japan

INTERNATIONAL CONGRESS ON

ROBOTICS & ARTIFICIAL INTELLIGENCE

ICRAI-2020

JUNE 22-23, 2020 | OSAKA, JAPAN

Date: February 25, 2020

Letter of Invitation

Dear Naziha Fatma ADJALI,
Faculty of Exact Sciences, University of Bejaia, Bejaia, 00006, Algeria.

Greetings from Phronesis LLC!!!

We cordially invite you to attend the "**International Congress on Robotics and Artificial Intelligence**" to be held during **June 22-23, 2020 at Osaka, Japan**. In this regard, on behalf of the **Phronesis LLC**, we are delighted to inform you that your research work has been accepted to present an **Invited Oral Presentation** on "**Improve Recommender systems in the era of Big Data: Deep learning based approach**" under the session **Machine Learning Methods** at **ICRAI - 2020**.

The **International Congress on Robotics and Artificial Intelligence** is being conducted at Osaka, Japan. ICRAI-2020 is a scientific congregation which brings together researchers, scientists, and key decision makers, industry professionals in the same physical space for a brief yet intense period of discussion, collaboration, and addressing related problems in research. We believe this conference will be a highly rewarding educational and networking experience for all. Additionally, we encourage you to take this opportunity to explore the many facets of Osaka and to experience the unique Japan culture.

We look forward to seeing you in **Osaka, Japan!!**

For more details about **ICRAI-2020**, have a glance PS: https://phronesisonline.com/robotics-artificial-intelligence-conference/

Regards,

Arizona Grey, Phronesis LLC, 5 Great Valley Pkwy, STE 235, Malvern PA 9355, USA

Organizing Committee Members:

Dr. Cheng Siong Chin Newcastle University in Singapore, Singapore	**Dr. Fairouz Kamareddine** Heriot-Watt University, United Kingdom

Note:** This invitation is only to attend **ICRAI-2020** which is during **June 22-23, 2020** at **Osaka, Japan