Web Recommender System for Job Seeking and Recruiting

Thesis · February 2018

DOI: 10.13140/RG.2.2.26177.61286

1 author:

Lionel Ngoupeyou Tondji
African Institute for Mathematical Sciences Senegal
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Web Recommender System for Job Seeking and Recruiting

LIONEL NGOUPEYOU TONDJI (lionel.ng.tondji@aims-senegal.org)
African Institute for Mathematical Sciences (AIMS)
Senegal

Supervised by: Pr. Ndeye Niang Keita
Conservatoire National des Arts et Métiers, France

31 January 2018
Submitted in Partial Fulfillment of a Masters II at AIMS

Dedicace

To my Father, Tondji Martin

My Mother, Touomgam Tsewalo Tondji Fride

And all my Family.

Acknowledgements

It is always a pleasure to finish a research project on a high note. It gives extra pleasure while looking back working with kind, humble and wise people.

First of all, I would like to thank the Almighty God for helping me to successfully finish this project. His blessings guided me through some of the hard times in my life and showed me the correct path.

I would like to express my heartfelt gratitude to my supervisor Pr. Ndeye Niang Keita for allowing me to work with her, helping me to see it through to completion. She was very supportive, encouraging, patient during the final thesis preparation. She took the pain of reading through the initial drafts of my thesis during her busy schedule and I am greatly thankful to her for the suggestions regarding the structure and content of the final document. I thank my examiner Dr. Franck Kalala for reading through the final paper and making the necessary arrangements for the successful presentation. I would like to thank the initiator of the AIMS program in the person of Pr. Neil Turok, I also thank the president of AIMS-Senegal Pr. Aissa Wade for her many efforts, encouragement and advice that she has given me throughout this stay. I say a thank you to the administrative staff in particular to Ms. Barbara Diagne, Ms. Kadidjatou Dramé for their support. I would like also to express my gratitude to the tutors and more particularly to Dr. Lamine and Mr. Ignace Minlend, who have had the patience to direct my work to this day.

In May 2017, Mr. Momar Diop and Youssef Hassani, co-founders of the company African digital native, trusted me to carry out the internship during which was conceived this thesis subject. I would like to thank them and Mr. Pascal Elingui for allowing me to live this experience, from which I am out growing both professionally and humanely. When I arrived, I did not imagine learning as much about business life and human relations.

I would like to thank my teachers from the Department of Mathematics of the University of Yaounde 1, thanks to them I have always had confidence in me and for the love of mathematics that they transmitted to me in particular I thank Pr. Tonga Marcel, Pr. Blaise Tchapnda, Pr. Nkuimi C. Jugnia. I want to thank Dr. Emmanuel Fouotsa. I thank all my classmates for my promotion of AIMS Senegal and the University of Yaounde I for all these wonderful times we have lived together and this fraternity. I also want to say a thank you especially to all those who from near or far contributed to the writing of this Thesis.

I kept the best for last; the ones I love more than anything in the world my family, I say once again thank you for your excessive support throughout my studies.

Abstract

In general, looking for a job while scanning a lists of hiring positions on recruitment sites, which really cost a lot of time and money is an annoying thing to do Althought most of the time those jobs are not always suitable with users, or users are not satisfied. By doing this, recruiters waste their time by making sure that they are qualified or not. This thesis seeks to address a very important issue in the recruitment process which is about matching jobs seekers with jobs offers. Nowadays, the matching process between the applicant and the job offers is one of the major problems companies have to handle. Shortlisting candidates and screening resumes are long time-consuming tasks for the company, especially when 80 percent to 90 percent of the resumes received for a role are unqualified. We have designed and proposed an hybrid personalized recommender system called skillake for job seeking and online recruiting websites adapted to the cold start problem using a clustering predictive algorithms.

Keywords: Text mining, Recommender systems, Clustering, Cold-start problem, Unsupervised learning.

Résumé

En général, la recherche d'un emploi tout en scannant un poste de recrutement sur des sites de recrutements, qui coûte beaucoup de temps et d'argent est une chose agaçante à faire. Bien que la plupart du temps ces emplois ne conviennent pas toujours aux utilisateurs, ou les utilisateurs ne sont pas satisfaits. En faisant cela, les recruteurs perdent leur temps en s'assurant qu'ils sont qualifiés ou non. Cette thèse vise à aborder une question très importante dans le processus de recrutement qui consiste à faire correspondre les demandeurs d'emploi avec des offres d'emploi. À notre époque, le processus d'appariement entre le demandeur et les offres d'emploi est l'un des principaux problèmes auxquels les entreprises doivent faire face. Présélectionner des candidats et des curriculum vitae sont des tâches qui prennent beaucoup de temps pour l'entreprise, surtout lorsque 80 à 90 pourcent des CV reçus pour un rôle ne sont pas qualifiés. Nous avons conçu et proposé un système de recommandation hybride personnalisé appelé skillake pour la recherche d'emploi et des sites de recrutement en ligne adaptés au problème du démarrage à froid en utilisant un algorithme prédictif de clustering.

Mots-clés: Exploration de texte, Systèmes de recommandation, Clustering, Problème de démarrage à froid, Apprentissage non supervisé.

Declaration

I, the undersigned, hereby declare that the work contained in this essay is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

LIONEL NGOUPEYOU TONDJI, 31st January 2018

Contents

List of Tables

List of Figures

List of Acronyms

Iot	: Internet of Things
RS	: Recommender Systems, Recommendation Systems
CRM	: Customer Reletionship Management
IR	: Information retrieval
CBF	: Content-Based Filtering
KBR	: Knowledge-based recommendation
CF	: Collaborative Filtering
CV	: Curriculum Vitae
JRS	: Jobs Recommender Systems, Jobs Recommendation Systems
KBR	: knowledge-based recommendation
KNN	: K Nearest Neighbors
SVM	: Support Vector Machine
Tf-IDF	: Term frequency-Inverse document frequency
IQ	: Intelligence Quotient
PCA	: Principal Component Analysis
SVD	: Singular Value Decomposition
DBN	: Deep Belief Networks
DBSCAN	:Density-based clustering
LSH	: Locality-sensitive hashing
LDA	: Latent Dirichlet Allocation
CSV	: Comma Separated Values

1. Introduction

Recent advances in technologies such as internet of things [1] and electrical devices have rapidly grown for the last two decades allowing the creation of a large and variety repositories of information.This huge quantity of information rises new challenges from the fact that users cannot exploit available resources effectively when the amount of information requires prohibitively long user time spent on acquaintance with and comprehension of the information content. Thus, the risk of information overload of users imposes new requirements on the software systems that handle the information [2]. This problem of overload of information can be solve by using Recommender systems (RS) [3].

Recommender Systems are software system which provide the most valuable and appropriate information according to users preferences to avoid them to be overwhelmed by a large set of information.

1.1 Context and motivation

Nowadays more and more journal articles, jobs offers, books, web pages, musics and movies are created. As each new piece of information catches our attention, we are quickly overwhelmed by a huge amount of data and sometimes confronted to the problem of choices given the big number of options available, and ask for help in identifying the most interesting items, the ones that are more valuable, worthy of interest, or entertaining on which we should be spending our money and time. Many businesses including these well-known examples: LinkedIn, Amazon, Hulu, Netflix, Facebook, Twitter embed recommendation systems in their web sites, in order to study the tastes of their customers, and achieve some business objectives. Among other objectives we have:

- The increase of traffic to their web site,
- The elaboration of marketing policies tailored to their customers' tastes,
- Or simply the promotion of a given product.

The role that plays RS becomes more and more important. This was highlighted in 2006 during the competition organised by Netflix [4], a global provider of movies and TV-series. The purpose of this competition was to predict users ratings on movies based on previous ratings without any informations about the users or the movies. The price was based on the improvement of the Netflix's algorithm. The winner team has been able to improve the algorithm around ten percent and was awarded by one million dollars.

These systems also play an important role in decision-making, helping users maximize profits or minimize risks. Today, recommendation systems are used in many information-based companies. These recommendations can be used in many ways depending on the business activity. The Google company has an App called Google news [5] which recommend a list of news you may be interested in. The world's most popular online video community Youtube [6] and Netflix company recommend videos and movies you may like based on your activity on the site.

In [7], companies such as Facebook or Twitter embed recommender systems which can suggest you a list of people whom you might know, who are similar to you based on your friends list, friends of friends in your close circle, a list of people in the same geographical location as you, people posseding same

skillsets as you, groups, liked pages, and so on. The LinkedIn company [8] embed on their website recommender systems able matching job applicants with employers.

1.2 Problem Statement

Today [9], a number of platforms which make easier the recruitment process are made available to companies, from the posting of the advertisement to the management of the applications received. The use of the Internet for recruitment has grown considerably since the last two decades, resulting in a simultaneous increase in the number of recruitment channels and the volume of people that can be reached by this media. One of the popular online services for job seekers as well as for employers is E-recruitment. Nevertheless, the recruitment process is not easy, especially when it comes to searching for profiles and talents, as many approaches are currently limited to keyword research, which is no longer effective when the size of data becomes huge. In Senegal, these online recruitment platforms have become the main channel for most companies. Beyond the fact that these platform reduce the time of recruitment and the cost of advertising, they suffer from traditional methods, as a result many recruiters missed the opportunity to recruit talent and job seekers missed the opportunity to be recruited.

1.3 Methodology

In this thesis, we are interested in the stage of diffusion of the announcement, and the follow up of the recruiter during this phase [9]. However, most of the tools for analysis of the latter are limited in terms of decision support.

These limitations are due to the existence of barriers to the processing and automatic analysis of job offers. Indeed, the diversity of employment sites leads to a diversity of structures specific to these. Today there is no uniform structure accepted by all actors in the field of human resources for the information contained in job offers [9].

In this context, our work has two purposes:

- The analysis, through the restructuring of information from job offers and resume published on job boards in particular the job board Skillake to make it easiest usable for our predictive algorithms.
- The development of a predictive algorithm.

This work will lead to the development of a decision support tool for recruiters. With the developed algorithm, we will be able to provide the recruiter with a limited numbers of talents corresponding to his job offer and in the same time provide the job seeker with the ideal advertisement corresponding to his profile. To achieve this goal, our approach will require the automation of the processes used from a global point of view.

The remaining part of the document is organized as follows:

Chapter 1 provides an introduction to the internet recruitment market. Here we try to define the context and the problematic related to the recruitment market, the problems facing by recruiters and job seekers and the need for referral systems.

Chapter 2 begins with a formulation of the recommendation problem, followed by a state of the art for the recommendation systems in general, by a literature review of the differents methods used to design a recommendation system and the differents types of existing referral systems. This chapter also presents the challenges encountered when designing a recommendation system.

In Chapter 3, we give a short review on jobs recommendation systems and we make a comparative study on existing jobs recommendation systems then we present some prerequisites for the implementation of a job recommendation systems. After we present some data mining methods for recommender systems such as text mining, text processing etc.

Chapter 4, we present an overview of the data used for our model, the recommendation process used by our predictive algorithm. We have described the materials and packages used for the implementation of our recommender system, followed by a brief presentation of the results obtained by our prediction algorithm.

Chapter 5 provides a discussion about the real usage of our implementation by recruiting website, a conclusion of our work and perspectives.

In Chapter 6, we have writen an appendix which serves as mathematical background and a reference for those who want to get the python code designed throughout this thesis.

2. Literature Review on Recommendation Systems

Recommender systems are the software tools and techniques that provide suggestions. The main goal of recommender systems is to provide suggestions to online users with a highly relevant set of items so that they can make better decisions from many alternatives available over the Web [7].

2.1 Recommendation Problem

Let $U = \{u_1, u_2, \ldots, u_m\}$ a set of all users and let $I = \{i_1, i_2, \ldots, i_n\}$ a set of all possibles items that can be recommended such a jobs offers, jobs seekers, music files, images, movies etc. Here the space I can be equal to the set U and it can be very large.

The recommendation problem can be formulated [2] as follows :

Let f be a utility function that measures the usefulness of item i to user u,

$$f : U \times I \longrightarrow R$$

where R is a totally ordered set (e.g the set of ranking recommendation). Then for every user $u \in U$, we want to predict an item $i^{'} \in I$ that maximizes the user utility function. In other words:

$$\forall u \in U, i^{'}_{u} = \arg\max_{i \in I} f(u, i) \tag{2.1.1}$$

Depending to the predictive algorithm used, the input to a RS belongs to one of these categories [2]:

- Ratings (also called votes), which give the opinion of users on items
- Demographic data, which refer to user's informations, sometime difficult to get it. It is normally collected explicitly from the user
- Content data, based on content analysis

The output of a recommender system can be [2] a prediction if it is expressed as a numerical value or can be a recommendation if it is expressed in term of a list of N ranked items, where $N \leq n$.

The following part reviews the state of the art of the main approaches to designing RSs that address the problems caused by information overload.

2.2 Background of recommender systems

There exist different algorithms and methods to construct recommender systems able to do personalized recommendations or non personalized recommendations.

2.2.1 Non personalized Recommender Systems.

In [10], the most simple type of recommendation systems are the non personalized recommender systems. The recommendations produced by these systems are identical for each customer because it does not take into account the personal preferences of the users. The recommendations are independent of the customer.

These systems use differents kinds of statistical studies and in some case the product association by using external data take from other data sources, community or social networks. In e-commerce for instance the recommendation can be made based of the best seller product, the most liked or the most popular. They can also take into account the community opinion, for example the number of people who like the restaurant or the best university of the town.

In [10], aggregated opinion recommender and Basic product association recommender are the two types of algorithms used in non-personalized recommender systems.

Figure [2.1] summarizes roughly the different methods used by non-personalized recommendation systems.

Figure 2.1: A non personalized system
(Source: https://www.slideshare.net/anastasiiakornilova/recommender-systems-30843916)

Despite the fact that this method is very simple it still presents advantages and drawbacks.

Advantages

1. Easy to implement
2. Easy to collect data for these recommender systems

Disadvantages

1. Recommendations are the same for all users
2. Lack personalization

For [11], personalization concerns adapting to the individual needs, interests, and preferences of each user, from a business point of view it is a part of Customer Reletionship Management (CRM). All of the personalized recommendation approaches can be divided in these four categories which are widely used: Content-based filtering, Collaborative filtering, hybrid filtering, Knowledge-based [see Fig 2.2].

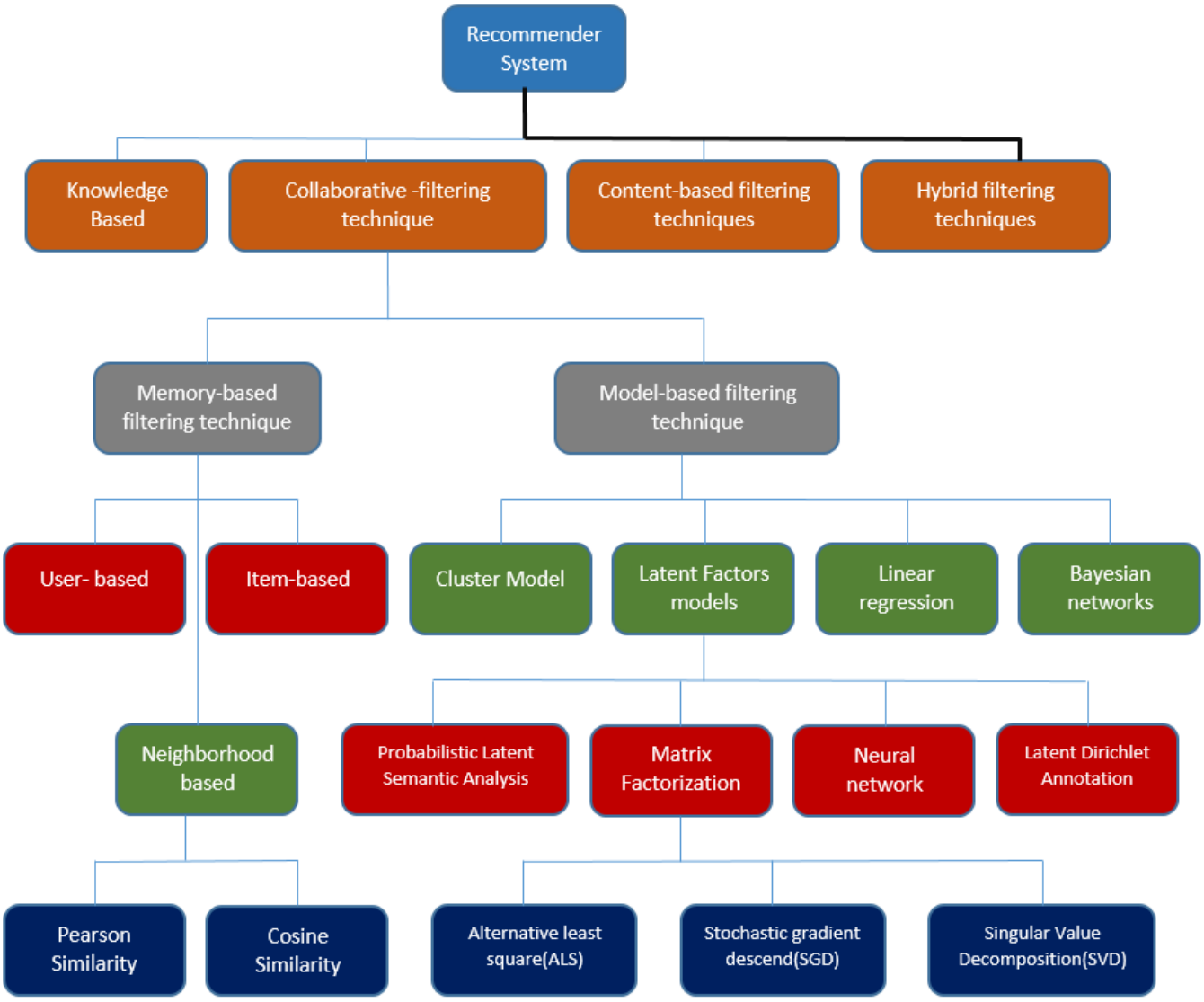

Figure 2.2: Taxonomy of Recommender Systems

The following session will give an insight into the first three approaches above mentioned.

2.2.2 Content-based recommender systems.

The recommendation process used in this method can be divided in three parts.

- **Content analyzer** : Here the main task is to represent the content of items and extract the information or specific features from the item by feature extraction techniques.
- **Profile learner** : In charge to construct the user profile after generalizing data representative collected from users preferences
- **Filtering components**: This process will try to match the features of the user profile with the features of the items. And then, the system will recommend items that fit for the user.

This system is working with data provided by the users explicitly or implicitly, from these data is constructed a profile of each user. Items are recommended to users by taking the similarity of items. In other words, the system recommends items similar to those that the user has liked in the past, and this situation is represented by the figure [2.3].

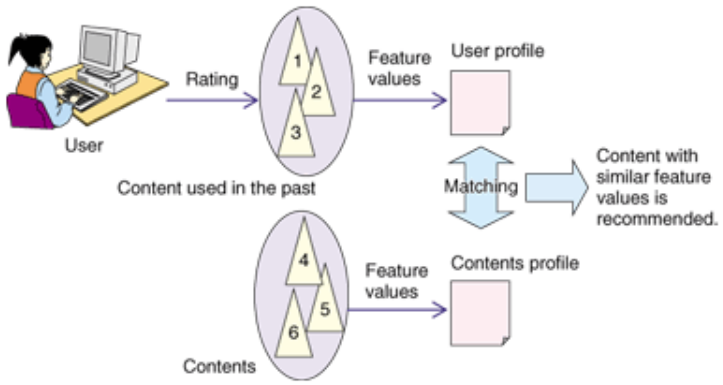

Figure 2.3: Content based filtering
(Source: http://findoutyourfavorite.blogspot.sn/2012/04/content-based-filtering.html)

The accuracy of the recommandation increase with the quantity and the quality of informations provided by users.

According to [8], content-based filtering is using the technique to analyze a set of documents and descriptions of items previously rated by a user, and then build a profile or model of the users interests based on the features of those rated items. Using the profile, the recommender system can filter out the suggestions that would fit for the user.

This modeling user approach is called content based learning and it take into account the fact that user's behavior do not change through time, then the content of past user actions may be used to predict the desired content of their future actions.

As an example, we can suppose that, if a user has positively rated a song that belong to jazz, then the system can learn to recommend other songs from the same music genre.

Information filtering can be viewed as the process of filtering information relevant to an individual's information needs, in order to come to terms with the ever increasing amount of information whereas

Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). Although, IR has significant differences with recommender systems, however, content-based algorithms are driven from IR [12]. Some IR technics such as Boolean retrieval, Rocchio's algorithm or correlation-based schemes, Probabilistic retrieval systems, Natural language query are useful for recommender systems and can be used in content based filtering. Other techniques for content-based recommendation use Pattern Recognition/Machine Learning approaches such as Bayesian classifiers [13], clustering methods, decision trees, and artificial neural networks.

The questions we have to take into consideration while building content-based recommendation system are as follow [7]:

- How do we create similarity between items?
- How do we create and update user profiles continuously?

Content based technics (CBF) are used in some recommender systems applications to provide recommendations to users.

We have:

1. StubleUpon which is a recommender system that assist user in web browsing by providing some web content to its users, they can discover and rate Web pages, pictures and videos that are personalized to their tastes and interests using peer-sourcing and social-networking principles.
2. Last.Fm which is a music website using a music content based recommendation called Audioscrobbled and predictions are provided to users based on previous items that they rated over time.
3. Google News [5] which provide news recommandations by using both methods content based filtering and collaborative filtering.

This technique doesn't take into consideration the user's neighborhood preferences. Hence, it doesn't require a large user group's preference for items for better recommendation accuracy. It only considers the user's past preferences and the properties/features of the items.

So, compared to collaborative filtering, there are some advantages and drawbacks of content-based filtering that we could note [[14],[15]].

Advantages

1. User independence : Uses the item's content to predict the user's interest, no needs other users' rating to find the similarity between the users and give the suggestions
2. Accurate method : recommendation quality improve as the review/items content data cumulates
3. Ability to recommend users with unique taste
4. Ability to recommend new and unpopular items
5. Transparency : content-based method can tell you they recommend you the items based on what features, in order word it can provide you explanation for recommendations

Disadvantages

1. Limited content analysis: if the content does not contain enough information to discriminate the items precisely, the recommendation will not be precisely at the end
2. Impossible to predict the totally distincts types of items the particular user has never expressed interest in
3. Overspecialization : items that are highly correlated with user profile or interest are only recommended
4. Unable to use quality judgements from other users
5. Sparsity of data

2.2.3 Collaborative methods.

Another very popular personalized recommendation approach is collaborative filtering and in the next few lignes we will try to give an overview on collaborative filtering approach, what is it particularity?, how its works, how it could be implemented. At the end of this paragraph like every methods, we will give advantages and disadvantages.

Instead of comparing user interest with items, in collaborative based filtering the comparison is made between user and user to find the similarity as it's shown in Fig [2.4].

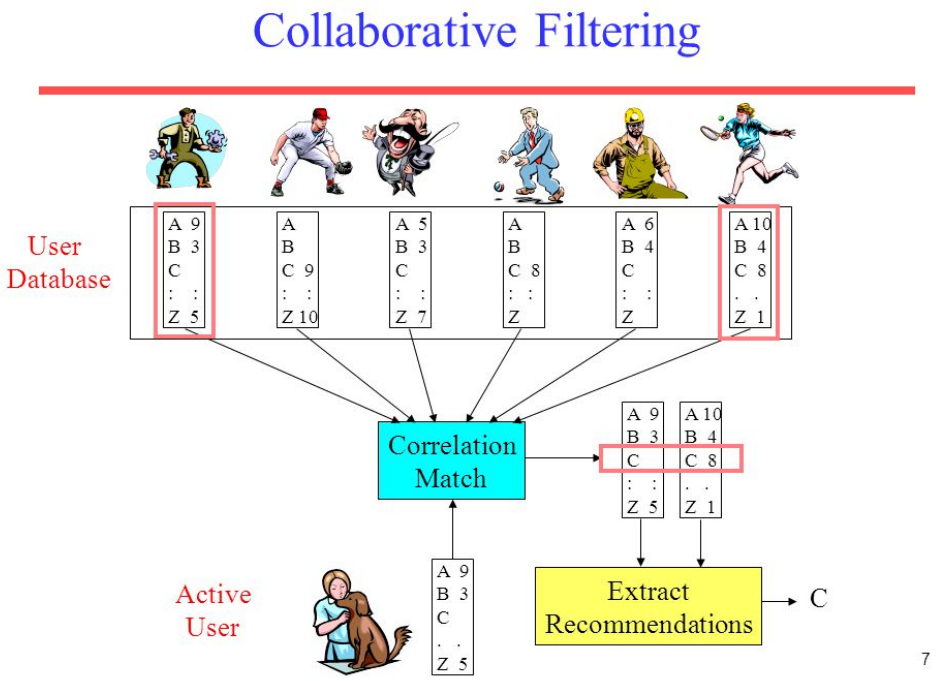

Figure 2.4: Collaborative based filtering
(Source: http://slideplayer.com/slide/5692490/)

Content based filtering works in a different way and Fig [2.5] try to highlight the difference between these two approaches.

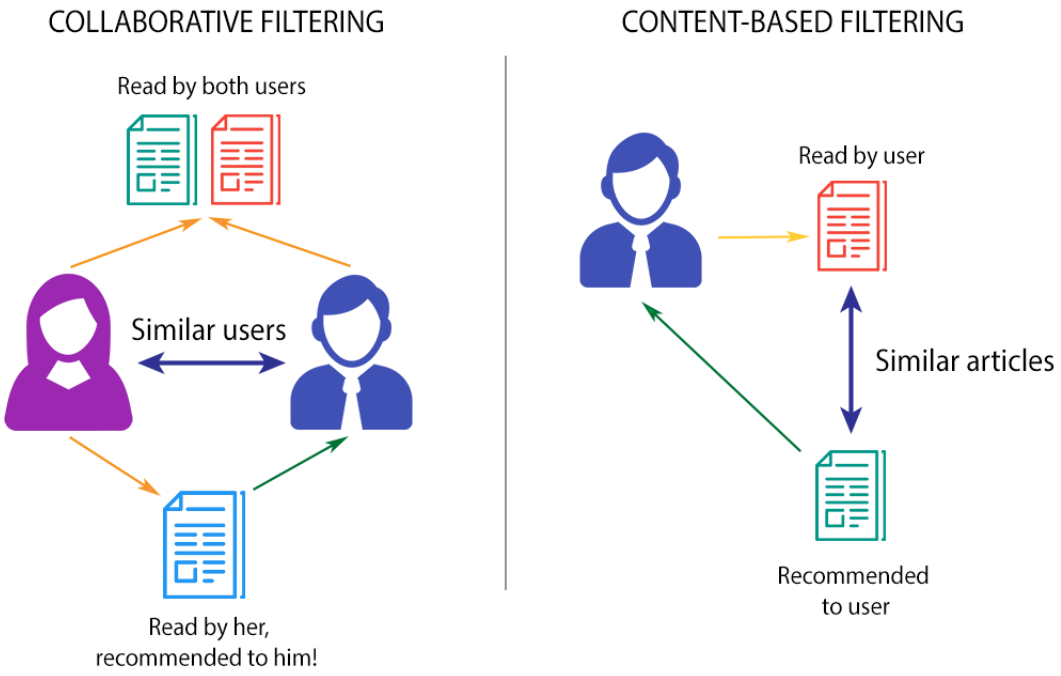

Figure 2.5: Content based filtering vs Collaborative filtering
(Source: https://www.themarketingtechnologist.co/building-a-recommendation-engine-for-geek-setting-up-the-prerequisites-13/)

These ideas about collaborative filtering are based on the fact that similar users prefer similar items or a user expresses similar preferences for similar items. If two users give the same rate n on an items or have similar behavior then they will rate others item in the same ways. Collaborative methods could be implemented with two approaches memory based or model based .

Memory based approach is based on the user database which is kept in memory and each recommendation is perfomed on the whole database. In memory based collaborative filtering many metrics such as pearson correlation metric or cosine similarity can be used to compute the similarity between users or items. The pearson correlation similarity metric to estimate the proximity among users performs better than the cosine similarity [16]. Memory-based CF methods can be further divided into two groups, namely user-based and item-based methods.

User-based methods look for users similar (also called "neighbors") whereas item-based methods look for similar items for an active user.

This approach have a good accuracy when the set of user preferences it is large but it is time consuming when the database become huge. Among the first papers proposed with this approach, [17] stated that Goldberg on 1992 developed certain type of memory-based CF system which is one of the earliest recommender systems, a manual CF mail system called Tapestry.

[18] present a CF recommendation engine for finding news articles called GroupLens.

[19] present a personalized music RS called RINGO, similarities between the tastes of different users are utilized to recommend music items. This approach have two majors parts Item/Object-based filtering and User/Customer-based filtering.

Model based approach does not have limitations of memory based approach because its use the underlying data to learn a probabilistic model such as a cluster model or a Bayesian network model, using statistical and machine learning techniques. Then, they use the model to make predictions. For [2], the clustering model works by clustering similar users in the same class and estimating the probability that a particular user is in a particular class.

While dealing with collaborative filtering recommender systems, we will learn aspects like [7]:

- Calculate the similarity between users
- Calculate the similarity between items
- Deal with new items and new users whose data is not known

So, compared to content-based filtering, there are some advantages and drawbacks of collaborative filtering that we could note[[14],[15]] :

Advantages

1. Works without item attributes
2. Predict items through similar user patterns,even if the particular user has a short review history
3. It does not depend on analysing the content provide by the user
4. Stability : the methods are not significantly affecte by the constant addition of users and items in a large commercial applications and do not require retraining
5. Can deal with multimedia content
6. Can recommend serendipity item
7. Efficiency : the methods require no costly training phases and storing nearest neighbors of a user requires very little memory. Thus it is scalable to millions of users and items.

Disadvantages

1. Cold start problem for the new users
2. Sparsity ratings problem on the same item
3. First rater : cannot recommend an item that has not been previously rated
   (a) New items
   (b) Esoteric items
4. Popularity Bias : cannot recommend items to someone with unique taste.
   - Tends to recommend popular items
5. Recommendations are difficult for users with distinct tastes; these users are called **black sheep** or **gray sheep**

Apart from these four approaches, we also have hybrid approach wherein we can combine the above mentioned algorithms to improve the performance of recommender systems.

2.2.4 Hybrid Methods.

A hybrid recommender system is one that combines multiple techniques together to achieve some synergy between them. For [20] in hybrid approach, two or more filtering methods are combined to gain better performance over CBF and CF approaches when they are applied separately. The three recommenders systems mentioned above (non personalized system, content based methods, collaborative methods) exploit different sources of inputs , and in different cases they may work well. For instance content based methods rely on the content description and the target user's own ratings whereas collaborative filtering systems rely on user ratings and community rating. It is to note that these systems use differents types of input, and have different strengths and weaknesses. Some recommender systems, such as content based methods, are more effective in cold-start setting where a significant amount of data is not available. Other recommender systems, such as collaborative methods are more effective when a lot of data is available.

Examples

1. By combining content-based methods, where the model fails because of the limited content, short review history of the item, with collaborative filtering systems, where similar user patterns is available, new items can be recommended more accurately and efficiently.

2. By combining collaborative filtering methods, where the model fails because of the cold start problem when new items don't have ratings, with content-based systems, where feature information about the items is available, new items can be recommended more accurately and efficiently.

In some cases where data are available, we have the possibility of using differents types of recommendation systems for the same task, in such case many opportunities exist for hybridization where we combined two or more types of systems to achieve better performance by dropping out the drawbacks of each technique separately. Burke in [21] has categorized hybridization methods into seven different types including: (1) mixed approach, (2) weighted, (3) feature combination, (4) cascade, (5) feature augmentation, (6) meta-level hybridization approach, (7) switching.

The Description of these methods can be see in Table 2.1

Hybridization method	Description
Mixed	Recommendations from several different recommenders are presented at the same time
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-level	The model learned by one recommender is used as input to another.
Switching	The system switches between recommendation techniques depending on the current situation.

Table 2.1: Methods of Hybridization

2.3 Collecting Knowledge About User Preferences

In order to make a personalized recommendations corresponding to the user preferences, recommender systems collect ratings of items by user and build user-profiles depending on how the data was collected. Its can be done through three approaches:

- Implicit approach, which is based on recording user behavior,
- Explicit approach, which is based on user interrogation,
- Mixing approach, which is a combination of the previous two.

2.3.1 Implicit approach.

This method does not require active user involvement in the knowledge acquisition task but instead the user behavior is recorded and we look at how user react to each incoming items. The aim of this approach is to learn the relevance of an items according to the user reaction. It also uses tracking user action by observing the links clicked on, the time spent on a webpage, the scrolling activity in addition to study user browsing activity.

2.3.2 Explicit approach.

In this approach it is required to the user to indicate his preferences with respect to each item, by indicating with a brief summary and a ratings often from 1 to 5 or then to give ratings in terms of number of stars to which are associated a weight or a numerical value for instance $R_{i,j} \in [1, 2, 3, 4, 5]$. Generally a low star rating given to an item corresponds to a low estimate of the item while a high star rating corresponds to a high interest of the item. This method requires additional efforts to the user that is why users generally avoid giving notes to items. The ratings performed on these scales will allow statistical studies to be done.

2.3.3 Mixing approach.

In this method the two previous approaches are combined. NewsWeeder [22], is a netnews-filtering system that addresses the problem of the maintenance of the user profile, which describes the user's interests. It is an example of mixing systems that uses both approaches, explicit and implicit to collecting knowledge about user preference which operate by letting the user rate his or her interest level for each article being read (1-5), and then learning a user profile based on these ratings.

2.4 Fundamental Problems of Recommender Systems

In this section, we present the issues and challenges that are common in recommender systems. These include: : (1) Cold Start Problem, (2) Synonymy, (3) Polysemy,(4) Shilling Attacks, (5) Privacy, (6) Limited Content Analysis, (7) Overspecialization, (8) Grey Sheep, (9) Sparsity, (10) Scalability, (11) Latency Problem, (12)Evaluation and the Availability of Online Datasets, (12) Context-Awareness.

1. Cold start Problem: It arise when a new entity enters the system for the first time and in this situation the recommender system can not provide recommendation because of the lack of information about the current entity which are : new items and new users.
2. Synonymy : There are many ways to refer the same object.
3. Polysemy : Most words have more than one distinct meaning
4. Shilling Attacks : people may give positive ratings for their own items and negative ratings for their competitors in a manner advantageous to them.
5. Privacy : employing user's private information to recommend to others.
6. Limited Content Analysis : this method is based on text, but not all content is well represented by keyword (picture, videos, music, blog, taste, etc.). It is a complicated task to generate the attributes for items.
7. Overspecialization : recommends only items that match user's content profile. People might have multiple interests.
8. Gray Sheep User Problem : most of users falls into the class of so called white sheep which are those who have high correlations with others users, we have another class called black sheep which correspond to the groups of users that have very few or no correlating users and Gray sheep which correspond to the users that have their own unusual tastes and low correlations with others.
9. Sparsity : limited amount of historical information for each user and for each item.

10. Scalability : some algorithms require computations that grows with both the number of users and the number of items.

3. Jobs Recommendation Systems

The fast growth of the Internet caused a matching growth of the amount of available online information that increased the need to expand the ability of users to manage all this information. This encourages a substantial interest in specific research fields and technologies that could benefit the managing of this information overload [23]. The most important fields are information retrieval and information filtering. Information retrieval deals with automatically matching user's information and information filtering aims to assist users eliminating unwanted information [24]. In order to solve this information overload, scientist designed what we call recommender systems that originated from information retrieval. The interest in this area still remains high because it is composed of a rich- problem research area and has a wealth of practical applications. Recommender systems are being widely accepted in many areas such as e-commerce, e-recrutement, social networks and so one in order to expand customer services, increase selling rates, decrease customers search time and to improved the user experience. A wide range of compagnies such as the online book retailer amazon.com, the movies and series provider Netflix, the professional networking LinkedIn or the social networking Facebook have successfully set up commercial recommender systems and have increased web sales and improved customer fidelity.

3.1 Research Motivation and Problem Description

Recently the increase of digital data and the emergence of e-business has led to a reform in the way companies operate their business in different aspects. In the field of recruitment, job offers were posted in the career session of the website for most companies. Due to the feedback received and the experience gained, they developed platforms specifically designed for recruitment. These platforms are used by job seekers to create profiles and to be able to apply whenever a new job posting is published. As a result, for a job, thousands of applications are received by the company, resulting in a huge and significant supply of jobs and CVs available online. This has caused a huge need for systems of recommendations.

In the other direction, looking for a new job is a difficult and time consuming process. The most common approach for a user is to search for job offers by keywords on a job posting site and then is returned a list of job postings containing the keywords whose he will evaluate according to his profile and his preferences [25]. Although having obtained the results the candidate is unable to know if he is fit or not for the proposed job offer.

3.2 Related Work

A job recommender (JRS) is designed to match jobs to users, removing the need for manual search. The recommender should evaluate a user's suitability for jobs and recommend those that advance a user's career [26]. This section presents different research areas that are related to our proposed algorithm. We start by presenting a brief review of job recommendation approaches, followed by a comparative study for differents JRSs.

3.2.1 Review of Job Recommendation Approaches.

The JRS has been studied from many aspects both from academia purposes or business purposes, it has attracted a lot of research attention and has played an important role on the online recruiting website.

Traditional recommendation systems recommend items to users whereas job recommender systems recommend one type of users (e.g. job applicants) to another type of users (e.g. recruiters). A comparative study of four online jobs recommenders systems is made in [27] by doing a comparison and analysis at the product level. It also highlight the differences between a JRS and a generic RS and develop a novel JRS by clustering the users and finding out the appropriate recommendation approach for each user group. Different recommendation strategies are used by online recruiting systems.

Theses strategies include non personalized systems, content-based filtering (CBF), collaborative filtering (CF), knowledge-based recommendation (KBR) etc. Among theses techniques, CBF is frequently the most used by online recruiting systems, due to the convenience of collecting users' demographic information. However, we often use hybrid methods because it is difficult to achieve a good accuracy only with one strategy. A comprehensive investigation is made in [27] on four online job recommender systems from four different aspects: user profiling, recommendation strategies, recommendation output, and user feedback.

The four well known online JRSs are Casper, Proactive, Prospect and eRecruiter, coming from Germany, France and Hong Kong, we will make an investigation for a comparative studies. Casper is a classical job application system used for enhancing the performance of JobFinder[1], a website which was developed to help job seekers easily browse the best jobs, learn about employers, and get advice on applying for those jobs. The Proactive has different recommendation modules applied to its own website[2], they believe that having the right staff is one of the most important aspects of running a successful business and understand the necessity of finding the right applicant within a tight time frame. The Prospect[3] is developed by analyzing and mining the resume. eRecruiter is designed for expanding the functionality and improving the accuracy of the Absolventen.at[4].

3.2.2 Comparative Study on JRSs.

In this section, we make a comprehensive comparison of theses four above online JRS shown in Table[3.1, 3.2] and based on their related literatures [[28], [29], [30], [31]] and websites.

Jobs recommendation systems				
	User Profiling	Approach	Layout	User Feedback
Casper	Individual information and behavior	CFR, CBR	Comprehensive list	Apply Collect
Proactive	Individual information	CBR, KBR	Modular list	Apply
Prospect	Individual information	CBR	Comprehensive list	Lack of website
eRecruiter	Individual information and behavior	CBR, KBR	Comprehensive list	Email

Table 3.1: A comparative study on JRSs

[1] http://www.jobfinder.com
[2] http://www.proactiverecruitment.co.uk
[3] http://www.prospects.ac.uk
[4] http://www.absolventen.at

JRSs	Advantages	Disadvantages
CASPER	Hybrid profile and approach. User can set the feature importance. Update profile based on user feedback.	Content of profile is simple. Use one way recommendation.
Proactive	Hybrid approach.Provide four recommendation modules. Use ontology to classify jobs.	Single profile. Knowledge engineering problem. Only email about user feedback.
PROSPECT	Resume miner. Batch processing	Single profile and approach. Simple resume match. Use one way recommendation.
eRecruiter	Hybrid profile and approach. Use ontology to classify jobs and users.	Single method of calculating similarity. Use one way recommendation

Table 3.2: Advantages and Disadvantages of Online JRSs

3.3 Data Mining Methods for Recommender Systems

Data can be define as a collection of objects and their attributes, where an attribute is defined as a property or characteristic of an object. Real-life data typically needs to be preprocessed (e.g. cleansed, filtered, transformed) in order to be used by the machine learning techniques in the analysis step. In this section, we will present popular data preprocessing techniques, data-mining techniques, and data-evaluation techniques commonly used in recommender systems. The first part of this section deals with how a data analysis problem is tackled, followed by data preprocessing steps such as similarity measures and dimensionality reduction. The next part of the section is about data mining techniques and their evaluation techniques.

Solving a data analysis problem : realizing this operation involves a bunch of steps such as:

- Well posed the problem

  Start with a clearly defined problem, begin with the right questions. Questions should be measurable, clear and concise. Design your questions to either qualify or disqualify potential solutions to your specific problem or opportunity.

- Set Clear Measurement Priorities

  This step can be done by Deciding what to measure, and Deciding how to measure it.

- Data collection

  After the question clearly defined and the measurement priorities set, it's time to collect your data, by Identifying data sources and data variables suitable for the analysis.

- Analyze Data

  After collecting the data to answer your question, it's time for deeper data analysis. Data preprocessing or a cleansing step, such as identifying missing values, quantitative and qualitative

variables and transformations, performing exploratory analysis to understand the data, mostly through visual graphs such as box plots or histograms.

- Interpreting Results

  After analyzing the data and possibly conducting further research, it's finally time to interpret the results. By following these five steps in our data analysis process, we make better decisions for our research study, business or government agency because our choices are backed by data that has been robustly collected and analyzed.

Data preprocessing steps : One of the core step for any data analysis problem is data preprocessing, because by doing that we ensure about the accuracy of the model because it mostly depends of the quality of the data. It involve data consolidation (collect data, select data, integrate data), data cleaning (impute missing data, reduce noise in data, eliminate inconsistencies data), data transformation (normalize data, discretize /aggregate data, construct new attributes), data reduction (reduce number of variables, reduce number of cases, balance skewed data). After the well execution of these steps the preprocessed data which is well formed data can be fed into a machine learning algorithm. In this part, we will focus on data preprocessing techniques including similarity measurements (Pearson coefficient, cosine distance, euclidean distance, Tf-Idf measure), sampling and principal component analysis, singular value decomposition (SVD), Deep Belief Networks, Stacked Autoencoders which are a part of dimensionality reduction techniques.

3.3.1 similarity measures.

Different measures of distance or similarity are convenient for different types of analysis, and they are used in recommender systems to compute the similarity between items and users. In this section we will explore some similarity measures such as : Tf-Idf measure, Pearson coefficient, cosine distance, euclidean distance.

TFIDF weighting

TFIDF [32] is the most common weighting method used to describe documents in the Vector Space Model. This weighting function has been particularly related to two important machine learning methods: KNN and SVM.

Tf-Idf [33] is a measure that is often used when dealing with textual data to information retrieval in particular for text mining and to calculate document similarities or to find relevant documents among a set of documents.

The Tf-idf word can be divided into two words which are: Tf and Idf.

Tf stands for term frequency and Idf stands for inverse document frequency.

Let us define:

- df(t) : number of document containing the term t which is $\sharp\{d_i,\, t \in d_i\}$
- c(t,d) : Number of times term t appears in a document d
- N(t) : total number of terms in the document
- M : total number of document

So we can express $TFIDF_{t,d}$ in terms of tf and idf by :

1. TF: Term Frequency, which measures how frequently a term occurs in a document.

$$TF(t,d) = \frac{c(t,d)}{N(t)} \tag{3.3.1}$$

2. IDF: Inverse Document Frequency, which measures how important a term is.

$$IDF(t) = 1 + log_e\left(\frac{M}{df(t)}\right) \tag{3.3.2}$$

According to [34], the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

In order words in [35], $TFIDF_{t,d}$ given by :

$$TFIDF_{t,d} = TF_{t,d} \times IDF_t \tag{3.3.3}$$

Assigns to a term **t** a weight in document **d** that is :

- Highest when **t** occurs many times within a small number of documents (thus lending high discriminating power to those documents)
- Lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal)
- Lowest when the term occurs in virtually all documents.

Example:

Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3 in Table 3.3

documents terms	Doc1	Doc2	Doc3
table	27	4	24
car	3	33	0
wife	14	10	29
best	14	0	17

Table 3.3: Table of tf values

We will compute the $tfidf$ of the terms t_1 = wife and t_2 = best in the document Doc3, but before we need to have df(t), c(t,d), N(t) and M.

Refering to the table

- $df(t_1) = 3,\ df(t_2) = 2$
- $c(t_1, d_3) = 29,\ c(t_2, d_3) = 17$

- $N(t_1) = 70,\ N(t_2) = 70$
- M = 3

We can easily deduce that

1. $tf - idf_{t_1,d_3} = \frac{29}{70} \times (1 + ln(\frac{3}{3})) = 0.4143$
2. $tf - idf_{t_2,d_3} = \frac{17}{70} \times (1 + ln(\frac{3}{2})) = 0.3413$

This result tell us that the term $t_1 =$ wife is more important in the document Doc3 that the term $t_2 =$ best.

Another similarity measure is **Cosine similarity**. It considering items as document vectors and compute their similarity by taking the cosine of the angle formed by these vectors. It is given by the following formular:

$$Similarity(x, y) = cos(\theta) = \frac{< x, y >}{\|x\|\|y\|} \tag{3.3.4}$$

where $< ., . >$ is the inner product and $\|.\|$ is the norm. Similarity between two items can also be given by the correlation existing between their variables. There are several correlation coefficients that may be applied but the **Pearson correlation** is the most commonly used. It is given by

$$Pearson(x, y) = \frac{cov(x, y)}{\sigma_x \sigma_y} \tag{3.3.5}$$

where $cov(x, y)$ is the covariance between the data points x, y and $\sigma_.$ is the standard deviation.

Empirical studies showed that Cosine similarity performs well in item-based collaborative filtering whereas Pearson correlation performs well in user-based collaborative filtering. In conclusion, we saw a variety of similarities measures for data with diverses attributes but there is a suitable similarity measure when the data has binary attributes.

First we define M_{ij} as the number of attributes where x was **i** and y was **j**, so M_{10} will be the number of attributes where x was 1 and y was 0. A simple matching coefficient between x and y is defined by :

$$SMC(x, y) = \frac{M_{11} + M_{00}}{M_{01} + M_{10} + M_{11} + M_{00}} \tag{3.3.6}$$

The **Jaccard similarity** is give by :

$$JS(x, y) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \tag{3.3.7}$$

In general, the accuracy of a RS does not depend on the choices of the similarity measure, as a matter of fact [36] and in some context, using a random similarity measure sometimes yielded better results than using any of the well-known approaches.

3.3.2 Data Sampling.

For [37], data sampling is considering as a statistical analysis technique used to select, manipulate and analyze a relevant representative subset of data points in order to identify patterns instead of processing the larger data set. It is used in data preprocessing and in data interpretation and it may be used because processing on the entire data set is computationnally too expensive. Sampling allows data scientists,

predictive modelers and other data analysts to work with a small, manageable amount of data in order to build and run analytical models more quickly, while still producing accurate findings. In big data analytics applications where data sets are too large to efficiently analyze in full sample can be useful. An important consideration, though, is the size of the required data sample. In some cases, a very small sample can tell all of the most important information about a data set. In others, using a larger sample can increase the likelihood of accurately representing the data as a whole. As sampling techniques we have : random sampling, stratified sampling.

3.3.3 Dimensionality reduction.

High dimensionality and sparse data while building recommender systems are the faced problems most of the time. In highly dimensional space notion of density and distance between points become meaningful and this is critical for clustering methods and outlier detection, this problem is know as the Curse of dimensionality problem. Dimensionality reduction techniques help overcome this problem by transforming the original high-dimensional space into a lower-dimensionality and it is essential to increase the efficiency of data analysis and handling. In this part we will summarize the two most relevant traditional dimensionality reduction algorithms in the context of RS: Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) [8].

Visualising or interpreting data which have more than three variables is very crutial issues that disturb statistician. Fortunately, quite often the problem can be reduced by replacing a group of variables which measuring the same informations with a single new variable. PCA is one of the methods for dimentional reduction, It has been called one of the most valuable results from applied linear algebra. PCA is used abundantly in all forms of analysis from neuroscience to computer graphics.

Overview of the problem

Suppose we have n individuals, and on each of them we measure the same m variables. We say that we have n samples of m dimensional data. For the j-th individual record the m measurements as a vector $\vec{x_j}$ belonging to $\mathbb{R}^m$. For instance, we might ask 50 people their height(m), weight(kg), grades and their IQ. In this case, n=50 and m=4 and the measurement $\vec{x_1}$ might look like $\vec{x_1} = \begin{bmatrix} 1.5 \\ 65.8 \\ 18 \\ 220 \end{bmatrix}$

We could visualize this data as a plot of 50 points in $\mathbb{R}^4$

Here are some questions we aim to answer by way of this technique:

1. When m is very large, is there a simpler way of visualizing the data?
2. Which variables are correlated? In this example we would probably expect to see some correlation between grades and IQ.
3. Which variables are the most significant in describing the full data set?

If we let A be an $m \times n$ matrix of data and A^T its transpose. Then from proposition 6.2 in appendix, the matrices AA^T and the A^TA share the same nonzero eigenvalues.

This previous result is very powerful in the sense that if m and n are very different in size, let us say $n \ll m$. For instance if A is a 700×2 matrix, then there is a quick way to find the eigenvalues of the 700×700 AA^T by doing the following steps:

- Find the eigenvalues of A^TA which is only a 2×2 matrix
- The others 698 eigenvalues of AA^T are all zeros.

Principal Component Analysis

From the notation in part 6.3 we can store the mean of all the m variables as a single vector in $\mathbb{R}^m$:

$$\mu = \frac{1}{n}(x_1 + x_2 + ... + x_n) \tag{3.3.8}$$

We translate the origin to the center of gravity, this will produces a zero mean data. Let A be the $m \times n$ $(n \leq m)$ matrix whose j-th column is $\vec{x_i} - \vec{\mu}$ such that $A = [A_{ij}] = [B_{ij} - \vec{\mu}]$

We define the covariance matrix S which will be $m \times m$ as :

$$S = \frac{1}{n-1}AA^T \tag{3.3.9}$$

Then for instance if

$$\vec{x_1} = [a_i]_{1 \leq i \leq m}, \vec{x_2} = [b_i]_{1 \leq i \leq m}, \vec{x_3} = [c_i]_{1 \leq i \leq m}, \vec{\mu} = [\mu_i]_{1 \leq i \leq m}$$

so that $\forall i, j \in [1, m]$

$$S_{ij} = \frac{1}{n-1}((a_i - \mu_i)(a_j - \mu_j) + (b_i - \mu_j)(b_j - \mu_j) + (c_i - \mu_i)(c_j - \mu_j)) \tag{3.3.10}$$

From this equation we noticed that:

- We have the variance of the i-th variable at the i-th entry on the diagonal S_{ii} of S.
- The ij-th entry S_{ij} of S correspond to the covariance between the i-th and j-th variables.

According to the definition of S, it is a symmetric matrix, then by theorem 6.2 it can be orthogonally diagonalized and thanks to proposition 6.2, S and $S^T = \frac{1}{n-1}AA^T$ share the same non-zero eigen values which is exactly n eigenvalues because S^T is a $n \times n$ matrix and $n \leq m$.

Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$, the eigen values of S with corresponding orthonormal eigen vectors $\vec{u_1}, \vec{u_2}, ..., \vec{u_n}$.

These eigen vectors are called the principal components of the data set.

On one hand, we can observe that the trace of S is the sum of the diagonal entries of S, which is the sum of the variances of all m variables. On the other hand, the trace of a matrix is equal to the sum of the eigen values so $\sum_{i=1}^{n} S_{ii} = \sum_{i=1}^{n} \lambda_i$

Interpretation

- The direction in $\mathbb{R}^n$ given by $\vec{u_1}$ which is the principal component account for an amount λ_1 of the total variance and it is $\frac{\lambda_1}{\sum_{i=1}^{n} S_{ii}}$ fraction of the total variance.
- The vector $\vec{u_1} \in \mathbb{R}^n$ points in the most significant direction of the data set.

Perform PCA on Data

1. After ordering the eigenvalues from highest to lowest to get the components in order of significance, the eigenvector with the highest eigenvalue is the principle component of the data. We want to reduce the dimension of the data (to do compression for example).
2. Our aims is to project the n dimensional data on a p dimensional subspace ($p \leq n$), minimizing the error of the projections (sum of squared difference) meaning to project on the p eigenvectors that corresponds to the highest p eigenvalues.
3. How to get the new data set ?

   Let us called U the $n \times p$ matrix where the columns are the eigenvectors, the new data F is given by :
   $$F(m,p) = A(m,n) \times U(n,p) \tag{3.3.11}$$
   We can see that the new data is a p-dimensional feature space
4. How to get old data back ?

   we will get exactly the same data, if we keep all eigenvectors.

   We know that
   $$F(m,p) = A(m,n) \times U(n,p) \tag{3.3.12}$$
   So using Proposition 6.3
   $$\begin{aligned} A(m,n) &= F(m,p) \times U(n,p)^{-1} \\ &= F(m,p) \times U(n,p)^{T} \end{aligned}$$
   we add the mean to vector to A, to get B, and we can noticed that the variance along the Other Component has gone (a lossy compression)

Singular Value Decomposition is a powerful technique for dimensionality reduction. It is a particular realization of the Matrix Factorization approach. For rectangular matrices, a closely related concept is Singular Value Decomposition (SVD) [Part 6.3].

Summary on PCA and SVD

The goal here is to find the principal components P of a data matrix A(m,n).

1. Zero mean the column of A
2. Apply PCA or SVD to find the principal components P of A.

   PCA:

   - Compute the covariance matrice $C = \frac{1}{n-1} AA^T$
   - P = the eigen vectors of C
   - The variance in each new dimension is given by the eigenvalues

   SVD:

   - Compute the SVD of A

- P = the singular vectors
- The variances are given by the squaring of singular values

3. Get the new data set $F = A \times P$

In traditional PCA and SVD approaches, it's just the matrix factorization involved which is linear. But in other non-linear techniques and deep learning techniques such as Deep Belief Networks(DBN) and Stacked Autoencoders complex functions are used to deal with non-linear data and of course linear data too.

In [38], the advantages of deep learning techniques over traditional techniques are:

1. Uses unsupervised training which eliminates the need of labels for training.
2. Data can be separated more easily.
3. Local optima can be prevented.
4. Meaningful representations can be made

Data scientists use many different kinds of machine learning algorithms to discover patterns in big data that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they learn about data to make predictions: supervised and unsupervised learning. Supervised machine learning is the more commonly used between the two. It includes such algorithms as linear and logistic regression, multi-class classification, and support vector machines. Supervised learning is so named because the data scientist acts as a guide to teach the algorithm what conclusions it should come up with.

1. **Regression** : we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function
2. **Classification problem** : we are instead trying to predict results in a discrete output. In other words we are trying to map input variables into discrete categories.

Some popular examples of supervised machine learning algorithms are:

1. Nearest Neighbors
2. Decision Trees for regression problems
3. Ruled-Based Classifiers
4. Bayesian Classifiers
5. Logistic Regression for regression problems
6. Support Vector Machines for classification problems
7. Artificial Neural Networks

Most of the time we approach the data without any information about the data property. Unsupervised learning allows us to approach problems with or no idea what our results should look like. We can derive structure from data where we do not necessarily know the effect of the variable. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. Scaling a classifier for finding the best k-nearest neighbors involve computing distances with is done by a amount of operations. Dimensionality reduction is a possible solution to it, but with that we still have to compute some distances, this is where clustering algorithms can come into play. Unsupervised learning problems can be further grouped into clustering and association problems:

1. Clustering [8] consists of assigning items to groups so that the items in the same groups are more similar than items in different groups: the goal is to discover natural (or meaningful) groups that exist in the data. The goal of a clustering algorithm is to minimize intra-cluster distances while maximizing inter- cluster distances. Amount clustering algorithms categories we have hierarchical and partitional. We can cite k-means, density-based clustering (DBSCAN), Message-passing clustering, Hierarchical Clustering, Locality-sensitive hashing (LSH), Latent Dirichlet Allocation (LDA) etc...
2. Association Rule Mining

   An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

3.3.4 Evaluating data-mining algorithms.

Previously, we have seen various data mining techniques used in recommender systems. In this part, we will see how to evaluate those models built using data mining techniques which is very important for any data analytics models to perform well on future data. This could be achieved only if we build a efficient and robust model.

While evaluating any model, the most important things we need to consider are as follows:

- Whether the model is over fitting or under fitting
- How well the model fits the future data or test data

Under fitting, also known as bias, is a scenario when the model doesn't even perform well on training data whereas over fitting is a scenario when the model performs well on training data, but does really bad on test data.

Any fitted model is evaluated to avoid previously mentioned scenarios using cross validation, regularization, pruning, model comparisons, ROC curves, confusion matrices, and so on.

- **Cross validation**

  A very popular technique for model evaluation for almost all models. In this technique, we divide the data into two datasets: a training dataset and a test dataset. The model is built using the training dataset and evaluated using the test dataset.

- **Regularization**

  In this technique, the data variables are penalized to reduce the complexity of the model with the objective to minimize the cost function.

- **Confusion matrix**

  This technique is popularly used in evaluating a classification model. We calculate precision and recall/sensitivity/specificity to evaluate the model.

  Precision: This is the probability whether the truly classified records are relevant.

  Recall/Sensitivity: This is the probability whether the relevant records are truly classified.

  Specificity: Also known as true negative rate, this is the proportion of truly classified wrong records.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
Predicted	Negative	False Negative (FN)	True Negative (TN)

Table 3.4: Confusion matrix

Let's understand the confusion matrix write in Table 3.4 :

1. TRUE Positives (TP): number of instances predicted as belonging to class A that really belong to class A
2. True Negatives (TN): number of instances predicted as not belonging to class A and that in fact do not belong to class A
3. False Positives (FP): number of instances predicted as class A but that do not belong to class A
4. False Negatives (FN): instances not predicted as belonging to class A but that in fact do belong to class A

The most commonly used measure for model performance is its Accuracy defined as the ratio between the instances that have been correctly classified and the total number of instances:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3.13}$$

Other common measures of model performance is Precision is a measure of how many errors we make in classifying samples as being of class A. It is defined as:

$$P = \frac{TP}{TP + FP} \tag{3.3.14}$$

we also have what we call Recall which measures how good we are in not leaving out samples that should have been classified as belonging to the class, it is defined as:

$$R = \frac{TP}{TP + FN} \tag{3.3.15}$$

4. Experiments

In this chapter, we describe the matching approaches that we have implemented. We developed a local JRS on the platform of the skillake website. The RS can extract the user profile automatically and provide the function of searching based on the database. Besides, the RS provides different lists of recommended jobs for different job applicants.

4.1 Plateform insight

Skillake is a new vision of Jobboard also called job portal that deals specifically with employment or careers. It is designed to allow employers to publish job requirements for a position, also offering career, job search tips by describing different jobs or employers. For their research, job seekers can through the platform locate and fill out a job application or submit CVs to find the job of their dream.

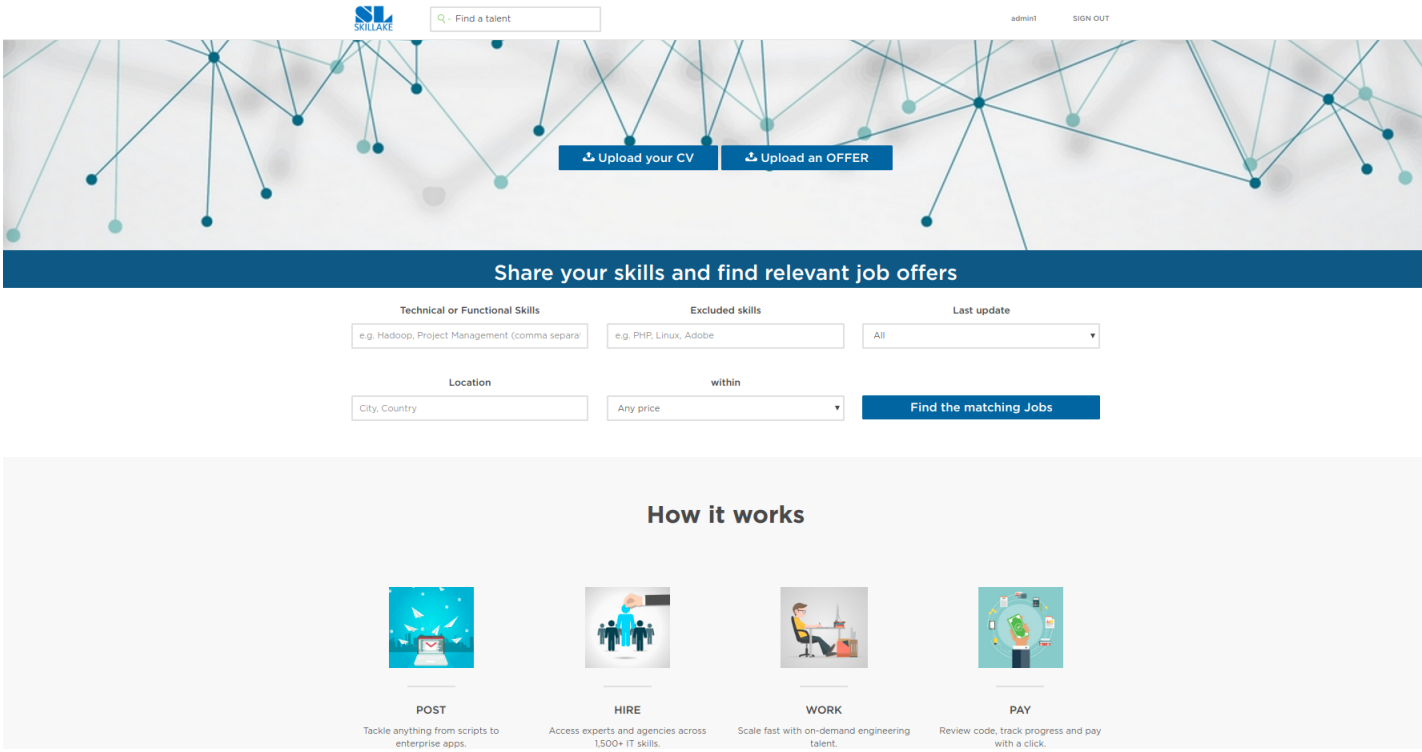

Figure 4.1: A Screenshot of Skillake view

Fig 4.1 give us an overview of the website.

Figure 4.1 shows the homepage of the job posting platform called skillake. This interface can be divided vertically into three large parts.

The first part corresponds to the header of the home page where the connectivity status is displayed where the user can see if he is actually connected or not and see his profile and user name, at the extreme right of the header, the user can disconnect and at the extreme left he can do research profiles to which he is interested.

The second part of this interface corresponds to the two buttons where the user can either upload his CV if he is looking for a job offer or he can upload a job offer to find potential candidates who correspond the requirements of the job offer.

The third part, allows the user to search for a job offer according to their requirements.

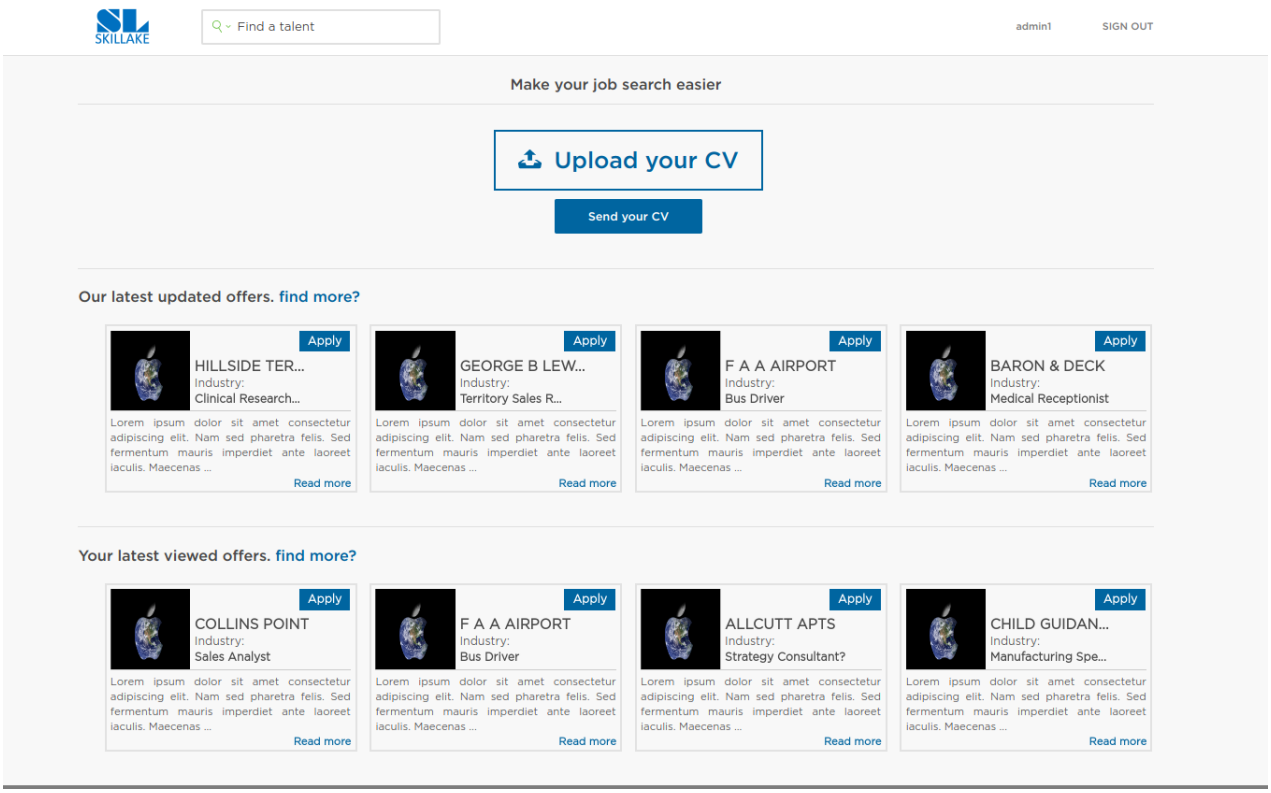

Figure 4.2: A Screenshot of Skillake overview

The figure 4.9 corresponds to the interface where the connected user or not can upload his CV and be able to find the job offer that corresponds to the skills described in his CV. On this page the user can see the latest job offers published on the website and if he is logged in he can see the offers he consulted during his last visit on the skillake platform.

After this step follows the document processing step and the recommendation step. The uploaded document will be parsed and vectorized to be passed in the model where it will make a prediction of the cluster to which the uploaded document belongs. The existing model in the database is a model created in advance by data that are present on the website built by clustering methods.

Then the model will find the best cluster corresponding to the documents similar to the one uploaded.

Then the elements of this cluster will be sorted in order of similarities, and then is send back to the user one of the 4 best job offers in the case where the user is a job seeker or then return the 4 best profiles in the case where it is a recruiter. Any time the user can modify information that is presented to him to improve the recommendation according to his tastes and his preferences.

4.2 DataSet

To the best of our knowledge, there are no job offers in the literature which can be freely exploited. Furthermore, as this work is conducted in order to meet the needs in the field of human resources, we will conduct our experiments on an extraction of job offers from a public limited company with a base of a few thousand people CVs of about eighty domains.

In order to test our algorithms, we need to use data sets that are realistic representations of the daily operation of job offers.

As far as we know, [39] are explicitly discussed the challenges of creating and using data sets specifically for testing recommender algorithms. They stress that it is "very important that the tasks your algorithm is designed to support are similar to the tasks supported by the system from which the data was collected" website.

The data set we used is collected from a job hunting website including 8868 jobs and 1000 jobs seekers resume from 100 differents field of specialization. Its available on the popular website Kaggle[1].

Kaggle is the world's largest community of data scientists. They compete with each other to solve complex data science problems, using the latest and varied applications of machine learning. All user ids in the dataset are anonymized to preserve the privacy of the users. The fields in the dataset are comma separated values (CSV). The structure of our data base is shown as fallow:

1. Jobs Offers Data

   The data is divided into three main categories which are: job offers, company data, user profiles. these data contain 8868 job offers coming mostly from France. The jobs offers data contains different fields present in the table below.

Field	Description
1	OrganizationID
2	Posted by
3	Candidates
4	Job title
5	Country
6	City
7	Job Description
8	Employee role
9	Skills
10	Contract type
11	Deadline
12	Status
13	userID

2. Organizations

   These data contain information about companies with the different fields present in the table below.

[1] https://www.kaggle.com

Field	Description
1	Name
2	Sector
3	Logo
4	Size
5	Email
6	Address
7	Description
8	Country
9	Employees
10	OrganizationID

3. Users Profile

   These data contains 1000 resume containing more than 90 trades.

Field	Description
1	UserID
2	Picture user
3	Gender
4	Birthday
5	Phone
6	Country
7	Address
8	Hourly rate
9	Desired contract
10	Availability
11	Bio
12	Language
13	Skills
14	last viewed offers
15	last viewed profiles

4.3 Recommendation process on skillake

The skillake platform has a recommendation system that makes it easier for users of the platform to save time by offering them topics that may be of interest to them. This system presents job offers to job seekers according to the profile and preferences of each user and presents profiles that most closely match the job offers posted by employers looking for qualified staff.

In our system, there are four steps:

- Data Preprocessing : This is the step where we clean the raw data to filter useless data
- Data Understanding : The main goal here is to gain general insights about the data that will potentially be helpful for the further steps in the data analysis process.

- Data Analysis : is a method in which data is collected and organized so that one can derive helpful information from it. In other words, the main purpose of data analysis is to look at what the data is trying to tell us. For example, what does the data show or do? What does the data not show or do?
- Ranking : Sort the Top N
- Recommendation engine : Recommend the top sorted N items to the user

The recommendation system works in a pipeline of three processes :

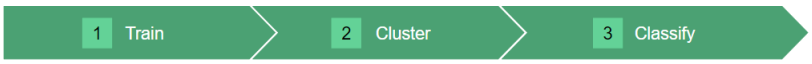

Figure 4.3: Recommendation process on skillake
(Source:
https://nycdatascience.com/blog/student-works/restaurant-recommendations-groups-people/)

In the training part, the preprocessing of the data is done to make it usable by the machine learning algorithms that were designed then the data analysis is done. At the end of this process a clustering is performed on data for groups of information having the same characteristics and this clustering will be registered as a model and it will allow us to make a classification of future data which plays a very important role in the recommendations made.

At this step, a document may be represent as a vector with one component corresponding to each term in the dictionary, together with a weight for each component that is given by Equation 3.3.3. For dictionary terms that do not occur in a document, this weight is zero. This vector form will prove to be crucial to scoring and ranking.

We have used the vector space model which consist to represent a set of documents as vectors in a common vector space, it captures the relative importance of the terms in a document and it is fundamental in information retrieval, text processing operations as well as on document clustering.

In the data preprocessing part an extraction sequence is made on the file to collect some information that will be needed to make the recommendation. As a result, the different extraction algorithms were built, first the extraction of the content is done, then this content is divided into lexical units, then from this result is extracting important information like the name, first name, email address, phone number, country, skills. With the obtained set of lexical units, the suppression of the stop words is done then comes the vectorization step where each document is represented in a vector space by using the vector space model where vector base is the set of lexical units. These transformations are made on these documents to be able to transform them into vector and use a combination between the Tf-IDf and cosines similarity to calculate the degree of similarity between the two documents.

We can not compute the similarity between all the elements available in the database because it will take so much time and the user will have to wait for a long time to get the results of his recommendation, instead we have adapted rather the Kmeans algorithm to group all the job offers and profiles present in the database in different clusters. By doing so, the documents belonging to the same cluster are very close to each other and this result is most closely to the output of a collaborative filtering results. This method allows us to save time and make recommendations very quickly in a matter of seconds despite the size of the data available in the database

4.4 Implementation

As we have seen in Chapter 2 , setting up a job recommender sytems faced us some challenges. The materials, packages used and app structure have been choose, made and justified according to these challenges.

In this part, in a first hand we will outline the materials and packages used to build our predictive algorithm.

In a second hand, we will present the app structure composed of front-end and back-end and others framework that was used to implement our algorithm.

4.4.1 Materials and packages used.

Throughout this work we have for most of the time working mostly with .txt, .pdf, .docx files. We used **Pypdf** which is a pure-python PDF toolkit. It can extract data from pdf files, or manipulate existing pdfs to produce a new file.

We also use **mimetype**, the mimetypes module converts between a filename or URL and the MIME type associated with the filename extension.

We used The Python **re** module provides regular expression support which are a powerful language for matching text patterns.

Another useful packages that we used is **python-docx** which is a Python library for creating and updating Microsoft Word (.docx) files. These librairies was used to perfomed tasks like keys words extraction or features extraction on data which are the part of data preprocessing, data understanding and data analysis.

Others librairies are used more for data analysis like **numpy** and **scipy** which are fundamental package scientific libraries for Python. We also used the **pickle module**, it allows to save in a file, in binary format, any Python object.

We also used **scikit-learn** which is a simple and efficient tools for data mining and data analysis, accessible to everybody, and reusable in various contexts.

The skillake platform was designed mainly with the **Django framework** and the **python programming language**, the database to be integrated into the site is **postgreSql** which is the most advanced open source database in the world.

4.4.2 App Structure.

Our App as shown in Figure 4.4 marries a Django front end with a Python back end to provide recommendations. Our product, Skillake, is a multiuser jobs and profiles recommendation engine. The front end is built on Django, a Python web application microframework, in combination with templating languages including Bootstrap, Jquery, CSS, and HTML.

Bootstrap which is an open source toolkit for developing with HTML, CSS, and JS was used to build responsive, mobile-first projects on the web with the world's most popular front-end component library. Javascript was also used to employ Facebook, Google+ and LinkedIn API into our application. On the back end, we used both advanced machine learning framework, we also created some API to make the recommendation work easiest and quickly regardless of the size of the data present in the database.

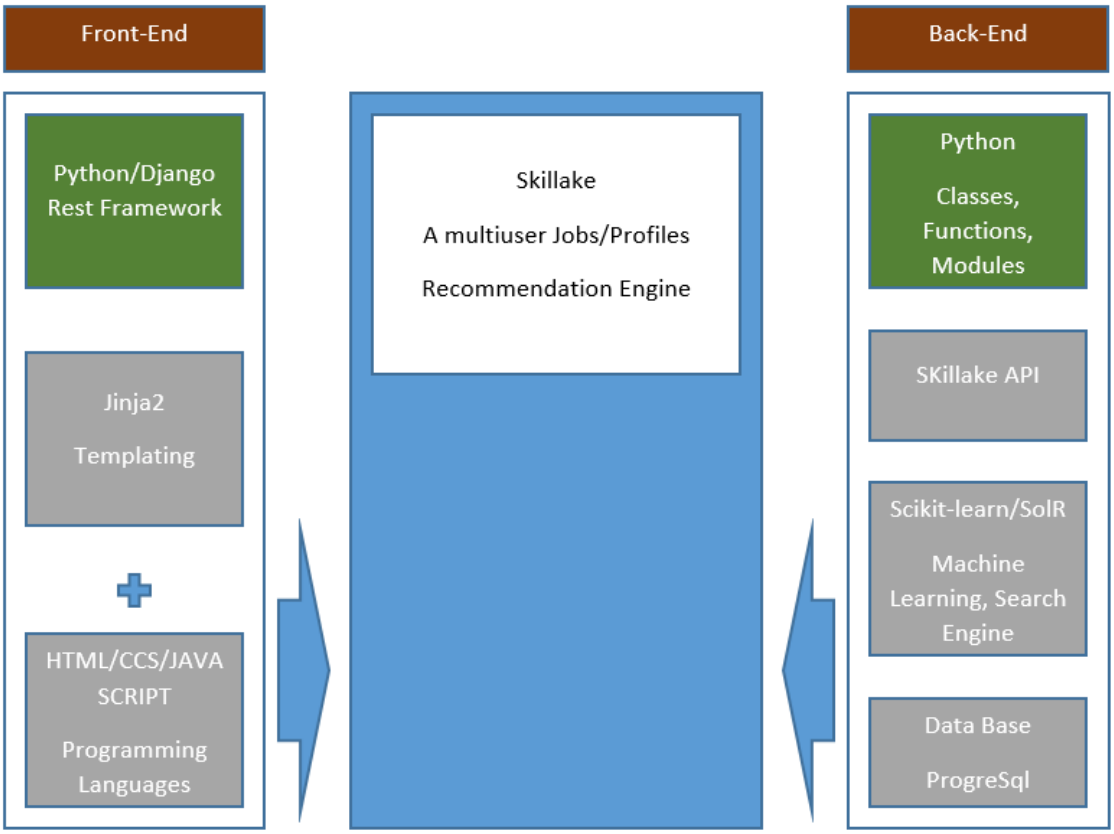

Figure 4.4: Skillake App Structure

4.5 Experiments

The recommendation system implemented on skillake is an hybrid system. This choice has been made and justified to fill the different weaknesses of each method seen in chapter 2.

The system in place is a hybridization of content based filtering and collaborative filtering, more precisely the model based filtering,the memory based filtering which are both sub techniques of collaborative filtering and user-based, item-based which are sub methods of content based filtering.

The memory based filtering still calls neighborhood based can be broken down into two parts: the user-based and the item-based whereas the model based can be broken down into parts such as: bayesian model, clustering, neural network etc. [Cf. Fig 2.2.]

We have developed a model using data mining, machine learning algorithms to find patterns based on training data. This model has the advantage of better managing the sparsite than the algorithms based on the memory and it improves the evolutivity and the prediction performances but is expensive during the construction of the models.

The second approach used is the memory based collaborative filtering which uses the entire data of users and jobs to make different neighborhoods.

These different neighborhoods of users are users with a similar profile and similar skills, the neighborhood of jobs is similar job offers.

Next, user based collaborative filtering and item based collaborative filtering algorithms are used to recommend users, so the profile and skills match best with the job offer proposition.

4.6 Computational Results

This part presents a simulation of a user's interaction on the platform, which comes on the platform to search for a job offer. At first the process of extraction is presented which is done in the back-end then the recommendation process up to the display in the front-end.

The figure 4.5 shows an overview of the user's CV header used to search for a job offer. this resume contains almost all of the user's information regarding his working, research experience and skills.

After the introduction of the file into the platform, the content is extracted automatically see figure 4.6,

then preprocessing is performed on this content see figure 4.7

KEY COMPETENCIES AND STRENGTHS

- Over 7 years research and working experience in Machine Learning and Data Mining field.
- Strong data analytical and programming skills especially with Matlab and Python.
- Excellent English writing and oral presentation skills.
- Strong team-work spirit with experience of working in highly international environments for years.
- Native Mandarin speaker with Permanent Finnish Resident and Working Permit.

WORKING & RESEARCH EXPERIENCE

2014 - Now **Data Analyst at Verto Analytics Inc. (Area: data analytics and image recognition)**
I am working on versatile projects at Verto Analytics Inc. - a Finnish Pioneer in Digital Media Research and Measurement Industry.
My responsibilities include: **1)** developing and implementing machine learning algorithms for mobile-end App image recognition; **2)** collaborating with marketing professionals for writing market insights reports; **3)** data quality assurance, data cleaning and curation, data visualisation, and data production.

2011 - 2014 **Nonnegative Learning for Data Clustering (Area: algorithms and optimisations)**
I designed several Machine learning algorithms using matrix factorisation models to better detect groups or clusters in various data sets. The algorithms can be directly applied for, e.g., Recommendation Systems and Market Segmentation. I published the results in 6 scientific journals and papers.

2011 - 2014 **Understanding the Emotional Impacts of Images (Area: image processing)**
I developed several image processing methods to predict emotional impacts of artistic images. The methods can improve the performance of Affective Image Classification and Retrieval systems. I published the results in 4 scientific journals and papers.

2008 - 2010 **PinView - A Proactive Personal Information Navigator (Area: multimedia retrieval)**
I developed a Gaze-and-Speech-enhanced Content-Based Image Retrieval system that can infer the user's search interests based on his or her feedbacks such as eye tracking

Figure 4.5: A Screenshot of an overview of the user's CV

```
In [16]: get_pdf_content(cv_file)
```

```
Out[16]: "CV for Data Scientist\nHE ZHANG \n Data Scientist, PhD in Machine Learning\nAddress: Innopoli\n 2, FI-0215
0, Espoo, Finland\nTel: +358-505188888 Email: klarke4001@gmail.com\nBorn: 19.08.1981, Changchun, P. R. Chin
a\nKEY COMPETENCIES AND STRENGTHS\nOver 7 years research and working experience in Machine Learning and Dat
a Mining eld.\nStrong data analytical and programming skills especially with Matlab and Python.\nExcellent
English writing and oral presentation skills.\nStrong team-work spirit with experience of working in highly
international environments for years.\nNative Mandarin speaker with Permanent Finnish Resident and Working
Permit.\nWORKING & RESEARCH EXPERIENCE\n2014 - 2015 Data Analyst at Verto Analytics Inc. (Area: data
analytics and image recognition)\n I am working on versatile projects at Verto Analytics Inc. - a Finnis
h Pioneer in Digital \n Media Research and Measurement \nIndustry. \n My responsibilities include:
\n1) \ndeveloping and implementing machine learning algorithms \n for mobile-end App image recognitio
n;\n 2) collaborating with marketing professionals for \n writing market insights reports; \n3) data qua
lity assurance, data cleaning and curation, \n data visualisation, and data production.\n2011 - 2014
Nonnegative Learning for Data Clustering (Area: algorithms and optimis\nations)\n I designed several Mac
hine learning algorithms using matrix factorisation models to \n better detect groups or clusters in
\nvarious data sets. The algorithms can be directly \n applied for, e.g., Recommendation Systems and Mar
ket Segmentation. I published the \n results in 6 scientic journals and papers.\n2011 - 2014 \n
Understanding the Emotional Impacts of Images (Area: image processing)\n I developed several image proce
ssing methods to predict emotional\n impacts of artistic\n images. The methods can improve the performan
ce of Affective Image Classication and \n Retrieval systems. I published the results in 4 scientic journ
als and papers.\n2008 - 2010 \nPinView - A Proactive Personal Information Navigator (Area: multimedi
a retrieval)\n I developed a Gaze-and-Speech-enhanced Content-Based Image Retrieval system that \n ca
n infer the user's search interests based on his or her feedbacks such as eye tracking \n data. I also i
mplemented a client-side browser extension using JavaScript and managed \n to publish the results in 2 s
cientic conferences.\n2005 - 2007 Research Assistant in the Multimedia Laboratory, Jilin University,
China\n I developed matrix transformation techniques for colour image\n and video
compression.\nCV: He Zhang, +358-50-5188888, \nklarke4001@gmail.com\n Page 1/3\nLANGUAGE &
IT SKILLS\nEnglish: Excellent in Writing and Speaking.\n IELTS Score (2006): 7.5/9. I also have
\n English-Chinese Translator & Interpreter experience with certicate issued by \n China HR Minis
try.\nFinnish: Basic. I received full scores in 4 consecutive Aalto University Finnish Exams \n
2006-2007.\nChinese: Native.\nProgramming: \nMatlab, Python, SQL, \nPerl, JavaScript, C / C++, L
aTeX (for document writing)\nPOSITIONS OF TRUST\n2014 Programme Committee Member \nin 2014
International Conference on Articial\n Neural Networks (ICANN), Hamburg, Germany.\
n2014 Reviewer for Scientic Journals, \ne.g., IEEE Transactions on Neural Network\ns
and Learning Systems, Information Sciences, Neurocomputing, Journal of \n Optical
Engineering.\n2013 Membership in European Neural Networks Society (ENNS)\nREFEREES\nProfes
sor \nErkki Oja\n, PhD Supervisor Email: \nerk
ki.oja@aalto.\nDepartment of Computer Science, Aalto University School of Science, Espoo, Finland\nProfesso
r \nTimo Honkela\n, Research Collaborator Email: \ntimo.honkela@helsinki.\n
Department of Language, University of Helsinki, Helsinki, Finland\nSenior Scientist J\norma Laaksonen\n, (f
ormer) PhD Instructor Email: \njorma.laaksonen@aalto.\nDepartment of \nComputer Science, Aalto
University School of Science, Espoo, Finland\nEDUCATION\n2008 2014 Doctor of Science, Aalto University
School of Science, Finland\n Research areas: Machine Learning, Data Mining, and Ima
ge Processing\n Minor: Signal Processing for Tele-communications. PhD Advisor: Prof
. Erkki Oja\n2004 2007 Master of Science, Jilin University, China\n Major: Inf
ormation and Communication Systems\n Master Thesis: Matrix Transformation Technique
s for Color Image Compression\n2000 2004 Bachelor of Engineering, Jilin University, China\n
Major: Communication Engineering\nCV: He Zhang, +358-50-5188888, \nklarke4001@gmail.com\n
Page 2\n/3PUBLICATION LIST\nJournal Articles\n1.He Zhang, Zhirong Yang, and Erkki Oja. \nImproving Cluster
Analysis By Co-initialisations.\n Pattern \nRecognition Letters, 45: 71-77, 2014\n.2.He Zhang, Zhirong Yang
, and Erkki Oja. \nAdaptive Multiplicative Updates for Quadratic Nonnegative Matrix \nFactorisation.\n Neur
ocomputing, 134: 206-213, 2014\n.3.He Zhang, Mehmet Gnen, Zhirong Yang, and Erkki Oja. \nUnderstanding Emot
ional Impact of Images \nUsing Bayesian Multiple Kernel Learning.\n Neurocomputing, 165: 3-13, 2015\n.Confe
rence Papers\n4.He Zhang, Mehmet Gnen, Zhirong Yang, and Erkki Oja. \nPredicting Emotional States of Images
Using \nBayesian Multiple Kernel Learning.\n In Proceedings of the 20th International Conference on Neural
\nInformation Processing (ICONIP), Daegu, South Korea, 2013. \nOral presentation\n.5.He Zhang, Zhirong Yang
, Mehmet Gnen, Markus Koskela, Jorma Laaksonen, Timo Honkela, and Erkki \nOja. Affective Abstract Image Cla
ssication and Retrieval Using Multiple Kernel Learning.\n ICONIP 2013, \nDaegu, South Korea, 2013. \nOral p
resentation\n.6.He Zhang, Zhirong Yang, and Erkki Oja. \nAdaptive Multiplicative Updates for Projective Non
negative Matrix \nFactorisation.\n ICONIP 2012, Doha, Qatar, 2012. \nOral presentation\n.7.Zhirong Yang, \n
He Zhang, and Erkki Oja. \nOnline Projective Nonnegative Matrix Factorisation for Large \nDatasets.\n ICONI
P 2012, Doha, Qatar, 2012. Oral presentation.\n8.He Zhang, Tele Hao, Zhirong Yang, and Erkki Oja. \nPairwis
e Clustering with t-PLSI.\n In Proceedings of the \n22nd International Conference on Articial Neural Networ
ks (ICANN), Lausanne, Switzerland, 2012.\n Travel Grant Award\n.9.He Zhang, Mats Sjberg, Jorma Laaksonen, a
nd Erkki Oja. \nA Multimodal Information Collector for \nContent-Based Image Retrieval System.\n ICONIP 201
1, Shanghai, China, 2011. \nOral presentation\n.10.He Zhang, Eimontas Augilius, Timo Honkela, Jorma Laakson
en et al. \nAnalysing Emotional Semantics of \nAbstract Art Using Low-Level Image Features.\n In Proceeding
s of the 10th International Conference on \nAdvances in Intelligent Data Analysis (IDA), Porto, Portugal, 2
011. \nOral presentation\n.11.\nHe Zhang, Teemu Ruokolainen, Jorma Laaksonen, Christina Hochleitner, and Ru
dolf Traunller. \nGaze \nand Speech-Enhanced Content-Based Image Retrieval in Image Tagging.\n ICANN 2011,
Espoo, Finland, \n2011. \nPoster\n presentation.\n12.Zhirong Yang, \nHe Zhang, Zhijian Yuan, and Erkki Oja.
\nKullback-Leibler Divergence for Nonnegative Matrix \nFactorisation.\n ICANN 2011, Espoo, Finland, 2011. \
nOral presentation\n.Technical Reports\n13.He Zhang, Markus Koskela, and Jorma Laaksonen. \nReport on Forms
of Enriched Relevance Feedback. \nTechnical Report TKK-ICS-R10, Helsinki University of Technology, Departme
nt of Information and \nComputer Science. Presented at PinView meeting, University College London, 2008.\n1
```

Figure 4.6: A Screenshot of Extracted content

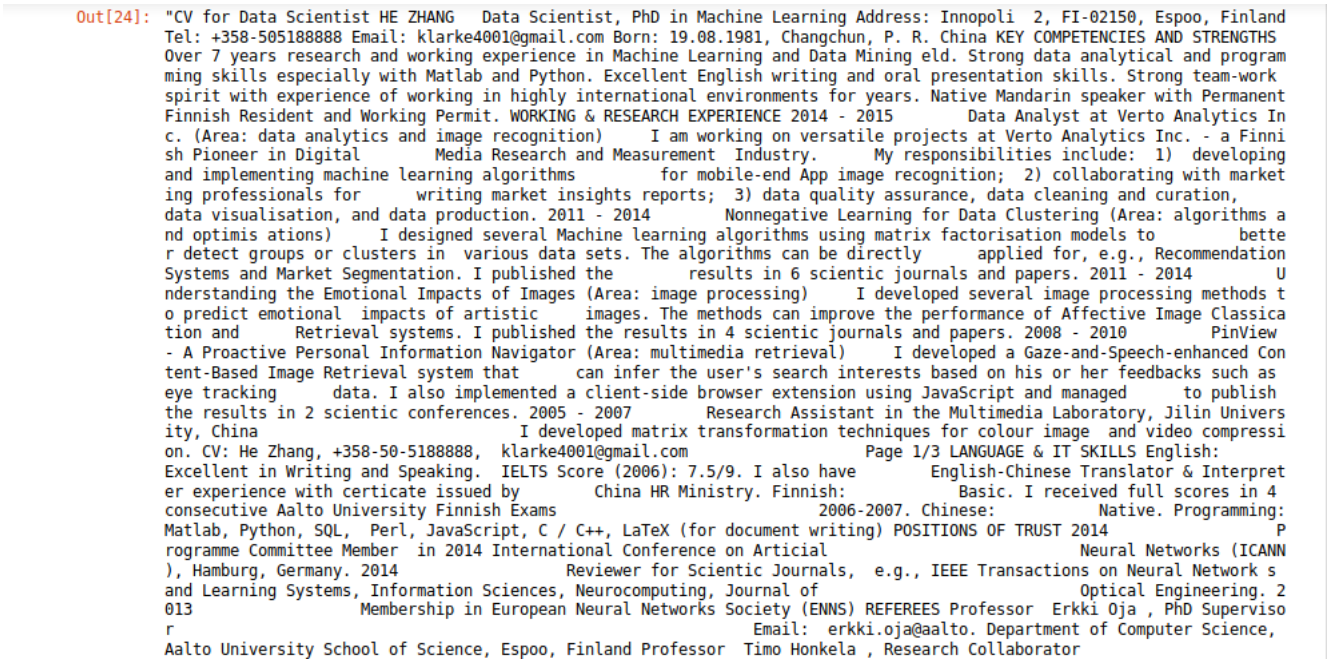

Figure 4.7: A Screenshot of Preprocessing content

In the Figure 4.8 below, we can see that the extraction algorithms were able to extract:

1. The type of the uploaded file
2. The country of residence of the user
3. The user's phone number
4. The email address of the user
5. The first and last user names
6. As well as the skills of the user.

```
In [5]: get_file_type(cv_file)

Out[5]: 'pdf'

In [6]: text = get_pdf_content(cv_file)
        extract_country(text)

Out[6]: 'Finland'

In [7]: extract_number(text)

Out[7]: '+358-505188888'

In [8]: extract_email(text)

Out[8]: 'klarke4001@gmail.com'

In [12]: extract_first_and_last_names(text)

Out[12]: ('HE', 'ZHANG')

In [14]: skills = extract_skills(text)
         skills

Out[14]: {'algorithms': 4,
          'analytical': 1,
          'analytics': 3,
          'communication': 2,
          'data analysis': 1,
          'data analytics': 1,
          'data cleaning': 1,
          'data clustering': 1,
          'data mining': 2,
          'data production': 1,
          'data quality': 1,
          'data visualisation': 1,
          'engineering': 3,
          'ielts': 1,
          'image processing': 3,
          'image recognition': 2,
          'image retrieval system': 2,
          'insights': 1,
          'javascript': 2,
          'machine learning': 4,
          'market segmentation': 1,
          'matlab': 2,
          'matrix factorisation': 2,
          'mobile': 1,
          'multimedia retrieval': 1,
          'networks': 3,
          'neural networks': 3,
          'nonnegative learning': 1,
          'perl': 1,
          'presentation': 4,
          'processing': 4,
          'programming': 2,
          'python': 2,
          'recommendation systems': 1,
          'retrieval systems': 1,
          'sql': 1,
          'technical': 2}
```

Figure 4.8: A Screenshot of Keys Informations

Figure 4.9 shows the output of the result of the recommendation made by our predictive algorithm where we can see the different percentages of matching between the user's CV and the proposed job offers.

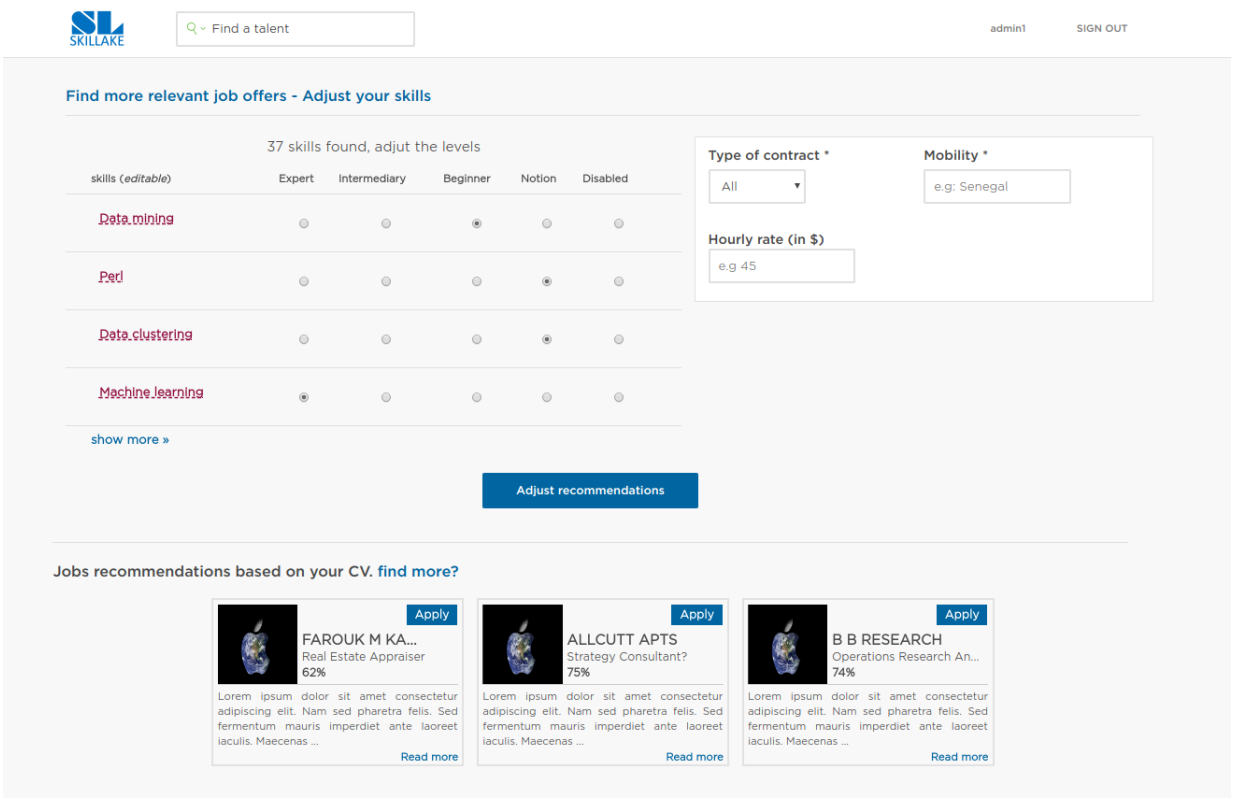

Figure 4.9: A Screenshot of Recommendation output

This process allows the user to save time, to be sure that it corresponds to the recommended job offer using the percentage rates displays and after these recommendations with one click the user will be able to apply for the job of these dreams as shown in the Figure 4.10 and Figure 4.11

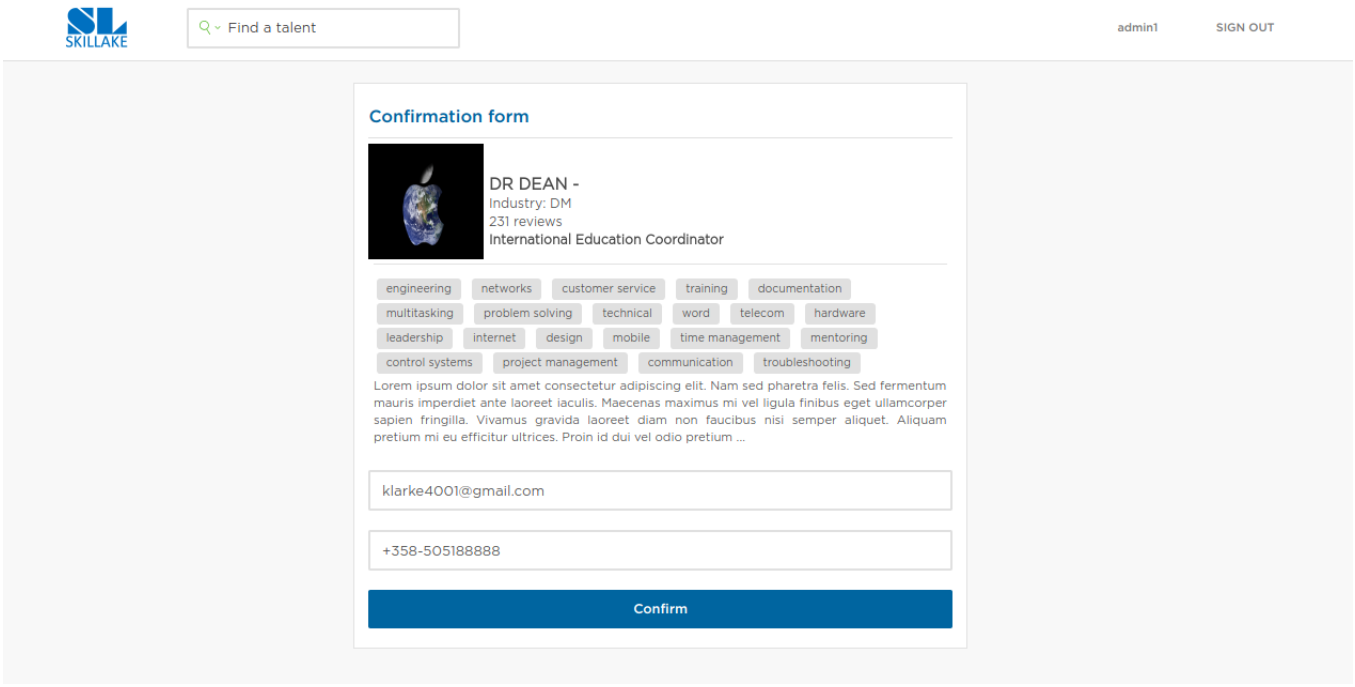

Figure 4.10: A Screenshot of Jobs Details

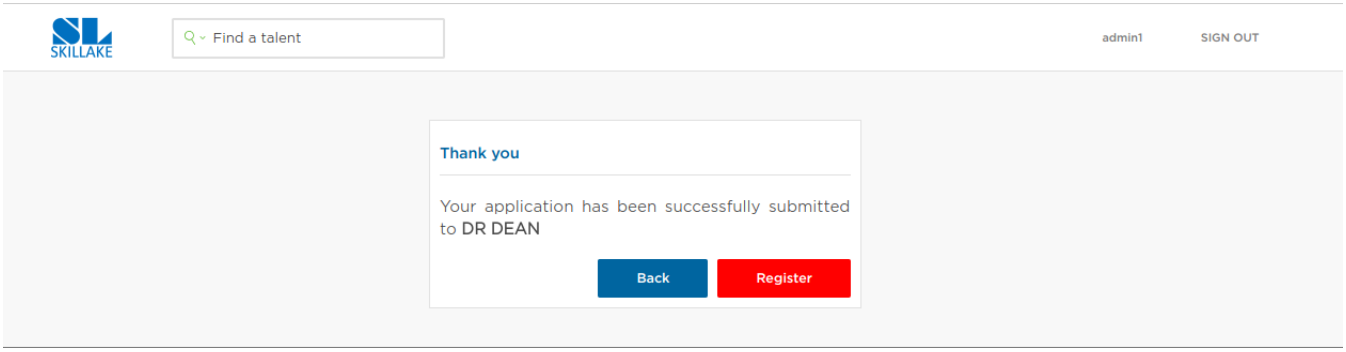

Figure 4.11: A Screenshot of Applied job

5. Conclusion and Perspectives

In this Thesis, we presented a hybrid, personalized recommender system for job seeking and recruiting websites. A clustering based ranking algorithm is taking into account the business requirements in the job seeking and recruiting process.

This chapter first summarizes the work put forward in the thesis. A discussion on the research issues and contributions is presented, with pointers to future work.

5.1 Conclusion

The work presented in this thesis consists of two main parts: As a first step, we conducted a literature review of four different online recruitment platforms, which allowed us to learn about the different existing methods and the most popularly used for designing an online recruitment recommendation system.

We did a comparative study of the different techniques used to design the recommendation systems. From this study, we decided to adopt the hybrid system for the design of the predictive algorithm, a choice that was made to overcome the various weaknesses of each technique.

We explored various data mining techniques useful for the design of existing recommendation systems by collecting CVs and job postings from various fields of specialization and similarity measures as well as unsupervised learning methods.

At the end of this exploration, we were able to combine these different techniques to implement the predictive algorithm used in the proposed recommendation system.

At first, the use of text mining techniques on CVs and job offers allowed us to structure the information they contain thanks to the use of supervised algorithms. Resumes and vacancies are now categorized.

In a second step, methods of extracting information from textual analysis to extract a set of keywords associated with the document to enrich the description within a predictive algorithm because it contributes to the performance of the prediction made by the algorithm.

For the implementation of the predictive algorithm, as an approach we have proposed a hybrid recommendation system adapted to the cold start problem which the platform has faced, this method proved superior during experiments on a real data set (collect initially) compared to standard methods.

5.2 Perspectives

Following these promising and encouraging results, our first recommendation concerns improving the quality of prediction obtained by considering the social impact in the network constitute by job seekers and recruiters. We want to improve the quality of keyword extraction combined with the use of similarity metrics.

We also want to improve the quality of clustering that will be done by combining the unsupervised learning methods for example instead of using the Kmeans unsupervised algorithm we use the DBSCAN algorithm that distributes clusters depending on the density of each data.

6. Appendix

6.1 Distance

6.1.1 Definition.

A distance **d** on a set **E** is a function **d** : **E**×**E**→ [0,+∞) which verify the following properties:

∀ x,y,z ∈ **E**

1. **d**(x,y) > 0 ∀ $x \neq y$ and **d**(x,y) = 0 iff x=y
2. **d**(x,y) = **d**(y,x)
3. **d**(x,y) ≤ **d**(x,z)+**d**(z,y)

The simplest and most common example of a distance measure is the **Euclidean distance**

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{6.1.1}$$

Where **n** is the dimension of **E** or the number of attributes for the dataset and x_i and y_i are the i^{th} components of data objects x and y, respectively. A generalization of the Euclidean distance is the **Minkowski distance** defined as

$$d(x,y) = \sqrt[1/p]{\sum_{i=1}^{n} (x_i - y_i)^p} \tag{6.1.2}$$

where p is the degree of the distance, for some value of p, the generic Minkowski distance is known with specific names:

- For p=1, the generic Minkowski is the **Manhattan distance** also called the L^1 norm.
- For p=2, we obtain the Euclidean distance also called the L^2 norm.
- For $p \to +\infty$, the generic Minkowski is the **supremum distance** or the L^∞ norm. It compute the maximum difference between any dimension of the data objects.

6.2 Linear algebra background

Let A be an $m \times n$ matrix of real numbers and A^T its transpose.

Theorem1 (Spectral Theorem)

If A is symmetric i.e $A^T = A$, then A is orthogonally diagonalizable and has only real eigenvalues. In order words, there exist real numbers $\lambda_1, \lambda_2, ..., \lambda_n$ called eigenvalues, and orthogonal, non-zero real vectors $\vec{v_1}, \vec{v_2}, ..., \vec{v_n}$ called eigenvectors such that $A\vec{v_i} = \lambda_i \vec{v_i}$

Proposition 1

If A is any $m \times n$ matrix of real numbers, then the matrix AA^T and A^TA are both symmetric.

Proof

We have $(AA^T)^T = (A^T)^T A^T = AA^T$ and $(A^TA)^T = A^T(A^T)^T = A^TA$

Proposition 2

The matrices AA^T and the A^TA share the same nonzero eigenvalues.

Proof

First of all AA^T and A^TA are both symmetric so they have non-zero eigenvectors .

Let $\overrightarrow{v}$ be a non-zero eigenvector of AA^T associated to the eigen value $\lambda \neq 0$ This means:

$$(AA^T)\overrightarrow{v} = \lambda \overrightarrow{v} \Rightarrow A^TA(A^T\overrightarrow{v}) = \lambda(A^T\overrightarrow{v}) \tag{6.2.1}$$

This means that the vector $A^T\overrightarrow{v}$ is an eigenvector of A^TA with the eigenvalue λ, the remain point is to show that $A^T\overrightarrow{v}$ is not the zero vector

suppose by contradiction that $A^T\overrightarrow{v}$ were zero, then $\lambda\overrightarrow{v}$ would be zero but neither λ and $\overrightarrow{v}$ are not equal to zero. So $A^T\overrightarrow{v}$ is a non-zero eigenvector

Proposition 3

The eigenvalues of AA^T and A^TA are nonnegative numbers.

Proof.

Let $\overrightarrow{v}$ an eigenvector of AA^T with eigenvalues λ Then

$$\begin{aligned} \left\|A^T\overrightarrow{v}\right\|^2 &= (A^T\overrightarrow{v})^T(A^T\overrightarrow{v}) \\ &= (\overrightarrow{v})^T(AA^T)\overrightarrow{v} \\ &= \lambda(\overrightarrow{v})^T\overrightarrow{v} \\ &= \lambda\left\|(\overrightarrow{v})^T\right\| \end{aligned} \tag{6.2.2}$$

Since the lengths are non negative, we can deduce that λ is non negative.

6.3 Statistics Background

Let us measuring a single variable A (such as the IQ of randomly selected individuals) n times(here, m=1). Let B (m,n) be the data matrix , m is the number of data points, n is the number of dimensions. We denoted by $a_1, a_2, ..., a_n$ the n measurements. Despite of the fact that the mean is rarely known in practice we can still estimate it as the sample average:

$$\mu_A = \frac{1}{n}(a_1 + a_2 + ... + a_n) \tag{6.3.1}$$

It tell us where the measurements are centered, we can also be interested in knowing how spread are the measures. This is quantify by the variance which is also generally unknown in practice. The sample

variance can be estimated as:

$$Var(A) = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_A)^2 \tag{6.3.2}$$

If we are measuring two variables A, B in a population, the natural question is to know the relation ship between A and B(For instance in our first example we want to know the relation ship between the grade and the IQ). We can use the **Covariance** of A and B defined as:

$$Cov(A,B) = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_A)(b_i - \mu_B) \tag{6.3.3}$$

Theorem 2

Given an $m \times n$ real matrix A, we can decompose it as

$$A = U \times \Gamma \times V^T \tag{6.3.4}$$

Where :

- U is a column orthonormal $m \times r$ matrix, with r the rank of the matrix A.
- Γ is a diagonal $r \times r$ matrix, where the elements are sorted in descending order.
- V is a column orthonormal $n \times r$ matrix

The Γ diagonal matrix contains the singular values, which will always be positive and sorted in decreasing order. The U matrix is interpreted as the "item-to-concept" similarity matrix, while the V matrix is the "term-to-concept" similarity matrix. This SVD decomposition is **unique** and the values of the diagonal Γ are called **singular values**.

Fact

The inverse of an orthonormal matrix is its transpose.

Theorem 3

If S is a real and symmetric matrix, then $S = U \times \Gamma \times U^T$, where the columns of U are the eigenvectors and the eigenvalues are the values of the diagonal of Γ.

Proof

Let U be the matrix where the column are the eigenvectors of S. We have $S \times U = U \times \Gamma$

$\Longrightarrow S = U \times \Gamma \times U^{-1} = U \times \Gamma \times U^T$.

Let define D, the $m \times m$ matrix such that $D = AA^T$, then by proposition 6.2, D is real and symmetric, so by theorem 6.3, one can say that the eigenvectors of D are the column of U and the eigenvalues are the values of the diagonal of Γ^2, which are the squares of the λ_i elements (The singular values of A). The elements of the column of U are called singular vectors similar to the principal components in PCA.

6.4 Code

If you are interesting about the python code designed throughout this thesis and you would like to get it, please do not hesitate to contact me at this mail address lionel.ng.tondji@aims-senegal.org

References

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[2] Aristomenis S Lampropoulos and George A Tsihrintzis. *Machine Learning Paradigms: Applications in Recommender Systems*, volume 92. Springer, 2015.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[4] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

[5] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.

[6] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM, 2010.

[7] Suresh K Gorakala and Michele Usuelli. *Building a recommendation system with R*. Packt Publishing Ltd, 2015.

[8] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[9] Julie Séguéla. *Fouille de données textuelles et systèmes de recommandation appliqués aux offres d'emploi diffusées sur le web*. PhD thesis, Paris, CNAM, 2012.

[10] Anil Poriya, Tanvi Bhagat, Neev Patel, and Rekha Sharma. Non-personalized recommender systems and user-based collaborative recommender systems.

[11] Alexandrina Singleton. Recommender systems : Collaborative filtering and content-based recommending. http://slideplayer.com/slide/5692490/, 2016.

[12] D. Jannach, Felfernig A. Zanker, M., and G. Friedrich, editors. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[13] M. Pazzani and D. Billsus. Learning and revising user profiles: the identification of interesting web sites. *Mach. Learn.*, 27(3):313–331, 1997.

[14] Richard Steinmetz David, Domingo Gregory, Liu Aiko, and Tzu-Yu Chen Amy. The restaurant dilemma: Personalized recommendations for groups of people. https://nycdatascience.com/blog/student-works/restaurant-recommendations-groups-people/, 2016.

[15] Daniel Tuan. Recommender systems-how they works and their impacts. http://findoutyourfavorite.blogspot.sn/2012/04/content-based-filtering.html, 2015.

[16] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[17] R. Decker and H. J. (Eds.) Lenz, editors. *Advances in Data Analysis: Proceedings of the 30th Annual Conference of The Gesellschaft FÃ¼r Klassifikation EV, Freie Universitat Berlin, March 8-10*. Springer Science and Business Media, 2006.

[18] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[19] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating 'word of mouth'. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[20] Shah Khusro, Zafar Ali, and Irfan Ullah. *Recommender Systems: Issues, Challenges, and Research Opportunities*, pages 1179–1189. Springer Singapore, Singapore, 2016.

[21] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[22] Ken Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95)*, 1995.

[23] Shaha T Al-Otaibi and Mourad Ykhlef. A survey of job recommender systems. *International Journal of Physical Sciences*, 7(29):5127–5142, 2012.

[24] Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User modeling and user-adapted interaction*, 11(3):203–259, 2001.

[25] Keith Bradley and Barry Smyth. Personalized information ordering: a case study in online recruitment. *Knowledge-Based Systems*, 16(5):269–275, 2003.

[26] Bradford Heap, Alfred Krzywicki, Wayne Wobcke, Mike Bain, and Paul Compton. Combining career progression and profile matching in a job recommender system. In *Pacific Rim International Conference on Artificial Intelligence*, pages 396–408. Springer, 2014.

[27] Wenxing Hong, Siting Zheng, Huan Wang, and Jianchao Shi. A job recommender system based on user clustering. *Journal of Computers*, 8(8):1960–1967, 2013.

[28] Danielle H Lee and Peter Brusilovsky. Fighting information overflow with personalized comprehensive information access: A proactive job recommender. In *Autonomic and autonomous systems, 2007. ICAS07. Third international conference on*, pages 21–21. IEEE, 2007.

[29] Rachael Rafter, Keith Bradley, and Barry Smyth. Personalised retrieval for online recruitment services. In *The BCS/IRSG 22nd Annual Colloquium on Information Retrieval (IRSG 2000), Cambridge, UK, 5-7 April, 2000*, 2000.

[30] Amit Singh, Catherine Rose, Karthik Visweswariah, Vijil Chenthamarakshan, and Nandakishore Kambhatla. Prospect: a system for screening candidates for recruitment. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 659–668. ACM, 2010.

[31] Matthias Hutterer. *Enhancing a job recommender with implicit user feedback*. na, 2011.

[32] Pascal Soucy and Guy W Mineau. Beyond tfidf weighting for text categorization in the vector space model. In *IJCAI*, volume 5, pages 1130–1135, 2005.

[33] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

[34] tf idf. Tf-idf. http://www.tfidf.com/, 2018.

[35] Cambridge University Press. Tf-idf weighting. https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html, 2009.

[36] Neal Lathia, Stephen Hailes, and Licia Capra. The effect of correlation coefficients on communities of recommenders. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 2000–2005. ACM, 2008.

[37] Margaret Rouse. Business analytics : Data sampling. http://searchbusinessanalytics.techtarget.com/definition/data-sampling, 2014.

[38] Flora S Tsai. A visualization metric for dimensionality reduction. *Expert Systems with Applications*, 39(2):1747–1752, 2012.

[39] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.