

APPLYING DATA MINING FOR JOB RECOMMENDATIONS BY EXPLORING JOB PREFERENCES

*Thesis submitted in partial fulfillment of the requirements for the award
of degree of*

Master of Engineering
in
Computer Science and Engineering

Submitted By
Anika
(Roll No. 801232003)

Under the supervision of:
Dr. Deepak Garg
Associate Professor and Head



**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004**

July 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Applying Data Mining For Job Recommendations By Exploring Job Preferences*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Deepak Garg* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.


(Anika)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.


(Dr. Deepak Garg)

Associate Professor

Computer Science and Engineering Department

Countersigned by


(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala


(Dr. S. K. Mohapatra)
Dean (Academic
Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my guide, **Dr. Deepak Garg**, Associate Professor and Head, Computer Science and Engineering Department, Thapar University, who has been very concerned and has aided for all the guidance essential for the thesis report. He has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to Dr. Ashutosh Mishra, P.G. Coordinator, for the motivation and inspiration that triggered me for the seminar work.

I would also like to thank the staff members and my colleagues who were always there in the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis.

Most importantly, I would like to thank my parents and the Almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.



Anika

801232003

ME (CSE)

ABSTRACT

Job recommender systems are desired to attain a high level of accuracy while making the predictions which are relevant to the customer, as it becomes a very tedious task to explore thousands of jobs, posted on the web, periodically. Although a lot of job recommender systems exist that use different strategies, here efforts have been put to make the job recommendations on the basis of candidate's profile matching as well as preserving candidate's job behavior or preferences. Firstly, the rules predicting the general preferences of the different user groups are mined. Then the job recommendations to the target candidate are made on the basis of content based matching as well as candidate preferences, which are preserved either in the form of mined rules or obtained by candidate's own applied job history. Through this technique, a significant level of accuracy, around eighty percent, has been achieved over other basic methods of job recommendations.

TABLE OF CONTENTS

<i>CERTIFICATE</i>	<i>ii</i>
<i>ACKNOWLEDGEMENT</i>	<i>iii</i>
<i>ABSTRACT</i>	<i>iv</i>
<i>TABLE OF CONTENTS</i>	<i>v</i>
<i>LIST OF FIGURES</i>	<i>vii</i>
<i>LIST OF TABLES</i>	<i>ix</i>
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Thesis Structure	4
CHAPTER 2: LITERATURE SURVEY	5
CHAPTER 3: PROBLEM STATEMENT	10
CHAPTER 4: PROPOSED ALGORITHM	13
4.1 Basic Terms Used	13
4.2 Assumptions	15
4.3 Basic Steps of Proposed Algorithm	16
4.3.1 Data Acquisition	16
4.3.2 Feature Selection	16
4.3.3 Data Categorization	18
4.3.4 Mining Decision Tree Induction Rules	22
4.3.5 Phase I Recommendation	26
4.3.6 Phase II Recommendation	29
CHAPTER 5: IMPLEMENTATION	33
5.1 Experimental Setup	33
5.1.1 Datasets	33
5.1.2 Environment	33
5.2 Data Mining Using Orange	34
5.3 Phase I Recommendation	37
5.4 Phase II Recommendation	39
CHAPTER 6: RESULTS AND DISCUSSIONS	42
6.1 Results Analysis	42

6.2 Discussions	44
CHAPTER 7: CONCLUSION.....	47
7.1 Answering the Research Question	47
7.2 Future Work.....	48
REFERENCES	49
LIST OF PUBLICATIONS	53

LIST OF FIGURES

Fig. No.	Name	Page No.
Figure 1.1	Proposed job recommender system framework	3
Figure 2.1	Basic classification of recommender system strategies	5
Figure 2.2	CASPER system architecture	7
Figure 4.1	Normal procedure proposed in making the job recommendations	15
Figure 4.2	Relevant features related to the candidate	17
Figure 4.3	Relevant features related to the job	17
Figure 4.4	Specification to generalization for a candidate	21
Figure 4.5	Specification to generalization for a job	21
Figure 4.6	Hierarchical representation of OOPs languages	22
Figure 4.7	A decision tree example	22
Figure 4.8	Evaluation results of various learning algorithms	23
Figure 4.9	Classification tree attribute selection criteria	24
Figure 4.10	Flow chart for calculation of decision tree induction rules	25
Figure 4.11	Steps for phase-I recommendation generation	27
Figure 4.12	Flow chart for final weight score calculation in phase-I recommendations	28
Figure 4.13	Steps for phase-II recommendation generation	30
Figure 4.14	Flow chart for calculation of cosine based similarity	31
Figure 4.15	Flow chart for final weight score calculation in phase-II recommendations	32
Figure 5.1	Experimental setup in Orange tool	34
Figure 5.2	Association rules example showing confidence and lift	35

Figure 5.3	Pictorial representation of a decision tree induction rule	36
Figure 6.1	Comparison of prediction accuracy for phase-I	42
Figure 6.2	Comparison of prediction accuracy for phase-II	43
Figure 6.3	Comparison of prediction accuracy for phase-I and phase-II	44

LIST OF TABLES

Table No.	Name	Page No.
Table 4.1	Categorization of candidate features	18
Table 4.2	Categorization of job features	20
Table 5.1	Sample matrix representation of mined rules against different job categories	36
Table 5.2	Sample jobs in the database	37
Table 5.3	Shortlisted jobs after step 1	38
Table 5.4	Calculation of cosine based similarity in step 2.	38
Table 5.5	Final weight calculation in step 4	39
Table 5.6	Final sorting of the jobs after step 5 of phase-I	39
Table 5.7	Generation of normalized rules weight using 4 preference matrices	40
Table 5.8	Final recommendation of phase-II after step 5	40
Table 6.1	Comparison of prediction accuracies during phase-I	42
Table 6.2	Comparison of prediction accuracies during phase-II	43
Table 6.3	Comparison of prediction accuracies for phase-I and phase-II recommendations	44
Table 6.4	Comparison between the different job recommender systems	45

CHAPTER 1

INTRODUCTION

1.1 Background

Recommender systems are being used in almost every internet based ecommerce websites. However the type of recommendations provided may vary according to the domain of its usage. For example, in an online shopping site for clothing, it will be more favorable to provide generalized recommendations regarding the latest trends and fashion in the market, as most of the users are expected to go along that way. However, this case is little bit different in e-recruitment sites. Here, it will be favorable to provide more personalized and profile based job recommendations. In job recommender systems, there are varieties of customers/ candidates, having different education level, experience and skills. Based on their respective background details, each one expects to get only those job recommendations which are highly relevant for the respective candidate.

A job recommender system is expected to provide recommendations in 2 ways: firstly recommending most eligible candidates for the specified job, to the recruiters and secondly, recommending jobs to the aspiring candidates according to their matching profiles. The focus of this paper is the second part only i.e. to recommend jobs to the candidates according to their matching profiles. However there can be seen some gap between the existing systems. Here an example is shown:

Suppose the profile of a candidate and job can be represented as shown below. These representations for the candidate as well as job will be used throughout this thesis.

1. Candidate Profile: {age, gender, marital status, education, major, education level, experience, current location, skills possessed}.
2. Job Profile: {field, required education, required experience, required skills, level of company (A (highest), B, C, D), position level offered by the company (A (top positions), B, C, D), pay-scale (High (H), Medium (M), Low (L)), Job Location}.

Example 1.1: There are 2 candidates with following profiles:

- 1) {27, male, unmarried, graduate, Computer Science, 65%, 6, New Delhi, (Python, Oracle, Machine Learning, English)}
- 2) {35, male, married, masters, Computer Science, 65%, 7, New Delhi, (Python, Oracle, Machine Learning, English)}

And there are 2 jobs with following requirements:

- 1) {CSE, graduate, 5, {Python, Machine Learning, English}, B, C, M, New Delhi}
- 2) {CSE, graduate, 5, {Python, Machine Learning, English}, B, B, M, Bangalore}.

Now, if both candidates are given option to select their prioritized jobs then the case may be that 1st candidate selects 2nd job as priority job whereas 2nd candidate may give priority to 1st job.

Explanation: Although both the candidates possess almost equivalent profiles, both also qualify for both the jobs, still their preferences regarding the jobs are different. The 1st candidate may have chosen for the 2nd job as he considered higher position (level B) and does not have a location constraint in his personal life. However, 2nd candidate, considering his age and marital status, he is normally expected to go with the first job as, amongst both the jobs, the only difference is of the position offered. Else the package offered, company level rest all is same. And also the location of 1st company is similar to his current location. Hence, he may compromise for the position offered and decide to choose 1st job as his preference job.

Talking about the human nature, as one is in the youth stage, the more enthusiastic he/she is, and may be ready to take any risk. But as he/ she grow older, a certain level of maturity is gained and one is more tilted towards stable and more promising and less risky decisions in his/her life. The same is applicable while taking decisions regarding one's source of living. While exploring the job recommender systems, interesting facts and figures were obtained regarding the nature of these job applicants. These job applicants, who belong to different age groups, gender etc, show a certain level of similarity, in nature, while applying for jobs. This only formed the basis of the research in this field. Here it is tried to explore these generalized job behavior of candidates having different genre, in the form of classification rules.

Later, these mined rules were applied, for providing initial recommendations, to the new candidate according to his genre.

As discussed in the example, two candidates having similar looking profiles may have different job tastes. Here, job taste can be defined as the preference criterion considered before applying for a particular job. For one, preference can be of getting a job in higher company, as opposed to the other who may be interested in having a job which offers higher salary. Considering this, the second phase of recommendations, are provided to the respective customer according to his/ her job taste or preferences. These job preferences are extracted from the already applied jobs basket of the candidate.

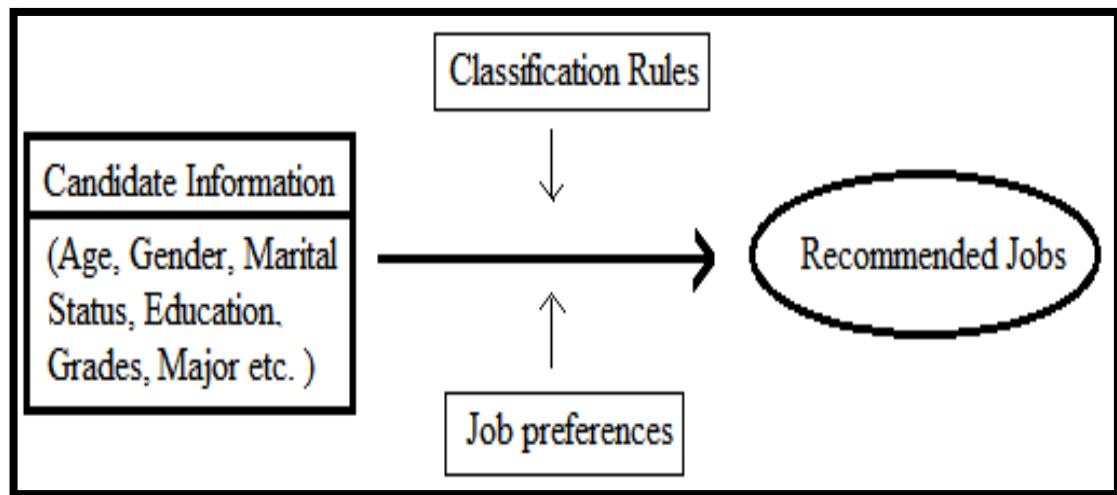


Figure 1.1: Proposed job recommender system framework

So, the efforts have been put to judge the gap between the candidate's choices, belonging to different groups, regarding the selection of different offered jobs. Here, it is tried to foresee the customer preferences regarding the jobs on four basic parameters of company preferences, position offered, pay-scale offered and job location. Instead of tracking the past history of the candidate, his current likings/preferences are focused upon. For a new customer, the system firstly tries to impose the general job preferences, obtained through mined rules, according to the age-group, gender, educational background, grades, current employer, salary pay-scale, location etc. under which the candidate lies and as the candidate becomes active within the system by applying for the suitable looking jobs, his/her own job preferences are taken into consideration, by looking over his past latest applied jobs history.

1.2 Thesis Structure

Rest of this thesis is arranged in the following manner:

- Chapter 2 tells about the earlier work done in the field of generalized recommender systems as well as job recommender systems. It summarizes the work done in the field of job recommender systems.
- Chapter 3 forms and defines the problem statement. Here gap existing between the current job recommender systems is stressed upon.
- Chapter 4 describes the experimental set up used while working on the problem. It deals with all the datasets, tools and technology used in finding the solution for the problem.
- Chapter 5 is concerned with the implementation part. It discusses how the problem discussed in Chapter 3 was solved. It shows how the steps used to solve the problem, were implemented.
- Chapter 6, Results and Discussions, focuses on the results obtained after the proposed algorithm was implemented and their analysis.
- Chapter 7 concludes the work and also discusses about the future scope of this work.

CHAPTER 2

LITERATURE SURVEY

The recommender systems are quite popular as they help to find the customer what they want within a very less time. The recommendations are the guesses made by the system about an item that a customer will most likely prefer. These help to increase the site's popularity as well as sales (in case of business sites). Although there are generalized recommender systems, but personalized recommender systems are more focused upon. Personalized recommender systems are expected to change the content or items according to the user's profile and preferences. Analogous to the personalized recommender systems, generalized recommender systems provide same content to all the users.

There are various types of recommender system strategies: Content Based, Collaborative Based, Demographic, Knowledge Based and Hybrid Recommender Systems as shown in the Figure 2.1 [1].

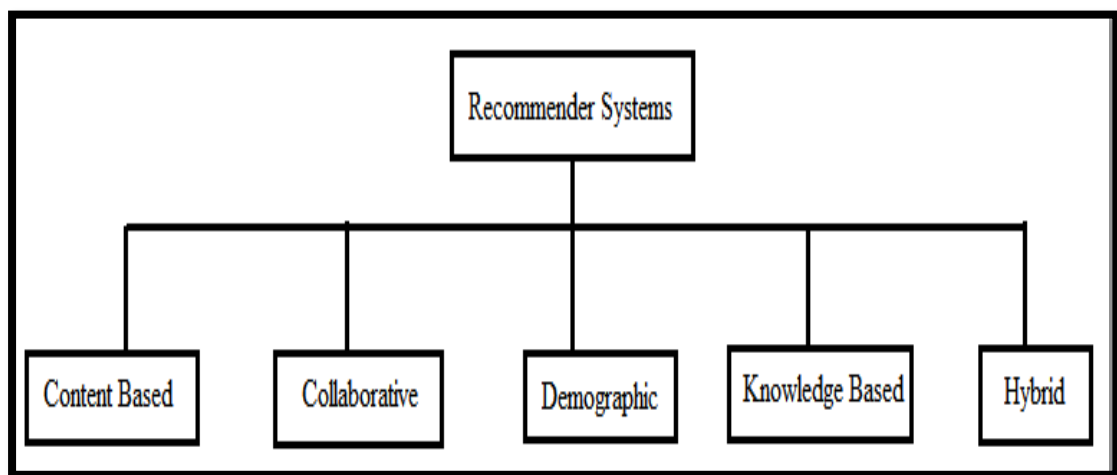


Figure 2.1: Basic classification of recommender system strategies

Content based systems deal with product features and user profiles and their matching. It first takes into account the product/item features or profile. After that it matches it with that of the user profile, taste or user requirements for the particular item or product. Based on the similarity index, the item that most satisfies the used need is recommended on the top [2].

Collaborative based systems focus on item-item similarity or user-user similarity. It calculates the similarity either between the items or between the user's profiles. It then looks for the user tastes. Then either it makes the recommendation on the basis of item-item similarity score i.e. if the user likes a particular item, then he might also like the items that are similar to that item. Looking the other way round, it can also make the recommendations on the basis of user-user similarity i.e. if this user has a particular taste, then it searches for other users that have similar tastes with the respective user and recommend their items to the target user [3-5].

Other than these basic strategies, Demographic systems aim to find out user's personal attributes through interactive dialog or other methods and then try to recommend the items [8]. Knowledge based system are based on the inferences drawn according to the user needs and preferences [6, 7].

However, now- a-days hybrid systems are more common in usage. The hybrid systems are a combination of two or more of the above systems with certain modifications, as per requirements [7-9].

Now concentrating upon the job recommender systems, a lot of research has been carried out in this field. A variety of job recommender systems already exist that try to explore one or the other aspect of the information by applying different methodologies [10, 11].

One of the earliest job recommender systems, CASPER tries to reduce the information overload of the jobs, by providing personalized recommendations to the candidates. CASPER ACF system tries to enhance the recruitment systems by using the CBR (Case Based Reasoning) and fuzzy techniques for searching and also ACF (Automated Collaborative Filtering) for making the personalized recommendations. Firstly the user profile is built by tracking its preference and then this information is used by ACF, for finding similar candidates as that with the respective candidate, for the personalized job recommendations to the respective candidate. However, CASPER PCR uses a two stage process where in first stage, the similarity between the candidate's query and the job is calculated, which is in turn is done on the server-side. In the second stage, on the client side, retrieved jobs relevance is calculated according to the target user profile and the jobs are finally sorted in the order of their relevance [12-13].

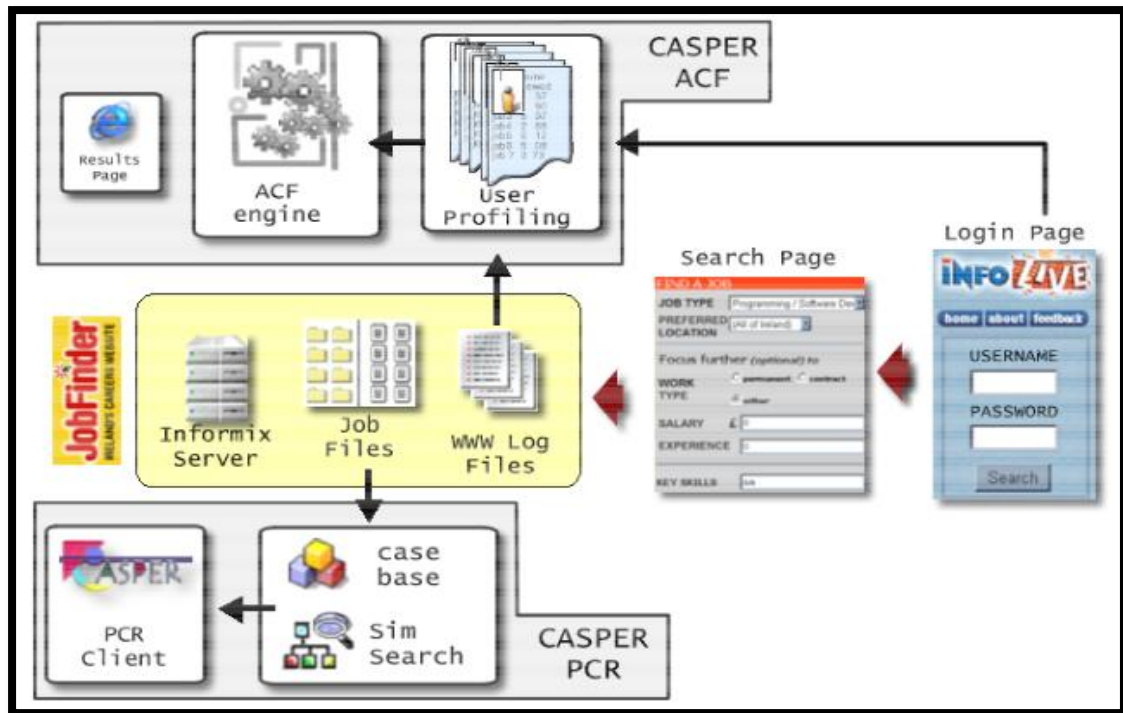


Figure 2.2: CASPER system architecture [13]

Proactive is an adaptive system that provides four different interfaces to capture the candidate's job tastes/ preferences. It integrates preferred jobs, recommended jobs, advanced search and most recent jobs in one system for providing efficiency in job recommender systems. First of all the recent jobs are shown to the candidate. After that the candidates preferences are stored by candidate's activity and these jobs are referred as the preferred jobs. Also the extra jobs that the candidate searches for are also trapped by the system and referred to the jobs of advanced search. And at last by seeing the preferred jobs and advanced search jobs, the recommendations of jobs are made to the candidate that are referred to as the recommended jobs [14].

The bilateral system is quite different from others as it considers both parties of the job recommender systems. Its main focus is to follow a bilateral approach in matching the jobs to the candidates as well as matching candidates to the jobs. It provides 2-sided recommendations, where first is the CV recommender and next is the Job recommender and afterwards integrates them into one system. The CV recommender here recommends the CVs to the recruiters, according to their preferences for the same job that they have earlier selected. The job recommender system works by recommending jobs that are built upon the previous ratings given by the candidates to the preferred jobs. Here it implements an explicit feedback mechanism of rating [15].

Although bilateral systems tried to integrate the two sided job recommender systems, the Reciprocal recommender system also tried to satisfy both the sides by judging their application, MEET, on four different parameters: reciprocity, limitedness, passiveness and sparsity. They integrated above mentioned four properties into one system. However these work by calculating the reciprocal value, after considering the candidate's resume information and candidate's interaction history. This is done with the help of bipartite graphs. The final recommendations are provided by ranking based on the reciprocal scores [15-17].

iHR system concentrates on user's characteristics. It clusters the candidates according to their activity within the system, into three major groups: proactive, passive and moderate. After judging the nature of these groups, it first clusters the candidates according to their profile matching with the respective activity graph of each group. After that it applies three different and major recommendation strategies, namely: content based recommendation, collaborative recommendation and hybrid recommendation on these according to their suitability [18].

Machine learned job recommenders focus on the past transition histories. They built up a supervised learning system and tried to learn from the past transition history of the candidate. They divided the samples into three different sets and applied DTNB, decision table/ naïve Bayes hybrid classifier on them [19]. This approach helped them to obtain a significant level of accuracy [20].

Dynamic user profile based system provided a solution to the job recommendations, from two basic perspectives of time and dimensionality. In this they dynamically updated the candidate's profile over the interested and non-interested feature sets, extracted over a time period, through the information gain concept. Here the focus is on the problem of non-updation of user profile for a long time period as most of the candidates don't update their profile timely. As a result the recommendations become stale and inefficient. It happens because the user preferences get changed according to the time and experience but the system is not updated accordingly. So they first extracted the important features from the candidate's preferred jobs using the TF-IDF values and those that crossed the threshold value were updated and appended in the candidate's profile. Thus two sets: interested and uninterested sets were created and feature extraction was done

periodically. However the very first recommendations were made on the basis of collaborative filtering with the only difference that a ranking index was created and jobs were sorted according to this ranking index [21].

Chien and Chen used data mining for effective personnel selection. They considered various features of the candidate as inputs: age, gender, marital status, degree, school/ school tier, major, work experience and the recruitment channel that can either be internal or external, then used this candidate's demographic data for predicting his/her work behavior. The work behavior comprised of job performance, retention and turnover reason. The significant rules were mined, having considerable lift and confidence, and then used to find out the job performance, turnover rate and retention rate of the different candidates of the organization [22].

Considering the other fields of the recommender systems, machine learning techniques are having noticeable effect on the provided recommendations. As in case of internet shopping mall, Cho et al used web usage mining, decision tree induction, association rule mining and product taxonomy to provide the recommendations. The procedure followed is as follows: firstly through the web usage mining, the system tried to predict the customer behavior. The product taxonomy is already assumed to be in place. After knowing the customer behavior, it is matched with those of the induction rules that are already stored. These rules help in judging the customer behavior as whether the respective customer is in the mood to buy that particular product or not. Along with these other factors like product taxonomy and environmental or the background information is also taken into consideration. And finally the strong product recommendations are made after considering all the above factors [23].

CHAPTER 3

PROBLEM STATEMENT

After considering the major work done in the related field of job recommender systems, let's have a look on the major gaps that exist in the present systems. When talking about the personalized job recommender systems, the task of recommendation becomes quite difficult as every candidate is unique. Everyone has some distinct circumstances, preferences, priorities, likes and dislikes that distinguish him/ her from others. Despite of that, there are some common facts that can be observed after considering the real world individual behaviors, belonging to a particular age group, gender etc.

Fact 1: *Decision Making*- With time, person's maturity level increases either through the various interactions, past experiences etc. that reflects in his/her decisions.

Explanation: A well known fact is: A person learns through his experiences, either good or bad. As one grows old, he/she attains a level of stability/maturity that is formed by his earlier interactions in this world. This knowledge is used in taking various decisions in his/her life, and when it comes to the carrier part these decisions play a very important role. This knowledge only forms the basis of the preferences formed by the candidate. For an example, the motive of a fresher in the IT world might be just to get a starting job with a moderate package irrespective of company and its location. Whereas the motive of a person, who is looking for a better job and having a 10 year experience in the IT field, can be either to have a higher package and position, compromising the status of the company or to have a better company compromising the package and position and many more.

Fact 2: *Group Behavior*- Same age level people (along with some other factors taken into consideration) show some common traits in decision making.

Explanation: Despite of the various differences existing amongst the decisions made by the people at different stages of their lives, there is something in common among them. For example, if we distinguish groups on the basis of gender a general observation is that the boys are comparatively more risk takers as compared to girls,

as girls often like to keep themselves at a safer side. Similarly distinguishing the groups on the basis of age, candidates near their twenties mostly do not consider location as a major factor while selecting a job whereas candidates near their forties, if willing to change a job might consider location as one of the major factors for job selection. Such common traits observed in the real world in particular group people, can be easily termed as group behavior.

Fact 3: Exceptions- Although there are group behaviors but exceptions exist everywhere.

Explanation: The carrier paths need not to be same for every person. There can be some who are behind their normal age groups in their carriers or it can be the case that some are more intelligent and may obtain big leaps in their carriers. For example, a person entering in the IT world at an age of 28 (near thirties) as a fresher might have preferences similar to the age group of early twenties. So, the carrier paths of such persons or exceptions might lag behind or lead ahead as compared to their normal group behaviors.

Fact 4: Varying preferences- 2 Candidates A and B may have similar looking profiles but can have different preferences/priorities.

Explanation: Despite of the common group behavior, candidate can still have different preferences or priorities. These preferences depend on various other contexts or factors, when taken into consideration. By context here it is meant by the extra factors like gender, education level, grades, current location etc. Location specification, company specification or salary specification can form some of the examples of this category.

Although there are various existing systems (as discussed in Chapter 2) that implement different strategies in order to make accurate personalized recommendations or predictions with respect to a particular candidate. Still they are not able to capture all the four above mentioned observed facts and thus somewhere lag behind in terms of system's recommendation accuracy.

Now, let's have a critical analysis on how the present systems lag behind in fulfilling the above facts and where the actual problem lies. The gaps present in the existing system are listed as follows:

Gap 1: Present systems that use content based filtering match the profiles only on the basis of content which violates the Fact 1.

Gap 2: Present job recommender systems that use memory based collaborative filtering, first shortlist the candidates that have similar looking profiles to the new user, then recommend jobs that these users have applied for and thus violate the Fact 3 and Fact 4.

Gap 3: The hybrid systems developed that use one or more of the recommendation techniques as discussed in chapter 2, either uses Fact 1 or Fact 2 as their basis but forget to capture the individual preferences of the candidates as described in Fact 3 and Fact 4.

While targeting a recommender system, the very first priority is to display only those recommendations on the top those are actually relevant for the respective candidate. In other terms this is also known as prediction accuracy. Prediction accuracy is one of the parameter that is used to judge the recommendations made by the system. The more useful personalized recommendations are made with respect to a candidate, the more successful the system is considered. The relevancy of the recommendations made by the system can be easily judged by the various direct and indirect existing methods like customer direct feedback or number of user clicks, time span of each click etc.

So, now the problem statement can be formulated as follows:

Problem Statement: Given a candidate c and job set $j \{ j_1, j_2, j_3, j_4, j_5, j_6 \dots j_n \}$, recommend jobs for c , $c \times j \rightarrow r$ where $r: \{ j_2, j_4, j_5 \dots j_m \}$ such that r satisfy the four parameters: *decision making*, *group behavior*, *exceptions* and *varying preferences* and fix in most of the gaps that are present in the existing systems and also improve or enhance the *prediction accuracy*. By satisfying the four parameters, it is meant that the recommendations made to the candidate must take care of the group as well as individual behavior or preferences regarding the jobs, as well as must take care of the exceptional carrier path candidates too. And thus must contribute for increasing the prediction accuracy rate or likelihood of the recommendations made to the respective candidate.

CHAPTER 4

PROPOSED ALGORITHM

A job recommender system is expected to provide recommendations in 2 ways:

- 1) Recommending most eligible candidates for the specified job, to the recruiters
- 2) Recommending jobs to the aspiring candidates according to their matching profiles

The focus here is the second part only i.e. to recommend jobs to the candidates according to their matching profiles. Here the efforts have been put to judge the gap between the candidate's choices, belonging to different groups, regarding the selection of different offered jobs. It is tried to foresee the customer preferences regarding the jobs on four basic parameters of company preferences, position offered, pay-scale offered and job location. Instead of tracking the past history of the candidate, his current likings/preferences are focused upon. For a new customer, the system tries to impose the general job preferences, obtained through mined rules, according to the age-group, gender etc. to which the candidate belongs and as the candidate becomes active, his/her own job preferences are taken into consideration.

The basic steps involved in making the final recommendations are as follows:

- Data Acquisition
- Feature Selection
- Data Categorization
- Mining of Decision Tree Induction Rules
- Phase-I Recommendation Generation
- Phase-II Recommendation Generation

4.1 Basic Terms Used

There are some basic terms that are used in this thesis and need some explanation. So the basic terms used are listed as follows:

- **Age:** Age refers to the present age of the candidate.
- **Gender:** It refers to the gender of the candidate. It can be either Male or Female.

- **Marital Status:** It represents whether the candidate is married or unmarried or divorcee.
- **Education:** It represents the educational degrees obtained by the candidate. It can be 12th, Bachelors, Post Graduate, Doctorate or higher.
- **Grades:** Marks scored by the candidate in his last/latest acquired degree.
- **Major:** Specialization field or discipline of the candidate.
- **Experience:** Candidate's experience in years.
- **Skills:** Extra knowledge/skills possessed by the candidate.
- **Current Location:** If the candidate is currently employed than this field specifies the current job location of the candidate and if unemployed than this field usually represents the home location of the employee.
- **Current employer:** If the candidate is employed than it represents the current company the candidate is working for.
- **Current pay-scale:** If the candidate has a job than it represents the current pay-scale of the employee.
- **Current position:** It represents the current post/ position that the employer is working at, in his current job.
- **Employer/Company:** Company, that is offering the job.
- **Industry field:** major/discipline in which the job is offered.
- **Position Offered:** position offered for the job.
- **Pay Scale:** Salary offered for the job.
- **Job Location:** This field specifies the location that the candidate will get after joining the prescribed job. Thus it will refer to the target job location.
- **Cosine similarity:** A measure of similarity between 2 vectors (Here it refers to job and candidate).
- **Preference matrices:** These matrices represent the candidate's preferences on 4 different parameters: company selection, position selection, pay-scale selection and job location selection.
- **Rules Weight:** This refers to the weights assigned to the jobs after applying the group/candidate's own preferences for recommending jobs to the candidate.
- **Final Weight:** This represents the weighted score of a particular job, obtained after combining weighted sum of cosine similarity and rules weight. This only forms the basis of ranking of the jobs at last stage of job recommendations.

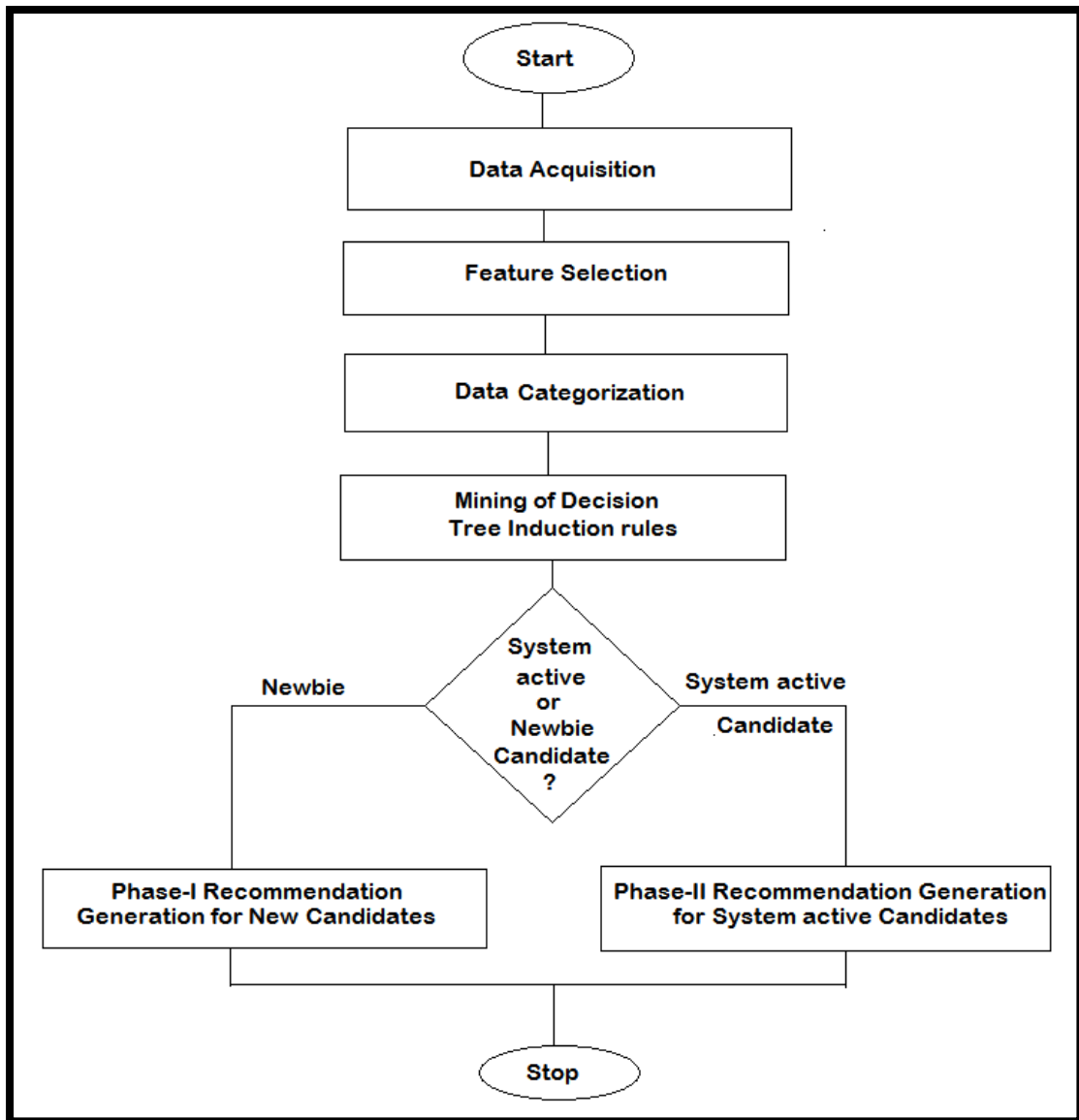


Figure 4.1: Normal procedure proposed in making the job recommendations

4.2 Assumptions

- It is assumed that all the text categorization is already in place and we have discrete and well labeled values that are easily understandable by the system. As resume mining is one of the research area in itself and just to focus on the proposed problem, it is neglected and that is why the text categorization is already assumed to be in place. That is interface is prepared which includes independent fields that keep the features distinct and ordered.
- It is assumed that either the candidate or the system itself updates the age, experience, education and skills field. For wanting the candidate to update the field, the candidate needs to be alerted periodically for updation of these fields. If

system is chosen for automatic updation, then it can be easily done with the help of Dynamic modification and extraction method used in [16]. It uses TF-IDF value for feature extraction and information gain with threshold value for feature addition.

Here updation of field is really important as the focus of the current system is concentrate on the present candidate status and preferences. And if the fields are not updated than it may result in completely vague results. Hence updation of fields timely plays an important role in this implementation.

4.3 Basic steps of the proposed algorithm

4.3.1 Data Acquisition

The very first step in the procedure is to acquire data regarding the jobs and candidates from various sources.

- For this purpose the data regarding the candidate was collected from the internet via the available resumes. The candidate's preferences all were collected and considered by tracking his past transition history as well as present employment details.
- The data regarding the offered job was collected from the various recruitment sites.

4.3.2 Feature selection

The second step is the selection of features on the basis of which the candidates profile will be matched with that of the job profile. So the features that were found relevant in a job scenario belonged to 2 major categories:

- Candidate
- Job

For candidate the features that were considered for judging his/ her behavior are: Age, Gender, Marital Status, Education, Grade, Major, Experience, Skills, Current Location, and Current employment status (if any). All the above features are already discussed in section 4.1. These features collectively form the candidate profile in the proposed system. These features relevant to the candidate are summed up and shown in Figure 4.2.

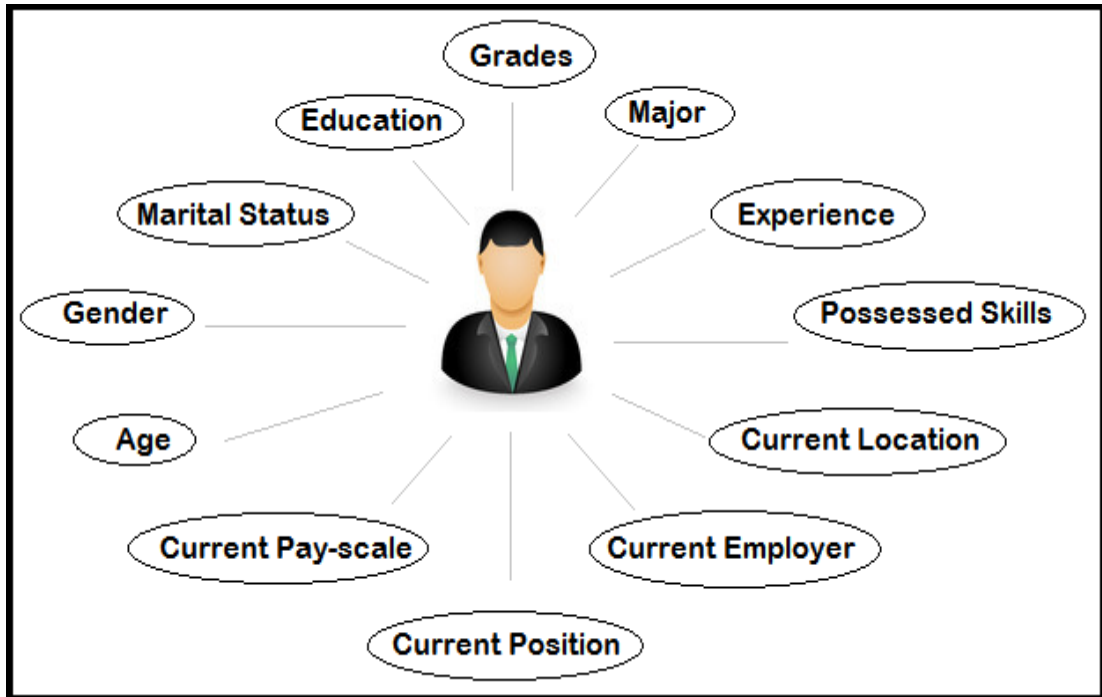


Figure 4.2: Relevant features related to the candidate

The features relating to the job are: Required Qualification and Experience, Skills requirement, Employer or the Company, Industry field, Position Offered, Pay Scale and Location. These above features collectively define a job in the proposed system. The features relevant to the job are collectively represented in Figure 4.3.

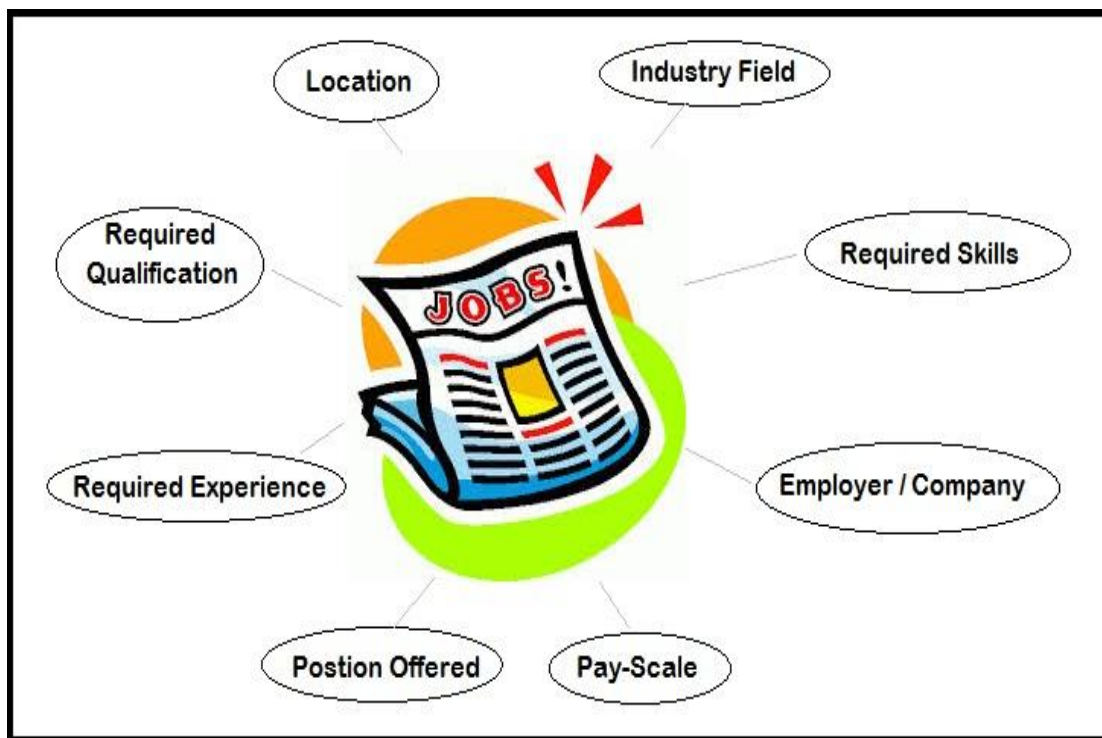


Figure 4.3: Relevant features related to the job

4.3.3 Data Categorization

As the objective was to find out the criteria on which the candidate, belonging to different age group, marital status, gender, education level, grades etc, focuses for selecting the offered job, the complete categorization or generalization was done. The candidate data as well as company data both are categorized into different groups for finding out the candidate's behavior belonging to a particular group for selecting a particular job on the basis of 4 parameters: Company group level, position offered, pay-scale offered and job location. The following list explains how the categorization was done of the above selected features for studying the candidate behavior.

- **Candidate Data**

The above mentioned selected candidate's features were categorized and later on used for mining relevant rules for different group's composition.

Table 4.1: Categorization of candidate features

S. No.	Feature	Categorization/ Generalization	
1.	Age	6 major groups.	Group 1: 20-25
			Group 2: 26-30
			Group 3: 31-35
			Group 4: 36-40
			Group 5: 41-45
			Group 6: <45.
2.	Gender	2 groups	Group 1: M
			Group 2: F
3.	Marital Status	3 groups	Group 1: Married (M)
			Group 2: Unmarried(U)
			Group 3: Divorcee(D)
4.	Education	3 groups	Group 1: Graduation (B)

			Group 2: Masters (M)
			Group 3: Doctorate or above (D)
5.	Grade	3 groups	Group 1: >80 (High or H)
			Group 2: 55-80 (Average or A)
			Group 3: <55 (Low or L)
6.	Major	This was not considered for grouping. This field relates to candidate's validation for a particular job.	
7.	Experience	6 groups	Group 1: 0 (N)
			Group 2: 1-5
			Group 3: 6-10
			Group 4: 11-15
			Group 5: 15-20
			Group 6: >20.
8.	Skills	This was not considered for grouping. This field instead helps in judging the similarity status between the candidate and job. So, only the keywords are considered. Here it is considered that skills are already in place and can be easily represented as vector space model.	
9.	Location	4 groups	Group 1: North India (N)
			Group 2: South India (S)
			Group 3: East India (E)
			Group 4: West India (W)

Example 4.1: A candidate having the following details: {27, male, unmarried, graduate, 65%, 2 years experience, New Delhi} can be easily grouped as follows {26-30, M, U, B, A, 1-5, N}.

- **Company Data**

The above mentioned selected job's features were also categorized and later on used for finding similarity score with that of the candidate possessed skills.

Table 4.2: Categorization of job features

S. No.	Feature	Categorization/ Generalization	
1.	Employer or the Company	4 types of company according to the company ranking	Group A (A) (top 25 %),
			Group B (B)
			Group C (C)
			Group D (lowest ranked).
2.	Industry field	Not considered for grouping. Only relates to the major or subject.	
3.	Position Offered	4 groups according to their level of importance	Level A (Highest Top positions)
			Level B
			Level C
			Level D (Initial Level)
4.	Pay Scale	3 Groups	High (H): (>15 L/ annum)
			Medium (M): (6-15 L/annum)
			Low (L): (<=5 L/ annum)
5.	Location	4 groups	Group 1: North India (N)
			Group 2: South India (S)
			Group 3: East India (E)
			Group 4: West India (W)

Example 4.2: A company having the following details: {TCS, Assistant Software Engineer, 4 L/per annum, New Delhi} can be easily grouped as follows: {Group A, D, L, N}.

Specification to generalization for the candidate and job are done with the help of Table 4.1 and 4.2 and are shown in Figure 4.4 and 4.5 respectively. This categorization, although comprised of some trade-offs (regarding different companies different level of designations, their pay-scales etc.), helped in mining of generalized rules for the candidate.



Figure 4.4: Specification to generalization for a candidate

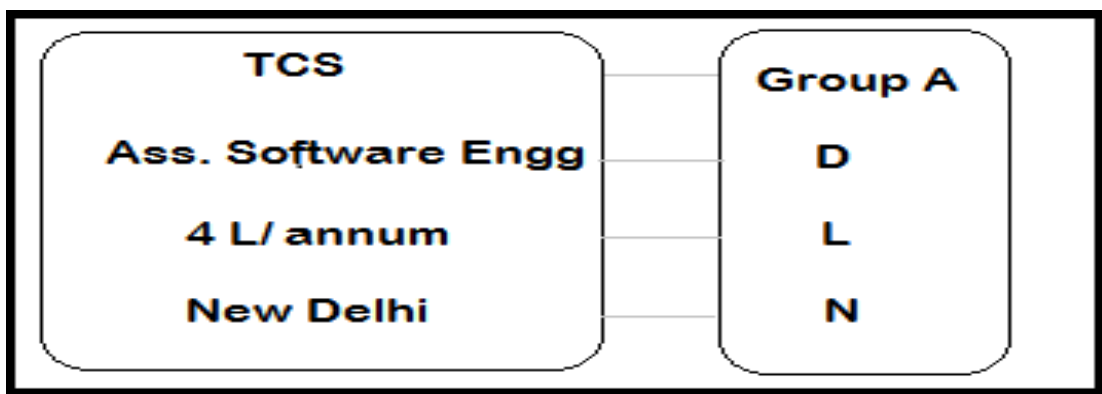


Figure 4.5: Specification to generalization for a job

- **Domain Knowledge**

Extra efforts have to be put in for collecting the domain knowledge regarding the job related fields as well as skills field in the candidate's features list.

1. For the job related fields, domain knowledge regarding the present companies rating need to be collected, linked and categorized correctly, for better results. Same is the case with position. A generalized idea of position levels is stored and categorized for mining efficient rules.
2. Skills field also need generalization. Let's understand it with the help of an example. A job may be requiring full knowledge of Java language. As Java is Object oriented language, so if a candidate is having even generalized knowledge of OOPs then he can be considered for this post as opposed to a candidate with little or no knowledge of OOPs. So, here it is assumed, as in [13], that even if a candidate is well versed in C++, he is also an equal contender for the offered job, as he is indirectly, but related to the concepts of OOPs. So, the hierarchical information regarding the domain needs to be stored. Here the domain knowledge

is being stored, for better matching of the skills field. As the fields are stored in hierarchical order, a child here even can have more than one parent depending upon whether the child belongs to multiple domains/ groups.

Example 4.3: Object Oriented Programming Languages include C++, Java, Python, and Smalltalk. So a hierarchical ordering needs to be stored. And even one of the matches is found amongst these than the candidate is considered to have qualified in this field and given complete score for that particular vector field in its system representation.

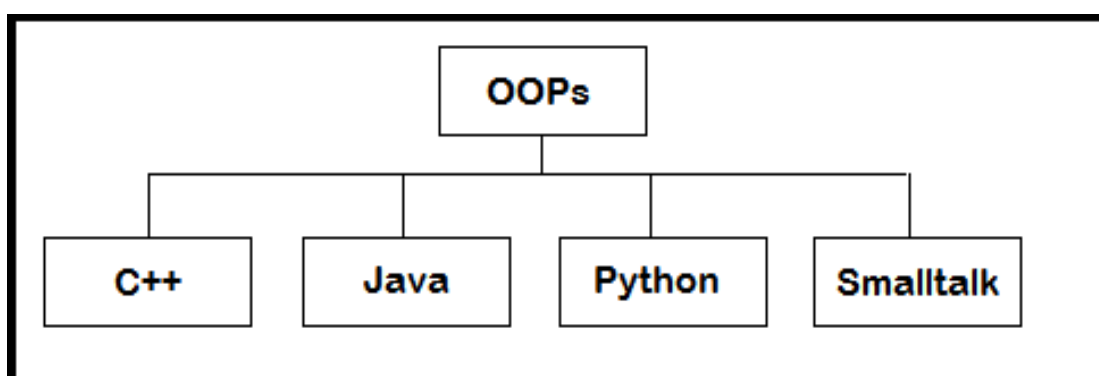


Figure 4.6: Hierarchical representation of OOPs languages

4.3.4 Mining of Decision Tree Induction Rules

Data Mining is one of the effective techniques for web personalization [24]. Here, decision trees have been for rule mining as per problem domain and suitability.

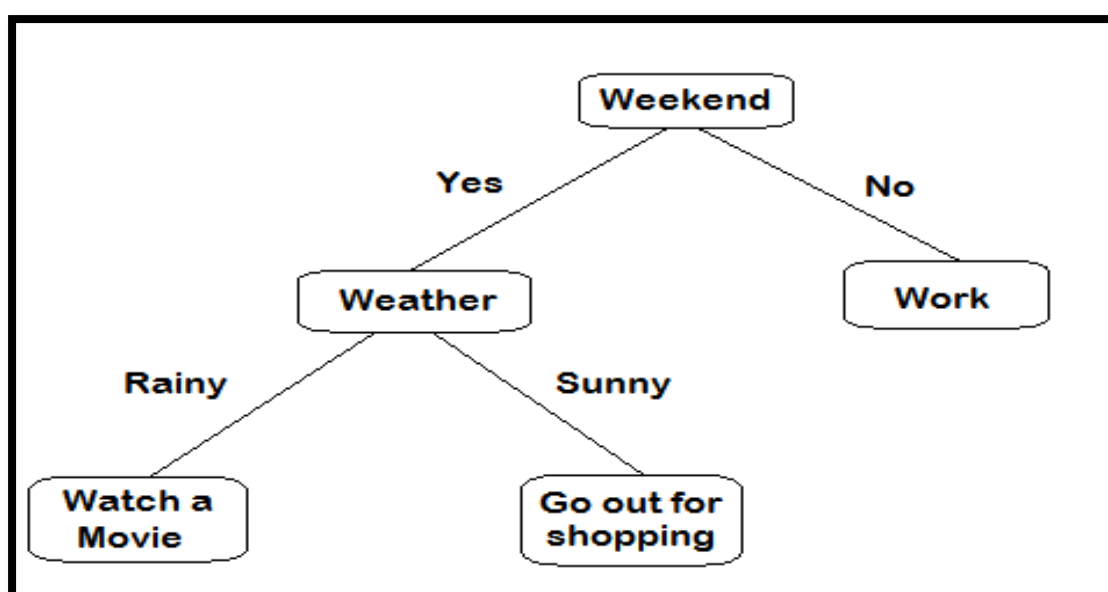


Figure 4.7: A decision tree example

Decision trees are tree like graphs or models that represent every possible outcome leading to a particular decision. In these each internal node represents a condition, every edge coming out represents the choice and every leaf node represents the classification or the decision. The path from root node to the leaf represents the decision rule in the form of if-then. These trees are used in data mining for supervised learning. These are used in classification problems. They are favorable, if the data that is to be tested is categorical.

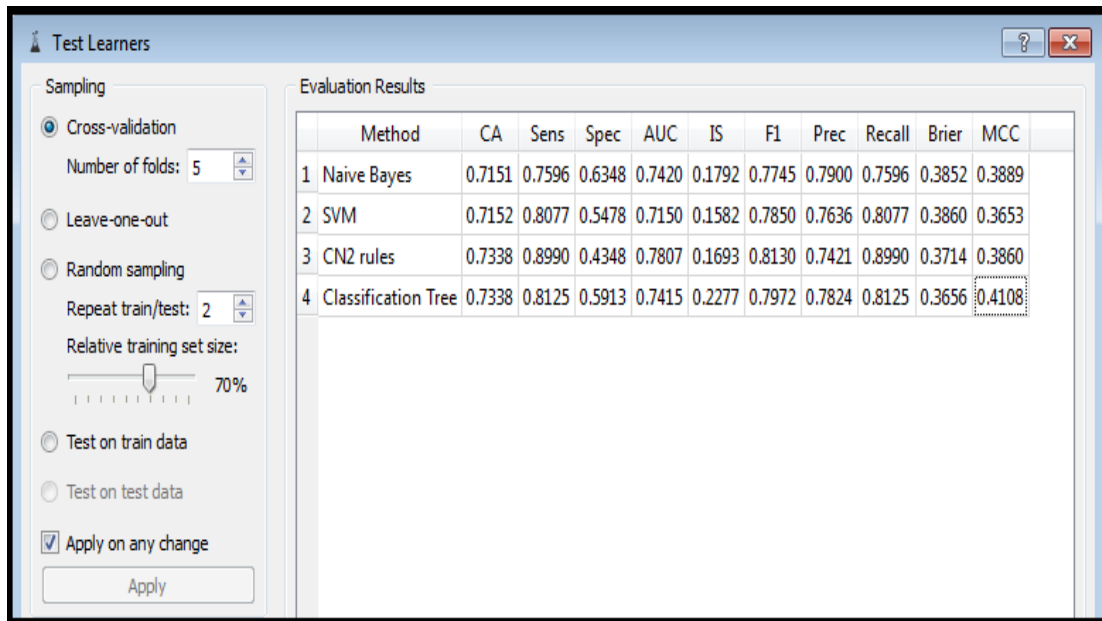


Figure 4.8: Evaluation results of various learning algorithms

Although the sample data was tested on various supervised machine learning algorithms, however Classification trees only outnumbered or performed better in most of the performance parameters as against: SVM, Naïve Bayes, CN2 Rules as shown in the Figure 4.8.

While using Classification or Decision trees, one of the important factors is: selecting the criteria for attribute splitting. Here C4.5 algorithm is used as it uses normalized information gain for attribute selection and splitting (Figure 4.9). It recursively search for such attributes, that divide the subspaces into highly enriched: one class or the other.

For the candidates, generalization/ categorization was done on the basis of various features as discussed earlier. For the jobs, they were divided into total 20 different meaningful categories, each having unique combination/ characteristic in terms of

company groups, position level, pay-scale and location. The data was collected and analyzed, for each job category, separately.

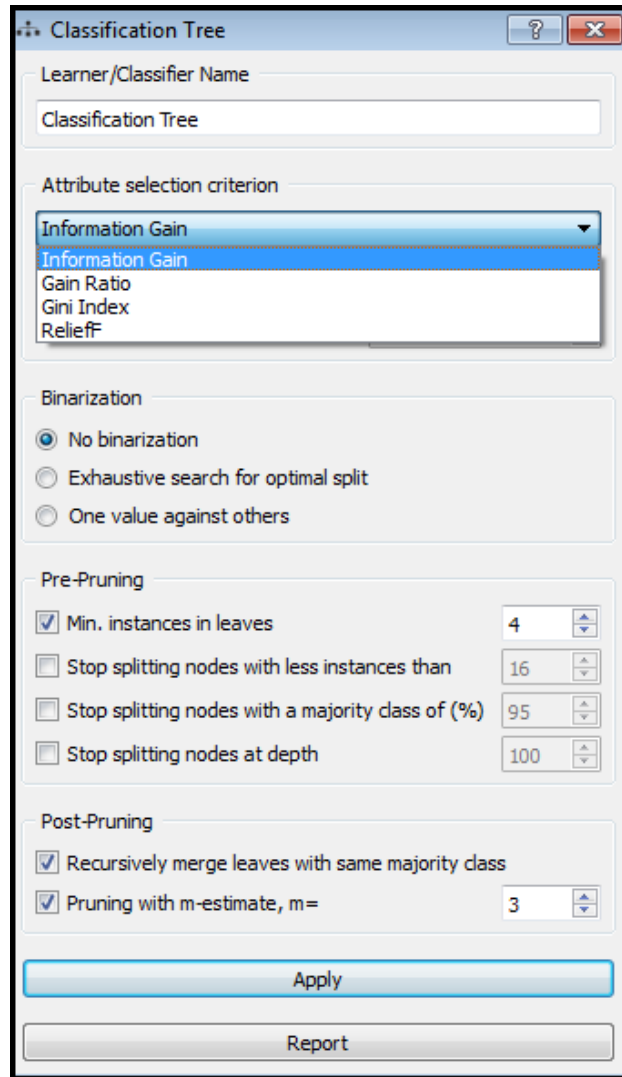


Figure 4.9: Classification tree attribute selection criteria

Exhaustive search was made for all the rules that can determine the choice of candidates job taste, belonging to a particular group, for the corresponding job category. The evaluation criterion taken for determining the strength of these rules are the lift, confidence and the sample size [10].

$$\text{Lift A (Rule i)} = \frac{P(\text{target class A} | \text{subset i})}{P(\text{target class A} | \text{population})} \quad (1)$$

$$\text{Confidence A (Rule i)} = P(\text{class A} | \text{subset data by Rule i}) \quad (2)$$

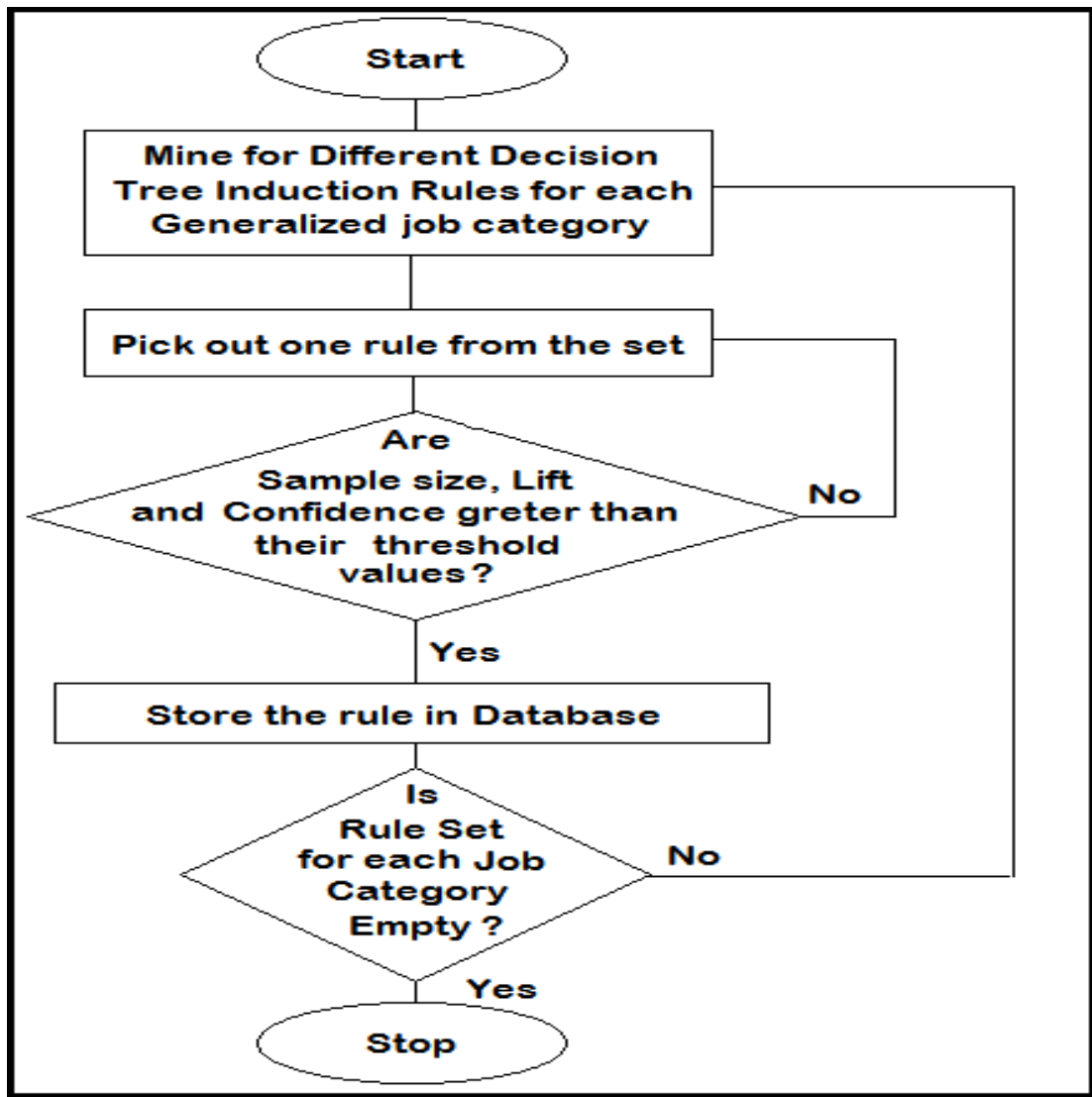


Figure 4.10: Flow chart for calculation of decision tree induction rules

Here, the lift value was considered to be greater than 1 and confidence percentage to be greater than 80%. A threshold value for the sample size was taken. If there exists a rule which has high lift and confidence but its sample size is less than the threshold value, then it is not taken into consideration, however strong the rule may be. All the rules that crossed the selection criteria were enlisted and checked for redundancy.

After that, a common matrix representing all the job categories preferences, for a particular rule was made. In this matrix, corresponding to a particular rule, if the rule exists for a particular job category, then the corresponding field of the job category is made 1 else 0.

$$\text{Matrix } [i, j] = \begin{cases} 1 & \text{(if } i^{\text{th}} \text{ rule exists for } j^{\text{th}} \text{ job category)} \\ 0 & \text{(else 0)} \end{cases} \quad (3)$$

After this 4 preference matrices were generated, each one preserving the authenticity of generated rules. And the scores/ weights assigned were normalized by dividing them with the total number of instances available for the jobs i.e. respective probabilities of selecting a particular field are stored.

Example 4.4: Preference matrix for company stores information regarding its 4 type of company's preference. 1st position for Group A companies preference, 2nd for Group B, 3rd for Group C and 4th for Group D as: $[p_A \ p_B \ p_C \ p_D]$.

The candidate's generalized preferences are now captured in the form of 4 preference matrices and thus can be applied when and wherever needed for judging and short listing the jobs for the respective candidate. These 4 matrices here represent the generalized group preferences as they are generated after mining the generalized group behavior.

4.3.5 Phase-I Recommendation Generation

The steps involved in the generation of recommendations when a candidate is new to the system are:

- Shortlist the jobs for which the candidate is currently eligible for
- Calculate the Content Based Similarity
- Apply the Decision Tree Induction Rules for the category to which the candidate belongs
- Generate the final weights
- Sort the jobs in descending order

Step 1: Shortlist the jobs for which the candidate is currently eligible for:

The fields considered for short listing are: major, minimum qualification required and minimum experience needed for the job. This is done to reduce the processing time over irrelevant jobs for the particular candidate. Because considering all the jobs for content based filtering, results in higher time complexity in case of large datasets.

Step 2: Calculate the Content Based Similarity:

Now calculate the similarity index for the short listed jobs with respect to the candidate. The similarity index is calculated in between the jobs desired skills field and candidate's possessed skill's fields. Here cosine similarity between these two is

considered. Cosine similarity is between 2 vectors a and b is calculated by the following formula:

$$\text{Cosine similarity}(a, b) = (a \cdot b) / (|a| |b|) \quad (4)$$

Firstly preferred matrix vector for job's skills requirement is created and then accordingly candidate's vector is created [11].

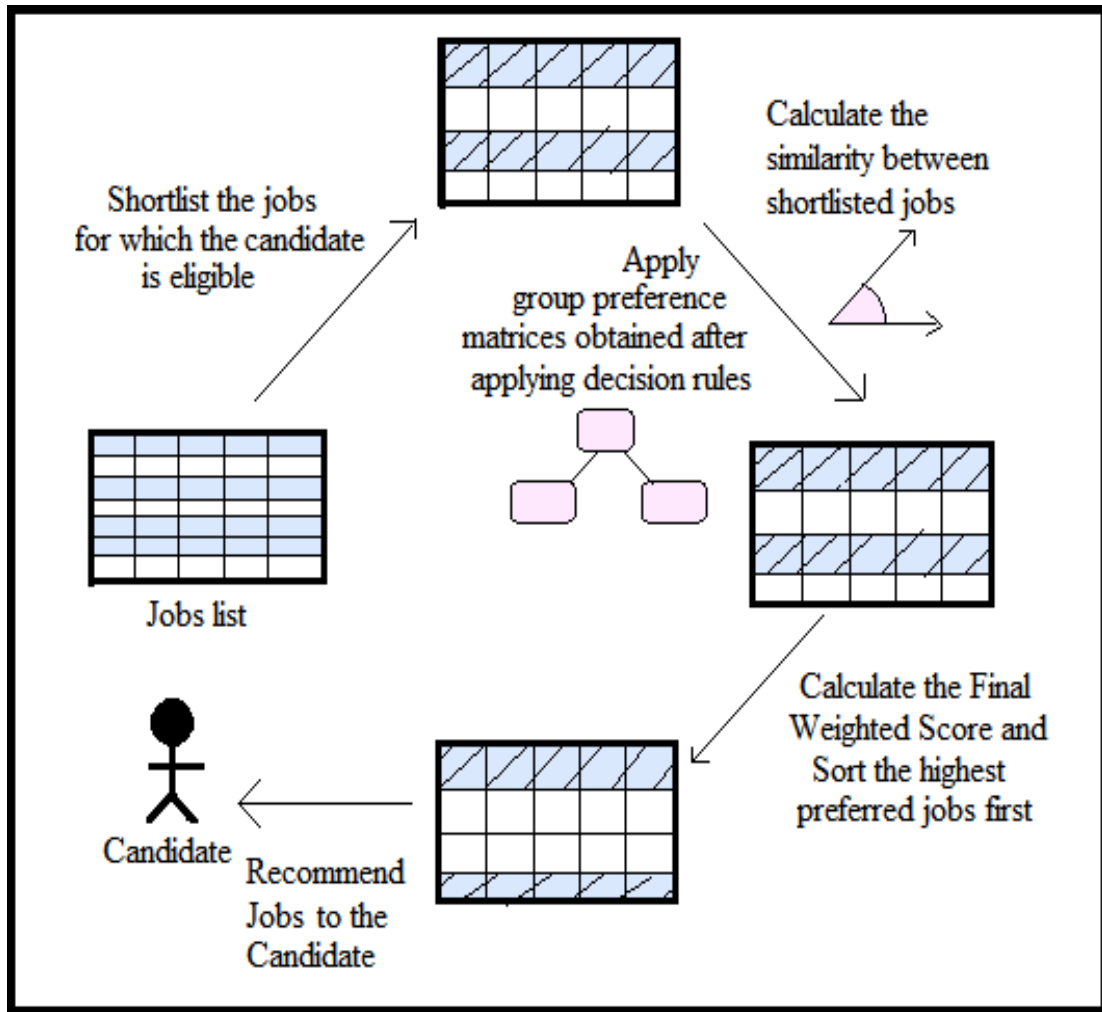


Figure 4.11: Steps for phase-I recommendation generation

However, if a level of proficiency is required in a particular language, then it can be represented as the corresponding weight in the vector. Example if the requirement is to have a proficiency level of 3 on a 5 scale, in java language then that can be represented with the weight 3 in the corresponding java field representation in the job vector. Also, if 2 languages/ skills match exactly or are related hierarchically and belong to same domain, then that is considered as an exact match and weight 1 is assigned else a 0 is assigned [12].

Step 3: Apply the Decision Tree Induction Rules for the category to which the candidate belongs:

Here, the basic categorization of jobs is done firstly. After that these categories are matched according to the preference matrices of the generated rules and assigned preference weights accordingly. Now, assign proper weights to the corresponding jobs, for building up accuracy in recommendations, by judging the general behavior of the respective group candidates. Normalize the rule's weight with the help of following equation:

$$\text{Normalized Weight} = (W_i - W_{\min}) / (W_{\max} - W_{\min}) \quad (5)$$

where W_i is the rule's weight assigned to the corresponding i^{th} job and $W_{\min}$ and $W_{\max}$ represent the min and max weights considering all the job's weight altogether.

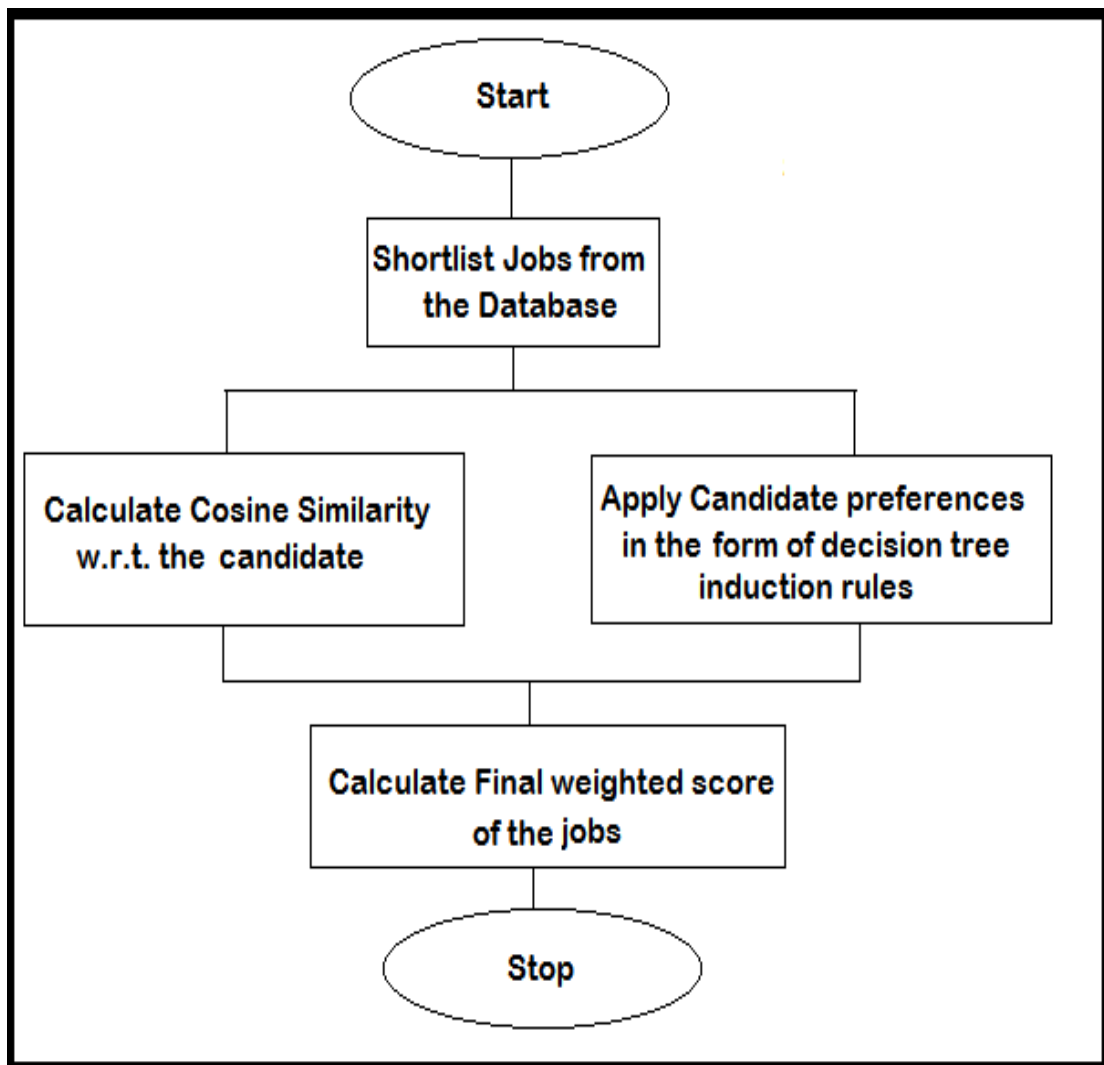


Figure 4.12: Flow chart for final weight score calculation in phase-I recommendations

Step 4: Generate the final weights:

Now calculate the final weight score by summing up values of cosine similarity and rules weight according to the following equation:

$$\text{Final Weight (i)} = w_1 \cdot \text{sim}_i + w_2 \cdot \text{rw}_i \quad (6)$$

where i represents i^{th} shortlisted job, sim_i stands for cosine similarity for i^{th} job, rw_i represent rules weight assigned to the i^{th} job, w_1 and w_2 represent the weights assigned to cosine similarity and rules weight for preserving their relevance. Here w_1 and w_2 both have the value as .5.

Here both cosine similarity as well as rules weight is equally weighted as assigning improper weights to them leads to vague results because the higher weight one overpowers the priority of other and thus induce vagueness in results.

Step 5: Sort the jobs in descending order:

According to the final score obtained, sort the jobs in descending order i.e. the job with maximum final weighted score at position 1 and job with least final weighted score at last.

4.3.6 Phase-II Recommendation Generation

This phase starts when the candidate has applied for at least 10 jobs. These jobs further fulfill the concept of mining new information, for recommending new jobs to the respective candidate, according to his/her job taste. The minimum jobs for rules creation is 10 and maximum jobs considered at a time were Θ where $30 < \Theta < 40$ as after Θ no accurate generalized results were obtained.

Again, the steps involved in phase 2 are:

- Shortlist the jobs for which the candidate is currently eligible for
- Calculate the Content Based Similarity
- Direct preference matrix creation according to the customer's latest preferences for jobs
- Generate the final weights
- Sort the jobs in descending order

This second phase recommender system is different from the phase one at only 3rd step. As in 3rd step there we apply the generalized mined rules for recommending

jobs to a candidate who is new to the system, whereas here direct customer preferences are used, by looking over its past applied jobs. Although the threshold of 10 is taken as less than 10 jobs were not qualifying for judging the exact candidate behavior. The general overview of the Phase-II Recommendation Generation is shown in Figure 4.13.

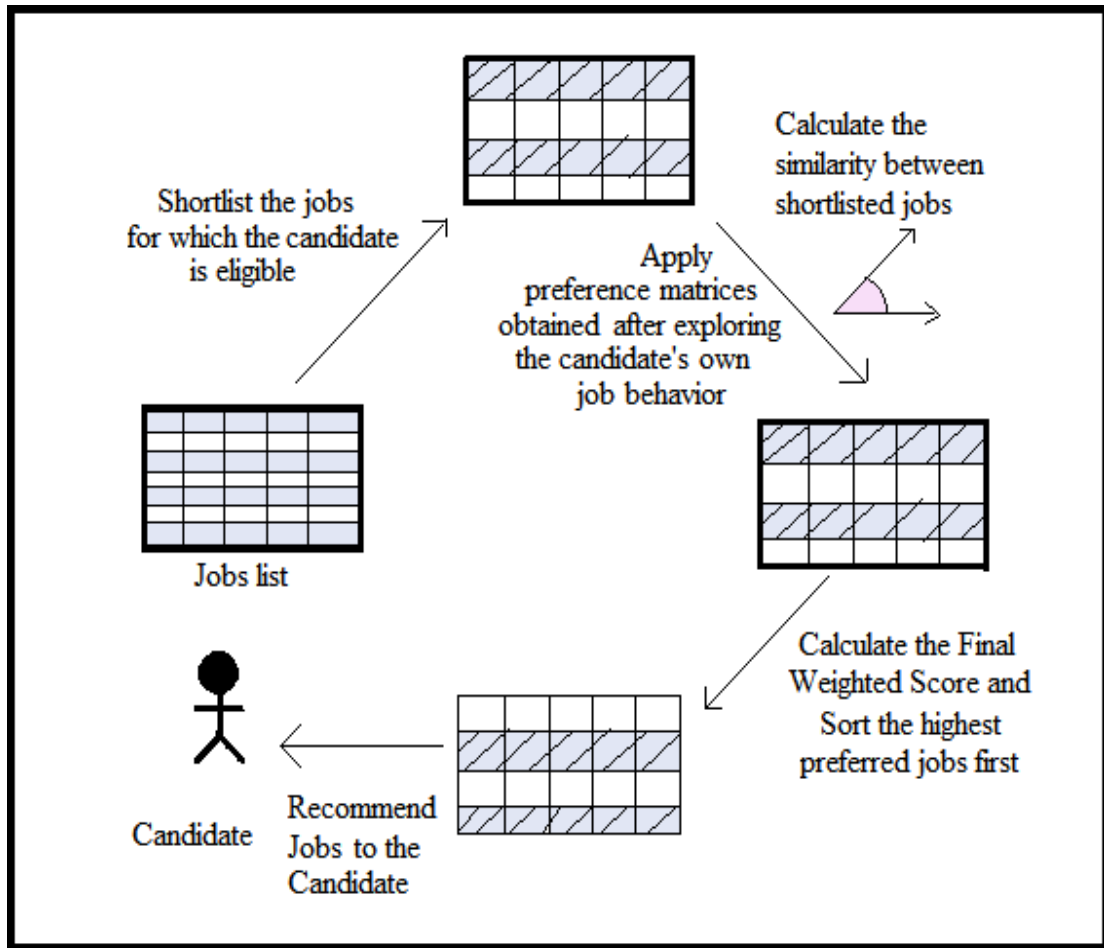


Figure 4.13: Steps for phase-II recommendation generation

Step 1: Shortlist the jobs for which the candidate is currently eligible for:

This step is same as earlier. Here also first the short-listing of jobs is done firstly, for working efficiently on the small jobs set, for which the candidate is currently eligible for.

Step 2: Calculate the Content Based Similarity:

Calculate the similarity index between the jobs required skills and candidate possessed skills using equation (4). So according to the job vector similarity, cosine similarity index is created for each job.

Step 3: Direct preference matrix creation according to the customer's latest preferences for jobs:

This step is different from the first phase's step third. Here preference matrices are created by directly judging candidate criteria behind selecting a job. The probabilities are calculated directly by keeping track of the company group, level of position offered, pay-scale applied for and the location of the job.

Again assign proper weights to the corresponding jobs, for building up accuracy in recommendations, by judging the personalized behavior of the candidate. After that, normalize the rule's weights by using the normalization equation (5).

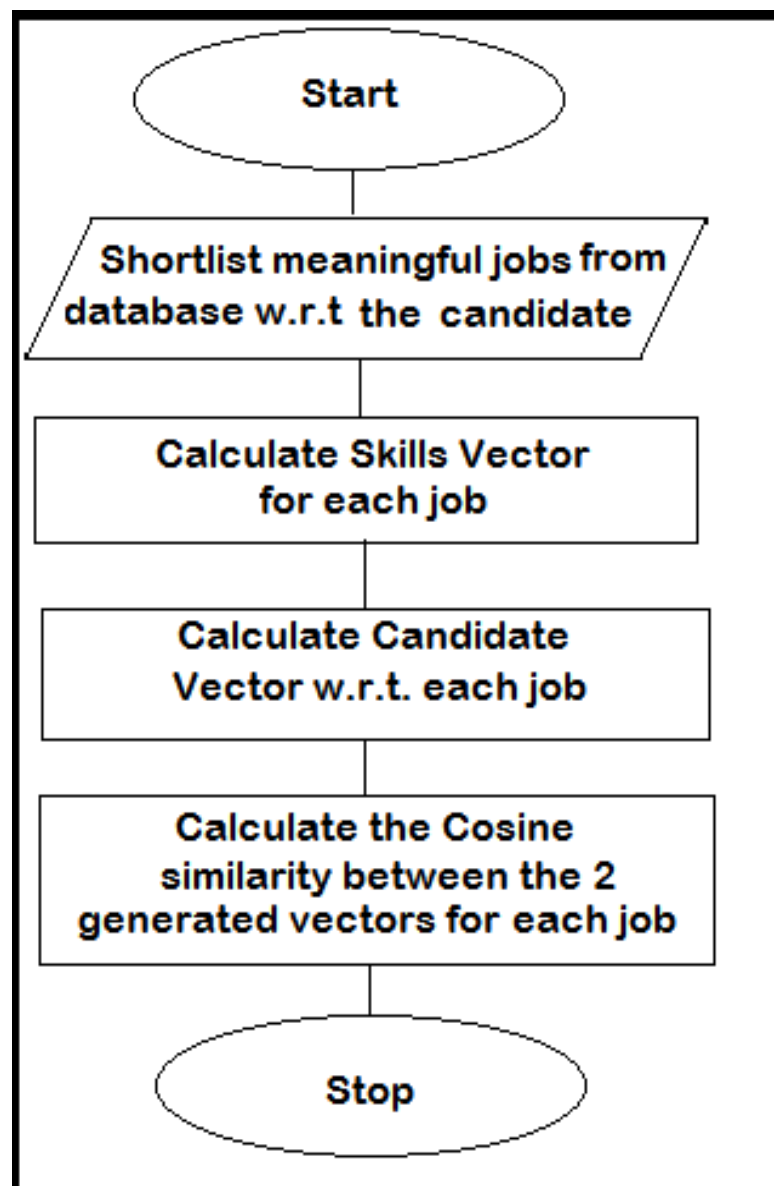


Figure 4.14: Flow chart for calculation of cosine based similarity

Step 4: Generate the final weights:

As in earlier phase, calculate the final weight score using equation (6).

Step 5: Sort the jobs in descending order:

Sort the jobs in descending order according to the final score and recommend them to the customer.

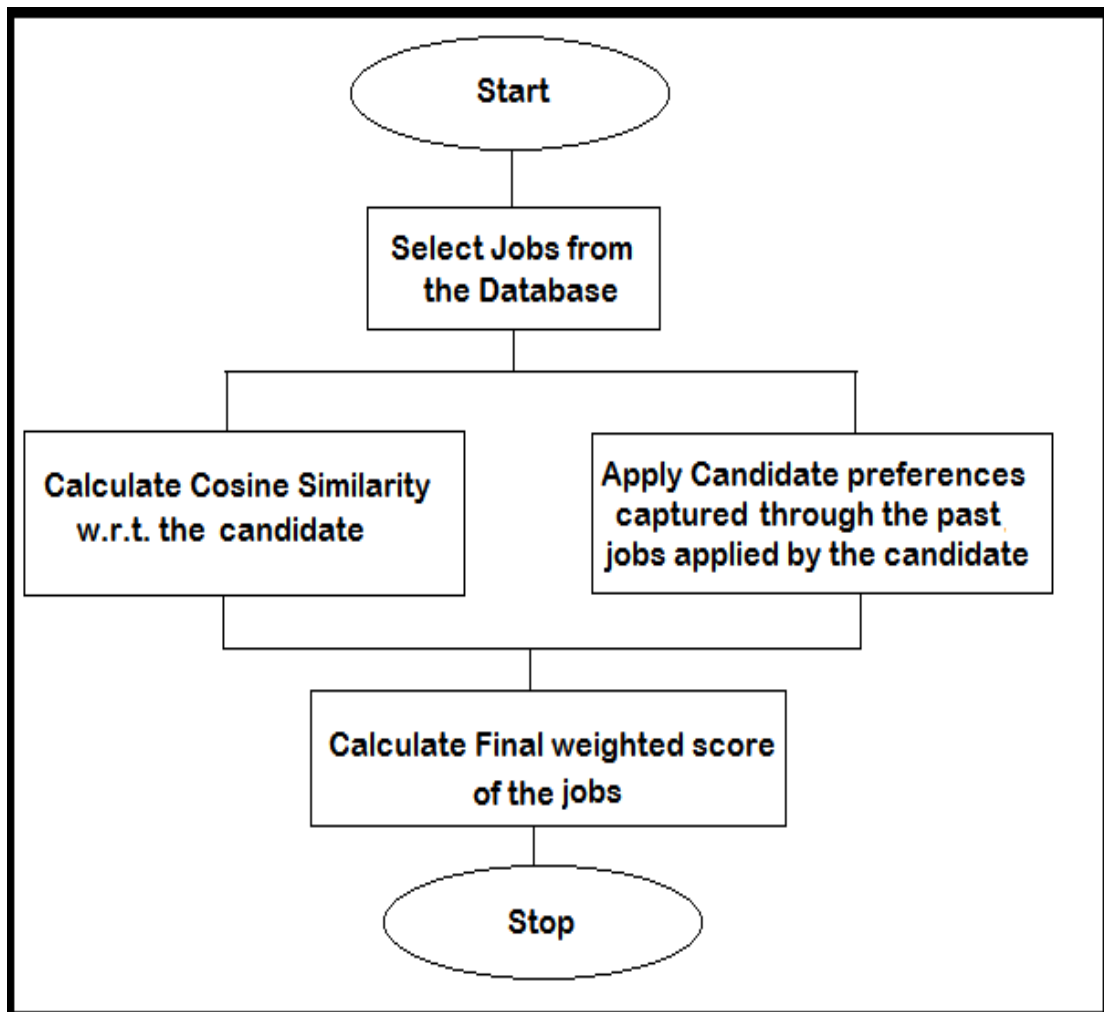


Figure 4.15: Flow chart for final weight score calculation in phase-II recommendations

Once the Phase-II is started for an active candidate, it goes forever and re-iterate over the last system allowable maximum jobs that the candidate has applied for and thus helps in recommending efficient personalized jobs to each of the candidate having different job preferences.

CHAPTER 5

IMPLEMENTATION

5.1 Experimental Setup

5.1.1 Datasets

As the research work required job related resumes of the candidates to be in place, a lot of resumes were surfed over the internet. The research oriented relevant information regarding the candidate was studied and extracted. The information related to their past history, transitions from one company to another, all were considered. As the resume mining was not the target of the research, so all the candidate's information data was kept in place in the database. Also the jobs related information was also extracted from the various recruitment sites available online [32]. It was also kept directly in the database.

So, the actual research dataset consisted of 1500 candidates and 500 jobs. 20 different meaningful categories of jobs were made according to the research categorization requirements and rules were mined accordingly.

5.1.2 Environment

Python was used as the main implementation language. As it is one of the most famous scripting languages, it also provides the libraries for making the recommendations, data mining, artificial intelligence and many more. It supports most of the machine learning operations with the help of its appropriate libraries. So, it can be said that it is one of the most evolving languages. So, considering the above factors, python was only used as the implementation language here for the following two purposes:

- Making the recommendations
- Data mining purpose

The Orange library of python was used for data mining purpose. The relevant rules were mined against each job category and stored as knowledge base for creating further recommendations [26].

5.2 Data Mining Using Orange

Orange the data mining tool, also available in the form of library in python, makes the task of data mining much simpler. It provides the ease of use to the developers, researches through its effective graphical user interface. As it is completely a component centric tool, it provides widgets in its interface. These widgets only form the basis of its component based interface. They take the input and process them internally and finally provide the output. They also have communicating interfaces, such that with the help of these only two or more widgets interact [25-28].

Figure 5.1 shows the experimental setup used during the data mining step of the proposed algorithm.

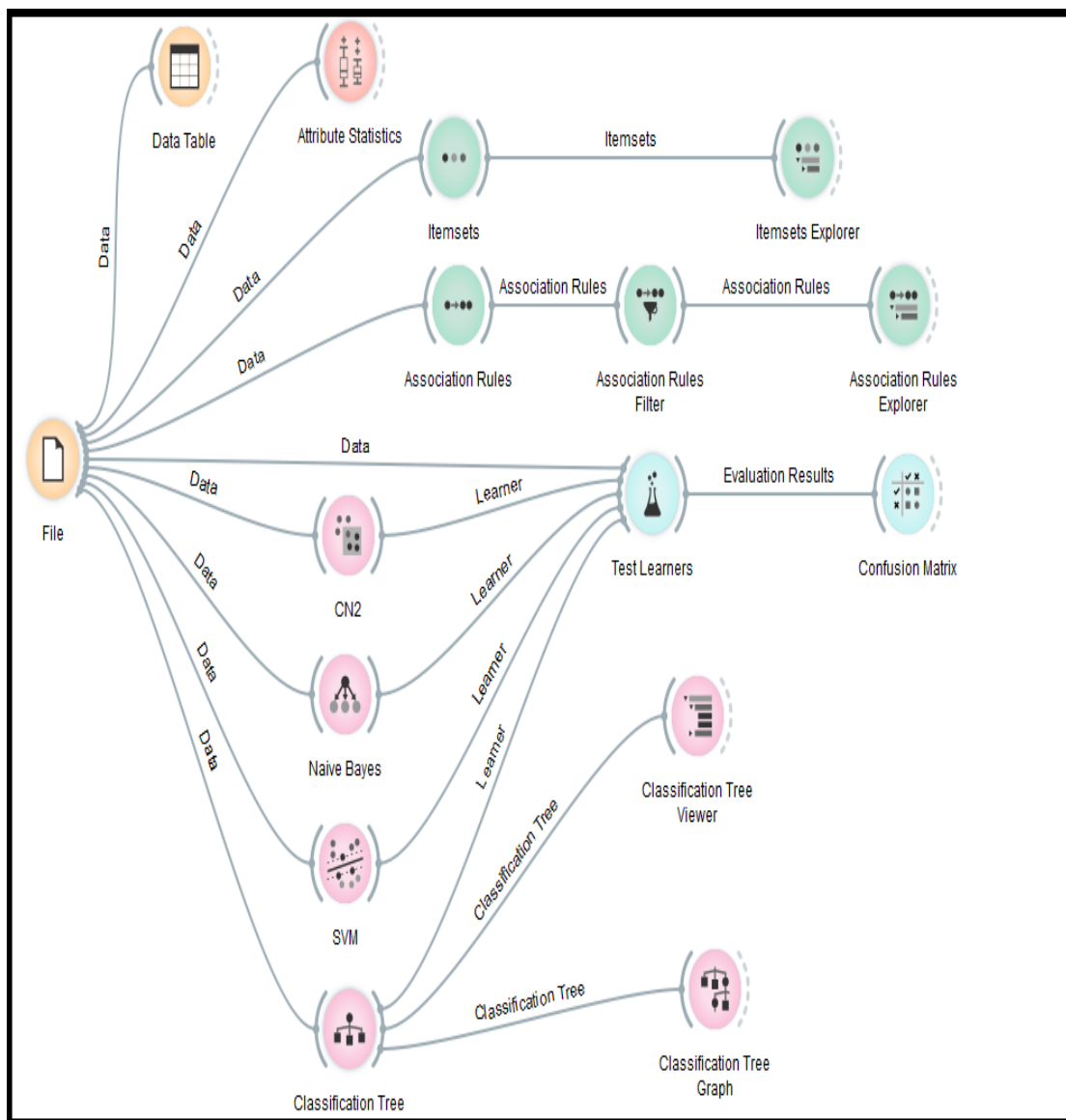


Figure 5.1: Experimental setup in Orange tool

It consists of the following important concepts (Figure 5.1):

- First of all, the target sample file is uploaded.
- The view of it can be taken in the Data Table Widget.
- Attribute statistics can also be viewed by its respective widget.
- As it is the supervised learning case, hence various classifiers are used.
- Then these classifiers are tested in the Test Learner widget.
- The Classification Tree classifier is the best one chosen for the data mining purpose after their evaluation on different parameters.
- After that the association rules were mined and stored in the database for later use in the recommendations (Figure 5.2).
- Then a common matrix representing all the job categories preferences, for a particular rule was made, as discussed earlier.
- After that, 4 job preference matrices were created.

Rules	Conf	Lift
Grades=h		
Grades=h -> CompanyAD=y	0.828	1.285
Exp=n		
Grades=h Exp=n -> CompanyAD=y	0.828	1.285
Age=a1		
Grades=h Exp=n Age=a1 -> CompanyAD=y	0.828	1.285
Age=a1		
Gender=m		
MStatus=u		
Exp=n		
Age=a1		

Figure 5.2: Association rules example showing confidence and lift

Example 5.1: In Table 5.1, for representation there are 5 rules and 12 job categories. 1st rule indicates that for the people belonging to age-group 20-25 and having

experience = null and grades = high, often selected the jobs belonging to the category J1, J2, J5 and J6.

Table 5.1: Sample matrix representation of mined rules against different job categories

Age	Gender	Exp	Grade	MStatus	Loc	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12
20-25	-	N	High	-	-	1	1	0	0	1	1	0	0	0	0	0	0
26-30	-	1-5	-	-	-	0	0	0	0	0	1	1	1	0	0	1	1
31-40	M	-	-	-	-	0	0	0	0	1	0	0	0	1	1	1	1
35-40	F	-	-	M	N	1	0	0	0	0	0	1	0	0	1	1	0
35-40	F	-	-	D	N	0	0	0	0	0	0	1	0	0	1	1	0

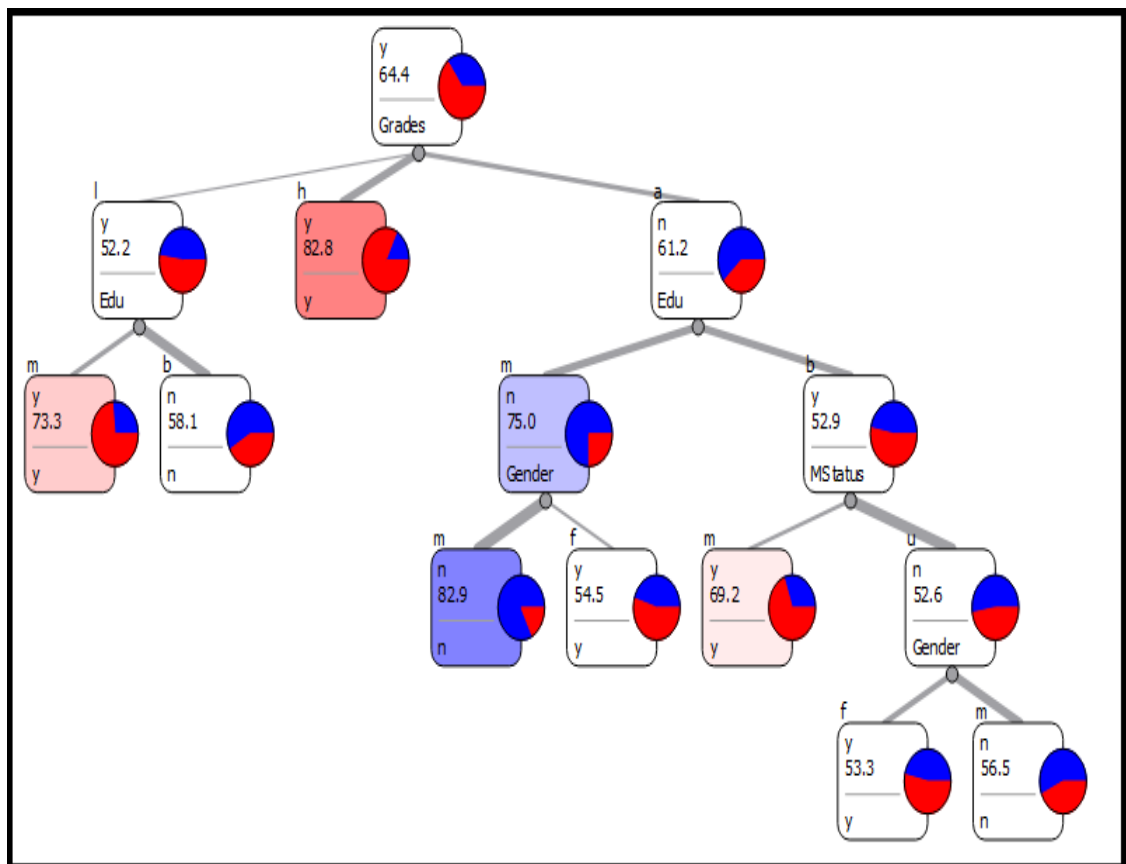


Figure 5.3: Pictorial representation of a decision tree induction rule

Example 5.2: Last Rule (Table 5.1), Age=35-40, gender=Female and MStatus=Divorcee. The 4 preference matrices are Company Matrix: $[0 \ 1/3 \ 2/3 \ 0]$, Position Matrix: $[0 \ 2/3 \ 1/3 \ 0]$, Income Matrix: $[2/3 \ 1/3 \ 0]$, Location Matrix: $[1/1 \ 0 \ 0 \ 0]$.

5.3 Phase-I Recommendation

Let us take an example to understand the complete procedure of how the system generates its first recommendations.

Example 5.3: Here is a candidate that has the following details: {27, male, unmarried, graduate, Computer Science, 65%, 2 years experience, New Delhi, (C++, Oracle, Data Structures, Algorithms, Machine Learning, English)} and we have these 10 jobs in our database: {J1, J2, J3, J4, J5, J6, J7, J8, J9, J10}, as shown in Table 5.2, with only those specifications which are relevant only for the first step of the algorithm.

Table 5.2: Sample jobs in the database

Jobs	Field/ Major	Qualification	Min. Experience
J1	CSE	Graduation	2
J2	CSE	Graduation	2
J3	M.E.	Graduation	2
J4	CSE	Graduation	1
J5	CSE	Graduation	4
J6	CSE	Graduation	1
J7	CSE	Graduation	0
J8	CSE	Graduation	2
J9	CSE	Graduation	3
J10	CSE	Masters	1

Step 1: After the short listing from the above jobs list: J3, J5, J9 and J10 get eliminated, as these jobs are not fit for the candidate either in case of major, minimum qualification required or minimum experience needed for the respective job. In some cases age can also be considered as an important factor, which is currently ignored here. So, we are left with only 6 jobs in our basket {J1, J2, J4, J6, J7 and J8}. The shortlisted jobs after Step 1 are shown in Table 5.3. These all jobs are, at this step, found appropriate for the candidate and are considered for further filtering and processing. Here, redundancy in jobs, if any, is also removed.

Table 5.3: Shortlisted jobs after step 1

Jobs	Field/ Major	Qualification	Min. Experience
J1	CSE	Graduation	2
J2	CSE	Graduation	2
J4	CSE	Graduation	1
J6	CSE	Graduation	1
J7	CSE	Graduation	0
J8	CSE	Graduation	2

Step 2: Now calculating the cosine similarity as shown in Table 5.4. As already discussed, it is calculated between the candidate's possessed skills and respective job's required skills.

Table 5.4: Calculation of cosine based similarity in step 2

Job	Skills	Skills Vector	Candidate Vector	Cosine similarity
J1	MySQL, PHP, Data structure, English.	[1,1,1,1]	[0,0,1,1]	.707
J2	OpenCL, Networking, English, German	[1,1,1,1]	[0,0,1,0]	.5
J4	Python, MySQL, Machine Learning, English, German	[1,1,1,1,1]	[1,0,1,1,0]	.774
J6	C++, Data Structure, Algorithm, English	[1,1,1,1]	[1,1,1,1]	1
J7	C++, Oracle, Algorithms, English.	[1,1,1,1]	[1,1,1,1]	1
J8	Smalltalk, DB2, JSP, Machine Learning, English	[1,1,1,1,1]	[1,0,0,1,1]	.774

Step 3: Calculating the rules weight by applying decision tree induction rules. So, for the above mentioned candidate: {26-30, M, U, B, A, 1-5, N}, the preference matrices for a candidate belonging to age group: 26-30 and having experience: 1-5 years, are: company [0,1/5,2/5,2/5], position [0,2/5,3/5,0] and pay-scale [1/5, 3/5, 0] according to the job categorization shown in Table 5.5.

Location preference matrix is not considered as it is assumed that the group has applied equally in all 4 regions and hence adding these will not result into any new information. At the end normalizing the rules weight using equation 5, we get the normalized rules weight.

Table 5.5: Final weight calculation in step 4

Job	Cosine similarity	Company	Position	Pay Scale	Rules Weight	Normalized Rules Weight	Final Weight
J1	.707	B	C	H	$\{1/5+3/5+1/5\} = 1$	.5	.6035
J2	.5	B	B	M	$\{1/5+2/5+3/5\} = 1.2$	.666	.583
J4	.774	D	B	M	$\{2/5+2/5+3/5\} = 1.4$	.833	.8035
J6	1	C	C	M	$\{2/5+3/5+3/5\} = 1.6$	1	1
J7	1	D	D	L	$\{2/5+0+0\} = .4$	0	.5
J8	.774	C	C	L	$\{2/5+3/5+0\} = 1$	.5	.637

Step 4: Calculate the final weights using equation 6 as shown in Table 5.5.

Step 5: Now finally sorting the jobs in decreasing order. The final recommendations provided to the candidate of Example 5.3, in sorted order are {J6, J4, J8, J1, J2, J7} as shown in Table 5.6.

Table 5.6: Final sorting of the jobs after step 5 of phase-I

Job	Cosine similarity	Rules Weight	Normalized Rules Weight	Final Weight	Final Ranking
J6	1	$\{2/5+3/5+3/5\} = 1.6$	1	1	1
J4	.774	$\{2/5+2/5+3/5\} = 1.4$	.833	.8035	2
J8	.774	$\{2/5+3/5+0\} = 1$	.5	.637	3
J1	.707	$\{1/5+3/5+1/5\} = 1$	.5	.6035	4
J2	.5	$\{1/5+2/5+3/5\} = 1.2$	.666	.583	5
J7	1	$\{2/5+0+0\} = .4$	0	.5	6

5.4 Phase-II Recommendation

This phase starts as soon as the candidate applies for at least 10 jobs i.e. when he/she become active in the system.

Step 1: Short list the jobs: Considering the Example 5.3, we are again left with only 6 jobs in our basket or job set: {J1, J2, J4, J6, J7 and J8}. For Example 5.3, it is same as calculated in Table 5.3.

Step 2: Calculate cosine similarity: This step is also same as that of phase 1 recommendations and hence is again same as calculated in Table 5.4.

Table 5.7: Generation of normalized rules weight using 4 preference matrices

Job	Cosine Similarity	Company	Position	Pay Scale	Loc	Rules Weight	Normalized Rules Weight
J1	.707	B	C	H	W	$\{1/10+3/10+2/10+2/10\}=.8$	0
J2	.5	B	B	M	S	$\{2/10+6/10+8/10+2/10\}=1.8$	.666
J4	.774	D	B	M	N	$\{2/10+6/10+8/10+7/10\}=2.3$	1
J6	1	C	C	M	W	$\{5/10+3/10+8/10+2/10\}=1.8$	.666
J7	1	D	D	L	N	$\{2/10+1/10+0+7/10\}=1$	.133
J8	.774	C	C	L	N	$\{5/10+3/10+0+7/10\}=1.5$	.466

Step 3: Calculate preference matrices from customer's latest preferences for jobs: Suppose we have the list of following 10 jobs that the customer has recently applied for: {J11, J12, J13, J14, J15, J16, J17, J18, J19, J20} and from these jobs, suppose the preference matrices are as follows:

- Company [1/10 2/10 5/10 2/10]
- Position: [0 6/10 3/10 1/10]
- Pay-scale: [2/10 8/10 0]
- Location: [7/10 2/10 0 2/10]

Table 5.8: Final recommendation of phase-II after step 5

Job	Cosine Similarity	Rules Weight	Normalized Rules Weight	Final Weight	Final Ranking
J4	.774	2.3	1	.887	1
J6	1	1.8	.666	.833	2
J8	.774	1.5	.466	.6203	3
J2	.5	1.8	.666	.583	4
J7	1	1	.133	.5665	5
J1	.707	.8	0	.3535	6

Now using these normalized rules weight can be calculated as shown in Table 5.7.

Step 4: Generate the final weights using equation 6.

Step 5: Sort the jobs in descending order: Final personalized list of the job recommendations for candidate (Example 5.3) after knowing his job preferences are: {J4, J6, J8, J2, J7, J1}.

Table 5.8 shows the final sorted list of jobs that are offered to the candidate of Example 5.3. So, the final ordering of jobs after phase-II recommendations are {J4, J6, J8, J2, J7, J1}. Note that this job ordering is different from the phase-I recommendations {J6, J4, J8, J1, J2, J7}, as these are generated only after the candidate becomes active in the system and applies for the minimum no. of jobs required by the system for preference matrices generation. So, phase-II recommendations are more personalized recommendations that are made after keeping the candidate's specific job preferences.

6.1 Results Analysis

After exploring the datasets for decision tree induction rules and creating up all the preference matrices and final weights, the personalized jobs were recommended to the candidate. These results obtained were compared to that of the basic existing techniques of making the recommendations i.e. content based filtering and memory based collaborative filtering. The prediction accuracy was used as a parameter to judge the importance of made recommendations [29-31].

Table 6.1 shows the prediction accuracy (in percentage) obtained after applying CBRS, CFRS and PRS-I to the same result set.

Table 6.1: Comparison of prediction accuracies during phase-I

Top N Recommendations	CBRS (%)	CFRS (%)	PRS-I (%)
Top 5	30	30	50
Top 10	43	42	58
Top 20	51	48	66
Top 40	56	52	72

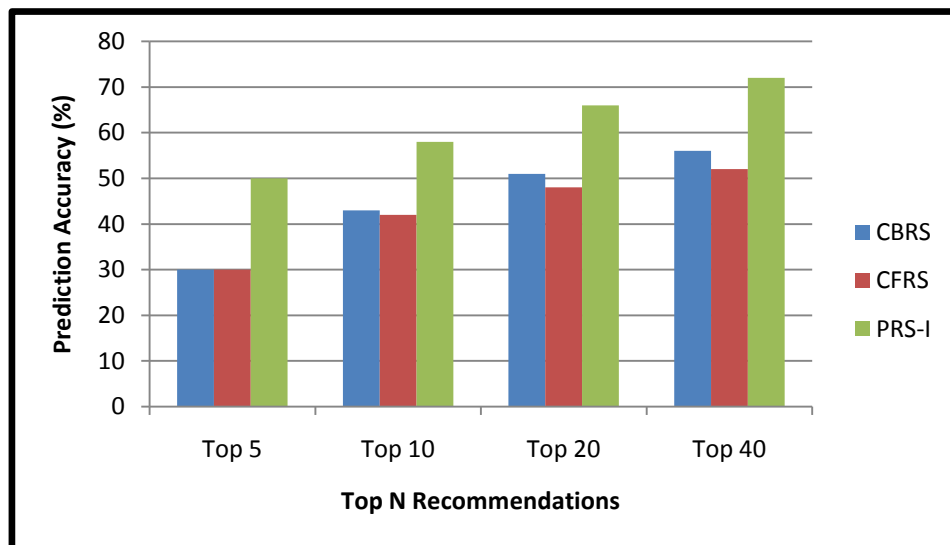


Figure 6.1: Comparison of prediction accuracy for phase-I

Here, CBRS stands for Content Based Recommender Systems, CFRS for memory based Collaborative Filtering Recommender Systems, PRS for our Proposed Recommender System. PRS-I stands for Proposed Recommender System for Phase I.

The prediction accuracies were calculated distinctly for Top 5, Top 10, Top 20 and Top 40 recommendations made to the candidate. In first phase, a prediction accuracy of about 72 percent was perceived as against lower percentages of traditional recommender methods. Figure 6.1 shows the resulting graph obtained after plotting the prediction accuracies of Table 6.1.

Table 6.2: Comparison of prediction accuracies during phase-II

Top N Recommendations	CBRS (%)	CFRS (%)	PRS-II (%)
Top 5	30	30	55
Top 10	43	42	62
Top 20	51	48	70
Top 40	56	52	81

Table 6.2 shows the comparison of prediction accuracies between the CBRS, CFRS and PRS-II i.e. Proposed Recommender System for Phase II. Here also, the prediction accuracies were calculated for Top 5, 10, 20 and 40 recommendations made to the candidate. The prediction accuracy of about 81 percent was obtained for the second phase. Figure 6.2 shows the comparison of prediction accuracies obtained after plotting the results of Table 6.2.

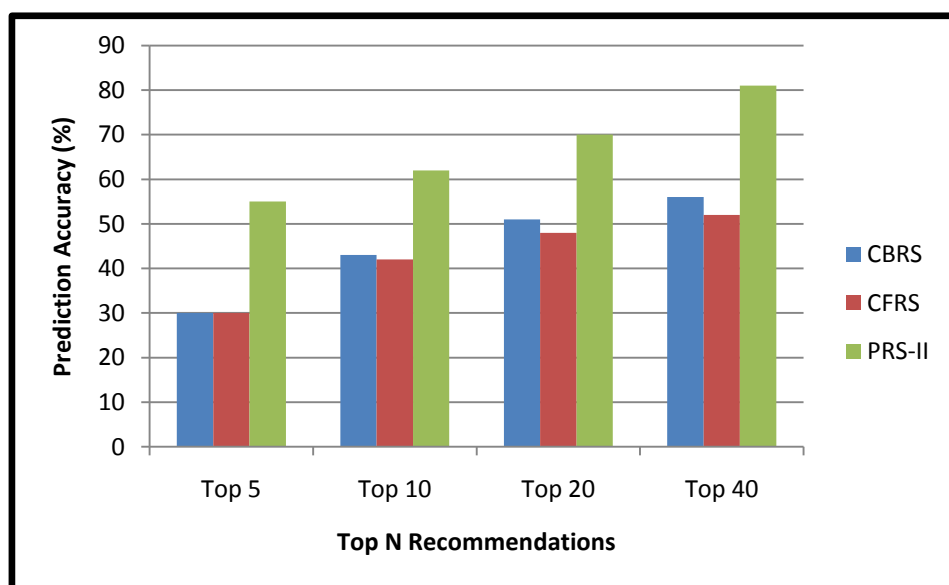


Figure 6.2: Comparison of prediction accuracy for phase-II

However when a comparison was made between the phase-I and phase-II results, the prediction accuracies obtained after phase-II were quiet higher as compared to that of phase-I. The Table 6.3 summarizes their results and Figure 6.3 represents this comparison in the form of line graph.

Table 6.3: Comparison of prediction accuracies for phase-I and phase-II recommendations

Top N Recommendations	PRS-I (%)	PRS-II (%)
Top 5	50	55
Top 10	58	62
Top 20	66	70
Top 40	72	81

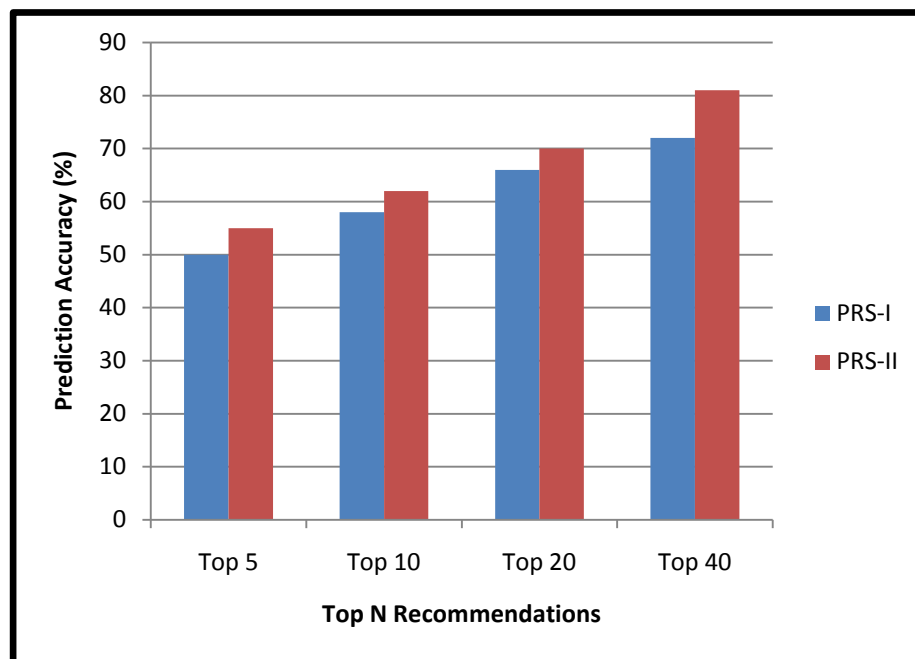


Figure 6.3: Comparison of prediction accuracy for phase-I and phase-II

6.2 Discussions

The reason behind the significant difference between the prediction accuracies of the CBRS, CFRS and PRS lies behind the fact that the proposed recommender system lays much stress on the generalized or current preferences of the candidate. It considers the candidate job preferences as against just matching its content blindly with the jobs present in the database, and then short listing the jobs on the basis of similarity or just suggesting those jobs that are in actually considered relevant by the similar profile candidates with that of the target candidate.

Now, concentrating on the reason behind the difference between the prediction accuracies of 2 phases lies in the fact that in first phase the filtering is based on content based filtering and the generalized rules that exist in the knowledge base for that particular candidate group. Whereas, in the second phase the recommendations are more candidate centric as those depend on the latest job preferences of the candidate.

It also proves to be efficient as it eliminates the irrelevant, outdated and stale jobs out of the basket and considers only latest or jobs that are freshly (according to the time period) applied by the candidate.

Table 6.4 compares the proposed system with the various existing systems discussed in Chapter 2. Here, the following abbreviations are used:

- CBR: Content Based Recommendation
- CFR: Collaborative Filtering Based Recommendation
- KBR: Knowledge Based Recommendation
- HyR: Hybrid Recommendation

Table 6.4: Comparison between the different job recommender systems

	CASPER [13]	PROACTIVE [14]	BILATERAL [15]	iHR [18]	Machine Learned [20]	Proposed System
Input For User Profile	Candidate Behavior and Search Query	Candidate Information	Candidate Information	Candidate Information and Behavior	Candidate Information	Candidate Information and Behavior
Recommendation Strategy	Memory based & Cluster based CFR, Case Based Reasoning	CBR, KBR	Probabilistic Hybrid Recommendation Engine	Different strategies for different groups (CBR, CFR, HyR)	Supervised learning Based DTNB Hybrid Classifier	CBR, Model based CFR, KBR
Output of Recommendation	One complete list of recommended jobs	4 different job recommendation lists	List of recommended jobs as well as list of recommended CVs	One complete list of recommended jobs	List of predictions of next job positions	One complete list of recommended jobs
User Feedback Mechanism	Implicit	Explicit	Explicit	Implicit	-	Implicit
Pros	1. Adapts to user needs 2. Implicit user preference	1. Four different recommendation modules 2. Uses	1. Two sided recommendations	1. Different recommendation strategies for different clusters of	1. Considers transition history for recommendations	1. Implicit user preference considered 2. Adapts to the user

	considered	ontology to cluster jobs		users		needs 3. Different recommendations for different group of users
Cons	1. User profile not much accurate 2. One way recommendation	1. One Way Recommendation 2. Explicit Feedback required 3. Knowledge engineering problem	1. Explicit Feedback Required 2. There are no perfect or standard methods/ measures defined	1. One Way Recommendation 2. Data Sparsity problem	1. One Way Recommendation 2. Data Sparsity problem	1. One Way Recommendation 2. Data Sparsity problem at initial level

Now concentrating on the limitations of the proposed recommender system, one important limitation is that while categorizing the company data on the basis of job positions, job ranking (levels) and pay-scale offered, there were certain trade-offs. For an example, considering 2 companies, one having level A in the job market, but offering a low package for a high position for its job. Whereas the other also of level A in the job market, but offering a high package for the same position for its job. So, here although the positions are same, company levels are same but still the packages are categorized into 2 different categories. This leads to a mismatch, unevenness of the normal trend in the existing categorization system. So, such trade-offs were made, in the data categorization part of the existing system.

The second limitation lies in the fact that during the first phase of the recommendations, if no rule in the knowledge base matches the candidate's group or category, then no preference matrices are generated and as a result, the algorithm of first phase merely reduces to content based recommender system, in worst case. However, as the candidate applies for a minimum of 10 jobs, the second phase starts over and overcome this limitation regarding the job preferences and efficiency is thus re-gained.

CHAPTER 7

CONCLUSIONS

Here the efforts were put to take into consideration the job preferences of the candidates along with the content based profile matching. It along with increasing the prediction accuracy also helped to solve the problem of providing direction to the candidates who are not clear about their job goals as general group preferences are imposed. However, in case of candidates having exceptional path carriers, the system adapts itself by focusing on their latest job preference behavior and providing them the list of recommendations accordingly. Tracking the present preferences of the candidate regarding the job, helps to prioritize only the relevant jobs as against the irrelevant jobs that are shortlisted after the content based matching of the candidate.

7.1 Answering the Research Question

The research work discussed in Chapter 4 and 5, serves well for the problem that was proposed in the Chapter 3. It fits better on all the four parameters/facts as well as tries to judge the gap between the existing systems and the target systems in the following ways:

- The system keeps track of the preferences which are reflected behind the decisions of the people, in applying for a particular job.
- The group behavior is preserved in the form of rules obtained after the data mining.
- Even the people, who are having exceptional carrier path curves, are also well considered and well adapted to the system.
- As every candidate's preferences are preserved separately, the system is able to distinguish even between those two candidates, who have got their similar looking profiles.

Thus the system helps to judge the gap by not only conserving the contents of the candidate's profile but also its job preferences, comparatively resulting into an efficient system.

7.2 Future Work

- As the proposed system is only for the job or better job aspiring candidates, the system can be extended for the recruiters also. In that the well qualified candidates can be suggested to the recruiters, for better personnel selection.
- As one of the assumptions here is that the resumes are all in place, whereas in real world this is not the case. The candidates used to submit their resumes at online sites. So, a resume miner can also be integrated for making the system complete.
- The granularity level used in data categorization may be increased for more accurate predictions.
- More features can be added or the contextual information can also be added to build more accurate job recommender systems.

REFERENCES

- [1] P. Melville, V. Sindhwani, “Recommender Systems,” *Encyclopedia of Machine Learning*, pp. 829-838, 2010.
- [2] M. J. Pazzani, D. Billsus, “Content-Based Recommendation Systems,” *The Adaptive Web*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4321, pp. 325-341, 2007.
- [3] J. S. Breese, D. Heckerman, C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” *Proceedings of the fourteenth conference on Uncertainty in Artificial Intelligence*, UAI'98, pp. 43-52, 1998.
- [4] J. Ben Schafer, D. Frankowski, J. Herlocker, S. Sen, “Collaborative Filtering Recommender Systems,” *The Adaptive Web*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4321, pp. 291-324, 2007.
- [5] X. Su, T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances in Artificial Intelligence*, vol. 2009, no. 4, pp. 1–20, Jan. 2009.
- [6] B. Smyth, “Case Based Recommendation,” *The Adaptive Web*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4321, pp. 342-376, 2007.
- [7] R. Burke, “Hybrid Recommender Systems: Survey and Experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, Nov 2002.
- [8] B. Smyth, “Hybrid Web Recommender Systems,” *The Adaptive Web*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4321, pp. 377-408, 2007.
- [9] A. Gunawardana, C. Meek, “A unified approach to building hybrid recommender systems,” *Proceedings of the 3rd ACM conference on Recommender systems*, RecSys '09, ACM, New York, NY, USA, pp. 117-124, 2009.
- [10] S. Zheng, W. Hong, N. Zhang, F. Yang, “Job Recommender Systems: A

- Survey,” *7th International Conference on Computer Science & Education*, Melbourne, VIC, pp. 920 – 924, Jul. 14-17, 2012.
- [11] S. T. Al-Otaibi, M. Ykhlef, “A survey of Job Recommender Systems,” *International Journal of the Physical Sciences*, vol. 7, no. 29, pp. 5127-5142, Jul. 26, 2012.
- [12] R. Rafter, K. Bradley, B. Smyth, “Automated Collaborative Filtering Applications for Online Recruitment Services,” *Adaptive Hypermedia and Adaptive Web-Based Systems*, Lecture Notes in Computer Science, vol. 1892, pp. 363-368, 2000.
- [13] R. Rafter, K. Bradley, B. Smyth, “Personalized Retrieval for Online Recruitment Services,” *Proceedings of the 22nd Annual Colloquium on Information Retrieval*, IRSG 2000, Cambridge, UK, 5-7, Apr. 2000.
- [14] D. H. Lee, P. Brusilovsky, “Fighting Information Overflow with Personalized Comprehensive Information Access: A Proactive Job Recommender,” *Third International Conference on Automatic and Autonomous Systems*, ICAS07, Athens, pp. 21, Jun. 19-25, 2007.
- [15] J. Malinowski, T. Keim, O. Wendt, T. Weitzel, “Matching People and Jobs: A Bilateral Recommendation Approach,” *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, HICSS '06, vol. 6, pp. 137c, Jan. 4-7, 2006.
- [16] L. Li, T. Li, “MEET: A Generalized Framework for Reciprocal Recommendation Systems,” *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM'12, ACM, New York, NY, USA, pp. 35-44, 2012.
- [17] H. Yu, C. Liu, F. Zhang, “Reciprocal Recommendation Algorithm for the Field of Recruitment,” *Journal of Information & Computational Science*, vol. 8, no. 16, pp. 4061–4068, 2011.
- [18] W. Hong, S. Zheng, H. Wang, J. Shi, “A Job Recommender System Based on User Clustering”, *Journal of Computers*, vol. 8, no. 8, pp. 1960-1967, 1, Aug.

2013.

- [19] M. Hall, E. Frank, “Combining naïve Bayes and Decision tables,” *Proceedings of 21st International Florida Artificial Intelligence Research Society Conference*, AAAI Press, Coconut Grove, Florida, USA, pp. 318-319, May 15-17, 2008.
- [20] I. Paparrizos, B. B. Cambazoglu, A. Gionis, “Machine learned Job Recommendation,” *Proceedings of the fifth ACM conference on Recommender systems*, RecSys ‘11, ACM, New York, NY, USA, pp. 325-328, Oct. 23, 2011.
- [21] W. Hong, S. Zheng, H. Wang, “Dynamic User Profile-Based Job Recommender System,” *8th International Conference on Computer Science & Education (ICCSE)*, Colombo, pp. 1499-1503, Apr. 26-28, 2013.
- [22] C. F. Chien, L. F. Chen, 2008, “Data Mining to improve personnel selection and enhance human capital: A case study in high-technology industry,” *Expert Systems with Applications: An International Journal*, vol. 34, no. 1, pp. 280-290, Jan. 2008.
- [23] Y. H. Chao, J. K. Kim, S. H. Kim, “A Personalized recommender system based on web usage mining and decision tree induction,” *Expert systems with Applications*, vol. 23, no. 3, pp. 329-342, 2002.
- [24] B. Mobasher, “Data Mining for Web Personalization,” *The Adaptive Web*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4321, pp. 90-135, 2007.
- [25] J. Demšar, B. Zupan, G. Leban, T. Curk, “Orange: From Experimental Machine Learning to Interactive Data Mining,” *Knowledge Discovery in Databases: PKDD 2004*, Lecture Notes in Computer Science, vol. 3202, pp. 537-539, 2004.
- [26] “Orange: Data Mining Fruitful and Fun”, *Orange*, [Online], Available: <http://orange.biolab.si>, Accessed on: Jun 23, 2014.
- [27] J. Demšar, T. Curk, A. Erjavec, “Orange: Data Mining Toolbox in Python,”

Journal of Machine Learning Research, vol. 14, pp. 2349-2353, 2013.

- [28] J. Demšar, B. Zupan, “Orange: Data Mining Fruitful and Fun - A Historical Perspective,” *Special Issue: 100 Years of Alan Turing and 20 Years of SLAIS Guest Editors*, vol. 37, no. 1, pp. 55–60, 2013.
- [29] F. H. d. Olmo, E. Gaudioso, “Evaluation of recommender systems: A new approach,” *Expert System with Applications*, vol. 35, pp. 790-804, 2008.
- [30] M. Ge, C. D. Battenfeld, D. Jannach, “Beyond Accuracy: evaluating recommender systems by coverage and serpendipity,” *Proceedings of the fourth ACM Conference on Recommender Systems*, RecSys '10, ACM, New York, NY, USA, pp. 257-260, Sep. 2010.
- [31] G. Shani, A. Gunawardana, “Evaluating Recommender Systems,” *Recommender Systems Handbook*, US: Springer, pp. 257-297, 2011.
- [32] S. Mittal, A. Singh. “E-recruitment In India: A Study Of Major Job-portals And Upcoming Trends.” *Golden Research Thoughts*, vol. 3, no. 2, Aug. 2013.

LIST OF PUBLICATIONS

Accepted Paper

1. Anika, D. Garg, “Applying Data Mining Techniques in Job Recommender System by Considering Candidate Job Preferences,” accepted in *3rd International Conference on Advances in Computing, Communications & Informatics, ICACCI 2014*, IEEE, GCET, Greater Noida, India, Sep. 24-27, 2014.