Recommendation System for Workers & Customers for Informal Jobs

Manasi Purkar[a],*, Omkar Joshi[a], Abhishek Salape[a], Ankush Patil[a], Varad Kulkarni[a], Dr. Pravin Futane[b]

aUndergraduate Student, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.
bProfessor, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

Abstract: Recommender systems are software applications that provide or suggest items to users. These systems use filtering techniques to provide recommendations. The Purpose of this recommendation system was to provide services to small or part-time workers. It has been observed by the team that recommendation systems that are studied have not focused on small workers like electricians and carpenters but the corporate people. Recent trends in technology have made us dependent on technology too much itself. To cope up with the problems of urbanization and employment trends in this ever-changing world, a well-suited system for corporate workers as well as skilled laborers should coexist. This model is designed to help recruiters to hire employees based solely on work type and rating. A Recruiter can hire a person for a specific task or time frame dependent on what type of employee he/she is expecting. It is designed in such a way that it will help to reduce the gap between both of them leading to a hassle-free experience for an employer as well as an employee. The content-based technique is adopted as it is used to know the content of both user and item. A Manually generated dataset is used by taking the reference of the standard dataset of the formal employees present on the Kaggle website. This system is based on the Vector Space Model and TF-IDF vectorizer. A literature survey of several research papers in the same domain was conducted. The recommendation system has been implemented in the python programming language. The results obtained were quite accurate which helps to recommend jobs to workers and workers to customers in the required work field.

Keywords—Recommender System, Content-Based Filtering, Job Recommendation, Machine Learning, Python.

1. Introduction

The recommender systems are being used in every possible system for example, movie recommendation, book recommendation, etc. However, the different types of recommendations may provide according to the domain of its use. Job Recommendation will provide job to the workers like painter, repairer, carpenter, etc. satisfy the need of the customer.

In job recommendation systems, there are varieties of workers, having different education levels and skills. Based on the worker's respective background details, each one of them expects to get only those job recommendations that are highly relevant for that particular worker. Our recommendation system is designed for jobs by making the use of ratings and by matching their skills with the required ones. The system may save a lot of time for both customers and workers for satisfying their needs. The main aim of this paper is to provide proper knowledge about the structure of the developed system.

By creating an easy job recommendation system where everyone will have a fair chance to access it properly and experiencing the flow. The system requires the details of the worker like name, working area, skills, rating like this and similarly, for customers which work, skills are required, location, name, etc. In this system, we used a dataset containing all these fields.

Based on the person-job fit premise, our system proposes a framework for job recommendations based on the professional skills of job seekers. Our system automatically extracted the skills from the job seeker profiles using a variety of text processing techniques. Therefore,

perform the job recommendation using TF-IDF. This system was implemented on two different datasets one for employees (workers) and the other for employers (customers). This system will provide jobs to small workers and also fulfil the need of customers for small jobs at home or workplace.

Recommendation system techniques are of different types based on input data provided and required results (B. Patel, P. Desai, & U. Panchal, 2017):
   1. Collaborative Recommendation Approach
   2. Content-Based Recommendation Approach
   3. Hybrid Recommendation Approach

In this paper further, section 2 defines the problem, section 3 gives the proposed system used for this recommendation system including dataset and other things used, section 4 gives the literature survey done, section 5 clarifies implementation of this recommendation system, section 6 gives results, section 7 mention conclusion and section 8 talks about the future scope of this recommendation system.

2. Problem Definition

Indeed, even in today's times of automation/ computerization, the greater part of the procedures involved in finding jobs as a small worker is generally manual. Making this procedure completely computerized will not just reduce the time in determination but will also add clinical accuracy to the process. This is also done presently but what makes difference is the proficiency with which it is done.

In the current job recommendation systems, they mainly focus on suggesting higher-end jobs such as jobs regarding IT sector, creative jobs such as graphic designer, etc. but none of them is providing service to find small jobs like electrician, painter, etc.

In today's world, people could not find these workers near their house. Workers also can't find their customers, so this project will help both customers and workers to satisfy the need.

3. Proposed System

3.1. Setup

1. Jupyter Notebook
2. Python==3.5
3. scikit-learn

Recommendation systems are active information filtering systems that provide personalized information to a user based on his interests, relevance of the information, etc. They are used widely for recommending, articles, movies restaurants, items to buy, and many more. The Collaborative Filtering approach was not the perfect one for our system because of some of its features.

Let's consider a set of customers and a set of workers. If the customer will get the recommendation of a particular worker based on the rating, and if the worker does not have any kind of rating at present, then it will not be recommended to the customer. So, to overcome this, we are using a content-based filtering approach.

Content-based filtering uses the skills of the workers to recommend other workers similar to what the customer likes, based on their previous actions or according to their entered requirement. For example, if a customer is interested in a particular worker, then based on that content, our system will recommend him/her the worker having a similar kind of skills.

3.2. Dataset

For formal jobs standard dataset available on Kaggle was used to train our system. But for our project requirement Dataset is generated by manually entering data and by taking help Mockaroo website as no dataset was already available for small workers like painter, driver, carpenter, etc. Mockaroo site is used generate random data. Using this random data, we trained this system.

3.3. Vector Space Model

The Vector space model is a way of representing text documents in multi-dimensional space as vectors. In this model, in which each term is a dimension and each document is a linear combination of vectors along the axes. The Vector space model is used in this recommendation system for the relevance of ranking and text similarity. The value of cosine ranges from -1 to +1. In the text, the similarity range will be 0 to 1.

Fig. 1 shows 2 vectors that are documents represented in the graph as Document1 and Document2. The user query is represented as a query. The similarity between Document1 and Query is proportional to the angle Cosine theta1 between them and the similarity between Document2 and Query is proportional to the angle Cosine theta2 between them.
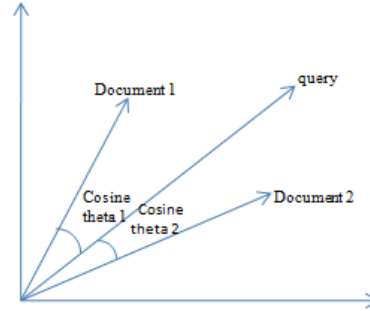
Fig. 1 - Representation of Cosine Similarity Graph

Mathematics formula (1) for calculating cosine similarity is given below:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

An example query would be:
If document D contains one instance of apple, 2 instances of word orange then the co-ordinates of the vector will be <1,1,2> then according to the above formula, cosine similarity will be 0.51.

3.4. Creating a TF-IDF Vectorizer

TF-IDF is a tool that calculates the relevance of words to a document in a collection of documents. It is used in machine learning as a weighing factor. TF-IDF calculates the ranking of words which is based on user input (A. Mishra and S. Vishwakarma, 2015). It has 2 terms TF and IDF.

The term TF which stands for Term Frequency is the number of times the single word or term appears in the document divided by the total number of terms in the document. The term IDF is the total no of documents let's say 'w' divided by the number of documents in our datasets that have the given term.

$$w_{x,y} = tf_{x,y} * \log(N / df_x) \quad (2)$$

$tf_{x,y}$ = Frequency of x in y
df_x = Number of documents that contain x
N = Total number of documents

The Given an example to understand the concept better, take two documents as D1 and D2.

D1 = He is intelligent.

D2 = he is not intelligent.

Table 1 – TF-IDF Vectorizer example

WORDS	TF		IDF	TF*IDF	
	D1	D2		D1	D2
He	1	1	log(2/2)	0	0
Is	1	1	log(2/2)	0	0
Intelligent	1	1	log(2/2)	0	0
Not	0	1	log(2/1)	0	log(2)

Here the word not is really important as it is the decision of the ranking. So, the equation should reciprocate that the word is quite important.

In the conclusion, it is reflected that the words 'He', 'Is' and 'Intelligent' are not significantly important as their Tf*Idf value is zero. The important word in these 2 documents that differentiates the ranking is the word 'Not'. for that one must import necessary packages which are pre-defined.

Packages include:

Import gensim

From gensim import corpora

From gensim.utils import simple_preprocess

In Python, using scikit-learn the TF-IDF calculation becomes pretty easy. The platform provides a pre-calculated TF-IDF score which is word by word for each document's description. By default, the words that add no significant value to the system like 'the', 'a', 'an' and 'is' are ignored by the system automatically.

4. Literature Survey

The Literature survey was the first step to gain the knowledge of existing research in the field of recommendation systems. It was beneficial to make the effective use of the time by focusing on appropriate resources and also helped to achieve the uniqueness of the project.

After analyzing several research papers, it was concluded that according to the project requirement, mainly collaborative based, content-based, and sometimes a hybrid-based system was used. The main focus was on the algorithms, input datasets, approaches, methodologies they have adopted to achieve the respective success rate. It was useful to know the popularity of a particular algorithm, different sources of the datasets.

In (Y. Zhang, C. Yang, & Z. Niu, 2014), Collaborative Filtering technique is used. Item-based collaborative filtering, Similarity calculation are some of the methods used in this project. As a dataset, they used the information of workers available on the internet.

In (S. Choudhary, S. Koul, S. Mishra, A. Thakur, & R. Jain, 2016), Naive Bayes algorithm is used. Data acquisition, data sanitization, count filtering, and Bayesian Ranking are some of the methods mentioned in this paper. Skills and previous job history details are used as an input dataset for this project.

In (S. Ahmed, M. Hasan, M. N. Hoq, & M. A. Adnan, 2016), Collaborative Filtering algorithm is used. Data analysis, data preprocessing and system design are some of the methods used in this project. Dataset was obtained from the career oriented social networking site, XING.

In (S. Dong, Z. Lei, P. Zhou, K. Bian, & G. Liu, 2017), Contextual Online learning algorithm is used. Methods such as Monte-Carlo Tree and Context Space Dividing are used in this project. Database from Work4 test data was used to build this system.

After analyzing all methods and their features content-based recommendation system was decided and further study on implementation of content-based technology in different fields was continued.

In (B. R. Cami, H. Hassanpour, & H. Mashayekhi, 2017), content-based filtering is used. Where interest's extraction technique is used for discovering the user interests and analyze the user profile. After that preference inferring technique is used to calculate the probability that the user selects a particular interest and a prediction module is used to provide a recommendation list. Clustering is used for analysis to gain valuable insights from data. These are some methods mentioned in this paper movie information for the dataset is gathered from IMDB. Similarly, the use of this technology in career recommendations and recommender for spoken documents was studied.

In (T. V. Yadalam, V. M. Gowda, V. S. Kumar, D. Girish, & N. M., 2020) NLP on feedback that has given by the users was used with following steps like tokenization in which the sentence was separated into singular words called as tokens. After that normalizing which contain lemmatizing and stemming. Data cleaning incorporates spelling and grammar revision and after that sentimental analysis is performed where the characterization of feelings was done. The dataset has been created by pre-processing and combining multiple databases. This project uses content-based technology using NLP as it thinks about the client's field of interest and the user's profile interest and abilities, which can produce applicable career suggestions for the user.

In (J. Wintrode, G. Sell, A. Jansen, M. Fox, D. Garcia-Romero and A. McCree, 2015) use ratings and features that characterize media to model users interests or information needed for making future recommendations. Content-based recommendation, speech retrieval, low resource, i-vectors are some of the methods mentioned in this paper. Thus, in the spirit of the music recommendation systems, a fully automatic

MANASI PURKAR, OMKAR JOSHI, ABHISHEK SALAPE,
ANKUSH PATIL, VARAD KULKARNI, DR. PRAVIN FUTANE

approach was adopted, that uses low and high resource automatic speech recognition (ASR) to tokenize the message. After that acoustic i-vectors to characterize speaker, language, and gender; speech and music activity detection to characterize proportions of each, zero resource discovery tools to characterize background acoustic events and bags of sub word units. Content-Based Features like English Bags of Words and Senones, Acoustic i-vectors, Speech and Music Activity, Pseudo terms, and Semantic Features are used in this paper. The data set consists of audio stripped from the Creative Commons website. After studying all these papers from different fields insides of the content-based technique were understood.

According to the survey, all the projects done on the recommendation system were for the formal employees but none of them focuses on the informal employees/ workers and this is what the opportunity lies in! The goal is to help the workers to get jobs using the recommendation system.

5. Implementation

5.1. Recommending jobs to workers

This is the first part of the application that is recommending jobs to workers. Here a dataset of customer requirements was used and jobs were recommended for workers using this dataset. In this application, a worker should enter the job title in his working area and jobs from requirements of customers related to that job title will be recommended to the worker.

Dataset of customer requirements contains customer name, job title, skills, location these fields. Jobs are recommended by combining job title and skills fields in a new field that is a job. In the implementation process of this first cleaning of the dataset was done and after that using Tfidf Vectorizer English sentences are converted into vectors and a sparse matrix from vectors of job field was created.

In the next step, sigmoid_kernel was used for finding cosine similarity. Using sigmoid_kernel, cosine similarity in jobs that is the similarity of one job to another job was found out bypassing same sparse matrix in sigmoid_kernel.

Indices to job titles were given. After that in the main recommendation function input of job title will be taken from the worker and finding its index in the dataset if that job title got in the dataset, using its index similarity scores were found out and sorting of that scores was done. The first 8 highly similar job titles were recommended to the worker.

But to find exactly similar job title in the dataset is sometimes difficult hence if exactly the same job title didn't get in the dataset of customer requirements, jobs containing that entered job title were founded out and similar jobs to every got job title were recommended to the
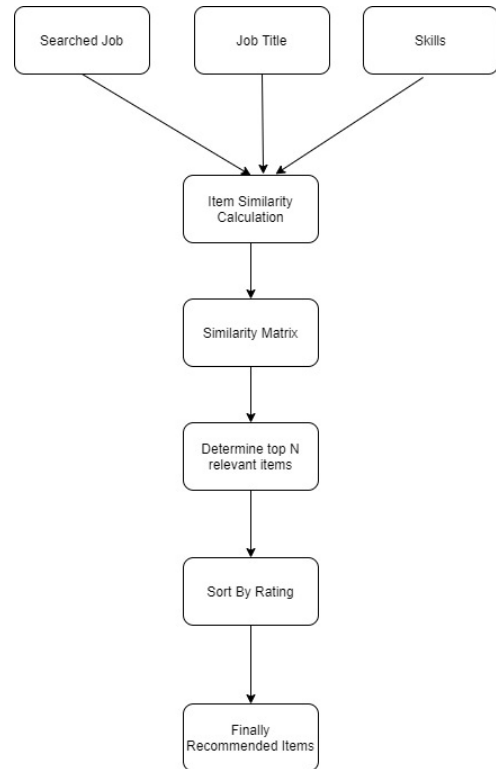
worker so that worker will get a range of different jobs in his entered working area. In this way, different available jobs in the working area of worker will be recommended to him/her.

5.2. Recommending workers to customers

This is the second part of the application that is recommending workers to customers. Here dataset of workers was used to recommend workers to customers according to their work requirements.

Here customer should enter a job title for what he/she requires worker and according to that this algorithm will recommend workers related to that job title to the customer with descending order of workers rating. Dataset of workers contains worker name, job title, skills in his working field, location, and rating given by people to him.

For finding similar workers to entered requirement same algorithm as part one was used and one additional thing in this part is after finding top N similar workers, they were sorted by their rating so we are getting top N top rated similar workers in the field of entered requirement. In this way, top-rated workers in the same working field were recommended to the customer according to their required entered working field. The flow is mentioned in Fig. 2.

Fig. 2 - Flow of algorithm used for recommending workers to the customers

MANASI PURKAR, OMKAR JOSHI, ABHISHEK SALAPE, ANKUSH PATIL, VARAD KULKARNI, DR. PRAVIN FUTANE

6. Results

In this application, we got correctly recommended jobs to workers and workers to customers in the required work field. If the entered job title is present in the dataset, we get precise recommendations related to that job title, and if not directly present we get the recommendation of workers or jobs in various fields related to entering one.

The algorithm used in the dataset was tested on a large dataset of formal jobs and also on the small manually created dataset of informal jobs. On both datasets, we got good results.

Some examples of results we got-

6.1. *Recommending jobs to workers for driver*

```
give_rec('driver')
```

```
    Name        Job title  \
68      Gui Faers       truck driver

88   Nadiya Unstead      truck driver

110    Elna Warrick      truck driver

217  Phillis Dunning     truck driver

496    Cody Gildea       truck driver

102  Delcina Dikles            truck

157   Regen Fliege  school bus truck

57     Maje Dudin              truck


                                      Skills        Loca
tion \
68       driving skill, excellent hearing and vision        Gu
ohua
88       driving skill, excellent hearing and vision      Guantá
namo
110      driving skill, excellent hearing and vision        Alk
maar
217      driving skill, excellent hearing and vision  Rueil-Malma
ison
496      driving skill, excellent hearing and vision        Li
cuan
102      driving skill, excellent hearing and vision    San Ant
onio
157  carefullness,skill, excellent hearing and vision        L
?chi
    57            skill, excellent hearing and vision
        Dallas
```

6.2. *Recommending workers to customers for cook*

```
give_rec('cook')
```

```
        worker name           job title  \
275     Filippa Dengel         fry/saute cook
492   Kimmy Sherington       kitchen manager
94        Zeb Elles         catering manager
438  Charlie Stevenson          broiler cook
438  Charlie Stevenson          broiler cook
263     Giffard Bolin            pantry cook
263     Giffard Bolin            pantry cook
227       Ripley Nix   soup and sauce cook
417    Pietro Parnham             fish cook
399    Natalina Doyle           butcher cook
```

```
                                  skills  rating  \
275             expert in fried sauteed items       4
492                    handles cooking/kitchen       4
94    handles complaint about cooking,watering service    4
438              grilled, boiled,roasted items       4
438              grilled, boiled,roasted items       4
263                         cold food items       3
263                         cold food items       3
227                     all soups and sauce       2
417             expert in fish/lobster cooking       0
399                    prepare cuts of meat       0


                location
275        4 Dayton Place
492        5783 Kings Trail
94    48 Stoughton Junction
438         563 Orin Point
438         563 Orin Point
263     00045 Mayer Avenue
263     00045 Mayer Avenue
227        691 Burrows Road
417       45 Hermina Avenue
    399     807 Mcbride Street
```

7. Conclusion

This paper presents a method to make a recommendation system in python based on content-based filtering technique. This recommendation system giving good recommendation results on datasets for both customers as well as workers. Recommendations are similar to the entered work area by customers or workers and also considering rating part for workers.

This recommendation system is based on content-based filtering using the vector space model and TF-IDF vectorizer. This application will be helpful for both customers and workers for informal jobs in our daily life. Many times, in our daily life we require workers for informal jobs like carpentering, plumbing, and many more.

This recommendation system will be useful for customers in finding people in the required working field and helpful for workers to find jobs. This recommendation system is different from others as it is designed for informal jobs for both customers and workers. There is the future scope for this recommendation system which is explained in the next section.

8. Future Scope

There is a lot of scope of enhancement to the existing recommendation system. An approach naturally solves the candidate and job cold-start problem in the absence of interaction data. The existing system has a job recommendation based on their skills and working field mentioned as a requirement.

As part of the future work, we can make a good UI for this system with some features like the portal can send email notifications to candidates about certain job availabilities, there can be a feedback or review section for the application.

The User Interface can be made more attractive and user friendly through more web-based capabilities. Many features can be added or contextual information can also be added to build more accurate job recommender systems.

Our plan is to extend the system by applying more algorithms and by converting a content-based system to a hybrid system that is by combining more techniques, by using more datasets or item features. As of now, recommendations using similar candidates and jobs based on the working field forms a part of content-based recommendations and the initial results seem promising. Finally, it would be interesting to extend this methodology to other recommendation systems. In the future, our plan is to provide a Career Path Recommender System which will help to find career domain from skills, interest, previous education and short experience.

REFERENCES

B. Patel, P. Desai and U. Panchal. (2017). Methods of recommender system: A review, *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, pp. 1-4, doi: 10.1109/ICIIECS.2017.8275856.

A. Mishra and S. Vishwakarma. (2015). Analysis of TF-IDF Model and its Variant for Document Retrieval, *International Conference on Computational Intelligence and Communication Networks (CICN)*, Jabalpur, pp. 772-776, doi: 10.1109/CICN.2015.157.

Y. Zhang, C. Yang and Z. Niu. (2014). A Research of Job Recommendation System Based on Collaborative Filtering. *Seventh International Symposium on Computational Intelligence and Design*, Hangzhou, pp. 533-538, doi: 10.1109/ISCID.2014.228.

S. Choudhary, S. Koul, S. Mishra, A. Thakur and R. Jain. (2016). Collaborative job prediction based on Naïve Bayes Classifier using python platform. *International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, Bangalore, pp. 302-306, doi: 10.1109/CSITSS.2016.7779375.

S. Ahmed, M. Hasan, M. N. Hoq and M. A. Adnan. (2016). User interaction analysis to recommend suitable jobs in career-oriented social networking sites. *International Conference on Data and Software Engineering (ICoDSE)*, Denpasar, pp. 1-6, doi: 10.1109/ICODSE.2016.7936143.

S. Dong, Z. Lei, P. Zhou, K. Bian and G. Liu. (2017). Job and Candidate Recommendation with Big Data Support: A Contextual Online Learning Approach. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, pp. 1-7, doi: 10.1109/GLOCOM.2017.8255006.

B. R. Cami, H. Hassanpour and H. Mashayekhi. (2017). A content-based movie recommender system based on temporal user preferences. *3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, Shahrood, pp. 121-125, doi: 10.1109/ICSPIS.2017.8311601.

T. V. Yadalam, V. M. Gowda, V. S. Kumar, D. Girish and N. M. (2020). Career Recommendation Systems using Content based Filtering. *5th International Conference on Communication and Electronics Systems (ICCES)*, COIMBATORE, India, pp. 660-665, doi: 10.1109/ICCES48766.2020.9137992.

J. Wintrode, G. Sell, A. Jansen, M. Fox, D. Garcia-Romero and A. McCree. (2015). Content-based recommender systems for spoken documents. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, pp. 5201-5205, doi: 10.1109/ICASSP.2015.7178963.