**APPLYING RECOMMENDATIONS IN JOB MARKETING**

Individual Project Report

Abhinav Balasubramanian

<[abhinav.balasubramanian@sjsu.edu](mailto:abhinav.balasubramanian@sjsu.edu)>

Student Id: 013853281

Department of Computer Engineering

San Jose State University

San Jose, CA

Applying Recommendations in Job Marketing

Abhinav Balasubramanian

**Abstract**— The search of jobs online has been blooming ever since the past decade. With the rise of professional social platforms like LinkedIn, Indeed etc., the search of jobs through professional social media has become inevitable. Most of the people seek these career oriented social platforms either to search for jobs or connect with users of similar educational background or interests. Thus, the need of recommendation systems in this field is increasing day by day. The different ways of implementing the recommendation engines in job marketing has been elaborated in this article with implementation. Additionally, the recommendation engines can be tuned to propose efficient job transition. Thus, this article not only aims at developing a typical recommendation engine but also tries at enhancing its functionality by implementing job transition recommendation.

**Index Terms**— Recommendation Systems, Job Transition, Content-based filtering, Collaborative filtering

# 1 Introduction

M

ost of the job opportunities these days are obtained with the help of career oriented social platforms. The rise of such platforms in the last decade has helped a lot of people with similar educational or professional backgrounds and interests connect to each other. This has also helped the recruiters in identifying suitable candidate with relevant experience to their specifications. However, the recommendation systems in this kind of social platforms are still under constant development. This makes it more important to research more on the recommendation systems in job marketing. A proper and efficient recommendation system would facilitate the recruiters and job seekers identify each other more accurately and precisely. This would help them in saving a lot of time. If the similarity between the users is calculated, it could be used by the employers to identify candidates who are similar to a particular candidate found suitable for a job position. The user similarity could also be used by individuals to identify users with similar educational/professional background to them. Similarly computing the similarity between jobs helps users in identifying the most similar jobs to which their qualification would be suitable. Collaborative filtering could be also applied upon the similar users identified such that if a particular similar user applies for a particular job posting, that job posting is recommended to the considered user. In the modern days, most of the people have career ambitions and thrives hard in achieving them. The recommender system in job marketing could be modeled to propose job transitions to the users. This would actually lead to the usage of recommendation system to its full potential.

# 2 Literature Study

An extensive study of literature with respect to recommender systems in job marketing is needed before proceeding towards the implementation. The survey [1] provides an overview about the various job recommendation systems in existence and how the common recommendation terminologies have been applied over the various recommendation systems. A study [2] developed a job transition model for job recommendation system. It also observed that the job transition model was capable of performing much better than the basic job recommendation system. It also employed the job transition model developed with cosine similarity to suggest more similar jobs to a specific user. Thus, these literatures help in getting a better understanding about recommendation systems in job marketing.

# 3 Data Extraction

To build a recommender system for job marketing that could function at its full potential; we would require both user and job related data. The best sites for scraping user related data is LinkedIn and the best site for scraping job related data is Indeed. The data from these are scraped with the help of selenium and scrapy packages in python.

The user and job related data extracted from LinkedIn and Indeed are given below:

TABLE 1

COLUMNS USED AND RELATED INFORMATION

|  |  |
| --- | --- |
| **User Related Columns** | **Job Related Columns** |
| User Name | Job Name |
| Current Position | Company |
| Past Positions | Company Description |
| Past Companies | Education |
| Year of experience | Experience |
| Summary about the user | Responsibility |
| Summary about previous job roles | Requirement |

Data collected from both LinkedIn and Indeed belongs to either one of the five technical domains listed below:

* Python developers and Data Scientist
* Cloud Developer
* Database Developer
* Full-stack Developer

However, there are a lots of problems associated with scraping data from LinkedIn. LinkedIn needs a user to be logged in to it to view other user profiles. It also blocks a user when increased activity is recorded. On the top of it, the website design of the page varies when it is viewed after logging in. So we needed to write a code which was capable of handling all this issues.

For Indeed, the URLs of various job posting were needed to be obtained from a search page which was accessed through the web scraping tools. Then those URLs needs to be accessed and necessary data for each job posting must be scraped from them.

# 3.1 Data Scraping for LinkedIn

For LinkedIn, two python files were used, namely the script.py and paramaters,py. The script.py contained the code for extracting user data from LinkedIn. The paramaters.py contained the information needed for the proper execution of the script.py like email id and password of LinkedIn etc. The codes used and the output dataset images are shown below:

**Script.py:**

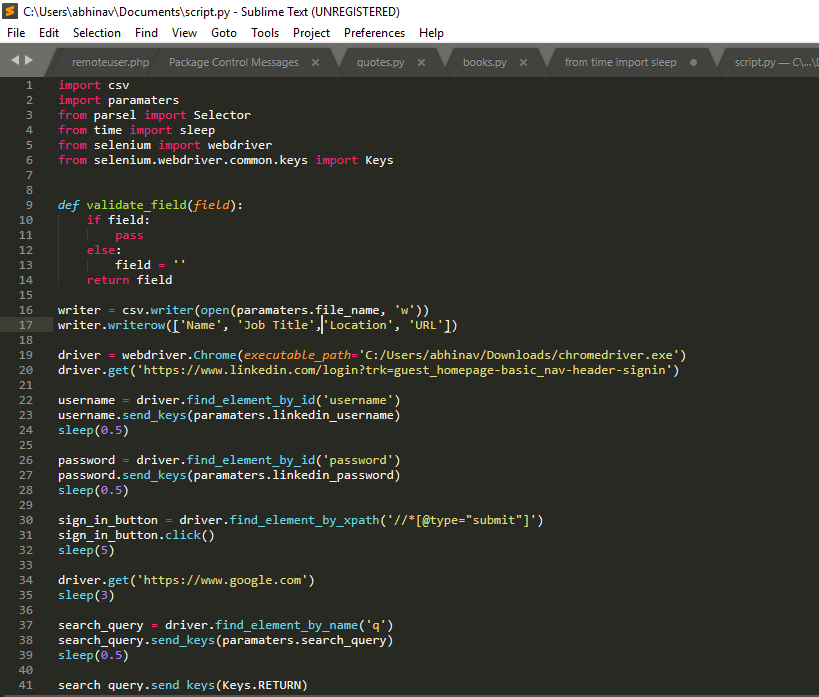


Fig 3.1 Data scraping for LinkedIn part one

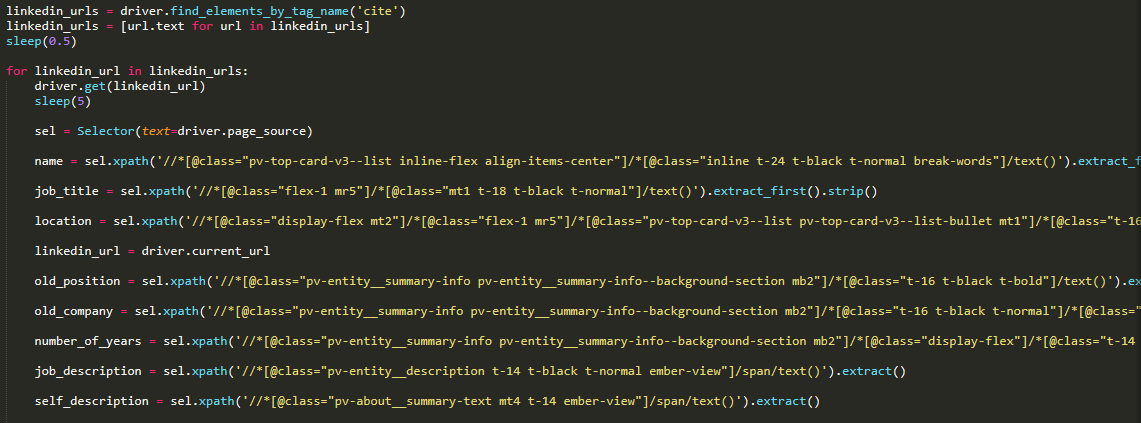
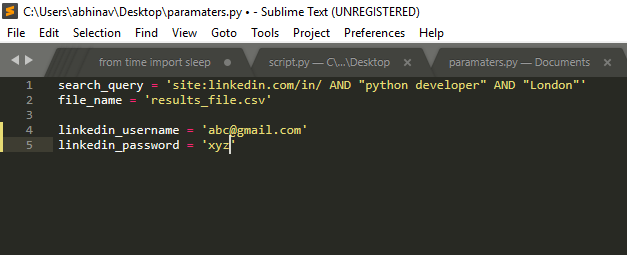
Fig Fig.3.2 Data scraping for LinkedIn part two



Fig.3.3 Data scraping for LinkedIn part three

**Parameters.py:**

Fig.3.4 Data scraping for LinkedIn part four

**Output Dataset for LinkedIn**:

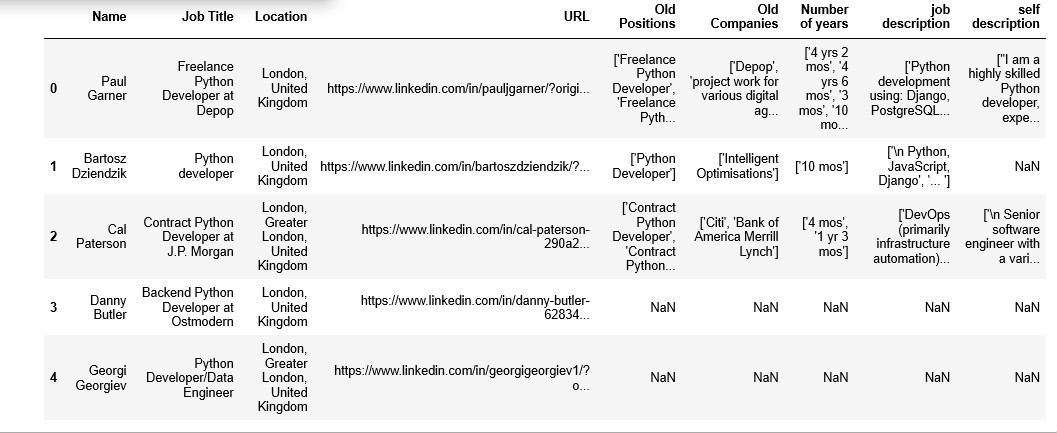
****

Fig.3.5 LinkedIn Dataset Output

# 3.2 Data Scraping for Indeed:

For indeed, a spider is created for the selected search page and inside that the logic stated above is coded. With the help of scrapy crawl command, all the information needed is fetched as dictionaries and is loaded in a csv file. The codes used and the output dataset images are shown below:

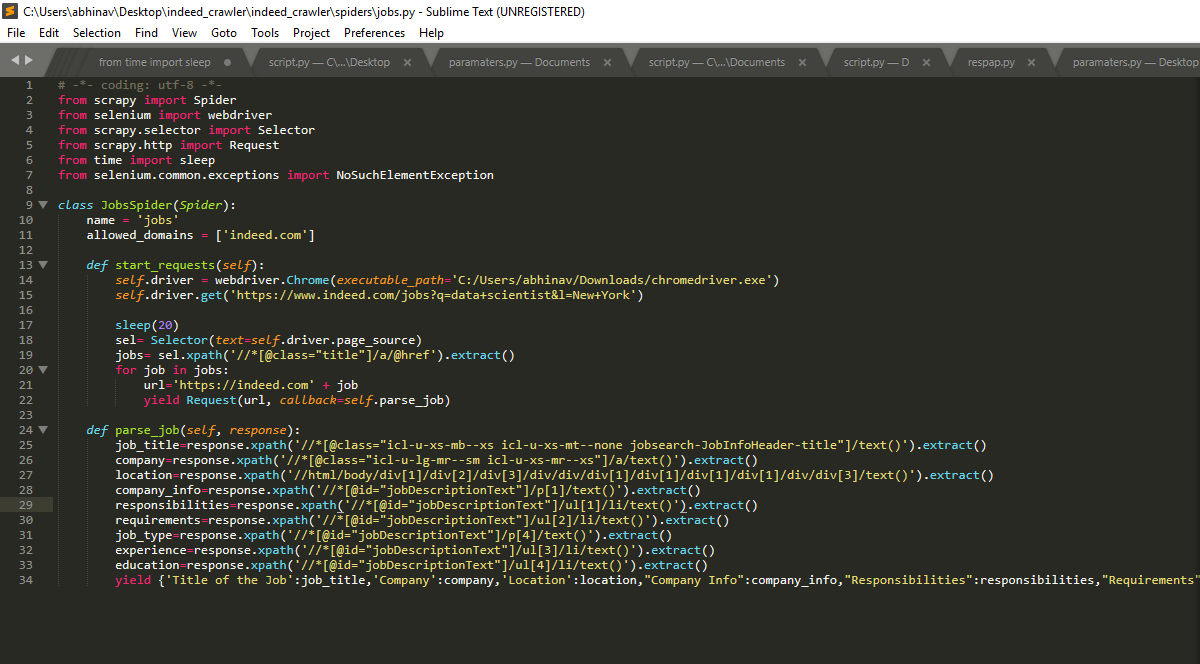


Fig. 3.6 Indeed Data Scraping

The columns which talks about the particular job responsibilities, requirements, experience is also extracted so that it could be text mined to obtain the important words and compared with the LinkedIn user data to build the job marketing based recommender system.

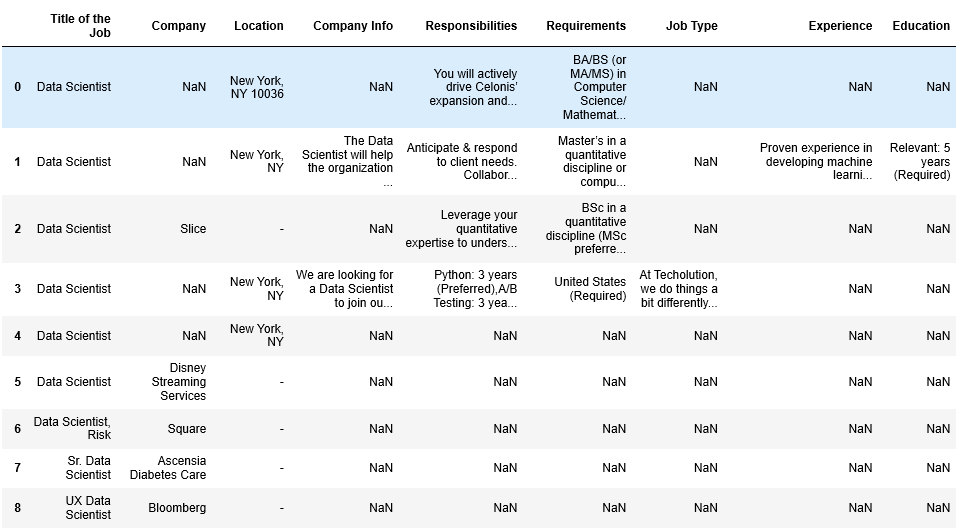


Fig. 3.7 Indeed Output Dataset

# 4 Data Pre-processing

Since the data collected from LinkedIn is in unprocessed form, we need to process the raw data obtained to build models with it. Two main pre-processing done on the LinkedIn data is

1. Splitting the job experiences as list within a single column into multiple columns with single values.

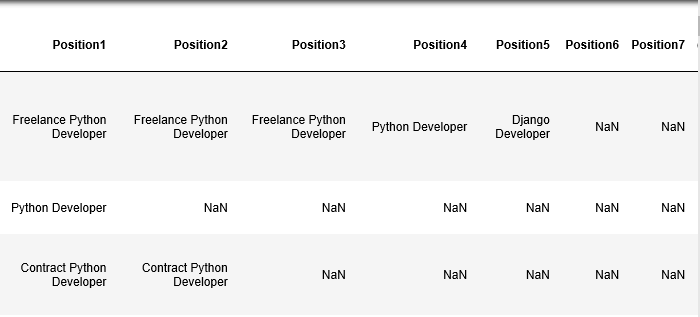


Fig.4.1. Splitting Previous Job Experience in Single Column to Multiple Columns

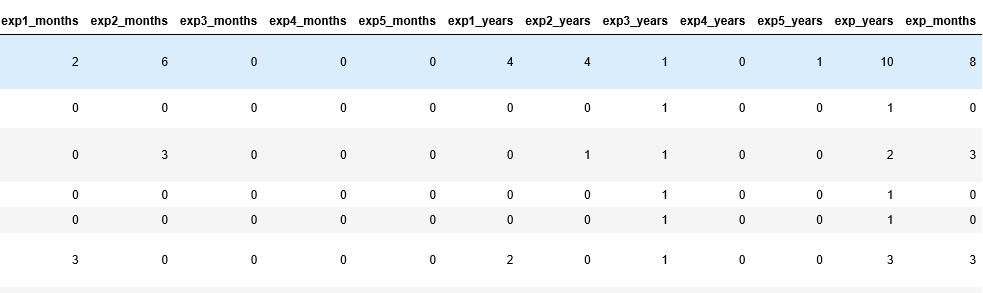
1. Splitting the years of experience as a list in single column to multiple columns and processing it to obtain the total years of experience.

Fig.4.2. Processing the years of experience column to obtain meaningful other columns

Apart from this, most of the LinkedIn and Indeed data have null values in description field which is necessary to perform content-based filtering. In those situations, the job title is substituted to perform the analysis.

# 5 CONTENT-BASED RECOMMENDATION

Content-based recommendation could be applied to both user and job dataset where a set of keywords is derived from:

1. Self and previous job descriptions for users
2. Experience, education, responsibility and requirement for jobs

Term Frequency-Inverse Document Frequency is applied on the set of keywords obtained to obtain the TF-IDF scores for users and job considered.

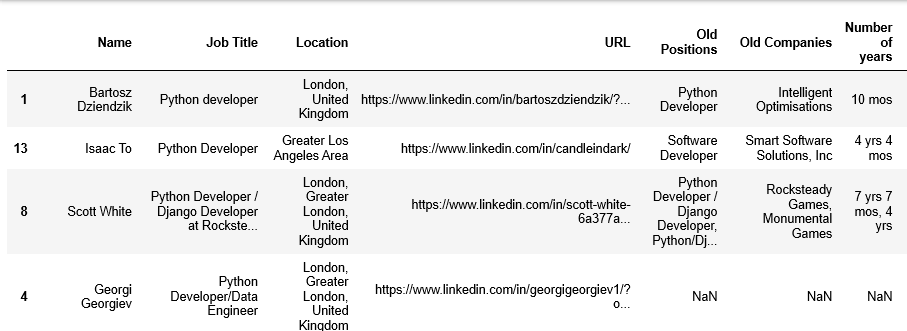
# Cosine similarity, Manhattan Distance and Euclidean Distance are used as similarity measures to find the similarity between users/jobs based on content.

# The output for users and jobs content-based recommendation for different similarity measures is shown below.

# 5.1 User Content-Based Recommendation

# 1. Cosine Similarity:

# Fig.5.1. Cosine Similarity based User recommendation System

1. **Euclidean Distance:**

# Fig.5.2. Euclidean Distance based User recommendation System

1. **Manhattan Distance:**

Fig.5.3. Manhattan Distance based User recommendation System

**5.2 Job Content-Based Recommendation**

1. Cosine Similarity:

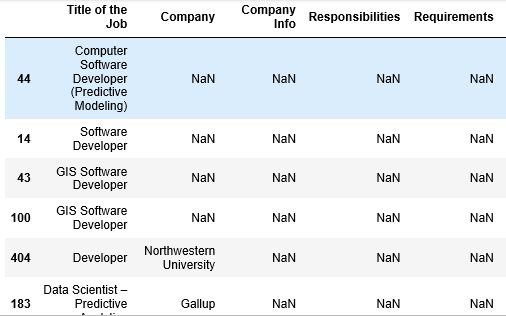
# Fig.5.4. Cosine Similarity based Job recommendation System

1. **Euclidean Distance:**

# 

# Fig.5.5. Euclidean Distance based Job recommendation System

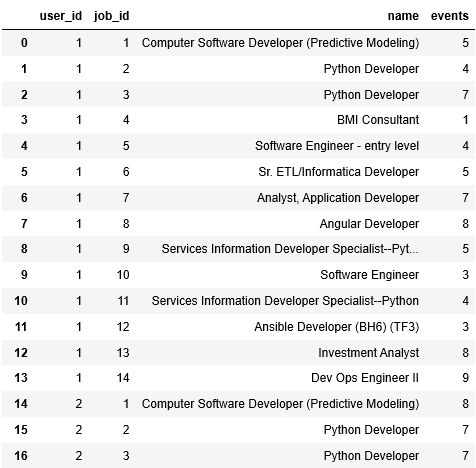
1. **Manhattan Distance:**



# Fig.5.6. Manhattan Distance based Job recommendation System

**6 Collaborative Filtering**

In case of collaborative filtering, we try to recommend those job postings that were applied by similar users to a particular user. Implicit ratings are actually needed to perform this analysis. In this case, we consider the number of application forms filled by similar users as the metric to perform collaborative filtering. Since we don’t have those columns, those columns are simulated with the help of numpy random number generation capability. The dataset used for collaborative filtering looks like:



# Fig.6.1. Collaborative Filtering Dataset

The column events provides the number of application form filled by the similar users and user\_id and job\_id denotes the ids assigned to jobs and users. The collaborative filtering is done on a very small subset of user and

job population.

Initially a sparse matrix is built up on the dataset considered and the sparsity of the matrix is analyzed. In this case, the sparsity was found to be 61.8%.

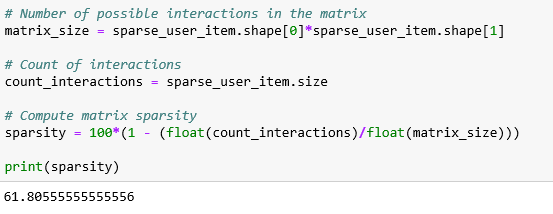


Fig.6.2. Sparsity value for the collaborative filtering dataset

Then Alternating Least Squares model with fixed parameters is implemented on the data and it is used to recommend similar jobs to the users on the basis of collaborative filtering.

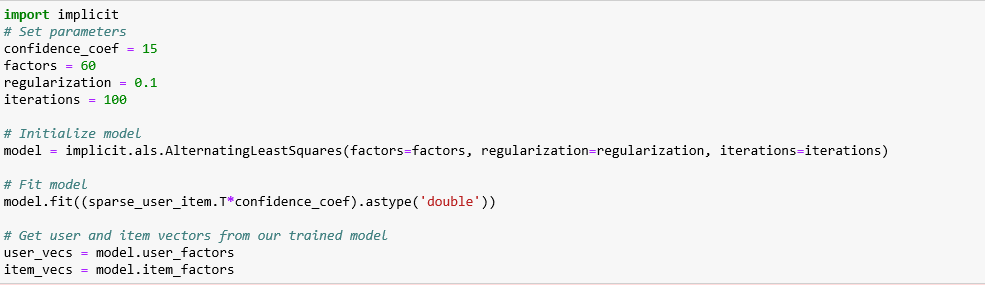


Fig.6.3. Alternating Least Square Model

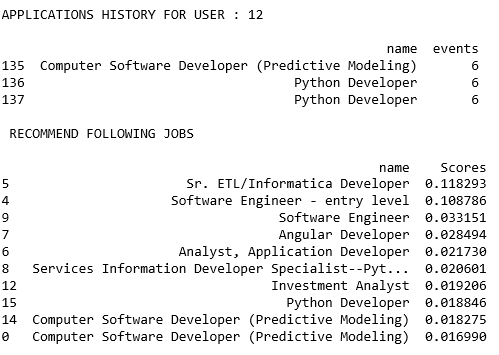


Fig.6.4. Output of Collaborative Filtering for user 12

**7 Job Transition Recommendation System**

For job transition recommendation system, we would need a dataset that contains a set of previous and current jobs along with time period. All this information is available in the LinkedIn dataset from which the data is extracted in the form needed.

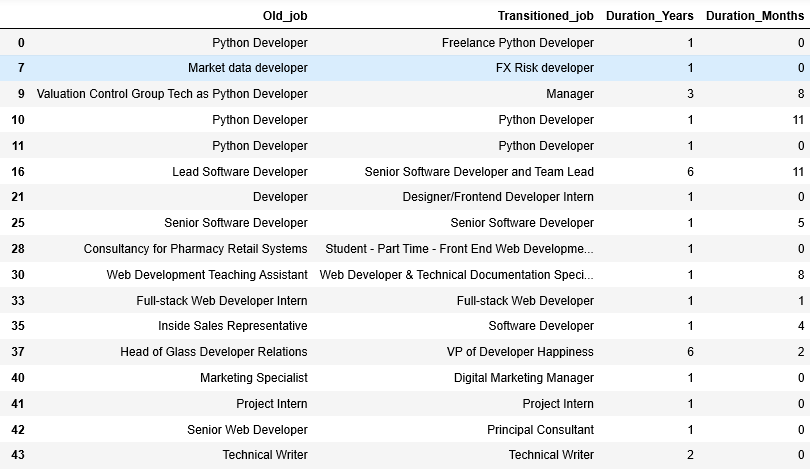


Fig.7.1 Dataset for Job Transition Recommender System

Once the data is obtained, we observe that for some values both the year and month of duration is 0 which is not a valid entry. This is handled by replacing those entries with 6 months experience as each employee would sustain in an organization for at least 6 months.

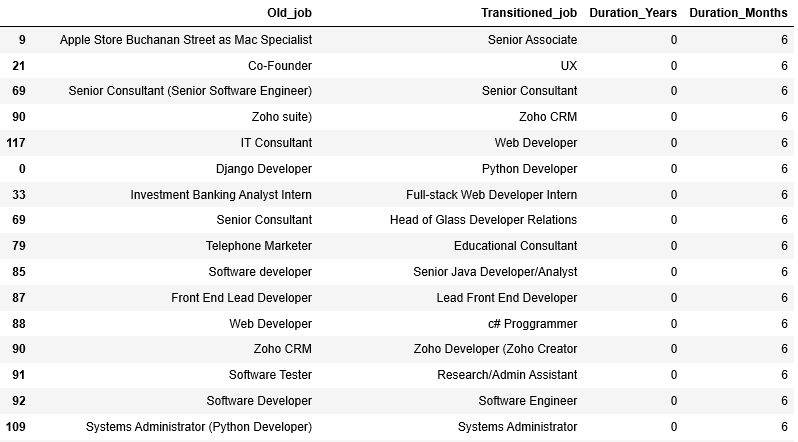


Fig.7.2 Handling the improper data

Once the improper data is handled, the job which contains the keyword searched by user is taken and sorted in ascending order of employment duration as a job that would lead to a desired job at lower time is preferred by most of the users generally.



Fig.7.3. Sample output when the user types ‘Python’ in search

**8 Conclusion and Future Works**

Thus, in this project, a simple job recommendation system is built along with job transition recommendation system. The lack of readily available data and data quality issue such as sparsity, null values decrease the performance of the recommendation system implemented. Thus, building a recommendation system after resolving all the above stated issues could end up giving even better results.

**References**

1. Z. Siting, H. Wenxing, Z. Ning and Y. Fan, "Job recommender systems: A survey," 2012 7th International Conference on Computer Science & Education (ICCSE), Melbourne, VIC, 2012, pp. 920-924.  
   doi: 10.1109/ICCSE.2012.6295216  
   keywords: {collaborative filtering;personal information systems;recommender systems;recruitment;job recommender systems;personalized recommender system;information overload;job recruiting domain;user profiles;Recommender systems;Collaboration;Feature extraction;Resumes;Data mining;Hidden Markov models;job matching;recommendation technology;job recommender system},  
   URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6295216&isnumber=6295013>
2. Heap B., Krzywicki A., Wobcke W., Bain M., Compton P. (2014) Combining Career Progression and Profile Matching in a Job Recommender System. In: Pham DN., Park SB. (eds) PRICAI 2014: Trends in Artificial Intelligence. PRICAI 2014. Lecture Notes in Computer Science, vol 8862. Springer, Cha