|  |
| --- |
| **FACULTY OF APPLIED SCIENCES & TECHNOLOGY**  **HUMBER NORTH CAMPUS** |
| **ITC-5402-0NB CAPSTONE PROJECT**  **GROUP PROJECT DOCUMENTATION** |
| TITLE: Job-O Recommendation Engine |
| Team Members: (Select one team member to be the main contact. Mark with ‘\*’) |
| Project Sponsors: Aditya Subramaniam Yasmin Bakhtiari Nahal Kazemian |
| Mentor/Supervisor: Prof. Orren Johnson |
| |  |  | | --- | --- | | Name | Humber Number & Email | | Sundeep Kaur | [sundeepkaur325@gmail.com](mailto:Sundeep-sundeepkaur325@gmail.com)  N01261642 | | \*Alisha Sharma | [xs.alisha@gmail.com](mailto:xs.alisha@gmail.com)  N01293681 | | Treesa Chacko | [treezachacko91@gmail.com](mailto:Tressa-treezachacko91@gmail.com)  N01369401 | | Raminderjit Kaur | [dhillonraminder46@gmail.com](mailto:Raminder-dhillonraminder46@gmail.com)  N01293146 | | Komalpreet Kaur | [kaurk7978@gmail.com](mailto:kaurk7978@gmail.com)  N01288629 | |
| Submission Date: April 17, 2020 |

**Contents**

1. Introduction 3

1.1. Approaches 3

2. Scope Statement 4

3. Group Highlights 6

4. Step by Step working of Model 7

4.1 Preparing Environment 7

4.2 Data Collection 7

4.3 Data Storage 8

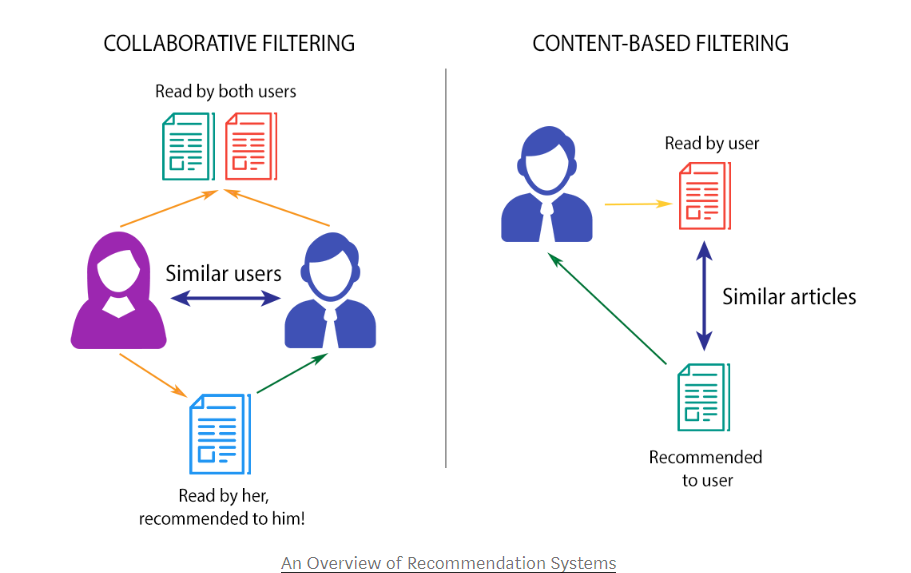
4.4 Evaluation 13

**1. Introduction:**

Most internet products we use today are powered by recommender systems. YouTube, Netflix, Amazon, Pinterest, and long list of other internet products all rely on recommender systems to filter millions of contents and make personalized recommendations to their users. Recommender systems are well-studied and proven to provide tremendous values to internet businesses and their consumers. In fact, I was shock at the news that Netflix awarded a $1 million prize to a developer team in 2009, for an algorithm that increased the accuracy of the company’s recommendation system by 10%.

**1.1 Approaches:**

Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user. It helps a user to make a decision. In general, there are three types of recommender system:

****

• Collaborative recommender system is a system that produces its result based on past ratings of users with similar preferences

• Content based recommender system is a system that produces its result based on the similarity of the content of the documents or items.

• Knowledge based recommender system is a system that produces its result based on additional and means–end knowledge.

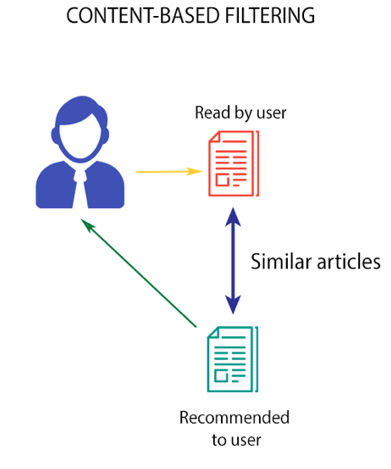
**2. Scope Statement**

This project intends to generate job recommendations for user based on certain features. User has got skills, rate for skills, job title, age, longitude, latitude used for calculating user’s distance from job location etc., and in the same way job posts have got skills, skills rate, job title, longitude, latitude, job total salary, job experience, job total hiring etc., than can be used against user parameters to generate meaningful job recommendations for user.

In this project the main aim is to create recommendation engine that can predict jobs for users using machine learning concepts. For this project machine learning - content based filtering approach has been used to generate recommendations. In content-based cosine similarity algorithm has been used.

The Content-Based Recommender relies on the similarity of the items being recommended. The basic idea is that if you like an item, then you will also like a “similar” item. It generally works well when it’s easy to determine the context/properties of each item.

A content-based recommender works with data that the user provides. Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

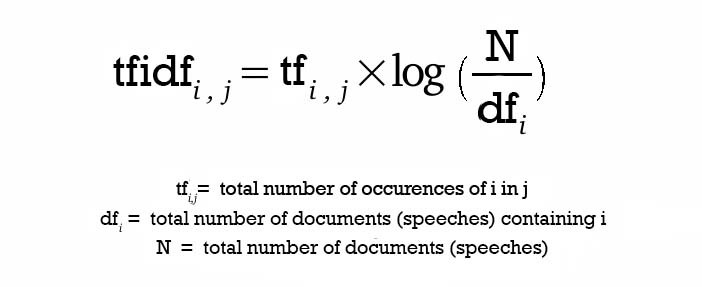


## The Math

The concepts of **Term Frequency (TF)** and **Inverse Document Frequency (IDF)** are used in content-based filtering mechanisms (such as a content-based recommender). They are used to determine the relative importance of a document or item.

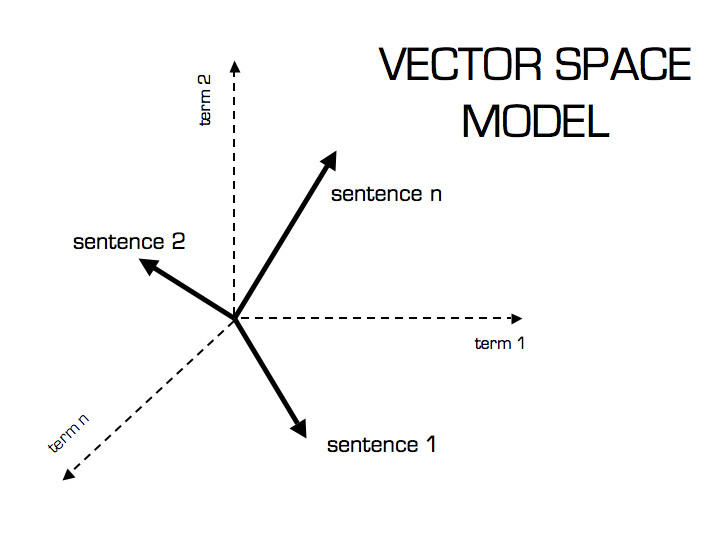
TF is simply the frequency of a word in a document. IDF is the inverse of the document frequency among the whole corpus of documents. TF-IDF is used mainly because of two reasons: Suppose we search for “**the results of latest European Soccer games**” on Google. It is certain that “**the**” will occur more frequently than “**soccer games**” but the relative importance of **soccer games** is higher than the search query point of view. In such cases, TF-IDF weighting negates the effect of high frequency words in determining the importance of an item (document).

Below is the equation to calculate the TF-IDF score:



TF-IDF Equation

After calculating TF-IDF scores, how do we determine which items are closer to each other, rather closer to the user profile? This is accomplished using the **Vector Space Model** which computes the proximity based on the angle between the vectors. In this model, each item is stored as a vector of its attributes (which are also vectors) in an **n-dimensional space** and the angles between the vectors are calculated to **determine the similarity between the vectors**. Next, the user profile vectors are also created based on his actions on previous attributes of items and the similarity between an item and a user is also determined in a similar way.



Vector Space Model

Sentence 2 is more likely to be using Term 2 than using Term 1. Vice-versa for Sentence 1. The method of calculating this relative measure is calculated by taking the cosine of the angle between the sentences and the terms. The ultimate reason behind using cosine is that the **value of cosine will increase with decreasing value of the angle** between which signifies more similarity. The vectors are length normalized after which they become vectors of length 1 and then the cosine calculation is simply the sum-product of vectors.

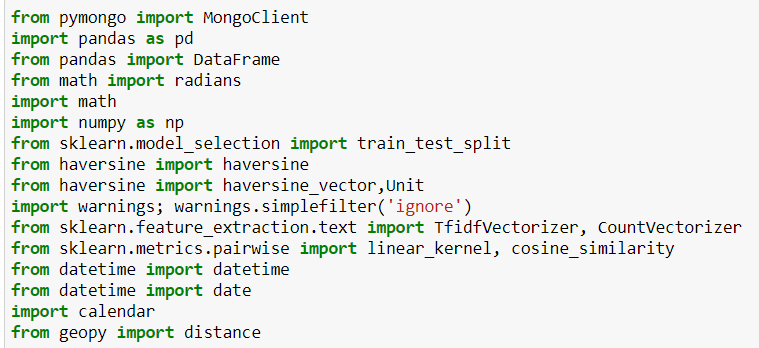
**3. Group Highlights**

**Accomplishment #1 Identification of relative and meaningful recommender engine and algorithm for our test case**

After having done through research on various machine learning recommender systems and algorithms, team identified the most relevant and meaningful recommender engine and algorithm for our test case. As in our test case it is easy to identify the context and properties of each user and also for jobs, so team decided to go with TF-IDF content-based filtering recommender engine and with cosine similarity algorithm.

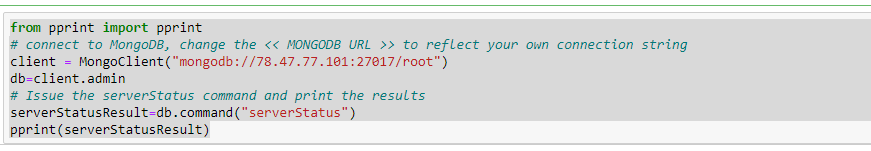
**4. Step by Step working of Model:**

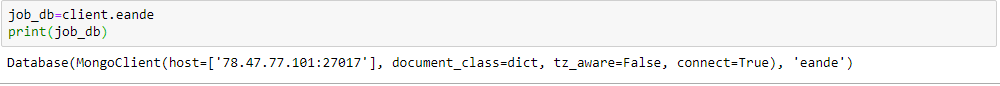
**4.1 Preparing Environment:**

****

**4.2 Data Collection:** This is the first and most crucial step for building a recommendation engine. The data can be collected by two means: explicitly and implicitly. Explicit data was provided by JobO.

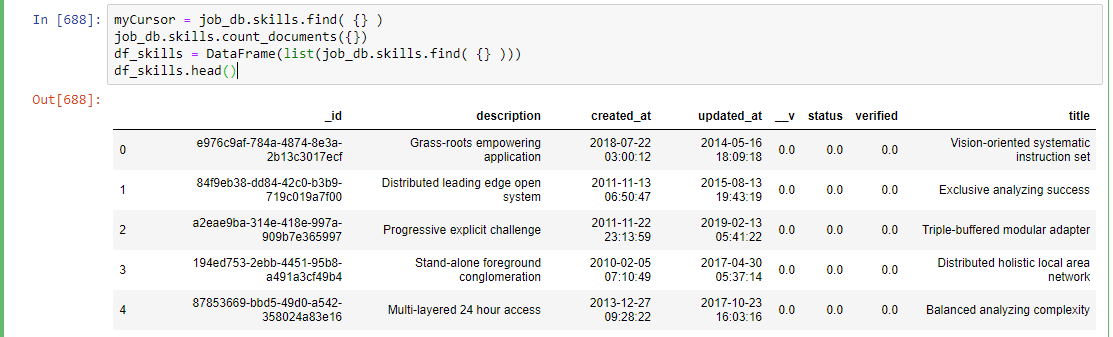
Successfully loaded and processed the Job-O data from their database using robo3T interface for accessing MongoDB database of Job-O





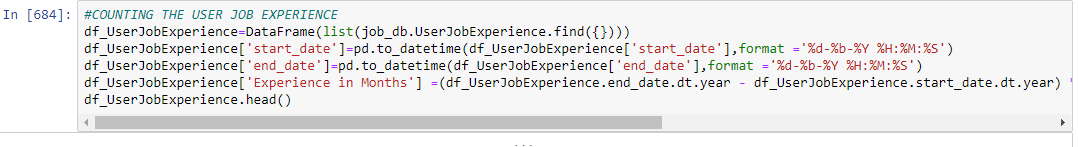
**4.3 Data Storage:**

1. Loaded the data from the skills table from database to data frame in order to get information about the skills of users.



As above we have loaded data from the below tables from database to data frame:

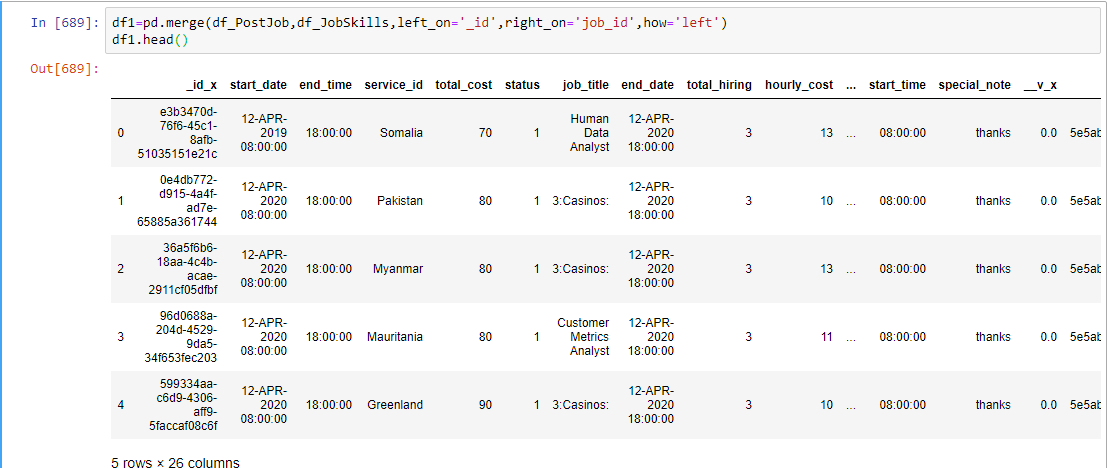
* PostJob: It has information about the job posted by companies
* JobSkills: It has information about the skills required for job
* UserJobExperience: It has started date of the job and end date of the job from which we calculated the Total experience of job as below:



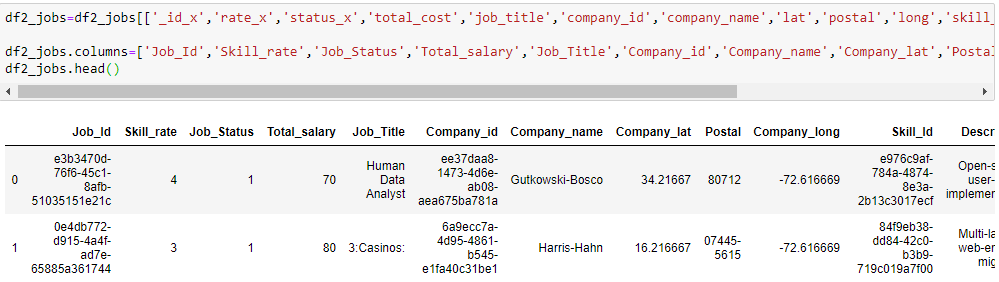
* UserSkills: It has information about the skills of user
* Company: It has information about the company like its name, latitude and longitude which will be required
* Users: It has all the information about the user.

1. Joining the data frames:

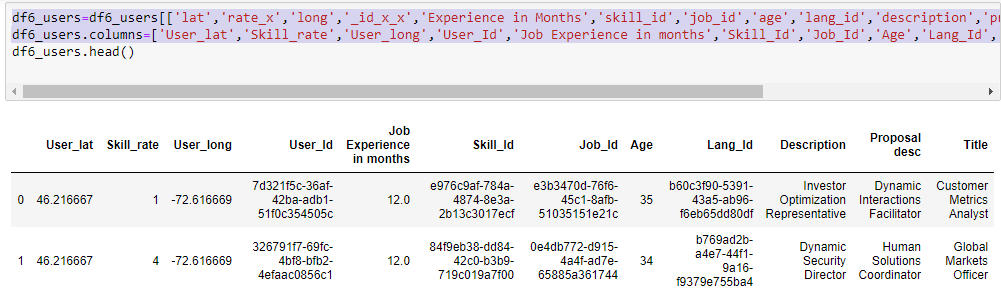
* Here we are joining two data frames df\_PostJob and df\_JobSkills on job\_id so that we can get Job posted by employers and skills required for a job in one data frame i.e. df1 in our model.



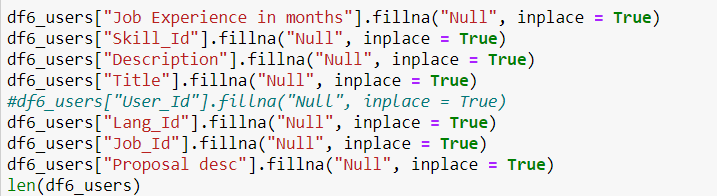
* Merging of df1 with df\_company on company\_id in order to get the all information of jobs posted ,skills and company in one data frame i.e. df2\_jobs in model, which is used as final data frame for jobs information in the model, holding below captured columns:



* Merging of df\_users on df\_Userskills on user\_id in order to get the data related to user skills and users in one data frame i.e. df3
* Merging of df\_UserLang and df\_UserJobExperience on user\_id giving data frame df4.
* Merge df4 and df3 giving df5\_users data frame which is further merged with df\_JobProposal on user\_id giving data frame df6\_users which in total have all the information about users.
* Calculated age from date of birth column from df6\_users
* For final data frame of users, we have selected below columns:

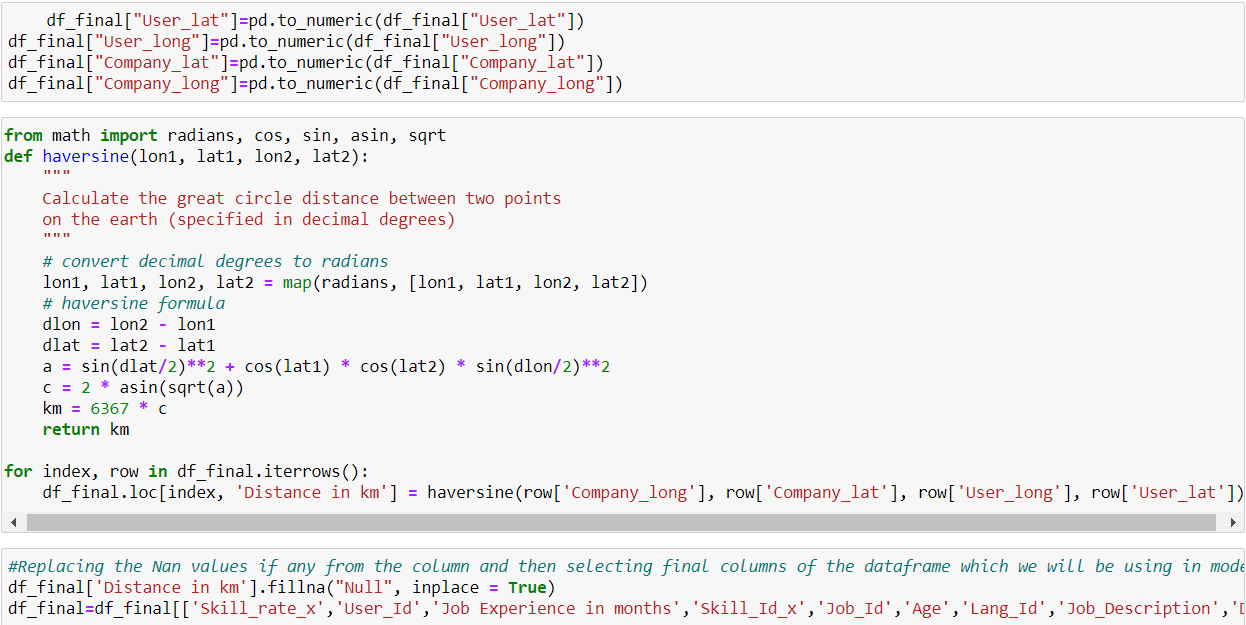


* After creating final data frame, we checked if there is any Nan value is there and than have replaced all the ‘Nan’ values from the data frame.

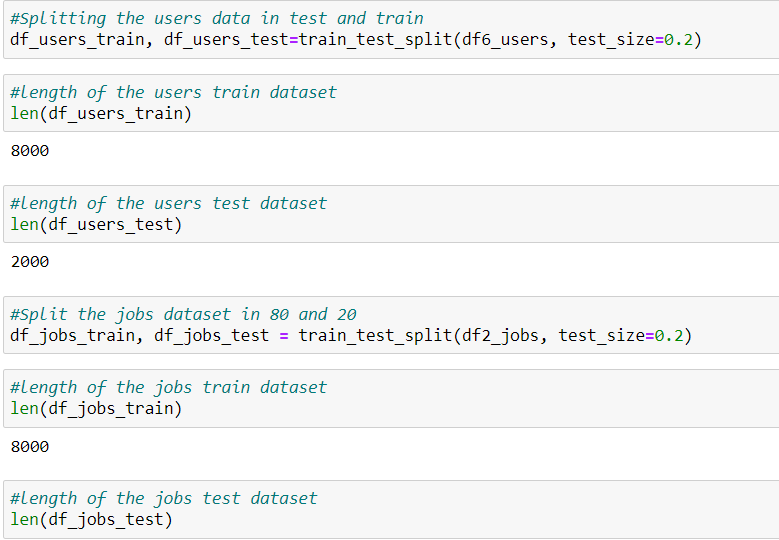


* We have further merged df2\_jobs and df6\_users on job\_id in df\_final data frame order to use it in the final output of a model.

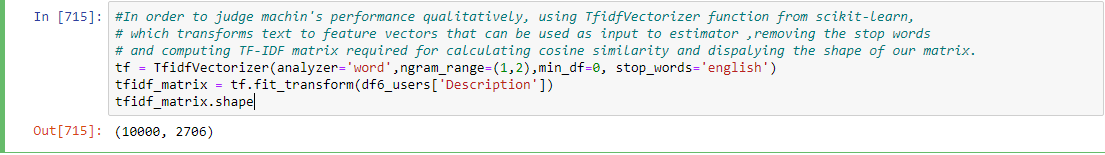


* Calculation of distance from haversine formula in which firstly we changed the latitude and longitude of user and company into numeric and then applied the formula. 

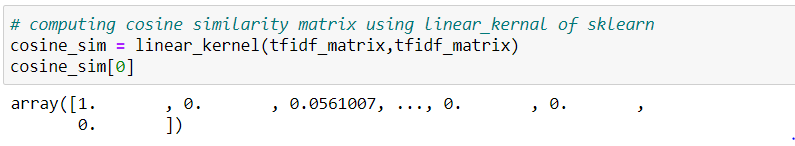
1. Split the users and jobs data in test and train in 20 and 80 ratios respectively.



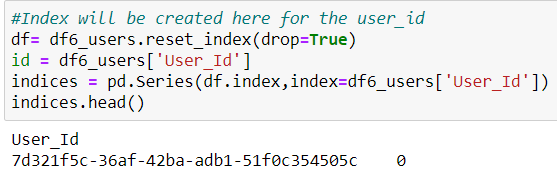
1. Taking input from title, description and proposal desc in df6\_users[descriptions] so that users can be matched on the basis of this value.
2. In order to judge machine’s performance qualitatively, using TfidfVectorizer function from scikit-learn, which transforms text to feature vectors that can be used as input to estimator ,removing the stop words and computing TF-IDF matrix required for calculating cosine similarity and displaying the shape of our matrix.



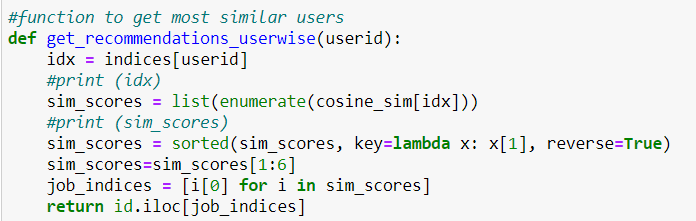
1. We computed cosine similarity matrix using linear\_kernal of sklearn.



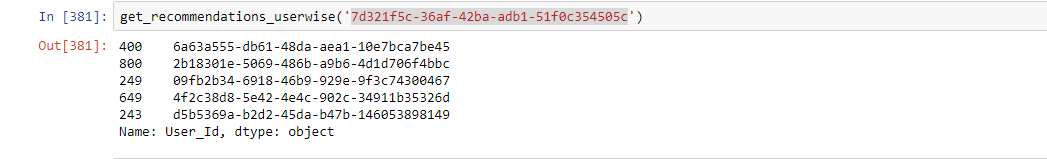
1. Index for the for the user\_id column of f6\_users is created here which will be further used in function to get most similar users.



1. We have created a function “get\_recommendations\_userwise” now which will recommend most similar users in which firstly it will take index of user\_id and will generate similar score of the indices from the cosine\_sim matrix which we have calculated and we have filtered the sim\_scores up to 6 which means it will give s top 5 similar users .



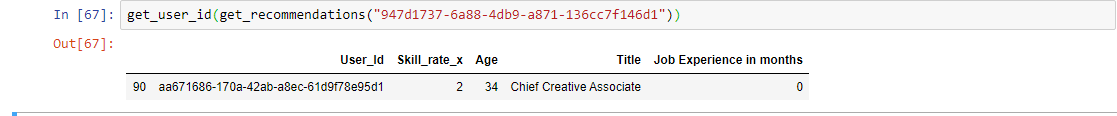
1. Now, we are calling our function in order to get most similar users in the output:



1. We have created a function “def get\_job\_id(usrid\_list)” which will recommend jobs to user on basis of skill\_id and filtering the output with distance and getting output as:



Similarly, we have created model that recommend users to employers. Similar approach has been used.



**Analysis for users:**

We have analyzed results for different user ids. As we can see below 10 users are getting respective recommendations as per the titles.

|  |  |  |
| --- | --- | --- |
| **User Id** | **Title** | **Recommendations** |
| **123c163c-3543-40ec-b907-b1e1bd39dfad** | **Corporate Metrics Analyst** | **Corporate Metrics Analyst** |
|  | **Corporate Research Consultant** |
| **b011a125-eab9-45e2-b80d-a2fe226c22e1** | **Forward Interactions Specialist** | **Product Web Representative** |
|  | **Human Marketing Director** |
|  | **Legacy Solutions Architect** |
| **cffac27b-b323-4530-a2bc-6b3110e41b54** | **Legacy Solutions Architect** | **Legacy Solutions Architect** |
| **District Directives Manager** |
| **Legacy Operations Technician** |
| **2b18301e-5069-486b-a9b6-4d1d706f4bbc** | **Customer Metrics Analyst** | **Customer Infrastructure Analyst** |
| **Corporate Metrics Analyst** |
| **c96b5519-00bf-44d6-adee-03c824fb568c** | **Internal Solutions Agent** | **Lead Response Strategist** |
| **Product Interactions Assistant** |
| **46dc73c9-1664-48c3-830c-0d434fb72125** | **Forward Intranet Manager** | **Global Configuration Associate** |
| **Principal Creative Designer** |
| **International Security Associate** |
| **e27e9023-c365-4469-8f72-3121efc2c12e** | **Customer Accountability Agent** | **Internal Paradigm Administrato** |
| **District Tactics Assistant** |
| **6d9daf03-6e84-4919-a7c5-53da5adbc4c2** | **Direct Quality Architect** | **Direct Quality Analyst** |
|  | **Direct Quality Designer** |
|  |  | **Central Accountability Administrator** |
| **feae463a-1773-40f5-bccf-1fac55da81ce** | **National Division Orchestrator** | **National Division Orchestrator** |
|  | **Dynamic Mobility Strategis** |
| **d3c45d80-d634-4119-8264-83569f1191fe** | **Global Configuration Associate** | **Global Configuration Associate** |

**Analysis for jobs:**

We have analyzed results for different job ids. As we can see below 10 job\_ids are getting respective recommendations of user as per their job\_description

|  |  |  |
| --- | --- | --- |
| **JOB\_ID** | **Job\_Description** | **RECOMMENDATIONS** |
| bcc06f99-bd0c-43ca-95ca-37f8c42d7e95 | Ameliorated secondary frame | Ameliorated secondary frame |
| 0e8299b9-d430-4c87-96d9-5576866f2196 | Diverse even-keeled portal | Diverse even-keeled portal |
| fe123f71-2c5a-425f-bd60-dfade202a452 | Centralized context-sensitive projection | Centralized context-sensitive projection |
| 502e883c-0ab9-491a-bb99-ad94a85caa4d | Up-sized neutral time-frame | Balanced mobile process improvement |
| 8db830ea-dfb5-4540-9a24-49695f603d92 | Progressive neutral capacity | Automated responsive moratorium |
| 9cc5255e-4d0f-4d44-b164-b31f462abd9f | Devolved analyzing alliance | Customizable grid-enabled function |
| 2c24e4c2-50a3-4601-9754-397d0a3e7a9f | Polarised executive capability | Centralized intermediate utilisation |
| 706e0489-6783-4492-b78b-90de3dff284f | Enterprise-wide logistical circuit | Reactive context-sensitive toolset |
| c76be6a4-0761-4216-a5fa-52446e28e62b | Multi-lateral needs-based conglomeration | Distributed intangible hierarchy |