IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

JOB RECOMMENDATION SYSTEM BASED ON SKILL SETS

G.Mahalakshmi[1], A.Arun Kumar[2], , B.Senthilnayaki[1], J.Duraimurugan[1]

[1]Teaching Fellow, Department of Information Science and Technology, Anna University Chennai, India
[2]MCA student, Department of Information Science and Technology, Anna University Chennai, India

Abstract Machine learning is a sub-field of data science that concentrates on designing algorithms that can learn from and make predictions on the data. Presently recommendation frameworks are utilized to take care of the issue of the overwhelming amount of information in every domain and enable the clients to concentrate on information that is significant to their area of interest. One domain where such recommender systems can play a significant role to help college graduates to fulfill their dreams by recommending a job based on their skill set. Currently, there are plenty of websites that provide heaps of information regarding employment opportunities, but this task is extremely tedious for students as they need to go through large amounts of information to find the ideal job. And many students are not aware of which job is suitable for them. Nowadays, the IT fields are in a boom. Many engineering students are learning some technical skills by doing some courses but they don't know which skill is for which job. Simultaneously, existing job recommendation systems only take into consideration the domain in which the user is interested while ignoring their profile and skillset, which can help recommend jobs that are tailor-made for the user. This paper examines the user's resume then compares the knowledge of degree, soft skills, hard skills, and the projects he has done and then only the system recommends the jobs for that user. The system not only recommends the jobs but also shows the score of his/her resume for the respective job. Then, the system also recommends skills to improve the scores of their resume.

Index Terms: **Skill set, Cosine Similarity, Jaccard Similarity, Euclidean Similarity**

I. INTRODUCTION

A recent report claims that most college graduates have difficulty in choosing their domain in their job. Many engineers are trying to shift the domain from their field to IT. So, they are doing some courses in online and randomly searching for a job. Nowadays, IT fields are the targets of many students but they don't know which domain is fit for them. To avoid this situation candidates, need a Job recommendation that analyses the skills to recommend a suitable job for the candidate. The solution is to design a system that reads a resume and their skills. The resumes are going through pre-processing to make the design more efficient. For pre-processing top words and porter stemmer, Porter Stemmer will make every word their root word, and stop words will remove every meaningless word. This makes the system more efficient. Using tf-idf vectorizer for both resume and job description. Then compare the skills in the resume and description.

For comparing, it uses the Cosine Similarity function and finds the scores of the resume for the respective jobs. Now it sorts the list in descending order with respect to their scores. Now, he got a hierarchical order of jobs from top tobottom. So, he can go with the first job or second which the skill he had already. He can be successful in that domain. The System not only shows the job but also recommends the skills to be improved for the job. Because of this, the candidate can train himself/herself for the future purpose and be a more achievable or talented person in his/her domain.

The proposed work focuses on predicting the suitable jobs for the candidates. It uses machine learning models to find similarities between jobs description and resumes to predict accurately. This application can be used by any candidates who need or who want to know about their suitable jobs and to improve themselves with both soft skills and hard skills. It will be helpful to them by not wasting their time searching for jobs. They can also grow their skills in their domain and grow faster in their domain.

1.1 DATASET

Two Kaggle resources and some from google searches were combined to produce a Job Dataset. I gather resume data from my friends. 13001 job descriptions are included in Job Dataset in .csv format. Each row has a different Id and three Features. The features include Job Title, ID, Query, and Description. Another dataset exists, it is called the skill dataset. The information for the skill dataset is gathered from Google by looking up each job. The job title and skills are included. There are 32 rows of specific job skills listed.

Table 1.1 Summary of the dataset

Type of Dataset	Number of rows
Job description	13000
Resumes	101
Job skills	32

1.2 METHODOLOGY

To find suitable jobs and their scores, this application receives the resume and has a dataset for a job with their description. It will pre-process the resume and job description with the stop words and porter's stemmer. Then it reduces into a meaningful bag of words. Now the application uses a tf-idf vectorizer to convert a raw text into a matrix which makes it easy while compare. The main step is comparing the two bag words. For that, it uses the Cosine Similarity function, which is an angle-dependent calculation. By using cosine, it has a list of jobs in descending order with respect to scores. The system will move on to the next progress which is finding the skills to be improved by the candidates. The system will take the resume and the skills dataset then compares both and display the skills which are all not in the resume.

The major contribution of this work is as follows: The large MNC businesses use the mechanism currently in place for employment recommendations. The method is employed by businesses, not by regular people. If not, they will charge a small subscription fee to check the user's career options. The system functions for the average guy from city to village to modify this predicament. Because the students would look for employment based on their own skills, this approach will reduce unemployment. This company will also grow more quickly, which will result in more job openings.

The goal of the proposed work is to suggest a job that is ideal for the user. It displays the hierarchical jobs that are best for the user, not just one job. Additionally, it suggests skills for the jobs that were suggested for the user. This project is intended for someone who simply has no idea what they are going to do. Additionally, there are no logins available because doing so increases the likelihood that users would reject you. The subsequent chapter goes over the specifics of the implementation. The rest of the paper organizes as follows: **Chapter 2** provides the literature review conducted for this project. **Chapter 3** presents the System Design and Architecture of the project along with the methodology. **Chapter 4** discusses the algorithms proposed in this project. **Chapter 5** presents the project conclusion and future works on this project.

II. LITERATURE REVIEW

Existing works are mainly found for the company to select a candidate who is fit for their vacancy[17]. There are many experiments for calculating the four recommendation algorithm but with a different distance formula namely the Minkowski distance [5]. And some others are tried a different recommender system like collaborative which only helps when there are more data to relate. That won't help for a person who is searching that which job is the correct choice for him/her. R.J. Mooney and L. Roy used Content-Based Book Recommending[1] where the content-based recommendation helps for a cold start. And some authors also say that a content-based recommender is best when they researched a comparison study of job recommendations [9].

A recommender system is not only the main part of accurate prediction. There are some other things like vectorizing the words and then similarity functions. Authors like Shouning Qu[3], and Li-Ping Jing[2] said that for text mining, tf-idf is the best approach for text feature selection. Ravali Boorugu has researched NLP and tried various text summarization techniques[14][19]. Some papers also say about similarity detection with many languages 8][10]. Jeevamol Joy and Renumol V G [16] discussed which similarity is the best one for a content-based recommending system. They finally concluded that cosine similarity is the best similarity for content based recommended system. Cosine similarity is not only used for recommender systems but is used to find the similarity functions between two sentences or two paragraphs[8][20]. Mohammad Alobed has tried "A Comparative Analysis of Euclidean, Jaccard, and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring"[21], and L. Zahrotum also compared jaccarJaccardidean and cosine similarity. They both said that Cosine similarity with all stemming types has the lowest error compared with the Jaccard and Euclidean similarity[7]. There is already a system that worked with both tf-idf and cosine similarity recommendations. It is used for patient support forums[6]. Tanya V. Yadalam, Vaishnavi, M. Gowda, and Vanditha researched those career recommendations content-based filtering which was mostly like my project but inside it, they mostly discussed security, transparency for the data, and the framework [15].

Most works are just built for the companies or for the purpose of making money from the people by giving some irrelevant choices. Many were using collaborative recommendation, which recommends the many searched jobs or the jobs which were chosen by some other. It only works if the system deals with more number of resumes which seems it can only be used by the companies. Some systems are asking to log in and some were asked to buy subscriptions. Logging in makes you redirect some spam mails.

In many papers, they have been solved through content recommender which is not enough. A literature paper[15] had done research on content recommender system, tfidf vectorizer, and cosine similarity in a row but in that the author doesn't think about the implementation process and only concentrated more on securing the data.

III. SYSTEM DESIGN

In this chapter, it discusses about a quick overview of the system's architectural design and an explanation of the architecture. Every module that is employed in the architecture design will be understood well.

3.1 OVERALL HIGH-LEVEL ARCHITECTURE

The high Lehigh-level architecture diagram Figure 3.1 displays the system's overall layout. A file with a job dataset contains the name and description of the job. Resume data is information that the user entered.

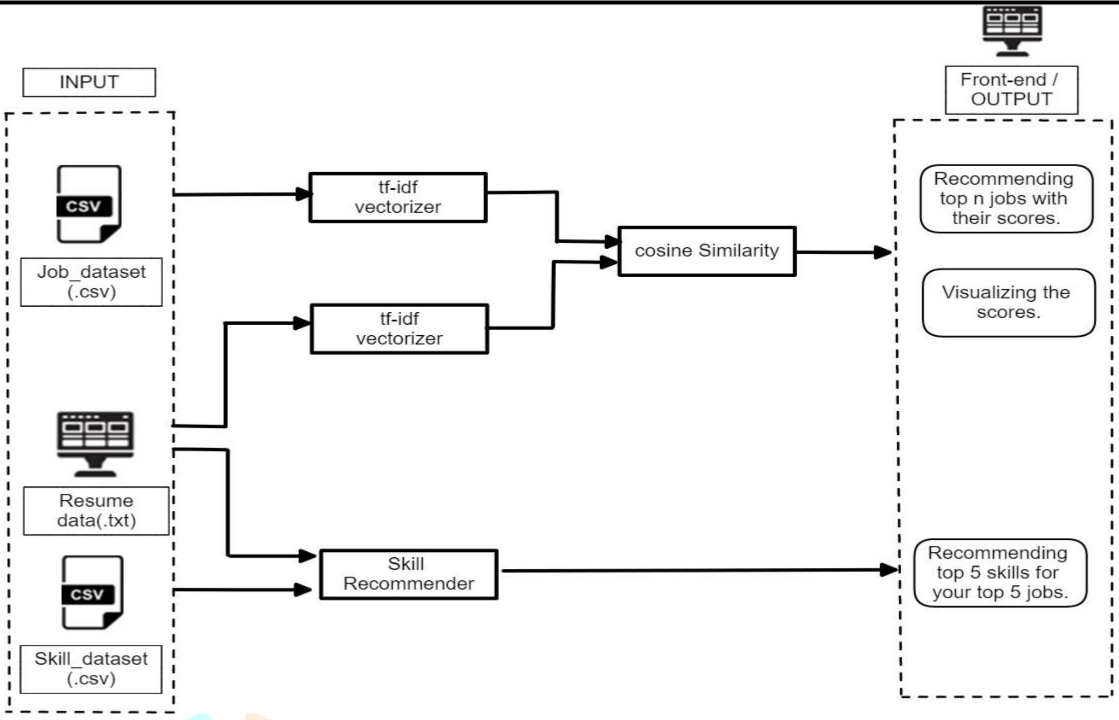


Figure 3.1 High Level Architecture Diagram

The folder was read and then saved for later use. Using the tf-idf vectorizer, a matrix can be created from raw data for both the employment dataset and the

user-collected resume. Then, using a method in similarity functions known as Cosine-Similarity, it will determine the scores between the job description of the job dataset and the resume. It involves calculating the total of the vectors dot products divided by the sum of their products divided by the products of their lengths. It will first display the top jobs and their scores in table format before visualizing the results in a pie chart. The system will then analyze the user's resume and the skill dataset to suggest skills that should be improved.

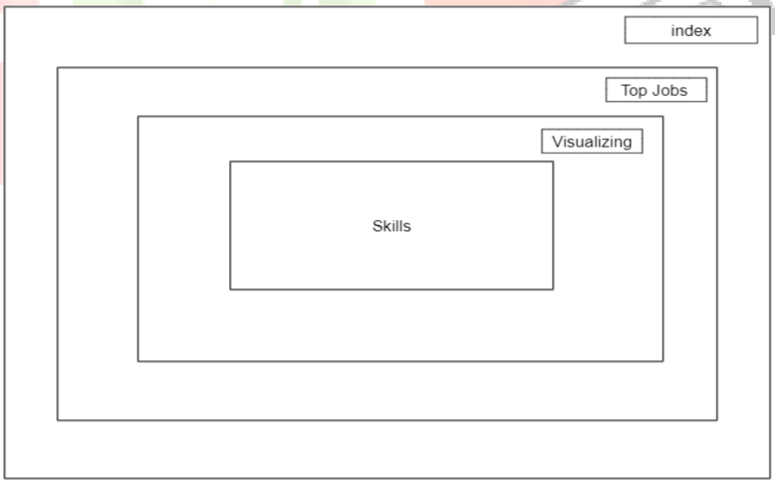


Figure 3.2: Modular View

The Modular view Figure 3.2 provides better visualization of the system design. Index is the area where the user uploads a resume. Top Jobs will be recommended in descending order with scores. Visualizing the recommended jobs and scores. Skills show the skills that need to be improved for the respective jobs.

3.2 DETAILED DESIGN

3.2.1 Preprocessing

To obtain the cleaned dataset, it processes the data using Porter Stemmer and Stop words in Preprocessing Figure 3.3. "ID," "Query," "Job Title," and "Job Description" are all included in the Job dataset file. In this case, Query superset Job Title as the job. Therefore, only the superset portion, Query, and its explanation are required in this system.

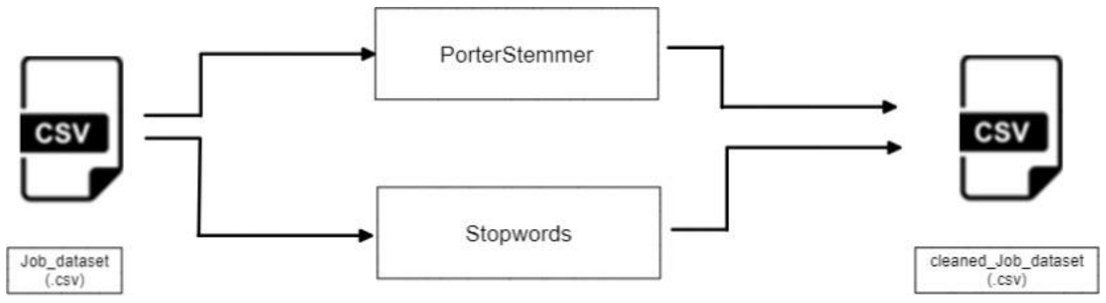


Figure 3.3: Preprocessing

To compare, the system doesn't require terms like "is" or "are" that don't give a score or points, therefore it only reads or chooses those two traits. Therefore, Stop words are employed to eliminate all the pointless/meaningless words. Porter Stemmer then used to lower the word frequency. An algorithm for stemming is Porter Stemmer. Stemming is the process of stripping a word down to its root, or lemma, which attaches to suffixes, prefixes, or other roots to form a base word. For instance, the basic forms run, runs, running, and ran are all variations of the same basic form; run is the lemma. In the appendices, the Porter Stemmer algorithm and comprehensive version are written. In this module, the old dataset will be processed to produce a new, cleaned dataset that contains just lemma terms. Utilizes nltk (Natural Language Toolkit).

3.2.2 Similarity Function Module

A resume that was obtained from a user in text format and a cleaned employment dataset make up the Similarity Function Module Figure 3.4. There is also a resume validation that, when the user tries to select a file, only displays the (.txt) file in the file manager. The machine selected cosine similarity with tf-idf vectorizer for this process.

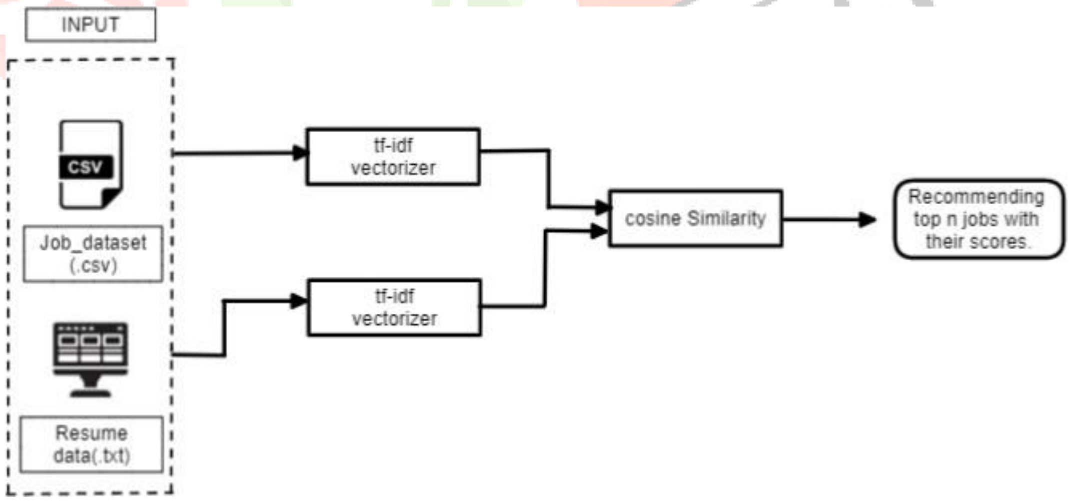


Figure 3.4 Similarity Function Module

The resume and the cleaned job dataset are both included in the Cosine Similarity function, which is processed by stopwords and porterStemmer (which was also used to analyse the job dataset's description). Now, a processed text file is also a resume. Use tf-idf, which stands for term frequency-inverse document frequency. Tf-idf was applied to both the processed resume and description. We are given a matrix of values that is already included in the raw text. The more often a term appears in a document, the higher it is regarded by the tf-idf. In order to account for the fact that a few terms appear more frequently overall, it isbalanced by the number of documents in the corpus that contain the word. After receiving a matrix value system, the cosine similarity formula can be used to continue. The formula is the sum of the vectors' dot products divided by the sum of the vector products divided by the vector products of lengths. Once the algorithm is operational, the system can produce scores indicating how closely the job description and the resume match. Change the order of the scores that it is descending to display the results. so that the user may quickly select the top-rated jobs and focus on them.

3.2.3 Skills Recommending Module

The Skills Recommending Module Figure 3.5 will demonstrate the workflow of the system that uses resumes and a skill dataset to identify the top five skills for each position. The following paragraphs go into the implementation's details.

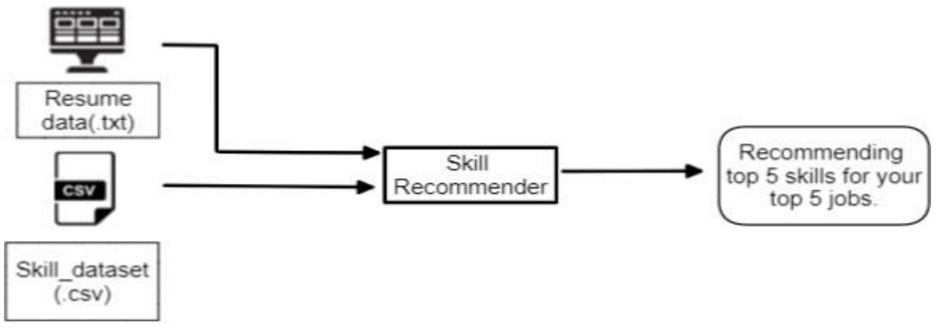


Figure 3.5: Skills Recommending Module

The skill dataset and resumes are the inputs for this module. Skills for the jobs are contained in a skill dataset. The algorithm will now compare the word list for both the skill dataset and the resume. Additionally, print the top five talents that don't match the resume. The system will list the top 5 occupations and top 5 abilities that the resume holder needs to develop in this module. There are several cleaning procedures for the skill dataset, such as deleting the prefix and suffix spaces used to match the words.

IV. PSEUDOCODE AND ALGORITHM

4.1 PREPROCESSING

The pre-processing module will clean the data of the job description and resume and give cleaned data. The algorithm is described in Algorithm 4.1.

Input: Job dataset (.csv file)

Output: Cleaned Job dataset (.csv file)

Algorithm 4.1 Preprocessing

```
1:    read job dataset file
2:        reading a file with pandas and store it in a variable k
3:    separate only job and description attributes from k by using split in python
4:        By using split operation it turns into list 5:    for each word
in description
6:        if word not in stopwords then
7:            use the word into porterstemmer
8:            then convert the list of stemmed words into a txt format 9:    else left
10:   save in a separate file
```

4.2 SIMILARITY FUNCTION MODULE

The similarity function module will get the resume from user and scan it. Then compare the resume and cleaned job description to give the top jobs and scores for it. The Algorithm for compressing the images and masks is described in Algorithm 4.2.

Input: Cleaned job dataset and resume

Output: Recommending N jobs in descending order with respect to their scores.

Algorithm 4.2 Similarity Function Module

```
1:      upload resume
2:          if resume uploaded then
3:              read and store a resume in a folder 4:    else indicates to
enter a file
5:      convert the resume into list and remove the spaces, tabs, and empty elements
6:      then the pre-processing module to the resume file
7:      apply both job description and resume matrices to cosine similarity formula.
8:          convert matrix into array
9:          Job description array as arr1 and resume array as arr2 10: for l1 in arr1 and
creating a list asa
11:             for l2 in arr2
12:             store sum of l1 in s1 and sum of l2 in  s2
13:             applying to formula s1.s2 / float(s1.s2)**0.5 and append to the list
14:         Return
15:     sum the duplicate jobs
16:     store it in a dictionary then average it with the total number of count 17:    sort in descending order
with respect to scores and display the results
```

The main work of my project were completed in **Algorithm 4.2** and only recommending skills are left.

4.3 SKILLS RECOMMENDING MODULE

This module is used to recommend top 5 skills of top 5 jobs to the user to upgrade themselves. The Algorithm for compressing the images and masks is described in Algorithm 4.3.

Input: Processed resume and skill dataset(.csv) file

Output: Top 5 skills for top 5 jobs

Algorithm 4.3 Skills Recommending Module

```
1:      read the file skill dataset
2:          reading a file with pandas and store it in a variable k 3:   appending job to a
separate list and skills to a separate list 4:    zip the job and skills from skill dataset
5:      pick the top 5 jobs from the output of similarity function module and save it in list
6:      for element in top5jobs
7:          skills = values of element from the skill dataset 8:        for element in
skills
9:              if element not in listofResume 10:
                store in recskills list
11:             else ignore
12:         slice the recskills list with only 5elements 13:    display the job
and skills to be improved
```

5 EXPERIMENTAL RESULTS AND OUTPUTS

The experiments and results of job a recommendation system with skills are discussed below.

5.1 PREPROCESSING

The job dataset will be processed by the system first. Using stop words and the porter stemmer process can be accelerated and streamlined.

5.1.1 Job Dataset

There are four characteristics: ID, Query, Job Title, and Description. The system will reduce it to only two attributes, Query and Description. Comparing and addressing job descriptions, after that, eliminate any words with no meaning or score using stop words. The words are then transformed into stem words using porter's stemmer prefixes and suffixes are eliminated.

5.2 SIMILARITY FUNCTION MEASURES

This module explains why the system chose to use Cosine similarity over other similarities. Cosine similarity, Euclidean similarity, and Jaccard similarity were the three similarities that the system examined. Additionally, the same cleaned Job dataset and a resume in (.txt) format were used to test these three similarities. The experiments involving the application of the three similarities are discussed in the chapter that follows.

5.2.1 Cosine Similarity Function

The resume and the cleaned job dataset are both included in the Cosine Similarity function, which is processed by stopwords and porterStemmer (which was also used to analyze the job dataset's description). Now, a processed text file is also a resume.

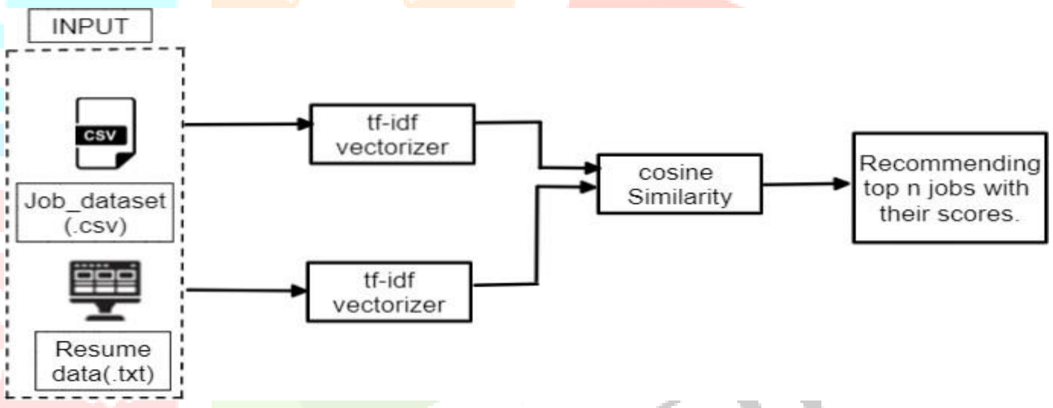
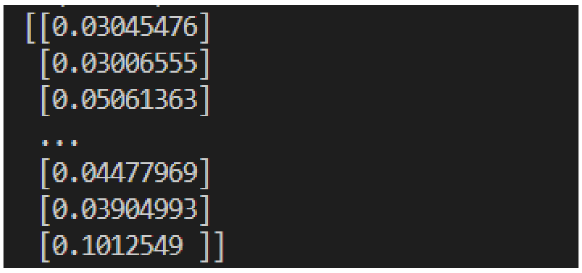


Figure 5.1: Cosine Similarity Function

The resume and the cleaned job dataset are both included in the Cosine Similarity function, which is processed by stop words and porter Stemmer (which was also used to analyze the job dataset's description). Now, a processed text file is also a resume. Use tf-idf, which stands for term frequency-inverse document frequency. Tf-idf was applied to both the processed resume and description. It is given a matrix of values that is already included in the raw text. The more often a term appears in a document, the higher it is regarded by the tf-idf. In order to account for the fact that a few terms appear more frequently overall, it isbalanced by the number of documents in the corpus that contain the word. After receiving a matrix value system, the cosine similarity formula can be used to continue. The formula is the sum of the vectors' dot products divided by the sum of the vector products divided by the vector products of lengths. Once the algorithm is operational, the system can produce scores indicating how closely the job description and the resume match. Change the scores order such that it is descending to display the results. so that the user may quickly select the top-rated jobs and focus on them. For eg. (the first job title is data scientist and there are 400 data scientist's description and it shows scores for every description). Then sum the scores of a job which are similar and divide it with the total number of description (i.e 400). After that store the job and description in dictionary has keys and values. Display the output by changing the scores into descending order. So that user can easily pick the top rated jobs and concentrate to it.

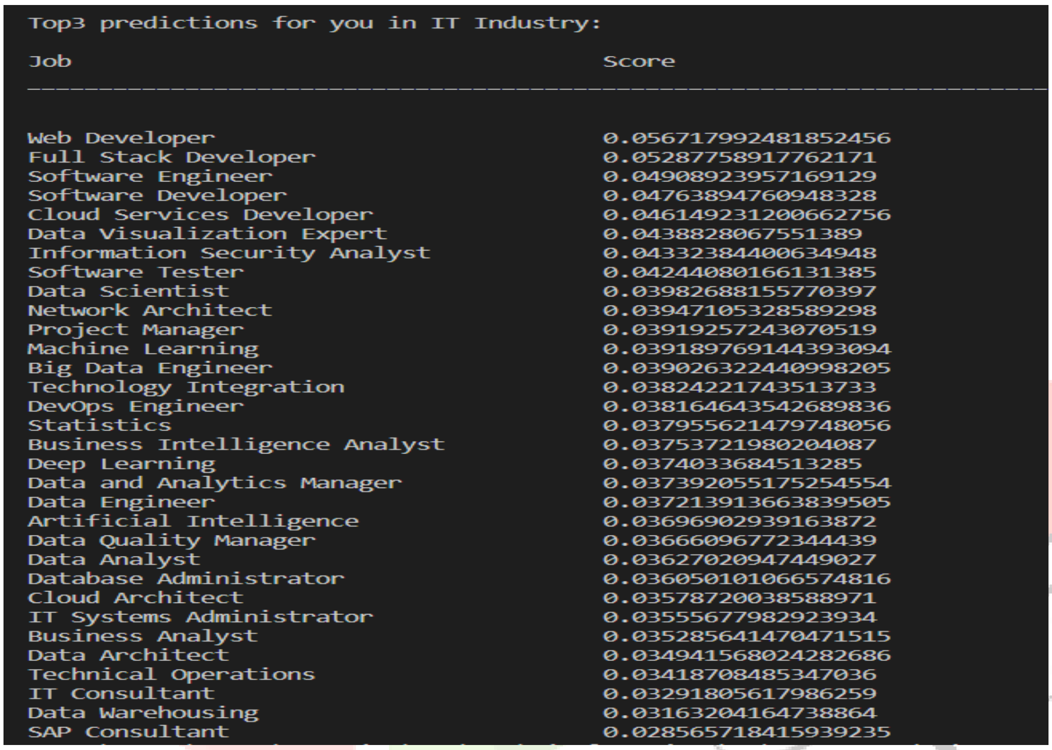
Output :



```
[[0.03045476]
 [0.03006555]
 [0.05061363]
 ...
 [0.04477969]
 [0.03904993]
 [0.1012549 ]]
```

Figure 5.2: Cosine Similarity in Array

The points in the array are shown in the previous picture Figure 5.2. This displays scores for all jobs, not just unique jobs. The array value should then be averaged to determine a score for each individual jobs.



```
Top3 predictions for you in IT Industry:

Job                                        Score
--------------------------------------------------------------------------


Web Developer                              0.056717992481852456
Full Stack Developer                       0.05287758917762171
Software Engineer                          0.04908923957169129
Software Developer                         0.04763894760948328
Cloud Services Developer                   0.046149231200662756
Data Visualization Expert                  0.0438828067551389
Information Security Analyst               0.04332384400634948
Software Tester                            0.04244080166131385
Data Scientist                             0.03982688155770397
Network Architect                          0.03947105328589298
Project Manager                            0.03919257243070519
Machine Learning                           0.039189769144393094
Big Data Engineer                          0.039026322440998205
Technology Integration                     0.03824221743513733
DevOps Engineer                            0.038164643542689836
Statistics                                 0.037955621479748056
Business Intelligence Analyst              0.03753721980204087
Deep Learning                              0.0374033684513285
Data and Analytics Manager                 0.037392055175254554
Data Engineer                              0.037213913663839505
Artificial Intelligence                    0.03696902939163872
Data Quality Manager                       0.03666096772344439
Data Analyst                               0.03627020947449027
Database Administrator                     0.036050101066574816
Cloud Architect                            0.03578720038588971
IT Systems Administrator                   0.03555677982923934
Business Analyst                           0.035285641470471515
Data Architect                             0.034941568024282686
Technical Operations                       0.03418708485347036
IT Consultant                              0.03291805617986259
Data Warehousing                           0.03163204164738864
SAP Consultant                             0.028565718415939235
```

Figure 5.3: Cosine Similarity Output

5.2.2 Euclidean Similarity Function

The Euclidean Similarity also consists of the same input which was given to the Cosine similarity function. The same process goes under Euclidean also but in cosine it used tf-idf vectorizer but here it uses count vectorizer. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. After converting the text into matrix value system is ready to proceed in a Euclidean similarity formula.

$$\textbf{Euclidean Similarity} = 1 / (1+(d(x,y)))\quad d(x,y) = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

where, x and y are coordinates. (5.1)

As soon as the formula begins to produce results, scores that indicate how closely the job description and the resume match are displayed. Change the scores' order such that it is descending to display the results. so that the user may quickly select the top-rated jobs and focus on them.

Output :

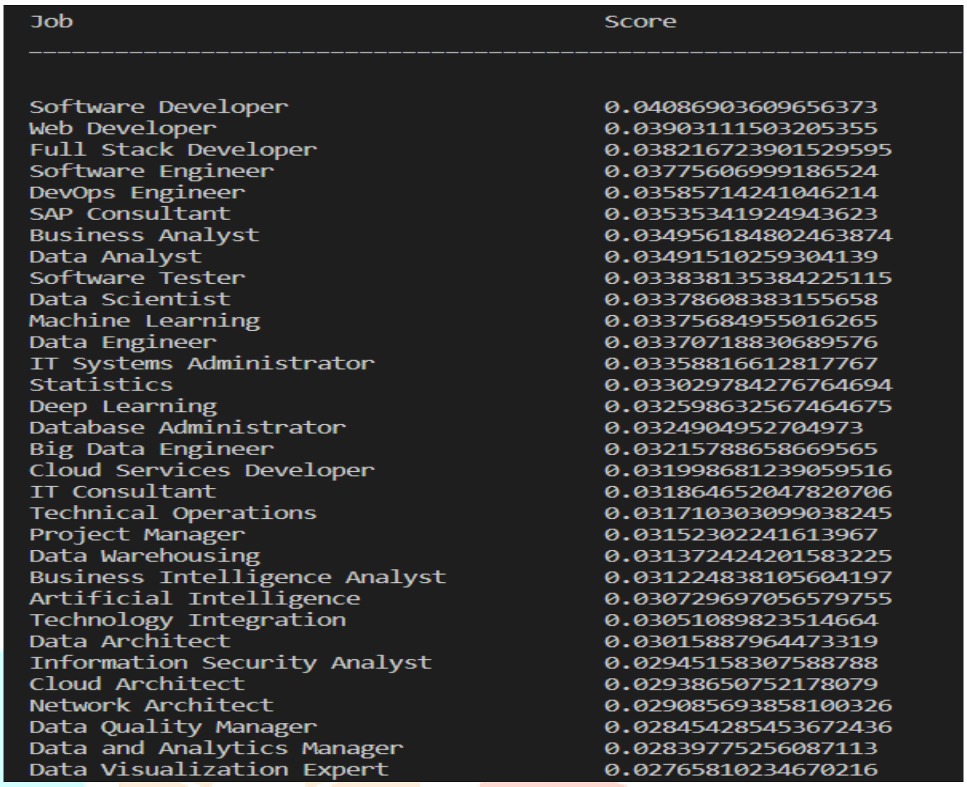
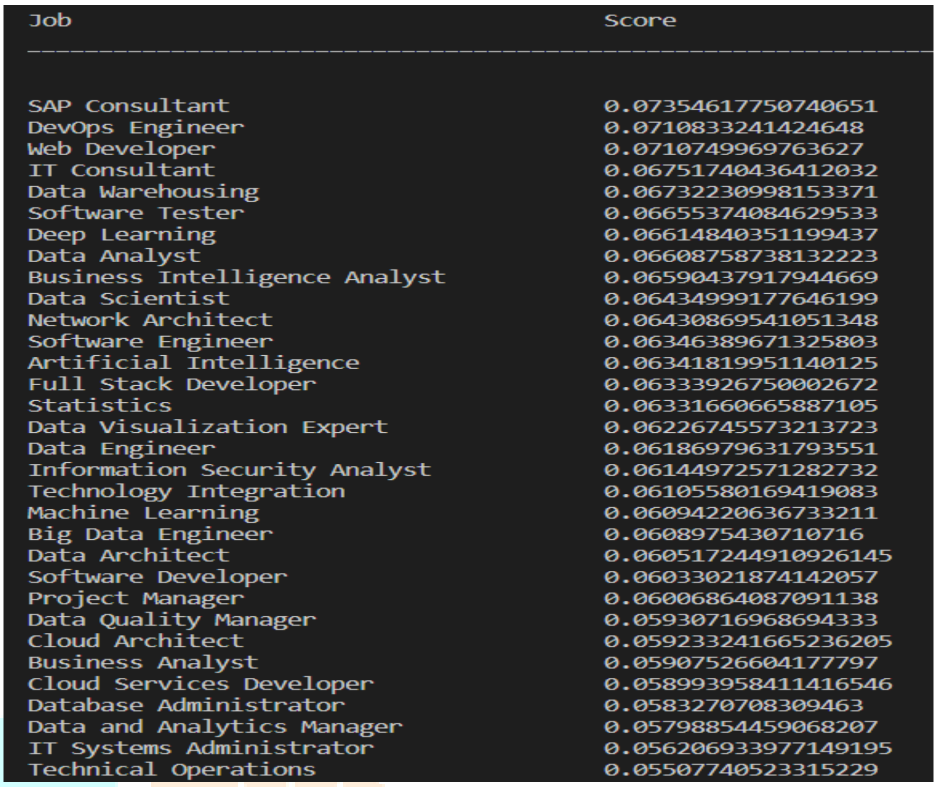

```
Job                                         Score
-----------------------------------------------------------------


Software Developer                          0.04086903609656373
Web Developer                               0.03903111503205355
Full Stack Developer                        0.038216723901529595
Software Engineer                           0.03775606999186524
DevOps Engineer                             0.03585714241046214
SAP Consultant                              0.03535341924943623
Business Analyst                            0.034956184802463874
Data Analyst                                0.03491510259304139
Software Tester                             0.033838135384225115
Data Scientist                              0.03378608383155658
Machine Learning                            0.03375684955016265
Data Engineer                               0.03370718830689576
IT Systems Administrator                    0.03358816612817767
Statistics                                  0.033029784276764694
Deep Learning                               0.032598632567464675
Database Administrator                      0.0324904952704973
Big Data Engineer                           0.03215788658669565
Cloud Services Developer                    0.031998681239059516
IT Consultant                               0.031864652047820706
Technical Operations                        0.031710303099038245
Project Manager                             0.03152302241613967
Data Warehousing                            0.031372424201583225
Business Intelligence Analyst               0.031224838105604197
Artificial Intelligence                     0.030729697056579755
Technology Integration                      0.03051089823514664
Data Architect                              0.03015887964473319
Information Security Analyst                0.02945158307588788
Cloud Architect                             0.02938650752178079
Network Architect                           0.029085693858100326
Data Quality Manager                        0.028454285453672436
Data and Analytics Manager                  0.02839775256087113
Data Visualization Expert                   0.02765810234670216
```

Figure 5.4: Euclidean Similarity Output

5.2.3 Jaccard Similarity Function

The identical input used by the Euclidean and Cosine similarity functions is also used by the Jaccard similarity function. The only difference is in the metrics the cosine and Euclidean similarity data flows are identical. Here, the system makes no use of a vectorizer; it merely measures the number of words in both the processed job description and processed resume. Apply the Jaccard formula next. The number of observations in both sets divided by the number in eitherset is the Jaccard formula. To put it another way, |AB| / |AB|.
As soon as the formula begins to produce results, scores that indicate how closely the job description and the resume match are displayed. Change the scores' order such that it is descending to display the results. So, that the user may quickly select the top-rated jobs and focus on them.

Output :

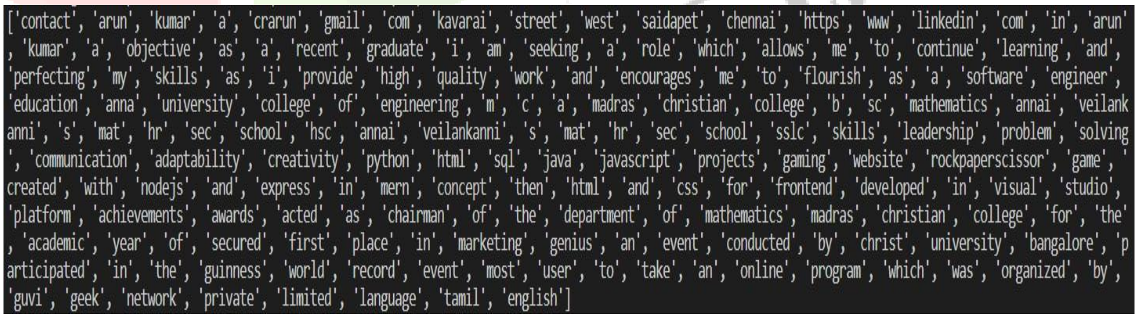


```
Job                                     Score
-------------------------------------------------------------------


SAP Consultant                          0.07354617750740651
DevOps Engineer                         0.0710833241424648
Web Developer                           0.0710749969763627
IT Consultant                           0.06751740436412032
Data Warehousing                        0.06732230998153371
Software Tester                         0.06655374084629533
Deep Learning                           0.06614840351199437
Data Analyst                            0.06608758738132223
Business Intelligence Analyst           0.06590437917944669
Data Scientist                          0.06434999177646199
Network Architect                       0.06430869541051348
Software Engineer                       0.06346389671325803
Artificial Intelligence                 0.06341819951140125
Full Stack Developer                    0.06333926750002672
Statistics                              0.06331660665887105
Data Visualization Expert               0.06226745573213723
Data Engineer                           0.06186979631793551
Information Security Analyst            0.06144972571282732
Technology Integration                  0.06105580169419083
Machine Learning                        0.06094220636733211
Big Data Engineer                       0.0608975430710716
Data Architect                          0.060517244910926145
Software Developer                      0.06033021874142057
Project Manager                         0.06006864087091138
Data Quality Manager                    0.05930716968694333
Cloud Architect                         0.059233241665236205
Business Analyst                        0.05907526604177797
Cloud Services Developer                0.058993958411416546
Database Administrator                  0.0583270708309463
Data and Analytics Manager              0.05798854459068207
IT Systems Administrator                0.056206933977149195
Technical Operations                    0.05507740523315229
```

Figure 5.5: Jaccard Similarity Output

5.2.4 Cosine or Euclidean or Jaccard

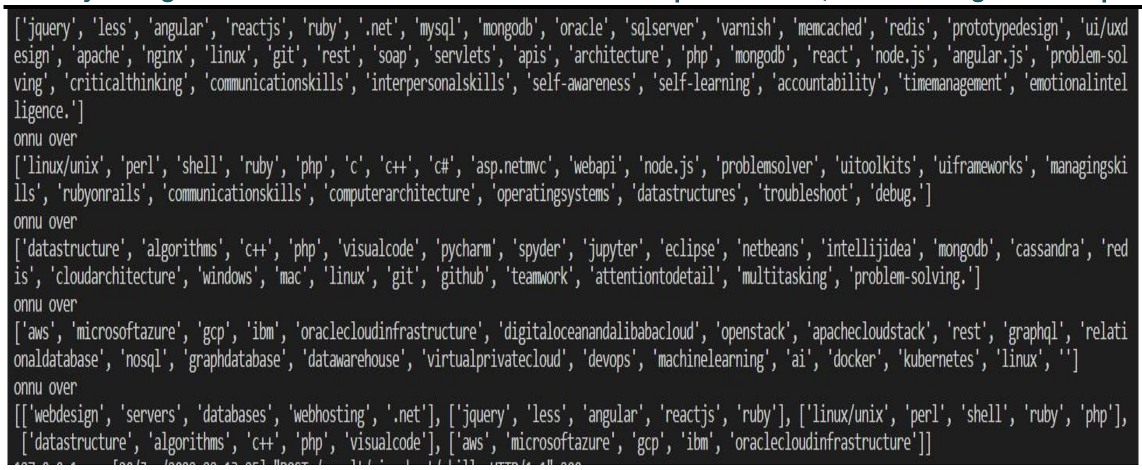
The positions that are advised for each of the three similarity functions. The system will now determine which is superior. Offered my guide or mam a CV, the job list, and requested her to order the position. The top ten jobs out of thirty two were ranked by mam, who then verified it with the three outputs. Mam's perspective and the cosine similarity recommendation are practically identical. It compares the Cosine, Euclidean, and Jaccard similarity functions in the research paper. Mohammad Alobed said in the paper that cosine similarity is superior to the other two. A Comparative Analysis of Euclidean, Jaccard, and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring" is the title of the paper that is related to the literature review.



```
['contact', 'arun', 'kumar', 'a', 'crarun', 'gmail', 'com', 'kavarai', 'street', 'west', 'saidapet', 'chennai', 'https', 'www', 'linkedin', 'com', 'in', 'arun'
, 'kumar', 'a', 'objective', 'as', 'a', 'recent', 'graduate', 'i', 'am', 'seeking', 'a', 'role', 'which', 'allows', 'me', 'to', 'continue', 'learning', 'and',
'perfecting', 'my', 'skills', 'as', 'i', 'provide', 'high', 'quality', 'work', 'and', 'encourages', 'me', 'to', 'flourish', 'as', 'a', 'software', 'engineer',
'education', 'anna', 'university', 'college', 'of', 'engineering', 'm', 'c', 'a', 'madras', 'christian', 'college', 'b', 'sc', 'mathematics', 'annai', 'veilank
anni', 's', 'mat', 'hr', 'sec', 'school', 'hsc', 'annai', 'veilankanni', 's', 'mat', 'hr', 'sec', 'school', 'sslc', 'skills', 'leadership', 'problem', 'solving
', 'communication', 'adaptability', 'creativity', 'python', 'html', 'sql', 'java', 'javascript', 'projects', 'gaming', 'website', 'rockpaperscissor', 'game', '
created', 'with', 'nodejs', 'and', 'express', 'in', 'mern', 'concept', 'then', 'html', 'and', 'css', 'for', 'frontend', 'developed', 'in', 'visual', 'studio',
'platform', 'achievements', 'awards', 'acted', 'as', 'chairman', 'of', 'the', 'department', 'of', 'mathematics', 'madras', 'christian', 'college', 'for', 'the'
, 'academic', 'year', 'of', 'secured', 'first', 'place', 'in', 'marketing', 'genius', 'an', 'event', 'conducted', 'by', 'christ', 'university', 'bangalore', 'p
articipated', 'in', 'the', 'guinness', 'world', 'record', 'event', 'most', 'user', 'to', 'take', 'an', 'online', 'program', 'which', 'was', 'organized', 'by',
'guvi', 'geek', 'network', 'private', 'limited', 'language', 'tamil', 'english']
```

Figure 5.6: Resume in List

The above figure 5.6 is showing the processed resume which is in list format.



```
['jquery', 'less', 'angular', 'reactjs', 'ruby', '.net', 'mysql', 'mongodb', 'oracle', 'sqlserver', 'varnish', 'memcached', 'redis', 'prototypedesign', 'ui/uxd
esign', 'apache', 'nginx', 'linux', 'git', 'rest', 'soap', 'servlets', 'apis', 'architecture', 'php', 'mongodb', 'react', 'node.js', 'angular.js', 'problem-sol
ving', 'criticalthinking', 'communicationskills', 'interpersonalskills', 'self-awareness', 'self-learning', 'accountability', 'timemanagement', 'emotionalintel
ligence.']
onnu over
['linux/unix', 'perl', 'shell', 'ruby', 'php', 'c', 'c++', 'c#', 'asp.netmvc', 'webapi', 'node.js', 'problemsolver', 'uitoolkits', 'uiframeworks', 'managingski
lls', 'rubyonrails', 'communicationskills', 'computerarchitecture', 'operatingsystems', 'datastructures', 'troubleshoot', 'debug.']
onnu over
['datastructure', 'algorithms', 'c++', 'php', 'visualcode', 'pycharm', 'spyder', 'jupyter', 'eclipse', 'netbeans', 'intellijidea', 'mongodb', 'cassandra', 'red
is', 'cloudarchitecture', 'windows', 'mac', 'linux', 'git', 'github', 'teamwork', 'attentiontodetail', 'multitasking', 'problem-solving.']
onnu over
['aws', 'microsoftazure', 'gcp', 'ibm', 'oraclecloudinfrastructure', 'digitaloceanandalibabacloud', 'openstack', 'apachecloudstack', 'rest', 'graphql', 'relati
onaldatabase', 'nosql', 'graphdatabase', 'datawarehouse', 'virtualprivatecloud', 'devops', 'machinelearning', 'ai', 'docker', 'kubernetes', 'linux', '']
onnu over
[['webdesign', 'servers', 'databases', 'webhosting', '.net'], ['jquery', 'less', 'angular', 'reactjs', 'ruby'], ['linux/unix', 'perl', 'shell', 'ruby', 'php'],
 ['datastructure', 'algorithms', 'c++', 'php', 'visualcode'], ['aws', 'microsoftazure', 'gcp', 'ibm', 'oraclecloudinfrastructure']]
```

Figure 5.7: Skills in List

The above figure 5.7 is showing the processed skills which are in list format.

Expert Recommended Output:

Table 5.1 Expert recommended top 10 jobs

S.NO	JOBS
1.	WEB DEVELOPER
2.	FULL STACK DEVELOPER
3.	SOFTWARE DEVELOPER
4.	SOFTWARE ENGINEER
5.	DATA VISUALIZATION EXPERT
6.	SOFTWARE TESTER
7.	DATA SCIENTIST
8.	DATA ANALYST
9.	INFORMATION SECURITY ANALYST
10.	MACHINE LEARNING

The Cosine Similarity Function is the best to use, the system can conclude after comparison with the table. Therefore, cosine was implemented, and the user interface showed the output. All of those will be seen in the following pages.

5.3 VISUALIZATION

After getting the top-recommended jobs and their scores. The system will display it as a pie chart. Google Charts were utilized for visualization. Pie-chart has been constructed by using the keys and values from the Similarity Functions solution. It is a three-dimensional pie chart. The label is included in a three-dimensional pie chart, and it is distinguished by different colors. The job names will then be listed at the side of the three-dimensional pie chart in descending order according to the result of cosine similarity functions. Moreover, a drop-down arrow will be present. The drop-down arrow is used to display all thirty-two jobs that appeared in the list.

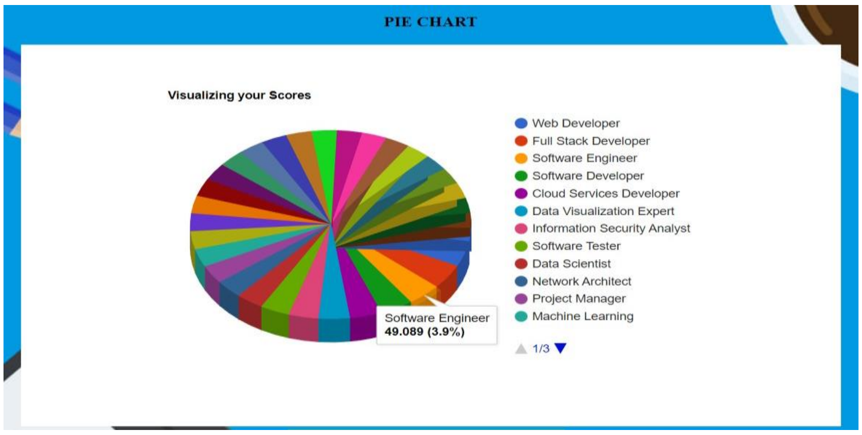
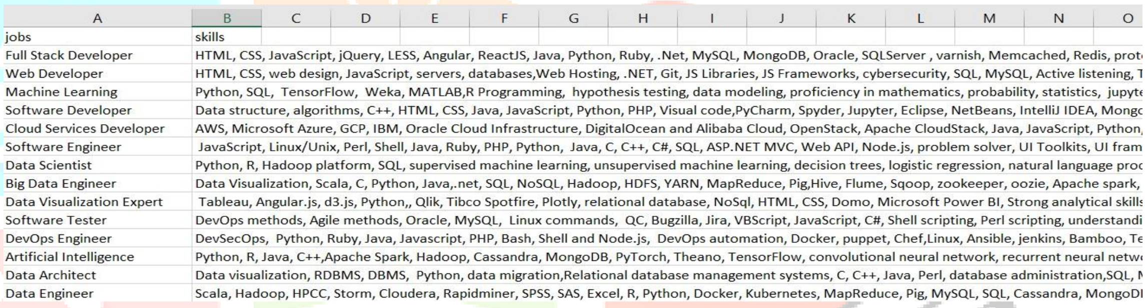


Figure 5.8: Pie Chart Output

5.4 SKILL RECOMMENDATION

A processed Resume and a skill dataset are required for skill recommendation. A resume with stop words and a porter stemmer will be processed and turned into a list of words. Job and skills are the two attributes of the skill dataset.

Actual Input

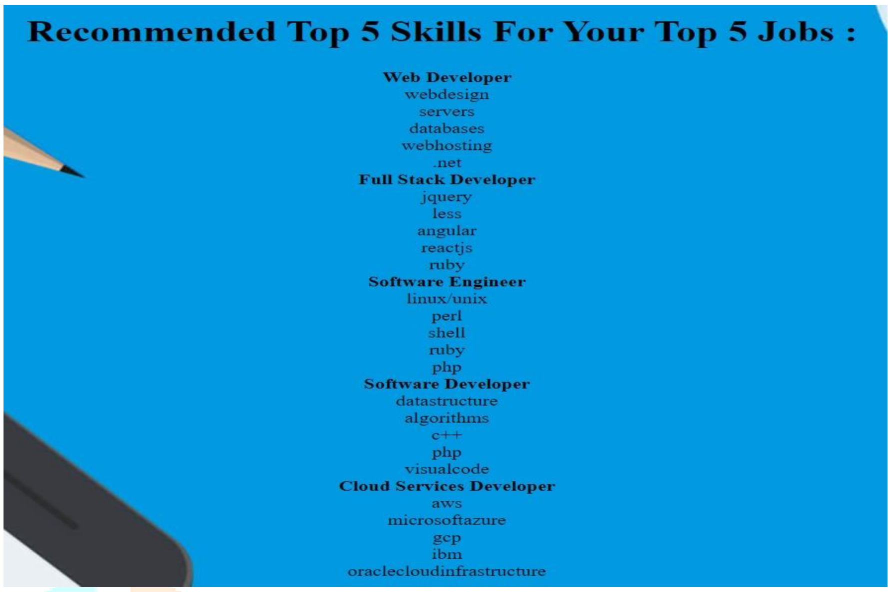


A	B
jobs	skills
Full Stack Developer	HTML, CSS, JavaScript, jQuery, LESS, Angular, ReactJS, Java, Python, Ruby, .Net, MySQL, MongoDB, Oracle, SQLServer , varnish, Memcached, Redis, prot
Web Developer	HTML, CSS, web design, JavaScript, servers, databases,Web Hosting, .NET, Git, JS Libraries, JS Frameworks, cybersecurity, SQL, MySQL, Active listening, T
Machine Learning	Python, SQL, TensorFlow, Weka, MATLAB,R Programming, hypothesis testing, data modeling, proficiency in mathematics, probability, statistics, jupyte
Software Developer	Data structure, algorithms, C++, HTML, CSS, Java, JavaScript, Python, PHP, Visual code,PyCharm, Spyder, Jupyter, Eclipse, NetBeans, IntelliJ IDEA, Mongo
Cloud Services Developer	AWS, Microsoft Azure, GCP, IBM, Oracle Cloud Infrastructure, DigitalOcean and Alibaba Cloud, OpenStack, Apache CloudStack, Java, JavaScript, Python,
Software Engineer	JavaScript, Linux/Unix, Perl, Shell, Java, Ruby, PHP, Python, Java, C, C++, C#, SQL, ASP.NET MVC, Web API, Node.js, problem solver, UI Toolkits, UI fram
Data Scientist	Python, R, Hadoop platform, SQL, supervised machine learning, unsupervised machine learning, decision trees, logistic regression, natural language proc
Big Data Engineer	Data Visualization, Scala, C, Python, Java,.net, SQL, NoSQL, Hadoop, HDFS, YARN, MapReduce, Pig,Hive, Flume, Sqoop, zookeeper, oozie, Apache spark,
Data Visualization Expert	Tableau, Angular.js, d3.js, Python,, Qlik, Tibco Spotfire, Plotly, relational database, NoSql, HTML, CSS, Domo, Microsoft Power BI, Strong analytical skills
Software Tester	DevOps methods, Agile methods, Oracle, MySQL, Linux commands, QC, Bugzilla, Jira, VBScript, JavaScript, C#, Shell scripting, Perl scripting, understandi
DevOps Engineer	DevSecOps, Python, Ruby, Java, Javascript, PHP, Bash, Shell and Node.js, DevOps automation, Docker, puppet, Chef,Linux, Ansible, jenkins, Bamboo, Te
Artificial Intelligence	Python, R, Java, C++,Apache Spark, Hadoop, Cassandra, MongoDB, PyTorch, Theano, TensorFlow, convolutional neural network, recurrent neural netw
Data Architect	Data visualization, RDBMS, DBMS, Python, data migration,Relational database management systems, C, C++, Java, Perl, database administration,SQL,
Data Engineer	Scala, Hadoop, HPCC, Storm, Cloudera, Rapidminer, SPSS, SAS, Excel, R, Python, Docker, Kubernetes, MapReduce, Pig, MySQL, SQL, Cassandra, MongoD

Figure 5.9: Skill Data

Skill Data Figure 5.9 display the skills required for every job which are all included.

Actual Output



Recommended Top 5 Skills For Your Top 5 Jobs :

Web Developer
webdesign
servers
databases
webhosting
.net
Full Stack Developer
jquery
less
angular
reactjs
ruby
Software Engineer
linux/unix
perl
shell
ruby
php
Software Developer
datastructure
algorithms
c++
php
visualcode
Cloud Services Developer
aws
microsoftazure
gcp
ibm
oraclecloudinfrastructure

Figure 5.10: Recommended Skills Output

VI. CONCLUSION AND FUTURE WORK

Job Recommendation System has a major role to play among recommending systems. With the presence of new algorithms and techniques, the system needs to evolve along with it. The main objective of this project is to recommend a suitable job for the candidates.

This project has two pre-processing methods, one text mining method and one similarity function. The pre-processing methods are stop words and porter stemmer. The text mining method is tf-idf. The similarity function is a cosine similarity function. Pre-processing methods are used with resumes and with jobs description, to make the system more efficient by avoiding some garbage words. Tf-idif is used in processed resumes and processed jobs descriptions to convert it from text to matrix to compare. Cosine Similarity will measure the similarity between the resume and each job description. Finally, it will display the scores for the jobs in a sorted way. There is also a pie chart which is used to visualize the percentage of the scores which is got by the candidate for the jobs. Then use a list compare method to compare the resume and job skills to recommend the skills to be improved by the candidate.

FUTURE WORK

In this proposed work, there is only job recommendations and skill suggestion for IT jobs. It can be improved by suggesting jobs and skills for the Non – IT jobs. In the future, some can find a better choice to find similarity than a cosine similarity. It makes the recommendation more accurate.

REFERENCES

[1] R. J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," in In Proceedings of DL '00: Proceedings of the Fifth ACM Conference on Digital Libraries, New York, NY, pp. 13-20, 2000

[2] Li-Ping Jing, Hou-Kuan Huang, Hong-Bo Shi, "Improved feature selection approach TFIDF in text mining", International Conference on Machine Learning and Cybernetics, pp. 944-946, 2002, doi:10.1109/icmlc.2002.1174522.

[3] Shouning Qu ,Sujuan Wang,Yan Zou, " Improvement of Text Feature Selection Method Based on TFIDF", International Seminar on Future Information Technology and Management Engineering, pp. 79-81, 2008, doi:10.1109/fitme.2008.25.

[4] I. A. Braga, "Evaluation of stopwords removal on the statistical approach for automatic term extraction," Seventh Brazilian Symposium in Information and Human Language Technology, pp. 142-149, 2009.

[5] Nikolaos D. Almalis, Prof. George A. Tsihrintzis, Nikolaos Karagiannis, Aggeliki D. Strati, "A new content-

based job recommendation algorithm for job seeking and recruiting", 6th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1-7, 2015, doi:10.1109/iisa.2015.7388018.

[6] Mohammad Alodadi and Vandana P. Janeja, " Similarity in Patient Support Forums Using TF-IDF and Cosine Similarity Metrics", International Conference on Healthcare Informatics, pp. 521-522, 2015, doi:10.1109/ichi.2015.99.

[7] L. Zahrotun, "Comparison jaccard similarity, cosine similarity and combined both of the data clustering with shared nearest neighbor method," Computer Engineering and Applications Journal. vol. 5. Pp. 11-18, 2016, doi:10.18495/comengapp.v5i1.160, 2016.

[8] Peng Yi, Cheng Yang ,Chen Li, Yingya Zhang, "A Job Recommendation Method Optimized by Position Descriptions and Resume Information", IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 762 -764, March 2017, doi:10.1109/rteict.2017.8256590.

[9] Minh-Luan Tran, Anh-Tuyen Nguyen, Quoc-Dung Nguyen, Tin Huynh, "A comparison study for job recommendation", International Conference on Information and Communications (ICIC), pp. 199-204, 2017, doi:10.1109/infoc.2017.8001667.

[10] Gokul P.P, Akhil BK, Shiva Kumar K.M, "Sentence similarity detection in Malayalam language using cosine similarity", 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 221-225, 2017, doi:10.1109/rteict.2017.8256590.

[11] Leah G. Rodriguez, Enrico P. Chavez, "Feature Selection for Job Matching Application using Profile Matching Model", IEEE 4th International Conference on Computer and Communication Systems (ICCCS), pp. 2-4 , Feb. 2019.

[12] Nunik Destria Arianti, Mohamad Irfan, Undang Syaripudin, Dina Mariana, Neny Rosmawarni, Dian Sa'adillah Maylawati, "Porter Stemmer and Cosine Similarity for Automated Essay Assessment", 5th International Conference on Computing Engineering and Design (ICCED)", pp. 1-5, 2019, doi:10.1109/icced46541.2019.91610.

[13] Garima Koushik, Dr. Prof. K. Rajeswari, Mr. Suresh Kannan Muthusamy, "Automated Hate Speech Detection on Twitter. 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)", pp. 421- 425, 2019, doi:10.1109/iccubea47591.2019.912.

[14] Ravali Boorugu, Dr. G. Ramesh, "A Survey on NLP based Text Summarization for Summarizing Product Reviews", Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 352- 356, 2020, doi: 10.1109/ICIRCA48905.2020.9183355.

[15] Tanya V. Yadalam,Vaishnavi, M.Gowda, Vanditha Shiva Kumar, Disha Girish, Namratha M, "Career Recommendation System Using Content Based Filtering", International Conference on Communication and Electronics Systems (ICCES), pp. 2-5, June 2020, doi: 10.1109/ICCES48766.2020.9137992

[16] Jeevamol Joy and Renumol V G, "Comparison of Generic Similarity Measures in E-learning Content Recommender System in Cold-Start Condition", IEEE Bombay Section Signature Conference (IBSSC)", pp. 175-179, 2020, doi:10.1109/ibssc51096.2020.9332162.

[17] M. Alamelu, D.Sathish Kumar, R. Sanjana, J.Subha Sree, A.Sangeerani Devi,
D. Kavitha, "Resume Validation and Filtration using Natural Language Processing", 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), pp. 412-430, 2021, doi:10.1109/IEMECON53809.2021.9689075.

[18] Swaranjali Jugran, Ashish Kumar, Bhupendra Singh Tyagi, Mr. Vivek Anand,"Extractive Automatic Text Summarization using SpaCy in Python & NLP", International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 582-585, 2021, doi:10.1109/icacite51222.2021.9404712.

[19] S. Prathyusha, S. Jadhav, K. Kommu, M.S. Velpuru, "Text summarization using NLTK with GUI interface", 4th Smart Cities Symposium (SCS 2021), pp. 435-442, 2021, doi: 10.1049/icp.2022.0369.

[20] Meenakshi A. Thalor, "A Descriptive Answer Evaluation System Using Cosine Similarity Technique", International Conference on Communication information and Computing Technology (ICCICT), pp. 1-12, 2021, doi: 10.1109/ICCICT50803.2021.9510170.

[21] Mohammad Alobed;Abdallah M M Altrad;Zainab Binti Abu Bakar, " A Comparative Analysis of Euclidean,

Jaccard and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring", Fifth International Conference on Information Retrieval and Knowledge Management (CAMP), pp.70-74, 2021, doi:10.1109/camp51653.2021.9498119.