SMART TALENT RESUME RANKER

Enrollment Nos	:	17104011	17104012	17104018
Name of Students	:	Kapil Israni	Ayush Nagar	Akshara Nigam
Name of Supervisor	:	Mr. Mahendra Gurve		

December, 2020

Submitted in the partial fulfilment of the Degree of

Bachelor of Technology

in

Information Technology

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Table of Contents

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement has been made in the text.

Place : Noida	Signature :
Date :	Name : Kapil Israni, Ayush Nagar, Akshara Nigam
	Enrollment no : 17104011, 17104012, 17104018

CERTIFICATE

This is to certify that the work titled "**Smart Talent Resume Ranker"** submitted by **Kapil Israni, Ayush Nagar** and **Akshara Nigam** in partial fulfillment for the award of B.Tech (I.T) of Jaypee Institute of Information Technology Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other university or institute for the award of any other degree or diploma.

Signature of the Supervisor
Name of the Supervisor : Mr. Mahendra Gurve
Designation :
Date :

ACKNOWLEDGEMENT

First and foremost we would like to thank our mentor Mr. Mahendra Gurve of Jaypee Institute of Information Technology, Noida for guiding us thoughtfully and efficiently throughout this project, giving us an opportunity to work at our own pace along our own lines, while providing us with very useful directions whenever necessary.

We would also like to thank our friends and classmates for being great sources of motivation and for providing us encouragement throughout the length of this project. We offer our sincere thanks to other persons who knowingly or unknowingly helped us in this project.

Signature of Students :

Name of Student:	Kapil Israni	Ayush Nagar	Akshara Nigam
Enrollment Number:	17104011	17104012	17104018

Date :

SUMMARY

We often notice that during the recruitment process, it becomes a very tedious task for the company to select the candidates just from their resumes. Even though the job description explicitly states the particular skills they are looking for, still the pile of resumes to screen is huge. To overcome this problem we have built a **Smart Talent Resume Ranker.**

The main idea is to recommend the best candidates suitable for a particular job. This is done first by classifying the resumes under various categories like Software Developer, Web Developer, HR, Finance, Sales , Medical and then ranking the resumes from those classified categories. This assists in minimizing the effort required by employers to manage and organize resumes, as well as to screen out irrelevant candidates. It is achieved by using K-nearest neighbours for classification and Similarity Matching between job description and resumes for ranking them.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 General Introduction : The main idea is to recommend the best candidates suitable for a particular job. This is done first by classifying the resumes under various categories like Software Developer, Web Developer, HR, Finance, Sales , Medical and then ranking the resumes from those classified categories. This assists in minimizing the effort required by employers to manage and organize resumes, as well as to screen out irrelevant candidates.

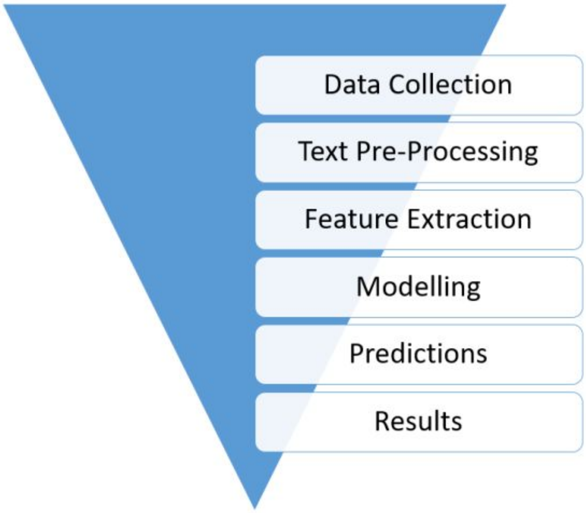
1.2 Problem Statement : To categorize the best suited candidates resumes for a particular job description.

1.3 Significance/Novelty of the Project : This project optimizes the conventional resume matching methods which parse the entire set of resumes and do the matching between the job description and the resumes. In this project a resume once fetched into the system is parsed and its job profile category is determined via the machine learning model which is then used to match with a Job Description looking for similar profile people.

1.4 Empirical Study : Due to the increasing growth in online recruitment, traditional hiring methods are becoming inefficient. This is due to the fact that job portals receive enormous numbers of unstructured resumes - in diverse styles and formats - from applicants with different fields of expertise and specialization. Therefore, the extraction of structured information from applicant resumes is needed not only to support the automatic screening of candidates, but also to efficiently route them to their corresponding occupational categories. This assists in minimizing the effort required by employers to manage and organize resumes, as well as to screen out irrelevant candidates. In this paper, we present Smart Talent Resume Matcher that exploits an integrated knowledge base for carrying out the classification task. Unlike conventional systems that attempt to search globally in the entire space of resumes and job posts, JRC matches resumes that only fall under their relevant occupational categories. To demonstrate the effectiveness of the proposed system, we have conducted several experiments using a real-world recruitment dataset.

Additionally, we have evaluated the efficiency and effectiveness of proposed systems against state-of-the-art online recruitment systems.

1.5 Brief Description of the Solution Approach :



(Fig 1 : Overview of the Model)

Resume Parsing : Whenever a new resume is fetched it is first cleaned and then fed to the training model. Pre-processing of the text is done by :-

- **Stemming :** Words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is an equal to or smaller form of the word. There are different algorithms that can be used in the stemming process, but the most common in English is Porter stemmer.

- **Lemmatization :** It involves resolving words to their dictionary form. It requires a lot more knowledge about the structure of a language, it's a much more intensive process than just trying to set up a heuristic stemming algorithm.

- **Text Cleaning :** This involves removal of punctuation marks and special characters and . The text is converted to lower case letters for better matching.

- **Skill Extraction** : This uses Named Entity Recognition from the existing data set of skills which is used to extract skills from the resume.

- **Relevant Information Extraction :**Other information such as phone number and email address is extracted from resume via regex functions.

Resume profile classification : After pre-processing, KNN algorithm is used to classify the resume profile and an array of classified profiles is derived from the KNN model on a particular resume.

Insertion into Database : This parsed resume is then inserted into the MongoDB database so that there is no need to parse the resume again.

Job Description Parsing : Similar to Resume a Job Description is parsed and appropriate fields are extracted such as skills , job description and job profile. Parsed JD is then inserted into the database.

Resume Matching : This basically shows the top best matched resume to the given job description . All the resumes with the given profile are fetched from the database and then score is calculated on the basis of similarity between skills of fetched resume and the job description.

1.6 Comparison with Existing Approaches : Many approaches and techniques have been proposed for addressing the e-recruitment challenges. In this context, some approaches attempt to overcome these issues associated with the matching process between candidate resumes and their corresponding job offers, while others attempt to classify resumes and job posts prior to starting the matching process.

Coming to another case study, a website called Top Resume has employed the usage of techniques like Natural Language Processing to analyze a prospective job seeker's resume. Here, the task of the candidate is to only upload their resume on the portal. With the help of Natural language Processing, only the text data is extracted from the resume and the strength of the candidate's profile is displayed in terms of percentage. While some papers have completely employed the use of API's for skill set recognition and classification while matching tasks are not done. Further Cosine Similarity that was used for matching job description and resume, it had flaws while matching skills.

So our model employs the use of Spacy pipeline along with Natural language processing of Named Entity Recognition (NER) for parsing the resume, using Selenium to extract skills from a renowned career website and the classifying the resumes to avoid performance issues and storing those results in the database. Now whenever a Job Description would be provided we would know the role (category) for which they are looking for candidates and from the database we would extract information, do the similarity matching between the skills and from the scores obtained sort to find top-k.

2. LITERATURE SURVEY

2.1 Summary of Paper

Paper I

Paper Title	A Job Recommendation Method Optimized by Position Descriptions and Resume Information.
Author	Peng Yi, Cheng Yang, Chen Li, Yingya Zhang
Publisher	IEEE
Year	2016
Summary	Job recommendation algorithms which utilize recommendation methods to filter positions that do not meet the requirements and recommend the proper positions for job hunters play an important role in the recruitment websites. Based on the analysis of real recruitment data and the comparison of the existing recommendation methods, item based collaborative filtering algorithm has been used as the basic algorithm for a job recommendation. This paper produced an optimization algorithm to improve the accuracy of job recommendations. Historical delivery weight calculated by position descriptions and similar user weight calculated by resume information were added as two influencing factors in the preference prediction. The experiments tested on real recruitment data have shown that the optimization algorithm has greatly improved the final recommendation result. The F1-score of the optimized algorithm produced 9.6% better results than the basic algorithm.

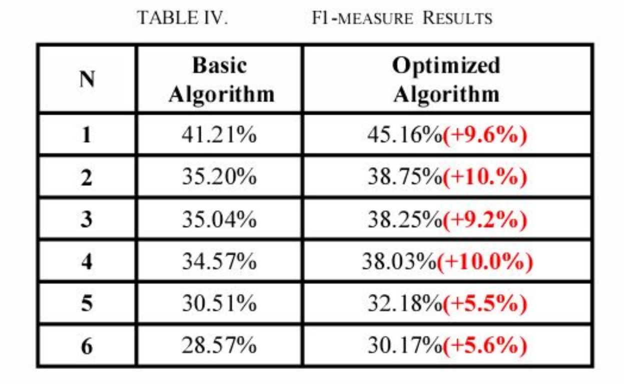


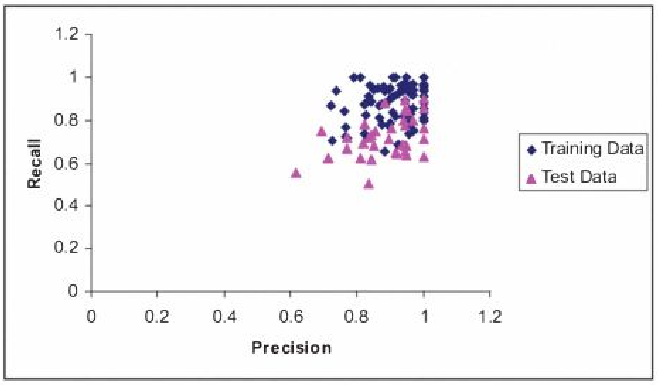
TABLE IV. F1-MEASURE RESULTS

N	Basic Algorithm	Optimized Algorithm
1	41.21%	45.16%**(+9.6%)**
2	35.20%	38.75%**(+10.%)**
3	35.04%	38.25%**(+9.2%)**
4	34.57%	38.03%**(+10.0%)**
5	30.51%	32.18%**(+5.5%)**
6	28.57%	30.17%**(+5.6%)**

Paper II

Paper Title	Web Application for Screening Resume
Author	Sujit Amin, Nikita Jayakar, Sonia Sunny, Pheba Babu, M.Kiruthika, Ambarish Gurjar
Publisher	IEEE, International Conference on Nascent Technologies in Engineering
Year	2019
Summary	This paper focuses on a web application for screening Resumes of various candidates. The recruiters from various companies can post the details of the job openings available in their respective companies. The interactive web application allows the job applicants to submit their resume and apply for the job postings they may still be interested in. The resumes submitted by the candidates are then compared with the job profile requirement posted by the company recruiter by using techniques like machine learning and Natural Language Processing (NLP). Scores can then be given to the resumes and they can be ranked from highest match to lowest match. This ranking is made visible only to the company recruiter who is interested to select the best candidates from a large pool of candidates. The scores as well as the rank list will only be visible to the recruiter and not to the candidates. The recruiter can then make an informed decision on when to select for the next round of the hiring process. The job description text file is retrieved from the database. After that, the relevant entities of the candidate resume text file as well as the job description text file are then compared and a score is assigned to the candidate.

Paper III

Paper Title	Automatic Extraction of Usable Information from Unstructured Resumes to Aid Search
Author	Sunil Kumar Kopparapu
Publisher	IEEE
Year	2010
Summary	This paper describes a system for automated resume information extraction to support rapid resume search and management. The system is capable of extracting several important informative fields from a free format resume using a set of natural language processing (NLP) techniques. A working system is described, for automatic resume management. The system is capable of extracting six major fields of information. Experimental results carried out on a large number of resumes show that the proposed system can handle a large variety of resumes in different document formats with a precision of 91% and a recall of 88%.  *Figure 5 Precision and recall plot for train dataset (♦) and test datasets (▲).*

Paper IV

<table>
<tr><td><b>Paper Title</b></td><td>A Machine Learning approach for automation of Resume Recommendation System</td></tr>
<tr><td><b>Author</b></td><td>Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia</td></tr>
<tr><td><b>Publisher</b></td><td>International Conference on Computational Intelligence and Data Science, Elsevier</td></tr>
<tr><td><b>Year</b></td><td>2019</td></tr>
<tr><td><b>Summary</b></td><td>The System produced in this paper, works with a large number of resumes first for classifying the right categories, then as per the job description top candidates would be ranked using Content-based recommendation using cosine similarity and KNN to identify the CV’s that are nearest to the provided job description. The classification was done using four different models and their accuracy score was recorded. 1) Random Forest 2) Multinomial Naive Bayes 3) Logistic Regression 4) Linear SVC

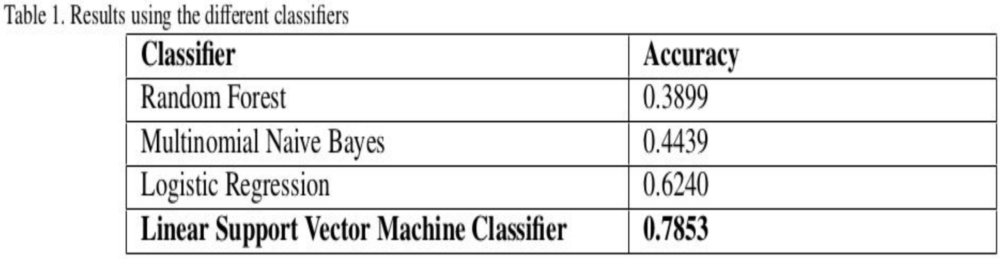
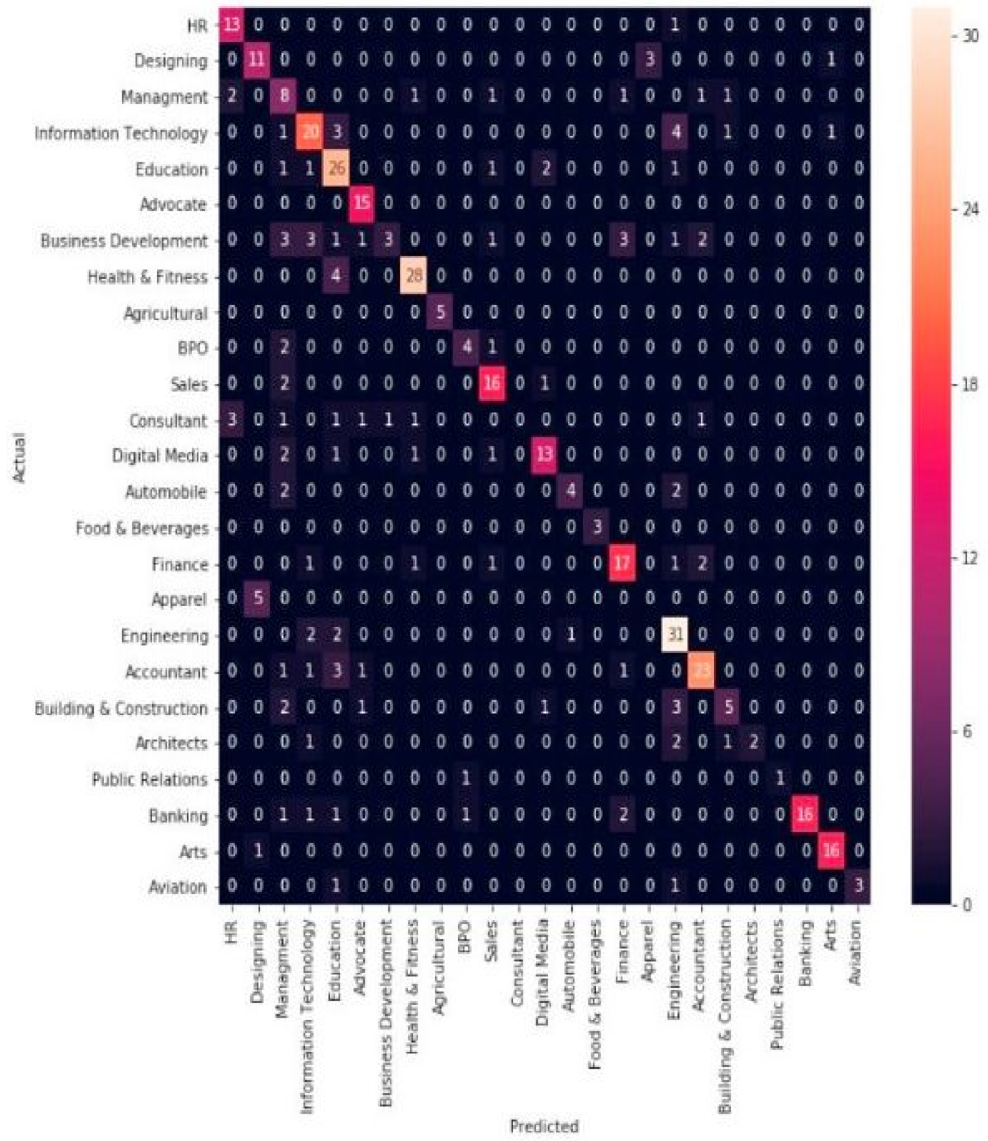



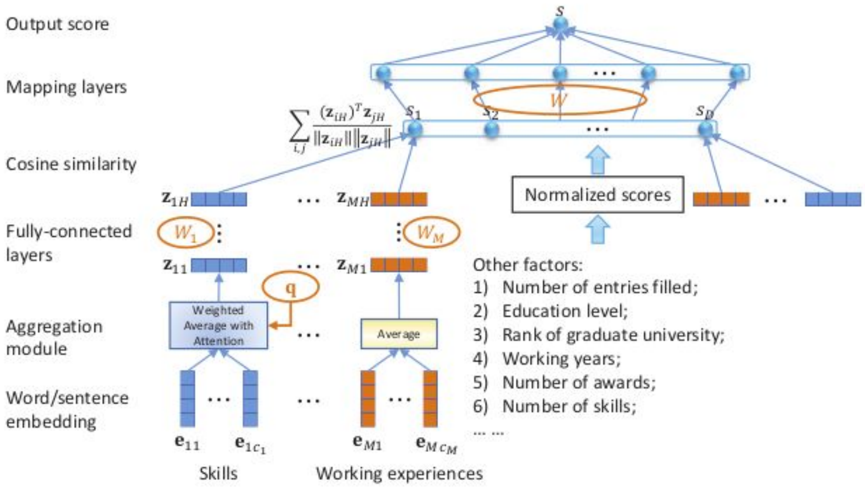
Table 1. Results using the different classifiers
<table>
<tr><th>Classifier</th><th>Accuracy</th></tr>
<tr><td>Random Forest</td><td>0.3899</td></tr>
<tr><td>Multinomial Naive Bayes</td><td>0.4439</td></tr>
<tr><td>Logistic Regression</td><td>0.6240</td></tr>
<tr><td><b>Linear Support Vector Machine Classifier</b></td><td><b>0.7853</b></td></tr>
</table>
The confusion matrix for the prediction done through Linear SVC is given below :</td></tr>
</table>



	HR	Designing	Managment	Information Technology	Education	Advocate	Business Development	Health & Fitness	Agricultural	BPO	Sales	Consultant	Digital Media	Automobile	Food & Beverages	Finance	Apparel	Engineering	Accountant	Building & Construction	Architects	Public Relations	Banking	Arts	Aviation
HR	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Designing	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0
Managment	2	0	8	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0
Information Technology	0	0	1	20	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0	0	1	0
Education	0	0	1	1	26	0	0	0	0	0	1	0	2	0	0	0	0	1	0	0	0	0	0	0	0
Advocate	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Business Development	0	0	3	3	1	1	3	0	0	0	1	0	0	0	0	3	0	1	2	0	0	0	0	0	0
Health & Fitness	0	0	0	0	4	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agricultural	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BPO	0	0	2	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sales	0	0	2	0	0	0	0	0	0	0	16	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Consultant	3	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Digital Media	0	0	2	0	1	0	0	1	0	0	1	0	13	0	0	0	0	0	0	0	0	0	0	0	0
Automobile	0	0	2	0	0	0	0	0	0	0	0	0	0	4	0	0	0	2	0	0	0	0	0	0	0
Food & Beverages	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
Finance	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	17	0	1	2	0	0	0	0	0	0
Apparel	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Engineering	0	0	0	2	2	0	0	0	0	0	0	0	0	1	0	0	0	31	0	0	0	0	0	0	0
Accountant	0	0	1	1	3	1	0	0	0	0	0	0	0	0	0	1	0	0	23	0	0	0	0	0	0
Building & Construction	0	0	2	0	0	1	0	0	0	0	0	0	1	0	0	0	0	3	0	5	0	0	0	0	0
Architects	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	2	0	0	0	0
Public Relations	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Banking	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	16	0	0
Arts	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0
Aviation	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3

Paper V

Paper Title	A Learning-based Framework for Automatic Resume Quality Assessment (RQA)
Author	Yong Luo, Huaizheng Zhang, Yongjie Wang, Yonggang Wen, Xinwen Zhang
Publisher	IEEE, International Conference on Data Mining
Year	2018
Summary	This paper throws light on the fact that from the talent perspective, many recruiters may want to know whether a resume is good enough or not. Therefore, the tool was developed to assess the quality of each resume automatically. Although there exist some resume quality assessment (RQA) websites (e.g., http://rezscore.com/), their underlying assessment schemes or algorithms are unknown and there is no public dataset for model training and evaluation. To tackle these issues, the authors had built a dataset and developed a general model for the same. The diagram of the system is given below :

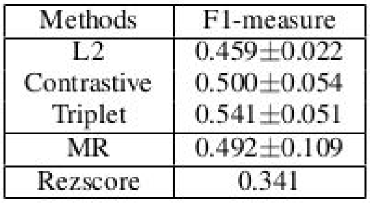


From the system designed, following conclusions can be drawn that :

1) Learning adaptive weights using the attention scheme to aggregate multiple embeddings is superior to the simple average in general.

2) Either using the designed pair/triplet-based loss or adding a regularization term to utilize unlabeled data can improve the performance, it seems that the model based on triplet loss achieves the best performance overall.

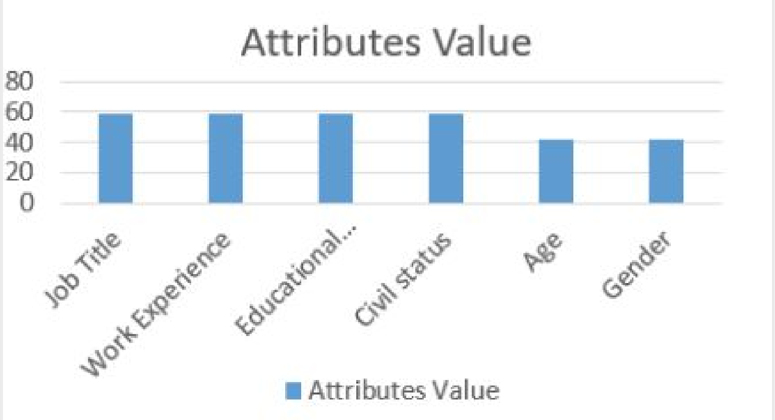
TABLE I
A COMPARISON OF OUR MODELS WITH THE OTHER APPROACH IN TERMS OF F1-MEASURE.



Methods	F1-measure
L2	0.459±0.022
Contrastive	0.500±0.054
Triplet	0.541±0.051
MR	0.492±0.109
Rezscore	0.341

Paper VI

Paper Title	Feature Selection for Job Matching Application using Profile Matching Model
Author	Leah G. Rodriguez, Enrico P. Chavez
Publisher	IEEE, 4th International Conference on Computer and Communication Systems
Year	2019
Summary	This paper aims to extract the relevant information from resumes and analyze it based on the different attributes. With the identification of the attributes, the proposed system is directed to adopt a clustering algorithm to match the profile of the job seekers against the requirements of the job posted by the prospect employers. Computing similarity scores between two profiles was the important task. For the similarity score, the values of common attributes in both profiles are extracted and their similarity scores are then computed and compared. Then, the obtained similarity scores are tuned in order to have more realistic scores that take into consideration the importance assigned to each attribute. By doing so, the new similarity value will tend to increase or decrease depending on the importance of each attribute. This tuning is an attribute based operation that outputs a new similarity score to each attribute by applying a weight to the computed similarity scores. The below graph shows the ranking of the attributes :

	Attributes Value 80 60 40 20 0 Job Title Work Experience Educational... Civil status Age Gender Attributes Value

Paper VII

<table>
<tr><td><b>Paper Title</b></td><td>A Research of Job Recommendation System Based on Collaborative Filtering</td></tr>
<tr><td><b>Author</b></td><td>Yingya Zhang, Cheng Yang, Zhixiang Niu</td></tr>
<tr><td><b>Publisher</b></td><td>IEEE, 7th International Symposium on Computational Intelligence and Design</td></tr>
<tr><td><b>Year</b></td><td>2014</td></tr>
<tr><td><b>Summary</b></td><td>This paper contrasts between user-based and item-based collaborative filtering algorithms to choose a better performed one. They take background information including students' resumes and details of recruiting information into consideration, bring weights of co-apply users (the users who had applied the candidate jobs) and weights of student used-liked jobs into the recommendation algorithm. It also takes into consideration four Methods of Similarity Calculation (i) Cosine Similarity (ii) Tanimoto Coefficient (iii) Log Likelihood (iv) The City Block Distance. The accuracy for both the filtering methods is given as follows:

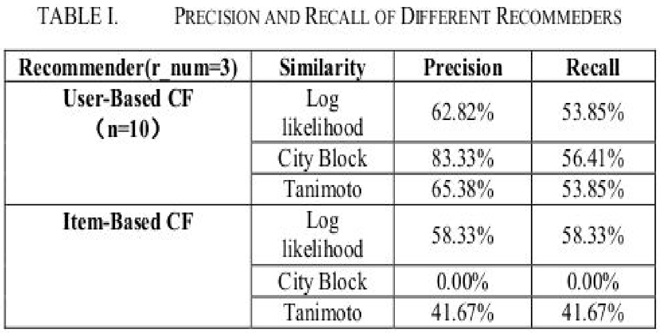
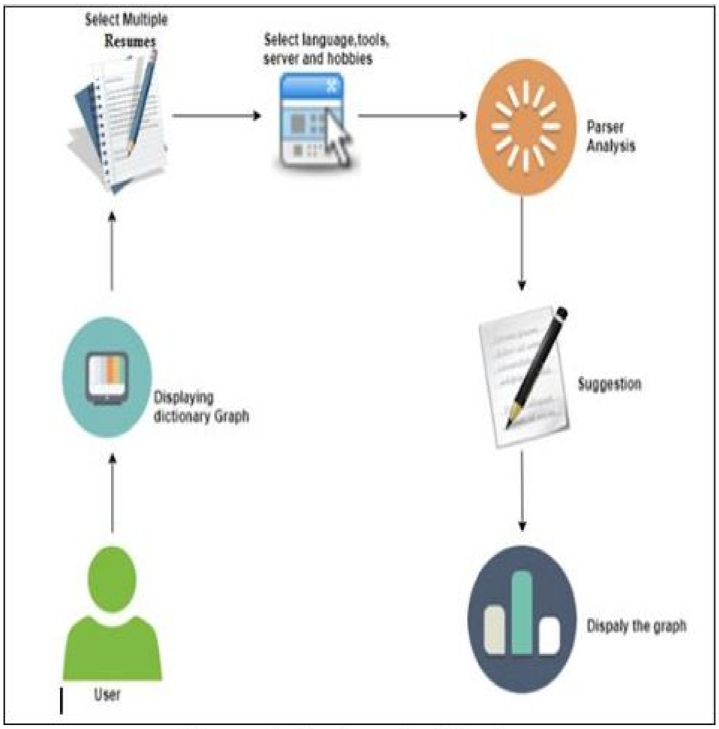



TABLE I. PRECISION AND RECALL OF DIFFERENT RECOMMEDERS
<table>
<tr><th>Recommender(r_num=3)</th><th>Similarity</th><th>Precision</th><th>Recall</th></tr>
<tr><td rowspan="3">User-Based CF
（n=10）</td><td>Log likelihood</td><td>62.82%</td><td>53.85%</td></tr>
<tr><td>City Block</td><td>83.33%</td><td>56.41%</td></tr>
<tr><td>Tanimoto</td><td>65.38%</td><td>53.85%</td></tr>
<tr><td rowspan="3">Item-Based CF</td><td>Log likelihood</td><td>58.33%</td><td>58.33%</td></tr>
<tr><td>City Block</td><td>0.00%</td><td>0.00%</td></tr>
<tr><td>Tanimoto</td><td>41.67%</td><td>41.67%</td></tr>
</table>
</td></tr>
</table>

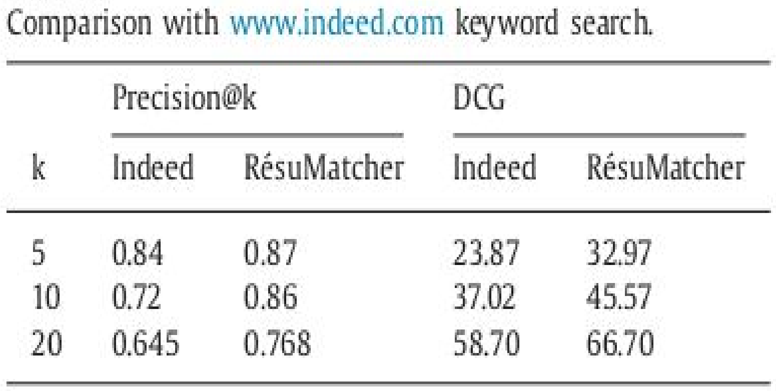
Paper VIII

Paper Title	Resume Parser
Author	Aneesha T Ibrahim, Annette J K, Geethika S, Archana Naik
Publisher	International Journal of Innovations in Engineering and Technology (IJIET)
Year	2018
Summary	This paper discusses developing a parsing application, for resumes received in multiple formats which include .docx, .doc, and .pdf. This application reduces the time and manual effort of searching through the multiple resumes for choosing the suitable resumes. The technique involved is known as resume parsing. Other names include, resume extraction, CV parsing, CV extraction, which allows the automated storage and analysis of resume information. Multiple resumes are uploaded into parsing software and the information is extracted so that it can be sorted and searched. Resume parsers first analyzes a resume, and then extracts the desired information. After the resume has been analyzed, a recruiter can specify the job skills required and get a list of relevant resumes as the output. Some parsers provide semantic search, which adds context to the search terms and tries to understand the intent in order to make the results more reliable and comprehensive.

	 Select Multiple Resumes Select language,tools, server and hobbies Parser Analysis Suggestion Dispaly the graph Displaying dictionary Graph User Figure 1: System Architecture

Paper IX

Paper Title	ResuMatcher: A personalized resume-job matching system
Author	Shiqiang Guo, Folami Alamudun, Tracy Hammond
Publisher	Elsevier
Year	2016
Summary	The proposed model, intelligently extracts the qualifications and experience of a job seeker directly from his/her resume, and relevant information about the qualifications and experience requirements of job postings. Using a novel statistical similarity index, ResuMatcher returns results that are more relevant to the job seekers experience, academic, and technical qualifications, with minimal active user input. The system comprises the following main components :- Job Data Processor, Search Interface, and the Resume Matcher. 1) The Job Data Processor executes a daily batch job of crawling the web for different job postings and building the Jobs Model. 2) The Search Interface provides an interactive front-end from which it accepts the resume of users and builds a Resume Model. 3) The Resume Matcher receives as input, a resume object from the Search Interface, and queries the job database using a novel similarity method to retrieve the most relevant jobs. The similarity between a resume object and a job object is calculated as a weighted sum of the computed features. The comparison is given below :-

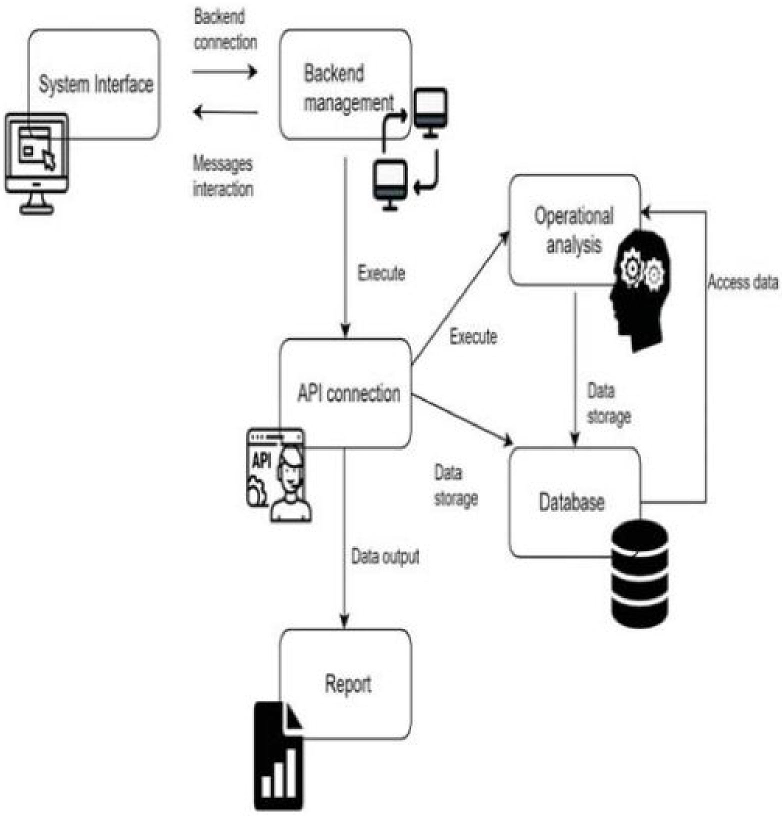


Comparison with www.indeed.com keyword search.

	Precision@k		DCG	
k	Indeed	RésuMatcher	Indeed	RésuMatcher
5	0.84	0.87	23.87	32.97
10	0.72	0.86	37.02	45.57
20	0.645	0.768	58.70	66.70

Paper X

Paper Title	A Resume Evaluation System Based on Text Mining
Author	Yi-Chi Chou, Chun-Yen Chao, Han-Yen Yu
Publisher	IEEE
Year	2019
Summary	This study developed an AI-based interviewing system to reduce the loss of talent caused by the emotional reactions and subjectivity of interviewers when viewing resumes. The designed system performs the function of resume assessment and explores the personality traits of candidates by classifying them into four dimensions of soft power, namely dominance, influence, steadiness, and compliance (DISC) after assessing the submitted electronic resumes. This system also assesses three dimensions of competence, namely education and experience, skills, and personality traits, which are indicated by the information contained in a resume. Finally, the designed system quantifies the aforementioned DISC data and three competency dimensions by scoring each resume. The results are then compiled into a report that contains the personal analysis, ranking, and distribution forecast for the candidate in question.

	 System Interface Backend connection Messages interaction Backend management Execute Operational analysis Access data Execute API connection API Data storage Data storage Database Data output Report Fig. 2. Backend architecture diagram

2.2 Integrated Summary :

SNo.	Paper	Algorithm/Model	Challenges	Drawbacks
1	Paper - 1	Item-based, user-based recommendation system User-similarity, major similarity	Due to the limited data, it is difficult to calculate the similarity of major just by the name of major.	It is a static based algorithm.
2	Paper - 2	NLP for data extraction. Spacy Pipeline for scoring the candidate resume.	To reduce the time complexity of comparing and matching the candidate's profile and job posted.	It converts the input data first into JSON format to be passed to the NLP pipeline for matching.
3	Paper - 3	A mix of NLP techniques and heuristics were used to build information extraction modules to aid extraction of useful information from resumes. The knowledge base was created using reference resumes	Automatic extraction of information from resumes with high precision and recall is not an easy task essentially because of the non-standardization of resume structure. In spite of constituting a restricted domain, resumes can be written	Extracting some information using only HR-XML

		and the system was tested on a large number of resumes which was not part of the reference resumes	in multitude of formats (e.g. structured tables or plain texts) and in different file types (e.g txt, .pdf, .doc(x) etc.)	
4	Paper - 4	It uses Content-based collaborative recommendation using cosine similarity and KNN for identifying the CV's closest to the job profile.	Different structure and format of every CV. Mapping the CV to the right job description	i) Model takes CVs in CSV format. ii) Generation of a summary using genism library might cause loss of important information due to compression of the text.
5	Paper - 5	Multi-layer neural network, Cosine Similarity	Since there is no public algorithm for RQA, we submit our labeled resumes to a website (http://rezscore.com/) , which can assign a grade for each resume.	Lacking a larger corpus that includes job-post information and identify more useful features for RQA
6	Paper - 6	Feature selection, Cosine Similarity, Weighted Similarity	One of the most challenging tasks of this type of job matching	Lack of training data set from job seekers and company, need

		for ranking.	was that there was a bulk of information to coordinate against and it was in free form.	to conduct tests of the clustering model to verify reliability and performance of the job matching system
7	Paper - 7	Collaborative Filtering, Cosine Similarity, Tanimoto Coefficient, Log Likelihood, The city block distance	Collaborative Filtering approaches often suffer from three problems: cold start, scalability and sparsity.	It's a comparison between the two kinds of recommendation systems so it doesn't rank the candidates, it just compares based on the accuracy.
8	Paper - 8	NLP for parsing the document and Regex Matching to extract specific information like Location, Phone Number etc.	The ambiguity problem caused by the polysemy of the phrases, in expressing the skills in the text of a resume	There is a limitation in the number of resumes that can be processed at a time. Further work involves overcoming that barrier and making it a highly efficient parser. Also the system needs to be made more large-scale

9	Paper - 9	NLP, Cosine Similarity	A significant amount of ambiguity exists between domain specific words and their respective interpretation. Resumes both contain richer and more complex words that cannot be described simply by keywords.	Firstly, they only consider the shortest path between concept pairs. When faced with more complex structures, such as multiple taxonomic inheritance, the accuracy similarity measures is significantly reduced. Another limitation of the path-based approaches is the assumption that all links in the taxonomy have uniform distance.
10	Paper- 10	Jieba, PDFMiner	To be able to create a real-time report and store the huge dataset.	It does most of the work using predefined API calls and may falter at response time of the API.

3. REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 Overall Description of Project : This project gives importance to the skills that an employer demands from a particular profile and via that it categorizes the profile of the resume and parses the resume for skills of a candidate and the information is stored in a database to avoid redundant parsing.

The Job Description is also parsed and all the resumes that have been classified to the job description's profile are brought up from the database and a score is generated based on the matching skills from both the job description and the resume. The top resumes are displayed along with the candidates email id and phone number.

3.2 Requirement Analysis :

- **MongoDB Atlas** : The cloud storage platform to perform mongodb queries and store the collection of resumes under categories.
- **Python 3 :** Python 3.0 is a new version of the language that is incompatible with the 2.x line of releases. You can install all the libraries using the command

```
pip3 install -r requirements.txt
```

where, requirements.txt is the text file containing all the libraries needed for the project to run successfully.

3.3 Solution Approach :

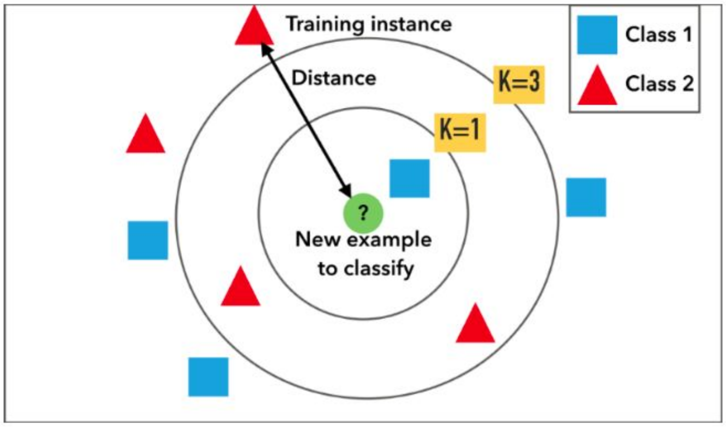
- Firstly, we use a web-scraping tool that is beautiful soup on Python, to create job skills and skill set csv. Job skills csv contains the category and the skills required for that job, while the skill set csv is a global csv consisting of all the possible skills in any field.
- Secondly, resume parsing is done through Spacy and regex matching to extract all the skills in a resume.

- Thirdly, classification using KNN is done and also the nearest neighbours are extracted because a resume can fall under more than one category. Then it is stored in the database.
- Lastly, resume matching extracts the skills from the job description and for the given category, it matches all the resumes in the database. After similarity matching it finds the top-k resumes and returns it.

Resume Filtering : In resume filtering first, the file is read to read its text content using textract. The text in utf-8 format is then processed and used to read the skills line by line to form an array of skills, name of the person, email address of the person and location. These skills are passed to find the top 3 category jobs for which the person's resume seems to best suited. These top 3 categories and skills array are sent to be stored in the database.

Database Storage : The MongoDB Atlas database based schema has three collections categories, jobs and resumes. The resume collection consists of 250+ resumes with skills in the form of an array which are extracted on parsing the resume using resume parser. These skills are used to determine the top-k resume categories for the profile they seem to be best suited. These categories are utilised to filter out the required resumes for the job description.

K-Nearest Neighbours : KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. A technique is said to be non-parametric when it does not make any assumptions on the underlying data distribution. In other words, the model structure is determined from the data. It does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This also means that the training phase is pretty fast . Lack of generalization means that KNN keeps all the training data. To be more exact, all (or most) the training data is needed during the testing phase. KNN Algorithm is based on feature similarity, i.e how closely out-of-sample features resemble our training set determines how we classify a given data point.



(Fig 2 : Working of KNN Algorithm)

Levenshtein Distance : The Levenshtein distance is a string metric for measuring difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other.

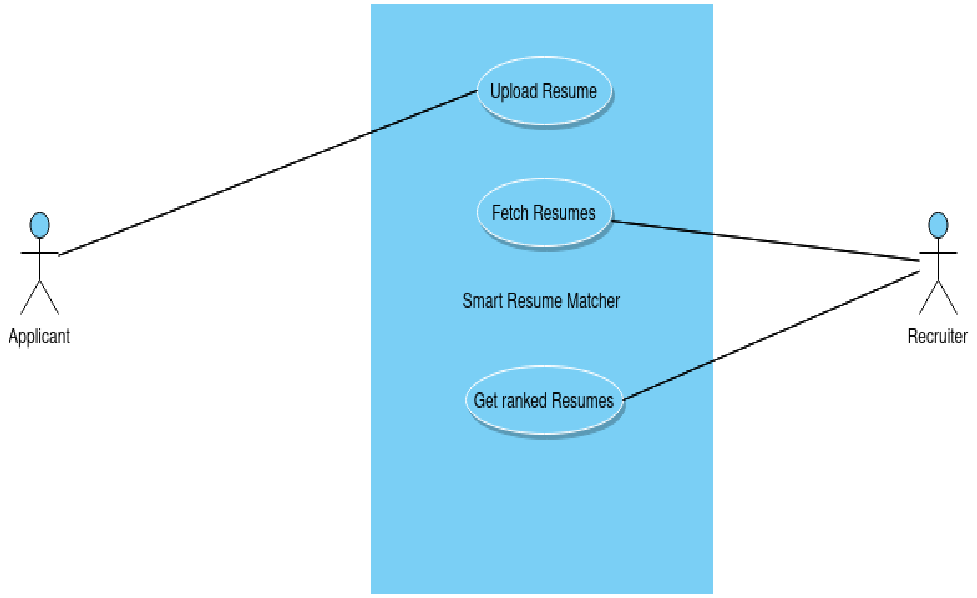
$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j)+1 \\ \operatorname{lev}_{a,b}(i,j-1)+1 \\ \operatorname{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

(Fig 3 : Mathematical Formula)

4. MODELLING AND IMPLEMENTATION DETAILS

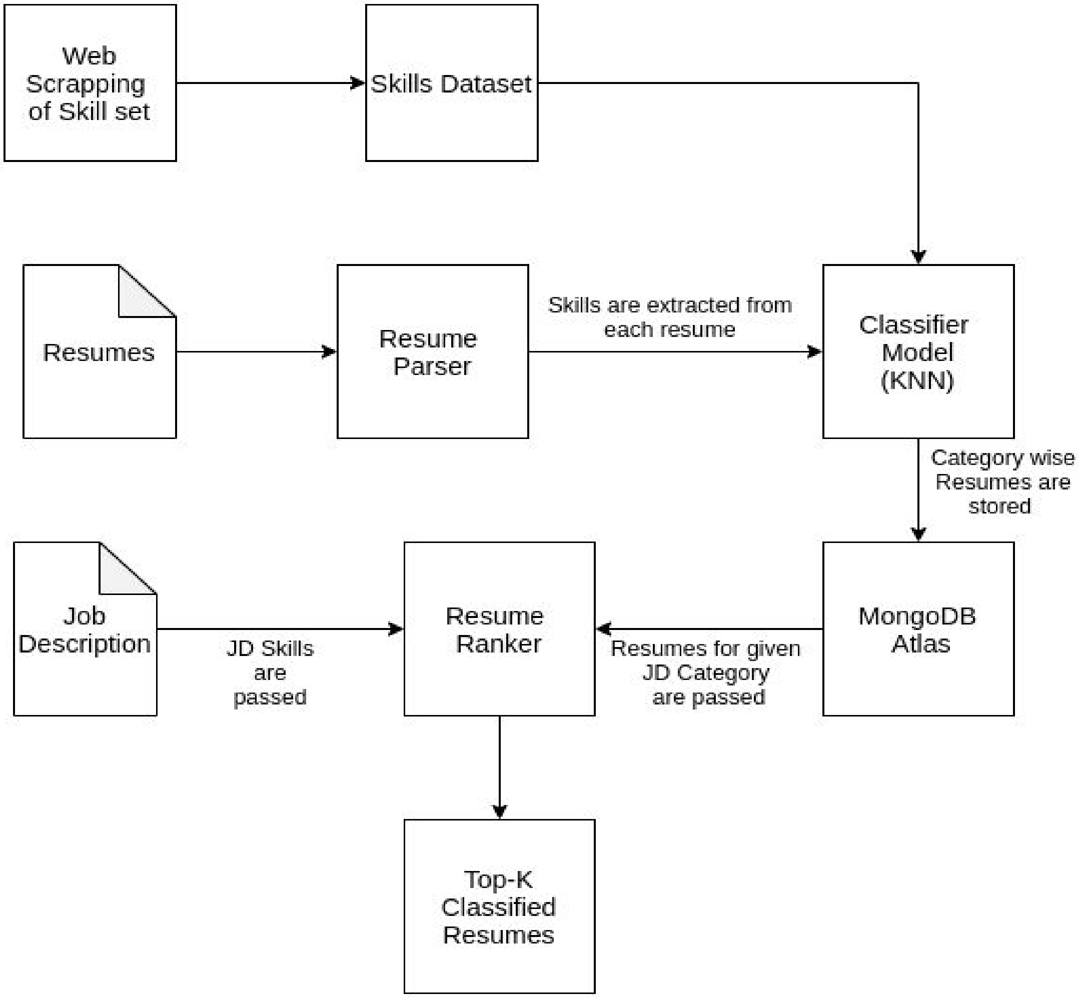
4.1 Design Diagrams :

4.1.1. Use Case Diagram



(Fig 4 : Use Case Diagram of the Model)

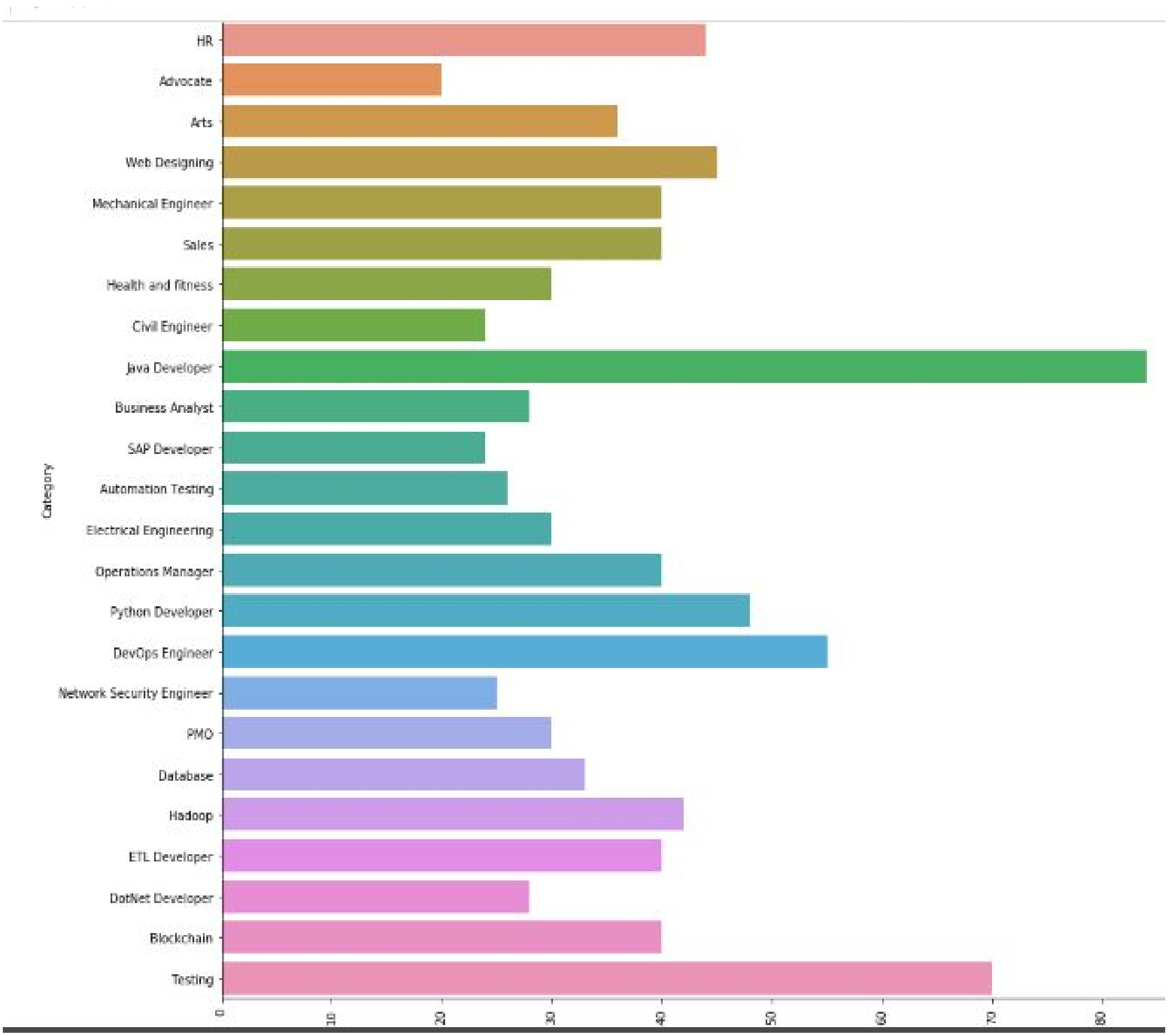
4.1.2 Activity Diagram



(Fig 5 : Activity Diagram of the Model)

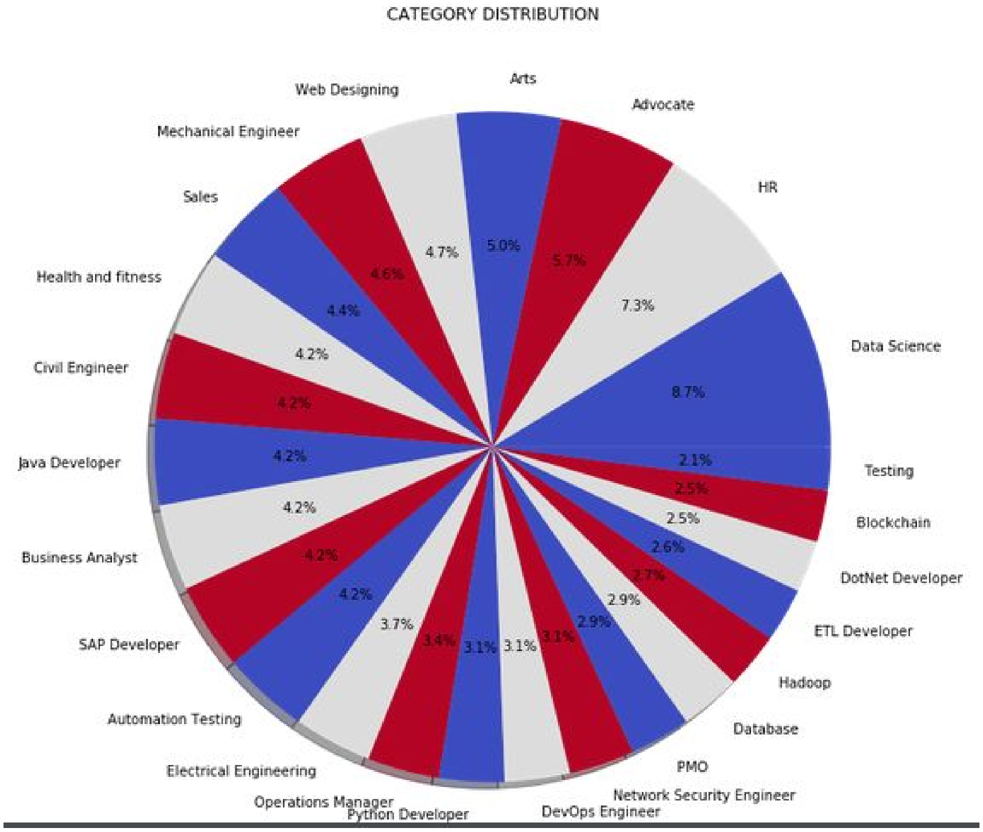
4.2 Implementation Details and Issues : Implementation starts first by analysing the data, then through word count analysis the terms that the parser recognizes, further extracting job descriptions and skills for each category in order to train the model for better results, then accuracy comparison for various models, then storing the resumes and categories in mongodb atlas and lastly find the top-k resumes for various job description on various fields like software development, hotel management, medical etc.

The Horizontal Bar graph below shows the count of each category in the dataset. The Y-axis represents the Categories like HR, Accountant, Web Designing while the X-axis represents the count of resumes in each category.



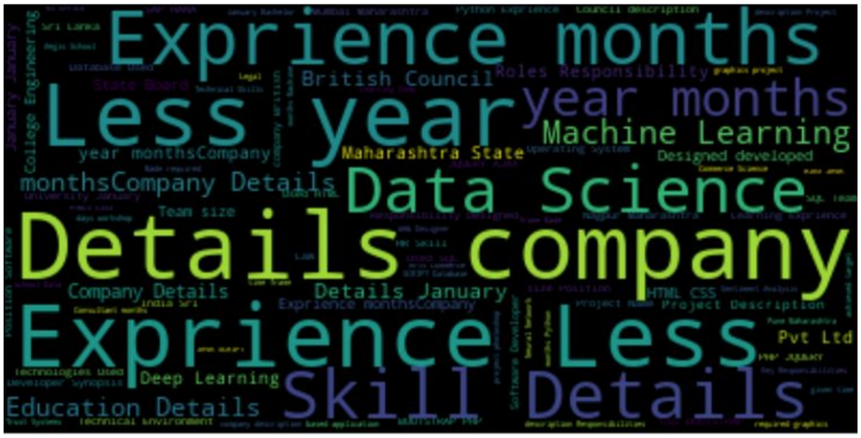
(Fig 6 : Plot of Category with Count)

The below pie chart shows the percentage distribution of each category present in the dataset.



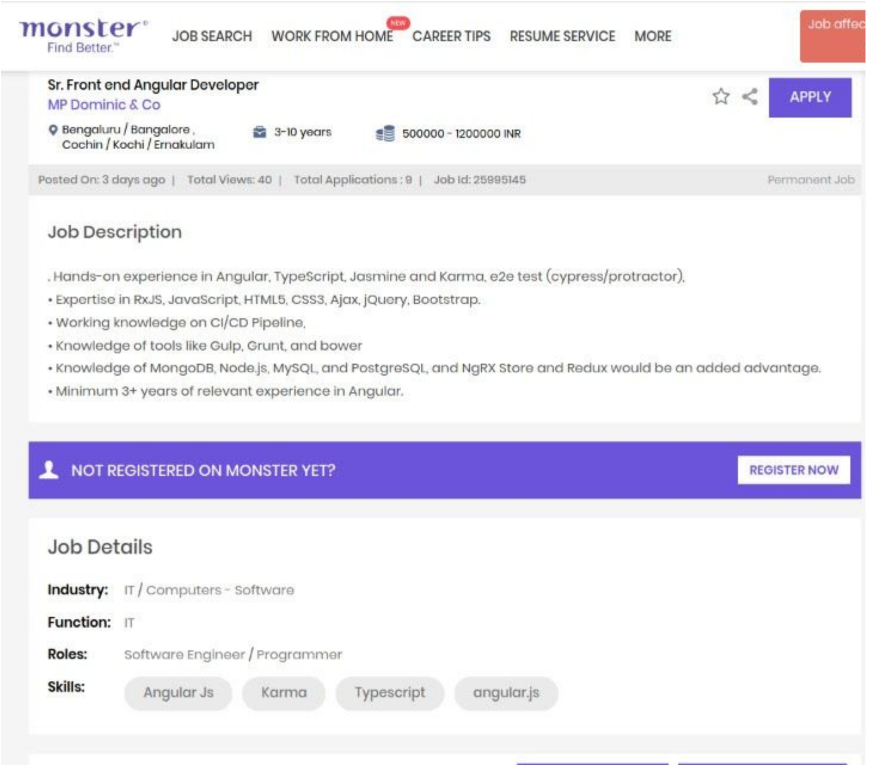
(Fig 7 : Category Distribution Graph)

The word cloud is plotted from the extracted words in all the resume along with the count using Count Frequency.



(Fig 8 : Word Cloud)

The website Monster Jobs was used to scrap all the possible job skills and job description for dataset creation.



Sr. Front end Angular Developer
MP Dominic & Co

Bengaluru / Bangalore , Cochin / Kochi / Ernakulam 3-10 years 500000 - 1200000 INR

Posted On: 3 days ago | Total Views: 40 | Total Applications : 9 | Job Id: 25995145 Permanent Job

Job Description

. Hands-on experience in Angular, TypeScript, Jasmine and Karma, e2e test (cypress/protractor),
- Expertise in RxJS, JavaScript, HTML5, CSS3, Ajax, jQuery, Bootstrap.
- Working knowledge on CI/CD Pipeline,
- Knowledge of tools like Gulp, Grunt, and bower
- Knowledge of MongoDB, Node.js, MySQL, and PostgreSQL, and NgRX Store and Redux would be an added advantage.
- Minimum 3+ years of relevant experience in Angular.

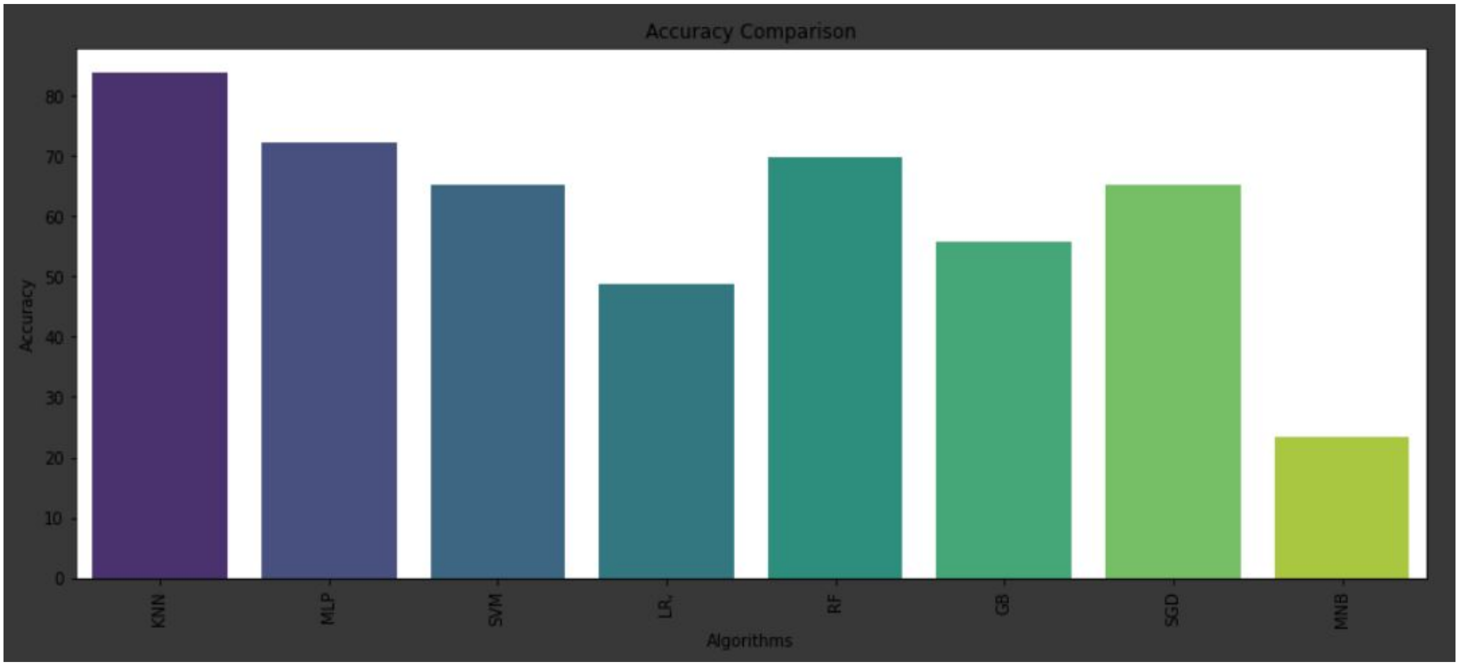
Job Details

Industry: IT / Computers - Software
Function: IT
Roles: Software Engineer / Programmer
Skills: Angular Js Karma Typescript angular.js

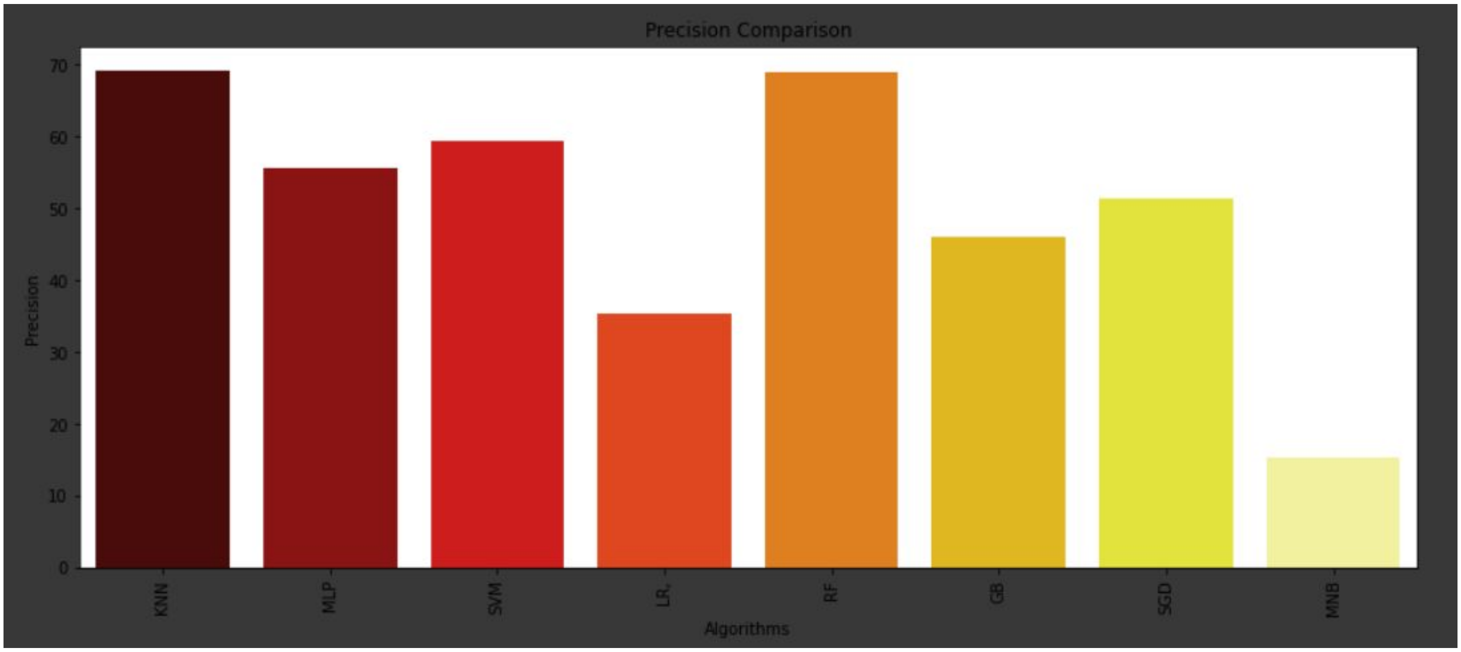
(Fig 9 : Scrapping skills from Monster Jobs)

The below bar plots are used to compare the Accuracy, Precision, Recall and F1 score of various classification Algorithms and this helped us reach a consensus.

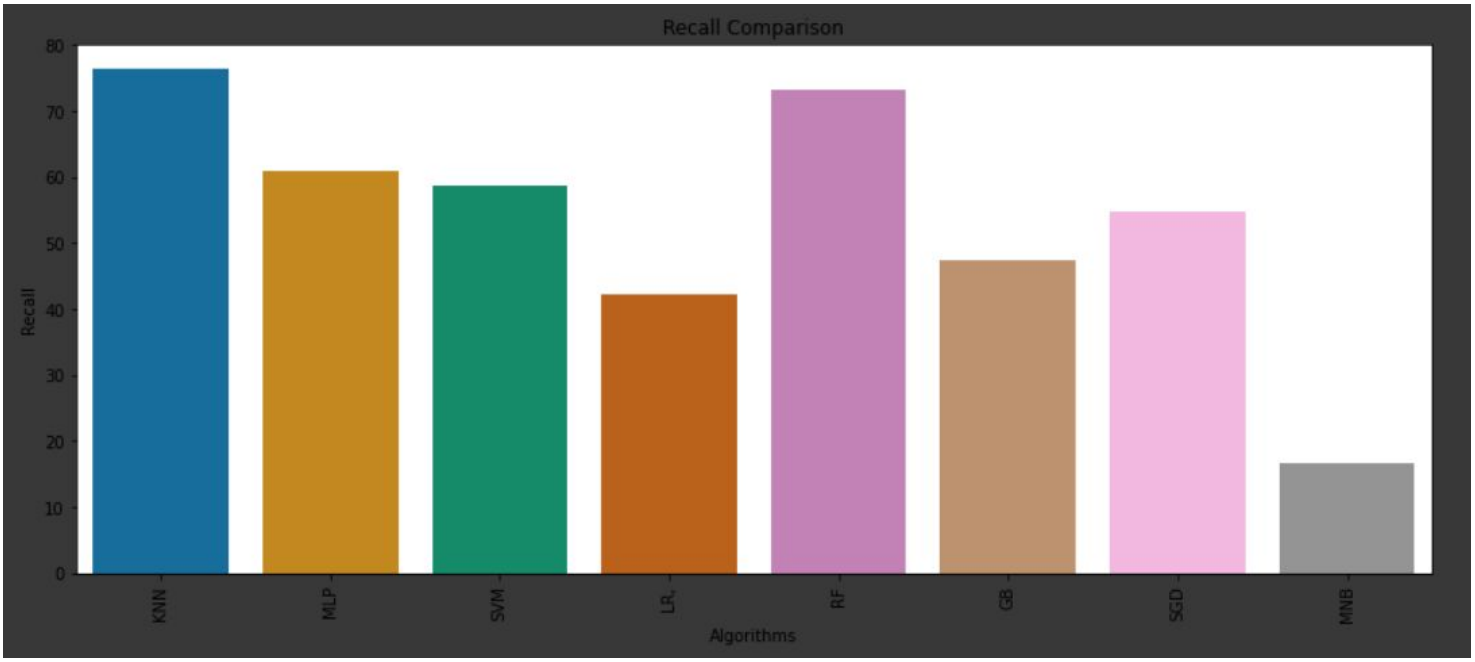
KNN	-	K Nearest Neighbours
MLP	-	Multi Layer Perceptron
SVM	-	Support Vector Machine
LR	-	Logistic Regression
RF	-	Random Forest Classifier
GB	-	Gradient Boosting
SGD	-	Stochastic Gradient Descent
MNB	-	Multinomial Naive Bayes



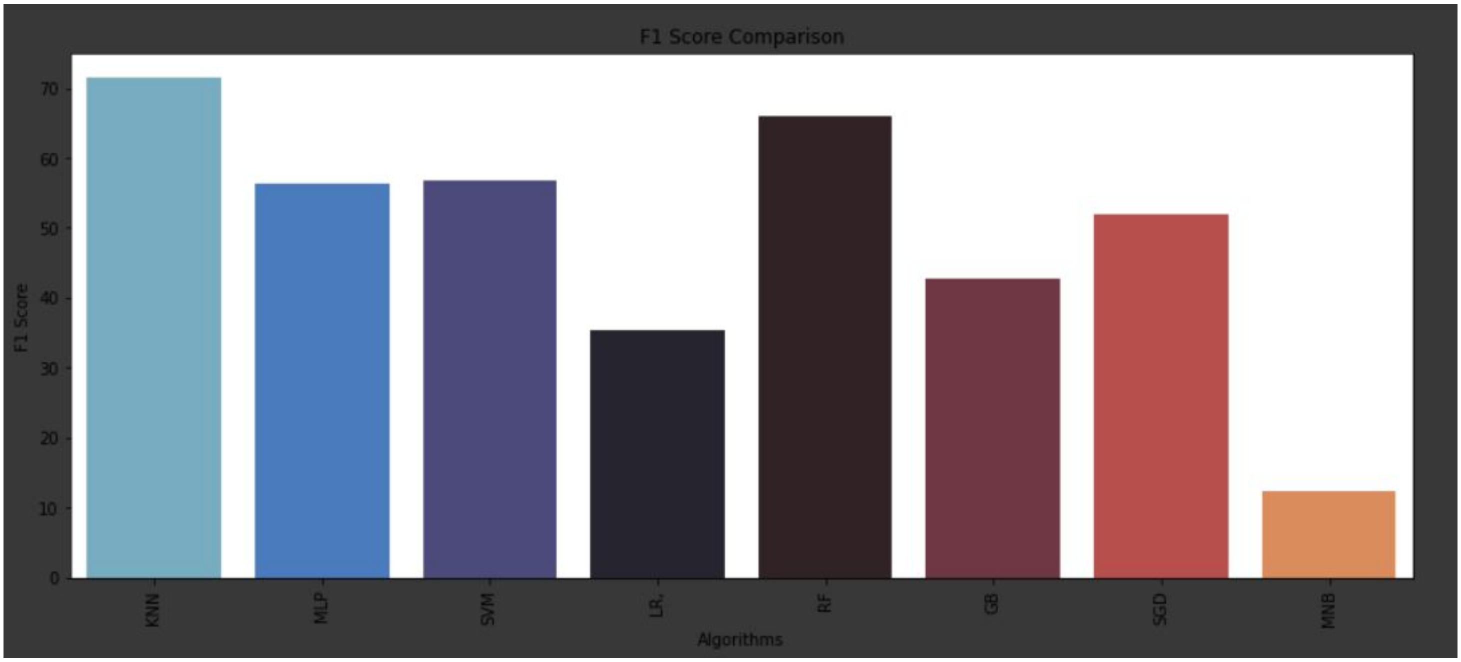
(Fig 10 : Comparison of Accuracy)



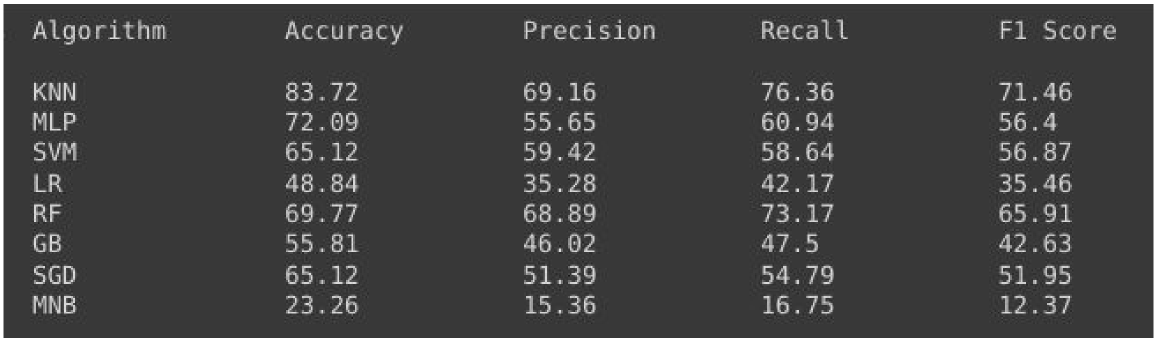
(Fig 11 : Comparison of Precision)



(Fig 12 : Comparison of Recall)



(Fig 13 : Comparison of F1 Score)

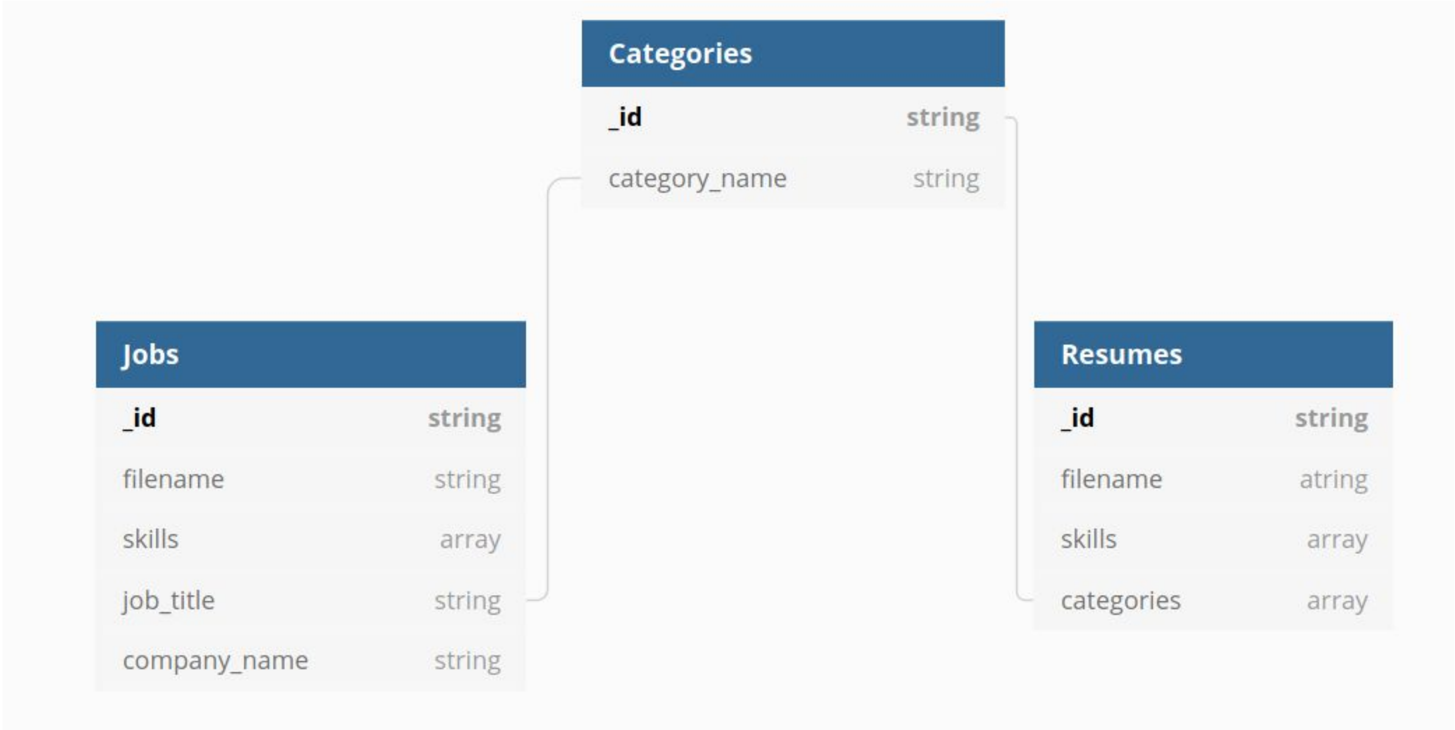


```
Algorithm          Accuracy         Precision        Recall           F1 Score

KNN                83.72            69.16            76.36            71.46
MLP                72.09            55.65            60.94            56.4
SVM                65.12            59.42            58.64            56.87
LR                 48.84            35.28            42.17            35.46
RF                 69.77            68.89            73.17            65.91
GB                 55.81            46.02            47.5             42.63
SGD                65.12            51.39            54.79            51.95
MNB                23.26            15.36            16.75            12.37
```

(Fig 14: Comparison of Accuracy, Precision, Recall and F1 Score)

The below diagram shows the database schema, the collections that were created after extracting job skills and accordingly after classifying the resumes under them. Since the database used is MongoDB, therefore _id acts as the primary key. The category_name in the Categories table acts as the foreign key and the categories attribute in the Resumes table acts as another foreign key.



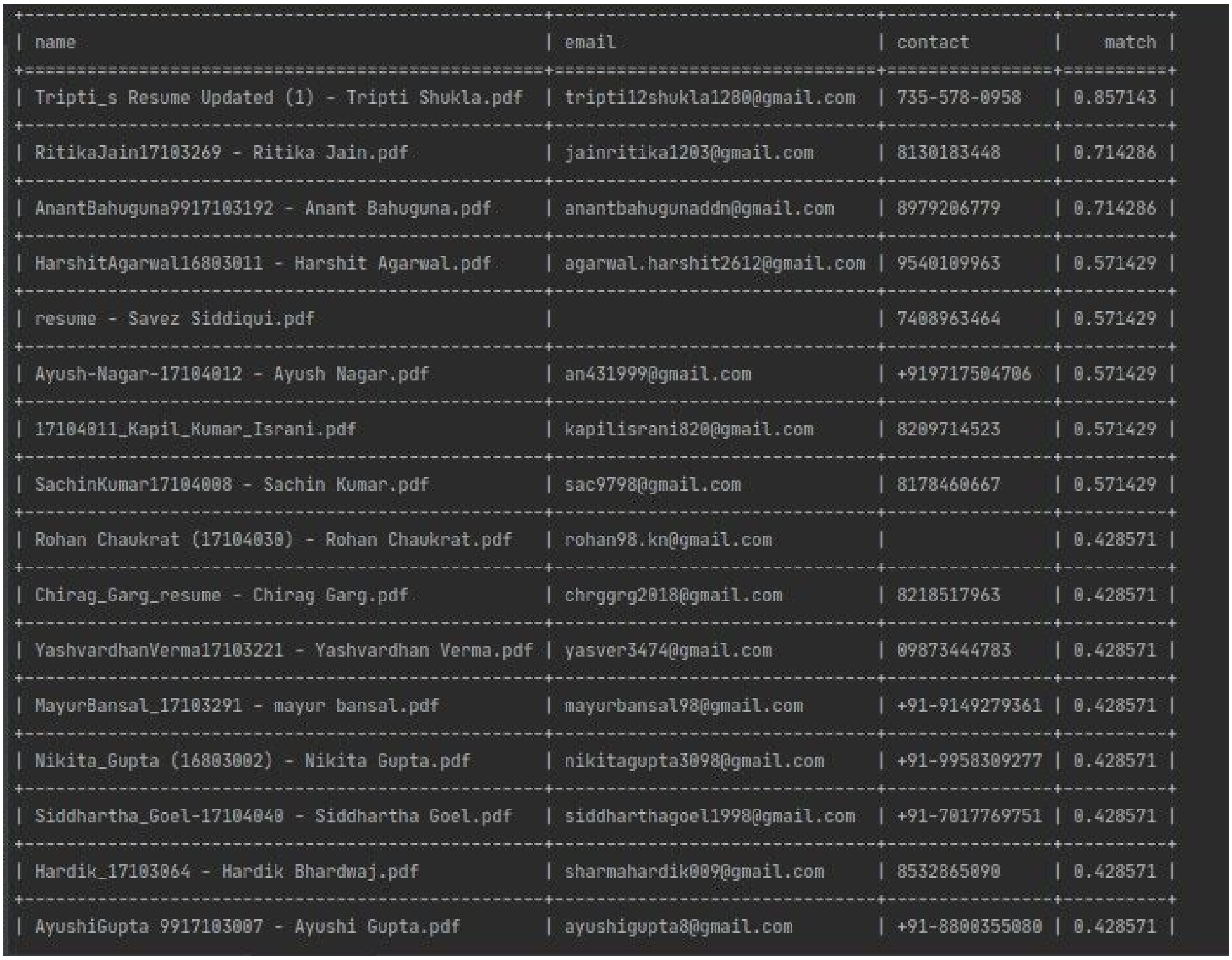
Categories	
_id	string
category_name	string

Jobs	
_id	string
filename	string
skills	array
job_title	string
company_name	string

Resumes	
_id	string
filename	atring
skills	array
categories	array

(Fig 15 : Database schema model)

The below table shows the ranked resumes for a given job description. Here, name indicates the name of the file which was uploaded, email and contact number were extracted from these resumes and match is the score of the similarity between the resume and the job description.



```
+--------------------------------------------------+-------------------------------+----------------+----------+
| name                                             | email                         | contact        |    match |
+==================================================+===============================+================+==========+
| Tripti_s Resume Updated (1) - Tripti Shukla.pdf  | tripti12shukla1280@gmail.com  | 735-578-0958   | 0.857143 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| RitikaJain17103269 - Ritika Jain.pdf             | jainritika1203@gmail.com      | 8130183448     | 0.714286 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| AnantBahuguna9917103192 - Anant Bahuguna.pdf     | anantbahugunaddn@gmail.com    | 8979206779     | 0.714286 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| HarshitAgarwal16803011 - Harshit Agarwal.pdf     | agarwal.harshit2612@gmail.com | 9540109963     | 0.571429 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| resume - Savez Siddiqui.pdf                      |                               | 7408963464     | 0.571429 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| Ayush-Nagar-17104012 - Ayush Nagar.pdf           | an431999@gmail.com            | +919717504706  | 0.571429 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| 17104011_Kapil_Kumar_Israni.pdf                  | kapilisrani820@gmail.com      | 8209714523     | 0.571429 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| SachinKumar17104008 - Sachin Kumar.pdf           | sac9798@gmail.com             | 8178460667     | 0.571429 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| Rohan Chaukrat (17104030) - Rohan Chaukrat.pdf   | rohan98.kn@gmail.com          |                | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| Chirag_Garg_resume - Chirag Garg.pdf             | chrggrg2018@gmail.com         | 8218517963     | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| YashvardhanVerma17103221 - Yashvardhan Verma.pdf | yasver3474@gmail.com          | 09873444783    | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| MayurBansal_17103291 - mayur bansal.pdf          | mayurbansal98@gmail.com       | +91-9149279361 | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| Nikita_Gupta (16803002) - Nikita Gupta.pdf       | nikitagupta3098@gmail.com     | +91-9958309277 | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| Siddhartha_Goel-17104040 - Siddhartha Goel.pdf   | siddharthagoel1998@gmail.com  | +91-7017769751 | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| Hardik_17103064 - Hardik Bhardwaj.pdf            | sharmahardik009@gmail.com     | 8532865090     | 0.428571 |
+--------------------------------------------------+-------------------------------+----------------+----------+
| AyushiGupta 9917103007 - Ayushi Gupta.pdf        | ayushigupta8@gmail.com        | +91-8800355080 | 0.428571 |
```

(Fig 16 : Resume Matching with Scores)

Issues : The issues that we faced during the project were mainly related to parsing of the resume text and extracting the relevant details like skills, name, and email addresses. It was regarding cleaning and reading the variety of files that are being uploaded by the applicant and due to this variety of unstructured data, it was a time consuming task. Also, we noticed that skills we needed to extract were not limited to one field, these resume skills could belong to any field for example HR, Sales, Health and Fitness, Business related, so it was very important to have a corpus pertaining to all these fields. Above all it was very important to store the data so that it could be accessed from any system via API call.

4.3 Risk Analysis and Mitigation : It was noticed that for a resume some categories may be at same distance, but due to the specified value of K given, those categories aren't considered. To mitigate this we maintained a hashmap with key as distance and value as categories. We start from lowest distance categories and if for the same distance there exist many categories, we consider them in top-k categories. It helped in lessening the chances of ignoring categories which were suited for the resume, but there's always a chance that every category found in the neighbours of the test resume are at a same distance, which hence increases the chances of inaccurate classification.

Further, it was found that some skills could be written in many formats like Node.js, Nodejs or Node, all these are the same skills but could get classified differently according to the training dataset. To avoid such a situation, we have fixed a nomenclature that the user would type the skills correctly as defined by the global convention. This prevented resumes from being wrongly classified while extracting skills and cleaning them.

Also, while cleaning the resume for skills we had to keep in mind that we do not remove punctuations that are used while defining the skillset, for example C#, C++, .Net, Node.js, HTML5 etc. So this had to be avoided because by removing the punctuations the skills would belong to a different category, like C# would become C, an entirely different skill set.

5. TESTING

5.1 Testing Plan :

- **Unit testing :** It takes the smallest piece of testable software in the application, isolates it, and determines if it behaves as expected. Each unit is tested separately before being integrated into modules. A large percentage of defects are identified during unit testing. Unit tests are written from a programmer's perspective. They ensure that a particular method of a class successfully performs a set of operations. Unit tests drive the design. In our project unit tests are performed on every function of the modules throughout the development.

- **Integration Testing :** It is a logical extension of unit testing. In integration testing, two individual units already tested are combined into a component and tested. The idea is to test combinations of pieces and eventually expand the process to test all the modules with those of other groups. Eventually all the modules making up a process are tested together. Integration testing is performed in three ways: the top-down, bottom-up, and umbrella approaches. For our project we followed a bottom-up approach, i.e the lowest level units were tested and integrated first.

Type of Test	Will it be performed ?	Explanation	Software Component
Requirement	NA	NA	NA
Unit	Yes	Individual components are being tested	Jd_reader and Resume parser can be tested individually
Integration	Yes	Ranking System check	Resume matcher can be tested for integration
Performance	NA	NA	NA
Stress	NA	NA	NA

Compliance	NA	NA	NA
Security	NA	NA	NA
Load	NA	NA	NA
Volume	NA	NA	NA

5.2 Component Decomposition and type of testing required :

SNo	List of Components	Type of testing required	Technique for writing test case
1	Classification	Unit Testing	Statement Testing & Decision Testing
2	Parsing / Skill Extraction	Unit Testing	Statement Testing, Decision Testing & Path Testing
3	Matching	Unit & Integration Testing	Decision Testing & Branch Testing

5.3 List of all Test Cases :

For Classification Component :

Test Case ID	Input	Expected Output	Status
1	Input test file at path **test-resumes/Ayush_resume-sept-21.pdf**	'category': ['DevOps Engineer', 'Web Designing', 'Database']	PASS

For Parsing/Skill Extraction Component :

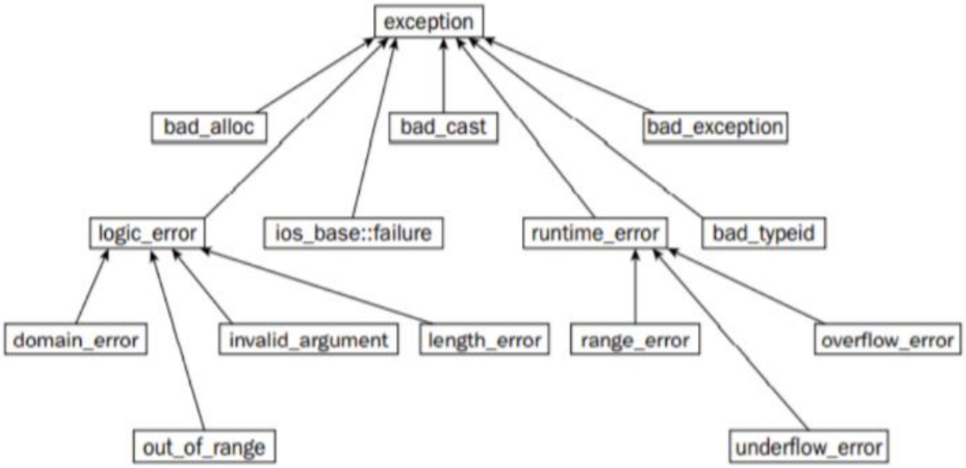
Test Case ID	Input	Expected Output	Status
2	Input test file at path **CV/Ayush_resume-sept-**	'skills': ['pvr', 'google', 'javascript', 'health', 'features',	PASS

	21.pdf	'twitter', 'foundation', 'flask', 'c++', 'pcm', 'reactjs', 'waiting', 'twitter api', 'cron', 'new features', 'speech', 'data science', 'it', 'rest', 'github', 'star', 'badges', 'iit', 'repository', 'api', 'mysql', 'responsiveness', 'angular', 'ml', 'web', 'python', 'nodejs', 'fixing', 'bootstrap', 'community health', 'shopping', 'nlp', 'application', 'refactoring', 'mongodb', 'git', 'history', 'dps', 'app', 'ticketing']	
3	Input test file at path **JD/jd02.pdf**	['python', 'github', 'reactjs', 'nodejs', 'javascript']	PASS

For Matching Component :

Test Case ID	Input	Expected Output	Status
4	Test on **jd01.pdf**	Got 21 sorted resume in the form of table (Refer Fig 16)	PASS
5	Test on **jd02.pdf**		PASS

5.4 Error and Exception Handling : The possible exceptions can be understood from the given diagram. We have handled them using the exception handling techniques (like assertion and try..catch block) and returned proper exceptions referring from the below diagram.



(Fig 17 : Hierarchy Diagram of Exception Handling)

Error Handling Techniques :

- Handling them locally
- Propagating the error using error codes.
- Logging debug information into a file
- Displaying error messages whenever the error is encountered
- Calling an error-processing routine/object
- Substituting the closest legal value
- Returning the same answer as the previous time
- Substituting the next piece of valid data

5.5 Limitations : Since the dataset that would have helped us extract Experience and Education Details was not available, therefore ranking algorithm was merely based on Skill Matching.

6. FINDINGS AND CONCLUSION

6.1 Findings : Through this project we found ways of using and storing unstructured data and how feature extraction works. The huge format of resumes and being able to parse them was indeed a challenging task. It thus helped us delve deeper into feature extraction and natural language processing.

6.2 Conclusion : This project has helped us develop a new perspective for commonly used machine learning algorithms and to extract more than just predictions. The natural language processing toolkit (NLTK) and KNN algorithms, have been helpful in carrying out the results. Also, working with huge unstructured data, with a variety of forms was a challenge, especially to extract correct information from it.

Moreover, Pandas, Numpy, Seaborn, Matplotlib and Sklearn were other Python libraries that were used. Selenium was helpful in scraping the skillset. The accuracy of each of the models was evaluated and the performance was compared to the latest and most successful existing methods which used the same dataset. The results show that the methodology presented in this project has better accuracy and reliability in comparison to other methods. This can effectively assist the employers in selecting the best candidates.

6.3 Future Scope : Though the project has stood up to our expectations yet there is always room for improvement. We hope to add a better ranking approach with the help of education and experience.

REFERENCES

[1] “A System for Screening Candidates for Recruitment”, Amit Singh, Rose Catherine, Karthik Visweswariah, Vijil Chenthamarakshan, Nanda Kambhatla, October, 2010, Toronto, Ontario, Canada.

[2] ”Matching People and Jobs: A Bilateral Recommendation Approach”, Jochen Malinowski, Tobias Keim, 39th Hawaii International Conference on System Sciences - 2006

[3] “Cluster based Ranking Index for Enhancing Recruitment Process using Text Mining and Machine Learning”, Mayuri Verma, International Journal of Computer Applications - 2017.

[4] "Automatic extraction of usable information from unstructured resumes to aid search", Kopparapu, Sunil Kumar, Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on. Vol. 1. IEEE.

[5] “Resume Parser: Semi-structured Chinese Document Analysis”, Zhang Chuang, Wu Ming, Li Chun Guang, Xiao Bo, WRI World Congress on Computer Science and Information Engineering, April 2009.

[6] “Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs”, Celik Duygu, Karakas Askyn, Bal Gulsen, Gultunca Cem, IEEE 37th Annual Workshops on Computer Software and Applications Conference Workshops, July 2013.

[7] "Algorithm AS 136: A k-means clustering algorithm", Hartigan, John A., and Manchek A. Wong. Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108

[8] “Collaborative Filtering based on Subsequence Matching: A New Approach”, Alejandro Bellogín, Pablo Sánchez, Informatics and Computer Science Intelligent Systems Applications, August 2017.

[9] “A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm”, Gerard Deepak, Varun Teja & A. Santhanavijayan, Journal of Discrete Mathematical Sciences and Cryptography, April 2020.

[10] “End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT”, Vedant Bhatia, Prateek Rawat, Ajit Kumar, Rajiv Ratn Shah, IEEE 2009.

[11] “Resume: A Robust Framework for Professional Profile Learning & Evaluation”, Clara Gainon de Forsan de Gabriac, Amina Djelloul, Constance Scherer, Vincent Guigue, Patrick Gallinari, Journal, 2000.

[12] “Resume Information Extraction With A Novel Text Block Segmentation Algorithm”, Shicheng Zu and Xiulai Wang, International Journal on Natural Language Computing (IJNLC) Vol.8, No.5, October 2019.

[13] “Learning to Rank Resumes”, Sangameshwar Patil, Girish K. Palshikar, Rajiv Srivastava, Indrajit Das, IEEE, 2012.

[14] “The Resume Corpus: A Large Dataset for Research in Information Extraction Systems”, 15th International Conference on Computational Intelligence and Security (CIS), 2019.