

Screening and Ranking Resumes Using NLP and Machine Learning Techniques

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology in Information Technology

By

MADESHWARAN B(Reg.No - 39120064)

SARATH L(Reg.No - 39120104)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

MAY - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —All grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Madeshwaran B (39120064)** and **Sarath L (39120104)** who carried out the Project entitled “**Screening and Ranking Resumes using NLP and Machine learning Techniques**” under my supervision from November 2022 to May 2023.

Internal Guide

Dr. SenduruSrinivasulu, M.Tech., Ph.D.,

Head of the Department

Dr. R.SUBHASHNI, M.E., Ph.D.

Submitted for Viva voce Examination held on

11/5/2023

Internal Examiner

External Examiner

DECLARATION

We, Madeshwaran B (Reg.No - 39120064) and Sarath L (Reg.No - 39120104) hereby declare that the Project Report entitled “Screening and Ranking Resumes using NLP and Machine learning Techniques” done by me under the guidance of Dr.SenduruSrinivasalu, M.Tech., Ph.D. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE: 11/5/23

PLACE: Chennai

Madeshwaran B,
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T. Sasikala M.E., Ph.D.**, Dean School of computing. **Dr. R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. Senduru Srinivasulu M.tech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish express our thanks to all teaching and Non teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project

ABSTRACT

The evolving technology is creating many chances of employment for many. Nowadays to apply for any job the most essential document is a resume tells a lot about the person's achievements and the skills in all walk of life. The person applying for the job highlights the strong points and skill sets required for the company. Multinational organizations receive thousands of emails from such people who send their resumes for them to apply for a certain post. Now the real challenge is to know which resumes is to be stored and shortlisted according to the constraints. One method is to manually check and sort the resume. Now, this method is the most time-consuming and also can lead to a lot of errors because of human interventions. Also, humans cannot keep on working continuously. Hence there is a problem of less efficiency as well. Thus, we have proposed a system that will easily find the required skill set by scanning the document or the to the skill sets which is a specified constraint of the organization. We are going to use the concept of Machine learning. Machine learning for recruiting is an emerging category of HR technology designed to reduce or even remove time-consuming activities like manually screening resumes.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	V
	LIST OF CONTENTS	Vi
	LIST OF FIGURES	Vii
1.	INTRODUCTION	9
1.1	INTRODUCTION	9
1.2	DESCRIPTION	10
1.3	OBJECTIVE	12
2.	LITERATURE SURVEY	13
3.	SYSTEM ANALYSIS	18
3.1	EXISTING SYSTEM	18
3.2	PROPOSED SYSTEM	18
3.3	REQUIREMENT SPECIFICATION	20
3.4	LANGUAGE SPECIFICATION	22
4.	SYSTEM DESIGN	25
	4.1 SYSTEM ARCHITECTURE	26
	4.2 USE-CASE DIAGRAM	27
	4.3 ACTIVITY DIAGRAM	28
	4.4 SEQUENCE DIAGRAM	29
	4.5 DATA FLOW DIAGRAM 0	29
	4.6 DATA FLOW DIAGRAM 1	30
	4.7 ER DIAGRAM	30

5.	MODULE DESCRIPTION	
	5.1 NATURAL LANGUAGE PROCESSING(NLP)	31
	5. 2 TF-IDF	31
	5.3 LATENT DIRICHLET ALLOCATION(LDA)	32
	5.4 METHODOLOGY	32
6.	TESTING	34
	6.1 SYSTEM TESTING	34
	6.2.TESTING TECHNIQUES	37
	6.3 SOFTWARE TESTING STRATEGIES	38
7.	CONCLUSION	42
8.	APPENDIX 1	44
9.	REFERENCES	46
10.	RESEARCH PAPER	48

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	SYSTEM ARCHITECTURE	25
4.2	USE-CASE DIAGRAM	26
4.3	ACTIVITY DIAGRAM	27
4.4	SEQUENCE DIAGRAM	28
4.5	DATA FLOW DIAGRAM 0	29
4.6	DATA FLOW DIAGRAM 1	29
4.7	ER DIAGRAM	30
9.1	SHORTLISTED RESUMES	44
9.2	SCORE AND RANK DISTRIBUTION	44
9.3	TOPICS AND TOPIC RELATED KEYWORDS	45
9.4	JOB NO AND JOB DESCRIPTION NAME	45

CHAPTER 1

INTRODUCTION

1.1 Introduction

Day by day there are so many new researches that are being carried out by so many organizations in many fields. Right from the IT sector to the Medical fields, there are so many researches as well as progress made in day to day life. These progress in these fields are creating new employment opportunities all over the world. Now we know that the hiring process across the world is the same. That is the candidate has to make the resume first and as most of the recruitment process is based on the candidates' resume. Now most of the organizations tell the candidates to send the resume via e-mails. Now after they receive the email the next job is to sort them according to the requirement. Now usually they sort these manually but this process is time consuming. Also there may be so many errors due to that. The efficiency of such sorting is extremely low. This may result in giving a chance to a candidate whose credentials are not up to the mark for the organization or this may also make the manual sorting miss the candidate who is extremely good for the organization. Hence a system which is intelligent enough to sort all these documents without any error and within time is needed which is exactly our aim. Machine learning for recruiting is the application of machine learning, such as the learning or problem-solving that a computer can do, to the recruitment function. This new technology is designed to streamline or automate some part of the recruiting workflow, especially repetitive, high-volume tasks. For example, software that applies machine learning to resumes to auto-screen candidates or software that conducts sentiment analysis on job descriptions to

identify potentially biased language.

1.2 Description

Recruitment is a 200-billion-dollar business. It deals with hiring the best-fit candidates having the relevant skills for a given job profile from an immensely large pool of candidates. If a company has any job opening for a position, scores of candidates mail their resumes to the company to apply for that opening. In the hiring process, the first task for any recruiter is to screen the resumes of all the job applicants. Any company having a job opening for a particular position will have their mail inboxes bombarded with thousands of emails from aspiring job applicants every single day. Selecting the prospective candidates for that job position from a large pool of candidates for any recruiter is very tedious. It is an extremely daunting task for the recruiters of a company to manually go through thousands of resumes and select the most appropriate candidates for the job. Out of those thousands of resumes submitted to the company for the given job posting, about 75% of them do not showcase the relevant skills that are required for the job profile.

Due to this, the recruiters quite often find it really arduous to narrow down the most appropriate candidates from a large applicant pool. In recent years, there have been more than 50,000 e-recruitment sites have been developed. The developers of these online recruitment sites have used various approaches to identify the prospective candidates for a given job profile of a company. Some of these, have managed to employ classification techniques that will classify the candidate resumes into various categories for every job posting given by every company. In these approaches, every candidate's resume is tried to match with every given job posting on the recruitment site. The aim of these recruitment sites is to throw up the results to the candidate to which they are

the best fit. The techniques used by these sites have resulted in high accuracy and precision, but one of the major disadvantages is the factor of time complexity. If every candidate's resume is matched with every other job posting given on the online recruitment site, the time complexity for acquiring the results is very high.

The world of Artificial Intelligence [AI] and Machine Learning [ML] has grown significantly in recent times. The availability of large amounts of data brought about by advancements in technology which has made the internet cheap and accessible to previously inaccessible regions of the world has contributed to a great increase in the performance of the ML models in recent times. Software companies around the world exploit the advances in ML to drive automation and increase their productivity in areas that relied mostly on manual human labor. The approach discussed in this project is by using machine learning to train the dataset for a particular type of job position. It is also proposed to use section-based segmentation for data extraction using Natural language Processing (NLP). In order to improve the time efficiency of the web application, the candidate's resume will only be matched to those job openings where they are interested in and have applied to which will, in turn, reduce the time complexity. Besides, the results of the resume matching of all the candidates who have applied for the job opening will be visible only to the recruiter of that particular company. This is done with the aim to aid the recruiters of any company from the long and tedious task of viewing and analyzing thousands of candidates' resumes. In this intelligent-based approach, they will be given the option to view the candidate's resume as well as they will get the results of the best candidates suitable for the required job position. same disease, we use it as the score to rank the similarity among patients.

1.3 Objective

- The major objective of our system is to take the current resume ranking system to another level and makes it more flexible for both the entity.
 1. Candidates, who have been hired.
 2. Client company, who is hiring the candidates
- To propose an algorithm that provides a list of applicants with appropriate experience and then presents the high points of each selected resume, unlike the conventional way of applying filters and manually scanning resumes.
- To identify the most qualified candidates for a certain vacancy.
- To automate tasks that the human resources can do and recognize and identify human faces
- It proposes a resume ranking software that uses natural language processing (NLP) and machine learning.
- It could be used in MNCs where multiple resumes must be screened every single day for multiple jobs, government, and administrative offices.
- If an organization gets the right candidates in place in the company, human resources can focus their efforts on building a driven company culture, and onboarding the right human capital for the positions that come up.

CHAPTER 2

LITERATURE REVIEWS

TITLE 1: Resume Classification And Ranking Using KNN And Cosine Similarity

AUTHORS: Rajath V; Riza TanazFareed; SharadadeviKaganurmamath

DESCRIPTION:

The authors in [1] created an automated machine learning-based algorithm that recommends acceptable applicant resumes to HR based on the job description provided. The suggested methodology had two stages: first, it classified resumes into various groups. Second, it suggests resumes based on their resemblance to the job description. If an industry produces a high number of resumes, the proposed approach can be used to create an industry-specific model.

TITLE 2: Natural Language Processing based Jaro-The Interviewing Chatbot

AUTHORS: Jitendra Purohit; Aditya Bagwe; Rishabh Mehta;

Ojaswini Mangaonkar; Elizabeth George

DESCRIPTION:

The proposed system, JARO accelerates the interview process towards an unbiased decision-making process by proposing a chatbot that would conduct interviews by analyzing the candidate's Curriculum Vitae (CV), based on which, it then prepares a set of questions to be asked to the candidate. The system will consist of features like resume analysis and automatic interview processes.

TITLE 3:Design and Development of Machine Learning based Resume Ranking System

AUTHORS:Tejaswini K; Umadevi V; Shashank M Kadiwal; Sanjay Revann

DESCRIPTION:the top applicants might be rated using content-based suggestion, which uses cosine similarity to find the curriculum vitae that are the most comparable to the job description supplied and KNN algorithm is used to pick and rank Curriculum Vitae (CV) based on job descriptions in huge quantities.

TITLE 4: A Machine Learning approach for automation of Resume Recommendation system

AUTHORS: Pradeep Kumar Roy; Sarabjeet Singh Chowdhary; Rocky Bhatia

DESCRIPTION: suggested that an automated way of "Resume Classification and Matching" could really ease the tedious process of fair screening and shortlisting, it would certainly expedite the candidate selection and decisionmaking process. This system could work with a large number of resumes for first classifying the right categories using different classifier, once classification has been done then as per the job description, top candidates could be ranked using Content-based Recommendation, using cosine similarity and by using k-NN to identify the CVs that are nearest to the provided job description.

TITLE 5:A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Post

AUTHORS: AbeerZaroor; Mohammed Maree; Muath Sabha

DESCRIPTION: The research work of authors in [7] presented a hybrid approach that employs conceptual-based classification of resumes and job

postings and automatically ranks candidate resumes (that fall under each category) to their corresponding job offers. In this context, the exploit an integrated knowledge base for carrying out the classification task and experimentally demonstrate - using a real-world recruitment dataset- achieving promising precision results compared to conventional machine learning-based resume classification approaches.

TITLE 6: An Empirical Study of Artificial Intelligence and its Impact on Human Resource Functions

AUTHORS: Garima Bhardwaj; S. Vikram Singh; Vinay Kumar

DESCRIPTION: The study work in [8] was conducted among 115 HR professionals at various IT sectors in Delhi/NCR region. A multiple regression method was used to test the hypothesis and confirmed a positive relationship between these two factors establishing about the increased use of AI at work results in better HR functional performance. However, AI has a significant relationship with innovativeness and also with the ease of use which reflects AI affects HR with innovations and ease of use.

TITLE 7: Intelligent Recruitment System Using NLP

AUTHORS: Anushka Sharma; Smiti Singhal; Dhara Ajudia

DESCRIPTION: The research work in [9] focuses on extracting data from resumes and performing the required analysis on the data to convert it into useful information for the recruiters. Thus, the Resume Parser would help the recruiters to select the best relevant candidates in a minimal amount of time, consequently saving their time and effort.

TITLE 8: ResumeNet: A Learning-Based Framework for Automatic Resume

Quality Assessment

AUTHORS:Yong Luo; Huaizheng Zhang; Yongjie Wang; Yonggang Wen; Xinwen Zhang

DESCRIPTION: The authors in [10] developed a method for automatic RQA. Since there is also no public dataset for model training and evaluation, we build a dataset for RQA by collecting around 10K resumes, which are provided by a private resume management company. By investigating the dataset, we identify some factors or features that could be useful to discriminate good resumes from bad ones, e.g., the consistency between different parts of a resume.

TITLE 9:Ethical Considerations in AI-Based Recruitment.

AUTHORS:Mujtaba, Dena F., and Nihar R. Mahapatra

DESCRIPTION: presents an overview of fairness definitions, methods, and tools as they relate to recruitment and establishes ethical considerations in the use of machine learning in the hiring space. Considering Deep Learning (DL) method recognize artificial Neural Network (NN) to nonlinear process, NLP tools become increasingly accurate and efficient that begin a debacle. Multi-Layer Neural Network obtaining the importance of the NLP for its capability including standard speed and resolute output. Hierarchical designs of data operate recurring processing layers to learn and with this arrangement of DL methods manage several practices

TITLE 10:Inflectional review of deep learning on natural language processing.

AUTHORS: Fahad, SK Ahmed, and Abdul Samad Ebrahim Yahya

DESCRIPTION: this resumed striving to reach a review of the tools and the necessary methodology to present a clear understanding of the association of

NLP and DL for truly understanding in the training. Efficiency and execution both are improved in NLP by Part of speech tagging (POST), Morphological Analysis, Named Entity Recognition (NER), Semantic Role Labeling (SRL), Syntactic Parsing, and Coreference resolution. Artificial Neural Networks (ANN), Recurrent Neural Network (RNN), Convolution Neural Networks (CNN), dealings among Dense Vector (DV), Windows Approach (WA), and Multitask learning (MTL) as a characteristic of Deep Learning.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system is a traditional Machine Learning based system. It gives lower rates of accuracy. It has lower efficiency. It gives lower inaccurate results. This system may lead to loss of human potential.

3.2 PROPOSED SYSTEM

Our system is a resume ranking software that uses natural language processing (NLP) and machine learning. This AI-powered resume screening program goes beyond keywords to contextually screen resumes. Following resume screening, the software rates prospects in real-time depending on the recruiter's job needs. The web application aims to order the resumes, by intelligently reading job descriptions as input and comparing the resumes which fall into the category of given Job Descriptions. It provides a ranking after filtering and recommends the better resume for a given textual job description. In order to match and rate candidates in real-time, the software employs natural language processing. Unlike generic processes, this app utilizes Mong for string matching, Cosine Similarity, Overlapping coefficient Natural Language. Our work takes a different approach as it focuses mainly on the content of the resumes where we perform the extraction of skills and related parameters to match candidates with the job descriptions. The interactive web application will allow the job applicants to submit their resumes and apply for 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. This is done with the aim to aid the recruiters of any company from the long and tedious task of viewing and analyzing thousands of candidate resumes. The calculated ranking scores can then be utilized to determine best-fitting candidates for that particular job opening. Since the dynamic model leverages NLP, it gives the output instantly. While going through all these pipelines, it will score each resume and give out accurate output with higher efficiency, precision, and accuracy. It works the following way -

Resumes from the Data set are parsed to remove white spaces, numbers, stop words like and, or, etc. TF-IDF vectorization is then applied to convert the words in the resumes to vectors. The text in the job description is also converted to vectors using the TF-IDF vectorizer. Cosine distance is computed to measure the similarity between the resume and the job description provided and Then Mong String algorithm is applied to identify the resumes which are closely matching with the JD provided by the recruiters. The main contributions of our work can be summarized as follows: Use of our framework is not limited to a single field of application and is useful for many more real-world applications. Providing a ranking-based approach after filtering Framework to highlight skills of a resume.

3.3 REQUIREMENT SPECIFICATION

3.3.1 HARDWARE REQUIREMENTS

Processor	: Pentium Dual Core 2.00GHZ
Hard disk	: 120 GB
RAM	: 2GB (minimum)
Keyboard	: 110 keys enhanced

3.3.2 SOFTWARE REQUIREMENTS

Operating system	: Windows7 (with service pack 1), 8, 8.1 and 10
IDE	:Anaconda
Language	: Python

3.4 PLATFORM SPECIFICATION– ANACONDA

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector,

Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

```
conda env list
```

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

```
conda create -n <your_environment_name>
```

You can replace `<your_environment_name>` with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

```
proceed ([y])/n)?
```

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

```
conda create -n <your_environment_name> python=3.6
```

Similarly, if you want to create an environment with a particular package, you can use the following command:

```
conda create -n <your_environment_name> pack_name
```

Here, you can replace `pack_name` with the name of the package you want to use.

If you have a `.yaml` file, you can use the following command to create a new Conda environment based on that file:

```
conda env create -n <your_environment_name> -f <file_name>.yaml
```

We have also discussed how you can export an existing Conda environment to a `.yaml` file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command:

```
conda activate <environment_name>
```

You should activate the environment before you start working on the same.

Also, replace the term <environment_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

```
conda deactivate
```

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

```
conda install <pack_name>
```

Replace the term <pack_name> with the name of the package you want to install in your Conda environment while using this command.

Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

```
conda update
```

The above command will update all the packages present in the environment.

However, if you want to update a package to a certain version, you will need to use the following command:

```
conda install <package_name>=<version>
```


Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you'll first have to activate the same and run the following command:

```
conda env export ><file_name>.yml
```

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

```
conda remove -n <env_name><package_name>
```

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

```
conda remove <package_name>
```

Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

```
conda env remove --name <env_name>
```

The above command would delete the Conda environment right away.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This diagram is nothing but a simple description of all the entities that have been incorporated into the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. You can simply call it a visual or the whole process and its implementation. All functional correspondences are explained in this diagram.

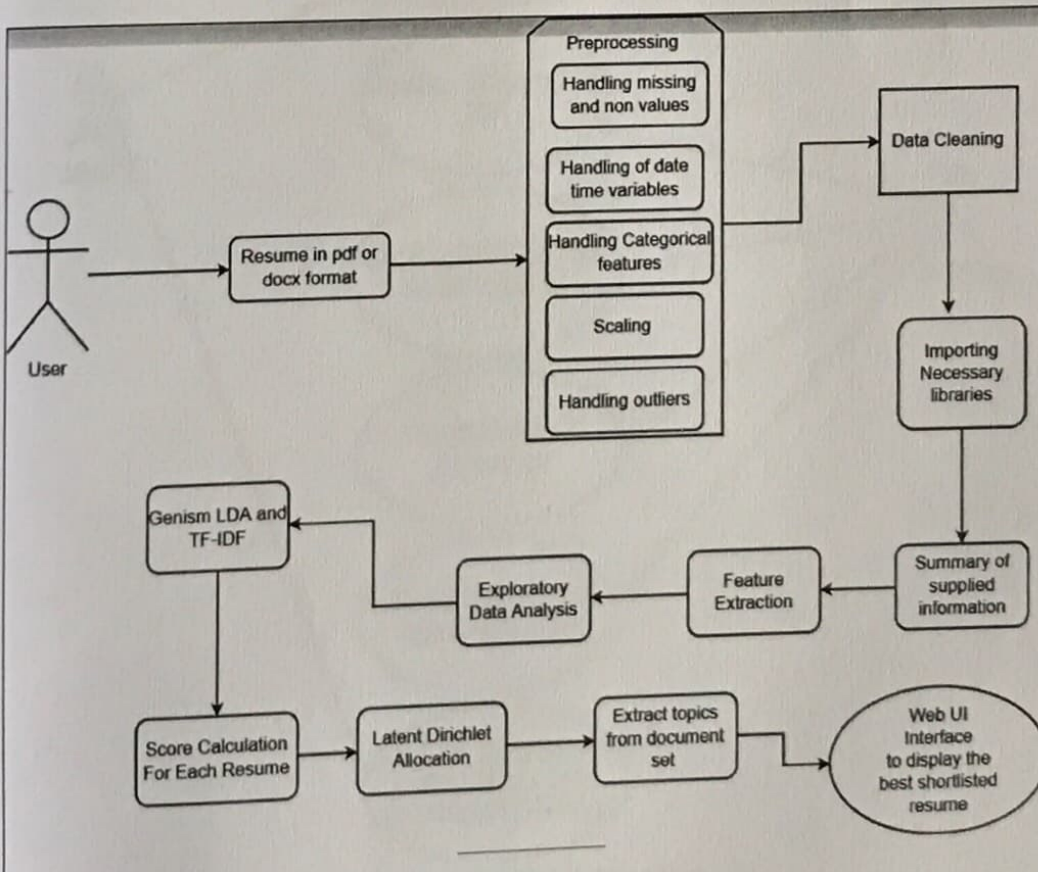


Fig 4.1 – Architecture Diagram

4.2 USE-CASE DIAGRAM

A use case diagram is a standard diagram that shows all interactions between the user, dataset, and algorithm used. It is developed in the early stages of the process.

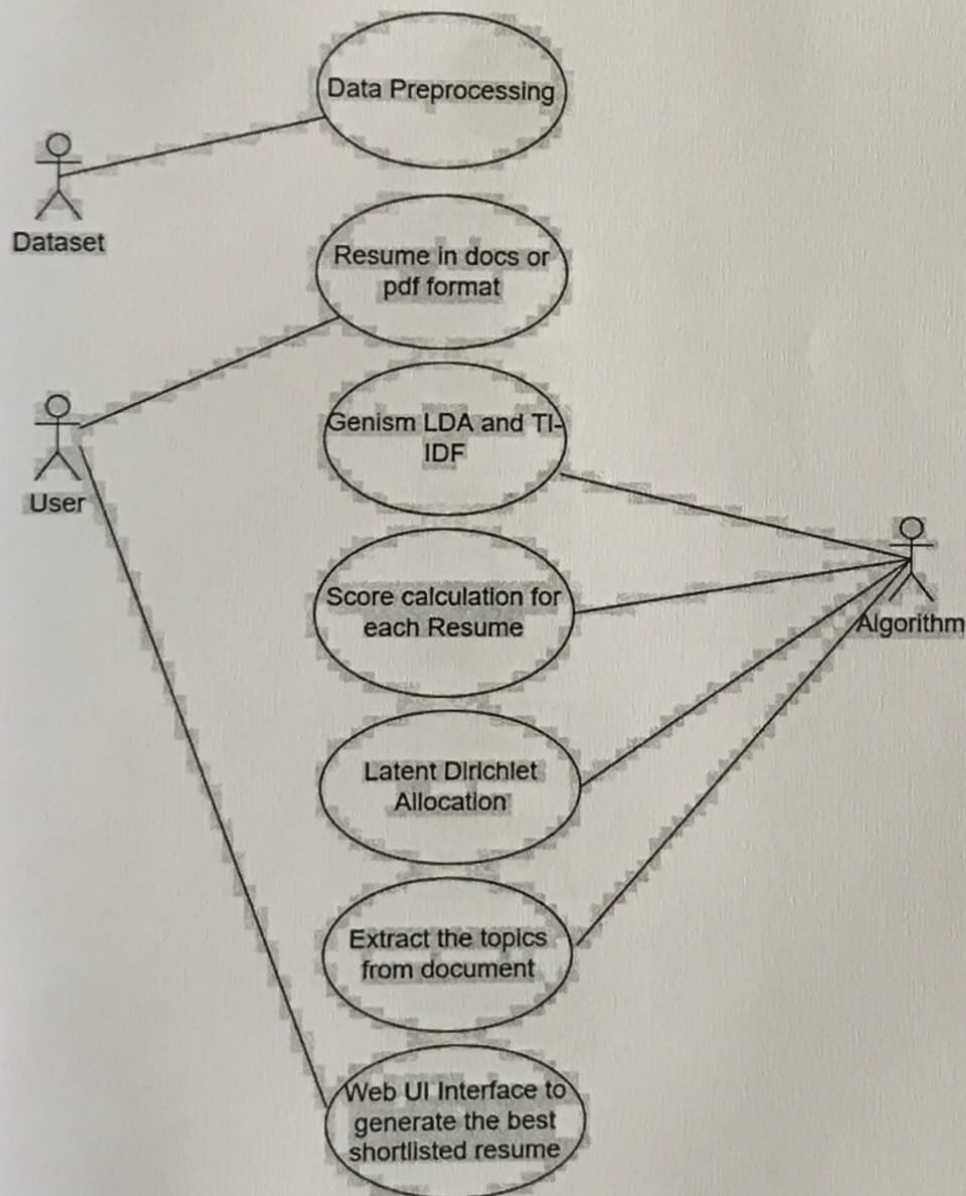


Fig 4.2 – Use-Case Diagram

4.3 ACTIVITY DIAGRAM

In simple terms, a diagram that represents the order of all activities is called the activity diagram. It shows the workflow between different activities that take place in the whole process. However, these are not exactly flowcharts but are similar.

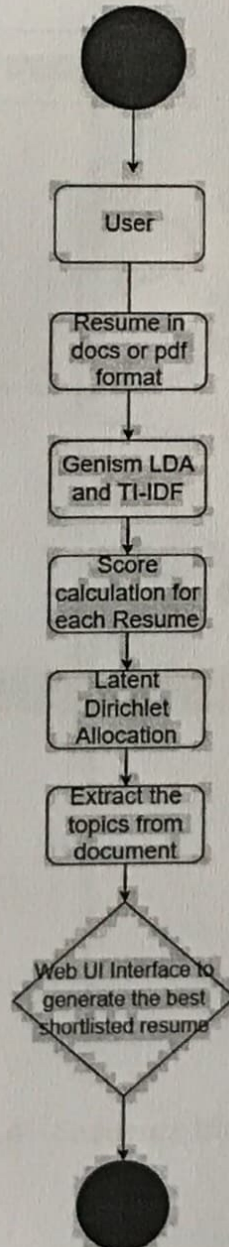


Fig 4.3 – Activity Diagram

4.4 SEQUENCE DIAGRAM

These are other kinds of interaction-based diagrams that show how all the operations are carried out. They capture the context of collaborations between objects and processes.

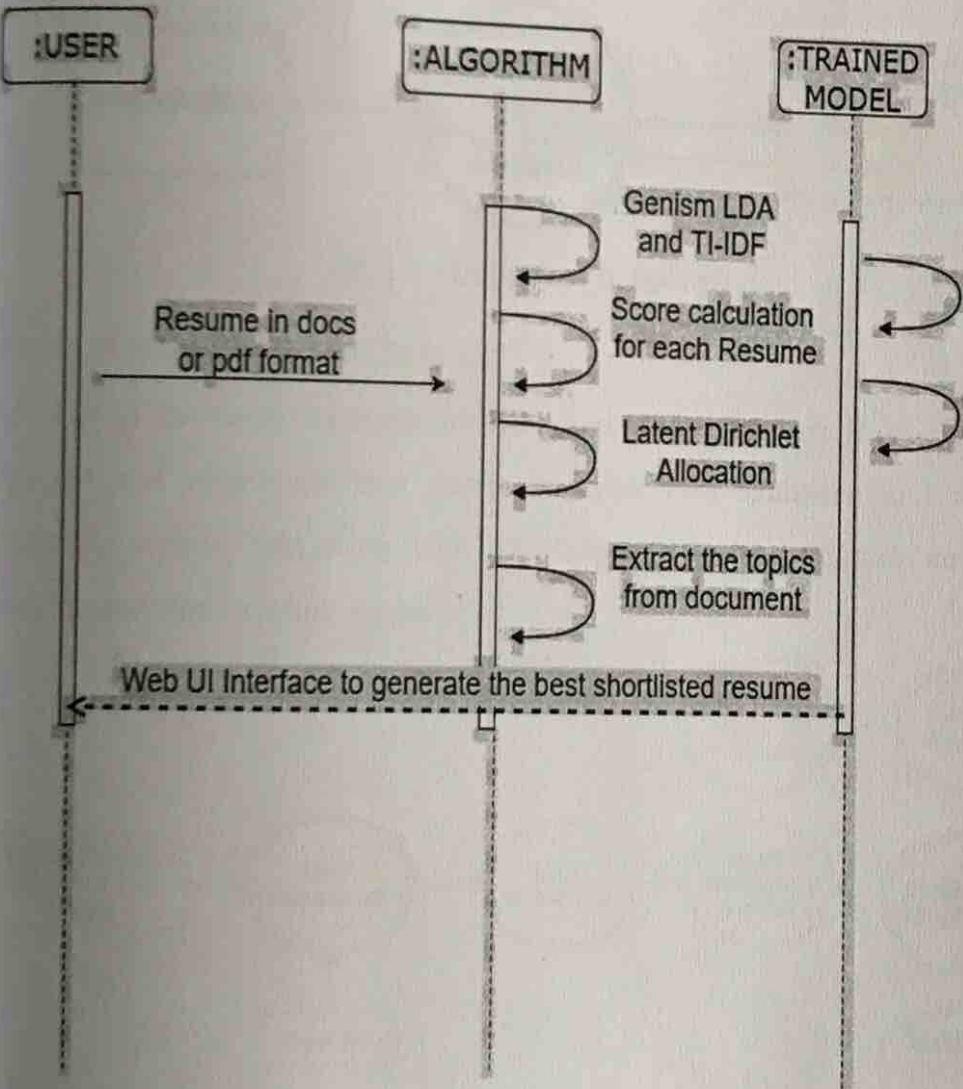


Fig 4.4 – Sequence Diagram

4.5 DATA FLOW DIAGRAM LEVEL 0

This is basically a contextual diagram, also referred to as a “context diagram”. It only represents the top level or the 0 Level in the whole process. it gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities.

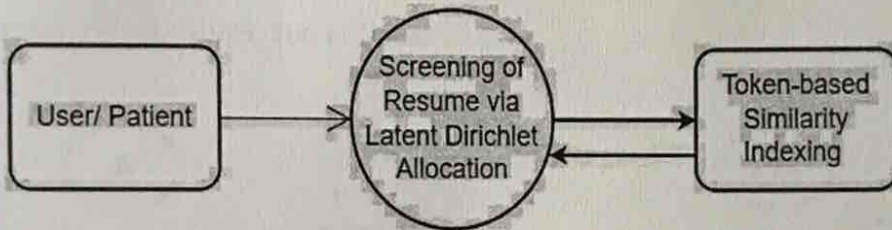


Fig 4.5 – Data Flow Diagram 0

4.6 DATA FLOW DIAGRAM LEVEL 1

Level DFDs represent the complete system as a single process. it rotates every process and subprocess that comes together in a sequence to form the complete system. This along with) and 2-level data flow diagrams comprise the “fundamental system model”.

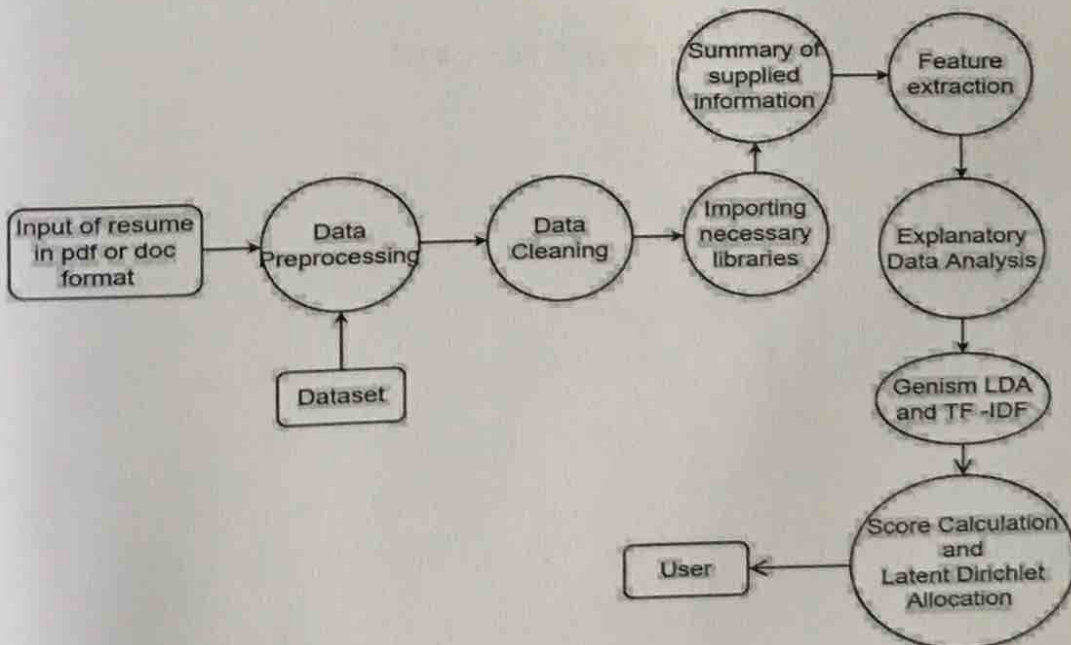


Fig 4.6 – Data Flow Diagram 1

4.7 ER DIAGRAM

ER stands for Entity Relationship. These diagrams display the relationship of entities that are used and stored in the database. They explain the structure of the whole process. These diagrams can be made using three basic concepts, attributes, relationships, and entities.

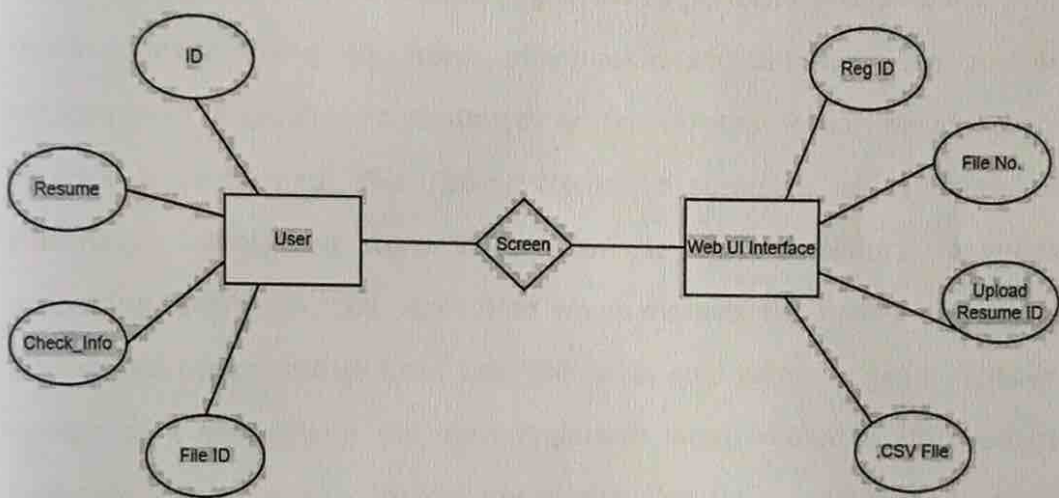


Fig 4.7 – ER Diagram

CHAPTER 5

MODULE DESCRIPTION

5.1 NATURAL LANGUAGE PROCESSING (NLP)

Resume collection is being performed and folder Structure creation is being done. Data Cleaning such as removing clutter and unnecessary punctuation would be taken off, Feature Engineering would be performed for enhancement. This includes removing stop words, punctuation, and stemming. This process will construct a graph with sentences as the vertices. Importing necessary libraries is performed. This library creates a summary of the supplied information within the word limit. With the help of Natural language processing techniques, the application which extracts the names of people, places, and other entities from text, the main goal being to get a reduced version of it that retains the most important information. It is followed by Exploratory Data Analysis. Exploratory data analysis (EDA) is often a necessary task in uncovering hidden patterns, detecting outliers, and identifying important variables and any anomalies in data.

5.2 TF-IDF

At this stage, a dynamic Script for the Tf-Idf approach is written. Term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection. TF-IDF is word frequency scores that aim to emphasize phrases that are more interesting, e.g., common in a text but not across texts, without delving into the arithmetic. The TF-IDF Vectorizer tokenizes texts, learns vocabulary, inverts frequency weightings, and allows encoding new ones. It provides information on a word

frequency in the documents. Higher the TF- IDF score of a term which is computed using the above equations represents more relevance in a document.

5.3 LATENT DIRICHLET ALLOCATION (LDA)

Latent Dirichlet Allocation A tool and technique for Topic Modeling, Latent Dirichlet Allocation (LDA) classifies or categorizes the text into a document and the words per topic, these are modeled based on the Dirichlet distributions and processes. Latent Dirichlet Allocation has been used in the application for the following functions- Discovering the hidden themes in the data. Classifying the data into the discovered themes. Using the classification to organize/summarize/search the documents. The application, then deals with the calculation of the score for a candidate's resume according to the job posting they have applied for. According to the score each candidate's resume receives, a rank list will be made with the candidate receiving a higher score placed higher as compared to the candidate receiving a lower score.

5.4 METHODOLOGY

Data Collection: Collect a dataset of resumes and job requirements for different job positions.

Data Preprocessing: Preprocess the dataset by cleaning the text data, removing stop words, stemming, and lemmatization.

Keyword Extraction: Use NLP techniques such as bag-of-words and TF-IDF to extract relevant keywords from the job requirements and resumes.

Classification: Use ML classification algorithms such as SVM, Naive Bayes, and

Decision Trees to classify resumes into different categories based on job requirements. Train the classification model using a set of labeled data.

Ranking: Use ML ranking algorithms such as PageRank and HITS to rank the resumes based on their relevance to the job requirements. Train the ranking model using a set of labeled data.

Clustering: Use ML clustering algorithms such as k-means clustering and hierarchical clustering to group similar resumes together.

Evaluation: Evaluate the performance of the system using metrics such as accuracy, precision, and recall. Fine-tune the system by adjusting the parameters of the algorithms and testing the performance.

CHAPTER 6

TESTING

Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

6.1 SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise, the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as a walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

Unit Testing

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were

heavily employed for unit testing.

Functional Tests

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

Performance Test

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

Stress Test

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

Structure Test

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.

- Checking attributes for their correctness.
- Handling end of file conditions, I/O errors, buffer problems, and textual errors in the output information

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules that makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach evolved from the unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that interface dispenses can be easily found and corrected.

The major error that was faced during the project is a linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

6.2 TESTING TECHNIQUES / TESTING STRATEGIES

Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet-undiscovered error. A successful test is one that uncovers an as-yet-undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hangs together. System testing requires a test consisting of several key activities and steps for running a program, string, and system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise, the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that has a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

White-box testing

This testing is also called Glass box testing. In this testing, by knowing the specific functions that a product has been designed to perform tests can be conducted that demonstrate each function is fully operational and at the same time search for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test

cases. Basis path testing is white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

Black box testing

In this testing by knowing the internal operation of a product, a test can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black-box test case design are:

- Graph-based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

6.3 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason, a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works "outward" toward the integration of the entire computer-based system.
- Different testing techniques are appropriate at different points in time.

- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with it. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem, of course, is "putting them together"- interfacing. There may be the chances of data loss across another's sub-functions when combined may not produce the desired major function; individually acceptable impressions may be magnified to unacceptable levels; global data structures can present problems.

Program Testing

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax errors. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items, and invalid combinations. Since the compiler s will not deduct logical errors, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or an arithmetic expression. The condition testing method focuses on testing each condition in the program the purpose the of condition test is to deduct not only errors in the condition of a program

but also other errors in the program.

Security Testing

Security testing attempts to verify the protection mechanisms built into a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attacks must also be tested for invulnerability from rear attacks. During security, the tester places the role of the individual who desires to penetrate the system.

Validation Testing

At the culmination of integration testing, the software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that is reasonably expected by the customer. Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. After the validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics conform to specifications and are accepted.
- * A validation from the specification is uncovered and a deficiency is created.

Deviation or errors discovered at this step in this project are corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

User Acceptance Testing

User acceptance of the system is a key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective systems and users at the time of developing and making changes whenever required. This is done in regard to the following points.

- Input screen design.
- Output screen design.

CHAPTER 7

CONCLUSION

Our algorithm was successfully able to screen and shortlist the best candidates with the help of NLP. Highly accurate results were obtained by using Latent Dirichlet Allocation to display the best-shortlisted resume on Web Ui. The web application was successfully able to order the resumes, by intelligently reading job descriptions as input and comparing the resumes which fall into the category of given Job Descriptions. The organization receives a large number of applications for each employment opening. Finding the right candidate's application from a sea of resumes is a time-consuming endeavor for any company these days. The classification of a candidate's resume is a laborious, time-consuming, and resource-intensive process. To address this problem, we created an automated machine learning-based algorithm that recommends acceptable applicant resumes to HR based on the job description provided. The suggested methodology had two stages: first, it classified resumes into various groups. Second, it suggests resumes based on their resemblance to the job description. If an industry produces a high number of resumes, the proposed approach can be used to create an Industry-specific model. By engaging domain experts such as HR professionals, a more accurate model may be built, and HR professionals' feedback can be used to iteratively enhance the model. The results from the model are encouraging. The resume classifier application is successful in automating the manual task of project allocation to the new recruits of the organization based on the interests, work experience, and expertise mentioned by the candidate in the profile. The Resume Screening System replaces ineffective manual screening, ensuring that no candidate is overlooked. The need for efficient and effective resume screening is at the

heart of every excellent recruitment strategy. The system will be able to accept or reject a job applicant based on two factors: the company's requirements must match the skills listed in the applicant's resume, and the test evaluation will be based on the applicant's skills, ensuring that the resumes uploaded by the applicant are genuine and the applicant is truly knowledgeable about the skills. Using NLP(Natural Language Processing) and ML(Machine Learning) to rank the resumes according to the given constraint, this intelligent system ranks the resume of any format according to the given constraints or the following requirement provided by the client company. We will basically take the bulk of input resume from the client company and that client company will also provide the requirement and the constraints according to which the resume should be ranked by our system. Besides the information provided by the resume, we are going to read the candidate's social profiles (like LinkedIn, GitHub, etc.) which will give us more genuine information about that candidate. The application automates the task of project allocation, thereby eliminating the tedious and redundant affair of opening and analyzing the resumes manually by the HR team of the organization.

FUTURE ENHANCEMENTS

The application can be extended further to other domains like Telecom, Healthcare, E-commerce, and public sector jobs. We also wish to put into effect and present a smart evaluation in the consistent database to survey with the present models. Efforts can be made to explore whether it is possible to identify in advance what lists might be ranked worse than the current baseline and to investigate whether there is another way to transform word embeddings into document embeddings, such as using doc2vec, that could have a greater impact on LTR results. Another area we would want to focus on

is ranking the resumes of people in different languages. Be it Dutch, German, French, or Hindi, we want to explore further ways to analyze and operate our model on major languages of the world.

APPENDIX 1

Top Ranked Resumes

Rank	Name	Scores
1	Vivek Kumar.docx	92.33500549981277
2	Shambhai Mishra.docx	86.34327804453588

Fig 9.1 Shortlisted resumes

The above image depicts the top scores on the respective person's resume.

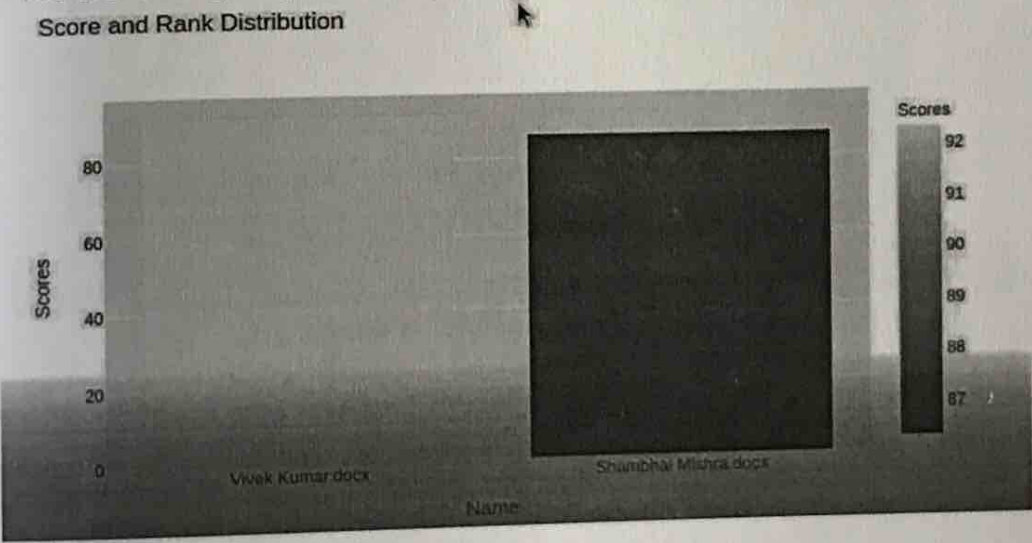


Fig 9.2 Score and Rank Distribution

The above image displays the score and rank distribution among vivekkumar and shambhaimishra.

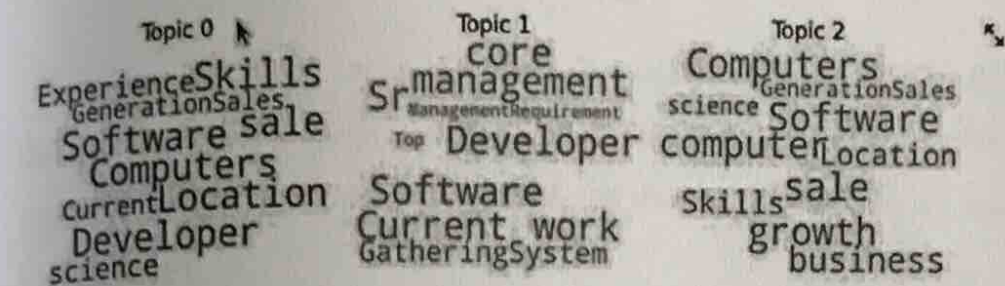


Fig 9.3 Topics and topic related keywords

The above Wordcloud representation shows the topic number and the top keywords that constitutes a topic and this further is used to cluster the resumes

Job No.	Job Desc. Name
0	Director of Engineering.docx
1	Senior Software Developer.docx
2	HTML Developer.docx
3	Revenue Reporting Data Analyst.docx
4	IT Project Manager.docx
5	Lead Technical Program Manager.docx
6	Billing cum Logistics Manager.docx
7	Data Scientist.docx
8	Senior Product Manager.docx

Fig 9.4 Job no and Job description name

The above table depicts the job description along with the job role and this data is used in finalizing resumes based on the scores that we get from the output.

REFERENCES

[1] Rajath V; Riza Tanaz Fareed; SharadadeviKaganurmath," Resume Classification And Ranking Using KNN And Cosine Similarity",international JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY, Volume 10, Issue 08, AUGUST 2021

[2] Jitendra Purohit; Aditya Bagwe; Rishabh Mehta; OjaswiniMangaonkar; Elizabeth George, "Natural Language Processing based Jaro-The Interviewing Chatbot", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), August 2019

[3] Tejaswini K; Umadevi V; Shashank M Kadiwal; Sanjay Revanna," Design and Development of Machine Learning based Resume Ranking System", Global Transitions Proceedings, October 2021

[4] Pradeep Kumar Roy; Sarabjeet Singh Chowdhary; Rocky Bhatia," A Machine Learning approach for automation of Resume Recommendation system", Procedia Computer Science Volume 167, Pages 2318-2327, 2020

[5] AbeerZaroor; Mohammed Maree; Muath Sabha," A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Post", Smart Innovation, Systems and Technologies book series (SIST, volume 72), 2017

[6] Garima Bhardwaj; S. Vikram Singh; Vinay Kumar," An Empirical Study of Artificial Intelligence and its Impact on Human Resource Functions", International Conference on Computation, Automation and Knowledge

Management (ICCAKM), 2020

[7] Anushka Sharma; Smiti Singhal; Dhara Ajudia," Intelligent Recruitment System Using NLP", International Conference on Artificial Intelligence and Machine Vision (AIMV), 2021

[8] Yong Luo; Huaizheng Zhang; Yongjie Wang; Yonggang Wen; Xinwen Zhang," ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment", 2018 IEEE International Conference on Data Mining (ICDM), December 2018

[9] Mujtaba, Dena F., and Nihar R. Mahapatra. "Ethical Considerations in AI-Based Recruitment." 2019 IEEE International Symposium on Technology and Society (ISTAS). IEEE, 2019.

[10] Fahad, SK Ahmed, and Abdul Samad Ebrahim Yahya. "Inflectional review of deep learning on natural language processing." 2018 International Conference on Smart Computing and Electronic Enterprise (ICSEE). IEEE, 2018

SCREENING AND RANKING RESUMES USING NLP AND MACHINE LEARNING TECHNIQUES

Madeshwaran B

Dr.SenduruSrinivasulu

Sarath L

Department of IT

Department of IT

Department of IT

Sathyabama Institute of Science and
Technology

Sathyabama Institute of Science and
Technology

Sathyabama Institute of Science and
Technology

Chennai, India

Chennai, India

Chennai, India

madeshbaskar@gmail.com

sarathsaro555@gmail.com

ABSTRACT:The purpose of a resume screening is to detect the most qualified contenders for a work opening. The system needs real-time ratings based on both machine learning and natural language processing. In order to evaluate applications, we employ a mixture of natural language processing (NLP) and machine learning algorithms. The output would be a resume for a top applicant who meets the requirements specified in the input resumes and job postings. Data from the output can be accessed instantly. Mong will be used for TF-IDF, Cosine Similarity, and string matching. Although current methods are simple and effective, they fall short in terms of precision, efficiency, and computing power.

Keywords: *Deep Learning, Natural Language Processing, Machine Learning, and TD-IDF*

1. INTRODUCTION

The worldwide recruiting market is estimated to be worth \$200 billion. To locate the candidates who are most suited for a certain job description, a large pool of applications must be combed through. A company will get dozens of applications for the position when it posts a job offer. Every recruiter's first step in the hiring process is to screen resumes. When there is an opening at a company, human resources probably receives thousands of resumes and cover letters every day.

Since it requires human engagement, this method currently takes the longest and is the most error-

prone. Humans are limited in ways that make it impossible for them to work continuously. Hence, decreasing efficiency is also a problem. Thus, we have proposed a way to quickly find the essential skill required by evaluating a document or CV and sorting its findings by ability collection bypassing the company's constraint. One tactic we want to use is machine learning. Human resources technology is expanding into the domain of recruitment, with the goal of automating or significantly streamlining processes like resume screening.

New research are conducted daily in a huge number of universities across many different fields. Everyday living benefits from a vast array of research and discoveries, from the field of information technology to that of medical. Due to these changes, many industries today have more opportunities accessible than ever before. It is reasonable to suppose that similar trends exist in other countries' hiring procedures. Although the resume serves as the foundation of the applicant's application, it is the candidate who must build it in the beginning.

Most modern companies prefer to have applicants submit their resumes via an online platform. Once the email has been received, the responses will be organised accordingly. That takes a long time because it's usually sorted by hand. To the contrary, it's possible that many errors are brought on by it. This type of sorting is rather ineffective. This might result in employing someone whose credentials are insufficient for the post or it could result in the manual sorting procedure missing a great contender.

Finding the best candidates in a large pool is a recruiter's most tedious task. For human recruiters to go through hundreds of resumes to locate the top candidates might be daunting when a company gets them. There might be hundreds of candidates for each given position at the company. 75 per cent of advertisements don't show the necessary competence for the job being advertised, on average.

In the age of social media and online talent databases like LinkedIn and Facebook, thousands of candidates may apply for a single post. Automated systems are starting to prevail in modern HR departments. The field is streamlined and improved with resume screening. Hence, human resources professionals have more time to engage in fruitful conversations with both potential workers and candidates. Our suggested design is a digital tool for HR departments that effectively screens the resumes of applicants using NLP technology. Token-Based Similarity and Latent Dirichlet Allocation (LDA) are 2 of the finest ways for comparing information on resumes in the field with that on job descriptions. The best similar resumes are then provided to the recruiter.

2. LITERATURE REVIEW

K-NN, Content-based Recommendation, and cosine similarity are used to evaluate the best applicants in accordance with the job description in (1), and the CVs that match the job description the closest are identified.

In (2), it is hypothesised that a virtual assistant called JARO may conduct interviews by reading the applicants' curriculum vitae (CVs). A list of inquiries to ask the requestor will then result from this. In order to ensure an objective judgement, this expedites the interview process.

In (3), in order to determine whether or not a particular resume is relevant, it uses a neural-network model in conjunction with a few text-processing methods. To address the label shortage issue, we provide a variety of model solutions that either use the pair- or triplet-based loss algorithm or a semi-supervised methods of learning to learn from data.

Multiple regression was used in (4) to assess the concept. The findings demonstrate how greater usage of AI in working contexts increases HR functional performance by demonstrating a substantial positive association between the two criteria.

The research of a hybrid technique for curriculum vitae screening conducted by the authors was detailed in (5). The authors demonstrate empirically, using a dataset based on real-world

recruiting, that their integrated knowledge base performs the classification task with promising accuracy results when compared to existing machine learning-based algorithms for resume categorization.

KNN, Linear SVC, and XGBoost are only a few of the machine-learning techniques employed in the system (6). To accurately forecast specific job profiles based on a textual description, every one of these techniques are stacked into a two-level model and integrated into a single model. The usage of this approach may help both employer's waitlist competitors and job seekers determine whether their resumes are sufficiently organized for the system to identify the right job profiles.

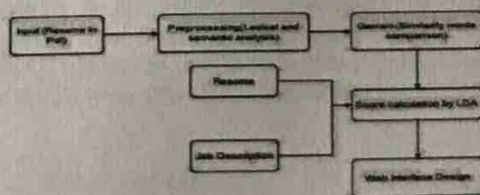
Candidates can apply for open vacancies and submit their resumes through the website's interactive application form (7). NLP and machine learning are used by the corporate recruiter to compare resumes to job requirements.

In (8), we use its web tool to categorise and rank applications. The uploaded resumes are compared to the job description using a tool called Natural Language Processing to determine which ones are best suited for the position (NLP). Then, the resumes are graded and rated in order of best match to worst match. Only corporate recruiters, who are tasked with choosing the best applicants from a wide pool, have access to this ranking.

We can score individual CVs by comparing total fields in CVs with needed fields, and the authors of (9) anticipate that automating this screening process through in-depth analysis of NLP would transform the entire recruitment process by reducing employer duties. In addition, their categorization model has a precision of 74%.

In order to create content that employers can use, the writers assemble data linked to resumes and do the necessary analyses on it (10). The Resume Parser would make it easier for hiring managers to locate the best candidates without wasting time.

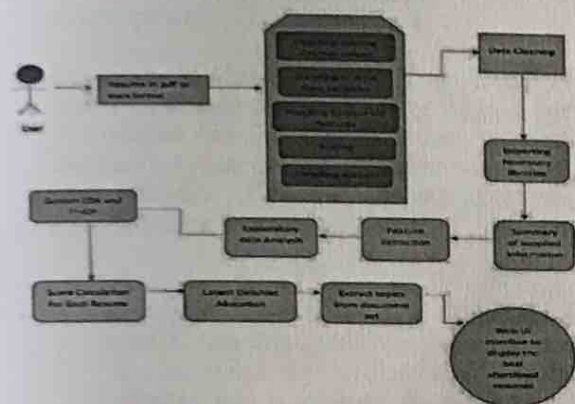
WORKFLOW ARCHITECTURE



The functionality of the suggested model is depicted in graphical form by the system flow diagram. The PDF-based study serves as the initial input, followed by a lexical and semantic analysis as part of the preprocessing phase. Genism

generates a CV and job description by analysing keyword similarities and assigning a score using LDA. All of these steps ultimately serve to shape the look of the website's user interface.

SYSTEM ARCHITECTURE



User and system interactions are shown in the architecture diagram. Data preparation is done once the user submits the resume in pdf or docx format. The relevant libraries are imported, and data cleaning is done. The characteristic is extracted after a summary of the data. The shortlisted resume is shown using the online UI after the score computation.

3. EXISTING SYSTEM

Nowadays, resume screening is done manually through the evaluation of applications. It is physically challenging to go through the resumes of the millions of job seekers in India's enormous employment market to find someone who is a suitable match. This results in a labor hiring procedure that is slow and ineffective, which costs businesses resources. HR must personally sift through each résumé in order to find the applicant whose qualities most closely match those listed in the job description. Because this method requires an abundance of assets and is vulnerable to errors, during the course of the selection process, it is possible that the person who possesses the most relevant experience and skills for the position will be passed over.

4. PROPOSED METHODOLOGY

Our method for rating applications makes use of NLP and machine learning. This AI-powered resume analyzer takes things a step further than keyword searches. After assessing applications, the software provides instant feedback on candidates based on the recruiter's specific requirements. The online application makes an attempt to classify the

applications by contrasting them to the Work Specifications that are provided and using an intelligent understanding of the Job Descriptions itself as input. Following the application of several filters, it lists and recommends the best resumes that are a match for a certain text job description. The matching and rating of applicants takes place instantly thanks to natural language processing in the programme. This programme use Mong in place of more standard approaches to problems like string matching, cosine similarity, overlap coefficient, and NLP.

We use an alternate approach for our work since it focuses mostly on the information that is contained inside the resumes. From these resumes, we extract relevant skills and other information that helps us match individuals with specific job descriptions.

5. PROPOSED TECHNIQUE

Module 1: Collecting data and Pre-Processing data

a. Data Collection

The term "data collection" is used to describe the steps used to amass data for use in training deep neural networks. The data should be stored in a format suitable for the difficulty level. Kaggle provides us with the resume dataset in CSV format. We use this data set to both train and test our model.

b. Data Pre-Processing

A technique used in data mining known as "data preprocessing," it cleans and organises raw data in preparation for future analysis. When data is preprocessed, it is transformed into a format that enables it to be processed more rapidly and effectively in subsequent steps of machine learning, data mining, and various other data science activities.

The methods are frequently used right from the start of the machine learning and AI development pipeline to assure accurate results. In spite of the fact that there is a wide variety of software and methods available for data preparation, we have decided to focus on the following criteria for our dataset:

a. Data Cleaning

The term "data cleaning" refers to the act of rectifying data that has been corrupted, improperly organised, duplicated, or found to be otherwise lacking in quality. We have used a number of techniques to examine this when integrating multiple data sources to determine

whether any data is duplicated or miscategorized.

This segment fixes structural problems, handles missing information, and removes duplicates from the database. As a last resort, we get rid of observations that have missing values, but we know that doing so can ruin the overall picture.

On the basis of the other facts, we come up with an estimate for the missing figures, but this method can introduce bias into the data because it is based on speculation rather than hard evidence. To clean up our data, we followed these procedures:

- **Removing HTML Tags:** When we extract data from different resumes, the data is frequently written with emails, websites, and work links. In this context, HTML tags like header, body, paragraph, strong, and many others may be encountered and processed as part of the data extraction process. We can easily strip the HTML tags from the text using regular expressions. The objective is to create a regular expression that will remove all of the tags from the document. Therefore, the syntax that looks like this is used:

```
without_html = re.sub(pattern=r"<.*>", repl=' ', string=text)  
print(f'{without_html}')
```
- **Removal of Hashtags:** Hashtags must be removed in order to stop them from obscuring our process. We are aware that hashtags never contain any whitespace and are always composed of letters, numbers, and the # sign. Tokens will be created by dividing the text whenever whitespace occurs. Using the `re.sub` technique, we achieve the same result. The `re.sub()` function may be found in the Python Regular Expressions (`re`) library. Any occurrences of the supplied pattern that are found identical to the returning phrase will have the replace string applied to them.
- **Removing Punctuations:** When working with the texts in the dataset, because we run into problems so regularly, we often have to take particular characters out of the text. This is necessary because of the regularity with which we come into

problems. In this section, the punctuation marks that should go in lieu of characters are replaced with regex. We eliminate every punctuation using a specific regex and substitute an empty text in its stead. The starting material is printed after initializing the string and adding any appropriate punctuation. The `re.sub` command is then used to build a function that eliminates any punctuation. At this stage, the result is written out without any punctuation being added to it.

b. Removing Extra Whitespace

When analysing text datasets, it occasionally occurs that there is a plurality of space amongst words in documents, which is generally unwanted. This can be a problem because it makes it more difficult to read the texts. These scenarios might happen at any time. These kinds of problems are a possibility in web development, and when they emerge, they usually need fixing.

The `join()` method, the `strip()` function, and the `split()` function can all be used together to complete this operation. This is accomplished in two distinct phases. After chopping up the phrase into a set of terms, we make use of the `connect` function to put a single space in between each of the newly created links.

c. Removing Stop Words

When indexing documents to be searched and retrieved in response to a search query, a search engine can be set up to skip over instances of certain words, or "stop words," that are used frequently.

These words ought not to be saved in our database, nor should they be processed at the moment. Maintaining a record of "stop words" enables us to get rid of them in a way that is both speedy and simple. A lexicon of "stop words" for sixteen different languages is included in the Natural Language Toolkit (NLTK) module for the Python programming language. The NLTK data directory is where you'll find them. As a result, we employ the same method to scrub our data set of stop terms. To begin, we use the following command to verify the English dictionary's list of stop words:

```
import nltk  
from nltk.corpus import stopwords  
print(stopwords.words('english'))
```


In the new code, we bring in the `nlk.corpusstopwords` and the `nlk.tokenize` word tokenize packages. After that, we used the command to add the English stop words to the file:

```
stop_words= set(stopwords.words('english'))
file1 = open("text.txt")
```

So that our collected resume data is free of stop words, we read the file's content as a stream and append the file.

d. Tokenization

The technique of producing tokens by extracting them from a string of text is known as tokenization. Tokens can be thought of as pieces in the same way that words can function as tokens in phrases and sentences can work as tokens in paragraphs.

Tokenizing phrases and words in a text is a breeze with NLTK's `tokenize` module. To begin, we'll bring in the necessary NLTK library functions. Nevertheless, `nlk` must be imported first. Next we use `nlk.tokenize` to bring in words to be tokenized. We wait to print the result.

e. Plotting The Graph

The categories versus. counts graph is plotted using Matplotlib. The preferred library for data visualization in Python is Matplotlib. Without using cumbersome graphical user interfaces, we can easily create graphs and charts as well as static raster or vector files.

To install Matplotlib, we use the "pip install matplotlib" command. We import `matplotlib.pyplot` as `plt` into our notebook to finish. We can quickly create a plot with only one easy Matplotlib command.

In order to put the finishing touches on our plot, we will use the `xlabel()`, `ylabel()`, and `title()` methods to label our axes with descriptive information and to give our diagram an appropriate title (). Adding the `plt.show()` call to the very end of your Python script will accomplish this.

We are able to construct graphs with the dimensions and dots per inch (DPI) that we choose because to the `fig` size and `dpi` parameters that Matplotlib provides. The figure size is the product of the image's width (in inches) and height (in inches). The result is given in inches. The `dpi` refers to the image's resolution (pixel-per-inch).

In the absence of `figsize` and `dpi`, Matplotlib will fall back on its defaults. With Matplotlib, We are capable of producing graphics of a professional standard in a number of kinds of files (png, jpg, svg, pdf, and so on). The graphical representation will be saved when you use the `savefig()` method.

f. Creating Word Cloud

A "Word Cloud" is a specific sort of text-based visualisation in which the quantity of every word indicates the frequency with which they appear or the importance they hold. A word cloud can be an effective tool for highlighting important details hidden within a body of text. Word clouds are frequently used in the analysis of data collected from social networks. Word clouds can be made using Python using the `matplotlib`, `pandas`, and `word cloud` modules. To begin, we use the commands `pip install matplotlib`, `pip install pandas`, and `pip install word cloud` to get these packages set up.

Next, `pandas` are imported as a `pd` and `matplotlib.pyplot` as a `plt`. We also import `word cloud` and `stopwords` from `word cloud`. Next, we used `df= pd.read_csv()` to read our pre-processed CSV file (). The next step is to construct a text variable with a single string, including each review. Python's `join` function can be used for this purpose. The word cloud generation process has reached a crucial moment! Our word cloud will only include the most vital components.

In our prototype, We make use of the command to generate a word cloud, and while doing so, we define its dimensions, as well as the largest font size that should be used for the sentences that are most important, and we leave out any stop words. Emphasis less words like that, was, is, and they are examples of stop words in our dataset. Also, we selected the backdrop colour from a range of hues. It is currently set to black by default. In addition, the color theme was set using the "colormap" code. The produced wordcloud is now plotted using the `imshow()` function of Matplotlib, which is the last step. As a result, the dataset from the resumes contains all the crucial terms.

Module 2: Model Construction and Instruction

a. Creating The LDA Model

To find the top candidates for the open post, we'll create a training LDA model, an extraction TF-IDF model for the job

description, and a resume comparison and selection mechanism.

The Term Frequency - Inverse Document Frequency (TF-IDF) metric can be utilised to ascertain a word's level of significance in a given context. It gives a TF-IDF score that indicates the relevance of the term in the overall context of the analysis, depending on the amount of times it shows up in the data. This value is determined by the frequency with which the word appears in the dataset.

It is possible to calculate the frequency of a word by first counting the number of instances it comes up in a particular phrase followed by dividing that number by the total amount of phrases used in the expression. The following is a formula for determining the frequency of the terms. Divide the average of the document's total sentences by the number of lines that include the phrase of interest to determine the reverse article frequency. This will give you the inverse document frequency. We acquire a number in the format of a vector by multiplying the TF equation by the IDF formula. This gives us value. A graph illustrating its relative relevance along with an array of terms will be used to indicate its value.

$$TF(t, d) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d}$$

$$IDF(t) = \log \frac{N}{1 + df}$$

$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$

The graphical representation of TF-IDF vectorization shows that the more important word is inserted first, in declining order of its significance, producing an inverted growth curve. This is because the greater essential word has a higher TF value. With this method, you can easily identify the most relevant sets of terms connected to specific occupations and skill sets, and then utilise those terms to get relevant data and train an LDA model.

Latent Dirichlet allocation (LDA) is a generative statistical model used in natural language processing to explain similarities between data sets by means of unobserved groups. If, for the sake of the contention, sightings are terms that are collected into records, then it is plausible to suppose that any given document contains a range of thoughts and that the presence of each word is related to some feature of those ideas. This is because each document has words, and words

are associated with ideas. Topic models like LDA are used as an illustration.

Using Dirichlet distributions and processes, LDA is both a tool and a method for subject modelling. After that, the software evaluates the applicant's responses to the job advertisement and gives each résumé a score based on how well those responses meet the specifications of the posting. A score will be assigned to each candidate's resume, and the candidates in ascending order of score will be shown first.

b. Training Our Model

Before beginning the process of training the model, the data was initially organised into a set to be used for training and a testing set. Estimating the efficacy of machine learning algorithms well suited for prediction-based algorithms/applications is done with the train-test split. Using this uncomplicated method, the results of our very own model for machine learning can be contrasted to those obtained by an automated system. The Test set is composed of seventy per cent simulated data, while the Training set is composed of thirty per cent real data.

The performance of our model for machine learning may be evaluated by first dividing the information set into training and testing sets, and then comparing the results of the two sets. Adjustments are made to the model by employing the train collection, and its individual parts are singled out. The second set, which is referred to as the data from the test set, is the sole one that is used for creating projections.

The following procedure is the one that we use to partition our dataset into sets for training and testing. We bring in the packages for both Sklearn and Pandas. The train test split splitter function can be found in the scikit-learn library's model selection module (). The CSV file is subsequently imported using the read_csv() function. The df variable now contains the data frame.

Then we define the test size as 0.3, which means that thirty per cent of our data is taken for testing, and the rest seventy per cent is used for training data. We also put the random_state=0 so that data is randomly split into these two datasets.

The next step is to train our dataset. Following the partitioning of the data into a training set and a testing set, we use an algorithm based on

machine learning to evaluate the performance of the training set to determine how well it is performing. Scikit is used to perform data fitting in this investigation. When it comes to Python machine learning libraries, Scikit-learn is your best bet. The Sklearn toolbox gives users access to a wide variety of machine learning as well as statistical modelling skills. A few examples of these capabilities include grouping, regression, classification, and dimensionality reduction.

One of these is the 'fit' method in scikit-learn. Following the initialization of the model, the 'fit' technique is used to train the algorithm using the provided data. In essence, that's all it does. So, the training data is utilised by the sklearn fit method to train the machine learning model. Following this, we may then continue with the method of machine learning by utilising more scikit learn algorithms, including forecast and score.

The following is a description of how we apply the fit technique to the sample data. As can be seen in the example shown below, the model instance must first be defined before the fit function can be used to analyse the data. In the final parenthesis, we list the features and label vector used in the training dataset. You may also hear these datasets referred to as X train or Y train. For optimal model fitting, X train, the X-input to fit(), is a 2-dimensional format. Otherwise, there is a flaw in our model.

The performance of our LDA model is then assessed. The accuracy is the percentage of correct classifications, and it can range from 0 to 1. Intuitively, we check if the model correctly classified the data by comparing the predicted class to the actual class. We use Sklearn's accuracy score function for this purpose. The following command is used for the same purpose:

```
print(accuracy_score(y test, y pred)).
```

Module 3: Creating A Web App

After a model has been trained and its weights have been recorded, a web application can be built using Streamlit to provide a user interface for the model (UI). Streamlit is a Python framework that is open-source and is used for the development of web-based applications for machine learning and data science.

Using Streamlit, we can quickly build and deploy new web applications. The online software Streamlit streamlines the iterative process of

programming and observing results. Streamlit is used to construct Python-based apps.

The Streamlit Python package is installed with the pip install streamlit command. The following step is to run the streamlit commands by running the produced Python script with the streamlit run ourscript.py> command. This will complete the current step. We were able to make use of a wide variety of the widgets that Streamlit offers, such as the st.selectbox, st.checkbox, st.slider, and others.

The st.sidebar command makes it simple to rearrange any number of widgets or pieces of information. With this method, we were able to better arrange the information in the left-hand margin. We only used st.sidebar.selectbox to show a pickbox in the left panel.

Then, we make a title and a DataFrame in the web app with streamlit. We can now type streamlit run streamlit app.py into our console. A URL is returned after the command has been run. If we select the link, we will be redirected to our site or app.

Users can upload multiple resumes and specify the position they are applying for on our website, and then receive a resume score that can be used to determine which candidates to forward to the next round of the hiring process.

6. OUTPUT SCREENSHOTS

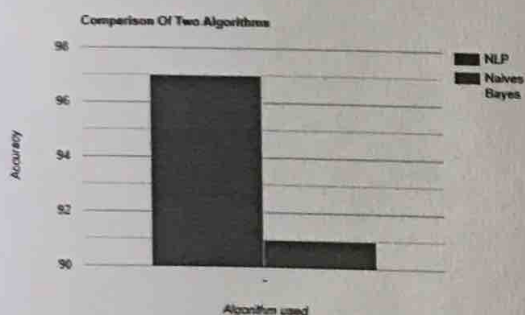
Output 1:The WordCloud represents the most frequently occurring keywords in the model used to screen resumes. The phrases used most frequently include computers, software, skills, management, and current job.



Output 2: It displays a list of the many job profiles we added to our model and their descriptions.

Job No.	Job Description
0	Director of Engineering dept
1	Senior Software Developer dept
2	AI/ML Developer dept
3	Revenue Reporting Data Analyst dept
4	IT Security Manager dept
5	Lead Technical Program Manager dept
6	Billing team Logistics Manager dept
7	Data Scientist dept
8	Senior Product Manager dept

7. ALGORITHM COMPARISON



We undertake calculations to determine the accuracy rates of both models and conduct a comparison of their performance metrics. NLP is proven to be superior to Naive Bayes in terms of accuracy.

8. CONCLUSION

The results of the model look good. Automating the procedure of placing freshmen in positions that make the most effective use of their passions, experiences, and self-reported talents is made significantly easier by the existence of algorithms such as the resume classifier. The Resume Screening System eliminates time-consuming and inaccurate manual screening, guaranteeing that no qualified applicant is overlooked. The outcomes of an effectiveness test meant to evaluate both the validity of the applicant's CV and his or her expertise in the necessary subject matter are considered by the system when determining whether or not to accept an applicant for a position. The system bases its decision on the requirements of the firm as well as the results of the test.

9. FUTURE SCOPE

The prediction makes it possible to choose the application without human intervention. This model educates the system to assess and pick the appropriate expertise from a CV that is given as the input, which helps to boost accuracy over the long term. This leads to robust and effective prediction enhancement.

REFERENCES

1. Pradeep Kumar Roy; Sarabjeet Singh Chowdhary; Rocky Bhatia, "A Machine Learning technique for automation of Resume

Recommendation system," *Procedia Computer Science* Volume 167, Pages 2318-2327, 2020

2. Jitendra Purohit; Aditya Bagwe; Rishabh Mehta; Ojaswini Mangaonkar; Elizabeth George, "Natural Language Processing based Jaro-The Interviewing Chatbot," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), August 2019

3. Yong Luo, Huaizheng Zhang, Yongjie Wang, Yonggang Wen, and Xinwen Zhang, "ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment," 2018 IEEE International Conference on Data Mining (ICDM), December 2018 [ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment]

4. 2020 International Conference on Computation, Automation, and Knowledge Management (ICCAKM): Garima Bhardwaj, S. Vikram Singh, and Vinay Kumar, "An Empirical Study of Artificial Intelligence and Its Impact on Human Resource Functions."

5. Mohamed Maree, Muath Sabha, and Abeer Zaroor's book chapter from 2017's Smart Innovation, Systems and Technologies volume 72

6. Ranikka Ransing Nikita Bhrugumaharshi Emberi, Kailas Mahavarkar, and Akshaya Mohan, "Screening and Raking resumes based on Stacked Model, 2021, IEEE

7. Amin Sujit Web Application For Screening Resumes, Nikita Jayakar, Sonia Sunny, Pheba Babu, M. Kiruthika, and Ambarish Gurjar, 2019 IEEE.

8. M. Alamelu D. Sathish Kumar; R. Sanjana; J. Subha Sree; A. Sangeerani Devi; D. Kavitha "Resume Validation and Filtration using NLP, 2021, IEEE IEMECON

9. M.F. Mridha; Rabeya Basri; Muhammad Mostafa Monowar; Md. Abdul Hamid "A Machine Learning Approach for Screening Individual's Job Profile Using Convolutional Neural Network", 2021 IEEE

10. Anushka Sharma, Smriti Singhal, and Dhara Ajudia, 2021 AIMV, Artificially Intelligent and Mechanics Voices