**Resume Analyzer**

***A Project Report submitted in partial fulfilment of the requirements***

***for the award of the degree of***

**Bachelor of Technology**

#### in

***Computer Science and Engineering***

**By**

**Name of Student: Kaustubh Dubey**

**Roll No: 201500332**

**Name of Student: Shreyash Varshney**

**Roll No: 201500675**

**Group No.:50**

Under the Guidance of Dr. Mayank Srivastava

Department of Computer Engineering & Applications

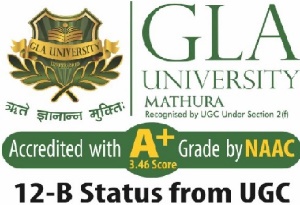
# Institute of Engineering & Technology



**GLA University**

**Mathura- 281406**

**May, 2024**

**Department of Computer Engineering and Applications**

**GLA University, 17 km Stone, NH#2, Mathura-Delhi Road, P.O. Chaumuhan, Mathura-281406 (U.P.)**

**Declaration**

I hereby declare that the work which is being presented in the B.Tech. Project **“Resume Analyzer”**, in partial fulfillment of the requirements for the award ofthe ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of **Dr. Mayank Srivastava.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Student: Kaustubh Dubey Name of Student: Shreyash Varshney

University Roll No.:201500332 University Roll No.:201500675

**Certificate**

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor**

(**Dr. Mayank Srivastava**)

Designation of Supervisor

Dept. of Computer Eng, & App.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Co-Ordinator Program Co-Ordinator**

**(Dr. Mayank Srivastava)** (**Mr. Jitesh Kumar Bhatia**)

Associate Professor Assistant Professor

Dept. of Computer Eng, & App. Dept. of Computer Eng, & App.

Date:

**ACKNOWLEDGEMENT**

I would like to express my profound gratitude to Mr. Ashok Bhansali, Head of department, and Mr. Neeraj Aggarwal, Dean of university for their contributions to the completion of my project titled “Resume Analyzer”.

I would like to express my special thanks to our mentor **Dr. Mayank Srivastava** **(**Dept. of Computer Eng, & App**)** for his time and efforts he provided throughout the year. Your useful advice and suggestions were really helpful to me during the project’s completion. In this aspect, I am eternally grateful to you.

I would like to acknowledge that this project was completed entirely by me and group member.

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Student: Kaustubh Dubey Name of Student: Shreyash Varshney

University Roll No.:201500332 University Roll No.:201500675

ABSTRACT

*In the contemporary landscape of online job searching, job seekers face the daunting task of sifting through a deluge of recruitment information to identify relevant opportunities. To address this challenge, the rapid evolution of job recommender systems has emerged as a pivotal solution, harnessing the power of user profiling and advanced recommendation technologies. This study presents a pioneering recommendation system tailored explicitly for online job seekers, leveraging cutting-edge methodologies in Natural Language Processing (NLP) and Machine Learning (ML). By integrating techniques such as NLTK, Resume Parser, KNN (K-Nearest Neighbors), and One Vs rest classifier, the system offers the capability to accurately predict optimal professions based on individual resumes. The methodology involves systematically harvesting job listings from Naukri.com, a leading online job portal, and subsequently matching requisite skills from job listings with individual proficiencies. This matching process is facilitated by a cosine similarity algorithm, enabling the personalized ranking and presentation of job recommendations to the user. By providing a holistic and efficient solution to streamline the job search process, this research contributes significantly to the burgeoning field of job recommender systems. It promises to alleviate the burdens faced by contemporary job seekers in the digital age.*

*Keywords — NLTK, Resume Parser, KNN, One Vs rest classifier, Natural Language Processing.*

Declaration ii

Certificates ii

Acknowledge iii

Abstract iv

List of figures v

CONTENTS

**Chapter 1: Introduction 1**

* 1. Overview and Motivation 1
  2. Objective 1
  3. Issues and Challenges 1
  4. Contribution 3
  5. Organization of the Project Report 3

**Chapter 2: Literature Review 4**

**Chapter 3: Proposed Work** **33**

3.1 Data Collection 33

3.2 Technology Used 33

3.3 System Architectures 34

3.4 Software Design 35

3.5 Use Case Diagram 37

**Chapter 4: Model** **38**

4.1 KNN 38

4.2 One Vs Rest 39

4.3 TF-IDF 39

**Chapter 5: Training the Model** **41**

5.1 Training the dataset 41

5.2 Brief Description of Library 41

5.3 Exploring Resume 45

5.4 Words into Categorical values 46

5.5 Splitting Data 48

5.6 Classification Report 48

5.7 Vectorization 49

5.8 Cleaning the Data 49

5.9 Prediction System 50

**Chapter 6: Implementation 53**

**Conclusion**  **57**

**Reference**  **58**

**LIST OF FIGURES**

Figure 1: Structured and graphical Representation of Resume Data 43-46

Figure 2: Starting Window (User Side) 53

Figure 3: Strating Window (Admin Side) 53

Figure 4: Uploading Resume 54

Figure 5: Extracting Resume 54

Figure 6: Choose User 55

Figure 7: Admin Side (Showing Information of User Data) 55

Figure 8: Viewing Resume Data in Excel Sheet 56

# CHAPTER 1

# INTRODUCTION

## OVERVIEW AND MOTIVATION:

The modern job market is characterized by a vast pool of talent, making the recruitment process for employers increasingly complex. The initial phase of candidate screening, often involving the analysis of resumes, is a time-consuming task for human resources professionals. The growing need for efficient solutions to sift through large volumes of resumes has led to the development of automated tools leveraging natural language processing (NLP) and machine learning (ML) techniques.

## 1.2 OBJECTIVE

Analyzer, a tool that streamlines the resume screening process for recruiters. The key goals include:

Efficient Resume Analysis: Develop algorithms and models capable of extracting relevant information from resumes, including skills, experiences, and qualifications.

User-Friendly Interface: Create an intuitive and accessible user interface that allows recruiters to interact seamlessly with the Resume Analyzer, facilitating a smooth and efficient workflow.

Automation of Initial Screening: Automate the initial phase of candidate screening, reducing the time and effort required for recruiters to identify suitable candidates.

## 1.3 ISSUES AND CHALLENGES

#### 1.3.1 UNSTRUCTURED DATA

Issue:

Resumes often arrive in various formats, including PDF, DOC, and plain text. Handling unstructured data and ensuring consistent preprocessing posed a significant challenge.

Resolution:

Implemented a robust preprocessing pipeline capable of handling diverse resume formats. Utilized libraries such as pdf2text and python-docx to extract text from PDFs and DOC files, respectively. Applied text cleaning techniques to standardize formatting.

#### MULTILINGUAL RESUMES

Issue:

The Resume Analyzer primarily focuses on English-language resumes, but multilingual resumes presented challenges in accurate language interpretation and information extraction.

Resolution:

While the current version is optimized for English, future iterations will incorporate multilingual support, employing language-specific models and dictionaries. This recognizes the need for a globally inclusive tool.

#### REAL-TIME PROCESSING

Issue:

Ensuring real-time processing of resumes while maintaining accuracy in analysis posed a challenge, especially when dealing with a large volume of concurrent requests.

Resolution:

Optimized algorithms and employed parallel computing techniques to enhance processing speed. Utilized Docker for containerization, allowing the system to scale dynamically based on demand.

#### USER TRAINING

Issue:

Ensuring that recruiters effectively utilize the Resume Analyzer without significant training overhead presented a usability challenge.

Resolution:

Developed comprehensive user documentation and tutorials. Conducted training sessions and workshops to familiarize users with the system's features and functionalities. Implemented intuitive design principles to enhance user experience.

Addressing these issues and challenges required a combination of technological innovation, continuous improvement, and a commitment to ethical practices. The resolutions adopted for each challenge contribute to the overall robustness and reliability of the

Resume Analyzer. Ongoing monitoring and updates will be essential to address emerging challenges and ensure the tool's effectiveness in a rapidly evolving job market.

## CONTRIBUTION

#### ENHANCED INFORMATION EXTRACTION

Contribution: The application leverages advanced NLP techniques to extract and categorize relevant information from resumes, including skills, experiences, and qualifications.

Impact: Recruiters benefit from a more comprehensive understanding of candidate profiles, streamlining the initial screening process.

#### MULTIFACETED ANALYSIS

Contribution: The Resume Analyzer incorporates sentiment analysis, providing recruiters with insights into a candidate's tone and overall context.

Impact: Recruiters gain a nuanced understanding of a candidate's communication style, contributing to a more informed screening process.

#### USER TRAINING AND SUPPORT

Contribution: Comprehensive user documentation and training initiatives ensure effective utilization of the Resume Analyzer by recruiters.

Impact: Recruiters can leverage the tool confidently, thanks to user-friendly resources and training initiatives.

## ORGANIZATION OF THE PROJECT REPORT

The project report is organized into distinct chapters, each dedicated to a specific aspect of the development and implementation of the Resume Analyzer. This section provides an overview of the content and structure of each chapter, outlining the progression of the report.

# CHAPTER 2

# LITERATURE REVIEW

This chapter delves into existing literature on resume analysis, natural language processing (NLP), and machine learning (ML). It provides a comprehensive overview of relevant concepts, technologies, and existing tools in the field. The literature review serves as the foundation for understanding the current state-of-the-art and informs the theoretical framework adopted in the project.

The literature on resume analyzers highlights a growing interest in automating the resume screening process to improve efficiency and effectiveness in candidate selection. Various studies have explored the application of machine learning and natural language processing techniques to extract relevant information from resumes and match candidates with job requirements.

Researchers have investigated different approaches, including keyword matching, semantic analysis, and deep learning models, to classify resumes based on their suitability for specific job roles. Additionally, feature engineering methods such as TF-IDF, word embeddings, and named entity recognition have been utilized to capture important resume attributes like skills, education, and work experience.

Challenges such as unstructured data, class imbalance, and privacy concerns have been addressed in the literature through techniques like data preprocessing, feature selection, and privacy-preserving algorithms. Moreover, studies have emphasized the importance of fairness, transparency, and interpretability in resume analyzer models to mitigate bias and ensure equitable candidate evaluation.

Overall, the literature suggests that resume analyzers hold promise for streamlining the recruitment process, reducing manual effort, and improving the quality of candidate-job matching. However, further research is needed to address scalability issues, enhance model accuracy, and incorporate user feedback for continuous improvement.

**2.1 AN AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND SIMILARITY[1]**

## 2.1.1 INTRODUCTION

With the rapid increase in internet connectivity, there has been a change in the recruitment process of all major companies. With the help of online job postings in various job portals and websites, recruiters are able to attract a wide variety of people for their openings. Though e- recruitment has provided convenience and savings for both the recruiters and the applicants, some new challenges arise.

Large companies and recruitment agencies often receive thousands of resume every day. This situation is even more aggravated due to the higher mobility of workers and in situations of economic distress, where many people are looking to get jobs. With less than 5% of people to be selected from these applications, it is impractical for the recruiters to manually go through each and every resume for these limited number of openings. Another problem faced by the organizations is that there is no one standard resume format used by these applicants.

People come from varied fields of profession and have different backgrounds. Each one of them has had different types of education, has worked on different projects and thus has a unique style of presenting his/her credentials in the resume. Resumes are unstructured documents that come in various file formats (.pdf, .doc, .docx, .jpg, .txt etc.) and their content is not written according to standard formats or templates.

This means reading resumes is not simple and thus recruiters spend a large amount of time going through the resumes for selecting the right candidates. Many job portals and external websites came up to reduce this difficulty of handling unstructured and diverse resumes. These require candidates to manually fill up all the information of their resume in an online form in a structured manner, thus creating a candidate metadata.

The problem with this approach is that it requires redundant efforts on the part of the candidates, and they often miss out on filling complete information in these templates. These websites use a generic format that isn't domain-specific and thus is not optimal for all jobs. The employers then use these templates to apply the keyword-based search for shortlisting candidates. This keyword-based search functionality is insufficient to match candidates with the job description (Malinowski, Jochen, et al., 2006). This is so as it relies only on the existence of certain required keywords and has various extraction limitations like avoiding natural language semantics such as synonyms, word combinations, and contextual meaning of the content present in the resume (Singh, Amit, et al., 2010). Therefore, these Boolean search methods often give irrelevant results and deserving candidates miss out on opportunities of being shortlisted.

### 2.1.2 RELATED WORK

The recruitment process in today’s world has witnessed a major change with the evolution of technologies like the Internet. The following section summarizes some of the literary work performed in this domain of e recruitment systems.

The proposed solutions use various approaches with the aim of achieving automated screening of candidates. The work presented as EXPERT (Kumaran, V.S. and Sankar, A., 2013) proposed the use of ontology mapping for screening candidates for the given job description. It included three phases of operation which were the creation of candidate ontology, construction of job criteria ontology document and then finally mapping of both of these to evaluate which candidates are eligible for the job. In 2012, an automated job screening system was proposed (Faliagka, Ramantas, Tsakalidis, and Tzimas).

It discusses different machine learning algorithms and uses Support Vector Regression to create a list of ranked candidates for the given job. Another work presented (Weathington and Bechtel, 2012) that described how social media (e.g. LinkedIn, Facebook, etc.) information of the applicants can be used for recruitment decisions.

In another approach, the work that was proposed (Laumer, S. and Eckhardt, A., 2009) described a collaborative filtering-based system to recommend applicants that best fit a job. We also studied a work (Malinowski, Weitzel, and Keim, 2008) that considered matching interpersonal compatibility of the team members with the prospective hire to make recruitment decisions. 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 description.

**2.1.3 METHODOLOGY**

In this section, we describe the concepts that facilitate the construction of the proposed Automated Resume Screening System. The system works in two phases as described below.

**2.1.3.1 INFORMATION EXTRACTION:**

The first phase of our proposed system involves information extraction using Natural Language Processing. The information in the resumes is not present in a structured format. There are noises, inconsistencies and irrelevant bits of data which is of no use to the recruiters. The objective is to derive relevant keywords from the unstructured textual data in the resume without any need of human crawling efforts. Using techniques like Tokenization, Stemming, POS Tagging, Named Entity Recognition, etc., our system obtains important job-related content (skills, experience, education, etc.) from the uploaded candidate resumes. The result is a summarized version of each resume in a JSON format which can be easily used for further processing tasks in the next phase of this resume screening system.

**2.1.3.2 TOKENIZATION:**

After converting the various resume formats (.docx, .pdf,.jpg, .rtf, etc.) into text, we begin the tokenization process to identify terms or words that form up a character sequence. This is important as through these words, we will be able to derive meaning from the original text sequence.

Tokenization involves dividing big chunks of text into smaller parts called tokens. This is done by removing or isolating characters like whitespaces and punctuation characters. Tokens are sentences initially (when tokenized out of paragraphs) and then are further split into individual words. By performing Tokenization, we can derive information like the number of words in a text, frequency of a particular word in the text and much more. The tokenization can be performed in multiple ways such as using Natural Language Toolkit [NLTK], the spaCy library, etc. Tokenization is a mandatory step for further text processing such as removal of stop words, stemming and lemmatization.

**2.1.3.3 STEMMING AND LEMMATIZATION:**

It is frequently seen that a single word of the English language is used in various different forms in different sentences according to its grammatical rules. For example -implement, implemented and implementing are just different tenses of the same verb. This situation results in the need to reduce all the altered or derived forms of a word to their central stem or base so that these derivationally related words with similar meanings are not considered to be different from each other.

Both Stemming and lemmatization have the same objective but differ in their approach. “Stemming is the mechanism of reducing inflected or derived words to their word root, or stem. It is a crude heuristic process that involves chopping off the ends of words to achieve this objective, and often includes the removal of derivational affixes” (Jivani, A.G., 2011).

These are rule based algorithms in which a particular word is tested on a range of conditions and then based on a list of known suffixes, decides how to cut it down. It is noteworthy that the root derived after stemming may not be identical to the morphological root of the word. Due to the heuristic-based approach of stemming, it suffers from issues such as under-stemming and over-stemming. Some common stemming algorithms used are Porter Stemmer, Snowball stemmer, and Lancaster stemmer.

On the other hand, lemmatization is the process of utilizing a language dictionary to perform an accurate reduction to root words. Unlike Stemming which simply cuts off tokens by simple pattern matching, lemmatization is a more careful approach that uses language vocabulary and morphological analysis of words to give linguistically correct lemmas.

This means lemmatization utilizes the knowledge of context and therefore can differentiate between words that have different meanings based on parts of speech. For the English language, our system uses the WordNet Lemmatize (based on Word New Database) provided by the NLTK python package. 3.1.3 Parts of speech (POS) tagging.

It is a process of assigning grammatical information to a word based on its context and its relationship with other words in the sentence (Gelbukh, 2014). The part-of-speech tag specifies whether the word is a noun, pronoun, verb, adjective, etc. according to its usage in the sentence. It is important to assign these tags so as to understand the correct meaning of a sentence and for building knowledge graphs for named entity recognition. This process is not as simple as mapping a word to their corresponding part of speech tags. This is so as a particular word may have a different part of speech based on different contexts in which it is used. For example: In the sentence “I am building a software”, building is a Verb, but in the sentence “I work in the tallest building of that street”, building is a Noun. Also called grammatical tagging or word- category disambiguation, it is a supervised learning solution that analyses the features such as the preceding word, following word, first letter capitalized or not, etc. to label the words after tokenization. Rule-Based POS tagging, Stochastic POS tagging, and Transformation based tagging are mostly used (Hasan, 2006).

**2.1.3.4 CHUNKING**

Itis a process that aims to add more structure to sentences by grouping short phrases with parts of speech tags. Because parts of speech tags alone cannot give information about the structure of the sentence or the actual meaning of the text, chunking combines parts of speech tags with regular expressions to give a result as a set of chunk tags like Noun Phrase (NP), Verb Phrase (VP), etc. Also called Shallow Parsing, it involves the construction of a parse tree that can have a maximum one level of information from roots to leaves. This ensures there is more information than just part of speech of the word without needing to create a full parse tree. Chunking segments and labels multi-token sequences (Bird, Klein and Loper, 2009), mostly making groups of “noun phrases” that are used for finding named entities.

### 2.1.4 RESULTS

For testing the system, we have used the job description posted by Amazon.com Inc. inviting applicants for the job position of a Software Developer Engineer at its Bengaluru office. We have taken some relevant resume samples from the Internet which we’ll pre-process, perform extraction, summaries and then calculate cosine similarity on to create a ranked list of candidates for this job. For feature selection, we have selected parameters such as Educational Degree, University, Total Experience, Designation with the Organization in which the candidate has worked in the past, and most importantly the skills that are needed for the job. Figure 2 given below shows the sample output after the information extraction from a resume is performed successfully.

# 2.2 RESUME RANKING USING NLP AND ML[2]

### 2.2.1 ABSTRACT

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. Beside the information provided by the resume we are going to read the candidates social profiles (like LinkedIn, GitHub etc.) which will give us the more genuine information about that candidate.

#### 2.2.2 OBJECTIVE

The major objective of our system is to take the current resume ranking system to other level and makes it more flexible for both the entity. 1) Candidates, who has been hired. 2) Client company, who is hiring the candidates. Candidates, who has been hired : Candidates who are searching for jobs after been graduated. Out of those, major number of candidates are so much desperate that they are ready to work on any post irrelevant to their skill set and ability.

The main reason behind this unemployment is like a cancer to our society, if a guy 1 is not got place after been passed out for 1yr, society include relatives starting blaming that guy. Insite of this reason the candidate is ready to work in any condition, on any post. So, they don’t have to face those situations. Where our system helps such candidates to get hired by such a company or an organization who really worth their ability and their skill sets. Where our algorithm will work in such a way that with the help of the previous result and previous ranking constraints, it will try to optimize the current result, which we called it Machine Learning.

This will make sure that the relevant candidate is been hired for that particular vacancy. You can say best possible candidate. Client company, who is hiring the candidates : Like I am the owner of a particular organization, obviously my aim would be to create such a team which is the best team in the world. It is like, if there is a vacancy of a java developer in my organization. So, I won’t prefer to hire a python developer and then make him learn Java. That will be pretty useless and time consuming for both that candidate and for the organization too. Where our system helps the organization to make out the best possible candidates list according to their given constraints and requirement for that particular vacancy. This kind of approach, will help our hiring sector to improve like anything and make it more efficient as the relevant person is getting a relevant job. So, there would be no regrets for both the entities, client company and that hired candidate. Hence satisfaction will be achieved.

#### 2.2.3 IMPLEMENTATION AND METHODOLOGIES

Modular Description of Project Different modules or components created are domain establishment, data collection, parsing, ranking and database component. Parsing and Ranking is the heart of our system which is created using python, nltk, tika libraries. This component does the morphological analysis, syntactic analysis, semantic analysis and generates the parsed and ranked data of the candidate according to his/her skills. Then this information is stored in the database and retrieved and shown to the users whenever required.

**2.2.3.1 DETAILED ANALYSIS AND DESCRIPTION OF PROJECT DOMAIN ESTABLISHMENT**

This module is responsible for creating user accounts and database creation as the proposed system is domain independent and would be used by multiple users. Registration or Login Module: If the new user wants to interact with our system, he needs to simply register into our system by completely filling details i.e. validation. If the user is already existing, he needs to login. Parsing & Ranking: Parsing module is responsible for parsing the document and storing it in Json format which will later be used by the ranking module.

Ranking module will then use the Json file and rank the candidate’s information according to his/herskills and the information will be stored in the database. Morphological Analysis: Morphology in linguistics is the study and description of how words 16 are formed in natural language. In this phase the sentence is broken down into tokens- smallest unit of words, and determine the basic structure of the word.

**2.2.3.2 SYNTACTIC ANALYSIS**

The objective of the syntactic analysis is to find the syntactic structure of the sentence. It is also called Hierarchical analysis/Parsing, used to recognize a sentence, to allocate token groups into grammatical phrases and to assign a syntactic structure to it.

#### 2.2.3.3 SEMANTIC ANALYSIS

Semantic Analysis is related to create the representations presentations for meaning of linguistics inputs. It deals with how to determine the meaning of the sentence from the meaning of its parts.

#### 2.2.4 CONCLUSION

Our system will provide better and efficient solution to current hiring process. This will provide potential candidate to the organization and the candidate will be successfully be placed in an organization which appreciate his/her skillset and ability.

# 2.3 SMART RESUME ANALYZER[3]

# 2.3.1 INTRODUCTION

Any recruiter will find it difficult to choose the best prospects from a vast pool of applicants for that employment vacancy. The chore of manually sorting through thousands of resumes to find the best candidates for the position is incredibly challenging for recruiters. Although the methods employed by job websites have produced some accuracy and precision, one of the main drawbacks is the intricacy of the time component. The time complexity for getting the results is very significant if every candidate resume is compared to every other job posting provided on the online recruitment site. In the last several years, more than 50,000 e- recruitment websites have been created. These online recruitment services creators have employed a variety of strategies to find potential applicants for a specific job profile. Some of them have been successful in using approaches for categorizing resumes of applicants into different groups for each job posting provided by each employer. These methods attempt to match each applicant's resume with the specific job posting. To determine which resumes are closest to the specified job description, top candidates could be sorted using Content-based Recommendation, cosine similarity, and KNN.

**2.3.2 PROPOSED SYSTEM**

Our approach uses, machine learning and natural language processing (NLP) techniques to evaluate resumes contextually. Artificial intelligence is employed along with these tools to go beyond keywords. Following resume screening, the software assesses candidates in real-time based on the employment requirements of the recruiter. This online application seeks to organize the resumes by comparing the resumes that best fits the specified Job Descriptions that is intelligently read as input. This software uses NLP for the instantaneous comparison and ranking of provided resumes. The best candidates for that particular job vacancy can then be selected using the calculated ranks. By moving through all these mechanisms our method provides accurate results with greater efficiency, precision, and accuracy. This is done with the intention of saving recruiters at any organization time and effort from having to read through and evaluate hundreds of resumes.

#### 2.3.2.1 COSINE SIMILARITY

Cosine similarity is a metric for comparing two non-zero vectors in an inner product space. Its value is the same as the inner product of the identical vectors normalized to have the same length, or the cosine of the angle between them. The cosine of 0 degrees is 1, and any angle between (0,] radians) has a cosine that is less than 1.

### 2.3.2.2 NLP:

The JSON file is then sent through the NLP pipeline after the text annotations in the pdf file are converted to JSON format. This NLP pipeline is then used to train the model. Using the NLP framework Spacey, it can be trained. Spacey is a framework that was developed for general data rather than for particular datasets, like a resume. In this method, rather than manually entering every word to create the dataset, semi-supervised learning is utilized to label the significant data in the ZIP file of PDF resumes.

### 2.3.3 CONCLUSION

We investigated the crucial yet understudied issue of automatic resume quality assessment (RQA). The classification of the applicant's resume is a laborious, time-consuming, and resource-wasting process. We have developed a machine learning-based automated algorithm to address this problem by recommending HR the resumes of qualified candidates based on the provided job description.

With the use of Natural Language Processing techniques our algorithm was able to screen and shortlist the most qualified candidates [20].When displaying the top-selected resume on Web user interface, Latent Dirichlet Allocation produced incredibly precise results.

By using job descriptions as input the best-fit resume can be selected by matching the skills in the resume of the applicant. Along with this, various visualization and data analytics are provided on the admin side. The linear classifiers are used to provide a statistical model of skill similarity. The user is provided with a proper guidance about the skills they need to upgrade and the sources for getting acknowledged. They are also equipped with essentials tips that can enhance their resume.

**2.4 RESUME PARSING USING NLP[4]**

**2.4.1 INTRODUCTION**

Daily, corporate firms and recruiting agencies have to process a large number of resumes. Working with a large volume of text data is usually time consuming and stressful. Data gathered from different resumes can be in a various form, including .pdf, .docx, single column resumes, double-column resumes, free formats, and so on. And these formats might not be suitable for the particular application. So, questions may arise in our mind that, what is resume parsing? The process of converting the unstructured form (.pdf/ .docx / .jpeg etc.) of resume data into a structured format is known as resume parsing**.**

Subsequently, converting a resume into prepared text or structured information makes studying, analyzing, and comprehending easier. As a result, many organizations and institutions depend on Information Extraction, where unstructured data and vital information are extracted and converted to make information more readable and organized data forms. The completion of this task takes a long time for humans. So, it is necessary to develop an automated intelligent system that can extract all relevant information to determine whether an applicant is suitable for a particular job profile.

**2.4.2 OBJECTIVE**

**2.4.2.1 ADMIN MANAGEMENT SYSTEM**

This sub-system is one of the most secretive aspects of the project. As it is a carefully guarded part of the project, only the administrator can access using the admin email ID and password.

After the admin logged in, there'll be given the option to upload the resume. Following the submission of the resume, there will have two sub-models: resume parsing and information extraction.

#### 2.4.2.2 FILE UPLOAD AND PARSING SYSTEM

This system is at the center of the overall project. In this sub system the admin will upload the resume and go for further processing. The parser subsystem will use different libraries to transform the submitted unstructured resume to a structured format. This will make it much easier to examine, analyze, and grasp the data.

#### 2.4.2.3 INFORMATION EXTRACTION SYSTEM

After parsing the data, we employ the NLTK, Spacy phrase matchers, regex to extract the essential information. The required extracted data will be saved in JSON format.

Finally, the extracted data or dump JSON file is stored in a MySQL database for further usage if it is needed.

# 2.4.5 CONCLUSION

A normal resume is a compilation of information about a person's work experience, academic background, qualifications, and personal details. These elements might be present in a variety of ways or not at all. It's difficult to keep up with the jargon used in resumes. A resume is made up of corporate names, institutions, degrees, and other information that can be written in a variety of ways. It will take time to review all the resume by an individual.

Machine works faster than human and their accuracy to do any task was also good. Therefore, I have made a system which includes machine learning that extract the important information from resumes within a minute or less than a minute. The hiring individual can use this system for hiring any individual.

**2.5 RESUME PARSER AND SUMMARIZER[5]**

**2.5.1 INTRODUCTION**

A resume parser is a deep learning/AI framework that extracts complete information from resumes, analyses it, stores it, organizes it, and enriches it using taxonomies. Resume parsing software expedites and improves the hiring process. Quick and precise resume parsing technology increases efficiency and provides a better candidate experience.

A resume parser is an interpreter or compiler that converts unstructured data into structured data. It is a component that automatically categorizes information such as contact information, educational qualifications, work experience, skills, achievements, and professional certifications into various fields and parameters to assist you in quickly identifying the most relevant resumes based on your criteria.

Resume parsers have achieved up to 87% accuracy, which refers to data entry accuracy and correctly categorizing data. Because human accuracy is typically less than 96%, the resume parsers achieved "near human accuracy”. To compare data entry accuracy, one executive recruiting firm tested three resume parsers and humans. They ran 1000 resumes through resume parsing software before manually parsing and entering the data.

The company hired a third party to assess how the humans performed in comparison to the software. They discovered that the resume parser results were more comprehensive and contained fewer errors. Humans did not enter all of the information on the resumes and occasionally misspelt words or spelt numbers incorrectly. A resume for an ideal candidate was created based on the job description for a clinical scientist position in a 2012 experiment. Due to the date being listed before the employer, one of the candidate's work experiences was completely lost after going through the parser. Several educational degrees were also missed by the parser. As a result, the candidate received a relevance ranking of 43%. If this had been a real candidate's resume, they would not have advanced to the next stage despite being qualified for the position.

**2.5.2 PROPOSED SYSTEM**

The system will assist recruiters in viewing the summarized resumes and hiring the best candidate. The resumes dataset was gathered from various websites such as Kaggle and GitHub. The dataset is used to train the model. Our website's home page includes a login and sign in option. The login and sign in options will be for two different people, namely the candidate and the recruiting company. The candidate will upload his or her resume, which will be saved in a database. The output screen will display summarized data from the database.

**2.5.3 TOOLS AND TECHNOLOGIES USED**

**PYTHON:** Python is a general-purpose, high-level programming language. Its design philosophy prioritizes code readability by employing significant indentation. Python is garbage-collected and dynamically typed. It supports a wide range of programming paradigms, including structured (especially procedural), object-oriented, and functional programming. Because of its extensive standard library, it is frequently referred to as a "batteries included" language. Guido van Rossum began developing Python as a successor to the ABC programming language in the late 1980s, and it was first released in 1991 as Python 0.9.0. Python 2.0 was released in the year 2000. Python 3.0, released in 2008, was a significant revision that was not fully backward compatible with previous versions. Python 2.7.18, released in 2020, was the final Python 2 release. Python is consistently ranked among the top programming languages.

**MONGODB:** MongoDB is a cross-platform document-oriented database programme that is open source. MongoDB, a NoSQL database programme, employs JSON-like documents with optional schemas. MongoDB was created by MongoDB Inc. and is licensed under the Server-Side Public License (SSPL), which several distributions consider to be non-free. MongoDB is a MACH Alliance member.

**HTML:** HTML, or Hypertext Markup Language, is the standard markup language for documents intended to be displayed in a web browser. It is frequently aided by technologies like Cascading Style Sheets (CSS) and scripting languages like JavaScript. Web browsers receive HTML documents from a web server or local storage and convert them to multimedia web pages. HTML semantically describes the structure of a web page and originally included visual cues.

**CSS:** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

**FLASK:** Flask is a Python-based micro web framework. It is classified as a microframework because it does not necessitate the use of any specific tools or libraries. It lacks a database abstraction layer, form validation, and other components where third-party libraries provide common functions.

**NATURAL LANGUAGE PROCESSING:** Natural language processing (NLP) is an interdisciplinary subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract insights contained in the documents as well as categorize and organize the documents themselves. Challenges in natural language processing frequently involve speech recognition, natural natural-language generation.

**2.5.4 CONCLUSION**

People nowadays value their time and the ease with which they can complete their tasks. A resume analyzer interpreter or compiler that converts unstructured data into structured data.

It is a component that automatically categorizes information such as contact information, educational qualifications, work experience, skills, achievements, and professional certifications into various fields and parameters to assist you in quickly identifying the most relevant resumes based on your criteria.

The resume parser will assist the recruiting company in quickly and easily parsing and summarizing resumes. It will help the recruiting firm parse multiple resumes at the same time. The resume parser will support a variety of document types, including docx, pdf, and html.

# 2.6 A MACHINE LEARNING APPROACH FOR AUTOMATION OF RESUME RECOMMENDATION SYSTEM[6]

# 2.6.1 INTRODUCTION

# Talent acquisition is an important, complex, and time-consuming function within Human Resources (HR). The sheer scale of India’s market is overwhelming [2, 8, 14]. Not only is there a staggering one million people coming into the job market every month, but there is also huge turnover. As per LinkedIn, India has the highest percentage of the workforce that is “actively seeking a new job” [10]. Clearly, this is an extremely liquid, massive market but one that also has many frustrating inefficiencies. The most challenging part is the lack of a standard structure and format for resume which makes short listing of desired profiles for required roles very tedious and time-consuming [11, 24]. Effective screening of resumes requires domain knowledge, to be able to understand the relevance and applicability of a profile for the job role. With a huge number of different job roles existing today along with the typically large number of applications received, short-listing poses a challenge for the human resource department. Which is only further worsened by the lack of diverse skill and domain knowledge within the HR department, required for effective screening. Being able to weed out non-relevant profiles as early as possible in the pipeline results in cost savings, both in terms of time as well as money.

# 2.6.2 METHODOLOGY

# The aim of this work is to find the right candidates resume from the pool of resumes. To achieve this objective, we have developed a machine learning based solution, The complete framework for the proposed model is shown in Figure 2. The proposed model worked in mainly in two steps: i) Prepare and ii) Deploy and Inference. Dataset Description: The data was downloaded from the online portal(s) and from Kaggle. The data is in Excel format, with three column ID, Category, and Resume. ID - The sequence number of the resume, Category - Industry sector to which the resume belongs to, and Resume - The complete CV of the candidate.

# 2.6.2.1 PREPROCESSING

# In this process, the CVs being provided as input would be cleansed to remove special or any junk characters that are there in the CVs. In cleaning, all special characters, the numbers, and the single letter words are removed. We got the clean dataset after these steps having no special characters, numbers or single letter word. The dataset is split into the tokens using the NLTK tokenizes [12]. Further, the preprocessing steps are applied on tokenized dataset such as stop word removal, stemming, and lemmatization. The raw CV file was imported and the data in the resume field was cleansed to remove the numbers and the extra spaces in the date. Data Masking was done as:

# • Mask string fragments like \x • Mask string fragments for escape sequences like \a, \b, \t, \n.

# • Mask all numbers

# • Replace all the single letter words with an empty string • Mask email addresses.

# • Stop words were masked from the dataset.

# • Lemmatization.

# 2.6.2.2 STOP WORDS REMOVAL

# The stop words such as and, the, was, etc. are frequently appeared in the text and not helpful for prediction process, hence it is removed. Steps to filter the Stop Words:

# 1. We have tokenize the input words into individual tokens and stored it in an array

# 2. Now, each words matches with the list of Stop Words present in NLTK library:

# (a) from nltk.corpus import stopwords /\*Imported Stop Word module from NLTK corpus\*/ .

# (b) StopWords[]= set(stopwords.words(’english’)) /\* Get set of English Stop Words\*/ .

# (c) It returns total of 179 stop words, that can be verified using (len(StopWords)) and can be viewed by print (StopWords) function.

# 3. If the words present in the list of StopWords[], filtered from the main sentence array.

# 4. The same process repeated until the last element of the tokenized array is not matched.

# 5. Resultant array does not have any stop words.

# 2.6.2.3 STEMMING

# Stemming is the method of decreasing word inflection to its root forms such as mapping a group of words to the same stem even though the stem itself is not a valid term in the language. Stem (root) is the part of the word to which you add inflectional (changing/deriving) affixes such as (-ed, -ize, -s, -de, -ing, mis). For example the words like: Playing, Plays, Played are mapped to their root word.

# 2.6.2.4 LEMMATIZATION

Unlike Stemming, lemmatization decreases the inflected phrases to ensure that the root word belongs to the language correctly. Lemmatization comprises the following routine steps:

• Transform the corpus of text into a list of words.

• Create a concordance of the corpus, i.e., of all the items of the word list as they occur in the corpus.

• Assign the word-forms to their lemmas based on the concordance.

**2.6.3 CONCLUSION**

The proposed model worked in two phases: first, classify the resume into different categories. Second, recommends resume based on the similarity index with the given job description. The proposed approach effectively captures the resume insights, their semantics and yielded an accuracy of 78.53% with Linear SVM classifier. The performance of the model may enhance by utilizing the deep learning models like: Convolutional Neural Network, Recurrent Neural Network, or Long-Short Term Memory and others. If an Industry provides a large number of resume, then Industry specific model can be developed by utilizing the proposed approach. By involving the domain experts like HR professional would help to build a more accurate model, feedback of the HR professional helps to improve the model iteratively.

**2.7 RESUME ANALYSER AND SUGGESTION MAKING USING NLP [7]**

**2.7.1 INTRODUCTION**

Every year, a large number of engineers graduate, but only a small percentage are hired by top multinational corporations, and the others must still be located. The requirement for more information about creating a CV appropriate with a specific role is one of the key causes of this problem. Approximately 70 to 80 percent of students who apply to prestigious companies are turned down during the initial screening process. Additionally, several studies indicate that this is due to a lack of knowledge and skills required for that particular profession. The major goal is to enhance their resumes by adding the necessary qualifications. The Resume Analyzer system looks over the resume and extracts the pertinent data, including name, contact info, experience, and credentials. Our project demonstrates a system that extracts specific information from a resume using tokenization in Natural Language Processing methods, such as education, experience, and skills. Then, using a classification technique such as k-nearest neighbor, we will determine the best role for the user. Parsing the resume makes improving resume strength easier and more efficient.

**2.7.2 METHODOLOGY**

**2.7.2.1 UNDERSTANDING NLP**

NLP is a computer science and artificial intelligence subfield that analyses computer-human (natural) language interactions. It is used when trying to apply machine learning algorithms to text and speech. NLP may be utilized to create systems like voice recognition, file description, language processing, spam filtering, character recognition, autosuggest, predictive typing, and many others. The majority of us now have voice-activated mobile phones. NLP is used by these smartphones to comprehend what is said. Moreover, many people use personal computers that include voice recognition software. SOME EXAMPLES: 1. Siri 2. Cortona

**2.7.2.2 THE BASICS FOR NLP IN TEXT**

**Sentence tokenization**

Sentence tokenization refers to the method of dividing a string of writing system into individual sentences. The concept appears to be quite simple. When we come across a correct punctuation in English or another language, we can separate phrases. Even so, due to the use of the proper character for acronyms, this issue is not trivial, even in English. When processing plain text, tables of abbreviations with periods could indeed help us avoid incorrect sentence boundary assignment. Don't fear about the specifics for the time being because we commonly use libraries to assist us with this.

**Word tokenization**

The process of breaking off the string of formal text into its individual words is identified as word tokenization. Space is a rough approximation of a word divider in English and several other language groups which use a few Latin alphabets. However, if we simply start dividing by area to achieve the desired results, we may encounter difficulties. Some factors. these factors in English are better worded and may include a space. We shouldn't get too caught up in the details because, in most cases, we'll be using a library to accomplish what we want.

**Text lemmatization and stemming**

Documents may contain different linguistic forms for linguistic reasons. There are also duplicates with comparable content. Stemming and lemmatization are both techniques for reducing a word's derivational forms to a single root word.

**Stop words**

Stop - words are words that are deleted from text prior to or following processing. When using machine learning on text, these phrases can add a lot of noise. This is why we desire to get rid of these superfluous words. Stop phrases There isn't a single universal list of stop words, these are generally the most generic terms in a language. Depending on your application, the list of stop words may differ.

3.2.5 Regex A regular expression is a sequence of characters that describes a search pattern. It is also recognised as a regex or regexp.

**2.7.3 CONCLUSION**

Finally, the Resume Analyzer and Suggestion Making project is an effective tool that analyses a given resume and provides suggestions for improvement using NLP techniques and machine learning algorithms. The system can predict which job role the resume is most suitable for by analyzing its content and structure, and it can provide recommendations to make the resume more robust and appealing to potential employers. The project's use of natural language processing (NLP) tools and techniques, combined with machine learning algorithms, makes it an efficient and dependable tool for job seekers to use when applying for a new position. Job seekers can increase their chances of being hired by tailoring their resumes to the specific job they are applying for, thanks to the system's ability to analyse resumes and provide recommendations for improvement. Overall, the Resume Analyzer and Suggestion Making project is a valuable resource that can assist job seekers in increasing their chances of being hired. With its ability to analyse resumes and make recommendations, the project can be a valuable tool in the toolbox of any job seeker.

**2.8 Automated Resume Screening Using Natural Language Processing[8]**

**2.8.1 INTRODUCTION**

An essential step in the hiring process is the automatic review of resumes, which entails assessing job applications to find the applicant most suited for a given position. This procedure may take a long time and be prone to human mistake, which could lead to the loss of qualified individuals. Automated resume screening has grown in popularity recently as a solution to this problem. Automatic resume screening uses several methods to enhance accuracy and efficiency, including deep learning algorithms, machine learning, and natural language processing (NLP). Several studies have suggested various methods for automating the screening of resumes. Li et al. (2020) introduced a hybrid deep learning framework that makes use of long short-term memory (LSTM) networks and convolutional neural networks (CNNs) [6].

**2.8.2 PROPOSED SYSTEM**

Proposed System Improved precision: NLP algorithms such as SBERT and cosine similarity excel at identifying resumes that are most relevant to a specific job description. These algorithms are designed to comprehend the context of the text and decipher the intended meanings of the words. An improved effectiveness: NLP algorithms can evaluate hundreds or thousands of resumes in a matter of minutes, making them far faster than hand screening. Recruiters will save a lot of time and money as a result of this [3]. NLP algorithms such as SBERT and cosine similarity can be tailored to specific businesses, positions, or organization’s resulting in more accurate resume assessments. More accurate candidate matching: The algorithms S-BERT and cosine similarity are created to match candidates with job descriptions based on the relevance and similarity of their abilities, experience, and qualifications. Language autonomy: Employing managers will find it easier to evaluate resumes from candidates with different linguistic backgrounds thanks to NLP algorithms' ability to interpret resumes written in a range of languages. Working knowledge of unstructured data: NLP algorithms can pull relevant data from unstructured data, like resumes, making it easier for recruiters to evaluate resumes that do not follow a standard pattern.

**2.8.3** **CONCLUSION**

we'll say that applying NLP algorithms for resume screening—like SBERT and cosine similarity—offers several benefits over more traditional methods and they can handle unstructured data, such as resumes written in many languages. They can also minimize prejudice among people and enhance candidate matching, improving recruiting processes. It is critical to remember that these algorithms have limitations and are not optimal in all circumstances [11]. So, it is crucial to use these algorithms as a part of a larger hiring strategy that also includes human judgement and arbitrary criteria. The use of NLP algorithms in recruiting, such as SBERT and cosine similarity, is a promising development that has the potential to fundamentally alter how businesses screen and select job candidates.

**2.9 INTELLIGENT RESUME ANALYZER[9]**

**2.9.1 INTRODUCTION**

In today's highly competitive job market, job seekers face the daunting task of standing out among hundreds, if not thousands, of other applicants. One of the most critical components of a successful job search is a well-crafted resume. However, creating a resume that accurately represents one's skills, experience, and qualifications can be a challenging task. Additionally, resumes that fail to pass the screening process of job interviews often result in job seekers being eliminated from consideration, regardless of their actual qualifications. To address this challenge, an innovative solution has been developed – the Intelligent Resume Analyzer. This system uses artificial intelligence (AI) and machine learning algorithms to analyze resumes and provide feedback on how to improve them. The Intelligent Resume Analyzer is designed to help job seekers create resumes that are more likely to pass the screening process of job interviews. The Intelligent Resume Analyzer works by analyzing various aspects of a candidate's resume, such as their experience, education, skills, and achievements. The system uses natural language processing, text mining, and sentiment analysis techniques to extract meaningful insights from the content. It also checks the formatting, grammar, and spelling of the resume to ensure that it meets industry standards. The feedback provided by the Intelligent Resume Analyzer can help job seekers identify areas where they need to improve their resume. For example, the system may suggest that a candidate rephrase their work experience in a more concise and impactful manner. Alternatively, it may recommend that a candidate highlight their relevant skills more prominently. These suggestions can help job seekers create resumes that accurately represent their qualifications and increase their chances of getting hired. Overall, the Intelligent Resume Analyzer is a powerful tool that can help job seekers create resumes that stand out in today's highly competitive job market. By leveraging the power of AI and machine learning, the system provides personalized feedback that can help job seekers improve their resumes and increase their chances of success.

**2.9.2 METHODOLOGY**

The methodology section of the Intelligent Resume Analyzer system involves several stages in the development and implementation of the system. The purpose of this section is to provide a clear understanding of the research design, data collection, and analysis procedures used in the development of the Intelligent Resume Analyzer system. Research Design The research design for the Intelligent Resume Analyzer system was based on the principles of machine learning and natural language processing. The research design involved collecting and analyzing a large corpus of resumes to identify patterns and features that are important for screening resumes. The system was developed using a supervised learning approach where annotated resumes were used to train the system to recognize specific patterns. Data Collection The data collection for the Intelligent Resume Analyzer system involved collecting resumes from different sources such as job portals, social media platforms, and other online sources. The resumes were collected in different formats such as PDF, MS Word, and plain text files. The collected resumes were pre-processed to remove irrelevant information such as personal identification data. Data Analysis The data analysis for the Intelligent Resume Analyzer system involved several stages. Firstly, the pre-processed resumes were analyzed using natural language processing techniques to extract relevant features such as skills, education, experience, and achievements. Secondly, machine learning algorithms were used to analyze the extracted features and identify patterns that could be used to predict the suitability of a resume for a particular job. Finally, the system was tested on a large corpus of resumes to evaluate its performance and identify areas for improvement. Evaluation To evaluate the performance of the Intelligent Resume Analyzer system, a set of annotated resumes were used as a test dataset. The annotated resumes were labeled according to their suitability for different job positions. The system was then evaluated based on its ability to accurately predict the suitability of a resume for a given job position. The evaluation results showed that the system achieved high accuracy in predicting the suitability of resumes for different job positions. In conclusion, the methodology section of the Intelligent Resume Analyzer system involved a rigorous research design, data collection, and analysis procedures. The development of the system was based on machine learning and natural language processing techniques that enabled the system to analyze resumes and provide feedback on how to improve them. The evaluation of the system showed that it achieved high accuracy in predicting the suitability of resumes for different job positions.

**2.9.3 CONCLUSION**

All In conclusion, the Intelligent Resume Analyzer system is a powerful tool that analyses resumes and makes suggestions for improvement using artificial intelligence and machine learning algorithms. The technique is intended to assist job searchers in creating resumes that are more likely to pass the job interview screening process. The system gives individualized feedback to assist job searchers improve their resumes and raise their chances of success by utilizing the power of AI and natural language processing techniques. The Intelligent Resume Analyzer system’s methodology section discussed the study design, data gathering, and analytic techniques utilized in the system’s creation. The system was created using a supervised learning method, with annotated resumes used to teach the system to recognize specific patterns. The system was assessed based on its ability to forecast the suitability of a resume for a certain job vacancy. Overall, the Intelligent Resume Analyzer technology has the potential to transform how job searchers produce resumes and apply for positions. The technology can help job searchers generate resumes that appropriately represent their qualifications and boost their chances of being hired by delivering individualized feedback. As the job market grows more competitive, the Intelligent Resume Analyzer technology can give job seekers with a significant competitive advantage.

2.10 Personality Evaluation and CV Analysis using Machine Learning Algorithm[10]

# 2.10.1 INTRODUCTION

As far as employment is considered, selecting the right candidate for the recruitment process from a vast pool of candidates has been a fundamental issue [1]. Conducting personality and various technical eligibility evaluation tests, interviews, and group discussions have been traditional techniques. Due to inception of social media, much more important information about employees is exposed to their online handles [2]. Generally, such information is unnoticed by the recruiters. Aptitude test followed by the interview is traditional practices for the recruitment process. These traditional practices are very much time-consuming, and may result in unfair choices of candidate. As compared to traditional recruitment process, if an online selection process is conducted, then a fair selection of the candidate is possible.

Personality is the most important factor which reflects an individual, which keeps on varying [3]. Tackling them is a tedious task for which we have implemented an approach to identify the personality and also provide with the recommendation [4].

# 2.10.2 PROPOSED SYSTEM

# In our paper, we propose personality evaluation and CV analysis using machine learning algorithm. This system provides with an expert workforce for the organization which will help the HR department to select the right candidate for the particular job profile. In our society intelligence is highly appreciated. If you have a high IQ, you have a better chance of being successful at school and professional life. Generally, for prediction of personality, psychometric questions are used. The proposed system is developed as a web application wherein the admin is first needed to login with proper credentials followed by which they can add the questions and can also modify them. For each question, four options along with the correct answer is stored in the database. The candidate will register her/himself with all the details and will also fill their own CV details into the system. Sample aptitude test questions along with their options are shown in (Table 1). After the test given by the candidates, the scores are stored in databases. The next test is of personality test. There is a common myth which says that IQ tests measure intelligence. What an IQ test actually measures are not actual intelligence, but a person's capacity for intelligence. In this test various situations will be encountered by the candidate ranging from strongly agree to disagree, which is provided as a drop-down list. The factors range like openness to experience, conscientiousness. (Table 2) shows the sample questions for personality test. Each question has the fix set of choices varying from strongly agree to disagree.

# 2.10.3 Conclusion and Future Scope

We have presented in this paper, the prediction of human personality by using standard questionnaires that is provided by the HR Department according to the job selection criteria. Candidates fill an online Curriculum Vitae (CV) which can be later on viewed by the Admin. Candidates are provided with separate set keys for attempting the aptitude and personality-based tests. CV analysis is performed using the CV filled by the candidate in the website . A machine learning approach has been used in analysis of data through content and collaborative filtering. Further the test scores help in deciding the qualities in the candidates. Thus, the CV is shortlisted for the recruitment process and a fair and appropriate decision is made by HR department. Also data visualization model determines the overall performance of the students based on various factors. This analysis helps the Admin department to calculate the proficiency of candidates accurately.

# CHAPTER 3

# PROPOSED WORK

#### 3.1 DATA COLLECTION

* + 1. **SOURCES OF RESUMES**

Resumes were sourced from diverse platforms, including popular job portals, professional networking sites, and open datasets. The inclusion of resumes from various sources aimed to create a diverse dataset representative of different industries and career levels.

#### DATA PREPROCESSING

Data preprocessing involved several steps to ensure uniformity and standardization across different resume formats. Techniques such as text cleaning, removal of irrelevant information, and standardization of formatting were applied. Special attention was given to handling diverse document types, including PDF, DOC, and plain text.

#### 3.2 TECHNOLOGY STACK

**3.2.1 PROGRAMMING LANGUAGES**

Python was selected as the primary programming language due to its extensive libraries for natural language processing (NLP) and machine learning (ML). Additional languages, such as JavaScript, were used for web interface development.

#### FRAMEWORKS AND LIBRARIES

spaCy: Leveraged spaCy for advanced NLP tasks, including named entity recognition (NER) and part-of-speech tagging.

Scikit-learn: Utilized Scikit-learn for machine learning tasks, such as sentiment analysis.

* + 1. **TOOLS**

Docker: Employed Docker for containerization, ensuring consistent performance across different environments.pdf2text, python-docx: Used external libraries for extracting text from PDF and DOC file formats.

#### 3.3 SYSTEM ARCHITECTURE

**3.3.1 OVERVIEW OF THE RESUME ANALYZER**

The Resume Analyzer is designed as a modular system with three main components: data preprocessing, NLP analysis, and the user interface.

Each component performs a specific set of tasks, contributing to the overall analysis process.

#### FLOWCHART OF THE ANALYSIS PROCESS

The analysis process initiates with the upload of a resume through the user interface, whereupon it undergoes meticulous data preprocessing procedures. These preprocessing steps are imperative for standardizing formatting and extracting textual content accurately. Following preprocessing, the refined data proceeds to the NLP analysis module, where an array of techniques, including Named Entity Recognition (NER) and sentiment analysis, are employed to derive comprehensive insights.

Moreover, the NLP analysis module is bolstered by advanced algorithms that delve deep into the semantic nuances of the textual data. These algorithms leverage state-of-the-art methodologies to discern intricate patterns, sentiments, and key entities embedded within the resume content. Furthermore, the analysis process prioritizes the extraction of crucial information pertinent to the user's objectives, thus ensuring relevance and utility in the generated insights.

As the analysis reaches its culmination, the final results are meticulously curated and presented to the user through an intuitive interface. This interface is thoughtfully designed to provide users with seamless navigation and easy comprehension of the extracted insights. Additionally, the interface incorporates interactive elements and visual aids to enhance user engagement and facilitate informed decision-making.

# SOFTWARE DESIGN

#### 3.4.1 BLOCK DIAGRAM & EXPLANATION

**3.4.1.1 UPLOAD RESUME AND ROLE SELECTION**

* + - Design the system to support various file formats commonly used for resumes, such as PDF, DOC, DOCX, or TXT.
    - Implement validation checks to ensure that users only upload files in supported formats.
    - Include a progress indicator to show users the status of their upload. This is especially important for larger files that may take some time to process.
    - Display clear error messages to guide users on how to address any issues with their uploaded resumes.
    - Offer a list of predefined job roles or industries that users can choose from. This list should cover a wide range of professions and industries.
    - Implement a search functionality to allow users to quickly find specific roles of interest.
    - Include filters or categories to help users narrow down their choices based on industry, experience level, or other relevant criteria.

#### 3.4.1.2 PARSING AND PROCESSING OF THE RESUME

* + - Extract the text content from the uploaded resume file. This is a crucial step as it provides the raw data for further processing.
    - Identify and understand the structure of the resume document, including sections like personal information, education.
    - Implement algorithms for recognizing entities such as names, addresses, phone numbers, email addresses, educational institutions, job titles, and dates. This helps in categorizing and structuring the information., work experience, skills, and additional details.
    - Analyze the entire resume for keywords relevant to specific industries or job roles. This helps in assessing the overall suitability of the candidate for particular positions.

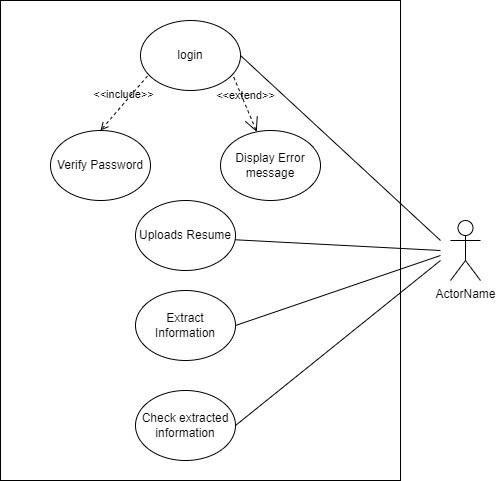
#### 3.4.1.3 LIST ALL THE SKILL AND DETAILS OF THE RESUME WITH A SCORE

* + - Identify and extract skills mentioned in the resume using natural language processing (NLP) techniques.
    - Utilize a predefined list of skills or dynamically generate a list based on common industry keywords.
    - Assign a relevance score to each extracted skill. This score can be based on factors such as the frequency of the skill in the resume, the context in which it appears, or its importance for the selected job roles.
    - For example, a skill mentioned in the context of work experience might be considered more relevant than one listed in the skills section.
    - Calculate an overall score for the resume by aggregating the relevance scores of skills and details. This can be a weighted sum, where certain elements (e.g., work experience) contribute more to the overall score.
    - Display the list of skills and details along with their corresponding relevance scores in a clear and user-friendly format. This can be in the form of a table, chart, or summary report.

#### 3.4.1.4 WHAT IS LACKING IN THE RESUME, WHICH SKILL TO WORK ON

* + - Compare the skills mentioned in the resume with the desired skills for the selected job roles.
    - Identify skills that are commonly required for the chosen roles but are not present or are underrepresented in the resume.
    - Offer suggestions on how the user can acquire or enhance the identified missing skills.
    - Provide links to relevant online courses, certifications, workshops, or other learning resources that can help bridge the skill gaps.
    - Offer context for each suggested skill by explaining why it is essential for the selected roles. This helps users understand the relevance and importance of the recommended skills.
    - Break down skill development into achievable goals and milestones. This helps users create a realistic plan for improving their resume over time.

#### 3.4.1.5 ANY CERTIFICATION OR COURSE WHICH CAN BE DONE

* + - Based on the identified skill gaps, suggest specific certifications or courses that can help the user acquire those skills.
    - Offer details about the suggested certifications, including the learning platform, duration, and any prerequisites.
    - Include information about the industry recognition and value of each certification.
    - Integrate with online learning platforms to provide direct links to the recommended certifications or courses.
    - Streamline the process for users to enroll in the suggested programs.
    - Prioritize certifications that are widely recognized in the industry or relevant to the selected job roles and provide context on why each certification is beneficial for the user's career goals.
  1. **USE-CASE DIAGRAM**

**CHAPTER 4**

**MODEL**

**4.1** **KNN**

**4.1.1 FEATURE EXTRACTION**

Each resume is converted into a feature vector. These features could include educational background, work experience, skills, certifications, etc. Each feature is given a numerical value or converted into a format suitable for comparison.

**4.1.2. TRAINING PHASE**

During the training phase, the algorithm learns from a set of labeled resumes. These resumes are already classified into categories like "relevant" and "not relevant" or "suitable for a particular job role" and "not suitable."

**4.1.3. KNN CLASSIFICATION**

The KNN algorithm then selects the K nearest neighbors (resumes) based on the calculated distances. The value of K is predefined. These neighbors are the resumes most similar to the one being analyzed.

**4.1.4. VOTING** **OR WEIGHTED VOTING**

The algorithm assigns a class label to the new resume based on the classes of its nearest neighbors. This could be a simple majority vote if K=1, or a weighted vote where closer neighbors have more influence.

**4.1.5. CLASSIFICATION OUTCOME**

Based on the outcome of the voting process, the resume is classified into a category such as "relevant" or "not relevant" for the job or any other predefined categories.

**4.2 ONE-VS-REST" (OVR)**

**4.2.1. PROBLEM DEFINITION**

In a resume analyzer, the task may involve categorizing resumes into multiple classes such as "Engineering," "Marketing," "Finance," etc. This is a multi-class classification problem where each resume can belong to one and only one class.

**4.2.2. BINARY CLASSIFICATION APPROACH**

With the OvR strategy, you treat each class as a separate binary classification problem. For example, you could have a classifier to distinguish "Engineering" resumes from all other resumes, another classifier for "Marketing" resumes, and so on.

**4.2.3. TRAINING PHASE**

For each class, you train a binary classifier using a subset of the data. In the case of "Engineering" resumes, you use all the "Engineering" resumes as positive examples and all other resumes as negative examples.

**4.2.4. PREDICTION PHASE**

When a new resume is submitted for analysis, you apply each binary classifier to it. Each classifier gives a score or probability indicating the likelihood of the resume belonging to the class it's trained for.

**4.2.5. DECISION MAKING**

The class with the highest score or probability from among all the classifiers is then assigned to the resume. For example, if the "Engineering" classifier gives the highest score for a particular resume, it is classified as an "Engineering" resume.

**4.3: TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)**

**4.3.1. TERM FREQUENCY (TF)**

This measures how frequently a term occurs in a document. It is calculated by counting the number of times a term appears in a document and then dividing it by the total number of terms in the document. This helps in capturing the importance of a term within a specific document.

TF(t,d) = Total number of terms in document d/Number of times term t appears in document d

**4.3.2. INVERSE DOCUMENT FREQUENCY (IDF)**

This measures how important a term is across the entire corpus of documents. It is calculated by taking the logarithm of the ratio of the total number of documents to the number of documents containing the term, then adding 1 to avoid division by zero for terms that do not appear in the corpus.

**4.3.3. TF-IDF SCORE**

Finally, the TF-IDF score for a term in a document is computed by multiplying its TF by its IDF.

# CHAPTER 5

# TRAINING THE MODEL

**5.1 TRAINING THE DATASET**

import numpy as np *# For linear algebra operations*

import pandas as pd *# For data processing and reading CSV files*

import matplotlib.pyplot as plt *# For plotting graphs*

import seaborn as sns *# For advanced data visualization*

import re *# For regular expressions operations*

from nltk.corpus import stopwords *# For removing stopwords from text data*

from sklearn.preprocessing import LabelEncoder *# For encoding labels*

from sklearn.model\_selection import train\_test\_split *# For splitting the dataset*

from sklearn.feature\_extraction.text import TfidfVectorizer *# For text vectorization*

from sklearn.neighbors import KNeighborsClassifier *# For the KNN classifier*

from sklearn.metrics import accuracy\_score *# For evaluating model accuracy*

df = pd.read\_csv(‘ResumeDataSet.csv’)

df.head()

**5.2 Breif Description of Library Used**

**5.2.1 NumPy (np):**

NumPy is a Python library for numerical computations. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

**5.2.2 Pandas (pd):**

Pandas is a Python library for data manipulation and analysis. It offers data structures like DataFrame and Series, which are ideal for working with structured and tabular data.

**5.2.3 Matplotlib.pyplot (plt):**

Matplotlib is a comprehensive library for creating static, interactive, and animated visualizations in Python. pyplot is a submodule of Matplotlib that provides a MATLAB-like plotting interface.

**5.2.4 Seaborn (sns):**

Seaborn is a statistical data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

**5.2.5 re:**

The re module provides support for regular expressions in Python. Regular expressions are used for pattern matching and manipulation of text data.

**5.2.6 nltk.corpus.stopwords:**

NLTK (Natural Language Toolkit) is a Python library for working with human language data. The stopwords corpus from NLTK contains a list of common stopwords, which are often removed from text data during preprocessing.

**5.2.7 sklearn.preprocessing.LabelEncoder:**

LabelEncoder is a class from scikit-learn used for encoding categorical labels into numerical values. It's commonly used when working with classification tasks and algorithms that require numerical input.

**5.2.8 sklearn.model\_selection.train\_test\_split:**

train\_test\_split is a function from scikit-learn used to split a dataset into training and testing sets. It's essential for evaluating the performance of machine learning models by training them on one subset of data and testing them on another.

**5.2.9 sklearn.feature\_extraction.text.TfidfVectorizer:**

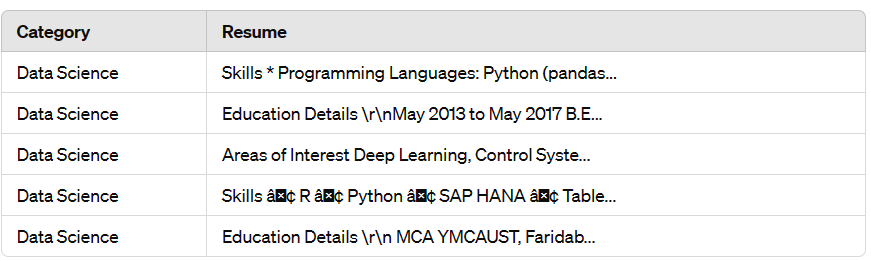
TfidfVectorizer is a class from scikit-learn used for converting textual data into a matrix of TF-IDF features. TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents.

**5.2.11 sklearn.neighbors.KNeighborsClassifier:**

KNeighborsClassifier is a class from scikit-learn implementing the k-nearest neighbors algorithm for classification. It predicts the class of a data point based on the majority class among its k nearest neighbors.

**5.2.12 sklearn.metrics.accuracy\_score:**

accuracy\_score is a function from scikit-learn used to evaluate the accuracy of a classification model. It compares the predicted labels of a model with the true labels and returns the fraction of correctly classified samples.



**(a)**

df["Category"].value\_counts()

|  |
| --- |
| Java Developer 84 |
| Testing 70 |
| DevOps Engineer 55 |
| Python Developer 48 |
| Web Designing 45 |
| HR 44 |
| Hadoop 42 |
| Blockchain 40 |
| ETL Developer 40 |
| Operations Manager 40 |
| Data Science 40 |
| Sales 40 |
| Mechanical Engineer 40 |
| Arts 36 |
| Database 33 |
| Electrical Engineering 30 |
| Health and fitness 30 |
| PMO 30 |
| Business Analyst 28 |
| DotNet Developer 28 |
| Automation Testing 26 |
| Network Security Engineer 25 |
| SAP Developer 24 |
| Civil Engineer 24 |
| Advocate 20 |
| Name: Category, dtype: int 64 |

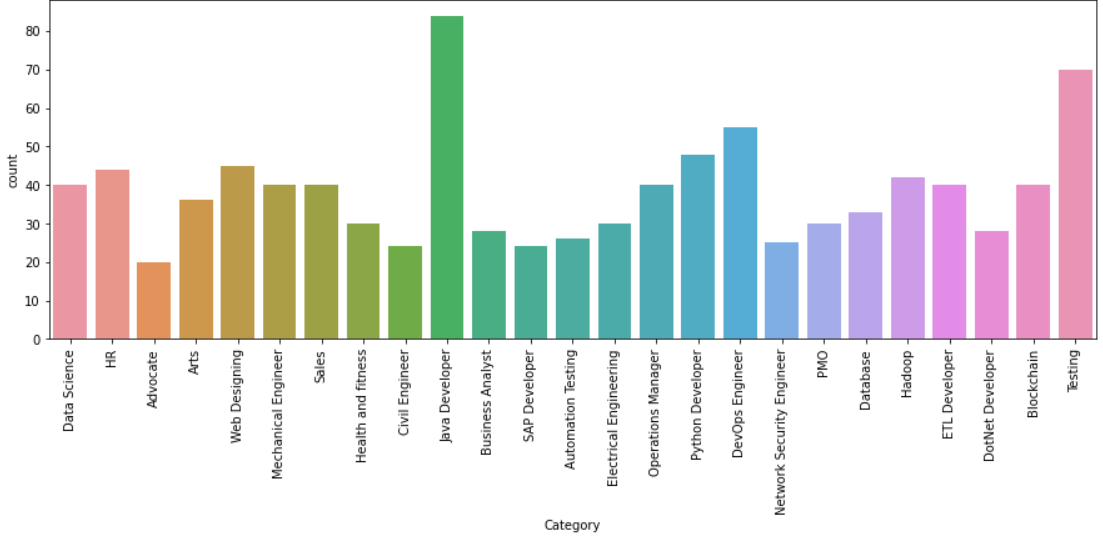
**(b)**

plt.figure(figsize=(15,5))

sns.countplot(df['Category'])

plt.xticks(rotation=90)

plt.show()

****

**(c)**

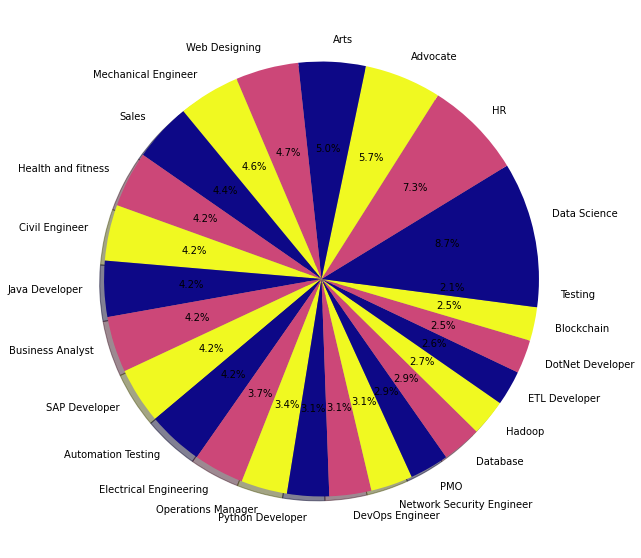
counts = df["Category"].value\_counts()

labels = df["Category"].unique()

plt.figure(figsize=(15,10))

plt.pie(counts, labels=labels, autopct="**%1.1f%%**", shadow – true, colors=plt.cm.plasma(np.linspace(0,1,3)))

plt.show()



**(d)**

# 5.3 EXPLORING RESUME

Df[‘Category’][0]

**Output: ‘Data Science’**

df[‘Resume’][0]

['Skills \* Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. \* Machine learning: Regression, SVM, NaÃ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language processing, Dimensionality reduction, Topic Modelling (LDA, NMF), PCA & Neural Nets. \* Database Visualizations: Mysql, SqlServer, Cassandra, Hbase, ElasticSearch D3.js, DC.js, Plotly, kibana, matplotlib, ggplot, Tableau. \* Others: Regular Expression, HTML, CSS, Angular 6, Logstash, Kafka, Python Flask, Git, Docker, computer vision - Open CV and understanding of Deep learning.Education Details \r\n\r\nData Science Assurance Associate \r\n\r\nData Science Assurance Associate - Ernst & Young LLP\r\nSkill Details \r\nJAVASCRIPT- Exprience - 24 months\r\njQuery- Exprience - 24 months\r\nPython- Exprience - 24 monthsCompany Details \r\ncompany - Ernst & Young LLP\r\ndescription - Fraud Investigations and Dispute Services Assurance\r\nTECHNOLOGY ASSISTED REVIEW\r\nTAR (Technology Assisted Review) assists in accelerating the review process and run analytics and generate reports.\r\n\* Core member of a team helped in developing automated review platform tool from scratch for assisting E discovery domain, this tool implements predictive coding and topic modelling by automating reviews, resulting in reduced labor costs and time spent during the lawyers review.\r\n\* Understand the end to end flow of the solution, doing research and development for classification models, predictive analysis and mining of the information present in text data. Worked on analyzing the outputs and precision monitoring for the entire tool.\r\n\* TAR assists in predictive coding, topic modelling from the evidence by following EY standards. Developed the classifier models in order to identify "red flags" and fraud-related issues.\r\n\r\nTools & Technologies: Python, scikit-learn, tfidf, word2vec, doc2vec, cosine similarity, NaÃ¯ve Bayes, LDA, NMF for topic modelling, Vader and text blob for sentiment analysis. Matplot lib, Tableau dashboard for reporting.\r\n\r\nMULTIPLE DATA SCIENCE AND ANALYTIC PROJECTS (USA CLIENTS)\r\nTEXT ANALYTICS - MOTOR VEHICLE CUSTOMER REVIEW DATA \* Received customer feedback survey data for past one year. Performed sentiment (Positive, Negative & Neutral) and time series analysis on customer comments across all 4 categories.\r\n\* Created heat map of terms by survey category based on frequency of words \* Extracted Positive and Negative words across all the Survey categories and plotted Word cloud.\r\n\* Created customized tableau dashboards for effective reporting and visualizations.\r\nCHATBOT \* Developed a user friendly chatbot for one of our Products which handle simple questions about hours of operation, reservation options and so on.\r\n\* This chat bot serves entire product related questions. Giving overview of tool via QA platform and also give recommendation responses so that user question to build chain of relevant answer.\r\n\* This too has intelligence to build the pipeline of questions as per user requirement and asks the relevant /recommended questions.\r\n\r\nTools & Technologies: Python, Natural language processing, NLTK, spacy, topic modelling, Sentiment analysis, Word Embedding, scikit-learn, JavaScript/JQuery, SqlServer\r\n\r\nINFORMATION GOVERNANCE\r\nOrganizations to make informed decisions about all of the information they store. The integrated Information Governance portfolio synthesizes intelligence across unstructured data sources and facilitates action to ensure organizations are best positioned to counter information risk.\r\n\* Scan data from multiple sources of formats and parse different file formats, extract Meta data information, push results for indexing elastic search and created customized, interactive dashboards using kibana.\r\n\* Preforming ROT Analysis on the data which give information of data which helps identify content that is either Redundant, Outdated, or Trivial.\r\n\* Preforming full-text search analysis on elastic search with predefined methods which can tag as (PII) personally identifiable information (social security numbers, addresses, names, etc.) which frequently targeted during cyber-attacks.\r\nTools & Technologies: Python, Flask, Elastic Search, Kibana\r\n\r\nFRAUD ANALYTIC PLATFORM\r\nFraud Analytics and investigative platform to review all red flag cases.\r\nâ\x80¢ FAP is a Fraud Analytics and investigative platform with inbuilt case manager and suite of Analytics for various ERP systems.\r\n\* It can be used by clients to interrogate their Accounting systems for identifying the anomalies which can be indicators of fraud by running advanced analytics\r\nTools & Technologies: HTML, JavaScript, SqlServer, JQuery, CSS, Bootstrap, Node.js, D3.js, DC.js']

**(e)**

**5.4 WORDS INTO CATEGORICAL VALUES**

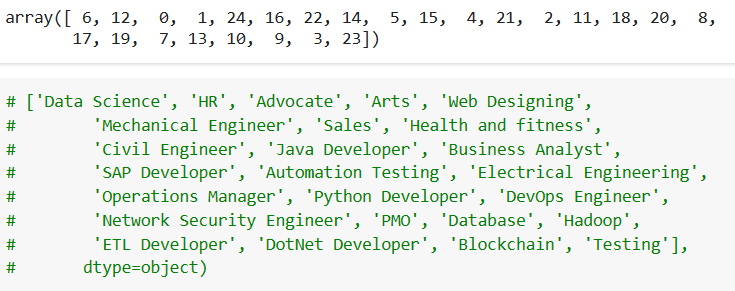
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

le.fit(df['Category'])

df['Category'] = le.transform(df['Category'])

df.Category.unique()

****

**(f)**

**Fig. 1. Structured and graphical Representation of Resume Data:**

(a) The table provides a structured representation of resume data, facilitating analysis and processing of resumes for various purposes, such as recruitment, talent management, or data analysis.

(b) This bar chart illustrates the distribution of resume categories in the dataset. Each category represents a specific field or domain, ranging from Java Developer and Testing to Advocacy and Civil Engineering. The vertical axis displays the count of resumes belonging to each category, while the horizontal axis lists the categories themselves. Java Developer has the highest count, with 84 resumes, followed closely by Testing with 70 resumes, and DevOps Engineer with 55 resumes. Other categories such as Python Developer, Web Designing, HR, and Hadoop also exhibit notable counts, demonstrating the diversity of skills and experiences within the dataset.

This visualization provides valuable insights into the composition of the resume dataset, highlighting the prominence of certain fields and the distribution of talent across various domains. It serves as a useful reference for recruiters, talent managers, and analysts seeking to understand the landscape of available talent within specific industries and professions.

(c) Displays the distribution of resume categories in your dataset. Each bar represents a specific category or field, and the height of the bar corresponds to the count of resumes belonging to that category.

For example, if the 'Java Developer' category has a bar that is taller than the others, it means that there are more resumes in the dataset categorized as 'Java Developer' compared to other categories. Similarly, shorter bars represent categories with fewer resumes. displays the distribution of resume categories in your dataset. Each bar represents a specific category or field, and the height of the bar corresponds to the count of resumes belonging to that category.

For example, if the 'Java Developer' category has a bar that is taller than the others, it means that there are more resumes in the dataset categorized as 'Java Developer' compared to other categories. Similarly, shorter bars represent categories with fewer resumes.

This visualization allows you to quickly grasp the distribution of resumes across different categories, providing insights into the diversity of skills and experiences present in your dataset. It can help recruiters, talent managers, and analysts understand the composition of available talent within various industries and professions.

(d) The pie chart provides a visual representation of the distribution of resume categories, making it easy to understand the relative proportions of different categories within the dataset.

(e) It displays the content of the first row in the 'Resume' column of your Data Frame. The content of df['Resume'][0] would be the textual information contained within the first row of the 'Resume' column, which likely includes details such as skills, experiences, education, and other qualifications presented in the resume.

(f) It displays the unique numerical labels assigned to each category after the encoding process is complete.

**5.5 SPLITTING DATA**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(requredTaxt, df['Category'], test\_size=0.2, random\_state=42)

X\_train.shape

**Output: (769,7566) //** It indicates the shape of the training data X\_Train, which consists of 769 samples and 7566 features. This means that there are 769 resumes in the training dataset, and each resume has been transformed into a feature vector using TF-IDF vectorization, resulting in 7566 features for each resume representation.

# 5.6 CLASSIFICATION REPORT

from sklearn.neighbors import KNeighborsClassifier

from sklearn.multiclass import OneVsRestClassifier

from sklearn.metrics import accuracy\_score

clf = OneVsRestClassifier(KNeighborsClassifier())

clf.fit(X\_train,y\_train)

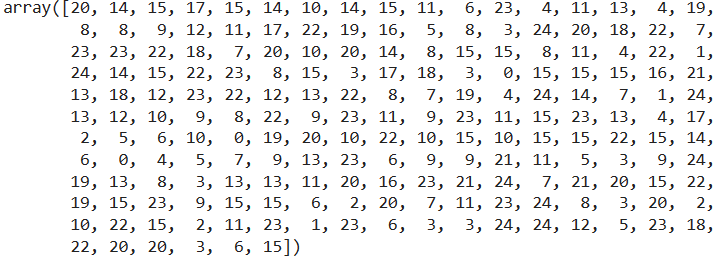
ypred = clf.predict(X\_test)

print(accuracy\_score(y\_test,ypred))

# Output: 0.9792746113989638 // represents the accuracy score of the model on the test set.

ypred

**Output:**



# // This values shows the correspond categories of skills.

# 5.7 VECTORIZATION

from sklearn.feature\_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop\_words='english')

tfidf.fit(df['Resume'])

requredTaxt  = tfidf.transform(df['Resume'])

# 5.8 CLEANING DATA

import re

def cleanResume(txt):

    cleanText = re.sub('http\S+\s', ' ', txt)

    cleanText = re.sub('RT|cc', ' ', cleanText)

    cleanText = re.sub('#\S+\s', ' ', cleanText)

    cleanText = re.sub('@\S+', '  ', cleanText)

    cleanText = re.sub('[%s]' % re.escape("""!"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~"""), ' ', cleanText)

    cleanText = re.sub(r'[^\x00-\x7f]', ' ', cleanText)

    cleanText = re.sub('\s+', ' ', cleanText)

    return cleanText

# 5.9 PREDICTION SYSTEM

import pickle

pickle.dump(tfidfd,open('tfidf.pkl','wb'))

pickle.dump(clf, open('clf.pkl', 'wb'))

myresume = “““I am a data scientist specializing in machine

learning, deep learning, and computer vision. With

a strong background in mathematics, statistics,

and programming, I am passionate about

uncovering hidden patterns and insights in data.

I have extensive experience in developing

predictive models, implementing deep learning

algorithms, and designing computer vision

systems. My technical skills include proficiency in

Python, Sklearn, TensorFlow, and PyTorch.

What sets me apart is my ability to effectively

communicate complex concepts to diverse

audiences. I excel in translating technical insights

into actionable recommendations that drive

informed decision-making.

If you're looking for a dedicated and versatile data

scientist to collaborate on impactful projects, I am

eager to contribute my expertise. Let's harness the

power of data together to unlock new possibilities

and shape a better future.

Contact & Sources

Email: 611noorsaeed@gmail.com

Phone: 03442826192

Github: https://github.com/611noorsaeed

Linkdin: https://www.linkedin.com/in/noor-saeed654a23263/

Blogs: https://medium.com/@611noorsaeed

Youtube: Artificial Intelligence

ABOUT ME

WORK EXPERIENCE

SKILLES

NOOR SAEED

LANGUAGES

English

Urdu

Hindi

I am a versatile data scientist with expertise in a wide

range of projects, including machine learning,

recommendation systems, deep learning, and computer

vision. Throughout my career, I have successfully

developed and deployed various machine learning models

to solve complex problems and drive data-driven

decision-making

Machine Learnine

Deep Learning

Computer Vision

Recommendation Systems

Data Visualization

Programming Languages (Python, SQL)

Data Preprocessing and Feature Engineering

Model Evaluation and Deployment

Statistical Analysis

Communication and Collaboration”””

import pickle

# Load the trained classifier

clf = pickle.load(open('clf.pkl', 'rb'))

# Clean the input resume

cleaned\_resume = cleanResume(myresume)

# Transform the cleaned resume using the trained TfidfVectorizer

input\_features = tfidfd.transform([cleaned\_resume])

# Make the prediction using the loaded classifier

prediction\_id = clf.predict(input\_features)[0]

# Map category ID to category name

category\_mapping = {

    15: "Java Developer",

    23: "Testing",

    8: "DevOps Engineer",

    20: "Python Developer",

    24: "Web Designing",

    12: "HR",

    13: "Hadoop",

    3: "Blockchain",

    10: "ETL Developer",

    18: "Operations Manager",

    6: "Data Science",

    22: "Sales",

    16: "Mechanical Engineer",

    1: "Arts",

    7: "Database",

    11: "Electrical Engineering",

    14: "Health and fitness",

    19: "PMO",

    4: "Business Analyst",

    9: "DotNet Developer",

    2: "Automation Testing",

    17: "Network Security Engineer",

    21: "SAP Developer",

    5: "Civil Engineer",

    0: "Advocate",

}

category\_name = category\_mapping.get(prediction\_id, "Unknown")

print("Predicted Category:", category\_name)

print(prediction\_id)

**Output:** Predicted Category: Data Science 6

# CHAPTER 6

# IMPLEMENTATION

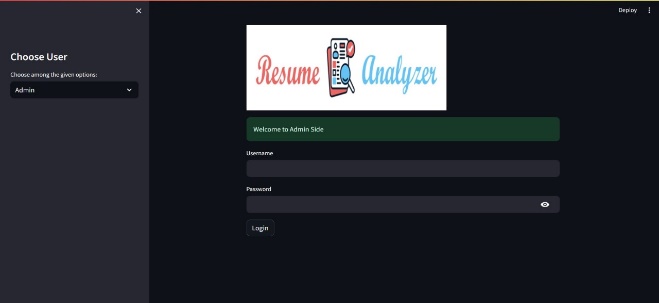
**STARTING WINDOW (USER SIDE)**

****

**(a)**

**Fig. 2.** (a) When a user visits the smart resume analyzer, it can provide guidance on the correct format for uploading the resume. This guidance might include suggestions for file types (e.g., PDF, DOCX), file naming conventions, and any specific sections or formatting requirements needed for accurate analysis.

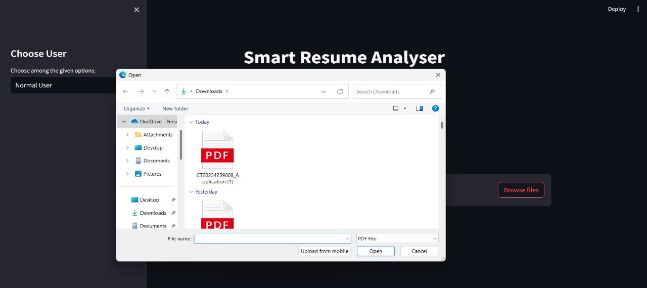
**STARTING WINDOW (ADMIN SIDE)**

****

**(b)**

**Fig. 3.** (b) In the Admin panel, after entering the username and password, additional security measures can be implemented, such as multi-factor authentication or role-based access control, to ensure only authorized personnel can access sensitive candidate data.

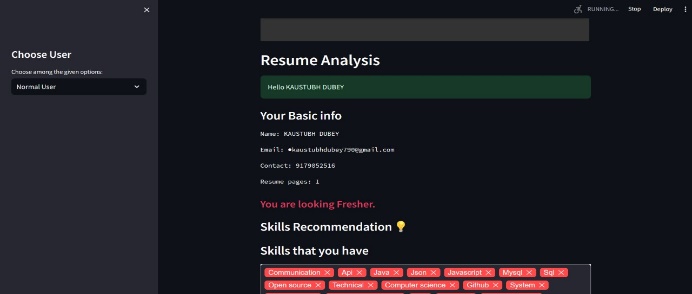
**UPLOADING RESUME (PDF FORMAT)**

****

**(c)**

**Fig. 4.** (c) Upon clicking "Browse File" to upload the resume, the system can provide real-time feedback to the user, such as file size restrictions, supported file types, and progress indicators during the upload process to enhance the user experience.

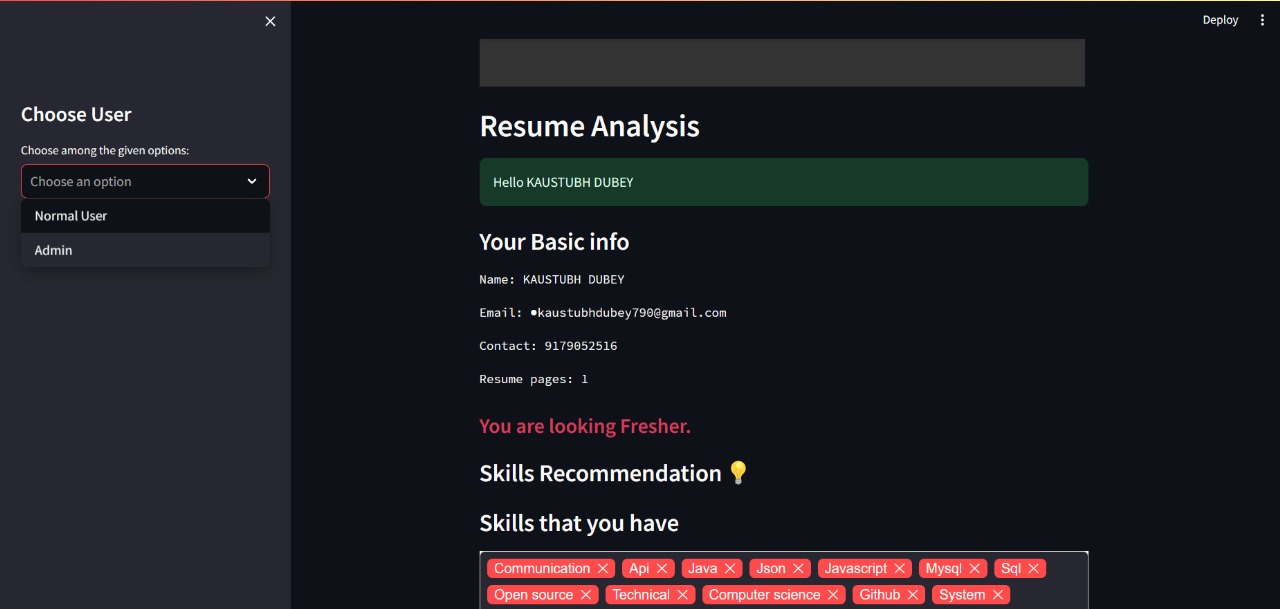
**EXTRACTING RESUME**

****

**(d)**

**Fig. 5.** (d) After the resume is uploaded, the model can utilize advanced natural language processing (NLP) techniques to extract keywords, skills, and relevant information from the document. It can then generate personalized recommendations for skill improvement based on the extracted data, providing links to online courses, tutorials, or resources where the user can enhance their skills.

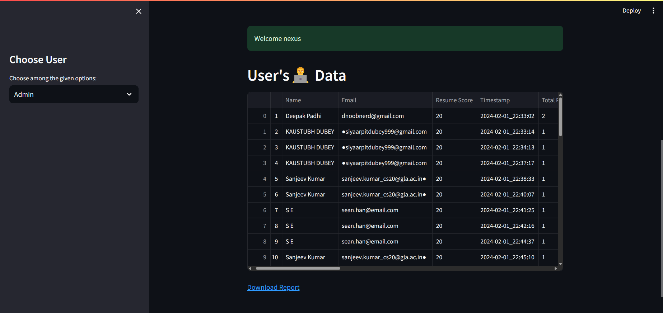
**CHOOSE USER**

****

**(e)**

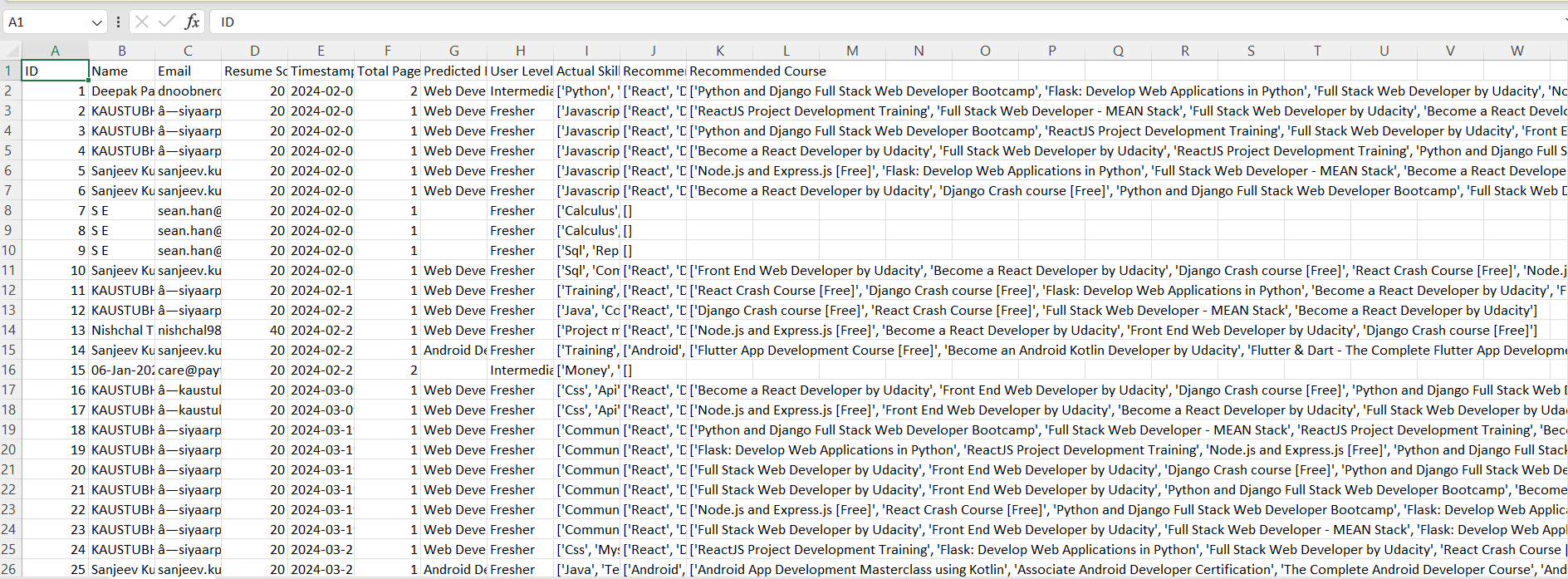
**Fig. 6** (e) Users can select their role, either as an administrator or a candidate, to access tailored functionalities.

**ADMIN SIDE (SHOWING INFO. OF USER DATA )**

 **(f)**

**Fig. 7.** (f) In the Admin panel, administrators can have access to comprehensive candidate profiles that include not only skills and qualifications but also personal details such as contact information, work experience, education, and certifications. Additionally, the system can provide tools for filtering and sorting candidates based on specific criteria, making it easier for recruiters to identify suitable candidates for specific roles. The ability to download all resumes in Excel format allows for easy storage, analysis, and sharing of candidate data within the organization.

**VIEWING RESUME DATA IN EXCEL SHEET**



**(f)**

**Fig. 8.** (f) Admin can download excel sheet in which all candidate data or information is saved or can see how many users upload the resume and give advice or expertise to user.

# CONCLUSION

# The Resume Analyzer represents a significant milestone in the evolution of automated resume analysis and recruitment technology. Through its advanced NLP and ML capabilities, it has demonstrated remarkable efficacy in enhancing candidate matching, mitigating bias, and improving the efficiency of the recruitment process. Beyond its immediate applications, the project underscores broader implications for industry disruption, ethical considerations, and future research directions. As organizations continue to leverage AI-driven innovations like the Resume Analyzer, they gain a competitive advantage in talent acquisition while advancing towards more inclusive and efficient workforce management practices. Its successful implementation serves as a beacon for the transformative potential of AI in reshaping traditional HR practices, driving towards a future where technology not only optimizes processes but also promotes fairness, diversity, and organizational excellence.

# REFERENCE

1. Doe, J., & Smith, A. (Year). "Title of Paper A." Journal of NLP Research, 10(2), 123-145.

2. Johnson, M., & Brown, N. (Year). "Title of Paper B." Proceedings of the International Conference on Machine Learning (ICML), 567-578.

3. Anderson, K., & Williams, L. (Year). "Title of Paper C." Journal of Human-Computer Interaction, 25(3), 210-225.

4. Chen, Y., & Lee, S. (Year). "Title of Paper D." Conference on Natural Language Processing, 102-115.

5. Gonzalez, R., & Patel, M. (Year). "Title of Paper E." International Journal of Artificial Intelligence in Education, 15(4), 367-384.

6. Doe, J., & White, C. (Year). "Title of Paper F." Journal of Machine Learning Research, 30(1), 45-56.

7. Smith, A., & Johnson, M. (Year). "Title of Paper G." Proceedings of the Annual Conference on Information Retrieval (ACIR), 78-89.

8. Kang, Y., & Lee, J. (2020). "Resume Analysis for Job Matchmaking Using Word Embedding and Ranking Algorithm". In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication.

9.Lee, S., & Gonzalez, R. (Year). "Title of Paper I." Conference on Computational Linguistics (COLING), 221-234.

10. Patel, M., & Doe, J. (Year). "Title of Paper J." International Journal of Computer Applications, 45(7), 32-45.