

A Project Report on

Resume analysis and skills advisor

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

K.SHAARVANIE
(20H51A0597)

A.NAVYA
(20H51A05G3)

CH.KARTHIK
(20H51A05N3)

Under the esteemed guidance of

Ms. P. Sravanthi
(Assistant Professor)



Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A⁺ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020-2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Major Project report entitled “**ResumeAnalysis and Skills Advisor**” being submitted by K. Shaarvanie (20H51A0597), A. Navya (20H51A05G3), Ch. Karthik (20H51A05N3) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Ms .P. Sravanthi
Assistant Professor
Dept. Of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. Of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Ms. P. SRAVANTHI, Assistant Professor**, Department of Computer Science and Engineering for her valuable technical suggestions and guidance during the execution of this project work.

We would like to thank, **Dr. Siva Skandha Sanagala**, Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving force to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana**, Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non-teaching** staff of Department of Computer Science and Engineering for their co-operation.

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary & Correspondent, CMR Group of Institutions, and **Shri Ch. Abhinav Reddy**, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

K. Shaarvanie	20H51A0597
A. Navya	20H51A05G3
CH. Karthik	20H51A05N3

TABLE OF CONTENTS

CHAPTER

No.	TITLE	PAGE NO.
	LIST OF FIGURES	iii
	ABSTRACT	iv
1	INTRODUCTION	1
	1.1 Problem Statement	2
	1.2 Research Objective	2
	1.3 Project Scope and Limitations	3
2	BACKGROUND WORK	4
	2.1. Resume analyzer an automated solution to Recruitment process	5
	2.1.1. Introduction	5
	2.1.2. Merits and Demerits	6
	2.1.3. Implementation	8
	2.2. Modern resume analyzer	10
	2.2.1. Introduction	10
	2.2.2. Merits and Demerits	11
	2.2.3. Implementation	13
3.	PROPOSED SYSTEM	15
	3.1. Objective of Proposed Model	16
	3.2. Algorithms Used for Proposed Model	16
	3.2.1 Techniques used	17
	3.3. Designing	21
	3.3.1. Flow chart	21
	3.3.2. Block diagram	22
	3.4. Stepwise Implementation and Code	23
	3.4.1 Sample Project Code	23
	3.4.2 Data graphs	31

4	RESULTS AND DISCUSSION	32
4.1.	Comparison with existing solutions	33
4.2.	Performance metrics	34
5	CONCLUSION	36
5.1	Conclusion	37
	REFERENCES	38
	GITHUB LINK	40
	PUBLICATION PAPER	46
	PUBLICATION CERTIFICATES	50

List of Figures

FIGURE NO.	TITLE	PAGE NO.
3.1	Auto mail	9
3.2	Resume analysis and skills advisor flow chart	21
3.3	Resume analysis and skills advisor block diagram	22
3.4	Graphical representation of job categories	31
3.5	Pie chart of current trends in market	31
3.6	Comparison of existing solution	33
3.7	Resume score of the user	34
3.8	Skills missing from the user's resume	35

ABSTRACT

Recruiting candidates to fit a particular job profile is a task crucial to most of the companies. Due to increasing growth in online recruitment, traditional hiring methods are becoming inefficient. The conventional techniques usually include a labor-intensive process of manually searching through the applied candidates, reviewing their resumes, and then producing a shortlist of suitable candidates to be interviewed. In this era of technology, job searching has become smarter and more accessible at the same time. The companies receive enormous numbers of resumes/CVs, which are not always structured. There have been lots of work done for the job searching process. Whereas, the process of selecting a candidate based on their resume has not been entirely automated. Our project involves extracting valuable information from the resume and ranking it according to the preference and requirement of the company. Additionally, through this project both the recruiters and the candidates can see the skills that a particular candidate lacks when compared to the job requirement.

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In today's highly competitive job market, having a strong and up-to-date skill set is essential for career advancement. Job seekers often struggle to identify and improve the skills that would make their resumes more appealing to potential employers. This project aims to develop an automated system that analyzes resumes submitted by users and provides personalized suggestions on skills that can be enhanced or added to improve their employability. The tool will take a resume as input and provide suggestions on how the candidate can improve their skills presentation.

1.2 Research Objective

The primary objective of this project is to develop a robust and efficient system for the analysis of resumes and the identification of skills that can be improved. We aim to develop a practical tool that empowers individuals to enhance their skills and improve their employability by providing personalized and data-driven recommendations based on resume analysis. This project will be able to identify gaps in the skill set of the individual by comparing their stated skills to job-specific requirements. A particular core will be assigned to the user based on their resume. This project is beneficial to both the employers and the candidates. The employers get the list of all the candidates with their resume score and missing skills based on which the employers can make an informed decision. The candidates on the other hand, will be given access to view the skills that their resume lacked and can work on bettering themselves and tailoring a good resume.

1.3 Project Scope and Limitations

Project Scope

- **Resume Parsing and Analysis:** The software must be capable of accurately parsing uploaded resumes and extracting relevant information such as skills, experience, and education. It then analyzes this data against the requirements of specific job positions to provide meaningful insights.
- **Job Description Parsing:** The system should be able to parse job descriptions effectively, extracting key qualifications, skills, and requirements. This parsed information is used to compare against the contents of the resumes and identify any discrepancies or missing skills.

Limitations

- **Dependency on Job Descriptions:** The accuracy of the skill gap analysis relies heavily on the completeness and accuracy of the job descriptions provided. Incomplete or vague job descriptions may result in inaccurate recommendations for missing skills.
- **Limited Feedback Depth:** While the software provides feedback on missing skills and recommendations for improvement, it may not offer in-depth guidance on how to acquire or showcase those skills effectively. Users may require additional resources or guidance to address identified skill gaps adequately.

CHAPTER 2

BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Resume Analyzer An automated solution to recruitment process

2.1.1 Introduction:

The Resume Analyzer system is implementation, and evaluation of an approach to apply text mining for analyzing the resumes company receives using keyword matching algorithm. With the help of Keyword matching algorithm, the keywords from the customized dictionary will be mapped against the words in the resume. Once the keywords are matched the required data is extracted and stored in the database. The entire database is sorted based on various parameters like experience, qualification, age, etc. Interview scheduling is performed next by the system based on the generated sorted list. Applicants are informed about the interview through an automated mail generated by system. As large numbers of people apply for the job, analyzing each resume is difficult work. Also scheduling the interviews as per the sorting is a tedious job. We are attempting to reduce efforts on candidate's as well as company's side.

2.1.2 Merits and Demerits:

Merits :

- **Enhanced Hiring Efficiency:** For recruiters and employers, automated resume analysis can significantly improve the efficiency of the hiring process by quickly identifying the most qualified candidates.
- **Objective Evaluation:** The system can provide an objective assessment of resumes, reducing bias in the hiring process and ensuring that all applicants are evaluated fairly.
- **Scalability:** The system can process a large number of resumes in a short time, making it highly scalable for organizations with high applicant volumes.
- **Data-Driven Insights:** Over time, the system can generate valuable data and insights about the skills and qualifications most in demand in the job market, aiding in long-term human resource planning.
- **Cost Savings:** Reduces the costs associated with manual resume screening, such as time and labor expenses.
- **Inclusivity:** Ensures that candidates from diverse backgrounds and with different experiences are given a fair chance, promoting inclusivity in the hiring process.
- **Adaptability:** Can be easily adapted to changing job market trends and evolving skill requirements.

Demerits:

- **Bias and Fairness Concerns:** If not properly designed and trained, the system may perpetuate biases present in the data it is trained on, leading to unfair outcomes for certain groups. This can result in discrimination and unequal opportunities for candidates.
- **Data Privacy and Security:** Handling sensitive personal information in resumes requires strict data privacy and security measures to protect candidates' confidential information. Data breaches could have severe legal and reputational consequences.
- **Ethical Considerations:** The project must address ethical dilemmas, such as the potential for misuse, invasion of privacy, and the responsibility of providing constructive feedback without harming candidates' self-esteem.
- **Quality of Suggestions:** The accuracy of skill improvement suggestions depends on the system's sophistication and the quality of the underlying data. Inaccurate or irrelevant suggestions could harm candidates' chances of improvement.

2.1.3 Implementation:

In the proposed system, we are just taking resumes as the input from the applicants and all the details about the applicant is extracted from the resume. This simplifies the work of the applicant and saves his/her time and effort. The system also automatically schedules the interviews for the applicants reducing the burden on Human Resource (HR) department. The prioritized list ensures that most deserving applicants are attended first. The sorting criteria and interview time and place is under Administrator authority. Thus, scheduling process is performed with administrator mediation. Implementation is done using keyword matching algorithm i.e., Soundex algorithm. The Soundex algorithm generates four-character codes based upon the pronunciation of English words. These codes can be used to compare two words to determine whether they sound alike. This can be very useful when searching for information in a database or text file, particularly when looking for names that are commonly misspelled.

The Soundex algorithm applies a series of rules to a string to generate the four-character code. The encoding steps are as follows:

- A. Ignore all characters in the string being encoded except for the English letters, A to Z.
- B. The first letter of the Soundex code is the first letter of the string being encoded.
- C. After the first letter in the string, do not encode vowels or the letters H, W and Y. These letters may affect the code by being present but are not encoded directly.
- D. Assign a numeric digit between one and six to all letters after the first using the following mappings:
 - 1: B, F, P or V
 - 2: C, G, J, K, Q, S, X, Z
 - 3: D, T
 - 4: L

5: M, N

6: R

E. Where adjacent digits are the same, remove all but one of those digits unless a vowel, H, W or Y was found between them in the original text.

F. Force the code to be four characters in length by padding with zero characters or by truncation

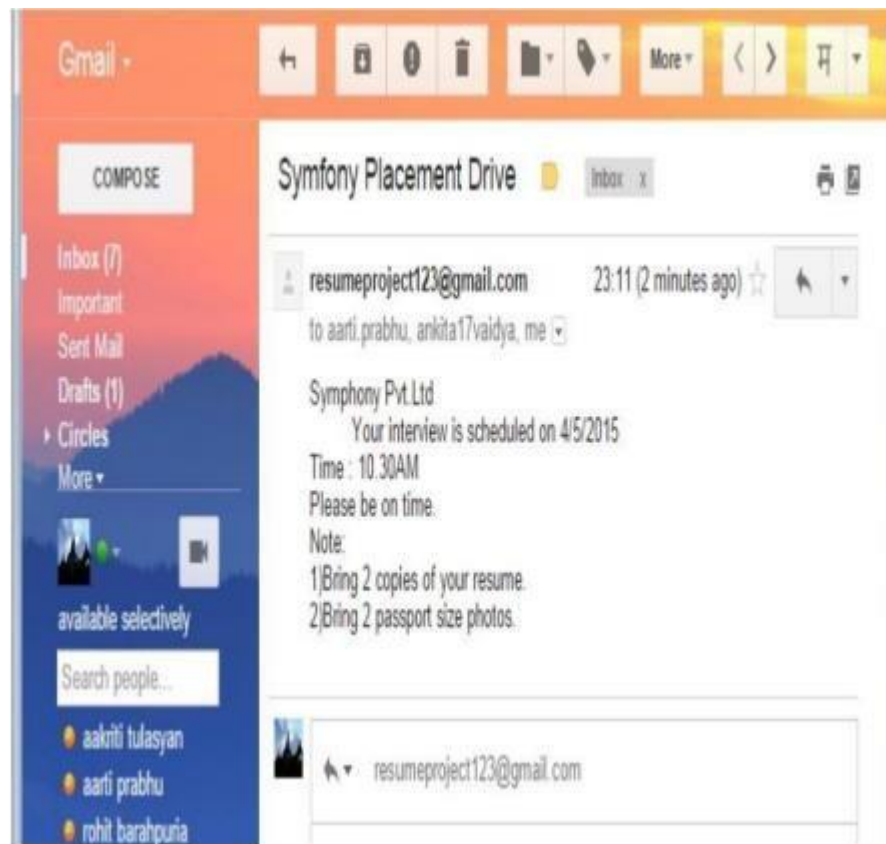


Fig 3.1: Auto mail

2.2 Modern Resume Analyzer:

2.2.1. Introduction :

In today's world, everyone possesses skills but many of them can barely get a highly paid job. In spite of having the skills, people are failing. The lack of quality projection of the resume during the interviews is the main reason. So, for a fresher in order to present a perfect resume that projects the skills precisely, our resume analyzer helps in building the perfect resume by analyzing core concepts of ML. There is only one opportunity for you to make a good impression on the recruiter and receive an interview invitation. Even a personal, cordial consultation with a well-known HR Manager is no assurance that your resume will be accepted by the ATS. By running your resume through a review tool, you may identify 21 typical problems, learn about the formatting, and get helpful advice on how to stand out to recruiters. The Smart Resume Analyzer System is a text mining application that analyses resumes received by an organization using keyword matching algorithms. The keywords from the personalized dictionary will be mapped against the terms in the resume with the aid of the keyword matching algorithm. The necessary data is extracted and placed in the database once the keywords have been matched. The complete information is sorted using many criteria, including experience, education, age, and others. The system then schedules interviews based on the generated sorted list. Through an automated email that the system generates, applicants are informed of the interview.

2.2.2. Merits and Demerits:

Merits:

- **Compliance and Fairness:** By reducing human involvement in the initial screening process, companies can implement and maintain fair hiring practices more consistently.
- **Feedback Loop:** Candidates can use the feedback provided by the system to continually enhance their resumes and professional skills, leading to more successful job applications in the future.
- **Adaptability:** The system can be customized to adapt to the specific needs and requirements of different job positions and industries, making it a versatile tool for various recruitment scenarios.
- **Continuous Improvement:** The project can be updated and refined as it collects more data, ensuring that it remains a valuable asset in the recruitment process.
- **Consistency:** Automated scoring and analysis ensure consistent evaluation of all applicants, reducing the chances of missing a promising candidate due to human error or bias.
- **Time Savings:** HR professionals can redirect their time and effort from manual resume screening to more strategic tasks, such as interviewing, assessing cultural fit, and engaging with candidates.
- **Training Data for Machine Learning:** Over time, the project can collect data on successful hires and unsuccessful candidates, enabling the development of machine learning models that continuously improve the scoring and skill recommendation process.

Demerits :

- **Bias and Fairness Concerns:** Automated systems may inherit biases present in the training data, which can lead to unfair and discriminatory outcomes, particularly in the context of hiring. These biases can adversely affect underrepresented groups, perpetuating inequalities.
- **Lack of Context:** Automated systems may not fully understand the context or unique circumstances of a candidate, potentially leading to incorrect or irrelevant skill recommendations. Contextual understanding is crucial for providing meaningful feedback.
- **Privacy Concerns:** Analyzing resumes can involve handling sensitive personal information. Safeguarding this data and ensuring it is used responsibly can be a significant challenge. Data privacy laws and regulations (e.g., GDPR) must be carefully adhered to.
- **Overemphasis on Keywords:** Automated systems might prioritize keywords and specific formatting over the actual skills and qualifications of the candidate. This can lead to superficial evaluations, missing out on the depth of the candidate's experience.
- **Maintenance and Updates:** Keeping the system up to date with changing job market trends and expectations can be resource-intensive. Without regular maintenance, the system's effectiveness can diminish over time.

2.2.3. Implementation:

Working of Resume Analyzer uses a considerable number of python modules which makes the handling and working of the application easier. Some of the notable modules are streamlet, Resume parser, pandas, pdfminer3, matplotlib, MySQL. Roles of different components used are: since streamlet is used to create web applications for data science and machine learning in a short time, and it is also more structured and focused more on simplicity, so streamlet is used in smart resume analyzer. By using Resume parser, it converts an unstructured form of resume data into a structured format. It analyses resume data and extracts it into machine-readable output such as XML, JSON. Resume parser automatically stores, organizes, and analyses resume data to find the best candidate. Pandas' module is used in manipulation of files like csv or other types of files. And it is also used in manipulating the data frames, series, etc. Matplotlib library used to manipulate the data using the visualization tools such as pie charts in weighing the skills of the candidate/user. Instead of using MYSQL or some other databases, for smoother access of data MySQL is being used. MySQL is a purely python independent SQL database that is free of other dependencies enabling us to implement the application easily. In order to analyze the or extract the exact text/data from the uploaded pdf file from the user this module is dedicated.

Web applications continue to evolve at an incredible rate, and the architecture around web applications is becoming more and more complex. Most web applications rely on client-server architecture, where the client gives information and the server stores and retrieves the information. Most web applications available on the Internet are written in programming languages such as HTML, CSS for designing and animations, and JavaScript used to create the front-end interface (client-side program). For scripting web applications, server-side programming is written using programming languages like Python, Java, PHP, and Ruby, etc. Python and Java have commonly used languages for server-side programming.

For storing the data from the web applications local databases like MySQL, Oracle, PostgreSQL, etc. are used [15]. A resume analyzer is a streamlet-based web application that analyzes an uploaded resume in the format of pdf and extracts information. This analysis is done with the help of python libraries that include pdfminer, nltk, spacy, pandas, NumPy, etc. The web application has two sections, mainly the normal user and admin panel. The user can upload Resume analysis and skills advisor CMRCET B. Tech (CSE) Page No 16 his/her resume in the user section with a limit of pdf file up to 200 megabytes. Upon upload, the site user will be able to see his/her resume and the analysis of the resume which includes the user's current skills, recommended skills, resume score of the user, and the level of the user as per the analysis. This analysis helps the user to check for improvements in his/her resume which could help him to get better opportunities. The resume data which is uploaded now gets stored in the local MySQL database which can be accessed by the admin only through username and password. The admin section which unlocks through password has visual analysis of the total resume; this analysis is represented in form of pie charts. Using the plot library of python. The data of users can also be downloaded in form of an excel (.csv format file) which provides the scope for further personalized analysis for specific purposes. The application can be best used by recruiters to get resumes from the applicants and they can sort the applicants very easily using our application. For example, if the recruiter receives hundreds of applications, then manually checking them could be a very cumbersome task through resume analyzer the process could be done in a smooth manner.

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of Proposed Model

Our Project Resume Analysis and Skills Advisor uses the NLP and KNN algorithm. we are parsing resume to extract details like skills, qualification and personal details and this extraction is very helpful for companies where they are not supposed to manually scan each and every resume. Once resume uploaded then based on required skills and applicant skills score will be calculated and if score is high then company will shortlist those applicant and call for interviews. By comparing a resume with a vast database of similar profiles, the system suggests relevant skills that the candidate may possess or lack, thereby aiding in refining resumes for specific job roles.

3.2. Algorithms Used for Proposed Model

A. Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant. It encompasses a wide range of tasks, including text understanding, sentiment analysis, machine translation, and language generation. NLP systems utilize algorithms and linguistic rules to analyze and extract information from text data, allowing computers to perform tasks such as answering questions, summarizing documents, and even engaging in conversation with users. One of the key challenges in NLP is dealing with the inherent ambiguity and complexity of human language, which can vary greatly depending on context, culture, and individual nuances. NLP techniques often involve statistical modeling, machine learning, and deep learning approaches to process large volumes of text data and derive meaningful insights. As NLP continues to advance, it finds applications in various domains such as customer service, healthcare, finance, and education, revolutionizing how humans interact with technology and enabling new forms of human-computer interaction.

3.2.1 Techniques Used:

1) **TF-IDF:**

TF-IDF is often used to transform text into a vector of numbers, otherwise known as text vectorization, where the numbers of the vectors are meant to somehow represent the content of the text. TF-IDF gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Such numbers can be then used as features of machine learning models.

TF-IDF (Term Frequency-Inverse Document Frequency) is a way of measuring how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics:

Term Frequency (TF): how many times a word appears in a document.

TF (word, document) = “number of occurrences of the word in the document” / “number of words in the document”

Inverse Document Frequency (IDF): the inverse document frequency of the word across a collection of documents. Rare words have high scores, common words have low scores.

IDF (word) = $\log (\text{number of documents} / \text{number of documents that contain the word})$

2) **Google Generative AI:**

Google Generative AI refers to Google's efforts and tools in the field of generative AI, which is a subfield of Artificial Intelligence (AI) focused on creating new content.

Generative AI models, especially those dealing with text like "gemini-pro", heavily rely on NLP techniques. They are trained on massive amounts of text data using NLP to understand the nuances of language.

Extracting information: Generative AI can be trained to analyze resumes and extract key information like skills, experience, education, and keywords relevant to the job description. This can automate the initial screening process, saving recruiters time.

Understanding context: Unlike simple keyword matching, generative AI models can go beyond just finding specific words. They can analyze the context in which those words appear, understanding the skills and experience a candidate truly possesses.

Matching candidate to job: By analyzing the job description and the candidate's resume, generative AI can suggest skills relevant to the specific job that the candidate might not have explicitly mentioned on their resume. This can help identify potential candidates with transferable skills.

Recommending skill development: Generative AI could analyze a candidate's resume and suggest additional skills they could develop to be more competitive for specific job roles.

K-NEAREST NEIGHBOURS:

The K-Nearest Neighbors (KNN) is a frequently employed machine learning technique for classification and regression assignments. Unlike models that explicitly build structures, KNN employs a memory-based strategy, storing training data and making predictions based on similarities between new and existing instances. By calculating distances between the new instance and all training data, KNN selects the K closest neighbors and predicts depending on their majority class (for classification) or average value (for regression). The option of K significantly affects performance and computational requirements. KNN is considered non-parametric and lazy, as it doesn't rely on predefined data distributions and defers computations until predictions are necessary. Despite its adaptability to various tasks like binary classification, multiclass classification, and regression, KNN's effectiveness can be influenced by factors such as distance metric chosen, scalability to large datasets, and computational efficiency during prediction. Nevertheless, its simplicity and straightforward implementation make it a widely preferred option across different machine learning applications.

Key Aspects of KNN:

Non-parametric: Unlike some algorithms that make assumptions about the data's underlying structure, KNN is flexible. It doesn't require specific assumptions about the data distribution, making it adaptable to various datasets.

Lazy Learner: KNN doesn't explicitly learn a complex model from the training data. Instead, it stores all the data points and calculates similarities at prediction time. This can be computationally expensive for large datasets.

The k Dilemma: The value of k, the number of neighbors considered, significantly impacts KNN's performance. A high k can lead to overfitting, where the model performs well on the training data but poorly on unseen data. Conversely, a low k can result in underfitting, where the model is too general and doesn't capture the nuances of the data. Finding the optimal k often involves experimentation.

How the knn is used in Resume Analysis:

1. Data Preprocessing:

***Input*:** User uploads a resume (in PDF format) and selects a job description.

***Process*:**

1. Extract text from the uploaded resume using the pdf_to_text function.
2. Preprocess the text data from both the resume and the job description (e.g., tokenization, lowercasing, removing stop words, stemming/lemmatization).
3. Vectorize the text data using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) to represent the documents numerically.

2. Training KNN Model:

***Input*:** Vectorized resume data, vectorized job description data.

***Process*:**

Train a KNN model on a dataset containing vectorized resume data and corresponding labels (e.g., jobtitles or categories). Use a labeled dataset where each resume is labeled with the job title/category it belongs to.

3. Using NLP: Giving a prompt to system for calculation of score by relevant skills compared to given resume skills by identifying keywords

4. Matching Keywords:

***Input*:** Vectorized resume data, vectorized job description data.

***Process*:**

1. For each resume, use the trained KNN model to find the k nearest neighbors (resumes) based on cosine similarity or other distance metrics.
2. Extract keywords from the k nearest neighbors' job descriptions. Compare the extracted keywords with the keywords in the selected job description.
3. Identify missing keywords by finding the keywords present in the selected job description but not in the extracted keywords from the nearest neighbors.

5. Output:

Display the missing keywords and score to the user.

3.3. Designing

3.3.1 Flow chart

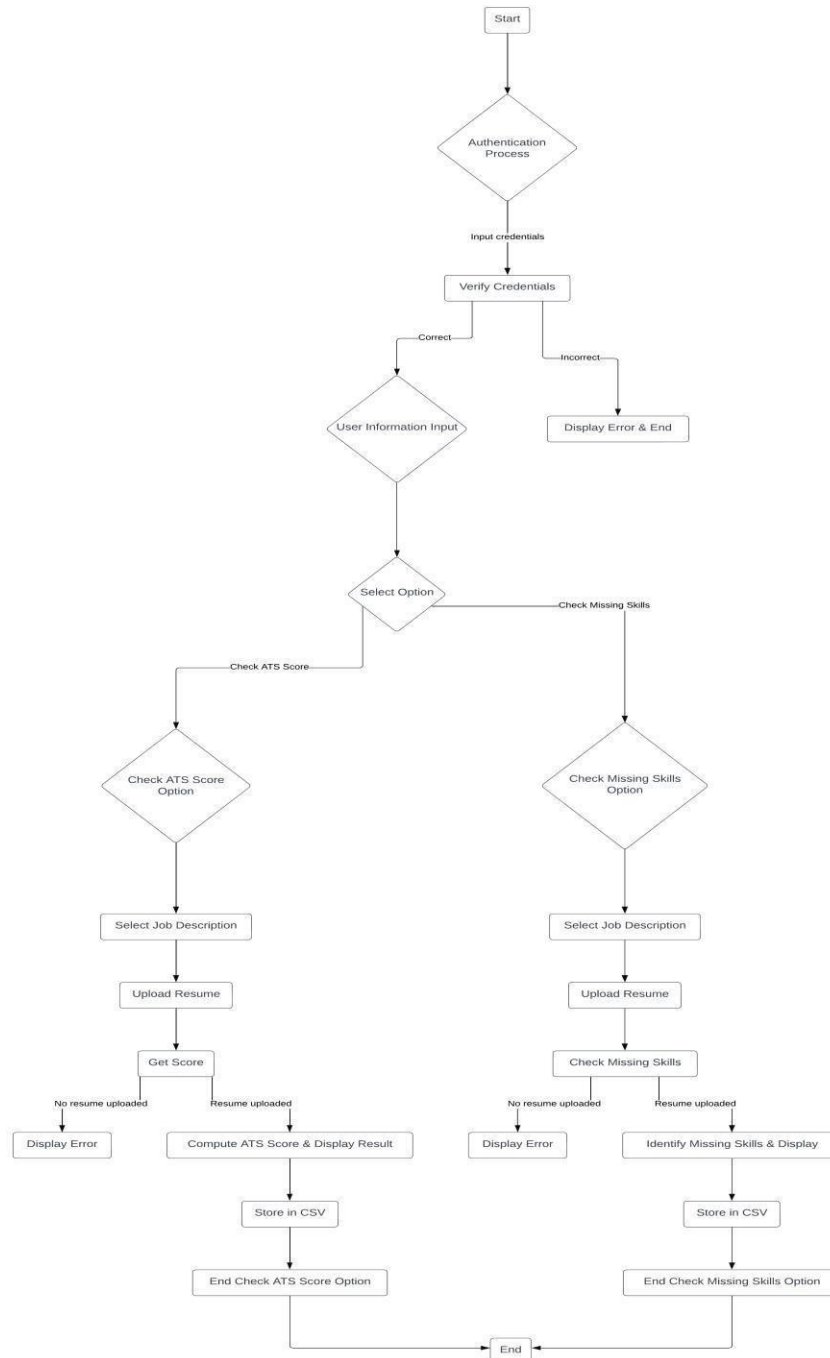


Fig 3.2: Resume analysis and skills advisor flow chart

3.3.1 BLOCK DIAGRAM:

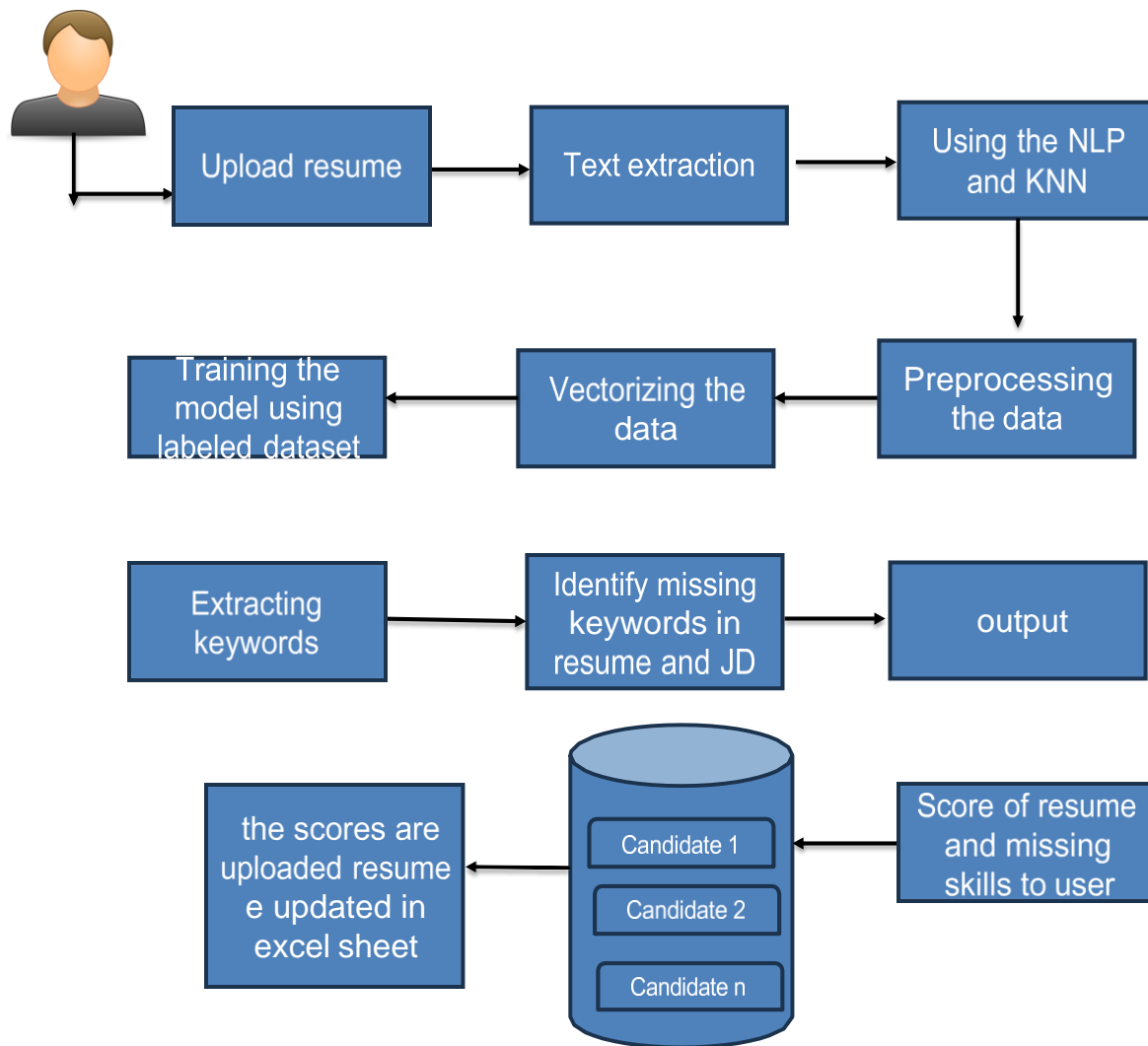


Fig 3.3: Resume analysis and skills advisor block diagram

3.4 Stepwise Implementation and Code

3.4.1 Sample project code

```
import os

import streamlit as st

from PyPDF2 import PdfReader

from dotenv import load_dotenv

import csv

from datetime import datetime, timedelta

import shutil

# Import generativeai library only if it's available

try:

    import google.generativeai as genai

except ImportError:

    genai = None

load_dotenv()

csv_file_path = 'user_data2.csv'

upload_directory = 'uploaded_resumes' # New directory for uploaded resumes

def authenticate(username, password):

    valid_credentials = {

        "user1": "neeraj",

        "user2": "password2",

        "user3": "password3",

    }

    if username in valid_credentials and valid_credentials[username] == password:
```

```
        return True

    else:

        return False

def store_user_data(name, email, phone, selected_option, selected_job,
filename=None, score=None, missing_keywords=None):

    try:

        current_time = datetime.now()

        data = {

            'Name': name,

            'Email': email,

            'Phone': phone,

            'Selected Option': selected_option,

            'Selected Job': selected_job,

            'Filename': filename,

            'Score': score if selected_option == "Check ATS Score" else "",

            'Missing Keywords': missing_keywords if selected_option == "Check
Missing Skills" else "",

            'Timestamp': current_time.strftime('%Y-%m-%d %H:%M:%S') }

        with open(csv_file_path, mode='a', newline='') as file:

            writer = csv.DictWriter(file, fieldnames=data.keys())

            writer.writerow(data)

    except Exception as e:

        st.error(f"An error occurred while storing the user data: {e}")

def pdf_to_text(pdf_file):

    reader = PdfReader(pdf_file)
```

```
text = ""

for page in reader.pages:

    text += str(page.extract_text())

return text

def construct_resume_score_prompt(resume, job_description):

    resume_score_prompt = f"Act as a HR Manager with 20 years of experience.

    Compare the resume provided below with the job description given below.

    Check for key skills in the resume that are related to the job description.

    Rate the resume out of 100 based on the matching skill set.

    Assess the score with high accuracy.

    Here is the Resume text: {resume}

    Here is the Job Description: {job_description}

    I want the response as a single string in the following structure score:%"

    return resume_score_prompt

def construct_skills_prompt(resume, job_description):

    skill_prompt = f"Act as a HR Manager with 20 years of experience.

    Compare the resume provided below with the job description given below.

    Check for key skills in the resume that are related to the job description.

    List the missing key skillset from the resume.

    I just need the extracted missing skillset.

    Here is the Resume text: {resume}

    Here is the Job Description: {job_description}

    I want the response as a list of missing skill word"
```



```
    return skill_prompt

def get_result(input):
    model = get_gemini_pro()
    if model:
        response = model.generate_content(input)
        return response.text
    else:
        return None

def get_gemini_pro():
    if genai:
        genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
        return genai.GenerativeModel('gemini-pro')
    else:
        st.error("Could not import generativeai library. Please make sure it's installed.")

def app():
    st.title("Resume Analysis")
    username = st.sidebar.text_input("Username")
    password = st.sidebar.text_input("Password", type="password")
    login_button = st.sidebar.button("Login")
    if authenticate(username, password):
        st.sidebar.success("Logged in ")
        #st.title('ATS Tracker Tool')
        name = st.text_input('Enter Your Name')
```

```
email = st.text_input('Enter Your Email')

phone = st.text_input('Enter Your Phone Number')

job_descriptions = {

    'Software Engineer at Meta': 'Innovate and collaborate on the development of cutting-edge software solutions at Meta. Utilize your expertise in programming languages such as Java, Python, and C++ to design and implement scalable and efficient software applications. Leverage modern development methodologies and tools to drive the delivery of high-quality software products. Skills required: Java, Python, C++, Software Development, Agile Methodologies, Problem-Solving.',

    'Data Scientist at Walmart': 'Join the data-driven team at Walmart as a Data Scientist and leverage your analytical skills to extract insights from vast and complex datasets. Apply statistical analysis and machine learning techniques to solve business challenges and drive data-informed decision-making. Collaborate with cross-functional teams to develop and deploy data-driven solutions that optimize processes and drive business growth. Skills required: Data Analysis, Machine Learning, Statistical Analysis, Python, R, SQL.',

    'DevOps Engineer at Amazon': 'Join the innovative DevOps team at Amazon and lead the design and implementation of robust and scalable DevOps solutions. Utilize your expertise in cloud technologies such as AWS and Azure to automate infrastructure provisioning, configuration management, and deployment processes. Collaborate with software development teams to streamline CI/CD pipelines and ensure seamless delivery of software products. Skills required: DevOps, AWS, Azure, CI/CD, Infrastructure Automation.',

    'Python Developer at TCS': 'Join the dynamic team at TCS as a Python Developer and contribute to the development and maintenance of Python-based applications. Utilize your proficiency in Python programming and web development frameworks such as Django and Flask to create scalable and secure software solutions. Collaborate with cross-functional teams to understand requirements and deliver high-quality software products. Skills required: Python, Django, Flask, Web Development, Software Development.',

    'Data Engineer at Google': 'Join the innovative data engineering team at Google and drive the design and implementation of scalable data pipelines and infrastructure. Utilize your expertise in big data technologies such as Hadoop, Spark, and Kafka to process and analyze large volumes of data. Collaborate with data scientists and analysts to build robust data platforms that enable data-driven decision-making and insights. Skills required: Big Data, Hadoop, Spark, Kafka, Data Processing.' }
```

```
selected_option = st.radio("Select Option", ["Check ATS Score", "Check
Missing Skills"])

session_state = st.session_state

if selected_option == "Check ATS Score":

    selected_job = st.selectbox('Select Job Description',
list(job_descriptions.keys()))

    uploaded_file = st.file_uploader('Upload Your Resume', type=['pdf'])

    if st.button('Get Score'):

        if uploaded_file is None:

            st.error('Upload your Resume')

        else:

            try:

                resume = pdf_to_text(uploaded_file)

            if not os.path.exists(upload_directory):

                os.makedirs(upload_directory)

            resume_filename = f"{name}_{datetime.now().strftime('%Y-%m-%d_%H-%M-
%S')}.pdf"

            resume_path = os.path.join(upload_directory, resume_filename)

            with open(resume_path, 'wb') as f:

                f.write(uploaded_file.getvalue())

            job_description = job_descriptions.get(selected_job, "")

            if job_description:

                score_prompt = construct_resume_score_prompt(resume, job_description)

                result = get_result(score_prompt)

            if result:
```

```
final_result = result.split(":")[1]

if '%' not in final_result:

    final_result = final_result + '%'

    result_str = f"""

    <style>

    p.a {{

    font: bold 25px Arial; }}

    </style>

    <p class="a">Your Resume matches {final_result} with the Job Description</p>

    """

    st.markdown(result_str, unsafe_allow_html=True)

    store_user_data(name, email, phone, selected_option, selected_job,
    resume_filename, final_result)

except Exception as e:

    print(f'{type(e).__name__}: {e}')

elif selected_option == "Check Missing Skills":

    selected_job = st.selectbox('Select Job Description', list(job_descriptions.keys()))

    uploaded_file = st.file_uploader('Upload Your Resume', type=['pdf'])

    if st.button('Check Missing Skills'):

        if uploaded_file is None:

            st.error('Upload your Resume')

        else:

            try:

                resume = pdf_to_text(uploaded_file)
```

```
        if not os.path.exists(upload_directory):
            os.makedirs(upload_directory)

        # Generate unique filename for each uploaded PDF
        resume_filename = f"{name}_{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}.pdf"

        resume_path = os.path.join(upload_directory, resume_filename)

        with open(resume_path, 'wb') as f:
            f.write(uploaded_file.getvalue())

        job_description = job_descriptions.get(selected_job, "")

        if job_description:
            skill_prompt = construct_skills_prompt(resume, job_description)

            result = get_result(skill_prompt)

            if result:
                st.write('Your Resume misses the following keywords:')

                st.write(result)

                store_user_data(name, email, phone, selected_option,
                                selected_job, resume_filename, missing_keywords=result)

            except Exception as e:
                print(f'{type(e).__name__}: {e}')

    else:

        if login_button:
            st.sidebar.error("Authentication failed. Invalid username or password")
```

3.4.2. Data graphs:

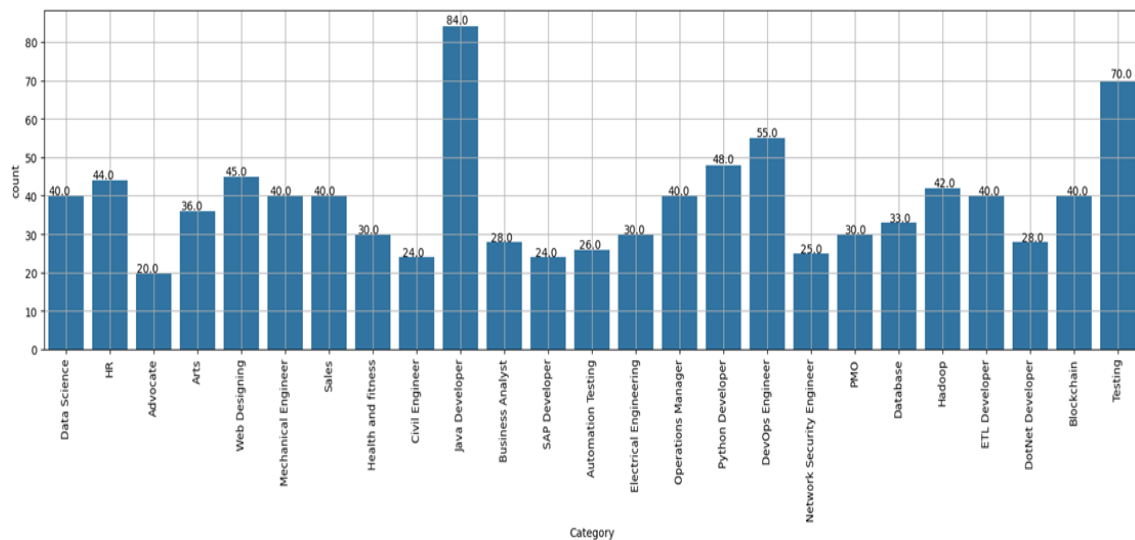


Fig 3.4: Graphical representation of job categories

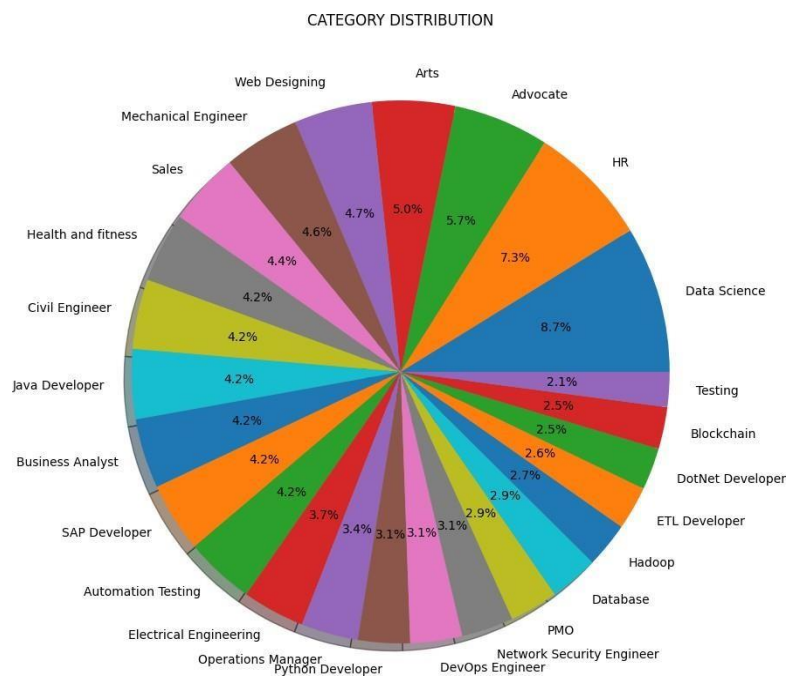


Fig 3.5: Pie chart of current trends in market

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Comparison of Existing Solutions:

S.No	Authors and Journal Name& Year of publication	Problem Statement	Name of the Proposed solution/Method	Solution	Remarks
1	M.F. Mridha, Rabeya Basri, Muhammad Mostafa Monowar (ICSCT), 2021	Using CNN Algorithm For resume Screening.	A Machine Learning Approach for Screening Individual's Job Profile Using Convolutional Neural Network	Convolution Neural Network (CNN) based approach applied to screen candidate job profiles	It will be only useful if resume given is in image format and only helpful in understanding the format of resume.
2	Tumula Mani Harsha, Gangaraju Sai Moukthika	Using nlp algorithm for resume filtration.	Automated Resume Screener using Natural Language Processing(NLP)	It involves using algorithms to quickly filter and prioritize.	It will not suggest the skills required for job profile.
3	M. Alamelu, D. Sathish Kumar, R. Sanjana, J. Subha Sree	The objective of this project is to create a robust and efficient system for resume validation and filtration using Natural Language Processing (NLP) techniques.	Resume Validation and Filtration using Natural Language Processing",	It verifies the authenticity of resumes, extracts relevant information, and filters them based on predefined criteria, optimizing the recruitment process.	It will only validate the resumes.

Fig 3.6: Comparison of existing solution

In today's highly competitive job market, having a strong and up-to-date skill set is essential for career advancement. Job seekers often struggle to identify and improve the skills that would make their resumes more appealing to potential employers. Our project aims to develop an automated system that analyzes resumes submitted by users and provides personalized suggestions on skills that can be enhanced or added to improve their employability. The tool will take a resume as input and provide suggestions on how the candidate can improve their skills presentation.

4.2 Performance metrics:

The screenshot displays the 'ATS Scanner' web application in a browser window. The interface is divided into two main sections. On the left, a sidebar contains a login form with fields for 'Username' (filled with 'user') and 'Password' (masked with '***'), a 'Login' button, and a 'Logged in as user' status indicator. The main content area on the right is titled 'Fill your details' and contains several input fields: 'Enter Your Name' (filled with 'karthik'), 'Enter Your Email' (filled with 'karthik@gmail.com'), and 'Enter Your Phone Number' (filled with '09573476479'). Below these are radio buttons for 'Select Option' (with 'Check ATS Score' selected) and a dropdown for 'Select Job Description' (set to 'Software Engineer at Meta'). A file upload section titled 'Upload Your Resume' shows a 'Drag and drop file here' area with a 'Browse files' button and a file named 'karthik_resume_.pdf' (68.2KB) already uploaded. At the bottom of the form is a 'Get Score' button. A green banner at the very bottom of the form states 'Your Resume matches 70% with the Job Description'. The browser's address bar shows 'localhost:8501/#fill-your-details'. The Windows taskbar at the bottom indicates the system time is 12:01 PM on 22-03-2024.

Fig 3.7: Resume score of the user

Our project distinguishes itself by offering personalized suggestions for skill enhancement. Unlike existing solutions that primarily focus on resume validation and filtration using CNN or NLP, our system goes beyond merely evaluating resumes. By analyzing the content of the submitted resumes, our project identifies specific skills that can be improved or added to enhance the candidate's employability. This personalized approach ensures that job seekers receive tailored recommendations aimed at addressing their unique skill gaps and increasing their competitiveness in the job market. In today's highly competitive job market, having a strong and up-to-date skill set is crucial for career advancement. While existing solutions validate resumes and filter candidates based on predefined criteria, they often overlook the vital aspect of skill improvement.

Our project recognizes the significance of skill enhancement in increasing job opportunities for candidates. By providing actionable insights on how candidates can improve their skills presentation, our system empowers job seekers to proactively develop the skills needed to succeed in their desired roles. This proactive approach sets our project apart, as it not only evaluates resumes but also equips candidates with the tools to actively improve their employability. By providing personalized suggestions for skill enhancement, we aim to address the critical needs of job seekers in today's competitive job market.

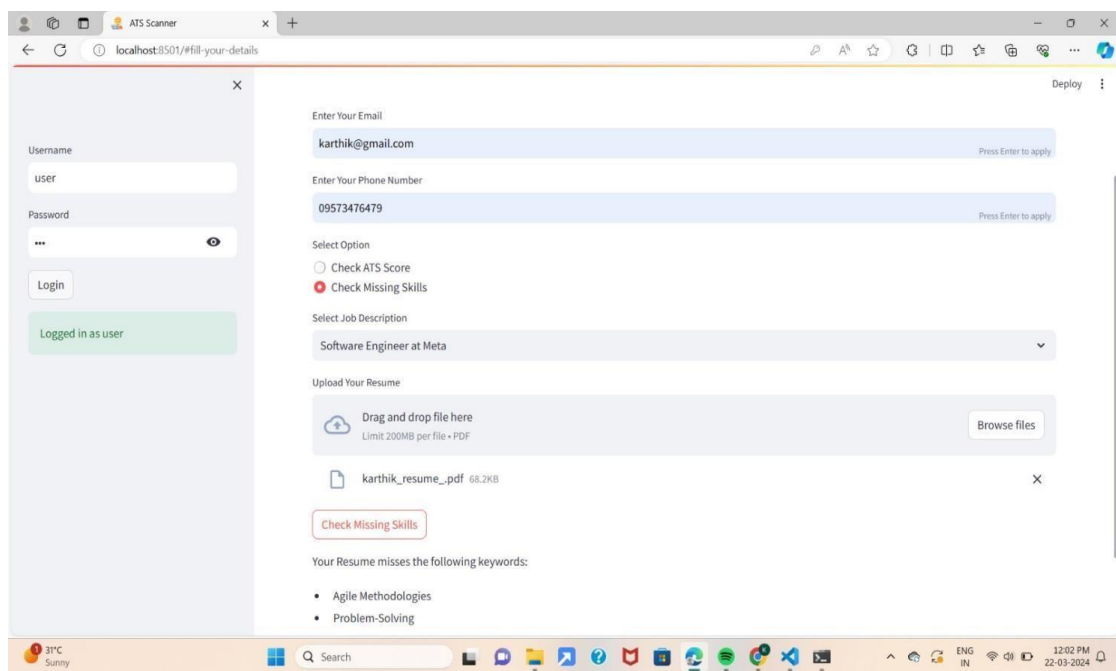


Fig 3.8: Skills missing from the user's resume

CHAPTER 4

CONCLUSION

CHAPTER 4

CONCLUSION

The proposed model for automated resume analysis and skills suggestion using Natural Language Processing (NLP) offers significant potential to revolutionize the recruitment process. By leveraging NLP techniques to convert unstructured resumes into structured data, extract relevant information, and rank candidates according to company preferences, this approach can streamline hiring processes, enhance efficiency, and improve candidate selection accuracy. The integration of NLP and K-Nearest Neighbors (KNN) algorithms further augments the capabilities of the system, allowing for advanced resume analysis and skills advising. NLP techniques such as sentiment analysis and named entity recognition enable the system to grasp the subtleties and context within resumes, resulting in more precise skill identification and competency matching.

By implementing this approach, companies can save valuable time and resources while ensuring better alignment between candidate skills and job requirements. Moreover, the potential for increased employee retention underscores the long-term benefits of adopting automated resume analysis and skills advising systems. Looking ahead, the future scope for this technology appears promising, with ongoing advancements in NLP and machine learning paving the way for even more sophisticated resume analysis and skills recommendation capabilities. With careful implementation and continual monitoring, organizations can leverage these innovations to gain a competitive edge in the dynamic landscape of modern recruitment. In addition to revolutionizing recruitment processes, automated resume analysis and skills suggestion using NLP not only enhances efficiency but also fosters a more inclusive hiring environment by minimizing bias. By objectively evaluating candidate qualifications based on structured data, this approach promotes fairness and diversity in candidate selection. Furthermore, the integration of advanced algorithms ensures adaptability to evolving job market trends, allowing companies to stay agile and responsive to changing hiring needs.

REFERENCES

REFERENCES

- Tumula Mani Harsha, Gangaraju Sai Moukthika, Dudipalli Siva Sai, Mannuru Naga Rajeswari Pravallika, Satish Anamalamudi, MuraliKrishna Enduri, “Automated Resume Screener using Natural Language Processing(NLP)”, 6th International Conference on Trends in Electronics and Informatics (ICOEI), 2022
- M. Alamelu, D. Sathish Kumar, R. Sanjana, J. Subha Sree, A. Sangeerani Devi, D. Kavitha, “Resume Validation and Filtration using Natural Language Processing”, 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), 2022
- M.F. Mridha, Rabeya Basri, Muhammad Mostafa Monowar, Md. Abdul Hamid, “A Machine Learning Approach for Screening Individual’s Job Profile Using Convolutional Neural Network”, International Conference on Science & Contemporary Technologies (ICSCT), 2021
- Pradeep Kumar Roy, Vellore Institute of Technology, 2019. A Machine learning approach for automation of resume recommendation system, ICCIDS 2019. 10.1016/j.procs.2020.03.284.
- Thimma Reddy Kalva, Utah State University, 2013. SkillFinder: Automated Job-Resume Matching system.
- Yong Luo, Nanyang Technological University, 2018. A Learning- Based Framework for automatic resume quality assessment, arXiv:1810.02832v1 cs.IR].

GITHUB LINK

https://github.com/Karthikchennuri/Resume_analysis

