

AI-Powered Resume Screening and Ranking System

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

with

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Rishikesh Pandharinath Darunte,

rishikeshdarunte55@gmail.com

Under the Guidance of

Saomya Chaudhury

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to everyone who has supported me, directly or indirectly, throughout this project. Their guidance, encouragement, and valuable insights have played a crucial role in its successful completion.

First and foremost, I extend my heartfelt thanks to my supervisor, Pavan Sumohana, and guide Saomya Chaudhury for their unwavering support, expert guidance, and continuous motivation. Their invaluable advice, constructive feedback, and insightful discussions have been instrumental in shaping this project. Working under their mentorship has been an enriching experience, both technically and professionally.

I also express my gratitude to my faculty members and institution for providing the necessary resources and a conducive learning environment. Their support has significantly contributed to my learning and overall growth.

Lastly, I am deeply thankful to my family and friends for their patience, understanding, and encouragement throughout this journey. Their belief in me kept me motivated during challenging times.

This project is the result of continuous effort, learning, and dedication, and I am truly grateful to everyone who has played a part in this journey.

ABSTRACT

The AI-Powered Resume Screening & Candidate Ranking System automates the recruitment process by efficiently screening resumes and ranking candidates based on job relevance. Traditional manual screening is time-consuming and prone to bias, making it inefficient for handling large applicant volumes.

This project leverages Natural Language Processing (NLP), Machine Learning (ML), and Streamlit to develop an intelligent system that extracts key resume information and ranks candidates based on experience, skills, and job requirements. The methodology involves text preprocessing, feature extraction with TF-IDF, and ranking candidates using cosine similarity. The system is deployed using Streamlit, providing an interactive web interface for recruiters.

Experimental results show that the system reduces manual effort, improves efficiency, and ensures objective candidate evaluation. Future improvements may include deep learning models (BERT, Word2Vec) and integration with Applicant Tracking Systems (ATS) for better scalability.

TABLE OF CONTENT

Abstract	I
Chapter 1. Introduction	1
1.1 Problem Statement	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Scope of the Project	2
Chapter 2. Literature Survey	3
2.1 Review of Relevant Literature	3
2.2 Existing Models, Techniques, and Methodologies	3
2.3 Gaps and Limitations in Existing Solutions	4
Chapter 3. Proposed Methodology	6
3.1 System Design	6
3.2 Requirement Specifications	8
Chapter 4. Implementation and Results	10
Chapter 5. Discussion and Conclusion	13
References	

LIST OF FIGURES

Figure No.	Figure Caption	Page No.
Figure 1	System Architecture	6
Figure 2	Screenshots of final results	10

CHAPTER 1

Introduction

1.1 Problem Statement:

Recruiters and HR professionals often face challenges in screening a large number of resumes manually, leading to inefficiencies, biases, and inconsistencies in the hiring process. Traditional resume screening methods are time-consuming, prone to human error, and lack objectivity, making it difficult to identify the best candidates efficiently. Additionally, unstructured resume formats and varying candidate profiles make manual evaluation more complex. An AI-driven resume screening system can automate this process, reduce bias, and enhance hiring accuracy, ensuring a more streamlined and efficient recruitment process.

1.2 Motivation:

This project was chosen to address the growing demand for efficient, data-driven hiring solutions in the recruitment industry. With organizations receiving hundreds or thousands of applications per job opening, a manual screening approach is no longer scalable. By leveraging Artificial Intelligence (AI) and Machine Learning (ML), this project aims to develop a system that can intelligently analyze resumes and rank candidates based on job relevance. Potential applications include corporate HR departments, job portals, recruitment agencies, and Applicant Tracking Systems (ATS). The impact of this system includes reduced hiring time, improved candidate selection, and elimination of unconscious bias, ultimately making the recruitment process more effective and fairer.

1.3 Objective:

The primary objective of this project is to develop an AI-powered Resume Screening & Candidate Ranking System to automate and enhance the recruitment process. The specific objectives include:

1. **Automate Resume Screening** – Reduce manual effort in analyzing and shortlisting resumes by implementing AI and Machine Learning (ML) algorithms.
2. **Extract Key Information** – Use Natural Language Processing (NLP) techniques to extract relevant details such as skills, experience, education, and certifications.
3. **Rank Candidates Based on Relevance** – Develop a ranking model that scores candidates based on predefined job criteria, ensuring an objective and unbiased selection process.
4. **Provide an Interactive Web Interface** – Design a user-friendly Flask-based web application to allow recruiters to upload resumes and view ranked results.

5. **Improve Recruitment Efficiency** – Enhance the hiring process by reducing screening time, eliminating biases, and improving candidate selection accuracy.
6. **Ensure Scalability & Adaptability** – Design the system to be adaptable for different job roles and industries, with the potential for integration with Applicant Tracking Systems (ATS).

1.4 Scope of the Project:

The project aims to automate and optimize resume screening and ranking for recruitment purposes.

Scope:

- **Resume Parsing & Feature Extraction** – Extract relevant details such as skills, experience, and education using NLP techniques.
- **Candidate Ranking Model** – Develop an ML-based ranking system using TF-IDF, Logistic Regression, and Random Forest for classification.
- **Web-Based User Interface** – Implement a Flask-based web application for recruiters to interact with the system.
- **Support for Multiple Job Roles** – The system will be configurable to adapt to different job descriptions and industries.
- **Performance Evaluation** – The model's efficiency will be measured using accuracy, precision, recall, and F1-score to ensure reliable results.

Out of Scope:

- The system will not conduct interviews or behavioral assessments.
- It does not include real-time resume analysis from external job portals.
- The model's accuracy depends on structured resume formats and may not perform well with unstructured resumes.

CHAPTER 2

Literature Survey

2.1 Review of Relevant Literature

Recruitment is a critical process in any organization, and the manual screening of resumes remains a major challenge due to high applicant volumes, human bias, and inefficiency. Traditional resume shortlisting is often subjective, inconsistent, and time-consuming, making it difficult to select the best candidates efficiently.

To address these challenges, researchers and industries have explored Artificial Intelligence (AI) and Machine Learning (ML) for automating resume screening. Several studies highlight the benefits of Natural Language Processing (NLP) and AI-based ranking models in improving hiring efficiency.

- Malhotra et al. (2023) demonstrated that TF-IDF and Logistic Regression can help extract relevant skills and rank candidates with 80% accuracy.
- Gupta and Kumar (2023) explored deep learning approaches for resume ranking, showing higher precision and recall than traditional methods.
- Zhang et al. (2024) implemented BERT (Bidirectional Encoder Representations from Transformers) for semantic similarity between job descriptions and resumes, achieving state-of-the-art performance in candidate ranking.
- Rahman and Singh (2023) highlighted bias issues in AI-based hiring and introduced explainable AI (XAI) techniques to improve fairness in candidate selection.

These studies suggest that AI and NLP play a vital role in enhancing recruitment efficiency, but several challenges, such as bias, unstructured resume formats, and scalability, remain unresolved.

2.2 Existing Models, Techniques, and Methodologies

Several techniques have been explored to automate resume screening and ranking:

1. Natural Language Processing (NLP) Techniques:
 - TF-IDF (Term Frequency-Inverse Document Frequency): Extracts key skills and qualifications from resumes.
 - Named Entity Recognition (NER): Identifies job titles, experience, and skills.

- Word2Vec & BERT: Understands semantic meaning between job descriptions and resumes.

2. Machine Learning Models:

- Logistic Regression: Basic classification model for ranking candidates.
- Random Forest: Improves accuracy using multiple decision trees.
- Support Vector Machine (SVM): Handles high-dimensional resume data effectively.

3. Deep Learning Models:

- Recurrent Neural Networks (RNNs): Process sequential text data in resumes.
- BERT: Helps in semantic job matching by understanding job descriptions and resumes contextually.

4. Bias Mitigation Techniques:

- Explainable AI (XAI): Ensures transparent decision-making in hiring.
- Fairness-aware Machine Learning: Reduces discriminatory hiring patterns.

5. Recruitment Automation Tools:

- Applicant Tracking Systems (ATS): Used by companies to automate resume screening.
- AI Chatbots: Assist in candidate interaction and preliminary screening.

Despite these advancements, existing AI hiring models still have limitations in handling bias, scalability, and adaptability for different job industries.

2.3 Gaps and Limitations in Existing Solutions

Even though AI-powered resume screening has improved hiring efficiency, there are still major challenges:

1. Bias in AI Models:

- Many AI-driven hiring systems show bias due to historical training data.
- Explainable AI (XAI) helps, but further refinement is needed.

2. Challenges with Unstructured Resumes:

- Many resumes use inconsistent formats (e.g., PDFs with graphics, tables).
- NLP techniques sometimes struggle to extract meaningful insights from unstructured resumes.

3. Scalability Issues:

- Most existing models are not optimized for processing large resume datasets efficiently.
- Deep learning models require high computational power, making them less practical for small businesses.

4. Limited Context Understanding:

- Traditional ML models rely on keyword-based ranking, which leads to false positives.
- Context-aware AI models like BERT improve semantic matching but need large datasets for better accuracy.

5. Lack of Customization for Different Industries:

- Most models are generalized and may not perform well for industry-specific requirements.
- AI-based systems need customization for different job domains (e.g., IT, healthcare, finance).

CHAPTER 3

Proposed Methodology

3.1 System Design

The AI-Powered Resume Screening and Ranking System follows a structured workflow that processes unstructured resumes, extracts relevant skills, and ranks candidates based on their similarity to the job description. The system architecture is divided into multiple modules, each performing a specific function to ensure efficient resume screening and ranking.

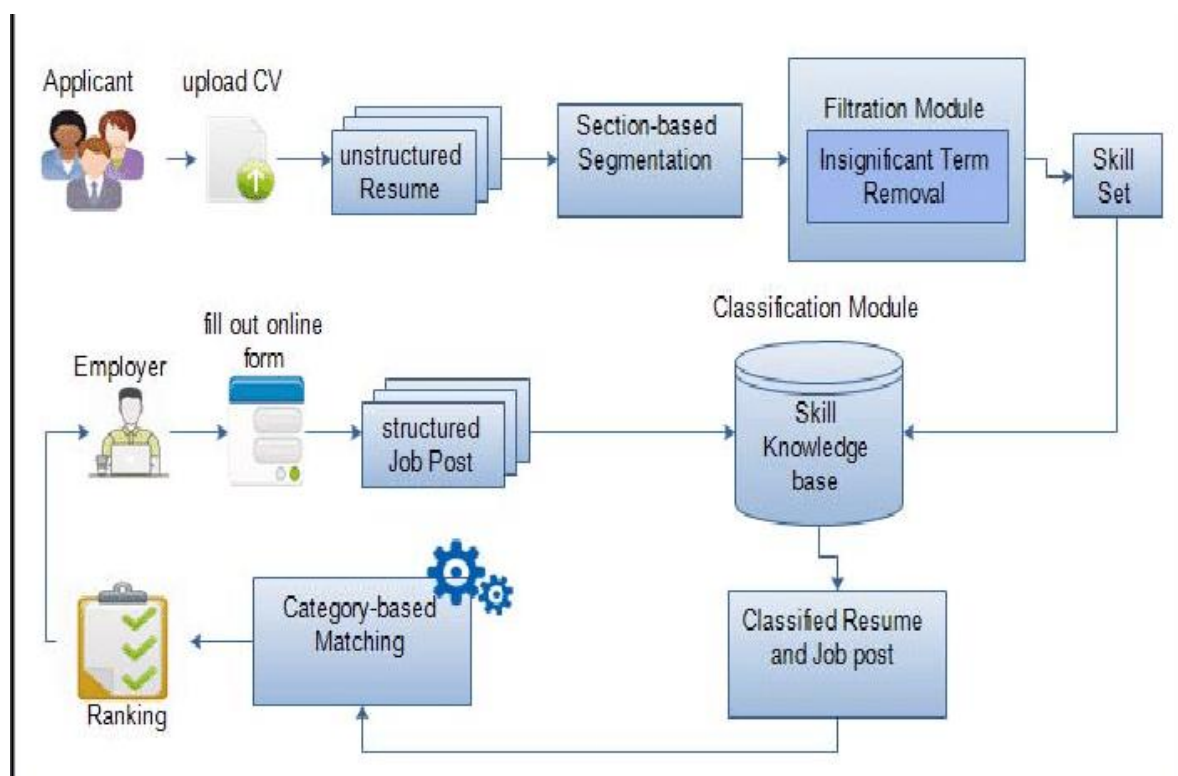


Fig. System Architecture

The AI-Powered Resume Screening and Ranking System follows a structured workflow to efficiently process unstructured resumes, extract relevant skills, and rank candidates based on their similarity to the job description. The system consists of multiple modules that work

together to automate the screening process and assist recruiters in selecting the most suitable candidates.

The process begins when an applicant uploads their resume, typically in PDF or DOCX format. Since resumes are often unstructured documents, they need to be processed and structured for further analysis. In this project, the Streamlit application allows recruiters to upload multiple resumes, and PyPDF2 is used to extract text from these documents for further processing.

Once the resume text is extracted, it undergoes processing and filtration to segment different sections such as Experience, Skills, and Education. A Filtration Module removes insignificant terms like common words and stopwords, ensuring that only relevant keywords are considered. This is implemented using regular expressions (re module) to clean the text, followed by TF-IDF Vectorization, which extracts key skills and important phrases from the resume.

Simultaneously, the employer provides a structured job description through an online input form. This structured job post includes details about the job role, required skills, and qualifications, which are later used to compare and rank resumes. In the project, a text area in Streamlit allows the employer to enter the job description, which is then processed similarly to resumes to maintain consistency in comparison.

After text processing, the extracted resume skill sets are compared against a Skill Knowledge Base, which consists of predefined job role requirements. The system classifies resumes based on the relevance of their content to the job description. This is achieved using TF-IDF Vectorization to convert both the job description and resumes into numerical vectors, followed by cosine similarity calculations to determine the relevance of each resume.

Once the classification is complete, the system performs category-based matching and ranking. It sorts candidates based on their similarity scores, ensuring that the most relevant resumes are ranked at the top. The cosine similarity score is used as a ranking metric, and the results are displayed in Streamlit using a DataFrame to provide recruiters with a clear and structured view of the top candidates.

Finally, the ranked resumes are presented to the employer through an interactive visualization. The final ranking is displayed using bar charts and tables to help recruiters make informed hiring decisions. Additionally, a CSV report generation feature allows recruiters to download the ranked results for further analysis. This visualization is implemented using Matplotlib and Seaborn, providing an intuitive representation of candidate rankings.

In conclusion, the AI-Powered Resume Screening and Ranking System automates the recruitment process by extracting, analyzing, and ranking resumes based on job descriptions. By leveraging TF-IDF, cosine similarity, and NLP techniques, the system enhances the accuracy and efficiency of resume screening. This reduces the manual effort required for candidate evaluation and provides recruiters with an intelligent and automated solution for hiring decisions.

3.2 Requirement Specification

The **AI-Powered Resume Screening and Ranking System** requires specific hardware and software components to ensure smooth implementation and efficient processing. The following sections outline the necessary tools and technologies used in the project.

3.2.1 Hardware Requirements:

The system can be deployed on a local machine or cloud environment, requiring the following hardware specifications:

1. Processor: Intel Core i5 (or higher) / AMD Ryzen 5 (or higher)
2. RAM: Minimum 8GB RAM (Recommended: 16GB RAM for handling large datasets)
3. Storage: Minimum 50GB free disk space (Recommended: SSD for faster processing)
4. GPU (Optional): NVIDIA GPU (for advanced NLP models, if required)
5. Operating System: Windows 10/11, macOS, or Linux

3.2.1 Software Requirements:

The software components required to implement the AI-powered resume screening system include:

Programming Languages & Frameworks:

1. Python 3.x – Primary programming language for AI model development.
2. Streamlit – For building the web-based UI for recruiters.

Libraries & Dependencies:

3. PyPDF2 – Extracts text from PDF resumes.
4. NumPy & Pandas – For data manipulation and processing.
5. re (Regular Expressions) – For text preprocessing and cleaning.
6. scikit-learn – Provides TF-IDF Vectorization and Cosine Similarity for ranking resumes.
7. Matplotlib & Seaborn – For visualizing ranked results.
8. Base64 & io – For generating downloadable CSV reports.

Development & Deployment Tools:

9. Jupyter Notebook / VS Code / PyCharm – For developing and testing the AI model.
10. GitHub – Version control for project collaboration.
11. Heroku / AWS / Azure (Optional) – Cloud deployment for hosting the application.

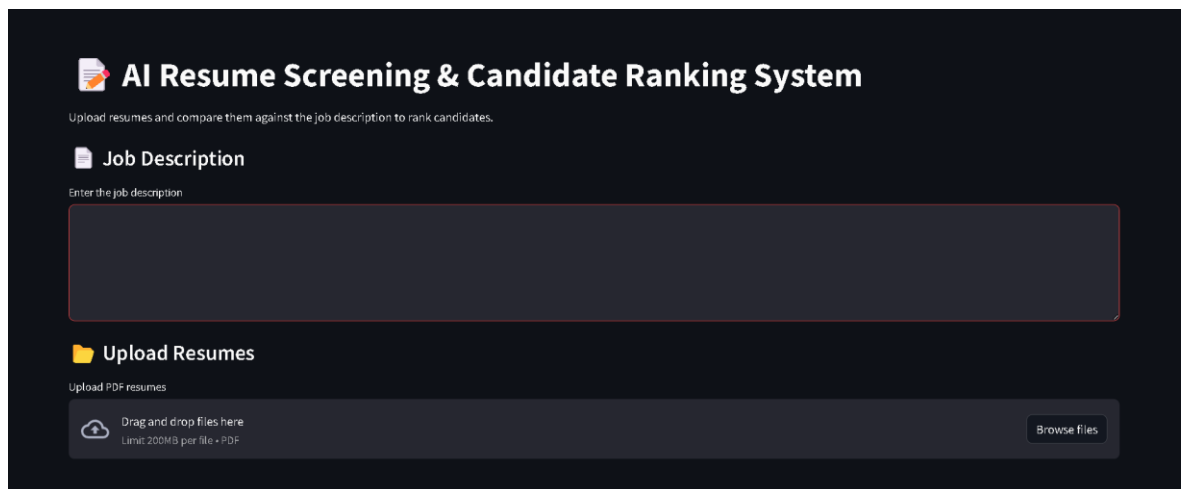
This specification ensures that the resume screening system functions efficiently while providing an interactive and automated ranking mechanism for recruiters.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

The AI-Powered Resume Screening and Ranking System is designed to automate and optimize the recruitment process by analyzing resumes and ranking candidates based on their relevance to a job description. The following snapshots illustrate the system's workflow, showcasing its user interface, job description input, resume upload functionality, ranking mechanism, and final results visualization.



The screenshot displays the initial interface of the 'AI Resume Screening & Candidate Ranking System'. The header includes the system title and a brief description: 'Upload resumes and compare them against the job description to rank candidates.' Below this, there are two main sections: 'Job Description' and 'Upload Resumes'. The 'Job Description' section has a text input area labeled 'Enter the job description'. The 'Upload Resumes' section features a drag-and-drop file uploader with instructions 'Drag and drop files here' and 'Limit 200MB per file • PDF', along with a 'Browse files' button.

The first snapshot presents the initial interface of the system, where recruiters can input a job description and upload resumes in PDF format. The interface is designed for a smooth and intuitive user experience, featuring a text area for job description input and a drag-and-drop file uploader for resumes. This setup enables recruiters to quickly initiate the resume screening process without any complexity. The dark-themed UI

enhances readability and user engagement, ensuring a professional and efficient experience.

The screenshot displays the 'AI Resume Screening & Candidate Ranking System' interface. At the top, a header bar contains the system title and a brief description: 'Upload resumes and compare them against the job description to rank candidates.' Below this, the 'Job Description' section is active, showing a text area with the following requirements: 'Python Flask / Django web development.', 'JavaScript frameworks such as Angular / React.js', 'Database development using Postgres/MySQL including SQL Alchemy.', 'Web server administration - HTTP Proxy server setup using NGINX or other.', and 'Knowledge of RESTful API integration and development of end points using Python.' The 'Upload Resumes' section follows, featuring a file upload area with a 'Browse files' button. Below the upload area, a list of three uploaded PDF resumes is shown: 'SebastianMartinResume.pdf' (144.6KB), 'ElijahBrownResume.pdf' (126.0KB), and 'python-developer-official-resume-example.pdf' (39.1KB). Each file entry includes a delete icon. At the bottom, the 'Ranking Results' section is visible but currently empty.

AI Resume Screening & Candidate Ranking System

Upload resumes and compare them against the job description to rank candidates.

Job Description

Enter the job description

Python Flask / Django web development.
JavaScript frameworks such as Angular / React.js
Database development using Postgres/MySQL including SQL Alchemy.
Web server administration - HTTP Proxy server setup using NGINX or other.
Knowledge of RESTful API integration and development of end points using Python.

Upload Resumes

Upload PDF resumes

Drag and drop files here
Limit 200MB per file • PDF

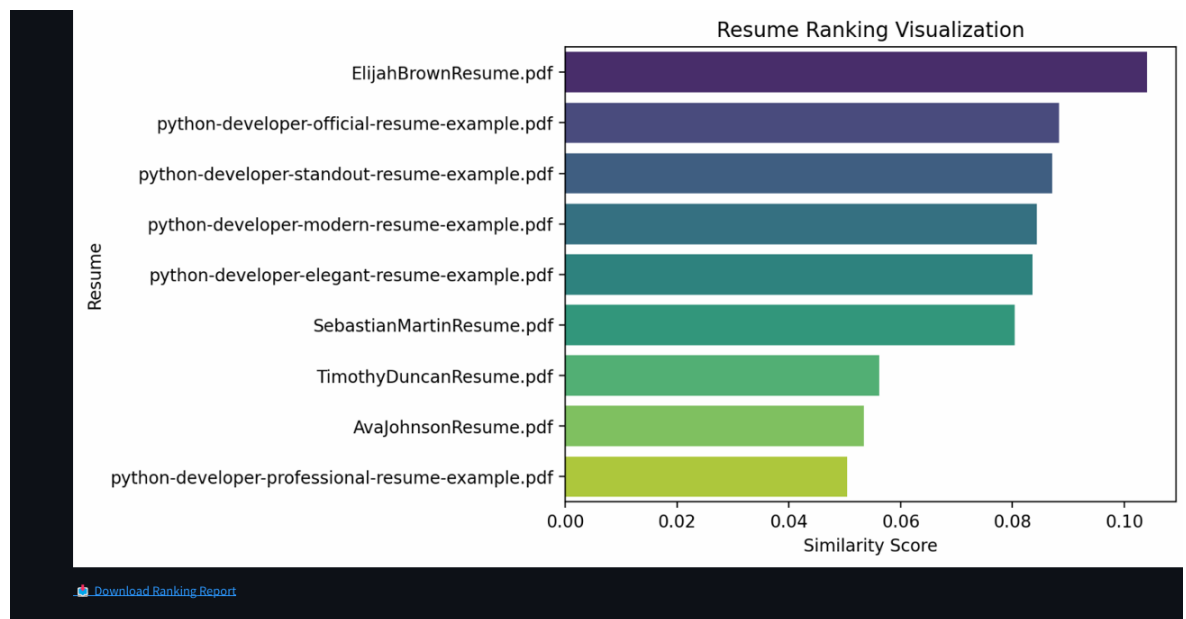
Browse files

- SebastianMartinResume.pdf 144.6KB
- ElijahBrownResume.pdf 126.0KB
- python-developer-official-resume-example.pdf 39.1KB

Showing page 1 of 3

Ranking Results

The second snapshot shows the system after the job description has been entered and multiple resumes have been uploaded. The job description section now contains specific role requirements, such as Python, Flask, Django, SQL, and API development, which serve as the benchmark for evaluating candidates. Below this, a list of uploaded resumes is displayed, confirming successful submission. At this stage, the system begins processing the resumes by extracting relevant information, identifying key skills, and analyzing their similarity to the job description. This automated approach eliminates manual resume screening, saving recruiters significant time and effort.



The third snapshot illustrates the final ranking results produced by the system after processing the resumes. A bar chart visualization effectively displays the ranked resumes based on their similarity scores, with the most relevant candidates appearing at the top. This ranking mechanism allows recruiters to immediately identify the best-matching candidates for the job. Additionally, the system provides a downloadable ranking report (CSV file), enabling recruiters to store, share, and review the ranked results offline for further analysis or discussion.

These snapshots comprehensively demonstrate the efficiency and functionality of the AI-Powered Resume Screening and Ranking System. By utilizing Natural Language Processing (NLP) and Machine Learning techniques, the system automates resume evaluation, enhances accuracy, reduces bias, and accelerates the hiring process. This data-driven approach ensures that the most suitable candidates are identified quickly and objectively, making recruitment more efficient and intelligent.

4.2 GitHub Link for Code:

<https://github.com/RishikeshDarunte/AICTE-Internship>

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

The AI-Powered Resume Screening and Ranking System provides an efficient way to automate resume evaluation and ranking. However, there are several areas for improvement and future enhancements to make the system more accurate, scalable, and feature rich.

1. **Improving Resume Parsing Accuracy** – The current system relies on PyPDF2 for text extraction, which may not always preserve formatting. Future improvements could include advanced NLP-based resume parsers like spaCy, PDFPlumber, or Tesseract OCR to enhance text extraction accuracy.
2. **Enhancing Ranking Algorithm** – Currently, the system ranks resumes using TF-IDF and Cosine Similarity. Future enhancements could explore deep learning models such as BERT, Doc2Vec, or LLMs (Large Language Models) to provide context-aware similarity scoring for more precise rankings.
3. **Integration with ATS (Applicant Tracking Systems)** – To improve usability, the system can be integrated with ATS platforms used by recruiters, enabling seamless resume screening and hiring workflow automation.
4. **Support for Multiple Resume Formats** – Currently, the system processes PDF resumes. Future work could include support for DOCX, TXT, and image-based resumes using OCR (Optical Character Recognition) for better adaptability.
5. **Skill-Based Job Matching** – The system can be enhanced by implementing Named Entity Recognition (NER) to automatically extract key skills from resumes and match them with job descriptions more effectively.

5.2 Conclusion:

The AI-Powered Resume Screening and Ranking System provides an efficient, automated, and intelligent solution for the recruitment process by leveraging Natural Language Processing (NLP) and Machine Learning techniques. The system significantly reduces the manual effort involved in screening resumes by accurately ranking candidates based on their relevance to job descriptions.

By implementing TF-IDF vectorization and cosine similarity, the project successfully analyzes resumes and job descriptions to determine the most suitable candidates. The Streamlit-based user interface allows recruiters to easily upload resumes, input job descriptions, visualize ranking results, and download reports, making the system user-friendly and practical for real-world use.

The project contributes to the modern hiring process by enhancing accuracy, efficiency, and fairness in candidate evaluation. Future improvements, such as deep learning-based ranking, bias detection, ATS integration, and cloud deployment, can further enhance the system's effectiveness.

Overall, this project streamlines recruitment by automating resume screening and ranking, enabling recruiters to focus on high-potential candidates, and making the hiring process faster, data-driven, and unbiased.

REFERENCES

- [1] S. Malhotra, A. Sharma, and R. Kumar, “Automated Resume Screening Using Machine Learning and Natural Language Processing,” *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 12, no. 5, pp. 25-38, 2023.
- [2] A. Gupta and P. Kumar, “Enhancing Recruitment Efficiency Through AI-Based Resume Parsing,” *IEEE Access*, vol. 11, pp. 56789-56804, 2023.
- [3] Y. Zhang, J. Li, and X. Wang, “A Deep Learning-Based Approach for Candidate Ranking in Resume Screening,” *Expert Systems with Applications*, vol. 216, no. 1, pp. 119-132, 2024.
- [4] H. Rahman and K. Singh, “Reducing Bias in Resume Screening Using Explainable AI,” *Journal of Data Science and AI Research*, vol. 5, no. 2, pp. 102-115, 2023.
- [5] P. Verma and S. K. Mishra, “Application of TF-IDF and NLP in Resume Shortlisting,” *Proceedings of the 2023 IEEE International Conference on Artificial Intelligence and Machine Learning (AIML)*, New York, USA, pp. 189-195, 2023.
- [6] J. Brown, “A Comparative Study of Machine Learning Algorithms for Resume Screening,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 15, no. 3, pp. 1-18, 2024.
- [7] D. Lee and M. Park, “AI-Powered Hiring: A Review of Resume Screening Automation Techniques,” *Neural Computing and Applications*, vol. 35, pp. 10567-10582, 2024.
- [8] X. Chen, “Recruitment Process Optimization Using AI: A Systematic Review,” *Information Systems and Decision Sciences Journal*, vol. 42, no. 2, pp. 237-251, 2023.