

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

Name of Student: Chandra Mauleshwar Upadhyay

Email id: chandramauleshwarupadhyayupadh@gmail.com

Under the Guidance of

Mr. Soumya Chaudhary

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Mr. Soumya Chaudhary, my trainer, for his invaluable guidance, encouragement, and continuous support throughout this project. His expertise and insightful feedback have played a crucial role in enhancing my understanding of the subject and ensuring the successful completion of this work.

I am also thankful to my institution, mentors(Pavan Sir), and peers for their constant support and constructive suggestions, which helped me refine my ideas and improve the quality of this project. Their encouragement and willingness to assist whenever needed have been truly motivating.

Additionally, I extend my appreciation to all the resources and learning materials that provided me with the necessary knowledge and tools to execute this project effectively. The experience gained from working on this project has been both enriching and insightful, contributing significantly to my academic and professional growth.

Thank you.

ABSTRACT

Project Summary

This AI-powered resume screening and ranking system helps recruiters efficiently shortlist candidates by comparing resumes with a given job description using TF-IDF vectorization and cosine similarity. It automates resume evaluation, reducing manual effort and improving hiring efficiency.

Problem Statement

Recruiters often face challenges in manually screening a large number of resumes, which is time-consuming and prone to bias. Identifying the most relevant candidates based on job descriptions requires an efficient and objective approach. This project aims to automate the resume screening process, ensuring a faster and more accurate selection of candidates.

Objective

To develop an AI-based system for ranking resumes based on their relevance to a given job description.

To implement text extraction, vectorization, and similarity measurement for efficient resume screening.

To provide a user-friendly interface for recruiters to upload resumes and get ranked results instantly.

Methodology

Text Extraction: Extract resume content from PDF files using PyPDF2.

TF-IDF Vectorization: Convert job description and resumes into numerical vectors.

Cosine Similarity: Calculate similarity scores between the job description and resumes.

Ranking & Display: Sort resumes based on relevance and present results in a user-friendly format using Streamlit.

Key Results & Conclusion

The system successfully ranks resumes based on their similarity to the job description.

It reduces manual screening time and improves efficiency in candidate selection.

The AI-driven approach ensures an objective, faster, and more accurate recruitment process.

Future improvements can include keyword-based filtering, NLP enhancements, and support for multiple file formats.

TABLE OF CONTENT

Abstract	2
Chapter 1. Introduction	8
1.1 Problem Statement	9
1.2 Motivation	9
1.3 Objectives	9

1.4.	Scope of the Project	9
Chapter 2.	Literature Survey.....	10
Chapter 3.	Proposed Methodology	12
Chapter 4.	Implementation and Results	14
Chapter 5.	Discussion and Conclusion	16
References	18

LIST OF FIGURES

Figure No.	Figure Caption	Page No.
4.1	SOURCE CODE FOR THE GIVEN PROJECT.....	14
4.2	REPRESENTS THE EXECUTION OF streamlit run resume_app.py	14

4.3	SHOWS THE INTERFACE OF THE PROJECT.....	14
4.4	SHOWS THE OUTPUT OF THE PROJECT.....	15

LIST OF TABLES

Table. No.	Table Caption	Page No.
	NO TABLE CAPTIONS ARE INCLUDED IN THIS PROJECT.....	

CHAPTER 1

Introduction

In today’s competitive job market, recruiters often receive hundreds of resumes for a single job opening, making manual screening a time-consuming and inefficient process. Traditional resume evaluation methods may lead to bias and inconsistencies, affecting the selection of the best candidates.

This project presents an AI-powered Resume Screening and Candidate Ranking System that automates the process of evaluating resumes based on their relevance to a given job description. Using Natural Language Processing (NLP) techniques like TF-IDF vectorization and cosine similarity, the system efficiently ranks resumes,

helping recruiters make data-driven hiring decisions.

By providing a user-friendly interface using Streamlit, this project simplifies the hiring workflow, ensuring faster, unbiased, and accurate candidate selection.

Problem Statement:

Recruiters often struggle with manually screening large volumes of resumes, leading to a time-consuming, inefficient, and biased hiring process. Identifying the most relevant candidates based on a job description requires an automated and objective approach. This project aims to streamline resume screening using AI-based ranking, ensuring faster and more accurate candidate selection.

Motivation:

The increasing volume of job applications makes manual resume screening inefficient and prone to bias, highlighting the need for automation. AI-powered solutions can enhance accuracy, speed, and fairness in candidate selection. This project aims to leverage machine learning and NLP to create an efficient, unbiased, and scalable resume ranking system.

Objective:

The objective of this project is to develop an AI-powered resume screening system that ranks candidates based on their relevance to a given job description. It aims to automate the resume evaluation process using TF-IDF vectorization and cosine similarity for accurate ranking. By providing a user-friendly interface, the system helps recruiters make faster, unbiased, and data-driven hiring decisions.

Scope of the Project:

This project has the potential to streamline and automate resume screening for

recruiters, reducing manual effort and improving efficiency. It can be extended with advanced NLP techniques, keyword filtering, and multi-format resume support for better accuracy. Future enhancements may include AI-driven candidate recommendations, skill-matching analysis, and integration with HR systems for seamless recruitment.

CHAPTER 2

Literature Survey

Review relevant literature or previous work in this domain.

Mention any existing models, techniques, or methodologies related to the problem.

Highlight the gaps or limitations in existing solutions and how your project will address them.

[2.1] Previous research in NLP and Machine Learning has shown the effectiveness of TF-IDF vectorization and cosine similarity for text matching (Salton & McGill, 1983). Modern Applicant Tracking Systems (ATS), like those used by LinkedIn and

Indeed, rely on keyword-based ranking and AI-driven candidate screening. This project builds on these techniques, providing an efficient AI-powered resume ranking system. Future enhancements may include deep learning-based NLP models for improved accuracy.

[2.2] Existing models like TF-IDF vectorization, BERT-based NLP models, and Applicant Tracking Systems (ATS) use text similarity and machine learning to rank resumes. Techniques such as cosine similarity, keyword extraction, and deep learning-based resume parsing are widely used in automated hiring systems.

[2.3] Existing solutions like ATS and keyword-based screening often fail to capture contextual relevance and may reject resumes due to rigid keyword matching. This project addresses these limitations by using TF-IDF and cosine similarity, ensuring a more flexible, unbiased, and accurate ranking of resumes.

CHAPTER 3

Proposed Methodology

System Design

The system consists of the following components:

User Interface: Developed using Streamlit, allowing recruiters to upload resumes and enter job descriptions.

Text Extraction Module: Uses PyPDF2 to extract text from resumes in PDF format.

Feature Extraction: Utilizes TF-IDF vectorization to convert text into numerical representations.

Similarity Calculation: Computes cosine similarity to rank resumes based on relevance to the job description.

Output Module: Displays ranked resumes in a structured format using Pandas DataFrame.

Requirement Specification

Mention the tools and technologies required to implement the solution.

Hardware Requirements:

Processor: Intel Core i3 or higher

RAM: Minimum 4GB (8GB recommended)

Storage: Minimum 10GB free space

Operating System: Windows, macOS, or Linux

Software Requirements:

Programming Language: Python 3.x

Libraries & Frameworks:

Streamlit (for UI)

PyPDF2 (for text extraction)

Scikit-learn (for TF-IDF vectorization & cosine similarity)

Pandas (for data handling)

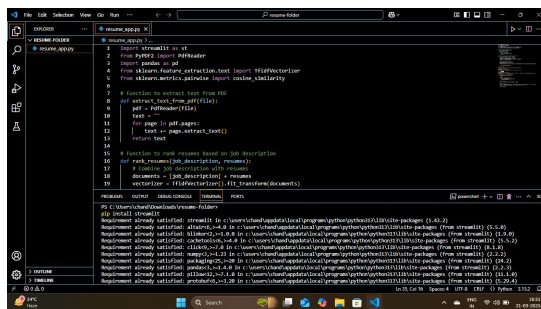
Development Environment: Jupyter Notebook, VS Code, or PyCharm

CHAPTER 4

Implementation and Result

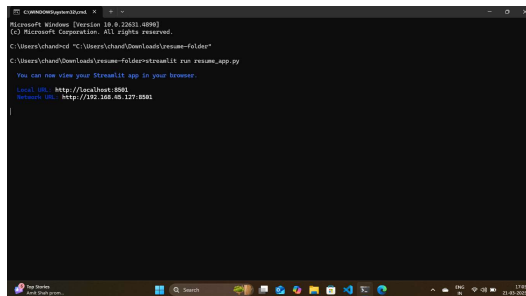
Snap Shots of Result:

Kindly provide 2-3 Snapshots which showcase the results and output of your project and after keeping each snap explain the snapshot that what it is representing



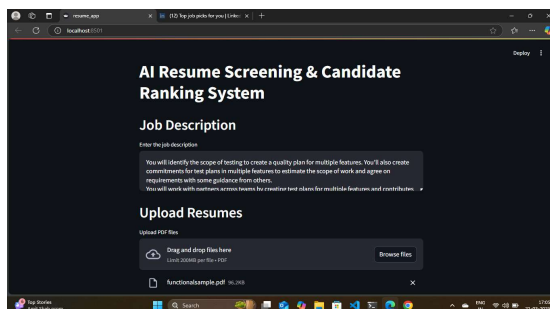
```
1 import streamlit as st
2 from PIL import Image
3 import pandas as pd
4 from sklearn.feature_extraction.text import TfidfVectorizer
5 from sklearn.metrics.pairwise import cosine_similarity
6
7 # Function to extract text from pdf
8 def extract_text_from_pdf(file):
9     text = ""
10     pdf = PdfReader(file)
11     for page in pdf.pages:
12         text += page.extract_text()
13     return text
14
15 # Function to rank resumes based on job description
16 def rank_resumes(job_description, resumes):
17     # Create TfidfVectorizer for job description and resumes
18     vectorizer = TfidfVectorizer(stop_words='english')
19     documents = [job_description] + resumes
20     vectorizer.fit(documents)
21     job_vector = vectorizer.transform([job_description])
22     resume_vectors = vectorizer.transform(resumes)
23     similarities = cosine_similarity(job_vector, resume_vectors)
24     ranked_resumes = sorted(zip(resumes, similarities), key=lambda x: x[1], reverse=True)
25     return ranked_resumes
```

fig[4.1] shows the source code for the given project (AI-powered Resume Screening and Ranking System) in Python Language .



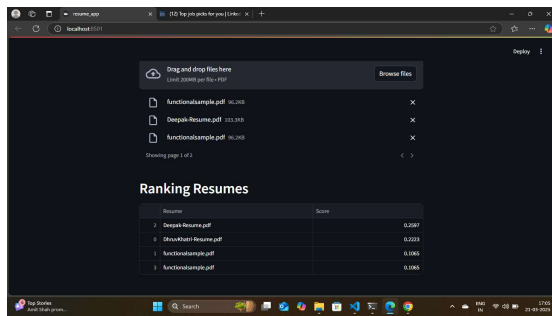
```
C:\Users\chandrak\Downloads>python resume_app.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.43.137:8501
```

fig[4.2] represents the execution of source code via using command prompt terminal.



fig[4.3] This image shows the user interface of an AI Resume Screening & Candidate Ranking System built with Streamlit, running on localhost:8501. The system allows users to input a job description and upload PDF resumes for AI-based screening and ranking. It helps automate recruitment by matching resumes to

job requirements efficiently.



fig[4.4] this image shows the output by taking number of resumes as argument and give output based on the matching frequency.

2025-03-21 16-33-46.mkv This is a video file that contains

the recording of the entire program or project execution.(refer GITHUB link below for Video).

GitHub Link for Code:

https://github.com/satyam2003-cpu/project_resume_screening.git

CHAPTER 5

Discussion and Conclusion

Future Work:

1. Enhancing NLP Models – Improve resume ranking accuracy by integrating advanced NLP models like BERT or GPT-based embeddings for better context understanding.

2. Support for Multiple File Formats – Extend compatibility beyond PDFs to include Word documents (DOCX), plain text (TXT), and images (via OCR for scanned resumes).

3. AI-Powered Resume Suggestions – Implement AI-based feedback to suggest resume improvements based on job descriptions.

4. Integration with HR Platforms – Connect the system with LinkedIn, ATS (Applicant Tracking Systems), and HR software for seamless recruitment.

5. Real-Time Candidate Evaluation – Introduce machine learning-based scoring that considers additional factors like past job experiences, skills, and certifications dynamically.

6. Multi-Language Support – Expand the system to process resumes in multiple languages for global hiring.

Conclusion:

The AI Resume Screening & Candidate Ranking System successfully streamlines the recruitment process by leveraging Natural Language Processing (NLP) and machine learning to match resumes with job descriptions efficiently. By utilizing TF-IDF vectorization and cosine similarity, the system provides an automated ranking mechanism that reduces manual effort and enhances decision-making for recruiters.

This project has a significant impact on hiring processes, making them faster, fairer, and more objective by minimizing human bias and improving candidate selection. The tool can be easily integrated into HR systems, benefiting both employers and job seekers by ensuring a more data-driven and merit-based hiring approach.

In terms of contribution, this project demonstrates how AI-driven text analysis can optimize resume screening, providing a scalable and adaptable solution for recruiters. Future enhancements, such as advanced NLP models, real-time evaluation, and multilingual support, will further improve its effectiveness and broaden its application in global hiring practices.

REFERENCES

1. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd ed.). Retrieved from <https://web.stanford.edu/~jurafsky/slp3/>
2. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
3. Scikit-learn Developers. (2024). Scikit-learn: Machine Learning in Python. Retrieved from <https://scikit-learn.org/stable/documentation.html>

4. PyPDF2 Developers. (2024). PyPDF2 Documentation. Retrieved from <https://pypdf2.readthedocs.io/>
5. Streamlit Developers. (2024). Streamlit: The fastest way to build data apps. Retrieved from <https://docs.streamlit.io/>
6. Analytics Vidhya. (2024). TF-IDF and Cosine Similarity Explained. Retrieved from <https://www.analyticsvidhya.com/>
7. Towards Data Science. (2024). Machine Learning Text Similarity Methods. Retrieved from <https://towardsdatascience.com/>
- 8.