AI-Powered Resume Ranker
Project Report

MUHAMMED RISHAD K

23-05-2025

INTRODUCTION

In today's fast-paced job market, recruiters often face the challenge of manually reviewing hundreds of resumes for a single job opening. To streamline this process, we developed an AI-powered resume ranker that evaluates and ranks resumes based on their relevance to a given job description.

The system leverages natural language processing (NLP) and machine learning techniques to understand both the content of resumes and the requirements specified in job descriptions. This enables automated, objective ranking of applicants, significantly reducing the time and effort required during the hiring process.

ABSTRACT

This project implements an AI-based resume ranking system that uses NLP and cosine similarity to evaluate how well a set of resumes matches a specific job description. Resumes are ranked from 1 to 10, and a downloadable PDF report is generated with the results and the original job description.

The backend is built using Python and Flask, while the frontend is implemented in Flutter, providing a responsive and user-friendly interface. The system also supports uploading multiple PDF resumes, extracting text, and scoring them dynamically.

TOOLS AND TECHNOLOGIES USED

- **Frontend:** Flutter (Dart)
- **Backend:** Flask (Python)
- **PDF Processing:** PyPDF2
- **Natural Language Processing:** spaCy
- **Text Vectorization:** Scikit-learn (TF-IDF)
- **Similarity Calculation:** Cosine Similarity
- **Report Generation:** FPDF
- **Data Handling:** Pandas (for future Excel support)
- **File Storage:** Local file system
- **Mobile File Picker:** file-picker package

STEPS INVOLVED IN BUILDING THE PROJECT

1. **Setup Backend Server**: A Flask API was created to handle resume uploads and ranking logic.
2. **Resume Upload and Text Extraction**: Users upload PDF resumes; PyPDF2 extracts the content for analysis.
3. **Job Description Input**: Recruiters provide a job description to compare resumes against.
4. **Text Preprocessing**: Both job description and resumes are cleaned and tokenized using spaCy.
5. **Vectorization and Ranking**: TF-IDF vectorizer converts text into numerical form, and cosine similarity scores each resume.
6. **Results Display**: Ranked list of candidates is shown in the Flutter app, along with score bars.
7. **Report Generation**: A downloadable PDF report is generated using FPDF, including names, scores, and job description.
8. **Testing and Debugging**: Rigorous testing was performed to ensure accuracy and robustness of the system.

CONCLUSION

The AI Resume Ranker successfully automates the initial screening process in recruitment by analyzing resumes and matching them to job descriptions. By integrating NLP and ML techniques, it offers a scalable and efficient solution for HR professionals.

Currently, the system provides a ranked list of candidates and generates a clean PDF report. In the future, enhancements like keyword highlighting, detailed analytics, and multi-format export will be added to improve usability and insight depth.

This project demonstrates how AI can simplify repetitive tasks and improve productivity in talent acquisition.

End of Report