AI POWERED RESUME ANALYSER & RANKER

CONCEPTS

⚠ Disclaimer

This document and its content are intended solely for educational, personal, and illustrative purposes.

All examples, strategies, and methodologies shared here are based on public datasets, generalized best practices, and open research. They are meant to demonstrate concepts in a learning context and do not reflect any confidential, proprietary, client-specific implementations or production-level implementations.

🚫 Commercial use, redistribution, or adaptation of this material for paid services, business solutions, client work, or monetized platforms is strictly prohibited without prior written permission.

If you're interested in adapting these insights or tools for commercial or enterprise purposes, please reach out for collaboration

About This Document

This Conceptual Study is part of a broader portfolio series designed to bridge the gap between technical execution and strategic understanding. While the main documentation explains *what* was built and *how*, this companion document explores the deeper *why* behind it.

You'll find here:

- Foundational concepts that support the project’s methodology
- Business relevance and real-world applications
- Algorithm intuition and implementation logic
- Opportunities for extension and learning paths

Whether you're a curious learner, a recruiter reviewing domain expertise, or a professional looking to adopt similar methods — this document is meant to offer clarity beyond code.
If you're eager to understand the reasoning, strategy, and impact of the solution — you're in the right place.

Happy Learning!

Introduction: AI in Resume Evaluation

In today's competitive hiring landscape, matching the right candidate to the right role requires more than just keyword matching or basic filtering. As companies receive thousands of applications for a single position, the pressure to screen efficiently, fairly, and accurately has grown significantly.

Traditional tools fall short on three fronts:

- **Lack of feedback** to candidates on what went wrong.
- **Opaque scoring systems** that miss context.
- **Scalability limitations** in batch processing.

This project explores how GenAI and Retrieval-Augmented Generation (RAG) can transform the resume screening experience with intelligence, transparency, and human-like personalization — all within a privacy-preserving local setup.

1. Algorithmic Foundations

A. Resume Scoring Using TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency) is a classic NLP method used here to:

- Convert resumes and job descriptions into weighted term vectors.
- Measure semantic similarity using **cosine similarity**.

Why it works:

- Captures importance of terms within a resume.
- Penalizes generic/common words (e.g., "teamwork") while rewarding domain-specific terms (e.g., "Docker", "OKRs").
- Outputs an interpretable score (0.0 - 1.0) showing how closely aligned the resume is with the JD.

This method is lightweight, explainable, and doesn't require model training — making it perfect for local execution.

B. AI Feedback with LLMs (Mistral via Ollama)

To go beyond scores, we use a **local LLM (Mistral-7B)** accessed via Ollama to:

- Analyze resume-JD pairs.
- Generate persona-specific improvement suggestions.
- Support free-text queries about resume content.

Prompting logic includes:

- Persona prompts ("You are hiring for an AI Product Manager...")
- Structured feedback patterns (missing skills, achievements, clarity)

LLM output is structured and context-rich, giving actionable feedback rather than generic commentary.

2. Retrieval-Augmented Generation (RAG)

LLMs are powerful, but context length is limited. To ensure our AI agent can access all resume content, we embed the resume using:

- **LangChain Document Loaders** + Character Text Splitter
- **Ollama Embeddings** for generating semantic vectors
- **FAISS** as an in-memory local vector store

With this setup, any question asked is handled by:

1. Chunking and indexing the resume into vectors
2. Finding relevant chunks based on the query
3. Injecting context + query into a custom prompt
4. Generating a high-quality response using the LLM

This allows:

- Deep dive queries ("What certifications are listed?")
- Match-specific feedback ("Does this person meet the data science requirements?")

3. Batch Ranking: Workflow and Concept

When analyzing resumes in bulk, performance and clarity become critical. This project introduces a comprehensive pipeline to support batch resume ranking with personalized feedback:

- **Multi-file upload**: Upload multiple resumes and a single JD file
- **Real-time score computation**: Each resume is analysed individually using TF-IDF and cosine similarity
- **Rank assignment**: Scores are sorted and assigned ranks (1st, 2nd, 3rd, ...)
- **PDF generation**: Role-specific feedback is created and downloadable for each resume

Key design enhancements:

- Unique keys for each Streamlit download_button to prevent ID clashes
- Sorted Data Frame generation for top-ranking resumes
- OCR fallback for scanned resumes using EasyOCR
- Structured Streamlit layout supporting batch uploads

Each resume is:

- Read and pre-processed
- Compared against the JD
- Matched on relevant skills
- Given improvement feedback via LLM
- Ranked and downloadable as PDF

This fully automated pipeline replicates the work of an entire recruiting team — with consistency and insight.

4. Mistral vs. Cloud LLMs: Why Local?

Using Mistral via Ollama provides:

- Full **offline execution**
- No need for OpenAI API keys or cloud billing
- Control over context length, prompt patterns, and inference flow
- Ability to integrate with local file systems and OCR pipelines

Mistral-7B is a powerful open model that balances:

- Strong language reasoning
- Lower compute requirements
- Faster inference speeds on consumer-grade hardware

This gives full transparency, security, and extensibility.

5. UX Design for Practical Use

Streamlit is chosen for its simplicity and visual interactivity. Key interface flows include:

A. Single Resume Flow

- Upload Resume + JD
- View Match Score
- See Extracted Skills
- Get Persona-Based Feedback
- Download PDF Summary

B. Batch Flow

- Upload multiple resumes and one JD
- Ranks auto computed and displayed
- Individual and grouped PDF downloads

C. Resume Q&A Agent

- Upload a resume only
- Ask a free-form question
- Get real-time LLM-based response

Each mode is designed to be:

- Fast (runs in seconds)
- Lightweight (runs locally)
- Insightful (AI + traditional ML)

Final Thoughts & Strategic Conclusion

This project demonstrates how modern NLP and LLM techniques can be orchestrated into a lightweight, fully local application that brings real impact to hiring and career development.

Rather than overcomplicating with black-box models or heavy cloud dependencies, it blends:

- Classical NLP (TF-IDF)
- Modern GenAI (Mistral)
- RAG (LangChain + FAISS)
- Simple UX (Streamlit)

The result: a practical, fast, explainable tool that benefits both hiring teams and candidates.

Whether embedded into an ATS, used in a university career center, or run offline by a job seeker — this resume agent represents the future of human-AI collaboration in hiring.