# AI Resume Ranker Report

## Introduction

Screening manually hundreds of resumes in today's recruitment process is inefficient and time-consuming. To solve this, we created an AI-based Resume Ranker that auto-ranks resumes by comparing candidate resumes against a specified job description based on Natural Language Processing (NLP). This reduces the recruitment process and improves candidate-job matching.

## Abstract

AI Resume Ranker is an online tool that utilizes NLP methods to compare resumes against a given job description. By preprocessing the text with SpaCy, vectorizing with TF-IDF, and then using cosine similarity, the system ranks resumes according to the relevance to the position. The friendly interface enables HR staff to upload multiple resumes, see ranked results, and download the final result as a CSV file. This solution minimizes manual effort and maximizes hiring accuracy.

## Tools Used

- **Python** – Core backend functionality
- **Flask** – Web application framework
- **SpaCy** – NLP processing (lemmatization, stopword elimination)
- **Scikit**-learn – TF-IDF Vectorizer and cosine similarity scoring
- **PyPDF2** – To extract text from PDF resumes
- **HTML/CSS** – Frontend layout
- **Pandas** – DataFrame operations and CSV export

## Steps Involved in Building the Project

1. **Environment Setup**
   Installed libraries required:
   bash
   Copy
   Edit
   ```
   pip install spacy pandas scikit-learn PyPDF2
   python -m spacy download en_core_web_sm
   ```
2. **Folder Structure Initialization**
   Added templates/, uploads/, and static/ directories to hold HTML, uploaded files, and stylesheets.

3. **Resume Parsing and Preprocessing**
   Resumes are uploaded in PDF format.
   Used PyPDF2 to extract raw text.
   Used SpaCy to clean and lemmatize the text, removing stopwords and punctuation.

4. **Job Description Input**
   Pasting a custom job description or choosing from predefined templates (Python Developer, Data Scientist, etc.) is possible for users.

5. **Similarity Scoring**
   Concatenated the job description and cleaned resume texts
   Applied Used TF-IDF Vectorizer to convert text to numeric feature vectors.
   cosine similarity to quantify relevance of every resume to the job.

6. **Ranking and Output**
   Ranked resumes in order of score, descending.
   Rendered results in dynamic HTML table.
   Enabled CSV download of ranked results (ranked_resumes.csv).

7. **Web Interface**
   Created easy-to-use HTML form with CSS styling.
   Give visual feedback for error, success, and file uploads.
   Added dropdown role selection to auto-populate job descriptions.

**Conclusion**

This project successfully showcases the coupling of NLP and web technology to address a practical problem in hiring. AI Resume Ranker is scalable, precise, and adaptable to various positions. It conserves precious HR time and yields data-driven insights into candidate-job compatibility. Areas to improve in the future might involve DOCX file support, email integration, and in-depth semantic analysis with transformer-based models such as BERT.