PROJECT TITLE - AI-Powered Resume Ranker

NAME : J.A.MAANSI **REG NO.: 22MIC0055**

Introduction

Recruiters often face the daunting task of reviewing hundreds of resumes to identify the best candidates for a job role. This manual process is time-consuming, prone to bias, and inefficient. To address this challenge, our project introduces an AI-Powered Resume Ranker, a system that automates the process of evaluating and ranking resumes based on a given job description. By leveraging Natural Language Processing (NLP), this solution enhances recruitment efficiency, saves time, and ensures that the most relevant candidates are highlighted for further review.

Abstract

The AI-Powered Resume Ranker is designed to automate the shortlisting process for HR professionals by analyzing the textual content of resumes and comparing it with the requirements of a job description. The system uses Python for development, SpaCy for text preprocessing, and Scikit-learn (Sklearn) for vectorization and scoring. A Flask-based web interface allows users to upload multiple PDF resumes and a job description. The resumes are ranked based on their relevance using a TF-IDF-based similarity score. The platform also provides an option to download a structured HR report containing the scores and ranking of each candidate. This intelligent automation tool reduces the effort required for initial screening and improves the recruitment workflow.

Tools Used

- Python: Core programming language used for the application logic.
- SpaCy: Used for natural language preprocessing like tokenization, lemmatization, and stop word removal.
- Scikit-learn (Sklearn): Utilized for implementing TF-IDF vectorization and cosine similarity scoring.
- Flask: A lightweight web framework used to build the user interface and handle backend processes.
- PyMuPDF (fitz): Used for extracting text from PDF files.

- HTML/CSS: For building a user-friendly front end.

Steps Involved in Building the Project

1. Extract Text from Resumes: Use PyMuPDF (fitz) to extract raw text from uploaded PDF resumes.
2. Preprocess Text with SpaCy: Perform cleaning, tokenization, stop word removal, and lemmatization to prepare the text for analysis.
3. Input Job Description: The recruiter provides a job description (via UI or text field), which is also preprocessed using SpaCy.
4. Vectorize using TF-IDF: Convert both job description and resume texts into TF-IDF vectors using TfidfVectorizer from Sklearn.
5. Score and Rank Resumes: Compute cosine similarity between each resume and the job description. Assign a score and sort resumes in descending order of relevance.
6. Build Flask Web Interface: Create a user-friendly UI for uploading resumes, entering the job description, and viewing ranked results.

Conclusion

The AI-Powered Resume Ranker is a practical application of NLP and machine learning techniques in the HR domain. It significantly simplifies the resume screening process by automating the evaluation of candidate profiles based on their relevance to a job description. By providing ranked outputs and downloadable reports, this system enables recruiters to make faster, data-driven decisions. The integration of SpaCy, Sklearn, and Flask ensures scalability and adaptability for real-world recruitment processes.