Project Report: Resume / Job Description Matching using NLP-09/10/25(Thursaday)

1. Executive Summary

The objective of this project was to develop a script capable of matching candidate resumes with job descriptions based on textual similarity. This was successfully achieved using core Natural Language Processing (NLP) techniques. By employing **TF-IDF (Term Frequency-Inverse Document Frequency)** for text vectorization and **Cosine Similarity** for scoring, the system generated accurate similarity scores, enabling the ranking of candidates for specific job roles.

2. Methodology and Key Techniques

The project followed a standard NLP pipeline across four main phases:

A. Data Collection & Preparation

- **Data Source:** Sample resumes (R1-R4) and job descriptions (J1-J3) were generated in-script to simulate real-world textual data.
- **Initial Cleaning:** Raw text was converted to a structured format for processing.

B. Text Preprocessing

This crucial step standardized the text to ensure consistent feature extraction:

1. **Lowercasing:** All text was converted to lowercase (e.g., "Python" → "python").
2. **Noise Removal:** Punctuation, special characters, and numbers were removed.
3. **Stopword Removal:** Common, non-essential words (e.g., "the," "is") were removed to focus on keywords.
4. **Lemmatization:** Words were reduced to their dictionary base form (e.g., "developers" → "developer") to group related skills.

C. Feature Extraction (Vectorization)

- **Technique: TF-IDF Vectorizer** was used to transform the cleaned text into numerical vectors. TF-IDF assigns a weight to each word, prioritizing skills that appear frequently in a specific document (resume/job) but rarely across the entire collection.
- **Feature Space:** Both resumes and job descriptions were transformed using the *same* fitted vectorizer to ensure they exist within the **same feature space**, which is essential for accurate comparison.

D. Similarity Computation

- **Metric: Cosine Similarity** was calculated between every resume vector and every job vector. This metric measures the cosine of the angle between two vectors, resulting in a score between 0 (no similarity) and 1 (identical).

- **3. Results and Findings**
- **A. Similarity Matrix (Heatmap)**
- The matrix below shows the raw similarity scores, with darker colors indicating a better match.

Resumes	J1 (Data Scientist)	J2 (Web Developer)	J3 (Data Analyst)
R1 (Python/ML)	0.46	0.05	0.14
R2 (React/JS)	0	0.42	0
R3 (SQL/Tableau)	0.13	0.09	0.47
R4 (Marketing)	0	0	0

4. Discussion and Future Work

A. Limitations of Current Approach

The current TF-IDF/Cosine Similarity model relies solely on **keyword overlap**. This leads to two main limitations:

1. **Ignores Context:** The model treats text as a "bag of words," failing to distinguish *how* a skill is used (e.g., "seeking to learn Python" vs. "5 years experience with Python").
2. **Skill Synonymy:** It cannot recognize that skills phrased differently (e.g., "Cloud Services" vs. "AWS and Azure") are semantically related, resulting in lower scores than deserved.

B. Suggested Improvements

To achieve higher matching accuracy, the following improvements are recommended:

1. **Semantic Matching (BERT/Sentence Transformers):** Replace TF-IDF with modern **contextual embeddings** (like **BERT**) to capture the *meaning* of phrases, not just the word count. This would allow the system to match synonyms and related concepts accurately.
2. **Weighted Skills:** Introduce a mechanism to assign higher importance to core technical skills (e.g., "Python," "React") versus generic terms (e.g., "experienced," "team player").

Project Summary: Resume-Job Matching (TF-IDF & Cosine Similarity):

This project developed a pipeline for matching resumes to job descriptions based on textual similarity. The core process converts text into numerical representations and then measures the angle between these representations.

Key Steps and Techniques

1. **Text Preprocessing:** Standardized text via **lowercasing**, **stopword removal**, and **lemmatization**.
2. **Vectorization (TF-IDF):** Transformed cleaned text into numerical vectors using **TF-IDF (Term Frequency-Inverse Document Frequency)**. This method weights words by importance: high for words frequent in one document but rare across the entire collection.
3. **Similarity Calculation (Cosine Similarity):** Measured the similarity between the resume vector (A) and the job vector (B).

Core Mathematical Equation

The similarity score is calculated using the **Cosine Similarity** formula, which measures the cosine of the angle (θ) between the two vectors. A score closer to 1 indicates a higher match (smaller angle).

Cosine Similarity(A,B)=‖A‖‖B‖A·B=∑i=1nAi2 ∑i=1nBi2 ∑i=1nAiBi

Outcome

The script successfully generated a **similarity matrix** and ranked candidates, demonstrating that Resume R3 (Data Analyst) was the top match for Job J3 (SQL/Data Analyst) with a score of **0.47**.