

## **Problem Statement: MATCHING OF JOB OPENINGS WITH SKILL DATABASE USING MACHINE LEARNING**

### **Dataset:**

The dataset is a set of resumes of job seekers and a set of jobs with descriptions and required skills.

**Approach:** The methodology for the problem statement is explained for various use cases.

### **Dynamic Job Matching:**

In this module, the job seeker will upload their resume and get the jobs suitable for their skills.

Approach:

1. The candidate's or job seeker's resume is taken as input for the application
2. The resume is parsed, and the features like name, skills, experience, etc. are extracted and converted to a feature vector. The resume parsing can be done using Named Entity Recognition.
3. Now, assuming that there is a list of fields, the feature vector is mapped to any of them. Topic Classification is used to do this. Each topic in the model is one of the fields in the list.
4. The topic or field which is the output from the previous step is used to scrap job postings from employment websites like Naukri.com, Indeed, and LinkedIn. This is done using the Requests, BeautifulSoup Library in Python.
5. The set of job postings extracted in the previous step is then filtered based on parameters like experience, salary, etc.
6. Retrieved job postings are then recommended to the job seeker with the URL to the application site.

Concepts used:

1. Named Entity Recognition: It is an NLP task that involves identifying a named entity within the text. This is useful in extracting information from the text. For this libraries used are spaCy.
2. Topic Classification: It is the NLP task of assigning predefined categories or labels to the input.

### **Automated Candidate Shortlisting:**

This module helps the recruiter shortlist the candidates who have applied for the specific job. The shortlisting is done based on quantitative factors like skill set, experience, etc., and qualitative factors like problem-solving ability, and career goals.

Approach:

1. Resumes are screened based on the skill sets of the candidates and the job description.
2. Then, we can have a weighted score by assigning different scores to different skills. The candidates are ranked based on the top scores. The top-ranked candidates can be used for further analysis.
3. Now, Collaborative filtering can be used to find employee referrals and find similar candidates from past successful recruitment.
4. Classification models can be incorporated to classify the candidate as suitable or not suitable.

### **Skill Gap Analysis:**

In this module, the gap in the skill set of the candidate is compared with the job description, and the missing skills are then recommended as a course or training program for the candidate.

Approach:

1. The data is the candidate's resume and job description. The skillset of the candidate is taken as a vector and the skills required are taken from the job description.
2. TF-IDF is used to convert the skill description into numerical values. A binary matrix is created for mapping the skills possessed with the skills required. 1 represents the candidate has a particular skill in his skill and 0 represents that the candidate doesn't possess the particular skill
3. Using the binary matrix the candidate's gap is analyzed and course and training program recommendations are made to fill the gap.

### **Role Transition Assistance:**

This application recommends a plan for the employee to acquire the necessary skills for the new role within the company.

Approach:

1. Data is collected as the skills possessed by the employee and skills and competencies required for different roles within the company.
2. Skill matching and mapping can be done using TF-IDF or Word embeddings. The vector similarity is calculated between the employee's skills and skills profile using cosine similarity.
3. Matching can be done using a rule-based method where set if-then rules are used for matching assuming the skills are well defined.
4. Recommendation generation can be done using ranking roles by skill overlap and skill importance.
5. Now, the internal training resources are recommended for the employee assuming the company offers an internal training program.

6. Regular analysis of employee progress in development activities is done and feedback is taken.

### **Matching Mentors and Mentees:**

In this module, Mentees are matched with mentors with relevant expertise to enhance their growth.

Approach: Combining collaborative and content-based filtering to match the mentors with the mentees.

Collaborative Filtering Approach:

1. Data Collection: Gathering data on mentors and mentees, including their skills, experiences, career goals, and interactions.
2. Data Preprocessing: Representing skills and attributes as numerical vectors.
3. User-Item Matrix: Creating a user-item matrix where rows represent mentees and columns represent mentors. Each cell contains interaction data (e.g., Ratings, Feedback)
4. Similarity Calculation: The similarity between users (mentees) is based on their preferences and behaviors. Here cosine similarity is used.
5. Recommendation Generation:
  - a. For a mentee, identifying mentors with similar preferences who haven't been interacted with.
  - b. Ranking mentors by their similarity scores and present the top candidates as recommendations.

Similarly, Content-based filtering is done

For better results content-based and collaborative filtering is done.