**SEMINAR ABSTRACT   
TOPIC : JOB RECOMMENDATION**

ASHNA KARIM

INT. MCA S9

**Job Recommendation**

Job recommendation systems play a crucial role in connecting job seekers with suitable employment opportunities. This study proposes a job recommendation approach utilizing the k-nearest neighbors (KNN) algorithm. KNN leverages a proximity-based recommendation mechanism to identify similar jobs based on the job seeker's preferences and qualifications. The algorithm calculates the distances between job features and recommends jobs with the shortest distances, indicative of similarity. The proposed system enhances job search efficiency by suggesting jobs that closely match the user's profile, thereby facilitating informed career decisions and fostering a more effective job seeking experience. Evaluation metrics such as accuracy, precision, recall, and F1-score demonstrate the effectiveness and reliability of the KNN-based job recommendation system, showcasing its potential to assist job seekers in finding their ideal employment opportunities. Future work involves exploring hybrid recommendation approaches to further improve recommendation accuracy and user satisfaction.

1.Data Collection and Preprocessing:

Gather a dataset containing job descriptions, titles, positions, and relevant features such as skills, qualifications, and experience.

Preprocess the data by cleaning, transforming, and encoding the features to make them suitable for KNN.

2.Feature Engineering:

Define the features (e.g., skills, qualifications) that are relevant for job recommendations.

Represent each job as a feature vector, with each feature representing a specific aspect related to the job.

3.Normalize Features:

Normalize the feature vectors to ensure that all features contribute equally to the distance calculations in KNN.

4.Splitting the Dataset:

Divide the dataset into training and testing sets to evaluate the performance of the KNN model.

5.KNN Model Training:

Train the KNN model on the training dataset, using the feature vectors and their corresponding job titles or positions.

Choose an appropriate value for 'k,' the number of nearest neighbors to consider.

6.Distance Calculation:

Implement a distance metric (e.g., Euclidean distance, cosine similarity) to calculate the distances between feature vectors.

7.Recommendation Generation:

Given a new job seeker's profile (feature vector), calculate the distances to all jobs in the dataset.

Select the 'k' nearest neighbors (jobs) based on the calculated distances.

8.Recommendation Ranking:

Rank the 'k' nearest jobs based on their distances in ascending order (closest to farthest).

9.Present Recommendations:

Present the top-ranked jobs to the job seeker as recommendations, helping them find suitable job opportunities based on their profile.

10.Evaluation and Tuning:

Evaluate the recommendation system using appropriate metrics (e.g., accuracy, precision, recall) on the testing dataset.

Fine-tune the model by experimenting with different values of 'k' and distance metrics to optimize the recommendation performance.

11.User Interface Implementation:

Develop a user interface where job seekers can input their profile and preferences.

Integrate the KNN recommendation model to provide real-time job recommendations based on the user's input.

12.Deployment and Monitoring:

Deploy the job recommendation system to a web platform or application where users can access and utilize the recommendation service.

Monitor system performance, gather user feedback, and continuously improve the recommendation algorithm for enhanced user satisfaction.