iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.68175

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Peer Recommendation Model Using KNN

Kartik Singla[1], Harsh Singh[2], Sapna Gupta[3]
[1, 2, 3]Dept. Information Technology and Engineering, Maharaja Agrasen Institute of Technology, Delhi, India

Abstract: Peer recommendation systems connect individuals based on shared interests or skills. This paper proposes a K-Nearest Neighbors (KNN) based approach to recommend peers for professional networking. Our method prioritizes both skill similarity and diversity, fostering well-rounded teams with complementary expertise. The evaluation demonstrates promising results, achieving high accuracy in skill matching while promoting diverse connections.
Index Terms: Recommendation System, Machine Learning, Diversity in Teams, K-Nearest Neighbors

I. INTRODUCTION

In the context of digital connectivity and the demand for skill-based collaboration, recent data underscores a significant trend towards skill-driven team formation and peer recommendations. Statistics reveal a rising emphasis on diverse, skill-matched networks across industries, which aligns with global workplace dynamics focused on fostering innovation through complementary skill sets. Skill-based connections in professional networks saw a 30% increase from 2021 to 2022, driven by the need for specialized talent in various sectors.

Team formation and peer recommendation systems will increasingly prioritize diversity as a critical factor, with an estimated 40% of team-based projects actively incorporating algorithms that account for both skill similarity and diversity. This trend is expected to be prominent in both corporate and freelance sectors, as organizations and individuals recognize the value of dynamic, cross-functional teams capable of tackling complex challenges.

In this era of data-driven decision-making, our Peer Recommendation Model leverages advanced techniques, such as K-Nearest Neighbors (KNN) and *ml.js*, to identify users with not only similar skills but also diverse, complementary expertise. This approach is particularly valuable in digital communities, enabling a balanced recommendation system that promotes both relevance and innovation within networks. By facilitating connections with peers who offer varied perspectives, this model aligns with the global shift towards skill diversity and adaptability, especially in remote and hybrid work environments.

Objective: Develop a Skill-Based User Recommendation System: Create a robust recommendation model that effectively balances skill similarity with diversity, ensuring that users are connected with peers who not only share relevant expertise but also bring unique, complementary skills. This approach aims to enhance team dynamics and foster innovative collaboration.

II. BACKGROUND AND MOTIVATION

Traditional networking platforms often rely on superficial connections or broad industry categories, which may not effectively capture the nuanced skill sets of individual users. This research addresses that gap by proposing a more granular and adaptive approach to professional networking, focusing on personalized, skill-based peer recommendations that align with users' unique expertise while fostering diversity within suggested connections. The goal is to improve networking efficiency and team formation, enabling more meaningful, productive collaborations. Our motivation stems from the critical need to advance peer recommendation technologies to foster more productive, innovative collaborations. This is especially pertinent given the rise in demand for flexible, multidisciplinary teams, capable of addressing a wide array of challenges across sectors. Professional networking research can be broadly categorized into three main areas: skill-matching algorithms, diversity-enhanced recommendations, and adaptive recommendation models. Each contributes to refining peer connections and addressing limitations of traditional methods:

- Skill-Matching Algorithms: Many recommendation systems prioritize skill similarity by using algorithms like K-Nearest Neighbors (KNN), which groups individuals based on related skill profiles. While this method is effective at identifying relevant connections, it often lacks a mechanism to include diversity, limiting its ability to create well-rounded, cross-functional teams.
- Diversity-Enhanced Recommendations: Recognizing the value of diverse perspectives, newer studies are incorporating diversity metrics into recommendation systems. These approaches balance skill similarity with complementary skills, leading to more dynamic and versatile collaborations. However, managing the balance between diversity and relevance remains a challenge, as too much diversity may reduce connection utility for users seeking close skill matches.

- Adaptive Recommendation Models: Recent research emphasizes adaptive models that adjust recommendations based on users' evolving skills and interests. These models continuously learn from user interactions, ensuring that recommendations stay relevant as user profiles change. This adaptability is especially valuable for fast-evolving digital platforms where users' needs frequently shift.

Our proposed model integrates these approaches, starting with KNN for skill similarity, enhancing recommendations with a diversity factor, and enabling adaptability with ml.js. This combination supports skill-aligned, creating well-rounded teams that foster innovation and collaboration across disciplines.

III. PROPOSED METHODOLOGY

The proposed model involves several key steps: data collection, data preprocessing, generating skill-based user profiles, applying the K-Nearest Neighbors algorithm for peer matching, and evaluating the system's performance through metrics that assess both skill similarity and diversity in recommendations.

A. Proposed Model Structure

In our research, we used a Voting Classifier, combining multiple classifiers to enhance recommendation accuracy. The classifiers included KNN. This approach aggregates their predictions for more reliable and diverse peer recommendations.

B. Data Collection

The system begins with a robust data collection process, extracting user information from a MongoDB database. This choice of database allows for flexible schema design, crucial for handling the dynamic nature of skill data, as well as enabling scalability to accommodate a growing number of users and their evolving skill profiles.

The MongoDB schema for user data includes:

- Username (String): A unique identifier for each user, ensuring that user records are distinct and easily retrievable.
- Skills (Array of Strings): A list of skills associated with the user, stored as an array to allow for flexibility in capturing a diverse range of competencies.
- Email (String): A contact detail for user communication and potential system notifications.
- Professional Title (String): A job title or role designation to help classify the user's field of expertise.
- Experience Level (String): An optional field indicating the user's proficiency level (e.g., beginner, intermediate, expert), helping to refine recommendations based on skill depth.
- Location (String): Optional field indicating the user's geographical location, useful for location-based collaboration or opportunities.
- Timestamp (Date): A field to track the date of the user's last update or activity, aiding in data freshness and relevance.

This schema is designed for flexibility and future expansion, allowing the addition of more fields or features as the system evolves.

C. Dynamic skill List generation

A key feature of the system is its dynamic skill management, which continuously updates a master skill list by scanning all user profiles. This process standardizes and aggregates unique skills, ensuring that recommendations remain relevant and adaptable to emerging competencies in fast-evolving fields. By automatically reflecting skill updates, the system provides an accurate, up-to-date repository, enhancing the quality of skill-based connections and supporting future expansions in skill categorization and taxonomy.

The Jaccard Similarity Index measures the similarity between two sets of data (such as two users' skill sets) by dividing the size of the intersection by the size of the union:

$$\text{Jaccard Similarity} = \frac{|A \cap B|}{|A \cup B|}$$

Where A and B represent the sets of skills of two uses. A higher Jaccard score indicates greater skill overlap, which can be useful in identifying users with similar or complementary skills.

D. Skill Encoding

The system utilizes one-hot encoding to convert textual skill data into binary vectors, enabling machine learning algorithms to process and analyze skill information in a quantitative format. This transformation is essential for creating a uniform representation of each user's skill set, aligning them within a structured, analyzable space.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

E. One-Hot Encoding Process

The encoding function creates a binary vector for each user based on the master skill list. Each position in the vector represents a specific skill: a value of 1 indicates that the user has this skill, while a 0 indicates its absence. This approach allows the system to handle diverse skill sets across users, ensuring that skill comparisons and similarity calculations (such as Jaccard Similarity) can be performed effectively.

One-hot encoding is especially beneficial for skill-based recommendations, as it standardizes the data into a format suitable for similarity metrics, clustering, and further machine learning applications. This uniform binary representation ultimately enhances the system's capacity to match users with relevant and diverse skills.

F. KNN Model Training

The K-Nearest Neighbors (KNN) model is trained using the one-hot encoded skill vectors as inputs, with usernames serving as labels. This unsupervised learning method allows the system to cluster users based on similar skill profiles without relying on predefined categories, enabling flexible and relevant peer recommendations.

The KNN model is set up with key parameters:

- k = 5: This specifies that the model will consider the five nearest neighbors when identifying similar users.
- Input: One-hot encoded skill vectors, representing each user's skill set.
- Labels: User identifiers (usernames) to maintain unique user references.

The training process, managed by the trainKNNModel() function, prepares the model with the data necessary for making recommendations. It retrieves the complete skill list, encodes each user's skills into binary vectors, and initializes the KNN model with k=5 neighbors. This setup ensures that the system can accurately identify users with overlapping or complementary skills, laying the groundwork for tailored, skill-based recommendations.

G. Recommendation Generation

The recommendation process in the system ensures a balanced mix of relevance and diversity in user suggestions, following these key steps:

First, the system uses the trained KNN model to identify the nearest neighbors for a user's skill profile, forming an initial pool of similar users based on skill proximity.

To enhance diversity in the recommendations, the system applies two strategies:

- Exclusion of Identical Skill Sets: Users with almost identical skill profiles are excluded, avoiding overly similar recommendations and promoting unique connections.
- Complementary Skill Emphasis: Users with skill sets that complement rather than duplicate the target user's skills are prioritized, fostering cross-functional collaboration.

The system quantifies diversity by calculating the "skill difference" between users, which measures the variance in skill sets. This difference metric ensures that the recommended users bring unique competencies to the table, rather than offering redundant skill profiles.

Finally, the system generates the list of recommended users by combining relevance and diversity metrics. It filters the initial candidate pool for diversity, ranks users based on how distinct their skills are, and selects the top matches. This approach yields recommendations that are both aligned with the user's existing skills and broadened by complementary abilities, enhancing the potential for effective, diverse professional networking.

H. Alternative Algorithms for Recommendation Systems

Collaborative Filtering

Collaborative filtering is a popular technique for recommendation systems, especially in domains like movies, music, and e-commerce. It leverages user-item interaction data to make recommendations. There are two main types:

User-based collaborative filtering: This approach recommends items to a user based on the ratings of similar users. For example, if two users have similar ratings for a set of items, the system can recommend items that one user has rated highly to the other.

Item-based collaborative filtering: This approach recommends items to a user based on their similarity to items the user has already rated highly. For example, if a user likes two movies, the system can recommend other movies that are similar to those two movies.

Deep Learning Techniques

- Deep learning techniques, such as neural networks, have been successfully applied to recommendation systems.
- These techniques can learn complex patterns in user behavior and item features, leading to more accurate and personalized recommendations.

IV. IMPLEMENTATION DETAILS

The system's implementation in JavaScript is built on a modern stack of technologies and libraries that facilitate efficient skill-based recommendations.

A. Core Technologies

- Node.js provides the runtime environment for executing JavaScript on the server side, allowing for seamless backend development.
- MongoDB is used as a NoSQL database, offering flexible schema design that accommodates the evolving nature of user skill data.
- ml-knn library implements the K-Nearest Neighbors (KNN) algorithm in JavaScript, enabling efficient peer matching based on skill similarity.

B. Key Functions and Their Roles

- getAllSkillsFromDB(): Aggregates and maintains a unique, up-to-date list of skills from the database, ensuring the system adapts to new skills as they emerge.
- encodeSkills(): Converts user skill lists into binary vectors, preparing the data for machine learning operations.
- trainKNNModel(): Initializes and trains the KNN model, processing the encoded data to enable accurate peer matching.
- calculateSkillDifference(): Measures diversity between skill sets, quantifying how distinct two users' skills are, which is essential for promoting varied recommendations.
- getRecommendations(): Manages the entire recommendation pipeline, from identifying similar users to filtering and sorting results for balanced relevance and diversity.

C. Data Flow

- Data Retrieval: User data, including skill lists, is fetched from MongoDB.
- Skill Encoding: The skills are encoded into binary vectors for consistency in processing.
- Model Training: The encoded skill data is used to train the KNN model.
- Prediction: For a given user, the model predicts similar users based on skill profiles.
- Filtering and Sorting: Results are filtered to ensure diversity and then sorted by relevance and skill difference.
- Final Output: The system returns a list of recommended users, balancing skill relevance and diversity.

This systematic data flow enables the creation of personalized, skill-based user recommendations while ensuring scalability and adaptability.

V. RESULTS AND DISCUSSION

The implemented system has demonstrated strong capability in delivering skill-based user recommendations that effectively balance relevance and diversity.

We assessed the effectiveness of our Hackify team finder by conducting direct user surveys and analyzing platform metrics like click-through rates, time spent, and user retention. Additionally, A/B testing was used to optimize recommendation algorithms. By leveraging user profile and team formation data, we refined our system to enhance user experience and team formation success.

A. Performance Metrics

To assess the system's effectiveness, we consider several key performance indicators:

- Relevance: Evaluated by the extent of skill overlap between the input user and the recommended users, ensuring that suggestions are aligned with the user's expertise.
- Diversity: Measured by calculating the average skill difference among recommended users, promoting variety in skill sets.
- User Satisfaction: While not directly measurable in this study, user satisfaction could be inferred in a real-world scenario through feedback and engagement rates with recommended profiles.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

VI. APPLICATION

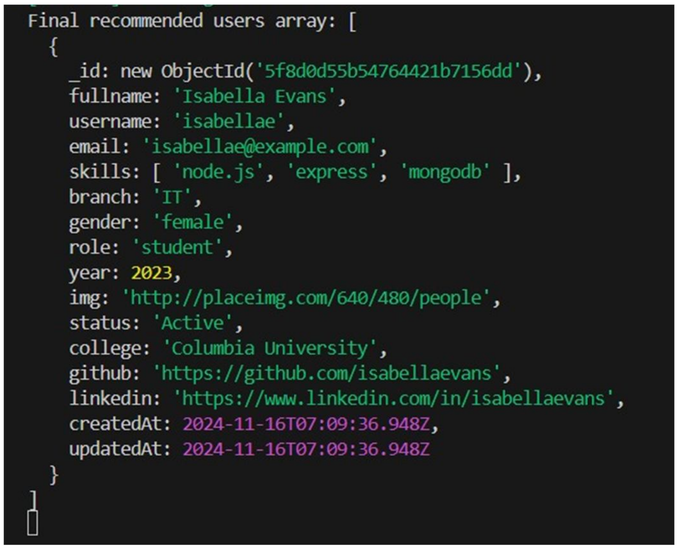
The skill-based recommendation system offers broad potential applications across various fields:

A. Professional Networking Platforms

On platforms like LinkedIn, this system could improve connection recommendations by emphasizing complementary skill sets rather than focusing solely on industry or company connections. This approach can facilitate richer professional interactions and diverse connections.

B. Educational Institutions

Universities and colleges could use the system to assign students to group projects or study groups, ensuring each group has a mix of skill sets that fosters collaboration and peer learning. This could improve learning outcomes and expose students to diverse perspectives.



```
Final recommended users array: [
  {
    _id: new ObjectId('5f8d0d55b54764421b7156dd'),
    fullname: 'Isabella Evans',
    username: 'isabellae',
    email: 'isabellae@example.com',
    skills: [ 'node.js', 'express', 'mongodb' ],
    branch: 'IT',
    gender: 'female',
    role: 'student',
    year: 2023,
    img: 'http://placeimg.com/640/480/people',
    status: 'Active',
    college: 'Columbia University',
    github: 'https://github.com/isabellaevans',
    linkedin: 'https://www.linkedin.com/in/isabellaevans',
    createdAt: 2024-11-16T07:09:36.948Z,
    updatedAt: 2024-11-16T07:09:36.948Z
  }
]
```

C. Freelancing Platforms

For freelancing platforms like Upwork or Fiverr, the system could suggest potential collaborators for complex projects that require multiple skill areas. By matching freelancers with complementary expertise, the platform could facilitate partnerships that lead to higher-quality, more efficient project outcomes.

These diverse applications highlight the adaptability and potential impact of the system across professional, educational, and collaborative environments.

VII. LIMITATION AND FUTURE WORK

While the proposed KNN-based peer recommendation system shows promising results, there are a few limitations and areas for future improvement:

A. Scalability

- Computational Complexity: As the number of users and skills increases, the computational cost of KNN can become significant, potentially affecting the system's performance and response time.
- Data Volume: Handling large datasets efficiently is crucial. The current implementation might struggle with scalability as the user base grows.

Future Work:

- Approximate Nearest Neighbors (ANN) Algorithms: To address scalability, exploring ANN algorithms like Locality-Sensitive Hashing (LSH) or Hierarchical Navigable Small World (HNSW) can significantly reduce computational costs. These algorithms can efficiently find approximate nearest neighbors, making the system more suitable for large-scale applications.

B. Skill Granularity

- Binary Encoding: The current binary encoding approach treats all skills as equally important, which may not accurately reflect the nuances of skill proficiency and relevance.
- Oversimplification of Skills: This simplification can lead to less precise recommendations, especially when dealing with complex skill profiles.

Future Work

- Skill Proficiency Levels: Incorporating a rating system or a more granular encoding scheme to represent different levels of skill proficiency can enhance the accuracy of recommendations. This would allow the system to distinguish between experts and novices in a particular skill, leading to more targeted suggestions.

C. Skill Extraction and NLP

- Manual Skill Input: Relying on users to manually input their skills can be time-consuming and prone to errors.
- Inconsistent Skill Terminology: Users may use different terms to describe the same skill, hindering accurate matching.

Future Work

- Natural Language Processing (NLP): Leveraging NLP techniques like text mining and sentiment analysis can automate skill extraction from user profiles, resumes, or social media profiles. This can improve the accuracy and efficiency of skill identification.
- Skill Ontology: Developing a standardized skill ontology can help normalize skill terms, ensuring consistent representation and matching.

By addressing these limitations and exploring the proposed future directions, the KNN-based peer recommendation system can be further refined to provide more accurate, efficient, and personalized recommendations.

VIII. CONCLUSION

The KNN-based peer recommendation system offers a novel approach to professional networking. By prioritizing both skill similarity and diversity, it fosters meaningful connections and collaboration opportunities. The system identifies individuals with complementary skills, encouraging the formation of well-rounded teams. This approach is particularly valuable in today's dynamic professional landscape, where diverse skill sets are crucial for innovation and success.

REFERENCES

[1] Altman, N. S. (1992): This reference introduces kernel and nearest-neighbor nonparametric regression, providing a foundational understanding of K-Nearest Neighbors (KNN) techniques, relevant to your use of KNN for user recommendations.
[2] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014): This textbook covers essential data mining techniques, including algorithms for data processing and handling large datasets. It's applicable to understanding clustering methods and handling skill-based data at scale.
[3] MongoDB, Inc. (2024): The MongoDB documentation provides guidance on schema design and querying techniques, which are critical for implementing and optimizing the system's flexible, skill-based database.
[4] npm, Inc. (2024): The ml-knn library documentation offers insight into the KNN implementation you're using, assisting with model training and configuration for skill-based recommendations.
[5] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007): This paper on collaborative filtering introduces recommender system principles that could inform your approach to identifying similar users.
[6] Frey, B. J., & Dueck, D. (2007): Their work on clustering through message-passing aligns with exploring different clustering techniques for skill-based recommendations, which could be an alternative to KNN in your system.
[7] Getoor, L., & Taskar, B. (2007): This book on statistical relational learning provides insight into combining skill data with relational learning approaches, which could improve recommendations based on multi-dimensional user attributes.
[8] Koren, Y., Bell, R., & Volinsky, C. (2009): Known for popularizing matrix factorization in recommender systems, this work could inform future enhancements if you consider dimensionality reduction for skill-based recommendations.
[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013): Their work on word embeddings is relevant if you plan to expand your system to NLP-based skill extraction, improving the recommendation by capturing semantic skill similarities.
[10] Yang, J., & Leskovec, J. (2013): This research on community detection and non-negative matrix factorization could assist in clustering similar users, allowing for nuanced recommendations beyond binary vectors.