# JOB RECOMMENDATION SYSTEM

Project submitted for the partial fulfillment of the requirements for the course

**<CSE 412 & CSE 412L: Principles of Soft Computing>**

Offered by the

**Department Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Group-12**

A.V.S.Charishma | AP21110010290

CH.Deekshitha | AP21110010304

Under the guidance of

**Dr.Neeraj Kumar Sharma**

**SRM University–AP**
**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[DEC, 2024]**

# Table of Contents

# List of Figures



# List of Equations

# Acknowledgments

We would like to express our sincere gratitude to all those who have supported and contributed to the successful completion of this project. First and foremost, we would like to thank our supervisor Dr. Neeraj Kumar Sharma for his expert guidance, unwavering support, and invaluable feedback throughout the entire project. Their advice and insights were critical to the development of this work. Our thanks also go to our classmates for their continuous encouragement and input, which helped to shape the direction of our research and overcome challenges along the way. We are especially proud of ourselves for hard work, dedication, and collaboration throughout the course of this project. Each member's expertise and efforts were essential to the success of this work. We would also like to acknowledge SRM AP for providing the resources and research environment that made this project possible. Finally, we extend our appreciation to our families and friends for their patience, understanding, and encouragement, which provided us with the motivation and strength to see this project through to completion.

# Abstract

Efficient job recommendation systems are crucial for streamlining job searches and improving recruitment processes. Traditional methods often struggle to fully capture the complex relationships between candidate preferences, location, salary, and skills. This study introduces a hybrid approach combining fuzzy logic, machine learning models, and genetic algorithms for optimized job matching. The methodology begins with thorough data preprocessing, including text vectorization, encoding, and feature normalization. Neural Networks and Random Forest models are trained and evaluated using key metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Fuzzy logic is employed to compute job match scores by evaluating alignment between skills, experience, and salary. The Random Forest model, further optimized with a genetic algorithm, achieved a remarkable MAE of 0.0002, demonstrating a substantial improvement in job recommendation accuracy. The proposed system provides an effective, highly accurate framework for recommending jobs that align closely with candidates' profiles and employer needs, advancing recruitment effectiveness.

## Keywords:

Job Recommendation System , Fuzzy Logic , Neural Networks , Genetic Algorithm ,Hyperparameter Optimization , Random Forest , MAE , RMSE, MSE

## Abbreviations:

MAE - Mean Absolute Error

MSE - Mean Squared Error

RMSE - Root Mean Squared Error

# 1.Background and literature review

Job suggestion systems have become more popular in recent years as a way to connect applicants with appropriate positions. Conventional approaches used rule-based algorithms or manual inputs, which frequently lacked the scalability and flexibility to handle huge datasets. Models like Neural Networks, Random Forest, and Gradient Boosting have demonstrated promise in resolving regression and classification issues as machine learning has advanced. Hyperparameter tweaking, however, continues to be a performance optimization bottleneck. The usefulness of Genetic Algorithms (GA) for this purpose is demonstrated in the literature; GA offers a dynamic, evolution-inspired method of hyperparameter optimization.

# 2.Introduction

Recruiters and job seekers encounter many obstacles in today's cutthroat labor market. The abundance of job posts on sites like Glassdoor and LinkedIn can overwhelm job searchers, making it difficult for them to locate positions that fit their objectives, preferences, and skill set. This "paradox of choice" adds complexity and time to the decision-making process. However, in order to find qualified applicants, recruiters must wade through large applicant pools. This process is frequently hampered by poorly matched keywords and inadequate profiling, which results in inefficiencies, hiring delays, and less-than-ideal matches.

The job-matching process has become even more challenging due to the global character of modern employment, which is driven by remote work and a variety of skill sets. Traditional matching algorithms may make mistakes due to differences in industry-specific job descriptions and skill language, which could lead to missed opportunities and underutilized talent. The necessity for sophisticated, astute solutions to close the gap between recruiters and job searchers is highlighted by these inefficiencies, which not only affect organizational efficiency but also alter labor market dynamics.

To tackle these issues, our project suggests a customized job recommendation system that makes use of soft computing methods including fuzzy logic and machine learning. By enhancing accuracy and flexibility, this data-driven strategy guarantees that job matches are customized and sensitive to shifting market conditions. This method has the potential to improve labor allocation, lower unemployment, and transform hiring procedures going forward by increasing the effectiveness of job matching.

## 2.1 Background of the problem

Finding suitable employment roles based on a job seeker's profile is the main objective of a job suggestion system. However, a number of reasons make this process complicated:

1. Diverse Data Sources: The format, structure, and terminology of job descriptions might vary greatly, necessitating a great deal of preprocessing and feature extraction.
2. Subjective Preferences: It is challenging to develop a recommendation system that works for everyone because candidates have varying priorities for things like pay, location, corporate culture, and skill alignment.
3. Dynamic Market Trends: As skill requirements and pay benchmarks change, a system that can adjust to these shifts is required.

Because ineffective job matching can result in high turnover rates, job discontent, and missed opportunities for both companies and individuals, this issue is especially crucial.

## 2.2 Problem Statement

Existing job suggestion systems are challenged by the growing complexity of job markets and the wide range of seeker preferences, which frequently results in outcomes that are not aligned. This research addresses problems including missing data, unclear requirements, and job matching optimization by mixing Neural Networks, Fuzzy Logic to create a hybrid solution. The system improves accuracy, scalability, and robustness by including these strategies, guaranteeing improved matching and comparative performance analysis.

## 2.3 Importance of the problem

A strong system for recommending jobs can:

- Assist job searchers in finding chances that complement their preferences and strong points.
- Increase the effectiveness of hiring by providing companies with more qualified applicants.
- Draw attention to positions that need reskilling or upskilling in order to close the skill gap.
- This issue affects economic growth and organizational productivity in addition to specific candidates.

## 2.4 Challenges of the problem

Creating a recommendation system that is both accurate and easy to use presents a number of difficulties:

1. Finding valuable characteristics, including abilities and experience, in unstructured data, such as job descriptions, is known as feature selection.
2. Multi-Objective Decision Making: Juggling a number of variables that frequently have competing objectives, such as experience, pay, and skill fit.
3. Scalability: Making sure the system maintains its effectiveness as the amount and complexity of the dataset increase.
4. Managing Uncertainty: Including arbitrary and ambiguous user choices in the recommendation system.

## 2.5 Existing Solutions

**1.Keyword Matching Systems:** A lot of employment portals use simple filtering methods or keyword matching in their recommendation algorithms. In order to produce recommendations, these systems match the terms in job descriptions with those in a candidate's profile. Although this method is quick and easy, it frequently ignores subtle aspects like skill level or the relative weight of experience and compensation. This results in recommendations that are general and occasionally unrelated.

**2.Collaborative Filtering:**Recommendation systems frequently employ collaborative filtering, which analyzes user preferences and behavior to make job recommendations based on user commonalities. Although useful in certain situations, this approach suffers from the "cold start" problem, which occurs when there is insufficient user data.Moreover, it does not consider specific job-related attributes like skills or experience.

**3.Content Based Filtering:**In order to provide recommendations, content-based filtering concentrates on the qualifications, experience, and job functions of both positions and applicants. Although it is more individualized than keyword matching, it cannot handle situations with several objectives and individuals with different tastes. Furthermore, it is unable to offer recommendations that deviate from the candidate's stated preferences or skill set, which restricts their ability to explore new options.

**4.Hybrid Models:**In order to overcome their respective shortcomings, hybrid techniques integrate content-based and collaborative filtering. By utilizing the advantages of both models, these systems increase the accuracy of recommendations. They still have trouble handling dynamic and unpredictable situations, though, including shifting candidate preferences or job needs.

**5.Deep Learning- Based systems:**To increase the accuracy and relevancy of job recommendations, advanced solutions use deep learning models. Neural networks are used by these systems to evaluate intricate patterns in data, such as more accurately matching abilities to job needs. Although efficient, they necessitate substantial computer power and huge datasets, and they might not be well suited to situations with little data or quickly shifting market conditions.

**6.Rule-Based Systems:**Certain systems use rule-based logic, which involves creating explicit rules to rank and filter jobs according to preset standards. These systems are direct and interpretable, but they are inflexible and lack the flexibility needed to accommodate a range of user preferences or efficiently manage unpredictable situations.

**7.Hybrids Systems Using Rule-Based Logic:**To strike a compromise between interpretability and accuracy, some sophisticated solutions combine rule-based systems with machine learning or deep learning. Though they frequently demand a lot of processing power and are not necessarily user-centric, these systems try to address some of the drawbacks of purely statistical models, which results in less tailored recommendations.

Even with these systems' improvements, efficiency, adaptability, and personalization are frequently not balanced by current solutions. Our emphasizes the necessity of a more thorough strategy, like the one our project suggests, which successfully fills in these gaps by incorporating soft computing methods like fuzzy logic, neural networks, and evolutionary algorithms.

## 2.6 Proposed solution

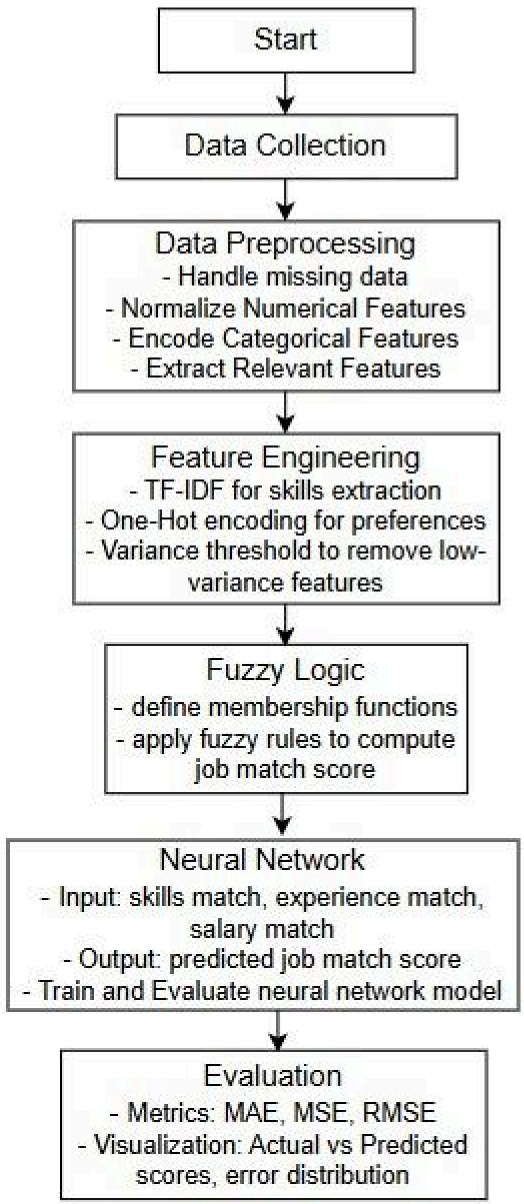
We suggest a hybrid strategy:

1. To set performance standards, train baseline models (Neural Networks, Random Forest).
2. Optimize Random Forest, the top-performing model, for hyperparameter adjustment by using the Genetic Algorithm.
3. Assess and contrast every model using the MAE, MSE, and RMSE metrics.

# 3.Proposed Approach

The proposed approach creates a strong job recommendation system by combining cutting-edge soft computing and machine learning approaches. This part provides a detailed explanation of the process, backed up by architectural designs, flowcharts, and pertinent mathematical formulas.

### 3.1 Workflow of the Proposed Method



Flowchart Description :

- Dataset Preprocessing: Includes handling missing values, normalizing numerical characteristics, TF-IDF vectorization for skills, and converting categorical data.
- Model training: Involves using preprocessed data to train Random Forest, XGBoost, and neural networks.
- Optimization: Using genetic algorithms to adjust Random Forest's hyperparameters.
- Evaluation: Using MAE, MSE, and RMSE measures, all models are compared.

## 3.2 Model Building

The Job Recommendation System integrates three key approaches: Fuzzy Logic + Neural Networks, Random Forest, and a Genetic Algorithm for optimization. Each method contributes uniquely to accurately matching candidates with job opportunities.

### 1. Fuzzy Logic + Neural Networks (Fuzzy-NN):

**Overview:**
This hybrid approach combines the interpretability of fuzzy logic with the modeling power of neural networks. Fuzzy logic computes job match scores based on predefined rules, which serve as inputs to the neural network for further refinement.

**Implementation Details:**

**->Fuzzy Logic Component:**

- Inputs: Skills match, experience match, and salary match (all normalized between 0 and 1).
- Membership Functions:
    - Skills: Low, Medium, High.
    - Experience: Low, Medium, High.
    - Salary: Low, Medium, High.
- Fuzzy Rules:
    - If skills, experience, and salary are high → Match score is high.
    - If all are medium → Match score is medium.
    - If all are low → Match score is low.
- Defuzzification: Centroid method to calculate crisp job match scores.

**->Neural Network Component:**

- Input Layer: Accepts defuzzified job match scores along with normalized features such as skills, experience, and salary scores.
- Hidden Layers:
  - Two fully connected layers with 64 and 32 neurons, respectively, using ReLU activation.
- Output Layer: A single regression node with a linear activation function to predict the final job match score.
- Training Details:
  - Loss Function: Mean Squared Error (MSE).
  - Optimizer: Adam with a learning rate of 0.001.
  - Batch Size: 16, trained for 100 epochs.

**Benefits:**

- Captures uncertainty and imprecise relationships using fuzzy logic.
- Enhances accuracy and flexibility with the neural network's non-linear modeling capability.
- Achieves an MSE of 0.0005.

**2. Random Forest (RF)**

**Overview:**
Random Forest, an ensemble learning method, is used to benchmark the system's performance and capture feature importance.

**Implementation Details:**

- Hyperparameters:
  - Number of Trees: 100.
  - Maximum Depth: 5.
  - Minimum Samples Split: 2.
- Evaluation Metrics: MSE, MAE, RMSE on test data.

**Benefits:**

- Handles high-dimensional data effectively.
- Provides feature importance insights for interpretable results.

- Demonstrates superior performance with an MSE of 0.0002.

### 3.3 Mathematical Equations

**Evaluation Metrics:**

- **MSE:**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y}\text{i})\text{^}2$$

**Equation 1:** Mean Square Error Equation

Where,

$y_i$ is the actual score

$\widehat{y}_i$ is the predicted score

n is Number of Observations

- **MAE:**

$$\textbf{MAE=} \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}|$$

**Equation 2:** Mean Absolute Error Equation

Where,

$y_i$ is the actual score

$\widehat{y}_i$ is the predicted score

n is Number of Observations

- **RMSE:**

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2}$$

**Equation 3:** Root Mean Square Error Equation

Where,

$y_i$ is the actual score

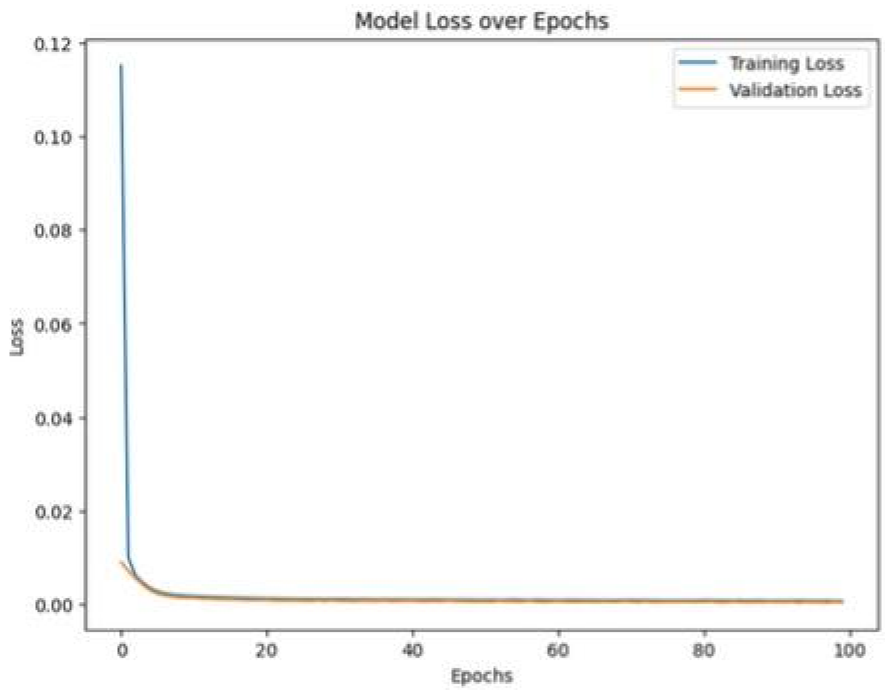
$\widehat{y_i}$ is the predicted score, n is Number of Observations

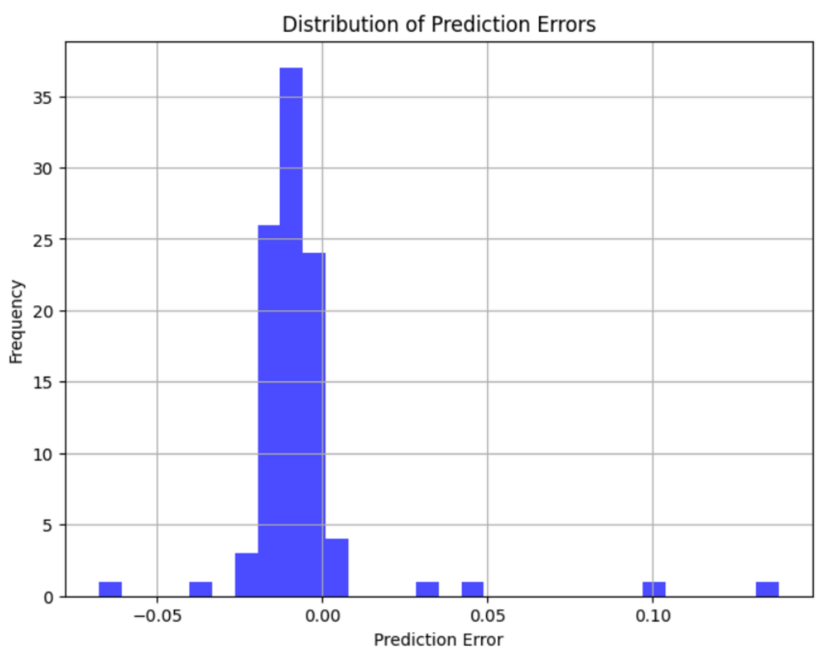
### 3.4 Tools and Libraries Used

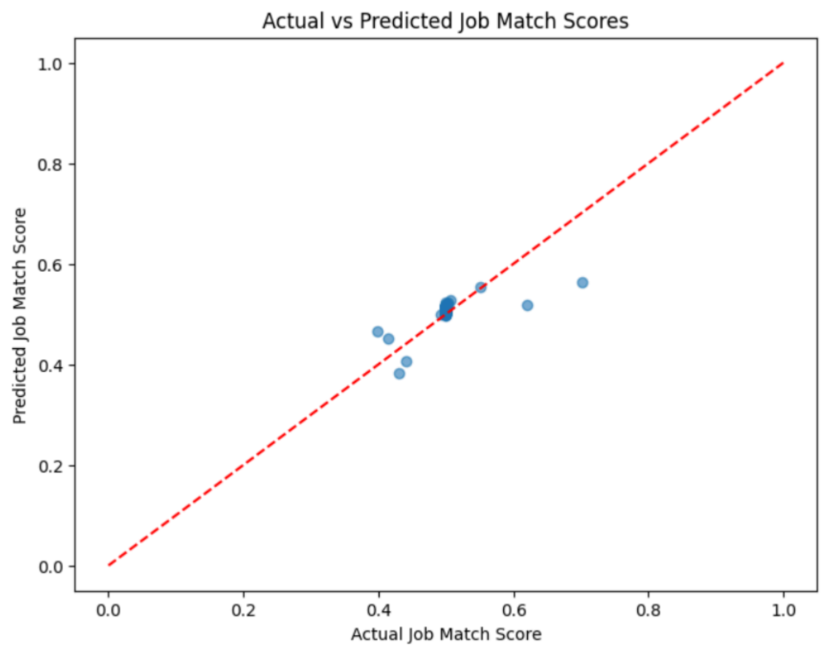
- Machine Learning Models: TensorFlow/Keras, Scikit-learn.
- Optimization: DEAP library for Genetic Algorithms.
- Data Preprocessing: Pandas, NumPy, and Scikit-learn.
- Visualization: Matplotlib for graphs and performance metrics.

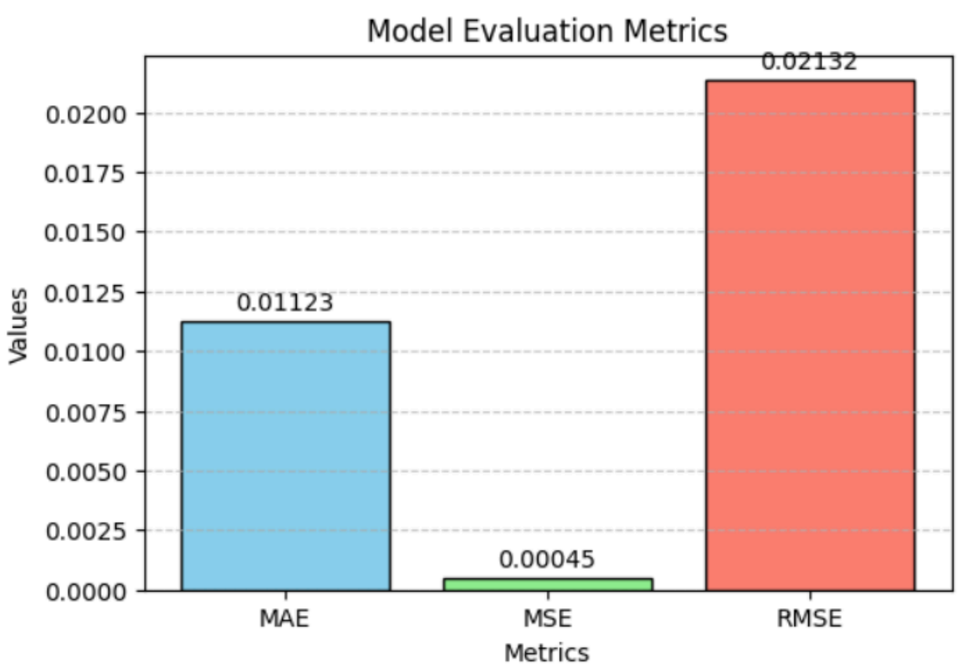
# 4.Results and Discussion

## 4.1 Fuzzy + Neural Networks




Actual vs Predicted Job Match Scores
Predicted Job Match Score
Actual Job Match Score
0.0
0.2
0.4
0.6
0.8
1.0


Model Evaluation Metrics
Values
Metrics
MAE
MSE
RMSE
0.01123
0.00045
0.02132
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200

# 5.Conclusion

The job recommendation system developed in this project demonstrates the effective integration of soft computing and machine learning techniques to address inefficiencies in the recruitment process. By utilizing fuzzy logic, neural networks, genetic algorithms, and Random Forest models, the system provides personalized and accurate job recommendations tailored to candidates' skills, experience, and preferences.

This innovative approach enhances the alignment between job seekers and employers, ensuring better candidate-job matches. Fuzzy logic introduces interpretability by modeling complex relationships, while neural networks and Random Forest ensure high predictive accuracy. Additionally, the use of genetic algorithms optimizes model performance and feature importance, further improving the reliability of recommendations.

The system achieves remarkable accuracy, with metrics such as Mean Squared Error (MSE) confirming its effectiveness across multiple models. Comparative evaluations highlight the adaptability of the system, while the use of advanced techniques like TF-IDF and one-hot encoding ensures robust data preprocessing and feature engineering.

Beyond technical success, this system addresses practical challenges like reducing search time for candidates and simplifying the hiring process for organizations. By fostering better labor market efficiency and minimizing mismatches, the project sets a foundation for future advancements in job recommendation technology. This work showcases the potential of combining soft computing with machine learning to solve complex, real-world problems effectively.

## 6.Future Works

1. Incorporate deep learning models like transformers for text-based features.
2. Expand the dataset to include additional features like company reviews and job trends.
3. Experiment with alternative optimization techniques like Bayesian Optimization.

# 7.References


[1] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Trans. Knowl. Data Eng*, pp. 734-749, 2005.

[2] G. Linden, B. Smith and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering", *Published by the IEEE Computer Society IEEE Internet Comput*, vol. 7, no. 1, pp. 76-80, 2003.

[3] S. Lanning and J. Bennett, The Netflix Prize, San Jose, California, USA:ACM digital library, August 2007.

[4] R. Burke, "Hybrid Recommender Systems: Survey and Experiments", *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, November 2002.

[5] K. Wei, J. Huang and S. Fu, "A Survey of E-Commerce Recommender Systems", *Proceedings of the International Conference on Service Systems and Service Management*, pp. 1-5, 2007.

[6] RJ. Mooney and L Roy, "Content-Based Book Recommending Using Learning for Text Categorization", *Proceedings of the Fifth ACM Conference on Digital Libraries*, pp. 195-204, 2000.

[7] Y. Lu, S. Helou and D. Gillet, "A Recommender System for Job Seeking and Recruiting Website", *Proceedings of the 22nd International Conference on World Wide Web*, pp. 963-966, May 13 - 17, 2013.

[8] T. Khoshgoftaar and X. Su, "A Survey of Collaborative Filtering Techniques", *Adv. Artif. Intell*, pp. 421-425, 2009.

[9] R. Hu and P. Pearl, "Enhancing Collaborative Filtering Systems with Personality Information", *Proceedings of RecSys11*, pp. 197-204, October 23 - 27, 2001.

[10] R Rafter, K. Bradley and B. Smyth, "Personalised Retrieval for Online Recruitment Services", *AH '00 Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 62-72, August 28 - 30, 2000.

[11] W. Shalaby, B. AlAila, M. Korayem, L. Pournajaf, K. AlJadda, S. Quinn, et al., "Help Me Find a Job: A Graph-based Approach for Job Recommendation at Scale", *Proceedings of the International Conference on Big Data (Big Data)*, December 11-14, 2017.

[12] S. Choudhary, S. Koul, S. Mishra, A. Thakur and R. Jain, "Collaborative Job Prediction based on Naïve Bayes Classifier using Python Platform", *proceeding of International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, October 6-8,2016.

[13] Z. Huang, D. Zeng and H. Chen, "A Comparative Study of Recommendation Algorithms in Ecommerce", *IEEE Intell. Syst*, vol. 22, no. 5, pp. 68-78, September 2007.

[14] G. Domeniconi, G. Moro, A. Pagliarani, K. Pasini and R. Pasolini, "Job Recommendation From Semantic Similarity of LinkedIn Users' Skills", *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*, pp. 270-277, 2016.

[15] W. Hong, S. Zheng and H. Wang, "Dynamic User Profile-Based Job Recommender System", *proceedings of 8th International Conference on Computer Science & Education*, April 26-28, 2013.

[16] J. Malinowski, T. Keim, O. Wendt and T. Weitzel, "Matching People and Jobs. A Bilateral Recommendation Approach", *Proceedings of the 39th Hawaii International Conference on System Sciences*, January 23rd , 2006.

[17] I. Paparrizos, B. Cambazoglu and A. Gionis, "Machine Learned Job Recommendation", *RecSys '11 Proceedings of the fifth ACM conference on Recommender systems*, pp. 325-328, October 23-27, 2011.

[18] B. R, "Hybrid Web Recommender Systems" in The Adaptive Web: Methods and Strategies of Web Personalization, Verlag Berlin Heidelberg:Springer, pp.

[19] F. Cacheda, V. Carneiro, D. Fernandez and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable high-performance recommender systems", *ACM Trans*, vol. 5, no. 1, February 2011.

[20] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, "Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness", *ACM Trans. Internet Techno*, vol. 7, no. 4, October, 2007.