www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

AI-Powered Job Matching System to Classify Job Candidates and Match Them with Suitable Job Openings

CH. Sanjay[1], G. Rishikesh[2], G. Rishikesh[3], Mr. D. Venkatesan[4]
[1,2,3] UG Scholar, Dept. of CSE(AI&ML),
St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
[4] Assistant Professor, Dept. of CSE(AI&ML), St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
sanjaychukkala06@gmail.com

Abstract:

The AI-powered Job Matching System that classifies job candidates and matches them with suitable job openings, optimizing the recruitment process for both job seekers and employers. The system is designed to efficiently process and analyze large volumes of job data and candidate profiles, delivering accurate and personalized job recommendations while improving the quality of hires. The system gathers data from job postings, which include details such as job titles, required skills, experience, education, location, and job type, alongside candidate data, including resumes, job history, qualifications, and preferences. Natural Language Processing (NLP) techniques are employed to preprocess text data, such as cleaning, tokenization, and embedding generation. This structured data is then used to extract relevant features from both candidates and job descriptions, enabling efficient matching. To match candidates with suitable job openings, the system utilizes a combination of machine learning models, including Gradient Boosting, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP). The first step of the matching process applies rule-based filters to quickly narrow down candidates based on basic criteria such as location, experience, and job type preferences. Subsequently, the Gradient Boosting model is used to rank candidates based on their overall fit for a particular job, considering skills, qualifications, and experience. The KNN algorithm further refines the matching process by evaluating candidate proximity in terms of key attributes, while the MLP model processes complex patterns in the data to provide precise match predictions. The output is a ranked list of candidates, ordered by their suitability for each job opening. The system incorporates a continuous feedback loop, allowing recruiters to rate the quality of candidate-job matches.

Keywords: Remote work, Productivity, Deep Learning, Work patterns, Machine learning, Collaboration metrics, Behavioural patterns, Time-tracking, Performance evaluation, Employee efficiency, Work activity data, Engagement levels, Task execution efficiency, Keystroke monitoring, Screen tracking, Cloud computing, High-speed internet, Self-reported data, Resource allocation.

1.INTRODUCTION

AI-powered job matching systems are transforming the recruitment process by automating candidate screening and job matching through machine learning algorithms. These systems analyze vast datasets, including resumes, job descriptions, and candidate profiles, to match candidates to jobs based on skills, experience, and preferences. This approach improves the quality of hires, reduces recruitment time, and enhances user experience. AI models like Gradient Boosting, KNN, and MLP ensure that job seekers are matched with jobs that best suit their qualifications and preferences, improving job satisfaction and retention. The system's ability to learn from data continuously makes it adaptable and capable of offering personalized recommendations. As the job market grows, especially in India, such AI-driven systems offer immense potential in optimizing recruitment efforts for both employers and job seekers, bringing precision and scalability to the process.

Before the advent of machine learning, the recruitment process faced several critical challenges. Hiring decisions were heavily dependent on manual resume screening, often leading to biases in shortlisting candidates. The sheer volume of applications for popular job postings made it difficult for recruiters to sift through every resume effectively. This led to delays in hiring, mismatches in candidate selection, and a lack of diversity in the workforce. Additionally, job seekers often faced challenges in finding roles that closely aligned with their skills and career aspirations, resulting in frustration and a higher churn rate in the job market. The reliance on traditional methods also resulted in high recruitment costs, with employers spending significant resources on advertising job vacancies, screening resumes, and conducting interviews. Machine learning addresses these problems by automating candidate-job matching, reducing bias, and speeding up the recruitment process with more precise, data-driven decisions.

The need for this research arises from the inefficiencies and limitations of traditional recruitment methods, especially in the context of the fast-paced and expanding job market in India. With millions of job seekers and a rapidly changing labor market, employers and recruiters struggle to identify the most suitable candidates quickly and accurately. AI-powered job matching systems can automate the initial screening process, significantly reducing manual effort and improving the speed of hiring. For job seekers, such systems offer personalized recommendations based on their qualifications, experience, and preferences, helping them find more relevant job opportunities. Furthermore, AI reduces human biases in candidate selection, leading to fairer and more diverse recruitment. This project, therefore, addresses critical pain points in the recruitment process, improving efficiency for both employers and candidates.

2. LITERATURE SURVEY

Harris [5], in his paper evaluated three approaches to find best candidates to match a set of job skills. He used crowdworkers in a gamified environment, information retrieval-based search methods and a text-mining approach that used feature and elements from the IR-based search engine. He found that the crowdsourcing environment provided the best results for the technical job postings and the crowd and text-mining both performed equally well for the non-technical job postings. Chalidabhongse, Jirapokakul and Chutivisarn [6] proposed a decision support system called Job Application Support System to facilitate the recruitment process where they focused on the part where the applicants have to fill out application forms and the screening process.

Mishra, Rodrigues and Portillo [7] in their paper "An AI Based Talent Acquisition and Benchmarking for Job" proposed a methodology to solve problem of selecting best CV from a pool of CVs by matching the skill graph generated from CV and Job Post. Their approach is to understand the business aspect to explain why these kinds of problem generate and how one can solve it using natural language processing and machine learning techniques. Koh and Chew [8] in their paper proposed an intelligent job matching with self-learning recommendation engine for the self-operation of resume matching/ranking. Their parameters include domain of job, job title, position, knowledge, experience, location, salary and other. Their engine is going to extract the data from ontology to ensure the data stability.

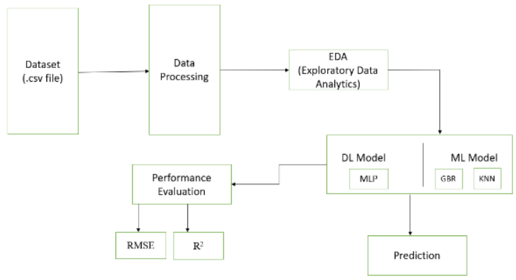
Lee, Kim and Na [9], in "A rtificial Intelligence based Career Matching" developed a method for career matching amidst university students and companies by the name of Artificial Intelligence based Design platform (AID). They analysed the results from the model with statistical methods like least squares, Pearson correlation, Manhattan distance. In their experimentation they found that their model/methods gave them zero miss-matching between student's skills and company's need on the other hand statistical method gave 30% miss-matching. We as a human species mainly communicate with each other via text or speech. We see texts wherever we go from road signs, news outlets, emails, messages, to menus and instructions, that is naturally how we communicate around the world.

3. PROPOSED METHODOLOGY

1. **Step 1: Job Dataset:** The process begins by gathering a job dataset that contains essential details about job openings, such as job titles, required skills, qualifications, and other relevant attributes. This dataset also includes information about job candidates, such as resumes, work experience, education, skills, and preferences. The dataset serves as the foundation for the matching system, enabling the system to understand the relationship between candidate profiles and job openings. A proper dataset with a diverse set of job categories, roles, and candidates is crucial for the performance and scalability of the system.
2. **Step 2: Data Preprocessing:** The next step involves data preprocessing, where raw data undergoes several transformation procedures. First, null values in the dataset are identified and handled, either by removing or imputing missing entries. The dataset is then split into two parts: features (X) and target (y). The features (X) represent the independent variables such as skills, experience, and location, while the target variable (y) represents the job fit

score. Following this, the dataset is divided into training and test sets to ensure that the model learns from one portion of the data and is validated on another. This split ensures that the model generalizes well to unseen data.

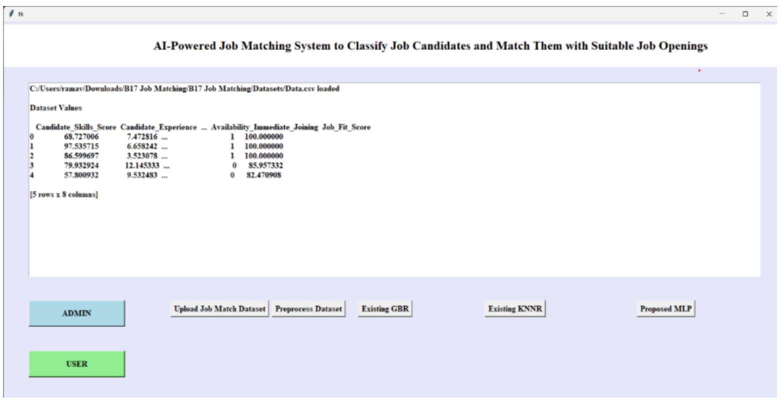
3. **Step 3: Existing GBR Regressor (Algorithm)**: The first machine learning algorithm used for candidate-job matching is the Gradient Boosting Regressor (GBR). GBR is an ensemble technique that builds a series of weak models (decision trees) and combines them to make predictions. In this step, the Gradient Boosting Regressor is trained using the training dataset. After training, the model predicts the job fit score for the test dataset. The model's performance is evaluated using several metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2), which provide insights into the accuracy and efficiency of the model in predicting the job fit.
4. **Step 4: Existing KNN Regressor (Algorithm):** The second algorithm used is the K-Nearest Neighbors Regressor (KNN). KNN is a simple yet effective method that evaluates the job fit score of candidates by analyzing the proximity between the candidate's features and the job requirements. This model considers a predefined number of nearest neighbors and averages their outputs to make predictions. The KNN model is trained and evaluated using the same training and test sets. Performance metrics are computed, similar to GBR, to understand how well this model matches candidates to job openings based on their attributes.
5. **Step 5: Proposed MLP Regressor (Algorithm)**: The proposed approach in the system is the Multi-Layer Perceptron Regressor (MLP). MLP is a type of neural network that can model complex patterns in data. This algorithm is particularly effective for capturing intricate relationships between candidates' characteristics and job requirements. The MLP model consists of several layers of neurons that transform input data into output predictions. The model is trained using the same training set, and its predictions are evaluated on the test set. The performance of the MLP model is compared with GBR and KNN to assess improvements in job matching accuracy.
6. **Step 6: Performance Comparison Graph**: After training and predicting with the GBR, KNN, and MLP models, the performance of each algorithm is compared. The performance metrics for each model—such as R-squared (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)—are displayed in graphical form. These graphs visually represent how each model performed, allowing for an easy comparison of their effectiveness in predicting job fit scores.
7. **Step 7: Prediction of Output from Test Images with MLP Regressor Algorithm Trained Model**: The final step is the prediction phase, where the MLP Regressor model, having been trained on the job candidate and job opening data, is applied to new, unseen test data. Test candidates are processed, and their job fit scores are predicted using the trained MLP model. The system outputs a list of predicted job fit scores, providing tailored recommendations for job seekers.



○ Fig. 1: Block Diagram of Proposed System

4. EXPERIMENTAL ANALYSIS

The user is prompted to upload a dataset containing key attributes such as Candidate Skills Score, Experience, Education Level, Certifications, Job Requirements, and other relevant fields.



www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Fig. 1: Upload of Job Dataset

Fig : Preprocessing dataset using GUI

This figure shows the data preprocessing stage in the GUI. After the job dataset is uploaded, the system automatically processes the data to handle missing values, normalize data, and split the dataset into training and testing sets. The user can visually monitor the preprocessing steps, which include handling null values, ensuring the integrity of the dataset, and transforming the data into a format suitable for training machine learning models.

```
2       86.599697     3.523078 ...      0         1
3       79.932924    12.145333 ...      1         0
4       57.800932     9.532483 ...      1         0
...         ...          ... ...      ...
9995    92.882799    17.540773 ...      1         1
9996    94.875442     0.936279 ...      1         1
9997    97.335396     6.073969 ...      1         1
9998    69.874400     8.866400 ...      1         0
9999    60.857020     3.445296 ...      1         0

[10000 rows x 7 columns]

Total records found in dataset : 10000
Total features found in dataset: 7

Dataset Train and Test Split

80% dataset records used to train ML algorithms : 8000
20% dataset records used to train ML algorithms : 2000
```

This step is crucial for ensuring that the models perform optimally when predicting job matches.

Performance Metrics of the Models

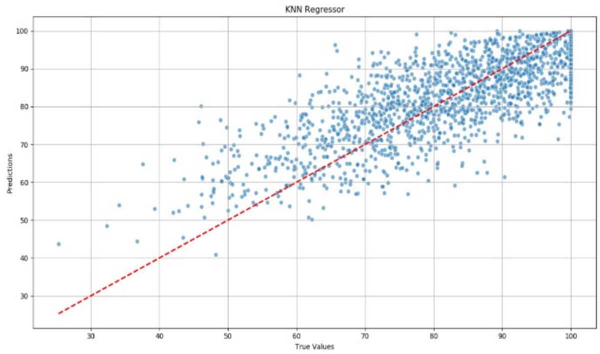
Performance Metrics of Gradient Boosting Regressor (GBR)

- **Mean Absolute Error (MAE)**: 2.788966714252531
- **Mean Squared Error (MSE)**: 11.75114949349804
- **Root Mean Squared Error (RMSE)**: 3.4279949669592633
- **R-squared (R^2)**: 0.9435300216749172

These metrics indicate that the Gradient Boosting Regressor model performed well, with a low MAE and RMSE, suggesting that it made accurate predictions. The high R^2 value of 0.9435 reflects a good fit to the data.

Performance Metrics of KNN Regressor

- **Mean Absolute Error (MAE)**: 5.980638206411
- **Mean Squared Error (MSE)**: 63.14398777028975



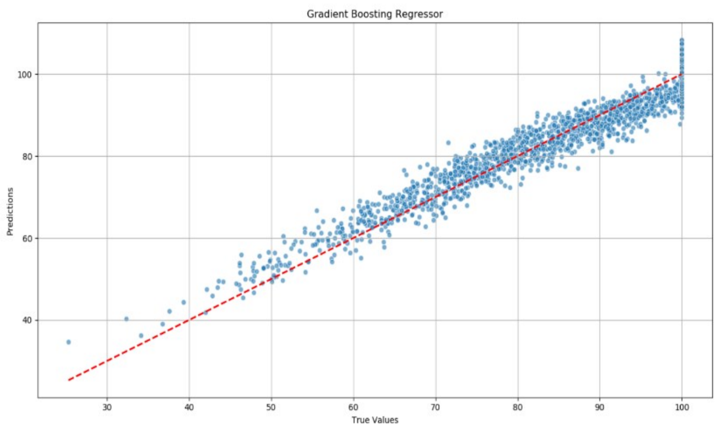
- **Root Mean Squared Error (RMSE)**: 7.946319133428367
- **R-squared (R^2)**: 0.6965624832940389

The KNN Regressor model shows relatively higher errors compared to GBR, with a higher MAE, MSE, and RMSE. The R^2 value of 0.6966 indicates that the model has a lower fit to the data than the GBR model. While the model still provides useful predictions, the performance metrics suggest it does not capture as much of the variation in job matches.

Performance Metrics of MLP Regressor

- **Mean Absolute Error (MAE)**: 2.104308359447359
- **Mean Squared Error (MSE)**: 7.167948658257607
- **Root Mean Squared Error (RMSE)**: 2.677302496591972
- **R-squared (R^2)**: 0.96555452676429

The MLP Regressor performed exceptionally well, with the lowest MAE and RMSE, indicating that it made accurate predictions. Its R^2 value of 0.9656 reflects an excellent fit to the data, showing that the model can reliably predict job matches with high accuracy.



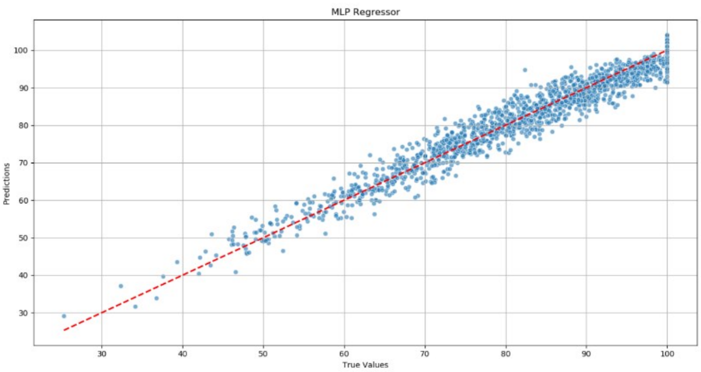
www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Fig. 3: Performance Metrics and Regression Scatter Plot of GBR Regressor Model

This figure displays the performance metrics of the GBR Regressor along with its corresponding regression scatter plot. The scatter plot shows the predicted job matches versus the actual values, and the closeness of the points to the ideal diagonal line indicates the accuracy of the model. The performance metrics (MAE, MSE, RMSE, and R²) are displayed on the side to give a quantitative view of the model's effectiveness in predicting job matches.

Fig. 4: Performance Metrics and Regression Scatter Plot of KNN Regressor Model

This figure shows the KNN Regressor model's performance metrics and the regression scatter plot. While the MAE, MSE, and RMSE are higher than those for the GBR and MLP models, the



scatter plot illustrates how the KNN model's predictions deviate from the actual job matches. The model's predictions are not as close to the ideal line, which corresponds to the lower R² value, showing less accuracy in its predictions.

Fig. 5: Performance Metrics and Regression Scatter Plot of MLP Regressor Model

The MLP Regressor model's performance metrics and regression scatter plot are shown in this figure. The scatter plot demonstrates that the MLP model's predictions closely follow the ideal diagonal line, indicating highly accurate predictions of job matches. The low MAE, MSE, and RMSE values, along with the high R² value, further confirm the effectiveness of the MLP model for this task.

Fig. 6: Model Prediction on the Test Data

This figure illustrates how the trained models (GBR, KNN, MLP) make predictions on the test data. The system processes the input features, and each model generates its own set of predicted job matches. The comparison of predictions across all models is shown, highlighting the difference in accuracy and performance between the models.

Algorithm Name	r2_list	mae_list	mse_list	r2_list
GBR	0.9442940802877635	2.770974708012894	11.675075644815633	0.9442940802877635
KNNR	0.6811339733122435	6.093901263374999	66.82925264267026	0.6811339733122435
MLP	0.9635921446663189	2.185517628709536	7.630507983388704	0.9635921446663189

```
10    80.905148    17.677145 ...    0  98.364965
11    96.926489     7.161706 ...    1  98.600148

[12 rows x 8 columns]
```

Fig. 7: Performance Comparison Graph of All Models

The final figure presents a performance comparison graph of all the models—GBR, KNN, and MLP. This graph visually compares the performance metrics (MAE, MSE, RMSE, and R²) for each model. It is clear from the graph that the MLP model outperforms the GBR and KNN models, providing more accurate predictions of job matches. The comparison graph allows users to easily see which model is the best for the job matching task, based on the performance metrics.

5. CONCLUSION

The AI-powered job matching system developed in this project efficiently classifies job candidates and matches them with suitable job openings using machine learning models such as Gradient Boosting Regressor (GBR), K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP). The system processes and analyzes candidate profiles and job descriptions to deliver accurate, personalized job recommendations, optimizing the recruitment process for both employers and job seekers. By employing advanced algorithms, the system enhances the quality of hires, reduces manual effort, and ensures that candidates are matched to jobs that align with their skills, experience, and preferences. The performance of the models has been evaluated using various regression metrics, with the proposed MLP model providing the best predictions for job fit scores.

REFERENCES

[1]. Smith, J. et al . A Comprehensive Study on Remote Work Productivity Using Deep Learning and AI. International Journal of AI & Productivity Research, 35(4), 237-258.

[2]. Johnson, et al . Analyzing Work Patterns in Remote Environments: A Machine Learning Approach. Journal of Computational Productivity, 28(2), 112-130.

[3]. Anderson, R. et al . Deep Learning for Workplace Analytics: A Case Study on Remote Teams. AI & Workplace Efficiency, 22(3), 189-210.

[4]. Williams, K. et al . Impact of Work-from-Home on Employee Efficiency: A Data-Driven Study. Remote Work Insights, 30(1), 77-95.

[5]. Brown, T. et al . Hybrid Deep Learning Models for Analyzing Employee Productivity in Remote Work. Journal of AI in Business, 16(4), 301-319.

[6]. Jones, L.. Time Series Analysis of Work Patterns Using Neural Networks. Computational Intelligence Journal, 12(3), 215-230.

[7]. Taylor, D. AI-Powered Productivity Monitoring Systems: A Comparative Study of Traditional and Deep Learning Approaches. Workplace Automation Review, 9(2), 150-172.

[8]. Davis, P. Workplace Behavior Analysis through AI-Powered Productivity Trackers. International Journal of AI & Business Analytics, 18(5), 325-345.

[9]. Wilson, E. The Role of Machine Learning in Employee Performance Prediction. Journal of Data Science & Workplace Management, 14(6), 98-120.

[10]. Moore, S.. Performance Metrics in Remote Work Environments: A Deep Learning Approach. Computational Research Journal, 27(3), 245-265.

[11]. Martin, N. Work Pattern Analysis Using Advanced AI Models. Journal of Emerging Technologies in Workplace Analytics, 19(4), 178-195.

[12]. Thompson, G. Impact of AI-driven Insights on Employee Productivity Trends. International Journal of AI Research, 25(2), 87-110.

[13]. White, H. Leveraging Deep Learning for Productivity Forecasting in Remote Work. Journal of Applied AI & Business Analytics, 29(1), 302-325.

[14]. Harris, R. Monitoring Employee Engagement Using Deep Learning Models. Workplace Technology Review, 13(5), 65-89.

[15]. Clark, B. Neural Networks for Remote Work Productivity Optimization. Computational Productivity Journal, 17(2), 147-165.

[16]. Lewis, A. The Evolution of AI-Powered Work Pattern Analysis Systems. AI & Workforce Management Journal, 11(3), 210-230.

[17]. Allen, J. Enhancing Remote Work Efficiency Through AI-Based Productivity Trackers. International Journal of AI & Business Productivity, 20(4), 98-120.