

AI Predictive Model to Suggest the Right Candidate for a Job Profile in HR

Divya Laxman Koli¹ and Ms. Yogita Kadbane²

¹Student, Master of Computer Applications,

²Assistant Professor, Master of Computer Application,
Sadhu Vaswani Institute of Management Studies for Girls, Pune
Savitribai Phule Pune University, Pune, Maharashtra, India.

Abstract: *Selecting a suitable candidate for a job can take time when many candidate and job details have to be considered together. Academic performance, skills, certifications, internship experience, job requirements and other factors may all influence whether a candidate is a suitable match. This project develops a machine learning based candidate-job matching system to support this process.*

The project uses a Skill-Job Matching Dataset containing 2,809 records and 22 columns. A stratified sample of 150 candidate-job records was used for the experiment, containing 102 non-matches and 48 matches. After removing Student_ID and Job_ID from model inputs, 19 features were used. Numerical features were standardized and categorical features were one-hot encoded.

Four classification models were tested: Logistic Regression, Decision Tree, Random Forest and Gradient Boosting. Random Forest achieved the highest accuracy of 76.67% and precision of 100%, but its recall was 30.00%. The notebook's final candidate recommendation pipeline uses Logistic Regression, which achieved 70.00% accuracy, 54.55% precision, 60.00% recall and 57.14% F1-score.

The final system calculates a match probability and converts it into three recommendation levels: Highly Recommended, Recommended and Not Recommended. In the 30-record test set, the recommendation output contained 4 highly recommended, 7 recommended and 19 not recommended candidates. The project demonstrates how machine learning can assist HR-style candidate-job matching and ranking.

Keywords: Candidate Recommendation, Job Matching, Human Resources, Machine Learning, Logistic Regression, Random Forest, Predictive Model.

I. INTRODUCTION

Recruitment involves comparing candidate qualifications with the requirements of available job roles. In a manual process, HR staff may need to examine academic performance, skills, experience, certifications, vocational background, job title, location and required skills. When the number of candidates increases, this comparison can become repetitive and time-consuming.

Machine learning can learn patterns from previous candidate-job matches and predict whether a new candidate-job pair is likely to be a match. Such a system can be used as a decision-support tool rather than as a replacement for an HR professional.

This project focuses on a candidate-job matching problem. The target variable is Job_Match, where 1 represents a match and 0 represents a non-match. The system also produces a match probability so that candidates can be ranked rather than viewed only as a binary result.



II. PROBLEM STATEMENT

The project aims to develop a machine learning based predictive model that can estimate whether a candidate is suitable for a given job profile and provide an understandable recommendation level for HR-oriented candidate screening.

III. OBJECTIVES

- Study candidate and job-related data used for matching.
- Preprocess numerical and categorical features for machine learning.
- Train and compare multiple classification algorithms.
- Evaluate the models using accuracy, precision, recall and F1-score.
- Select a practical prediction pipeline for candidate recommendation.
- Generate match probabilities for candidate-job pairs.
- Group predictions into Highly Recommended, Recommended and Not Recommended categories.
- Provide outputs that can support HR candidate screening and ranking.

IV. LITERATURE REVIEW

Machine learning has been applied to recruitment and HR analytics for tasks such as candidate screening, employee prediction and recommendation. Classification models can learn relationships between candidate attributes and historical outcomes. Logistic Regression provides a relatively simple classification approach, while tree-based ensemble methods can capture more complex patterns.

A candidate recommendation system can be useful when it ranks applicants according to a predicted probability instead of providing only a yes/no decision. However, the prediction should be treated as decision support because the training data and selected features determine what the model learns.

V. RESEARCH GAP

A basic job-matching process may stop after calculating whether a candidate matches a job. This project adds probability-based ranking and recommendation categories so the output can be more useful for screening. The notebook also compares several machine learning algorithms before producing the final recommendation output.

VI. RESEARCH METHODOLOGY

The project follows a supervised machine learning workflow. The dataset is first loaded and checked for missing values and class distribution. A stratified sample of 150 records is then created for the experiment. Identifier columns are removed from the model input, the target is separated, and the data is divided into training and testing sets.

Overall process: Skill-Job Dataset → Data Checking → Stratified 150-Record Sample → Remove ID Columns → Feature/Target Separation → Train/Test Split → Preprocessing → Model Training → Model Comparison → Final Recommendation → Candidate Ranking.

6.1 Dataset

The original Skill_Job_Matching_Dataset.csv contains 2,809 records and 22 columns. The columns include Student_ID, Age, Gender, Vocational_Program, Academic_Performance, Certifications_Count, Internship_Experience, five candidate skill fields, Job_ID, Job_Title, five required-skill fields, Min_Experience_Months, Location and Job_Match.

The dataset contains 1,918 non-match records (68.28%) and 891 match records (31.72%). The notebook reports no missing values in any of the 22 columns.

6.2 Experimental Sample

A stratified sample of 150 records was selected using random_state=42. The sample contains 102 non-match records (68%) and 48 match records (32%).



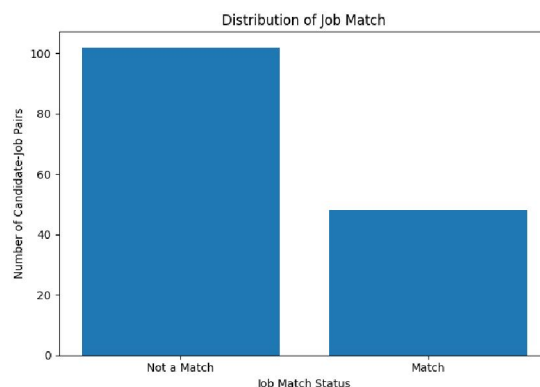


Figure 1. Job match distribution in the 150-record experimental sample.

6.3 Model Features

Student_ID and Job_ID were removed because the notebook treats them as identifiers rather than predictive inputs. The resulting model dataset contains 19 features and one target variable, Job_Match.

VII. DATA PREPROCESSING

7.1 Numerical and Categorical Features

The numerical features are Age, Academic_Performance, Certifications_Count, Internship_Experience, Skill_1 to Skill_5, Required_Skill_1 to Required_Skill_5 and Min_Experience_Months. The categorical features are Gender, Vocational_Program, Job_Title and Location.

The preprocessing pipeline standardizes numerical features using StandardScaler. Categorical features are converted into numerical form using OneHotEncoder with handle_unknown='ignore'. The transformed features are then passed to each classification model.

7.2 Train-Test Split

The 150 records were divided using an 80:20 stratified split. The training set contains 120 records and the testing set contains 30 records. The training target distribution is 82 non-match and 38 match records, while the test set contains 20 non-match and 10 match records.

VIII. MACHINE LEARNING MODELS

Logistic Regression: final recommendation pipeline, with balanced class weights and max_iter=1000.

Decision Tree: maximum depth 5 and balanced class weights.

Random Forest: 100 trees, maximum depth 5 and balanced class weights.

Gradient Boosting: 100 estimators, learning rate 0.05 and maximum depth 3.

8.1 Model Evaluation

The models were compared using accuracy, precision, recall and F1-score.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	70.00%	54.55%	60.00%	57.14%
Decision Tree	63.33%	45.45%	50.00%	47.62%
Random Forest	76.67%	100.00%	30.00%	46.15%
Gradient Boosting	63.33%	33.33%	10.00%	15.38%



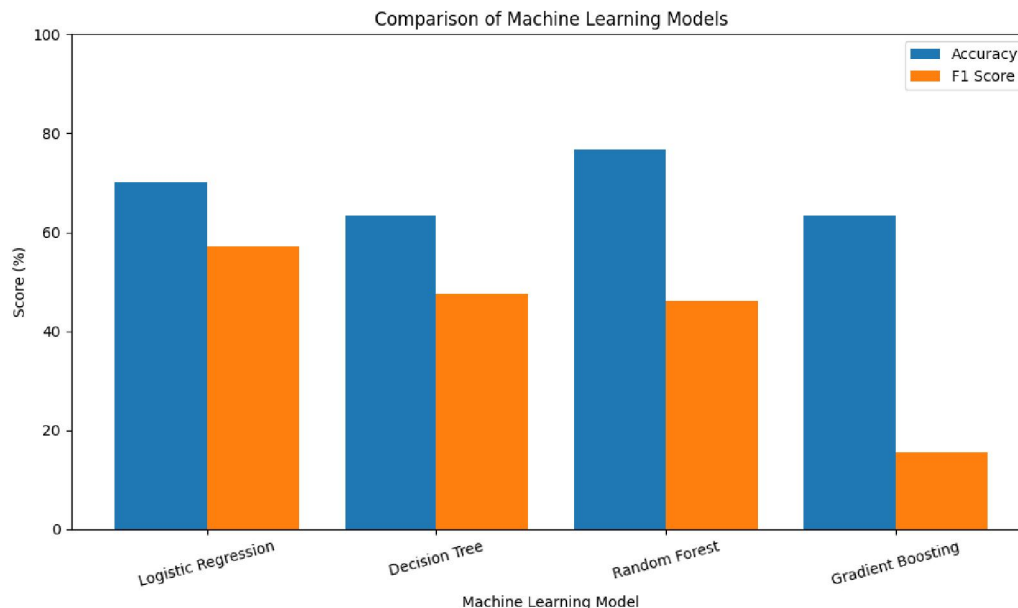


Figure 2. Comparison of accuracy and F1-score across the tested models.

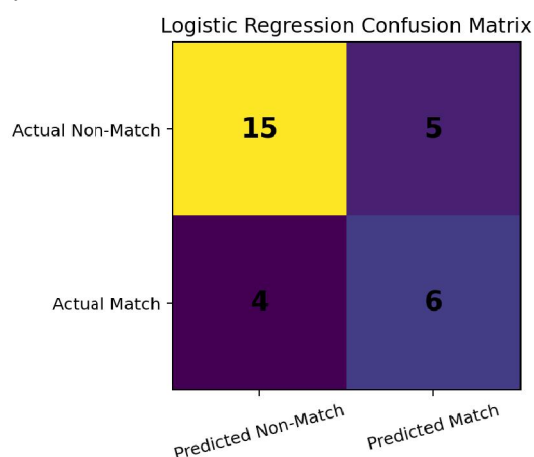
IX. MODEL SELECTION AND PREDICTION

The notebook's final project check identifies Logistic Regression as the selected model and uses it to generate candidate recommendation results. Its test accuracy is 70.00%, precision 54.55%, recall 60.00% and F1-score 57.14%.

Random Forest has the highest accuracy in the comparison at 76.67%, but its recall is 30.00%. The notebook does not state a separate formal reason for selecting Logistic Regression, so this paper reports the selection as implemented rather than adding an unsupported reason.

9.1 Confusion Matrix

The Logistic Regression confusion matrix is $\begin{bmatrix} 15 & 5 \\ 4 & 6 \end{bmatrix}$. Thus, 15 of 20 actual non-match records and 6 of 10 actual match records were correctly classified.



Confusion matrix derived from the Logistic Regression test results.



9.2 Match Probability

The final recommendation output uses Logistic Regression's predict_proba method to calculate the probability of Job_Match = 1. Candidate-job pairs are sorted from highest to lowest probability.

9.3 Recommendation Levels

Probability $\geq 75\%$ is Highly Recommended; 50% to below 75% is Recommended; below 50% is Not Recommended.

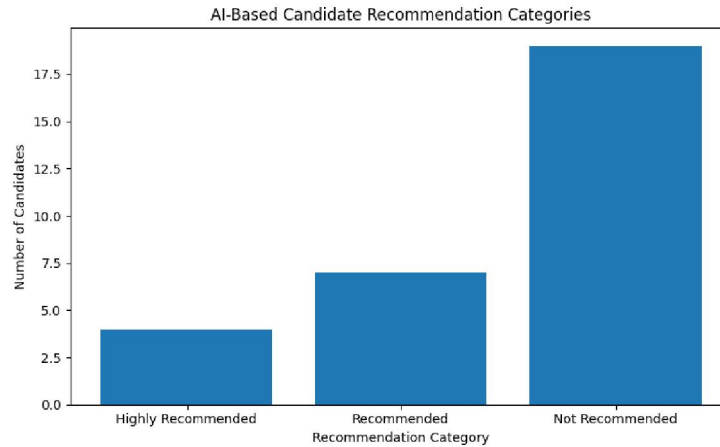


Figure 3. Recommendation categories in the 30-record test output.

X. CANDIDATE RECOMMENDATION RESULTS

The HR recommendation output combines Student_ID, Job_ID, Job_Title, Location, Academic_Performance, Vocational_Program, actual match, predicted match, match probability and recommendation information. Results are sorted by probability.

The notebook reports 4 Highly Recommended, 7 Recommended and 19 Not Recommended candidates among the 30 test records.

Examples of the highest displayed probabilities include S2133 – Electrical Technician at 99.93%, S0683 – IT Support Technician at 99.26%, S2495 – Welding Operator at 98.90% and S1793 – Plumbing Assistant at 75.86%.

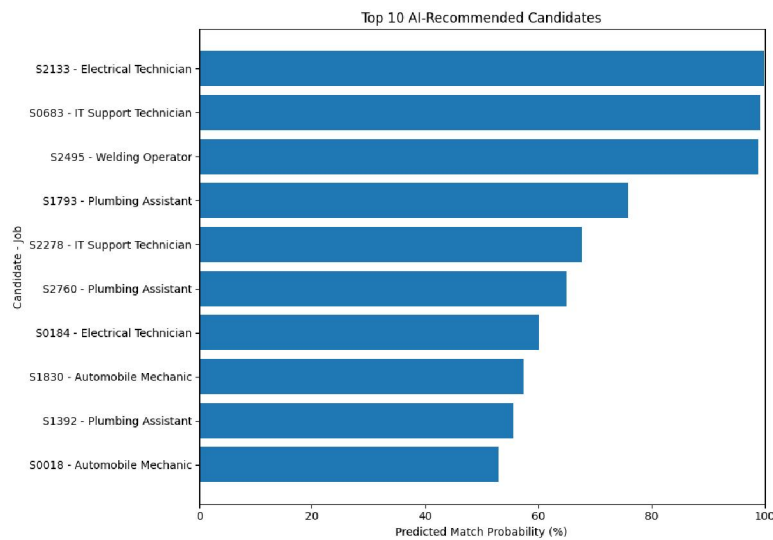


Figure 4. Highest displayed candidate-job match probabilities from the notebook output.



Top 10 candidate-job recommendations from the final Logistic Regression output.

Student ID	Job ID	Job Title	Location	Probability	Recommendation
S2133	J149	Electrical Technician	Chennai	99.93%	Highly Recommended
S0683	J038	IT Support Technician	Pune	99.26%	Highly Recommended
S2495	J203	Welding Operator	Pune	98.90%	Highly Recommended
S1793	J073	Plumbing Assistant	Pune	75.86%	Highly Recommended
S2278	J219	IT Support Technician	Bangalore	67.70%	Recommended
S2760	J039	Plumbing Assistant	Mumbai	65.10%	Recommended
S0184	J257	Electrical Technician	Pune	60.12%	Recommended
S1830	J018	Automobile Mechanic	Kolkata	57.41%	Recommended
S1392	J105	Plumbing Assistant	Chennai	55.57%	Recommended
S0018	J146	Automobile Mechanic	Mumbai	53.01%	Recommended

10.1 Interpretation

A high match probability means that the trained Logistic Regression model estimates a higher likelihood of the Job_Match = 1 class. The recommendation level is a presentation rule for this probability and does not independently verify a candidate's suitability.

XI. PRACTICAL USE

The system can support HR screening by ranking candidate-job pairs and highlighting high-probability matches. HR staff can then review the recommended candidates and make the final decision using additional information.

XII. GRAPHS AND VISUAL ANALYSIS

The notebook already contains visualization code for model comparison, job-match distribution, top candidate probabilities and recommendation categories. These outputs can be used directly in the research paper and presentation.

12.1 Model Comparison

Random Forest has the highest reported accuracy (76.67%), while Logistic Regression has the highest recall among the four tested models (60.00%) and is used for the final recommendation output.

12.2 Job Match Distribution

The 150-record sample contains 102 non-match and 48 match records, showing that the sample is imbalanced toward the non-match class.

12.3 Dashboard Status

The notebook creates a dashboard_summary table and exports Dashboard_Summary.csv. It contains total dataset records, experimental sample, training/testing records, job matches, no matches and recommendation counts. However, there is no complete visual dashboard application with cards, filters and interactive candidate search.

Recommended dashboard: Total Records card; Model Accuracy card; Match vs No-Match chart; Model Comparison chart; Recommendation Category chart; Top Candidate table; and filters for Job Title, Location and Recommendation Level.

13. Dashboard Implementation

For a college demonstration, Streamlit is a simple option. Load the saved CSV/model, show metric cards at the top, place the charts in the middle, and put the ranked candidate table below. Add dropdown filters for job title, location and recommendation level. If the project remains in Colab, the existing graphs can be displayed there, but a dashboard application will look more complete.



XIV. LIMITATIONS

The experiment uses a 150-record stratified sample from the 2,809-record dataset, so the reported performance is based on this sample and its 30-record test set. Model performance varies across algorithms. Random Forest has 100% precision but 30% recall, while the final Logistic Regression pipeline has 60% recall.

The model only learns from the variables in the dataset. Interview performance, communication skills, references, behavioural assessment and recruiter judgment are not represented. The match probability is therefore a machine learning estimate and should not be treated as a guaranteed hiring outcome.

XV. DISCUSSION

The results show that different algorithms behave differently on the same candidate-job matching problem. Random Forest gives the highest accuracy and precision in the reported experiment while Logistic Regression provides the final probability-based candidate ranking. The recommendation categories make the output easier to read, but the system should remain a decision-support tool.

XVI. PROPOSED DASHBOARD FLOW

Load Model → Select or enter candidate/job information → Calculate Match Probability → Show Recommendation Level → Display Top Candidates → Apply filters → Export recommendation table.

XVII. CONCLUSION

This project developed a machine learning based candidate-job matching system using a Skill-Job Matching Dataset. The original dataset contains 2,809 records and 22 columns, while a stratified sample of 150 records was used for the experiment. Nineteen predictive features were used after removing Student_ID and Job_ID.

Four classification algorithms were compared. Logistic Regression achieved 70.00% accuracy, 54.55% precision, 60.00% recall and 57.14% F1-score. Decision Tree achieved 63.33% accuracy, Random Forest achieved the highest accuracy of

76.67% with 100% precision and 30% recall, and Gradient Boosting achieved 63.33% accuracy.

The final notebook uses Logistic Regression for candidate recommendation. It calculates match probabilities, sorts candidate-job pairs and assigns recommendation levels. In the 30-record test output, 4 candidates were Highly Recommended, 7 were Recommended and 19 were Not Recommended.

Overall, the project demonstrates how machine learning can support initial HR candidate screening and job matching. A visual dashboard is the main presentation improvement needed to bring the model comparison, match distribution and candidate ranking together in one place.

XVII. FUTURE SCOPE

Future work can include evaluating the models on the complete dataset, using cross-validation, testing additional algorithms, improving recall for actual matches, adding more candidate and job attributes, applying explainable AI methods, adding recruiter feedback, creating a web-based dashboard and testing the system with larger real-world recruitment data.

REFERENCES

1. D. W. Hosmer, S. Lemeshow and R. X. Sturdivant, Applied Logistic Regression, Wiley.
2. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
3. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
4. T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, Springer.

