IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

Candidate Joining Prediction Using Random Forest and KNN

Monika Priya MK[*1] and Prof. Kusuma N[*2]
[*1]Student, Department of Computer Applications
[*2]Assistant Professor, Department of Computer Applications
Maharaja Institute of Technology Mysore, Mandya, Karnataka, India.

Abstract: *This paper presents a web-based machine-learning system for predicting whether a selected candidate is likely to join an organization after accepting an employment offer. The study uses 8,995 historical recruitment records with fifteen candidate- and offer-related factors, including notice period, offer-acceptance duration, compensation changes, relocation, candidate source, experience, location, and age. Categorical attributes are encoded and the dataset is divided using a stratified 80:20 train-test split. Random Forest and K-Nearest Neighbors (KNN) classifiers are trained and compared. The Random Forest model, configured with 250 trees and balanced subsampling, achieves 82.27% accuracy with a recorded training time of 3.12 seconds, while the optimized KNN model achieves 80.71% accuracy with a training time of 15.21 seconds. The selected Random Forest model is integrated with a Flask, MySQL, Jinja, Bootstrap, and pandas web application that supports administrator and HR-user workflows, dataset viewing, prediction, model comparison, graphs, and query management. Although the system demonstrates successful end-to-end integration, the confusion matrix shows low recall for the Not Joined class. Therefore, the output is intended as decision support for recruiter follow-up rather than an automated employment decision.*

Keywords: Candidate joining prediction, human resource analytics, Random Forest, KNN, recruitment, Flask, machine learning

1. INTRODUCTION

Recruitment outcomes are not fully realized when an offer letter is issued or accepted. Organizations obtain operational value only when the selected candidate reports for duty and enters onboarding. During the interval between acceptance and joining, candidates may receive competing offers, reconsider relocation, face delays in release from a current employer, renegotiate compensation, or decide that the position does not meet their expectations. These events create a measurable difference between offers released and candidates who actually join.

Traditional tracking practices depend on spreadsheets, status labels, recruiter judgement, and repeated follow-up calls. These methods support coordination, but they do not systematically learn from historical patterns. Machine learning can transform past recruitment records into a classification model that estimates the likely joining outcome of a new candidate based on a combination of offer, compensation, experience, relocation, source, location, and demographic factors [1], [2].

The proposed Candidate Joining Prediction System combines predictive analytics with a practical web application. Authenticated HR users can inspect the dataset and prediction factors, enter a candidate record, receive a Joined or Not Joined result, compare Random Forest and KNN models, view performance graphs, and communicate with an administrator. The system is intended to help recruiters prioritize follow-up and workforce planning; it is not intended to automatically reject, cancel, or deprioritize candidates.

DOI: 10.48175/IJARSCT-37781

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

II. RELATED WORK

Machine-learning research in recruitment has addressed candidate screening, placement success, talent matching, and prescriptive workforce decisions. Reddy et al. demonstrated recruitment prediction using candidate data [1], while Pessach et al. combined machine learning with mathematical programming for recruitment decision support [2]. Reviews of digital personnel selection emphasize the need to evaluate validity, usability, and applicant reactions rather than considering predictive accuracy alone [3].

Algorithmic employment systems may reproduce bias contained in historical data. Raghavan et al. examined how hiring tools are designed and validated, highlighting the risk of unsupported fairness claims [4]. Subsequent research has emphasized transparency, privacy, accountability, human rights, and human supervision in AI-enabled recruitment [5]-[10]. These concerns are especially relevant when the dataset contains age, gender, location, source, or related proxy attributes.

A practical gap remains between model-only experiments and role-based web systems focused specifically on the post-offer joining stage. The present work addresses this gap by integrating two supervised classifiers with a Flask and MySQL application. The system presents measurable model performance while preserving the HR employee as the final decision-maker.

III. METHODOLOGY

The methodology follows a complete machine-learning application lifecycle: data understanding, preprocessing, model training, evaluation, artifact serialization, web integration, and testing. Consistency between training and deployment is treated as a primary requirement. The feature definitions, category encoders, scaling objects, feature order, and classifier used in the web route must match the objects produced during training.

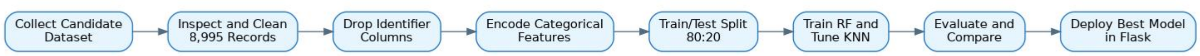

Figure 1: Proposed candidate-joining prediction methodology workflow

A. Dataset and Preprocessing

The source workbook contains 8,995 records and 18 columns. The identifier fields SLNO and Candidate Ref are removed. The remaining data contains fifteen input factors and one target variable, Result. The target distribution is 7,313 Joined records and 1,682 Not Joined records, corresponding to approximately 81.30% and 18.70%, respectively. No missing values are present in the submitted dataset.

Categorical input columns are transformed using saved label encoders, while numerical values are converted to floating-point form. The data is split into 80% training and 20% testing sets using random_state 42 and stratification. KNN inputs are standardized because distance-based classification is sensitive to feature scale. A production implementation should move all preprocessing into a Pipeline or ColumnTransformer and fit each transformation only on training folds.

B. Model Training

The Random Forest classifier uses 250 estimators, random_state 42, unrestricted depth, bootstrap sampling, and class_weight=balanced_subsample. Random Forest can model nonlinear interactions and mixed feature effects by aggregating predictions from many decision trees. The KNN model is optimized using five-fold GridSearchCV across neighbor counts of 3, 5, 7, 9, and 11; uniform and distance weighting; and Euclidean and Manhattan distance. The selected KNN configuration uses 11 neighbors, distance weighting, and Manhattan distance.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The application follows an MVC-like layered structure. A web browser forms the presentation layer. Flask routes receive requests, validate input, control sessions, interact with MySQL or model artifacts, and render Jinja templates. MySQL stores administrator accounts, HR employee accounts, and employee queries. The Excel workbook and serialized machine-learning files remain separate from transactional application data.

DOI: 10.48175/IJARSCT-37781

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

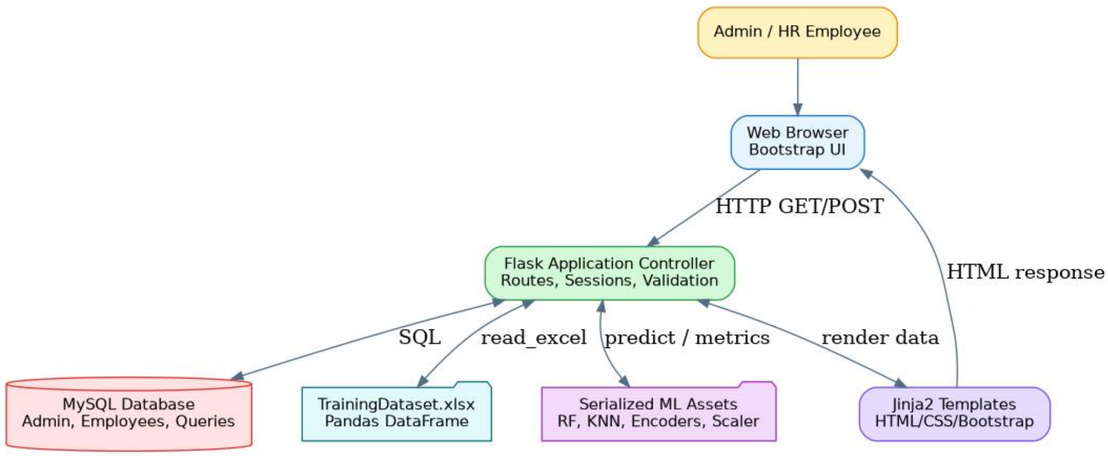

Figure 2: Overall architecture of the Candidate Joining Prediction System

Table 1. Tools and usage

Sl. No.	Technology	Purpose / Usage
1	Python	Model training, data processing, and application logic
2	Flask and Jinja2	Routing, sessions, validation, and dynamic templates
3	MySQL	Administrator, employee, and query data storage
4	pandas and openpyxl	Excel dataset loading and feature preparation
5	scikit-learn	Encoding, scaling, Random Forest, KNN, and evaluation
6	Bootstrap / HTML / CSS	Responsive user interface
7	Matplotlib and pickle	Graphs and model-artifact persistence

The administrator module supports login, employee account creation, viewing, editing, deletion, query replies, and password updates. The HR module supports login, dataset and factor viewing, candidate prediction, Random Forest and KNN result pages, comparison graphs, query posting, and password updates. Session checks restrict protected routes, and parameterized SQL is used for database operations. For production use, plaintext passwords and hard-coded secrets must be replaced with password hashing and environment-based configuration.

V. MODELING AND ANALYSIS

Table 2. Model configuration and performance

Model	Configuration	Accuracy	Training Time	Role
Random Forest	250 trees; balanced subsampling	82.27%	3.12 s	Deployed
KNN	k=11; distance; Manhattan	80.71%	15.21 s	Benchmark

Random Forest provides a 1.56 percentage-point accuracy advantage and a substantially lower recorded training time. It is therefore selected for the prediction route. Feature-importance analysis indicates that duration to accept the offer, offered CTC hike, expected CTC hike, CTC difference, age, relocation, relevant experience, and notice period are among the most influential model inputs. These values represent association with model decisions and should not be interpreted as causal effects.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 6, Issue 2, July 2026

Impact Factor: 8.2

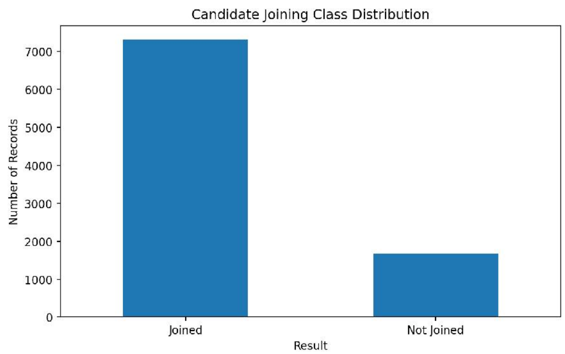

Figure 3: Candidate joining class distribution

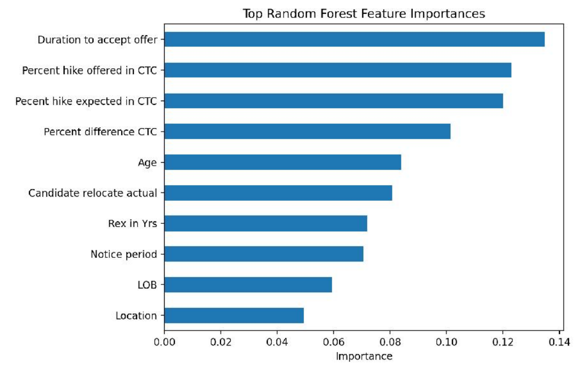

Figure 4: Random Forest feature importance

Class imbalance is a central evaluation issue. A classifier can achieve high overall accuracy by predicting the majority Joined class while failing to detect candidates at risk of not joining. Consequently, confusion matrix values, minority-class recall, precision, F1-score, calibration, and business costs should be considered alongside accuracy.

VI. APPLICATION SNAPSHOTS

The web interface converts the trained model into a usable HR decision-support workflow. The prediction page displays categorical dropdowns and numerical fields in the same feature order used during training. Input values are validated before encoding and inference. The comparison page presents model accuracy and training time and identifies the better submitted model.

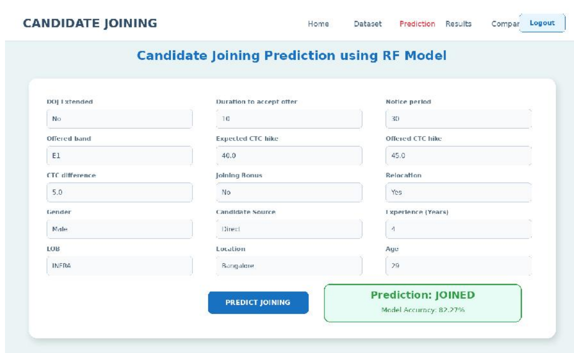

Figure 5: Candidate prediction interface

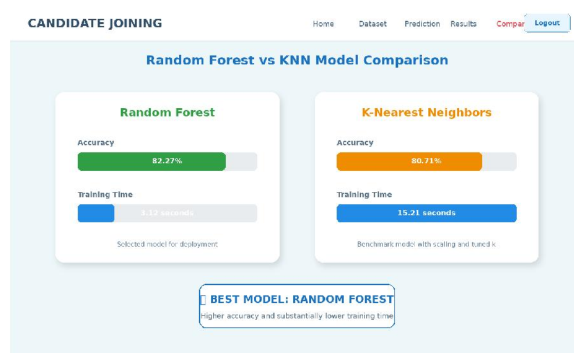

Figure 6: Random Forest and KNN comparison interface

The application also contains an administrator dashboard, HR dashboard, dataset table, factor list, result pages, graphs, and a two-way query module. This role-oriented interface is a key contribution because it converts an offline data-science experiment into a demonstrable application that supports routine HR review.

VII. RESULTS AND DISCUSSION

On the fixed test split of 1,799 records, the Random Forest confusion matrix contains 1,421 Joined records correctly classified as Joined, 42 Joined records classified as Not Joined, 59 Not Joined records correctly classified, and 277 Not Joined records classified as Joined. The resulting recall for the Not Joined class is approximately 17.56%. Thus, the model performs strongly on the majority class but misses many business-critical non-joining cases.

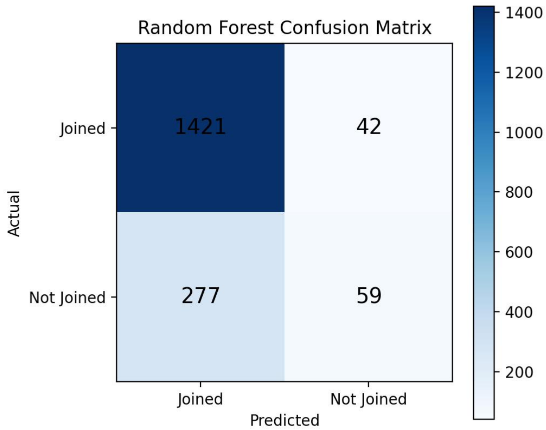

Figure 7: Random Forest confusion matrix

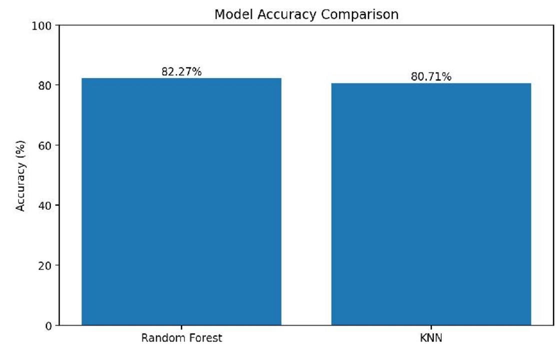

Figure 8: Accuracy comparison of the two models

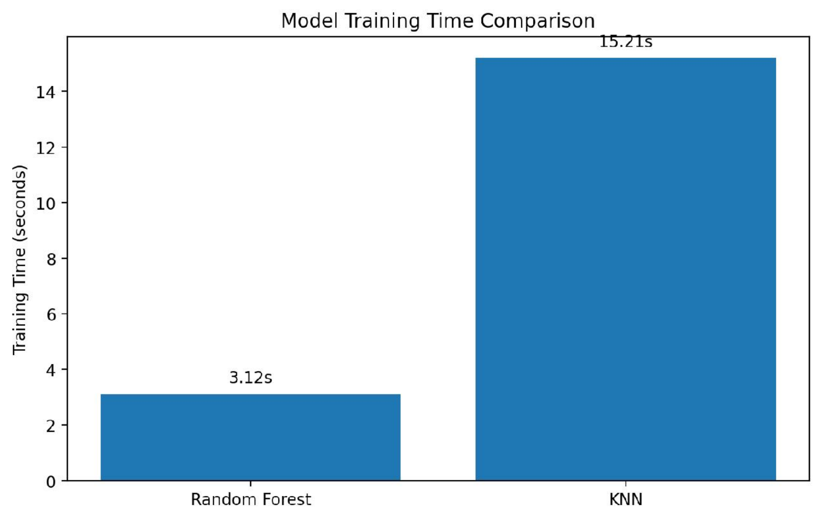

Figure 9: Recorded training-time comparison

The results demonstrate that Random Forest is a useful baseline for candidate-joining prediction and is more suitable than the submitted KNN model for interactive deployment. However, the current system should not be treated as production-ready. The class imbalance, limited minority recall, nominal label encoding, possible preprocessing leakage in the KNN workflow, and use of sensitive or proxy attributes require further validation. The software also requires password hashing, externalized secrets, CSRF protection, audit logging, production hosting, and route-level security review.

The appropriate operational use is risk-informed follow-up. HR employees may use the predicted outcome to review cases, verify circumstances, and prioritize communication. Human judgement and direct candidate interaction remain necessary. The system should not be used as the sole basis for employment-related actions, because algorithmic outputs can reflect historical patterns and bias [4], [6], [9], [10].

VIII. CONCLUSION AND FUTURE SCOPE

The Candidate Joining Prediction System demonstrates the end-to-end development of a machine-learning web application for an HR problem. Fifteen explanatory factors from 8,995 records are prepared for binary classification, Random Forest and KNN are trained and compared, and the better-performing Random Forest model is integrated into a role-based Flask and MySQL application. The model achieves 82.27% accuracy and provides immediate browser-based predictions, performance metrics, comparison graphs, and HR-administrator communication.

The study also shows why accuracy alone is insufficient. Low recall for the Not Joined class limits the model's ability to reliably identify joining-risk candidates. Future work should evaluate threshold optimization, class-weight tuning, SMOTE or balanced ensembles, gradient boosting models, calibrated probabilities, SHAP explanations, and group-wise fairness metrics. Preprocessing should be consolidated into a versioned pipeline, and sensitive features should be governed or removed where necessary. The application should also add secure authentication, audit trails, prediction history, outcome capture, automatic retraining, monitored deployment, and pilot evaluation with HR users.

ACKNOWLEDGEMENTS

The authors express their gratitude to the Department of Master of Computer Applications, Maharaja Institute of Technology Mysore, for providing the academic support and facilities required to complete this work. The authors also thank the faculty members, project guide, friends, and all individuals who contributed directly or indirectly to the development and documentation of the system.

REFERENCES

[1] D. J. M. Reddy, S. Regella, and S. R. Seelam, “Recruitment Prediction using Machine Learning,” Proc. 2020 5th Int. Conf. Computing, Communication and Security (ICCCS), pp. 1-4, 2020. DOI: 10.1109/ICCCS49678.2020.9276955.

[2] D. Pessach, G. Singer, D. Avrahami, H. Chalutz Ben-Gal, E. Shmueli, and I. Ben-Gal, “Employees Recruitment: A Prescriptive Analytics Approach via Machine Learning and Mathematical Programming,” Decision Support Systems, vol. 134, art. 113290, 2020. DOI: 10.1016/j.dss.2020.113290.

[3] S. A. Woods, S. Ahmed, I. Nikolaou, A. C. Costa, and N. R. Anderson, “Personnel Selection in the Digital Age: A Review of Validity and Applicant Reactions, and Future Research Challenges,” European Journal of Work and Organizational Psychology, vol. 29, no. 1, pp. 64-77, 2020. DOI: 10.1080/1359432X.2019.1681401.

[4] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, “Mitigating Bias in Algorithmic Hiring: Evaluating Claims and Practices,” Proc. 2020 Conf. Fairness, Accountability, and Transparency, pp. 469-481, 2020. DOI: 10.1145/3351095.3372828.

[5] E. Fumagalli, S. Rezaei, and A. Salomons, “OK Computer: Worker Perceptions of Algorithmic Recruitment,” Research Policy, vol. 51, no. 2, art. 104420, 2022. DOI: 10.1016/j.respol.2021.104420.

[6] A. L. Hunkenschroer and C. Luetge, “Ethics of AI-Enabled Recruiting and Selection: A Review and Research Agenda,” Journal of Business Ethics, vol. 178, no. 4, pp. 977-1007, 2022. DOI: 10.1007/s10551-022-05049-6.

[7] A. L. Hunkenschroer and A. Kriebitz, “Is AI Recruiting (Un)ethical? A Human Rights Perspective on the Use of AI for Hiring,” AI and Ethics, vol. 3, pp. 199-213, 2023. DOI: 10.1007/s43681-022-00166-4.

[8] P. N. Tinguely, J. Lee, and V. F. He, “Designing Human Resource Management Systems in the Age of AI,” Journal of Organization Design, vol. 12, no. 4, pp. 263-269, 2023. DOI: 10.1007/s41469-023-00153-x.

[9] A. Fabris et al., “Fairness and Bias in Algorithmic Hiring: A Multidisciplinary Survey,” ACM Transactions on Intelligent Systems and Technology, vol. 16, no. 1, 2025. DOI: 10.1145/3696457.

[10] C. Rigotti and E. Fosch-Villaronga, “Fairness, AI & Recruitment,” Computer Law & Security Review, vol. 53, art. 105966, 2024. DOI: 10.1016/j.clsr.2024.105966.

[11] M. Islam, M. M. Rahman, M. A. Taher, G. M. A. A. Quaosar, and M. A. Uddin, “Using Artificial Intelligence for Hiring Talents in a Moderated Mechanism,” Future Business Journal, vol. 10, art. 13, 2024. DOI: 10.1186/s43093-024-00303-x.

[12] S. M. U. Dadaboyev, J. Abdullayeva, N. Abbosova, A. Suleymenova, and K. Mamadjanova, “Role of Artificial Intelligence in Employee Recruitment: Systematic Review and Future Research Directions,” Discover Global Society, vol. 3, art. 99, 2025. DOI: 10.1007/s44282-025-00246-w.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, July 2026
Impact Factor: 8.2

[13] L. Roumbanis, "On the Present-Future Impact of AI Technologies on Personnel Selection and the Exponential Increase in Meta-Algorithmic Judgments," Futures, vol. 166, art. 103538, 2025. DOI: 10.1016/j.futures.2025.103538.

DOI: 10.48175/IJARSCT-37781