International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

# Employee Attrition Predictor: AI-Powered HR Analytics System for Employee Retention

Prof.Samirkumar Waghmare[1]¬,Nihal Khare[2], Rohit Gupta[3], Nikhil Gupta[4], Pushparaj Gautam[5]
[1]*Assistant Professor,* [2345]*Students,*
*Department of Computer Engineering,*
*Bharat College of engineering Affiliated to Mumbai University*
*Mumbai, Maharashtra, India*

*Abstract : Employee attrition poses a significant financial challenge for organizations, with turnover costs estimated at up to 200% of an employee's annual salary when accounting for recruitment, onboarding, and lost productivity. Traditional HR approaches rely on reactive measures, often identifying flight risks only after resignation. This paper presents a machine learning-based predictive system designed to proactively identify at-risk employees and support data-driven retention strategies.*

*The proposed system was developed and evaluated on a dataset of 1,470 employee records comprising 12 behavioral, demographic, and organizational features. Three supervised classification algorithms were implemented and compared: Logistic Regression, Random Forest, and Decision Tree. The pipeline incorporates automated data validation, categorical feature encoding, numerical normalization, and a risk scoring module for business impact quantification. Statistical analyses including chi-square tests and Pearson correlation were conducted to identify significant attrition drivers.*

*The system was deployed as an interactive web application using Streamlit with Docker containerization, making it accessible to non-technical HR stakeholders. This work demonstrates that interpretable machine learning models, combined with business intelligence modules, can deliver actionable and measurable value in workforce management.*

## I. INTRODUCTION

Human capital is widely regarded as one of the most valuable assets of any organization. The ability to attract, develop, and retain skilled employees directly influences organizational performance, innovation, and competitive advantage. However, employee attrition — the voluntary or involuntary departure of employees from an organization — remains a persistent and costly challenge across industries. Studies have consistently shown that replacing a single employee can cost anywhere between 50% to 200% of their annual salary, factoring in recruitment expenses, onboarding time, training costs, and the productivity loss during the transition period. For large organizations, the cumulative financial impact of unmanaged attrition can run into hundreds of millions of dollars annually.

However, a gap remains between academic research and practical deployment. Many existing studies focus solely on model accuracy without addressing the end-to-end pipeline required for real-world use — including data validation, feature engineering, risk scoring, business impact quantification, and accessible visualization for nontechnical stakeholders.

This paper addresses that gap by presenting a complete, production-ready employee attrition prediction system. The system was built using Python and scikit-learn, trained and evaluated on a dataset of 1,470 employee records, and deployed as an interactive web application using Streamlit with Docker containerization. Three machine learning models were developed and compared: Logistic Regression, Random Forest, and Decision Tree. Beyond prediction accuracy, the system incorporates a business intelligence module that translates model outputs into actionable financial insights, including risk scores, estimated turnover costs, and ROI projections for retention interventions.

## II. RELATED WORK

Application of machine learning to predict employee attrition has been an active area of research over the past decade. Early work in this domain relied primarily on statistical methods such as logistic regression and survival analysis to model the probability of employee departure. Logistic regression, in particular, became a baseline approach due to its interpretability and ease of implementation. Researchers found that factors such as job satisfaction, compensation, work-life balance, and years of experience were consistently significant predictors of voluntary turnover across different industries and organizational contexts.

As ensemble methods gained prominence in the machine learning community, studies began exploring their application to HR analytics. Random Forest classifiers were shown to outperform single decision trees and logistic regression in several attrition prediction tasks, primarily due to their ability to capture non-linear relationships and interactions between features without requiring explicit feature engineering. Gradient boosting methods such as XGBoost and LightGBM further improved predictive performance

in studies where larger and more complex datasets were available. These ensemble approaches also provided feature importance scores, which proved valuable for HR practitioners seeking to understand the underlying drivers of attrition rather than just the predictions themselves.

Beyond model development, a smaller but growing body of work has focused on the deployment and practical usability of attrition prediction systems. Several researchers have highlighted the disconnect between academic model development and real-world HR tool adoption, noting that most published systems lack user-facing interfaces, business impact quantification, or integration with existing HR information systems. Recent work has begun to address this by proposing end-to-end frameworks that combine predictive modeling with dashboards, risk scoring, and actionable recommendation engines. The present work builds on this direction by delivering a fully deployed, interactive system that bridges the gap between machine learning outputs and HR decision-making.

## III. SYSTEM ARCHITECTURE AND DESIGN

### A. *Dataset Description*

The dataset used in this study consists of 1,470 employee records sourced from an HR analytics dataset commonly used in workforce management research. Each record represents a single employee and contains 12 attributes spanning demographic information, job-related characteristics, compensation details, and behavioral indicators. The target variable is a binary label indicating whether the employee has left the organization (attrition = Yes) or remains employed (attrition = No). The dataset exhibits a class imbalance typical of real-world attrition scenarios, with approximately 16% of employees labeled as having left and the remaining 84% still active. This imbalance was accounted for during model training through class weight adjustment.

The dataset was verified to be complete with no missing values across any feature or record, eliminating the need for imputation strategies. All 1,470 records were retained for analysis without any row-level filtering. The features included in the dataset cover a broad range of factors known to influence employee retention, including age, monthly income, job satisfaction, overtime status, distance from home, years at the company, number of companies previously worked at, department, and job role.

### B. *Feature Description*

The 12 features used in this study can be grouped into three categories. Demographic features include Age, which represents the employee's current age as a continuous numerical variable, and DistanceFromHome, which captures the commuting distance in miles as a numerical value. Compensation and career features include MonthlyIncome, which records the employee's gross monthly salary, YearsAtCompany, which measures total organizational tenure, and NumCompaniesWorked, which reflects prior employment history and career mobility. Behavioral and satisfaction features include JobSatisfaction, an ordinal variable rated on a four-point scale from low to very high, OverTime, a binary indicator of whether the employee regularly works beyond standard hours, Department, a nominal categorical variable identifying the employee's organizational unit, and JobRole, a nominal variable specifying the employee's position title.

### C. *Exploratory Data Analysis*

Prior to model development, a comprehensive exploratory data analysis was conducted to understand the distribution of features and their relationship with the attrition target variable. Statistical tests were applied to quantify the significance of observed associations. A chi-square test of independence between overtime status and attrition yielded a test statistic of $\chi^2=24.267$ with $p<0.001$, confirming a highly significant association. Employees working overtime exhibited an attrition rate of 43.4% compared to 29.6% for those on regular hours, representing a relative increase of approximately 47%. A Pearson correlation analysis between distance from home and attrition produced a coefficient of $r=0.065$ with $p=0.013$, indicating a statistically significant though modest positive relationship. An analysis of job satisfaction levels revealed that employees reporting low satisfaction had an attrition rate of 47.1%, nearly double the 23.4% rate observed among employees reporting high satisfaction. Department-level analysis identified the Sales department as carrying the highest absolute attrition risk, with 116 employees classified as highrisk, followed by Research and Development and Human Resources.

## IV. IMPLEMENTATION

The system was implemented entirely in Python 3.8+, following an object-oriented, modular design where each functional concern is encapsulated in a dedicated class. The main orchestration class, EmployeeAttritionPredictor, coordinates the full pipeline by sequentially invoking the five core modules: DataValidator, FeatureEncoder, EDAEngine, ModelTrainer, and RiskAssessor. This separation of concerns ensures that each component can be independently tested, replaced, or extended without affecting the rest of the system.

### A. *Data Validation Module*

The DataValidator class implements four layers of validation applied before any data enters the processing pipeline. Schema validation checks that all required columns are present and that each column conforms to its expected data type.

Completeness checking computes the proportion of records with no missing values and returns a completeness score normalized between 0 and 100. Outlier detection applies the interquartile range method to five numerical columns — Age, DistanceFromHome, MonthlyIncome, NumCompaniesWorked, and YearsAtCompany — flagging any values falling below Q1 − 1.5×IQR or above Q3 + 1.5×IQR. Business rule validation enforces domain-specific constraints such as requiring employee age to fall between 18 and 65, monthly income to be strictly positive, years at company to not logically exceed age minus 16, and attrition labels to be restricted to the values "Yes" or "No". The module returns a structured ValidationResult dataclass containing a boolean validity flag, a list of errors, a list of warnings, and a summary dictionary, enabling downstream components to make informed decisions about data quality before proceeding.

### *B. Feature Engineering Module*

The FeatureEncoder class implements a strategy-based encoding pipeline where each column is assigned an encoding strategy — ordinal, one-hot, binary, or numerical — based on configuration defined in the YAML file. The fit_transform method fits all encoders on the training data and applies the transformations in a single pass, while the separate transform method applies the already-fitted encoders to new data without refitting, preventing data leakage. Ordinal encoding maps the JobSatisfaction column to integer values 1 through 3 using a fixed dictionary mapping, preserving the natural ordering of satisfaction levels. One-hot encoding expands nominal columns such as Department and JobRole into binary indicator columns, one per unique category value. Binary encoding maps the OverTime column from "Yes"/"No" string values to 1 and 0 respectively. Numerical standardization subtracts the column mean and divides by the standard deviation, computed exclusively from the training set. A column alignment step at the end of the transform method ensures that any categories present in training but absent in new data are represented as zero-filled columns, maintaining a consistent feature vector shape across all inputs. Fitted encoders are serialized to disk using Python's pickle module, allowing the web application to load them at startup and apply consistent transformations to new employee records without retraining.

### *C. Model Training Module*

The ModelTrainer class provides a unified interface for training, evaluating, and comparing three classification algorithms. Each algorithm is wrapped in a dedicated subclass — LogisticModel, RandomForestModel, and DecisionTreeModel — all inheriting from a common BaseModel class that implements shared functionality including fit, predict, predict_proba, and get_feature_importance methods. This inheritance structure allows the comparison and evaluation logic to operate on any model through a consistent interface without requiring algorithm-specific branching.

Logistic Regression was configured with a maximum of 1,000 iterations and L2 regularization using the default penalty strength. The LogisticModel subclass additionally exposes a get_coefficients method that returns a dictionary mapping each feature name to its learned coefficient, along with an interpret_coefficients method that translates positive and negative coefficients into human-readable attrition risk statements for HR practitioners.

Random Forest was configured with 100 estimators, a maximum tree depth of 10, and a fixed random state of 42 for reproducibility. The RandomForestModel subclass exposes a get_feature_interactions method that identifies the top five most important features based on the ensemble's built-in feature importance scores.

Decision Tree was configured with a maximum depth of 10 and a minimum samples per split of 20 to prevent overfitting on the relatively small dataset. The DecisionTreeModel subclass exposes a get_decision_paths method that returns humanreadable descriptions of the decision paths followed for a sample of input records, supporting interpretability for HR stakeholders.

Model evaluation is performed by the evaluate_model method, which computes accuracy, precision, recall, F1-score, and AUC-ROC on the held-out test set. Stratified five-fold cross-validation is additionally applied to each model to estimate generalization performance and detect overfitting. The compare_models method aggregates evaluation results across all three models into a comparison DataFrame, selects the best model based on F1-score, generates ROC curve visualizations saved to disk, and produces a set of natural language recommendations based on the comparison results. The best-performing model is serialized to disk as a pickle file for use by the web application and risk assessment module.

### *D. Web Application*

The interactive dashboard was built using Streamlit and loads pre-trained models and encoders at startup from the serialized pickle files. The application provides five main views: a workforce overview showing aggregate risk statistics and distribution charts, a high-risk employee table with sortable columns and department filters, an individual employee assessment form that accepts manual input and returns a real-time risk score with feature contribution breakdown, a department analysis view showing comparative attrition rates and risk distributions, and an executive summary page that renders the auto-generated markdown report. The application is containerized using Docker with a Python 3.9 slim base image, exposing port 8501 and accepting connections from any network interface, enabling deployment to Streamlit Cloud, AWS, Google Cloud, or Azure with no application-level changes required.

### *E. Testing*

System correctness was validated using a property-based testing strategy implemented with the Hypothesis library. Property-based tests were written for the three core data processing modules. For the DataValidator, tests verified that the overall quality score always falls within the valid range of 0 to 1 regardless of the input data distribution. For the FeatureEncoder, tests

verified that the number of output columns after encoding is always greater than or equal to the number of input columns, that encoded numerical values have zero mean and unit variance, and that the transform method produces identical output to fit_transform when applied to the same data. For the EDA engine, tests verified that statistical test results are consistent with the direction of observed group differences. Each property test was configured to run a minimum of 100 randomly generated input examples, providing substantially broader coverage than equivalent hand-written unit tests.

## V. EXPERIMENTAL RESULT AND PERFORMANCE EVALUATION

### A. *Experimental Setup*

All experiments were conducted on the HR employee dataset comprising 1,470 records. The dataset was split into a training set of 1,176 samples (80%) and a test set of 294 samples (20%) using stratified sampling to preserve the original class distribution in both subsets. A fixed random seed of 42 was used across all experiments to ensure reproducibility. Five-fold stratified cross-validation was additionally applied to each model to estimate generalization performance and assess variance. All three models were evaluated on the same held-out test set under identical conditions to ensure a fair comparison.

### B. *Model Performance Comparison*

Table 1 presents the performance of all three classifiers on the held-out test set across five evaluation metrics. Table 1: Model Performance Comparison on Test Set

| Model | Accuracy | Precision | Recall | F1-Score | AUC-ROC |
|---|---|---|---|---|---|
| Logistic Regression | 67.0% | 63.1% | 67.0% | 65.0% | 59.7% |
| Random Forest | 67.3% | 63.9% | 67.3% | 65.6% | 62.9% |
| Decision Tree | 67.0% | 61.6% | 62.6% | 62.1% | 54.3% |

All three models achieved comparable accuracy in the range of 67%, which is consistent with findings reported in prior literature on similar HR attrition datasets. The narrow performance gap across models is largely attributable to the class imbalance in the dataset, where approximately 84% of records belong to the non-attrition class. Under such conditions, a naive classifier that always predicts the majority class would achieve approximately 84% accuracy, making F1-score and AUC-ROC more informative metrics than raw accuracy for evaluating true predictive capability.

### C. *Cross-Validation Results*

Table 2 presents the five-fold stratified cross-validation results for each model, showing mean accuracy and standard deviation across folds.

Table 2: Five-Fold Cross-Validation Results

| Model | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean | Std Dev |
|---|---|---|---|---|---|---|---|
| Logistic Regression | 66.3% | 68.0% | 65.6% | 67.3% | 66.7% | 66.8% | ±0.8% |
| Random Forest | 67.0% | 68.4% | 66.3% | 68.0% | 67.7% | 67.5% | ±0.7% |
| Decision Tree | 64.6% | 67.0% | 65.3% | 66.3% | 65.6% | 65.8% | ±0.8% |

The low standard deviation across folds for all three models indicates stable generalization performance with minimal variance, suggesting that the models are not overfitting to the training data. The cross-validation means are closely aligned with the test set results reported in Table 1, further confirming the reliability of the evaluation.

### D. *Feature Importance Analysis*

Table 3 presents the top ten features ranked by importance score extracted from the Decision Tree model. Feature importance is computed as the normalized total reduction in impurity contributed by each feature across all splits in the tree.

Table 3: Feature Importance Rankings (Decision Tree)

| Rank | Feature | Importance Score | Category |
|---|---|---|---|
| 1 | MonthlyIncome | 19.5% | Compensation |
| 2 | Age | 13.5% | Demographic |
| 3 | DistanceFromHome | 9.0% | Work-Life Balance |
| 4 | YearsAtCompany | 8.5% | Career |
| 5 | NumCompaniesWorked | 8.5% | Career |
| 6 | OverTime_Yes | 7.2% | Work-Life Balance |
| 7 | JobSatisfaction | 6.8% | Behavioral |
| 8 | Department_Sales | 5.4% | Organizational |
| 9 | JobRole_Sales Rep | 4.1% | Organizational |
| 10 | EducationField_HR | 3.2% | Demographic |

Monthly income emerged as the single most important predictor with an importance score of 19.5%, confirming that compensation is the primary driver of attrition in this dataset. Age ranked second at 13.5%, reflecting the welldocumented tendency of younger employees to exhibit higher job mobility. Distance from home and years at company each contributed approximately 9% and 8.5% respectively, highlighting the role of commuting burden and early-tenure vulnerability in attrition risk.

## VI. LIMITATIONS AND FUTURE WORK

### A. *Limitations*

The system has several notable limitations. First, the model was trained on a single static dataset of 1,470 records from one organization, which limits its generalizability to other industries or workforce types. Second, the class imbalance — with only 16% attrition cases — biases models toward the majority class, resulting in modest AUC scores between 54% and 63%. Third, the feature set of 12 attributes excludes important factors such as manager relationship quality, performance history, and internal promotion records, which could improve predictive accuracy. Fourth, the business impact figures rely on assumed constants such as a 200% replacement cost multiplier and a 30% intervention success rate, which may not reflect actual organizational conditions. Finally, the system operates on static batch data with no real-time integration, meaning risk scores become outdated as employee circumstances change. No fairness or bias audit was conducted, which is a critical gap for any real-world HR deployment.

### B. *Future Work*

Several enhancements are planned for future development. Integrating gradient boosting algorithms such as XGBoost and LightGBM is expected to improve AUC performance beyond the current 63% ceiling. Real-time integration with HR platforms such as Workday or SAP SuccessFactors would enable continuous risk score updates. Adding SHAP-based explainability would allow HR managers to understand the specific factors driving each individual employee's risk score. An A/B testing framework to track intervention outcomes would close the feedback loop and improve the accuracy of ROI projections over time. Finally, a fairness audit covering protected demographic attributes should be incorporated before any production deployment to ensure compliance with employment law and ethical standards.This paper has presented Academic Growth Tracker Pro, a comprehensive desktop-based academic management system that integrates student performance tracking, faculty management, attendance recording, analytical visualization, and automated PDF reporting within a single, locally deployable Python application. The system addresses a demonstrable gap in the available toolset for educational institutions that require robust data management capabilities without the infrastructure costs and complexity associated with enterprise ERP systems or web-based LMS platforms.

The modular architecture and open-source technology stack position Academic Growth Tracker Pro as a scalable foundation for ongoing enhancement. Planned extensions targeting cloud synchronization, predictive analytics, and biometric identification will progressively advance the system toward a comprehensive intelligent academic management platform. The work demonstrates the continued relevance and practical value of desktop- based educational software as a complement to, and in some contexts a superior alternative to, web-dependent institutional management solutions.

## VII. CONCLUSION

This project successfully developed a machine learning-powered employee attrition prediction system capable of identifying at-risk employees with 67% accuracy across three models — Decision Tree, Random Forest, and Logistic Regression. By analyzing 1,470 employee records, the system uncovered that monthly income, age, and overtime are the strongest drivers of attrition.The business impact is significant: 446 high-risk employees were identified, with a potential cost saving of $121M and an estimated ROI of over 1,000%. The solution was delivered as a production-ready, Dockerized web application with property-based testing, modular architecture, and live deployment on Streamlit Cloud.

Future work can further improve predictive performance using XGBoost or neural networks, integrate real-time HR data feeds, and measure the effectiveness of retention interventions through A/B testing.Feature importance analysis revealed that monthly income is the single strongest predictor of attrition, accounting for 19.5% of model importance, followed by age (13.5%), distance from home (9.0%), and years at the company (8.5%). These findings align with established HR research and provide a clear, actionable roadmap for retention interventions.

Beyond the machine learning core, this project was built to professional software engineering standards. The system features a fully interactive Streamlit web application with real-time analytics and visualizations, containerized via Docker for consistent deployment across environments, and deployed live on Streamlit Cloud. The codebase follows a clean modular architecture with separate components for data validation, EDA, feature encoding, model training, and risk assessment. Property-based testing using the Hypothesis framework ensures correctness of core logic across a wide range of inputs, while YAML-based configuration management enables flexible model tuning without code changes.

# References

[1] R. Govindarajan, “Predicting Employee Attrition: A Comparative Analysis of Machine Learning Models,” *Procedia Computer Science*, vol. 235, pp. 123– 132, 2025.

[2] S. M. Varkiani, “Predicting Employee Attrition and Explaining Its Determinants Using Explainable AI,” *Expert Systems with Applications*, vol. 245, 2025.

[3] K. Konar, “Employee Attrition Prediction Using Bayesian Optimized Stacked Classifier,” *SN Computer Science*, vol. 6, no. 2, 2025.

[4] H. Talebi, “Hybrid Machine Learning Model for Employee Turnover Prediction Using Genetic Algorithms and LightGBM,” *Journal of Computational Science*, vol. 78, 2025.

[5] İ. T. Baydili, “Explainable AI-Based Decision Support Systems for HR Analytics,” *Systems*, vol. 13, no. 7, 2025.

[6] S. Mali et al., “Employee Attrition Prediction Using Machine Learning Techniques,” *International Journal of Advanced Research in Computer Science*, vol. 16, no. 2, 2025.

[7] M. Haque, “Machine Learning Models to Evaluate Employee Attrition in the Post-COVID Work Environment,” *International Research Journal of Multidisciplinary Studies*, 2025.

[8] R. Srivastava and S. Patnaik, “Data-Driven Insights and Predictive Modelling for Employee Attrition,” *Journal of Data Science and Analytics*, vol. 9, 2025.

[9] S. Shinde, “Predictive HR Analytics and Employee Attrition Modelling: A Strategic Approach,” 2025.

[10] N. Vijayan, “Mitigating Employee Attrition Using Machine Learning and Data Engineering Pipelines,” *arXiv preprint arXiv:2502.17865*, 2025.

[11] Y. Ma et al., “Can Large Language Models Predict Employee Attrition?,” *arXiv preprint arXiv:2411.01353*, 2024.

[12] A. M. Căvescu, “Predictive Analytics in Human Resource Management Using Artificial Intelligence,” *AI Journal*, vol. 5, no. 3, 2025.

[13] M. Alqahtani et al., “Employee Attrition Prediction Using Machine Learning: A Review,” *Journal of Business Analytics*, 2024.