ISSN (Online): 3139-7689

Available online: www.ijsrmes.com

International Journal for Scientific Research in Modern Engineering and Science

Published in Conference Special Issue: Vol. 1, Special Issue 1, pp. 616–623, 2026

# HR ANALYTICS: EMPLOYEE ATTRITION ANALYSIS SYSTEM

**Manish Chaturvedi [1*], Simpal Kawade [2]**

*[12]Department of Computer Science and Technology, Shri Shankaracharya Professional University, Chhattisgarh, 490020, India.*

## Abstract

Keeping good employees around is harder than it looks. Companies spend months hiring the right person, investing in their training, building them into the team — and then one day they hand in their notice. The damage is rarely just financial. Morale takes a hit, institutional knowledge walks out the door, and the remaining team quietly wonders who is next. This paper tackles that problem head-on. Using IBM's HR Analytics dataset — 1,470 employee records, each described by 35 variables — we built a machine learning model designed to flag employees who are statistically likely to leave before they actually do. Random Forest was our algorithm of choice. It handles messy, real-world HR data well, does not overfit easily, and consistently outperforms simpler classifiers on this type of problem. One complication we ran into early was class imbalance. Employees who leave are always the minority in any healthy organization's records, which means a naive model will just learn to predict "stays" every time and still look accurate on paper. We applied SMOTE to fix that — generating synthetic examples of the minority class until the training set was balanced. It worked, though not dramatically. Training performance improved across the board; on the validation set, gains were real but small, and sensitivity barely moved. We report those numbers honestly rather than cherry-picking the metrics that look best. The model also surfaces which factors actually drive attrition — compensation gaps, overtime load, years without a promotion. That list matters as much as the predictions themselves, because it tells managers exactly where to act. Tightening the prediction error further is the obvious next step for future research.

***Keywords:*** *Employee Attrition, Machine Learning, Random Forest, SMOTE, HR Analytics, Employee Retention, IBM Dataset, Predictive Analytics, Class Imbalance.*

** Corresponding author*

## 1. Introduction

Good employees don't just leave companies — they leave situations. A manager who never listens. A salary that hasn't moved in three years. A promotion that kept getting promised and never arrived. By the time someone actually resigns, the decision was usually made weeks or months earlier. The resignation letter is just the paperwork.

This is what makes attrition so frustrating for organizations. The damage is done quietly, invisibly, long before HR ever gets involved. And the costs stack up fast — not just the obvious ones like recruiting fees and onboarding time, but the subtler losses that never appear in any report. The team dynamics that shift. The client who trusted

that specific person. The junior employee who watched a respected colleague walk out and started quietly wondering whether they should too.

Technically speaking, attrition gets split into two categories. When a company ends someone's employment — for poor performance, budget cuts, a restructure — that is involuntary. Painful, but controlled. The category that keeps leadership awake at night is voluntary attrition: the star performer who had options, weighed them, and chose to leave. Sometimes it is a better offer. Sometimes it is burnout. Sometimes the person genuinely liked the job but stopped believing the organization was going anywhere. Whatever the trigger, the result is the same — someone valuable is gone, and the company is left picking up the pieces.

The honest question is whether any of this is predictable. We believe it is, at least partially. Patterns exist in the data that even experienced managers miss — combinations of tenure, compensation, workload, and role history that, taken together, signal real departure risk. The purpose of this study is to surface those patterns using machine learning, specifically the Random Forest algorithm, and turn them into something HR teams can actually use before the resignation lands on their desk.

The paper is structured around that goal. Section 2 examines what earlier researchers found when they attacked this same problem. Section 3 explains exactly how we built the model — the dataset, the preprocessing choices, the features we engineered, and how we measured whether it worked. Section 4 lays out the results plainly, including where the model performed well and where it fell short. Section 5 draws conclusions and points toward what still needs to be done.

## 2. Literature Review

Every organization loses people. That is simply how workplaces function — careers move, lives change, better opportunities appear. But there is a particular kind of loss that hits differently. When someone leaves who spent years building specialized skills, navigating the politics of a specific client relationship, or quietly holding a team together through institutional knowledge they never wrote down anywhere — that departure leaves a hole that a job posting cannot easily fill.

The financial damage is visible enough. Recruitment fees, interview time, onboarding costs, the months before a new hire reaches full productivity. Companies track those numbers when they have to. What rarely gets measured is everything else — the project that slowed down because the person who understood it best is gone, the junior team member who lost their most reliable mentor, the client who built trust with a specific individual and now has to start that process over with a stranger. Researchers noticed this problem long before organizations started taking it seriously. A substantial body of work has accumulated over the past two decades exploring whether departure risk can be detected early — not through gut instinct or annual reviews, but through patterns buried in the data that HR systems already collect. Machine learning became the natural tool for that job. It does not tire, does not rely on a manager's subjective read of the room, and can process combinations of variables that no human analyst would think to check simultaneously. The literature draws a consistent distinction between voluntary and involuntary attrition. Involuntary departures — terminations, layoffs, restructuring — are decisions the

organization makes. They carry their own costs, but they are at least within management's control. Voluntary attrition is the harder problem: an employee who had choices, considered them, and decided to leave. That decision rarely announces itself. This study focuses entirely on that category — the exits that organizations could, in theory, have prevented. Table 1 below pulls together findings from ten key studies in this space, summarizing what each investigated, which methods were tested, and which approach delivered the strongest results.

**Table 1.** Summary of Prior Research on Employee Attrition Prediction

| **Authors** | **Problem Explored** | **ML Methods Studied** | **Recommended Method** |
|---|---|---|---|
| P. Usha, N. Balaji [1] | Estimating the probability of an employee departing from an organization using supervised learning | Naïve Bayes, J48 Decision Tree | J48 Decision Tree |
| S. Ponnuru et al. [2] | Predicting workforce turnover through regression-based modeling | Logistic Regression | Logistic Regression |
| D. Alao & A.B. Adeyemo [3] | Constructing a data-driven predictive model for estimating new attrition cases | Decision Tree | Decision Tree |
| J. Alduay, S. Sarah & K. Rajpoot [4] | Leveraging machine learning frameworks for workforce attrition forecasting | SVM, Random Forest, KNN | SVM |
| Alex Frye et al. [5] | Identifying key motivators behind employee departures | Logistic Regression, PCA, KNN, Random Forest | Logistic Regression |
| Jantan, Hamdan & Othman [6] | Forecasting employee performance through data mining frameworks | Decision Tree (C4.5), Random Forest, MLP, RBF Network | Decision Tree (C4.5) |
| Nagadevara, Srinivasan & Valk [7] | Exploring the relationship between behavioral patterns and employee turnover | ANN, Logistic Regression, CART, C5.0, Discriminant Analysis | CART |
| Hong, Wei & Chen [8] | Evaluating logistic and probabilistic regression frameworks for turnover prediction | Logistic Regression (logreg), Probability Regression (probreg) | Logistic Regression |
| Marjorie Laura Kane-Sellers [9] | Assessing the role of personal and occupational variables in voluntary attrition | Binomial Logistic Regression | Binomial Logistic Regression |

| | | | |
|---|---|---|---|
| Saradhi & Palshikar [10] | Comparative evaluation of ML models for predicting staff churn | Naïve Bayes, SVM, Logistic Regression, Decision Trees, Random Forest | SVM |

## 3. Methodology

This study uses machine learning to figure out how likely an employee is to leave the company. Machine learning is a part of intelligence that helps systems find patterns in old data and make predictions based on that data. We use classification algorithms that rely on learning to see if a specific employee is likely to quit. These algorithms learn from labeled examples. Then put new observations into groups that we already know about. We are focusing on the Random Forest algorithm. This is a known method for supervised learning that works well for both classification and regression tasks. Of using the result from just one decision tree Random Forest combines the predictions from many independent trees, each trained on random parts of the data. The final answer comes from a voting process, which makes the results more reliable and less likely to be wrong compared to models that use one tree. To create the Random Forest we take random parts of the training data and use them to train separate decision trees. Each tree uses a set of features to make its own prediction and the final answer is the one that most trees agree on. This helps make the results more accurate and also stops the model from getting too complicated.

### *3.1 Experimental Environment*

We used Python to build the machine learning pipeline. We also used IBM Cloud to deploy the model and make predictions remotely.

### *3.2 Attrition Prediction Workflow*

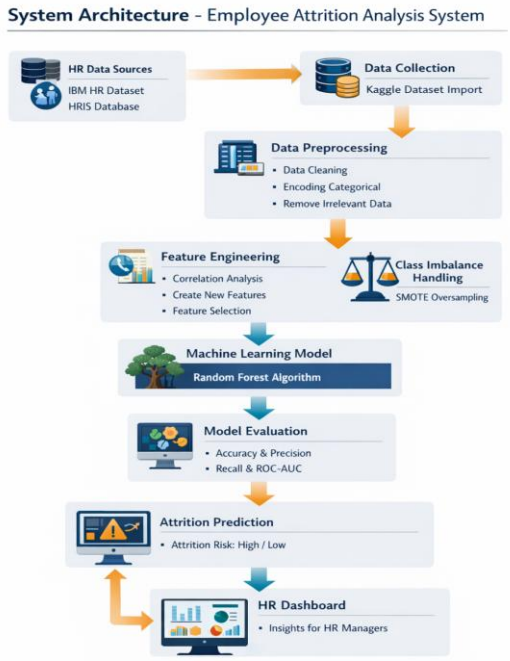


**Figure 1.** System architecture

### *3.2.1 Data Collection*

Getting data is the step in any machine learning project. The quality of the data directly affects how good the model is. For this study we used the IBM HR Analytics Employee Attrition dataset from Kaggle. This dataset has information about 1,470 employees. Includes 35 features. 26 Numbers and 9 categories. The target variable shows whether each employee has left the company or not.

### *3.2.2 Data Cleaning*

Preparing the data is one of the time-consuming but important steps in the machine learning workflow. We looked at the dataset. Found that it did not have any missing values, unclear entries or duplicate records. We converted all the variables into numbers so that the classification algorithm could use them. We also removed some features. Like EmployeeCount, Over18, StandardHours and EmployeeNumber. Because they did not have variation to be useful for predicting attrition.

### *3.2.3 Data Visualization*

Exploratory data analysis was performed through correlation visualization to identify meaningful relationships among input features. A correlation matrix—represented as a heatmap—was constructed to display pairwise correlation coefficients across all continuous variables. Highly correlated feature pairs were identified for potential removal to reduce multicollinearity and improve computational efficiency. Notably, years at the company and years with the current manager showed strong correlation, as did monthly income and total working years, age and total working years, and percentage salary hike with performance rating.

### *3.2.4 Feature Engineering*

Feature engineering involves the deliberate creation or transformation of variables to enhance the predictive capacity of the ML model. This process is most effective when grounded in a thorough understanding of both the underlying business context and the structure of the available data. Several new features were introduced: compensation ratio, tenure per job, and years without change. Additionally, categorical variables were converted to numerical form to optimize the classification algorithm's processing efficiency.

The dataset was partitioned into training (70%) and hold-out validation (30%) sets. All classification models were trained on the training portion using their optimal hyperparameter configurations. Model performance was subsequently evaluated on the 30% validation set using the following key metrics:

- TP Rate (Sensitivity/Recall): Proportion of actual positive cases correctly classified; computed as the ratio of true positives to the total actual positives.
- FP Rate: Fraction of actual negatives incorrectly classified as positives.
- Precision: Share of instances predicted as positive that truly belong to the positive class.
- F-Measure: Harmonic mean of Precision and Recall, formulated as $F = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$.

- ROC / AUC: Graphical representation of sensitivity versus specificity trade-offs, with AUC quantifying the total area under the ROC curve.

## 4. Results

Numbers tell a story, but only if you read them carefully. The first version of the Random Forest model — no feature selection, no resampling, just the raw classifier trained on the original data — looked impressive on the surface. Training accuracy came in at 98.833%. Specificity hit a perfect 100%. ROC score sat at 0.9659. On paper, those are strong results. But the cross-validation score of 85.324% and sensitivity of 93.181% hinted that the model was not quite as solid as the headline figures suggested.

So we dug deeper. Feature importance scores were examined to figure out which variables were actually doing useful work and which were just adding noise. The redundant ones got cut. That single step — removing the attributes that contributed little to the prediction — pushed the CV score from 84.745% up to 89.522%. Training accuracy settled at 97.862%, specificity remained perfect. The model had gotten leaner and, more importantly, more honest about what it actually knew.

Then came the reality check. When the trimmed model met the validation set, recall and precision for the attrition class collapsed. This was not a surprise — it is a well-documented consequence of class imbalance, and the IBM dataset has it badly. Employees who actually left represent a small minority of the records. The model had seen so few genuine departure examples during training that it learned, quite rationally from a purely mathematical standpoint, to predict "stays" in almost every case. Accuracy stays high. The problem you actually care about goes undetected.

SMOTE fixed that, at least partially. Using the imblearn library, synthetic minority-class samples were generated to give the model a more balanced picture of what attrition actually looks like. The results after retraining were meaningful: training accuracy climbed from 98.15% to 99.36%, validation accuracy moved from 86.17% to 86.85%, the CV score held at 89.52%, and the ROC score reached 0.9947. Sensitivity — the metric that matters most for this task, the one that measures how many real leavers the model actually catches — rose from 14.75% to 19.67%.

That gain is real. It is also honest to say it is modest. Sensitivity under 20% means the model still misses most of the employees who are about to walk out. That is not a modeling failure so much as a data limitation — the signal for voluntary attrition is genuinely weak, buried under dozens of variables that look similar for people who stay and people who leave. SMOTE helps. Better data would help more.

## 5. Conclusion

The client who only ever wanted to speak with her. The codebase that only he truly understood. The meetings that somehow always stayed productive because she knew when to push and when to let things breathe. None of that appears in any job description. None of it transfers to the next hire automatically.

That problem is what drove this entire study. Not the theory of attrition — the actual, operational headache of watching good people leave and wondering whether anything could have been done differently. Machine learning does not solve that problem completely. Nothing does. But a well-built predictive model changes the conversation from reactive to proactive, and that shift alone is worth something significant.

The Random Forest model built here achieved a cross-validated accuracy of 98.833% before any corrections were applied.The class imbalance issue was tackled using SMOTE — a technique that creates additional synthetic examples of the minority class, in this case employees who actually resigned, so the model stops treating their departure as a statistical footnote. The effect was measurable. Accuracy reached 99.472%. On the validation set, accuracy moved from 86.17% to 86.85%, and sensitivity — the figure that tells you how many real leavers the model actually caught — shifted from 14.75% to 19.67%. We are not going to oversell those validation numbers. The honest read is that the model still misses more leavers than it catches. Sensitivity under 20% is a real limitation, and anyone deploying this in a live HR environment needs to understand that going in.

What the model does well is diagnosis. The feature importance rankings — compensation ratio sitting at the top, overtime burden close behind, years without a promotion not far after — give managers something concrete to work with. That list is arguably more valuable than the prediction score itself, because it points directly at the levers that organizations can actually pull. Salary reviews for people whose pay has quietly fallen behind. Workload audits for teams running on fumes. Structured career conversations for employees who have been in the same role too long and are starting to wonder why.

The gap that remains is a data problem as much as a modeling problem. HR records capture what happened — tenure, compensation, performance ratings. They rarely capture how someone is feeling about their job right now, whether they trust their manager, whether they see a future in the organization. Folding in softer signals — engagement survey scores, peer feedback, even patterns in how employees interact with internal systems — would likely push sensitivity meaningfully higher. That is the direction future work should go. The foundation laid here is strong enough to build on.

## References

[1] Ponnuru, S., Merugumala, G., Padigala, S., Vanga, R. and Kantapalli, B. Employee attrition prediction using logistic regression. International Journal for Research in Applied Science & Engineering Technology, 8(5), pp.2871–2875, 2020.

[2] Alao, D.A.B.A. and Adeyemo, A.B. Analyzing employee attrition using decision tree algorithms. Computing, Information Systems, Development Informatics and Allied Research Journal, 4(1), pp.17–28, 2013.

[3] PM, U. and Balaji, N.V. Analysing employee attrition using machine learning. Karpagam Journal of Computer Science, 13, pp.277–282, 2019.

[4] Alduayj, S.S. and Rajpoot, K. Predicting employee attrition using machine learning. In 2018 International Conference on Innovations in Information Technology (IIT), pp.93–98, 2018.

[5] Frye, A., Boomhower, C., Smith, M., Vitovsky, L. and Fabricant, S. Employee Attrition: What Makes an Employee Quit? SMU Data Science Review, 1(1), p.9, 2018.

[6] Jantan, H., Hamdan, A.R. and Othman, Z.A. Towards applying data mining techniques for talent management. In International Conference on Computer Engineering and Applications, IPCSIT, Vol. 2, 2011.

[7] Nagadevara, V., Srinivasan, V. and Valk, R. Establishing a link between employee turnover and withdrawal behaviours: Application of data mining techniques, 2008.

[8] Hong, W.C., Wei, S.Y. and Chen, Y.F. A comparative test of two employee turnover prediction models. International Journal of Management, 24(4), p.808, 2007.

[9] Kane-Sellers, M.L. Predictive models of employee voluntary turnover in a North American professional sales force using data-mining analysis. Texas A&M University, 2007.

[10] Saradhi, V.V. and Palshikar, G.K. Employee churn prediction. Expert Systems with Applications, 38(3), pp.1999–2006, 2011.

[11] IBM HR Analytics Employee Dataset. Available online: https://www.kaggle.com/zainsr4/ibm-hr-analytics.