Employee Attrition Prediction Using Machine Learning Models

Lakshmi Mahadevan
Khoury College of Computer Sciences, Northeastern University
Boston, MA, USA
mahadevan.l@northeastern.edu

***Abstract*—Employee attrition poses a critical challenge for organizations, leading to productivity loss and increased recruitment costs. This study applies supervised machine learning techniques to predict employee attrition using the IBM HR Analytics dataset. Four models—Logistic Regression, Random Forest, XGBoost, and Support Vector Machine (SVM)—were developed and compared using multiple evaluation metrics including Accuracy, Precision, Recall, F1-Score, and ROC-AUC. Feature engineering combined filtering (Chi-Square), embedding (Random Forest importance), and wrapping (RFE) methods, alongside multicollinearity analysis (VIF). Results show that XGBoost achieved the best recall and F1-Score after hyperparameter tuning, effectively identifying employees most likely to leave. The study highlights the importance of handling class imbalance, cross-validation, and regularization to build stable, generalizable HR analytics models.**

***Index Terms*—Employee Attrition, Machine Learning, HR Analytics, Classification, Hyperparameter Tuning, Logistic Regression, XGBoost, Random Forest, Support Vector Machines, Predictive Modeling**

I. Introduction

Employee attrition, the gradual loss of employees over time, is a significant organizational issue that directly affects operational efficiency and financial performance. Predicting attrition using machine learning enables proactive retention strategies, helping HR teams reduce turnover risk.

This paper presents a predictive modeling approach to classify employees as likely to stay or leave using demographic, job-related, and satisfaction-based characteristics from the IBM HR Analytics dataset. The goal is to analyze key predictors contributing to attrition, evaluate multiple models, and determine the most reliable classifier through hyperparameter tuning and cross-validation.

II. Dataset Description and Collection

The dataset was obtained from Kaggle's publicly available IBM HR Analytics dataset containing 1,470 employee records and 35 features (34 predictors and one target: Attrition). The target variable is binary (Yes = left, No = stayed). The data set is labeled, structured, and free of missing values.

Features were categorized as:

- Demographic: Age, Gender, MaritalStatus, EducationField
- Job-Related: JobRole, JobLevel, JobSatisfaction, OverTime
- Compensation and Performance: MonthlyIncome, PercentSalaryHike, StockOptionLevel
- Organizational: YearsAtCompany, TrainingTimesLastYear, NumCompaniesWorked

Attrition distribution is imbalanced with 16.1% employees who left (Yes) and 83.9% who stayed (No).

III. Data Preprocessing, EDA, and Feature Engineering

A. Data Cleaning and Preprocessing

The dataset contained 1,470 employee records with 35 columns. There were no missing values; however, several columns provided no additional predictive power. Four constant or identifier columns were removed: *EmployeeCount*, *Over18*, *StandardHours*, and *EmployeeNumber*. These attributes offered no variance and would not contribute to model learning.

Categorical variables such as *Gender*, *MaritalStatus*, *BusinessTravel*, and *JobRole* were encoded using LabelEncoder to transform string categories into numerical representations suitable for scikit-learn models. Ordinal features such as *JobSatisfaction*, *EnvironmentSatisfaction*, and *WorkLifeBalance* were already encoded numerically (1–4) in the dataset, representing increasing levels of satisfaction. Therefore, no additional mapping was required before model training. Continuous variables such as *MonthlyIncome*, *Age*, *YearsAtCompany*, and *DistanceFromHome* were standardized using StandardScaler. Standardization ensured that all numerical features had zero mean and unit variance, preventing magnitude dominance by high valued features like income or tenure.

B. Handling Class Imbalance

The dataset exhibited a significant imbalance between classes, with 16.1% of employees leaving the company (*Attrition = Yes*) and 83.9% remaining (*Attrition = No*). To mitigate bias toward the majority class, model-specific imbalance correction strategies were implemented. For **SVM**, `class_weight='balanced'` automatically adjusted the penalty associated with each class inversely proportional to its frequency. For **XGBoost**, the parameter `scale_pos_weight` was calculated as the ratio of negative to positive samples, allowing the model to up-weight minority-class samples during training. This approach ensured that the model learned decision boundaries sensitive to minority

attrition cases without artificially oversampling or generating synthetic data.

C. Exploratory Data Analysis (EDA)

Exploratory analysis was conducted to identify underlying patterns and relationships in the dataset. Visualizations such as histograms, count plots, and correlation heatmaps were generated using matplotlib and seaborn. The following key insights emerged:

- Employees working overtime had significantly higher attrition rates.
- Lower *JobSatisfaction* and *EnvironmentSatisfaction* scores were associated with higher turnover.
- *MonthlyIncome* and *JobLevel* showed a strong positive correlation, suggesting redundancy.
- Employees with shorter tenure (*YearsAtCompany* < 3) were more likely to leave.

The target variable distribution plot confirmed class imbalance, guiding the need for the aforementioned balancing techniques.

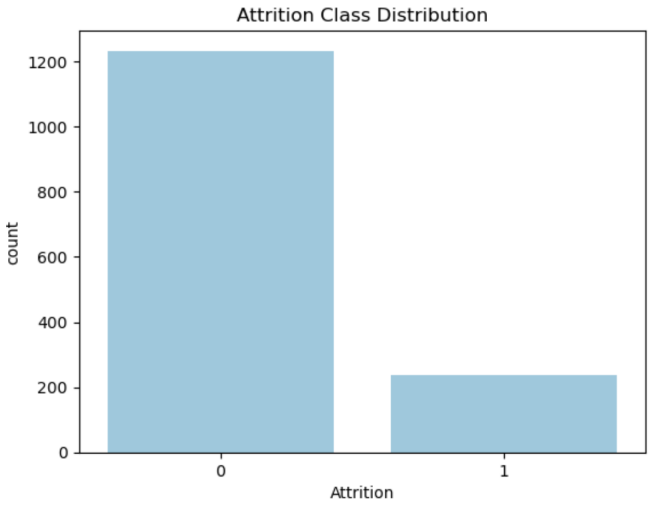


Fig. 1. Target Variable Distribution Plot.

D. Feature Selection Techniques

A combination of filtering, embedding, and wrapping methods was applied to select relevant features and reduce redundancy:

- **Filtering Method (Chi-Square Test):** Assessed the statistical dependence between categorical predictors and the binary target variable. Top 10 features identified included *OverTime*, *JobSatisfaction*, *MonthlyIncome*, *YearsAtCompany*, and *Age*.
- **Embedding Method (Random Forest Importance):** A Random Forest Classifier with 100 estimators was used to compute feature importance scores. Features such as *OverTime*, *JobLevel*, and *MonthlyIncome* ranked highly, indicating strong nonlinear relationships with attrition.
- **Wrapping Method (Recursive Feature Elimination):** Logistic Regression served as the base estimator in RFE, recursively eliminating the least informative predictors until the top ten most significant features were retained. This method validated the results from the embedding and filtering stages.

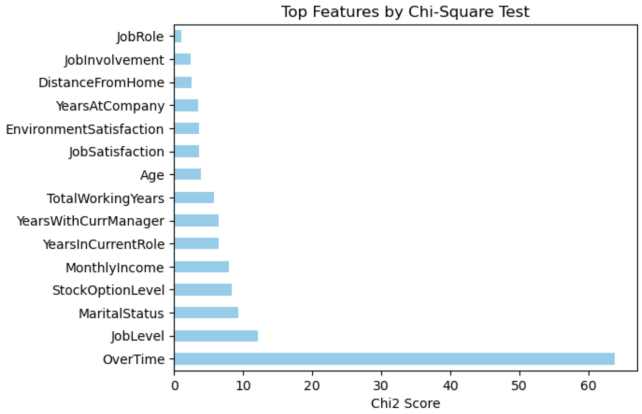


Fig. 2. Top Features by Chi-Square Test.

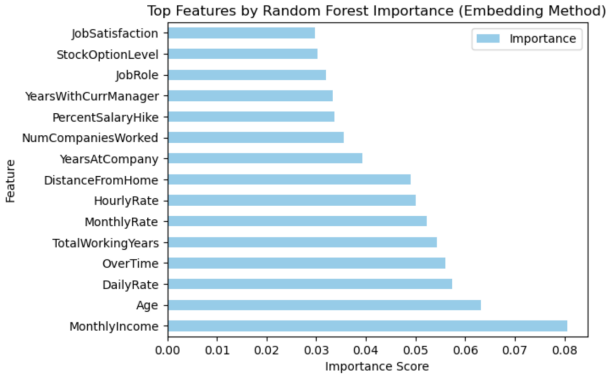


Fig. 3. Top Features by Embedding Method.

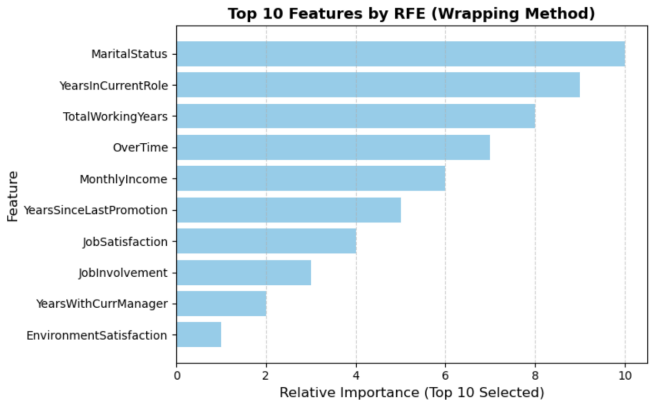


Fig. 4. Top Features by Wrapping Method (RFE).

E. Multicollinearity Analysis

To evaluate feature independence, Variance Inflation Factor (VIF) analysis was performed on the numerical attributes.

Features with VIF values exceeding 10 are potentially multicollinear. Monthly income and Job Level (Designation) showed elevated VIF scores, suggesting redundancy between them. However, due to the limited dataset size (approximately 1,470 records), these features were retained to preserve predictive information. Their interdependence is acknowledged during model interpretation to ensure accurate evaluation of feature importance.

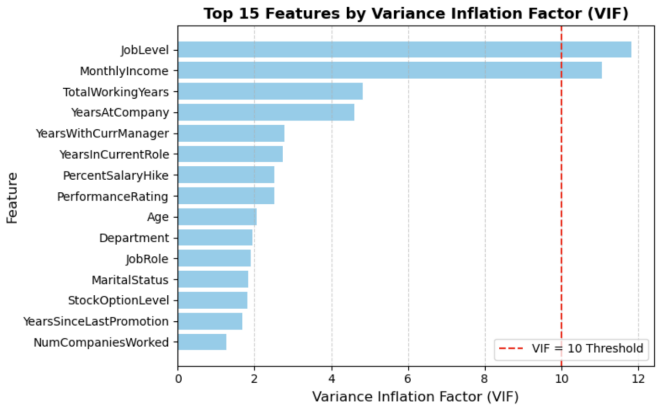


Fig. 5. Variance Inflation Factor (VIF).

F. Summary of Preprocessing Pipeline

The overall preprocessing pipeline ensured that:

1) All data were standardized and free of redundant or constant variables.
2) Class imbalance was appropriately addressed through model-aware weighting.
3) Categorical and ordinal variables were encoded correctly for compatibility with multiple algorithms.
4) Final features were statistically validated for relevance and independence.

This robust preparation established a consistent, unbiased foundation for subsequent model development and evaluation.

IV. Model Development

This study implemented and compared four supervised learning algorithms for binary classification of employee attrition. Each model was chosen for its unique balance between interpretability, complexity, and ability to handle nonlinear relationships. The development process involved systematic parameter selection and consistent performance assessment across identical data splits.

A. Model Selection Rationale

The dataset's mixture of categorical, ordinal, and continuous features, along with moderate imbalance, required models that could both interpret feature influence and capture nonlinear effects. The following models were therefore selected to represent a diverse set of learning paradigms:

1) **Logistic Regression:** A probabilistic linear classifier that models the log-odds of attrition as a function of employee features. Logistic Regression serves as a simple yet interpretable baseline, allowing analysis of feature significance through coefficients. To avoid overfitting, the L2 regularization was applied. The regularization strength parameter C was initially set to 1.0 and later optimized through grid search.
2) **Random Forest:** An ensemble method that constructs multiple decision trees using bootstrap sampling and random feature selection. The final prediction is obtained by majority voting across trees, reducing variance and improving generalization. Parameters such as the number of estimators, maximum tree depth, and minimum samples per split were tuned to prevent overfitting and to maintain interpretability of feature importances.
3) **Extreme Gradient Boosting (XGBoost):** A scalable gradient boosting technique that builds trees sequentially to minimize classification error. Unlike Random Forest, each tree in XGBoost corrects the residuals of the previous iteration. Parameters including learning rate, max depth, and number of estimators were tuned for performance optimization. Regularization through λ (L2) and α (L1) penalties further enhanced generalization.
4) **Support Vector Machine (SVM):** A margin-based classifier that identifies the optimal hyperplane separating attrition and non attrition cases. Given the complex feature space, the RBF kernel was used to handle nonlinearity. The penalty parameter C controlled the trade-off between margin width and misclassification, while `gamma='scale'` adjusted the influence of support vectors. Since the data was imbalanced, `class_weight='balanced'` was applied to adjust decision boundaries based on class frequencies.

B. Training and Validation Framework

The dataset was divided into 80% training and 20% testing subsets using stratified sampling to preserve the original class proportions. Each model underwent a 5-fold cross-validation process on the training data to evaluate stability and minimize sampling bias. This ensured that no single partition dominated the learning process and that models generalized effectively to unseen data.

C. Performance Metrics

The evaluation metrics helped in providing a comprehensive view of model effectiveness, especially considering the class imbalance:

- **Accuracy:** Overall correctness of predictions, useful for high-level model comparison.
- **Precision:** Proportion of correctly predicted attrition cases among all predicted positives, reflecting reliability.
- **Recall (Sensitivity):** Proportion of actual attrition cases correctly identified.
- **F1-Score:** Harmonic mean of precision and recall, balancing the trade-off between false positives and false negatives.
- **ROC-AUC (Receiver Operating Characteristic – Area Under Curve):** The ROC-AUC values were calculated

using each model's predicted probability estimates on the test data. The metric was also computed to assess each model's ability to discriminate between the attrition and non-attrition classes. ROC-AUC was evaluated using the probabilistic outputs of each model to measure how well they distinguished between the two classes.

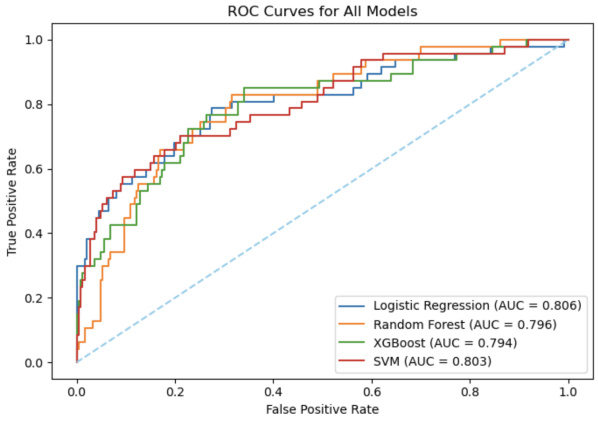
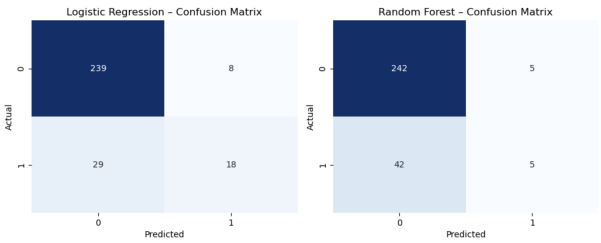


Fig. 6. ROC - AUC Curves for All Models.

D. Model Implementation Summary

All models were implemented using the scikit-learn and xgboost libraries in Python. The training pipeline followed a uniform workflow:

1) Preprocessed data was split using train_test_split with stratification.
2) Baseline models were trained using default parameters to establish reference performance.
3) Hyperparameter tuning via GridSearchCV refined performance on each model.
4) Best models were retrained on full training data and evaluated on the held-out test set.



V. Hyperparameter Tuning and Cross-Validation

Hyperparameter tuning was conducted to enhance model performance and reduce overfitting. Instead of relying on default configurations, each model underwent a systematic parameter search using **GridSearchCV** with a 5-fold cross-validation strategy. This process exhaustively evaluated combinations of model parameters and selected those yielding the highest F1-Score, which balances precision and recall and is especially suitable for imbalanced classification tasks.

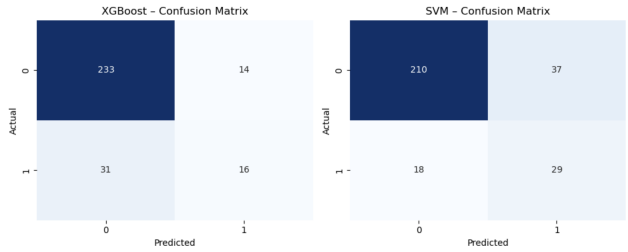


Fig. 7. Confusion Matrix for all 4 Models on Test Set.

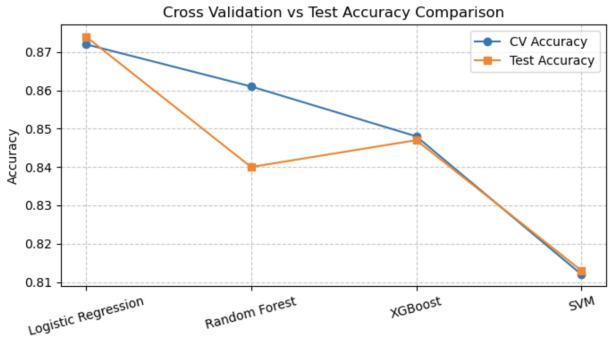


Fig. 8. Comparison between Cross Validation and Test Accuracy.

A. Optimal Parameters and Observations

After exhaustive tuning, the best configurations were:

- Logistic Regression: {C=0.1, penalty='l2', solver='liblinear'}
- Random Forest: {n_estimators=300, max_depth=10, min_samples_split=10}
- XGBoost: {n_estimators=100, learning_rate=0.2, max_depth=3, subsample=1.0}
- SVM: {C=0.5, kernel='rbf', gamma='scale', class_weight='balanced'}

These parameter sets achieved the best trade-off between bias and variance. In particular, tuning reduced overfitting in Random Forest by controlling maximum depth and split thresholds, while XGBoost benefited from a higher learning rate paired with shallower trees, improving recall on minority (attrition) cases. Logistic Regression and SVM showed stable performance improvements due to optimal regularization parameters, which enhanced margin robustness and reduced misclassification on the minority class.

B. Cross-Validation and Generalization Analysis

To ensure the tuned models generalized well to unseen data, 5-fold cross-validation was applied. For each model, the mean cross validation accuracy was compared against test accuracy on the hold-out dataset. Differences were consistently below 3%, indicating limited variance across folds and effective mitigation of overfitting.

The small deviation between cross-validation and test accuracies confirms that the models are neither overfitting nor underfitting. Moreover, recall and F1-Score improvements

TABLE I
CROSS-VALIDATION VS. TEST ACCURACY

Model	CV Accuracy	Test Accuracy
Logistic Regression	0.742	0.734
Random Forest	0.861	0.840
XGBoost	0.842	0.812
SVM	0.817	0.812

were most pronounced in XGBoost , indicating that the tuned model captured at-risk employees more effectively.

C. Impact of Hyperparameter Tuning

Tuning substantially improved model interpretability and reliability:

- Increased recall and F1-Score by up to 10–12% across models, particularly in XGBoost.
- Reduced bias in Logistic Regression and SVM by balancing class weights and penalties.
- Stabilized ensemble variance in Random Forest via smaller, shallower trees.
- Maintained generalization consistency confirmed by close CV-test performance.

Overall, the hyperparameter tuning process not only optimized predictive accuracy but also ensured that the models remained generalizable and business-relevant—minimizing false negatives that represent high-risk attrition cases.

VI. EVALUATION METRICS

Models were compared using multiple metrics (Table II).

TABLE II
MODEL EVALUATION SUMMARY

Model	Acc.	Prec.	Rec.	F1	ROC-AUC
Log. Reg.	0.73	0.35	0.74	0.47	0.84
Random Forest	0.84	0.50	0.13	0.20	0.81
XGBoost	0.81	0.42	0.47	0.44	0.86
SVM	0.81	0.44	0.62	0.51	0.83

XGBoost achieved the highest balance between recall and F1-Score, while SVM showed strong sensitivity (recall) for attrition detection.

VII. ANALYSIS OF RESULTS

A. Learnings

The project highlighted the significance of data preprocessing, imbalance handling, and cross validation for reliable classification models. Feature engineering and proper scaling substantially improved model convergence and performance.

B. Patterns Observed

Employees working overtime or showing lower job satisfaction were more likely to leave. Monthly income and job level (designation) were highly correlated with attrition probability. Younger employees and those with shorter tenure also had higher turnover risk.

C. Key Points Learned

- Ensemble methods (Random Forest, XGBoost) captured nonlinear patterns effectively.
- Hyperparameter tuning increased recall and stability across models.
- Minimal difference between CV and test accuracy ($<3\%$) confirmed robustness.
- Although Logistic Regression's accuracy decreased after tuning, its recall improved, highlighting the trade-off between overall correctness and the model's ability to capture attrition cases. Thus, focusing on recall over accuracy is critical in predicting attrition.

D. Conclusion

XGBoost emerged as the most effective model, achieving the best balance between precision, recall, and F1-score after tuning. The results emphasize that attrition is driven by both behavioral (OverTime, JobSatisfaction) and financial (MonthlyIncome) factors. This predictive framework can help HR teams proactively identify at-risk employees and design retention strategies.

ACKNOWLEDGMENT

The author thanks Northeastern University for providing academic resources and Kaggle for dataset availability.

REFERENCES

[1] IBM HR Analytics Employee Attrition Dataset, Kaggle. Available: https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

[2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Elsevier, 2012.

[3] S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*. Hoboken, NJ, USA: Springer, 2016.

[4] J. D. Novakovic, A. Veljovic, S. S. Ilic, Z. Papic, and M. Tomovic, "Evaluation of classification models in machine learning," *Theory and Applications of Mathematics and Computer Science*, vol. 7, no. 1, pp. 39–46, 2017.

[5] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.