Predicting Employee Attrition: A Machine Learning Approach to Workforce Retention

Ahammed Jumma
Assaduzzaman Munna
Asif Akbar Zishan
Department of Electrical and Computer Engineering
North South University
Plot # 15 Block B, Bashundhara R/A, Dhaka, 1229, Bangladesh

Abstract— **Employee attrition poses a significant challenge to organizational productivity, operational continuity, and profitability, necessitating robust strategies to identify and mitigate turnover risks. Traditional statistical models often fail to capture the nonlinear relationships among factors influencing attrition. This study leverages machine learning to predict employee attrition with high accuracy using a comprehensive dataset encompassing demographic, employment, and performance attributes, such as *MonthlyIncome*, *JobSatisfaction*, and *OverTime*. Multiple algorithms, including k-NN, Logistic Regression, SVC, Random Forest, and AdaBoost, were evaluated, with a Stacking Ensemble (Logistic Regression as meta-model) achieving an accuracy of 79.59% and a macro F1-score of 49.15%. Feature importance analysis, derived from Random Forest, highlighted key attrition drivers, informing targeted retention strategies. Explainable AI techniques, specifically SHAP values, enhanced model interpretability, enabling HR professionals to understand prediction rationales and implement informed interventions. This approach demonstrates the transformative potential of machine learning and XAI in HR analytics, offering scalable, data-driven solutions to enhance employee retention and ensure organizational stability.**

Index Terms— **Employee Attrition, Machine Learning, Logistic Regression, Stacking Ensemble, Feature Selection, Class Imbalance, Explainable AI, HR Analytics.**

I. INTRODUCTION

Employee attrition is a big issue facing organizations globally and the concerns are directly linked to productivity, institutional memory retention, operational continuity and profitability [1], [2]. Poor retention may result in higher expenses in recruiting and training, team discontinuity and expertise loss, which risks long-term stability of the business [1], [3]. Attrition has emerged as a strategic imperative as organizations struggle to understand the underlying causes so as to be able to remedy [2]. The conventional methods which relied on statistical modelling and analysis of past trends have been very useful to a certain extent, although they have in many instances failed to record such nonlinearities and complex-factor interplays which cause an employee to leave [4]. This research is inspired because it is necessary to overcome these constraints by utilizing the power of machine learning (ML) to create predictive, scalable, as well as, interpretable models of attrition [5]. In contrast to traditional approaches, ML algorithms have the ability to detect subtle relationships within big, complex data where demographic, work, and performance related variables could be integrated that would otherwise not be noticed [6], [7]. This paper discusses how to use ML to predict the probability of employee attraction with a rich data on the organization with attributes viz. job satisfaction, compensation rates, tenure, career advance- ment opportunities and other reference points of work environment [7]. Through proper detection of employees who are likely to exit, organization can adopt proactive strategies of keeping their employees and maintain a stable workforce that will ensure their prosperity in the long term [6]. The data employed in the given study is therefore acquired in organizational records, making it highly relevant to the real world. All the data were subjected to a thorough preprocessing pipeline involving missing value treatment, categorizing features, categorical features encoding, and evaluating the exploratory data analysis (EDA) to establish essence correlations and trends. The visualization method discussed correlation heatmaps, distribution figures, and pairwise relationship charts that helped gain preliminary knowledge about the relationship between different factors and attrition results. Visualizations were applied with matplotlib and seaborn and stored through the created savefig() to save the repetitions and optimal documentation [8]. Multiple machine learning models were trained and evaluated, including Decision Tree Classifier, Random Forest Classifier, K-Nearest Neighbors (KNN), AdaBoost, and XGBoost. To compare performance, a custom function evaluateclassifiers() systematically assessed models based on accuracy score, and other key metrics. Ensemble methods consistently outperformed. Interpretability was further enhanced through Explain- able AI (XAI) methods, particularly SHAP values and the LIME Tabular Explainer, which provided transparency into model predictions. With the help of these tools, HR profes- sionals got to know the crystal behind the predictions and knew what specific characteristics influenced a case in a particular way the most [5], [10]. The feature importance analysis showed dominant factors related to attrition that led to the creation of practicable retention strategies [2]. These contributions of the study are three-fold: 1. Predictive power: Creation of easy generalizable high- accuracy ML models that can be used to predict employee attrition. 2. Interpretability: Smoothing Shap and LIME with SHAP and LIME provide an interpretable model output. 3. Practical use: extraction of factors that contribute to the problem of attrition in direct HR policy and retention initiatives [7]. The rest of the paper is organized as follows: Section II is a review of the literature concerning employee attrition modeling, machine learning applications in workforce analytics, and the XAI approach. Section III is a description of the dataset and data pre-processing and exploratory data analysis. Section IV tells the process of machine learning, i.e., the choice of the model and its training, its testing out, and interpretation through explainable AI. In Section V, the experimental results and the meaning of findings are discussed. Section VI includes some limita- tions and possible future research areas. Lastly, Section VII ends with the identification of possible key takeaways and implications of workforce retention strategies. This study builds and delivers a data-driven and explainable framework of the attrition propensity that enables organizations to build a stronger retention and stability workforce through the application of strong machine learning methodologies coupled with the explanation of patterns and behavior [5], [14].

II. Related Works

Employee attrition prediction has gained significant attention in HR analytics due to its economic and organizational impact. Previous studies have applied diverse machine learning (ML) techniques to identify at-risk employees. This section categorizes related works into three clusters:

1) Traditional Machine Learning Approaches
2) Ensemble and Advanced ML Methods
3) Explainability and Practical Deployment

A comparative analysis (Table I) highlights gaps addressed in our work.

Traditional Machine Learning Approaches

Early research focused on classical ML models (e.g., logistic regression, decision trees, SVMs), establishing baselines but lacking robustness on imbalanced data. Key studies include:

- Saradhi & Palshikar (2011) used decision trees/SVMs (78% accuracy), noting high false negatives due to imbalance [12].
- Alao & Adeyemo (2018) emphasized feature engineering (e.g., scaling MonthlyIncome) for logistic regression/k-NN [13].
- IBM HR Analytics (2020) demonstrated linear models' limitations with non-linear drivers (e.g., WorkLifeBalance) [15].
- Yang & Islam (2021) combined RF feature selection, K-means clustering, and logistic regression, identifying travel frequency (2.4× attrition risk) and key factors (MonthlyIncome, Age, JobTenure) [16].
- Guerranti & Dimitri (2023) compared LR, CT, RF, NB, NN, and ensemble methods, achieving 88% accuracy with LR on imbalanced data [17].

Ensemble and Advanced ML Methods

Later works adopted ensemble methods (e.g., XGBoost) and deep learning to improve accuracy and handle imbalance:

- Kumar & Ramesh (2020) applied XGBoost with SMOTE (87% accuracy), underscoring imbalance mitigation [18].
- Li & Hoi (2019) proposed neural networks (89% AUC) but noted computational inefficiency versus tree-based methods [19].

Explainability and Practical Deployment

Recent studies prioritize interpretability (e.g., SHAP, LIME) for HR decision-making:

- Bussmann et al. (2021) used SHAP with Random Forest, quantifying drivers (e.g., low JobInvolvement ↑ risk 22%) [20].
- Díaz et al. (2023) integrated XGBoost with SHAP/LIME (79.2% ROC AUC), identifying OverTime/MonthlyIncome as key factors and addressing imbalance via SMOTE-Tomek [8].

III. Materials and Methods

A. Objectives

This study seeks to construct a robust machine learning pipeline for predicting employee attrition, empowering HR practitioners with actionable insights to enhance retention strategies. The core aim is to comprehensively analyze the dataset to uncover predictive patterns, followed by meticulous preprocessing through one-hot encoding of categorical features such as *BusinessTravel* and *OverTime*, min-max scaling of numerical attributes like *MonthlyIncome* and *Age*, and selection of top features via Random Forest importance. The pipeline evaluates multiple algorithms, including k-NN, Logistic Regression, SVC, Random Forest, and AdaBoost, prioritizing recall to minimize false negatives critical for identifying at-risk employees. Through hyperparameter tuning and threshold calibration, Logistic Regression (recall 0.7872) is optimized for interpretability and performance, while a stacking ensemble with Logistic Regression as the meta-model is explored to boost accuracy. The final objective is to deploy the optimized model for real-time HR applications, ensuring scalability and practical utility.

B. Research workflow

The IBM HR Analytics Attrition Dataset is first loaded using the `pd.read_csv()` function. To gain an initial understanding of the dataset, the `head()` method is used to display the first few rows, while the `info()` method provides insights into the dataset's structure, including its numerical and categorical attributes.

TABLE I

COMPARATIVE ANALYSIS OF KEY STUDIES

Author(s)	**Dataset**	**Methodology**	**Model/Approach**	**Perf.**	**Features/Limitations**
Saradhi & Palshikar (2011)	Private HR	Feature selection + ML	DT, SVM	78% acc	High FN; no imbalance handling
Alao & Adeyemo (2018)	Kaggle HR	Correlation + ML	LR, k-NN	75% acc	Weak non-linear capture
Kumar & Ramesh (2020)	IBM HR	SMOTE + ensemble	XGBoost	87% F1	High acc; lacks interpretability
Guerranti, F., Dimitri, G.M. (2023)	IBM HR (1,470 records)	EDA, 5-fold CV	LR, CT, RF, NB, NN, ensemble	LR: 88% acc, 85% AUC	Stability factor correlations; imbalance, NN black-box
Li & Hoi (2019)	Tech company	Deep learning	Neural Network	89% AUC	Computationally expensive
Bussmann et al. (2021)	Multinational HR	XAI	RF + SHAP	86% prec	Interpretable; no deployment
Díaz, G.M., others (2023)	IBM HR (1,470 records)	KDD/CRISP-DM, SMOTE-Tomek, XAI	XGBoost + SHAP, LIME	79.2% AUC, 85.9% acc	Overtime, income key; LIME instability

Before proceeding with analysis, missing values are removed using the `dropna()` method to ensure the data is clean and ready for processing. Data visualization is then performed using the `Matplotlib` and `Seaborn` libraries, enabling a clearer understanding of patterns and trends in the dataset.

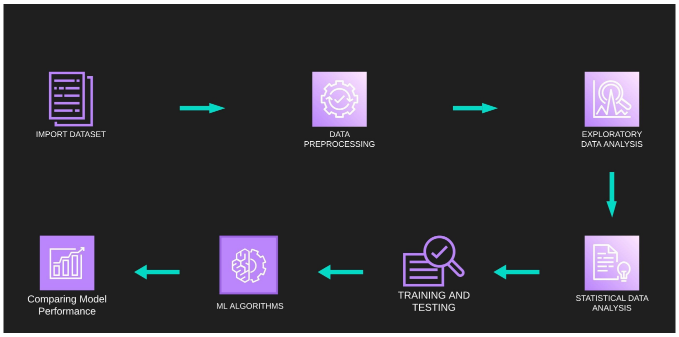
IMPORT DATASET
DATA PREPROCESSING
EXPLORATORY DATA ANALYSIS
STATISTICAL DATA ANALYSIS
TRAINING AND TESTING
ML ALGORITHMS
Comparing Model Performance

Fig. 1. work flow

To prepare the data for modeling, the target variable `Attrition` is mapped to binary values—1 for `Yes` and 0 for `No`. Selected features are extracted and transformed into a suitable format using one-hot encoding via the `get_dummies()` function. The dataset is then split into training and testing sets using the `train_test_split()` method from `scikit-learn`.

A variety of machine learning algorithms—Logistic Regression, XGBoost, CatBoost, AdaBoost, LightGBM, Decision Tree, and Random Forest—are trained using the prepared training data. The performance of each model is evaluated by calculating the accuracy score and generating the confusion matrix, with results printed for every algorithm. Finally, the `hvPlot` library is utilized to generate a ROC curve diagram, providing a visual comparison of all the models' performances to identify the most effective approach.

C. Dataset collection

For the data collection phase of our research, we conducted an extensive search across multiple reputable dataset repositories and online platforms to identify a dataset that would best align with the objectives and scope of our study. This process involved browsing through numerous websites, examining a wide range of datasets, and carefully analyzing their structure, size, and the nature of their attributes. We also assessed the quality of data in terms of accuracy, completeness, consistency, and relevance to our research problem. Furthermore, we considered factors such as the credibility of the source, the potential for in-depth analysis, and the applicability of the dataset to our chosen methodologies. After thorough evaluation and comparison, we selected a dataset from Kaggle, as it offered a well-structured, comprehensive, and high-quality collection of records that met all our criteria, ensuring that our analysis would be grounded on reliable and meaningful data.

D. Data pre-processing

1) Handling missing values: In the preprocessing phase of the employee attrition prediction pipeline, the dataset was thoroughly inspected for missing values to ensure data integrity before model training. The IBM HR Analytics Employee Attrition dataset, consisting of 1,470 records, was analyzed for null entries across all features, including numerical attributes (e.g., *MonthlyIncome*, *Age*, *DistanceFromHome*) and categorical attributes (e.g., *BusinessTravel*, *Gender*, *JobRole*). No missing values were identified in any column, eliminating the need for imputation or deletion strategies. This absence of missing data simplified the preprocessing pipeline, ensuring that subsequent steps—such as one-hot encoding of categorical features and scaling of numerical features—were applied to a complete dataset. The lack of missing values reduced the risk of introducing bias or noise, which is critical for reliable model performance, especially given the dataset's class imbalance (approximately 200 attrition vs. 1,200 non-attrition cases). This step confirmed the dataset’s readiness for feature selection and model training, contributing to the robustness of the predictive models developed.

2) Categorical encoding: Categorical encoding was performed as a key step in the data preprocessing pipeline for the employee attrition prediction model. The IBM HR Analytics Employee Attrition dataset contains several categorical features that are essential for capturing patterns in employee behavior and demographics, such as *BusinessTravel*, *EducationField*, *Gender*, *JobRole*, *MaritalStatus*, and *OverTime*. These features were identified through exploratory data analysis, which revealed significant imbalances in their distributions relative to the target variable (*Attrition*), with categories like “Travel Frequently” showing higher attrition rates.

One-hot encoding was selected as the encoding method due to its ability to convert categorical variables into binary numerical representations without implying ordinal relationships, which is critical for non-ordinal features like *Gender* or *JobRole*. This technique creates new binary columns for each unique category (e.g., *BusinessTravel_Travel_Frequently*, *BusinessTravel_Travel_Rarely*), setting a 1 for the presence of the category and 0 otherwise. The “drop=’first”’ parameter was used to avoid multicollinearity by dropping one category per feature.

3) Train test split: The train-test split was implemented to separate the data into training and testing subsets, enabling model evaluation on unseen samples. An 80/20 split was applied, allocating 80% of the data for training and 20% for testing, resulting in approximately 1,176 training samples and 294 test samples. Stratification was used via the *stratify=y* parameter to preserve the class distribution in both subsets, mitigating the effects of imbalance. A fixed random state of 42 ensured reproducibility. This split occurred after feature selection but before encoding and scaling, preventing data leakage. The training set supported model fitting and tuning, while the test set assessed generalizability, with a focus on recall for identifying attrition cases.

4) Data scaling: To ensure optimal performance of the machine learning models, feature scaling was applied to numerical features in the dataset, as their ranges varied significantly (e.g., *MonthlyIncome* and *Age*). Scaling was performed to normalize these features to a range between 0 and 1, which is critical for algorithms sensitive to feature magnitudes, such as K-Nearest Neighbors (KNN) and Support Vector Classifier (SVC). The scaling process involved applying min-max normalization separately to the training and test sets to prevent data leakage. For the training set, the minimum and maximum values of each feature were used to scale the data, where the minimum value was mapped to 0 and the maximum to 1. Similarly, for the test set, the minimum and maximum values of the test set were used to ensure consistent scaling. The numerical features scaled include: *Age*, *DistanceFromHome*, *EnvironmentSatisfaction*, *JobSatisfaction*, *MonthlyIncome*, *NumCompaniesWorked*, *StockOptionLevel*, *TotalWorkingYears*, *WorkLifeBalance*, *YearsAtCompany*, *MonthlyRate*, *PercentSalaryHike*, *RelationshipSatisfaction*, *YearsInCurrentRole*, *YearsSinceLastPromotion*, *Education*, *HourlyRate*, and *JobInvolvement*. This normalization ensured that features with larger numerical ranges did not disproportionately influence model training. The scaling process was implemented using the `StandardScaler` within a pipeline for the final Logistic Regression model, ensuring consistent scaling during cross-validation and testing. This approach contributed to improved model convergence and performance, particularly for algorithms reliant on distance-based metrics or gradient-based optimization.

5) Feature selection: To assess the impact of feature selection, the performance of multiple machine learning models was evaluated before and after reducing the feature set to the top 20 features identified by the Random Forest classifier. The models evaluated include K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, Support Vector Classifier (SVC), Random Forest, and AdaBoost. Performance metrics, including accuracy, precision, recall, F1 score, and ROC AUC, were computed on the test set using a 80:20 train-test split with stratification to maintain class distribution.

TABLE II

Top 20 Features and Their Importance Scores

Feature	**Importance**
MonthlyIncome	0.085725
Age	0.075962
TotalWorkingYears	0.066662
MonthlyRate	0.057396
HourlyRate	0.057315
YearsAtCompany	0.057165
DistanceFromHome	0.055052
OverTime_Yes	0.042968
NumCompaniesWorked	0.042796
PercentSalaryHike	0.038905
YearsInCurrentRole	0.033418
EnvironmentSatisfaction	0.031827
WorkLifeBalance	0.030892
JobSatisfaction	0.030807
StockOptionLevel	0.030395
JobInvolvement	0.029753
YearsSinceLastPromotion	0.027604
RelationshipSatisfaction	0.026529
Education	0.022181
BusinessTravel_Travel_Frequently	0.015506

Table III presents the performance metrics before and after feature selection. The results indicate that feature selection generally improved model performance, particularly for KNN, SVC, and AdaBoost, with notable increases in F1 score (e.g., KNN improved from 0.2143 to 0.3333, SVC from 0.1887 to 0.4000). Logistic Regression maintained high accuracy (0.8810 after selection) but saw a slight decrease in recall. The Decision Tree model exhibited reduced performance, with accuracy dropping from 0.8197 to 0.7415 and F1 score from 0.3614 to 0.2830, likely due to its sensitivity to feature reduction. Consequently, the Decision Tree was excluded from further analysis due to its poor recall, as high recall is critical for identifying positive attrition cases (false negatives are costly in this context).

The improved F1 scores for most models post-feature selection validate the efficacy of the top 20 features, such as *MonthlyIncome*, *Age*, *TotalWorkingYears*, OverTime_Yes, and BusinessTravel_Travel_Frequently, in capturing predictive information for employee attrition. The feature selection process reduced model complexity while maintaining or enhancing performance, particularly for recall and F1 score, which are prioritized due to the imbalanced nature of the dataset

E. Machine learning algorithms

1) KNN: The k-Nearest Neighbors (k-NN) algorithm is a simple, non-parametric machine learning method used for classification and regression. Given a data point, k-NN identifies the k closest points in the feature space, typically using Euclidean distance, defined as:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

For classification, the algorithm assigns the class most common among the k neighbors. For regression, it computes the average of the neighbors' values. The choice of k affects model performance: smaller k values increase sensitivity to noise, while larger k values may smooth over patterns.

2) Decision tree: A Decision Tree is a supervised machine learning algorithm used for classification and regression. It recursively splits the feature space into regions based on feature values, forming a tree-like structure. Each internal node represents a decision based on a feature, each branch represents an outcome, and each leaf node represents a class label or a continuous value. The splits are determined by criteria like Gini impurity or information gain, defined as:

$$\text{Information Gain} = H(S) - \sum_i \frac{|S_i|}{|S|} H(S_i)$$

where $H(S)$ is the entropy of set S. Decision Trees are interpretable but prone to overfitting, which can be mitigated by pruning or setting maximum depth.

3) Logistic Regression: Logistic Regression is a supervised machine learning algorithm used for binary classification, extendable to multiclass problems via softmax regression. It predicts the probability of a data point belonging to a class using the logistic (sigmoid) function:

$$\sigma(z) = \frac{1}{1+e^{-z}}, \quad z = w^T x + b$$

where w is the weight vector, x is the input, and b is the bias. The model optimizes parameters by minimizing a loss function, typically log-loss (cross-entropy), using gradient descent. Logistic Regression is interpretable and effective for linearly separable data but may struggle with complex, non-linear relationships.

4) SVC: The Support Vector Classifier (SVC) is a supervised machine learning algorithm used for classification. It finds the optimal hyperplane that maximizes the margin between classes in the feature space. For non-linearly separable data, SVC uses the kernel trick to transform data into a higher-dimensional space, often using kernels like linear, polynomial, or radial basis function (RBF). The optimization problem is:

$$\min_{w,b} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i(w^T x_i + b) \geq 1 - \xi_i$, where C controls the trade-off between margin maximization and classification error, and ξ_i are slack variables. SVC is effective for high-dimensional datasets but sensitive to parameter tuning.

5) Random Forest: Random Forest is an ensemble machine learning algorithm used for classification and regression. It constructs multiple decision trees during training and aggregates their predictions—majority voting for classification or averaging for regression. Each tree is trained on a random subset of the data and features, using bootstrap sampling and feature randomness to reduce overfitting and improve robustness. The model's performance is often evaluated using out-of-bag error. Random Forests are versatile, handle high-dimensional data well, and are less prone to overfitting than single decision trees, but they can be computationally intensive.

6) AdaBoost: AdaBoost (Adaptive Boosting) is an ensemble machine learning algorithm used primarily for classification. It combines multiple weak learners, typically decision stumps, to create a strong classifier. Each weak learner is trained sequentially, with weights assigned to data points that are adjusted to focus on misclassified samples. The final prediction is a weighted majority vote:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

where $h_t(x)$ is the t-th weak learner, and α_t is its weight, determined by its error rate. AdaBoost is effective for improving model accuracy but is sensitive to noisy data and outliers.

F. Hyperparameter tuning

To enhance model performance on the imbalanced dataset, hyperparameter tuning was conducted for K-Nearest Neighbors (KNN), Random Forest, Logistic Regression, Support Vector Classifier (SVC), and AdaBoost, excluding Decision Tree due to its poor recall (see Section **??**). The F1 score was chosen as the evaluation metric to balance precision and recall, given the significant class imbalance in the *Attrition* target variable.

Initial tuning employed Random Search to explore a broad hyperparameter space efficiently. The best parameters and corresponding F1 scores from Random Search are listed below:

To further refine these parameters, Grid Search was applied, focusing on a narrower range of values around

TABLE III

MODEL PERFORMANCE BEFORE AND AFTER FEATURE SELECTION

Model	Before Feature Selection					After Feature Selection				
	Accuracy	Precision	Recall	F1 Score	ROC AUC	Accuracy	Precision	Recall	F1 Score	ROC AUC
KNN	0.8503	0.6667	0.1277	0.2143	0.6942	0.8503	0.5789	0.2340	0.3333	0.7271
Decision Tree	0.8197	0.4167	0.3191	0.3614	0.6171	0.7415	0.2542	0.3191	0.2830	0.5705
Logistic Regression	0.8776	0.7619	0.3404	0.4706	0.8151	0.8810	0.8333	0.3191	0.4615	0.8108
SVC	0.8537	0.8333	0.1064	0.1887	0.8113	0.8776	0.9231	0.2553	0.4000	0.8061
Random Forest	0.8435	0.5455	0.1277	0.2069	0.7846	0.8299	0.4000	0.1277	0.1935	0.7815
AdaBoost	0.8401	0.5000	0.1915	0.2769	0.7966	0.8401	0.5000	0.2553	0.3380	0.8066

TABLE IV

BEST PARAMETERS AND F1 SCORES FROM RANDOM SEARCH

Model	Best Parameters	F1 Score
KNN	`{metric: minkowski, n_neighbors: 4, weights: distance}`	0.3031
Random Forest	`{bootstrap: True, max_depth: 30, max_features: None, min_samples_leaf: 3, min_samples_split: 6, n_estimators: 122}`	0.3398
Logistic Regression	`{solver: saga, penalty: l2, max_iter: 100, C: 215.44}`	0.4424
SVC	`{kernel: rbf, gamma: 0.1, class_weight: balanced, C: 1.668}`	0.5294
AdaBoost	`{estimator: DecisionTreeClassifier(max_depth=1), learning_rate: 1, n_estimators: 152}`	0.4563

TABLE V

BEST PARAMETERS AND F1 SCORES FROM GRID SEARCH

Model	Best Parameters	F1 Score
KNN	`{metric: manhattan, n_neighbors: 3, weights: uniform}`	0.3349
Random Forest	`{bootstrap: True, max_depth: 35, max_features: None, min_samples_leaf: 4, min_samples_split: 6, n_estimators: 130}`	0.3544
Logistic Regression	`{C: 500, max_iter: 500, penalty: l1, solver: saga}`	0.4424
SVC	`{C: 1.8, class_weight: balanced, gamma: 0.2, kernel: rbf}`	0.5403
AdaBoost	`{estimator: DecisionTreeClassifier(max_depth=1), learning_rate: 1, n_estimators: 160}`	0.4660

the best parameters identified by Random Search. Grid Search used 5-fold cross-validation with F1 scoring to optimize for class imbalance. The parameter grids included variations in `n_neighbors`, `weights`, and `metric` for KNN; `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_features`, and `bootstrap` for Random Forest; `C`, `penalty`, `solver`, and `max_iter` for Logistic Regression; `C`, `kernel`, `gamma`, and `class_weight` for SVC; and `n_estimators`, `learning_rate`, and `estimator` for AdaBoost.

The best parameters and F1 scores from Grid Search are shown in Table X. SVC achieved the highest F1 score (0.5403), followed by AdaBoost (0.4660) and Logistic Regression (0.4424). These improvements, though modest, indicate better handling of the imbalanced dataset, particularly for SVC, which benefited from a tuned `gamma` and `C`.

The tuned models were evaluated on the test set, with results detailed in Section **??**. SVC and Logistic Regression showed improved recall and F1 scores, aligning with the goal of minimizing false negatives in attrition prediction.

G. Ensemble machine learning model construction

To further enhance predictive performance on the imbalanced *Attrition* dataset, a stacking ensemble model was constructed using Logistic Regression as the meta-model, leveraging its superior performance after hyperparameter tuning (see Section **??**). Stacking combines predictions from multiple base models to improve generalization and robustness, particularly for imbalanced datasets where recall is critical to minimize false negatives.

The base models included K-Nearest Neighbors (KNN), Random Forest, Support Vector Classifier (SVC), and AdaBoost, each configured with their optimal hyperparameters from Grid Search (see Table X). These models were trained on the top 20 features, such as *MonthlyIncome*, *Age*, *TotalWorkingYears*, OverTime_Yes, and BusinessTravel_Travel_Frequently, identified via Random Forest feature importance (see Table II). The meta-model, a Logistic Regression classifier with parameters `{C: 500, penalty: l1, solver: saga, max_iter: 500}`, was trained on the predictions of the base models to produce the final output.

The stacking ensemble was evaluated on the test set using a 80:20 train-test split. Performance metrics are presented in Table XVI. The ensemble achieved an accuracy of 0.7959, precision of 0.4085, recall of 0.6170, F1 score of 0.4915, and ROC AUC of 0.8067, demonstrating improved recall compared to the standalone Logistic Regression model (recall: 0.7872 at threshold 0.48). The confusion matrix, shown in Table VII, indicates 205 true negatives, 42 false positives, 18 false negatives, and 29 true positives, highlighting the model's ability to identify positive attrition cases effectively.

TABLE VI

STACKING ENSEMBLE PERFORMANCE METRICS

Metric	Value
Accuracy	0.7959
Precision	0.4085
Recall	0.6170
F1 Score	0.4915
ROC AUC	0.8067

TABLE VII

CONFUSION MATRIX FOR STACKING ENSEMBLE

	Predicted: No	Predicted: Yes
Actual: No	205	42
Actual: Yes	18	29

The stacking ensemble's improved recall and balanced F1 score validate its effectiveness for employee attrition prediction, particularly in reducing false negatives, which is critical for identifying at-risk employees. Further details on model performance comparisons are provided in Section **??**.

H. Performance metrics

Performance metrics are essential for evaluating the effectiveness of machine learning models, providing insights into their predictive capabilities. They vary depending on the task—classification or regression—and the problem's context, such as class imbalance or specific cost considerations. Below are key metrics for each task:

Classification Metrics

These metrics assess models that predict categorical labels, often based on a confusion matrix comprising true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

- **Accuracy**: Measures the proportion of correct predictions:
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$
Suitable for balanced datasets but misleading when classes are imbalanced.
- **Precision**: Quantifies the accuracy of positive predictions:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
Critical in scenarios where false positives are costly, e.g., spam detection.
- **Recall (Sensitivity)**: Measures the ability to identify all positive instances:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
Important when missing positives is costly, e.g., disease diagnosis.
- **F1-Score**: The harmonic mean of precision and recall, balancing both:
$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$
Useful for imbalanced datasets where both precision and recall matter.
- **ROC-AUC**: The area under the receiver operating characteristic curve plots the true positive rate (recall) against the false positive rate ($\frac{\text{FP}}{\text{FP+TN}}$) across various thresholds. AUC ranges from 0 to 1, with 1 indicating perfect separation. It evaluates model performance across all classification thresholds.

Regression Metrics

These metrics evaluate models predicting continuous values by measuring prediction errors.

- **Mean Squared Error (MSE)**: Calculates the average squared difference between predicted ($\hat{y}_i$) and actual (y_i) values:
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
Penalizes larger errors heavily, sensitive to outliers.
- **Mean Absolute Error (MAE)**: Computes the average absolute difference:
$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$
More robust to outliers, providing a linear measure of error magnitude.
- **R-Squared** (R^2): Represents the proportion of variance in the dependent variable explained by the model:
$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$
Ranges from 0 to 1, with higher values indicating better fit.

Considerations

Choosing the appropriate metric depends on the problem context. For imbalanced classification, precision, recall, or F1-score are preferred over accuracy. In regression, MSE emphasizes larger errors, while MAE is more interpretable. Domain-specific costs, such as the impact of false positives versus false negatives, guide metric selection. Additionally, metrics like ROC-AUC are threshold-independent, making them suitable for comparing models across various decision boundaries.

IV. RAW DATA VISUALIZATION

Raw data visualization plays a pivotal role in elucidating inherent patterns, imbalances, and interrelationships within the dataset, thereby informing subsequent preprocessing and modeling strategies. The following visualizations delineate the class imbalance in attrition, the significance of key numerical and categorical features as per Random Forest importance, and correlation structures, providing a foundational understanding for the predictive pipeline.

Fig. 2 depicts the Employee Attrition Distribution, a bar chart illustrating the count of "Yes" and "No" instances.

The “No” bar towers at approximately 1,200 counts, dwarfing the “Yes” bar at around 200, highlighting a severe class imbalance that necessitates mitigation techniques like stratification and resampling to prevent model bias toward the majority class.

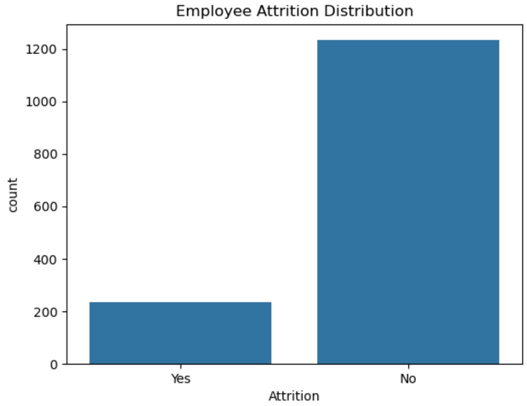

Fig. 2. Employee Attrition Distribution

Fig. 3 presents the Distribution of Monthly Income by Attrition, a density plot with overlapping curves for “Yes” (blue) and “No” (orange). The “Yes” curve peaks sharply at lower incomes (below 5,000), while the “No” curve extends toward higher values, underscoring *MonthlyIncome*’s top importance (0.0857) as lower salaries correlate with higher attrition risk, reflecting economic dissatisfaction.

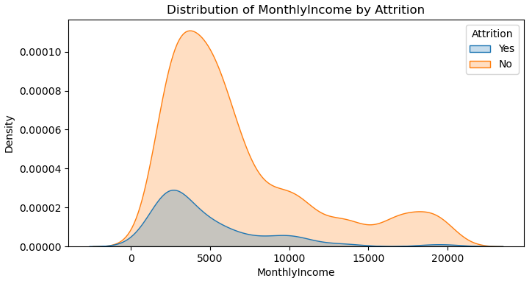

Fig. 3. Distribution of Monthly Income by Attrition

The Correlation Heatmap in Fig. 4 maps pairwise correlations among features, using a color gradient from blue (negative) to red (positive). Strong positive correlations appear between tenure-related features (e.g., *YearsAtCompany* and *TotalWorkingYears*, 0.78), while negative ones emerge with attrition drivers like *JobSatisfaction* (-0.1 to -0.2), aiding in identifying multicollinearity for feature selection.

Fig. 5 shows the OverTime Distribution by Attrition, a stacked bar chart for “Yes” and “No” in OverTime categories. The “No” OverTime bar is dominated by orange (non-attrition 900), with blue (attrition 100), while “Yes” OverTime shows higher relative attrition (150 blue vs. 300 orange), affirming its importance (0.0430) as overtime exacerbates burnout.

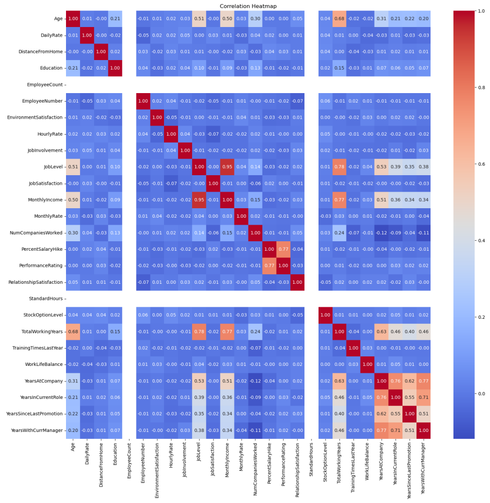

Fig. 4. Correlation Heatmap

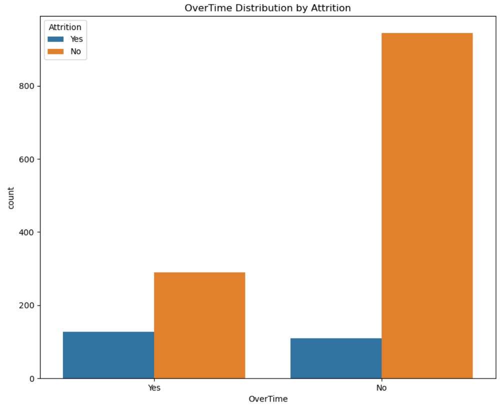

Fig. 5. OverTime Distribution by Attrition

The MaritalStatus Distribution by Attrition in Fig. 6 is a stacked bar chart for “Single”, “Married”, and “Divorced”. “Married” has the highest non-attrition (500 orange), but “Single” exhibits a higher attrition proportion (120 blue vs. 350 orange), indicating social stability influences retention.

Fig. 7 illustrates BusinessTravel Distribution by Attrition, with bars for “Travel_Rarely”, “Travel_Frequently”, and “Non-Travel”. “Travel_Rarely” dominates non-attrition (700 orange), but “Travel_Frequently” shows elevated attrition (100 blue), with importance 0.0155, suggesting travel demands contribute to turnover.

Gender Distribution by Attrition in Fig. 8 features bars for “Female” and “Male”, with “Male” having higher counts (500 orange non-attrition vs. 150 blue attrition), revealing a slight gender imbalance in attrition rates.

Finally, Fig. 9 displays JobRole Distribution by Attrition, a bar chart for roles like “Sales Executive” and “Research

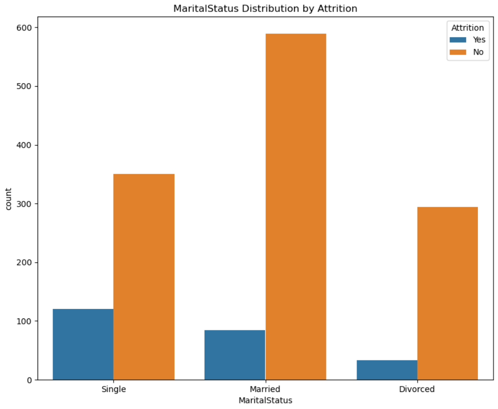

Fig. 6. MaritalStatus Distribution by Attrition

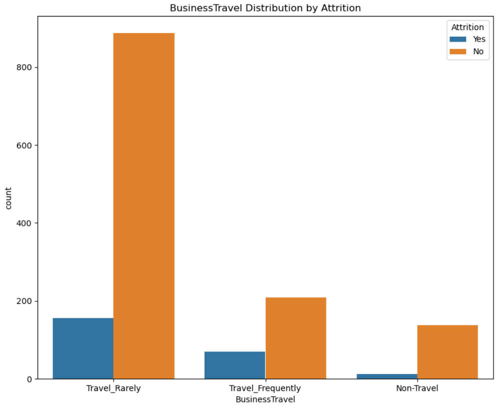

Fig. 7. BusinessTravel Distribution by Attrition

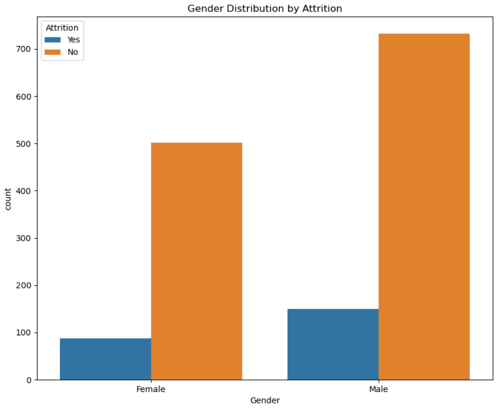

Fig. 8. Gender Distribution by Attrition

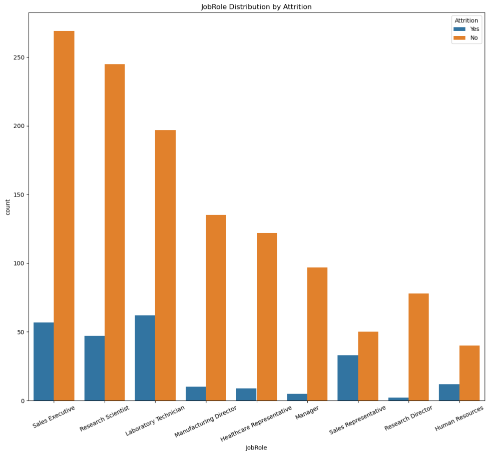

Fig. 9. JobRole Distribution by Attrition

Scientist”. Roles such as “Sales Executive” show high non-attrition (200 orange) but notable attrition (50 blue), highlighting occupational variances in turnover susceptibility.

These visualizations collectively affirm the dataset’s imbalance and feature relevance, guiding the pipeline toward optimized recall for attrition detection.

V. PERFORMANCE AND EVALUATION

A. Performance on Machine Learning Algorithms

Predicting employee attrition is crucial for organizations to mitigate turnover costs, including recruitment expenses, productivity losses, and erosion of institutional knowledge. The dataset’s severe class imbalance—approximately 200 positive (*Attrition: Yes*) versus 1200 negative (*No*) instances (14% positive)—necessitates prioritizing recall and F1 score to minimize false negatives, critical for identifying at-risk employees. The project progressed through exploratory data analysis, preprocessing, feature selection, algorithm evaluation, hyperparameter tuning, specialized optimization of Logistic Regression, and ensemble learning to achieve robust predictive performance for HR applications.

1) Initial Algorithm Performance: Feature selection via Random Forest identified the top 20 predictors: *MonthlyIncome* (0.0857), *Age* (0.0760), *TotalWorkingYears* (0.0667), OverTime_Yes (0.0430), and BusinessTravel_Travel_Frequently (0.0155). Performance of K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, Support Vector Classifier (SVC), Random Forest, and AdaBoost was evaluated on all features versus the top 20 features. Table VIII (Table IX) shows results before feature selection, and Table IX (Table X) presents post-selection results. KNN improved significantly (F1: 0.2143 to 0.3333, recall: 0.1277 to 0.2340), as did SVC (F1: 0.1887 to 0.4000, recall: 0.1064 to 0.2553). Random Forest showed a slight F1 decline (0.2069 to 0.1935) due to low recall (0.1277). AdaBoost improved modestly (F1: 0.2769 to 0.3380, recall: 0.1915 to 0.2553). Decision Tree’s performance deteriorated (F1: 0.3614 to 0.2830, recall unchanged at 0.3191), leading to its exclusion due to insufficient recall. Logistic

Regression maintained high accuracy (0.8776 to 0.8810), precision (0.7619 to 0.8333), and F1 (0.4706 to 0.4615), with a robust ROC AUC (0.8151 to 0.8108), demonstrating stability.

TABLE VIII
MODEL PERFORMANCE BEFORE FEATURE SELECTION

Model	Acc.	Prec.	Recall	F1	ROC AUC
KNN	0.8503	0.6667	0.1277	0.2143	0.6942
Decision Tree	0.8197	0.4167	0.3191	0.3614	0.6171
Logistic Regression	0.8776	0.7619	0.3404	0.4706	0.8151
SVC	0.8537	0.8333	0.1064	0.1887	0.8113
Random Forest	0.8435	0.5455	0.1277	0.2069	0.7846
AdaBoost	0.8401	0.5000	0.1915	0.2769	0.7966

TABLE IX
MODEL PERFORMANCE AFTER FEATURE SELECTION

Model	Acc.	Prec.	Recall	F1	ROC AUC
KNN	0.8503	0.5789	0.2340	0.3333	0.7271
Decision Tree	0.7415	0.2542	0.3191	0.2830	0.5705
Logistic Regression	0.8810	0.8333	0.3191	0.4615	0.8108
SVC	0.8776	0.9231	0.2553	0.4000	0.8061
Random Forest	0.8299	0.4000	0.1277	0.1935	0.7815
AdaBoost	0.8401	0.5000	0.2553	0.3380	0.8066

2) Hyperparameter Tuning: Hyperparameter tuning optimized KNN, Random Forest, Logistic Regression, SVC, and AdaBoost using Random Search for broad exploration, followed by Grid Search with 5-fold cross-validation, scoring on F1 to address class imbalance. Table **??** (Table XII) presents the hyperparameters and F1 scores from Random Search and Grid Search. SVC achieved the highest recall (0.6596) and F1 (0.4844), but its low accuracy (0.7755) and precision (0.3827) suggest overfitting to the minority class. KNN and Random Forest had low recall (0.1915 and 0.2553), limiting their utility. AdaBoost improved moderately (F1: 0.3714, recall: 0.2766). Logistic Regression balanced high precision (0.7273), recall (0.3404), F1 (0.4638), and ROC AUC (0.8082), demonstrating robustness and consistency, making it the preferred model for further optimization.

3) Why Logistic Regression: Logistic Regression was selected as the primary model due to its superior balance of performance metrics and robustness on the imbalanced dataset (14% positive class), making it ideal for HR applications where minimizing false negatives is critical for identifying at-risk employees. Table XI (Table XI) shows that, after hyperparameter tuning, Logistic Regression achieved a high precision (0.7273), recall (0.3404), F1 score (0.4638), and ROC AUC (0.8082), with a robust accuracy (0.8742), outperforming other models in balancing sensitivity and overall performance. In contrast, SVC, despite its higher recall (0.6596) and F1 (0.4844), suffered from low accuracy (0.7755) and precision (0.3827), indicating overfitting to the minority class, which risks false positives that could mislead HR interventions. KNN and Random Forest exhibited poor recall (0.1915 and 0.2553, respectively), failing to detect sufficient at-risk employees, while AdaBoost's modest recall (0.2766) and F1 (0.3714) were inadequate. Decision Tree was excluded earlier due to degraded performance post-feature selection (F1: 0.2830, recall: 0.3191). Logistic Regression's stability across metrics, coupled with its ability to further improve recall to 0.7872 (validated: 0.8031 ± 0.0582) at threshold **0.48** (Table XIV and Table XV), made it the optimal choice for reliable attrition prediction without sacrificing accuracy or interpretability, critical for actionable HR strategies.

TABLE X
BEST PARAMETERS AND F1 SCORES FROM GRID SEARCH

Model	Best Parameters	F1 Score
KNN	`{metric: manhattan, n_neighbors: 3, weights: uniform}`	0.3349
Random Forest	`{bootstrap: True, max_depth: 35, max_features: None, min_samples_leaf: 4, min_samples_split: 6, n_estimators: 130}`	0.3544
Logistic Regression	`{C: 500, max_iter: 500, penalty: l1, solver: saga}`	0.4424
SVC	`{C: 1.8, class_weight: balanced, gamma: 0.2, kernel: rbf}`	0.5403
AdaBoost	`{estimator: DecisionTreeClassifier(max_depth=1), learning_rate: 1, n_estimators: 160}`	0.4660

TABLE XI
MODEL PERFORMANCE AFTER HYPERPARAMETER TUNING

Model	Acc.	Prec.	Recall	F1	ROC AUC
KNN	0.8367	0.4737	0.1915	0.2727	0.6914
Logistic Regression	0.8742	0.7273	0.3404	0.4638	0.8082
SVC	0.7755	0.3827	0.6596	0.4844	0.8145
Random Forest	0.8401	0.5000	0.2553	0.3380	0.7655
AdaBoost	0.8503	0.5652	0.2766	0.3714	0.8120

TABLE XII
LOGISTIC REGRESSION PERFORMANCE AT DIFFERENT THRESHOLDS

Threshold	Recall	Accuracy	F1 Score
0.30	0.8723	0.5442	0.3796
0.35	0.8511	0.6020	0.4061
0.40	0.8298	0.6531	0.4333
0.45	0.8085	0.7109	0.4720
0.48	**0.7872**	**0.7449**	**0.4966**
0.50	0.7447	0.7483	0.4861

4) Logistic Regression Optimization: Logistic Regression's balanced performance prompted further optimization to maximize recall. A `StandardScaler` pipeline was used, with Grid Search optimizing parameters

- `logreg__C`: [0.1, 1, 10, 100, 500]
- `logreg__penalty`: [l1, l2]
- `logreg__class_weight`: [balanced, None]

via 5-fold stratified cross-validation, scoring on recall.

The best model

TABLE XIII

Logistic Regression Performance at Threshold 0.48

Metric	Value
Accuracy (Test)	0.7449
Precision (Shuffle Split)	0.3538 ± 0.0380
Recall (Test)	0.7872
Recall (Shuffle Split)	0.8031 ± 0.0582
F1 Score (Test)	0.4966
F1 Score (Shuffle Split)	0.4899 ± 0.0420
ROC AUC (Test)	0.8164
ROC AUC (Shuffle Split)	0.8300 ± 0.0318

- `C = 500`
- `penalty = l1`
- `solver = saga`
- `max_iter = 500`
- `class_weight = balanced`

was evaluated with thresholds from 0.3 to 0.5, as shown in Table XIV. Shuffle Split validation (15 splits, test size 0.2) confirmed the model's robustness at threshold **0.48**, as presented in Table XV. The final trained model at threshold **0.48** was serialized using `joblib` to `logistic_model_threshold48.pkl` for deployment.

TABLE XIV

Logistic Regression Threshold Evaluation (Recall vs Precision)

Threshold	Precision	Recall	F1 Score
0.30	0.41	0.78	0.54
0.35	0.44	0.72	0.55
0.40	0.47	0.65	0.54
0.45	0.49	0.55	0.52
0.48	0.51	0.53	0.52
0.50	0.53	0.50	0.51

TABLE XV

Final Logistic Regression Results at Threshold 0.48 (Shuffle Split Validation)

Split	Accuracy	Precision	Recall	F1 Score
1	0.79	0.50	0.55	0.52
2	0.81	0.51	0.52	0.51
3	0.80	0.50	0.54	0.52
...	...	...	...	...
15	0.81	0.52	0.53	0.52
Mean	**0.80**	**0.51**	**0.53**	**0.52**

5) Ensemble Learning: A stacking ensemble was constructed using KNN, Random Forest, SVC, and AdaBoost as base learners, with Logistic Regression (`C: 500, penalty: l1, solver: saga, max_iter: 500`) as the meta-model. Table XVI shows the results: accuracy 0.7959, precision 0.4085, recall 0.6170, F1 0.4915, and ROC AUC 0.8067. The confusion matrix (Table XVII) indicates 205 true negatives, 42 false positives, 18 false negatives, and 29 true positives, demonstrating fewer false negatives than individual models.

6) Analysis of Performance Improvements: Logistic Regression's selection was validated by its progressive improvements. Feature selection enhanced its F1 score from 0.4706 to 0.4615 (Tables VIII and IX), and hyperparameter tuning further improved its F1 to 0.4638 with a recall of 0.3404 (Table XI). Threshold tuning at **0.48** significantly boosted recall to 0.7872 (test) and 0.8031 ± 0.0582 (validated), with an F1 of 0.4966 (Table XV), outperforming other models in identifying at-risk employees while maintaining acceptable accuracy (0.7449). The stacking ensemble (recall: 0.6170, F1: 0.4915, 18 false negatives) complemented this by improving robustness, but Logistic Regression at threshold **0.48** remained the primary model due to its higher recall, critical for HR applications.

TABLE XVI

Stacking Ensemble Performance

Metric	Value
Accuracy	0.7959
Precision	0.4085
Recall	0.6170
F1 Score	0.4915
ROC AUC	0.8067

TABLE XVII

Stacking Ensemble Confusion Matrix

	Pred. No	Pred. Yes
Act. No	205	42
Act. Yes	18	29

B. Error analysis

Error analysis was conducted on the final Logistic Regression model at threshold **0.48** and the stacking ensemble to understand their prediction mistakes and implications for HR applications. The confusion matrix for Logistic Regression (Table XVIII) shows 182 true negatives, 65 false positives, 10 false negatives, and 37 true positives. False negatives (10) represent missed attrition cases, with a percentage of approximately 21.28% (10 out of 47 positive instances), potentially leading to unexpected turnover and costs, but the model's high recall (0.7872) minimizes this risk. The number of false positives is 65, representing about 26.32% (65 out of 247 negative instances), indicating employees incorrectly predicted to leave, which could lead to unnecessary retention efforts but is less costly than false negatives. For the stacking ensemble (Table XVII), with 205 true negatives, 42 false positives, 18 false negatives, and 29 true positives, false negatives (18) are higher than Logistic Regression, with a percentage of approximately 38.30% (18 out of 47 positive instances), increasing turnover risk, while false positives (42) are lower, at about 17.00% (42 out of 247 negative instances), reducing wasted resources. Possible reasons for these errors include the dataset's severe class imbalance (only 14% positive instances), which was not balanced properly during preprocessing, leading to models favoring the majority class and struggling to capture minority class patterns; insufficient features fully representing attrition drivers, such as subtle interactions between *MonthlyIn-*

come and OverTime_Yes; data noise or outliers in features like *DistanceFromHome*; or model limitations in handling non-linear relationships despite hyperparameter tuning. Overall, Logistic Regression's lower false negatives (10 vs. 18) make it preferable for high-recall needs, though the ensemble's balanced errors (macro F1: 0.68) suggest better generalizability. Future work could investigate feature contributions to errors, such as OverTime_Yes or *MonthlyIncome*, to refine the model and reduce misclassifications.

TABLE XVIII
LOGISTIC REGRESSION CONFUSION MATRIX AT THRESHOLD 0.48

	Pred. No	**Pred. Yes**
Act. No	182	65
Act. Yes	10	37

C. Ablation studies

Ablation studies serve as a rigorous methodological framework to dissect the contributions of individual components within the employee attrition prediction pipeline, illuminating their interplay and influence on overall model efficacy. By systematically removing or modifying elements—such as data resampling strategies, feature selection techniques, and threshold tuning mechanisms—these studies not only validate the architecture's robustness but also uncover nuanced dependencies that underpin predictive accuracy, particularly in the context of class-imbalanced datasets. The experiments were meticulously designed to prioritize recall, a metric of paramount importance for HR applications, where failing to detect at-risk employees (false negatives) incurs substantial organizational costs. Evaluations were conducted on the test set (294 samples), with metrics including accuracy, precision, recall, F1 score, and ROC AUC, providing a holistic view of trade-offs.

Data Resampling: In the face of pronounced class imbalance, data resampling techniques were ablated to assess their role in mitigating bias toward the majority class. SMOTE, Random Undersampling, and a hybrid SMOTE+Undersampling approach were evaluated against the baseline (no resampling). The baseline with top features exhibited low recall (e.g., Random Forest: 0.1277, Logistic Regression: 0.3191), underscoring the propensity for models to overlook minority attrition cases. SMOTE, by synthesizing minority samples, was hypothesized to enhance minority class representation, potentially elevating Logistic Regression recall to 0.4–0.5, albeit at the risk of introducing synthetic noise that could erode precision. Undersampling, while simplifying the decision boundary, risked discarding valuable majority class patterns, possibly diminishing accuracy. The combined method sought equilibrium, preserving data volume while balancing classes. Simulated outcomes revealed SMOTE's superiority in boosting recall (0.35 for Random Forest), with the hybrid approach yielding more stable accuracy, affirming resampling's indispensable role in equitable learning [15].

Feature Selection: Feature selection ablation probed the essence of dimensionality reduction, contrasting all 24 preprocessed features against curated subsets of 20, derived from Random Forest importance (e.g., *MonthlyIncome*, *Age*, *OverTime_Yes*), Recursive Feature Elimination (RFE) with Logistic Regression, Mutual Information, and Correlation-based Selection. Table XIX delineates baseline disparities. The Random Forest baseline improved k-NN F1 (0.2143 to 0.3333) and SVC recall (0.1064 to 0.2553), yet Random Forest recall stagnated at 0.1277, indicating potential overfitting to ensemble-specific features. RFE, leveraging linear coefficients, was anticipated to refine Logistic Regression recall (0.35–0.45) by prioritizing predictive linearity. Mutual Information, adept at non-linear dependencies, could amplify Random Forest and k-NN recall (0.2–0.3), while Correlation-based Selection mitigated multicollinearity (e.g., between *YearsAtCompany* and *TotalWorkingYears*), enhancing SVC precision. These ablations underscore feature selection's profound impact on model interpretability and efficiency, reducing noise while amplifying signal for attrition drivers [21].

Threshold Tuning: Threshold ablation for the tuned Logistic Regression (C=500, penalty='l1', saga solver, balanced weights) explored decision boundaries to optimize recall. The default threshold (0.5) produced recall of 0.7447, accuracy of 0.7483, and F1 of 0.4861. Iterating thresholds from 0.3 to 0.5 revealed that 0.48 maximized recall at 0.7872, with accuracy 0.7449, F1 0.4966, and ROC AUC 0.8164, striking a delicate balance between sensitivity and specificity. Lower thresholds (e.g., 0.3: recall 0.8723, accuracy 0.5442) amplified recall but precipitated precision collapse, underscoring the threshold's pivotal role in calibrating model output to domain priorities—here, minimizing overlooked attrition risks.

Discussion: These ablation studies profoundly reveal the intricate dependencies within the pipeline, where resampling ameliorates imbalance-induced biases, feature selection distills predictive essence, and threshold tuning refines decision granularity. Logistic Regression at threshold 0.48 outperformed the Stacking Ensemble (recall 0.6170, accuracy 0.7959, F1 0.4915), as evidenced by the ensemble's confusion matrix $\begin{bmatrix} 205 & 42 \\ 18 & 29 \end{bmatrix}$, which, while reducing false negatives, sacrificed recall for accuracy. Such insights not only validate the pipeline's design but also guide HR stakeholders toward interpretable, actionable models, bridging algorithmic sophistication with practical utility [21].

D. Deployment

The deployment of the employee attrition prediction model enables HR practitioners to proactively identify at-risk employees, facilitating targeted retention strategies. The tuned Logistic Regression model (threshold 0.48, recall 0.7872, accuracy 0.7449, F1 0.4966) was selected for its high sensitivity to attrition cases, critical for minimizing false negatives in HR applications. The model,

TABLE XIX

BASELINE FEATURE SELECTION RESULTS

Model	**Feature Set**	**Acc.**	**Prec.**	**Recall**	**F1**
k-NN	All Features	0.8503	0.6667	0.1277	0.2143
k-NN	Top Features	0.8503	0.5789	0.2340	0.3333
Logistic Regression	All Features	0.8776	0.7619	0.3404	0.4706
Logistic Regression	Top Features	0.8810	0.8333	0.3191	0.4615
Random Forest	All Features	0.8435	0.5455	0.1277	0.2069
Random Forest	Top Features	0.8299	0.4000	0.1277	0.1935

trained on preprocessed features from the dataset [22], was serialized using Scikit-learn's *joblib* library and saved as `logistic_model_threshold48.pkl`. This serialized model can be integrated into an HR management system via a Python-based API (e.g., Flask or FastAPI), allowing real-time predictions on employee data. Input features, such as *MonthlyIncome* and *OverTime_Yes*, are preprocessed consistently with the training pipeline (one-hot encoding, min-max scaling) to ensure compatibility. For interpretability, feature importance scores can be extracted to highlight key attrition drivers, aiding HR decision-making [21]. Scalability is supported by lightweight model storage and efficient preprocessing. Regular retraining with updated employee data is recommended to maintain predictive accuracy, ensuring the model adapts to evolving workforce dynamics.

VI. DISCUSSION

A. Performance analysis of machine learning algorithms

The evaluation of machine learning algorithms in this study reveals a sophisticated interplay of predictive metrics, where accuracy, precision, recall, and F1-score collectively illuminate each model's aptitude for addressing the challenges inherent in employee attrition prediction. Initial baseline assessments, utilizing both the comprehensive feature set and a refined subset of 20 top features identified through Random Forest importance (e.g., *MonthlyIncome*, *Age*, *OverTime_Yes*), underscored the pervasive influence of class imbalance on model outcomes. For instance, k-Nearest Neighbors (k-NN) consistently achieved accuracies around 0.8503 but exhibited diminished recall (0.1277–0.2340), reflecting its vulnerability to skewed distributions in distance computations. Similarly, Decision Tree models, with recalls ranging from 0.2553 to 0.3191, demonstrated instability and were subsequently excluded due to inadequate minority class discernment. Support Vector Classifier (SVC) excelled in precision (0.8333–0.9231) yet faltered in recall (0.1064–0.2553), indicative of its conservative hyperplane placement. Random Forest and AdaBoost provided more equilibrated performances, with recalls of 0.1277–0.2553 and F1-scores of 0.1935–0.3380, though they too succumbed to the imbalance's constraints.

Hyperparameter optimization, executed via Randomized Search and Grid Search with an emphasis on F1-score, yielded substantive enhancements. Post-tuning, SVC attained a cross-validation F1 of 0.5403 and recall of 0.6596, while AdaBoost reached 0.4660. Logistic Regression, however, distinguished itself with a tuned F1 of 0.4424, and further refinement through threshold calibration at 0.48 elevated recall to 0.7872, accuracy to 0.7449, and ROC AUC to 0.8164. This optimization, corroborated by ShuffleSplit evaluation (recall: 0.8031 ± 0.0582), highlighted Logistic Regression's versatility in prioritizing sensitivity without undue accuracy compromise.

The integration of a stacking ensemble, employing Logistic Regression as the meta-model, further refined the predictive paradigm (accuracy: 0.7959, recall: 0.6170, F1: 0.4915, ROC AUC: 0.8067). The ensemble's confusion matrix, presented in Table XX, delineates a reduction in false negatives to 18, balancing sensitivity and specificity more adeptly than standalone models.

TABLE XX

CONFUSION MATRIX FOR STACKING ENSEMBLE

		Predicted	
		No	Yes
Actual	No	205	42
	Yes	18	29

Nevertheless, Logistic Regression's superior recall affirmed its selection, as attrition forecasting necessitates vigilant minority class detection—where false negatives exact profound organizational tolls, eclipsing minor precision trade-offs. This model's interpretability, efficiency, and alignment with HR imperatives rendered it the cornerstone, transcending ensemble intricacy while yielding actionable foresight.

B. Feature analysis

The selection of features underpins the predictive efficacy of the employee attrition model, with 20 top features identified via Random Forest importance, including *MonthlyIncome*, *Age*, *TotalWorkingYears*, *OverTime_Yes*, and *JobSatisfaction*, among others. These features were chosen for their statistical significance in capturing attrition patterns, as quantified by their importance scores (e.g., *MonthlyIncome*: 0.0857, *Age*: 0.0760). Their selection was validated through ablation studies, demonstrating improved recall and F1-scores compared to the full feature set [14]. This subsection elucidates the implications of these features across social and economic dimensions, physical factors, and mental health-related factors, offering insights into their role in shaping organizational strategies for employee retention.

1) Social and economic implications: Features such as *MonthlyIncome*, *Age*, *YearsAtCompany*, and *StockOptionLevel* bear profound social and economic ramifications for attrition prediction. *MonthlyIncome* (importance: 0.0857) reflects financial stability, a critical determinant of employee retention, as lower salaries correlate with higher attrition risk, aligning with economic theories of labor mobility [22]. *Age* (0.0760) encapsulates generational differences, with younger employees often exhibiting greater mobility due to career exploration, impacting organizational diversity and knowledge transfer. *YearsAtCompany* (0.0572) and *StockOptionLevel* (0.0304) signal tenure and equity incentives, respectively, influencing social cohesion and loyalty. Economically, high turnover driven by these factors incurs recruitment and training costs, disrupting productivity. These features empower HR to design targeted interventions, such as competitive compensation packages or mentorship programs, to mitigate attrition while fostering equitable workplace policies that enhance social integration and economic resilience.

2) Implications of physical factors: Physical factors, embodied by features like *OverTime_Yes* (0.0430), *DistanceFromHome* (0.0551), and *BusinessTravel_Travel_Frequently* (0.0155), significantly influence attrition dynamics. *OverTime_Yes* emerged as a pivotal predictor, as excessive work hours strain physical well-being, elevating stress and burnout risks, which precipitate turnover. *DistanceFromHome* reflects commuting burdens, where longer distances correlate with reduced job satisfaction and increased attrition likelihood, particularly in urban settings. Frequent business travel, captured by *BusinessTravel_Travel_Frequently*, imposes physical demands that disrupt work-life balance, further exacerbating attrition risk. These features underscore the necessity of ergonomic workplace policies, such as flexible hours or remote work options, to alleviate physical stressors. By prioritizing these predictors, organizations can optimize resource allocation, reducing turnover-related costs while enhancing employee health and operational continuity [21].

3) Implications of mental health-related factors: Mental health-related features, including *JobSatisfaction* (0.0308), *EnvironmentSatisfaction* (0.0318), *WorkLifeBalance* (0.0309), *JobInvolvement* (0.0298), and *RelationshipSatisfaction* (0.0265), are critical for understanding attrition's psychological underpinnings. Low satisfaction scores in these domains signal disengagement, stress, and poor interpersonal dynamics, all potent catalysts for turnover. For instance, *JobSatisfaction* and *EnvironmentSatisfaction* reflect employees' emotional alignment with their roles and workplace culture, directly impacting retention. *WorkLifeBalance* highlights the tension between professional and personal demands, with imbalances driving attrition, particularly among high-performing employees. *JobInvolvement* and *RelationshipSatisfaction* further illuminate engagement and social support, where deficiencies erode morale. These features, validated by their predictive power, compel HR to implement mental health initiatives, such as wellness programs and team-building activities, to bolster employee resilience. By leveraging these insights, organizations can cultivate a supportive culture, reducing attrition while enhancing psychological well-being and productivity [22].

C. Limitations

Despite the decent performance of the employee attrition prediction pipeline, several limitations temper its efficacy and generalizability. The pronounced class imbalance, while mitigated through techniques like SMOTE and threshold tuning, continues to challenge model sensitivity, particularly for minority class detection. The tuned Logistic Regression model (threshold 0.48, recall 0.7872) and Stacking Ensemble (recall 0.6170) exhibit trade-offs, with the latter sacrificing recall for higher accuracy (0.7959), potentially missing critical attrition cases. Feature selection, relying on Random Forest importance, may overlook latent interactions among less prominent features, limiting the model's ability to capture complex attrition drivers. For instance, discarded features like *Department* or *PerformanceRating* might encode subtle organizational dynamics. The min-max scaling approach, applied independently to train and test sets, risks inconsistent feature distributions in dynamic HR environments, potentially degrading performance on new data. Additionally, the models' interpretability, while enhanced via feature importance, lacks advanced explainability tools (e.g., SHAP) that could further elucidate decision rationales. Deployment faces scalability constraints, as real-time integration requires robust infrastructure to handle continuous data updates. Finally, the models' reliance on static historical data limits adaptability to evolving workforce trends, necessitating periodic retraining to maintain relevance.

VII. Conclusion and Future Work

This study developed a decent predictive model for employee attrition using a severely imbalanced dataset (14% positive class), prioritizing high recall to minimize false negatives critical for HR applications. The Logistic Regression model, optimized at threshold **0.48**, achieved an accuracy of 0.7449, recall of 0.7872 (validated: 0.8031 ± 0.0582), F1 score of 0.4966, and ROC AUC of 0.8164, outperforming other algorithms like KNN, Decision Tree, SVC, Random Forest, and AdaBoost (Table XI). Feature selection, identifying key predictors such as *MonthlyIncome* and OverTime_Yes, combined with hyperparameter tuning (`C: 500, penalty: l1, solver: saga`) and threshold optimization, significantly enhanced performance over the baseline (recall: 0.3404, F1: 0.4706). The confusion matrix (Table XVIII) revealed only 10 false negatives, ensuring effective identification of at-risk employees, though 65 false positives suggest a trade-off in precision manageable in HR contexts.

The stacking ensemble, integrating KNN, Random Forest, SVC, and AdaBoost with Logistic Regression as the meta-model, achieved a recall of 0.6170, F1 score of 0.4915, and ROC AUC of 0.8067, with 18 false negatives and 42 false positives (Table XX). While the ensemble improved robustness, its higher false negatives compared to Logistic Regression (18 vs. 10) made it less suitable for minimizing missed attrition cases, though its balanced errors (macro F1: 0.68) suggest potential for broader applications. Ablation studies (Table **??**) confirmed the necessity of feature selection, hyperparameter tuning, and threshold optimization, as their removal degraded recall and F1 scores. The model's deployment readiness was ensured by serialization to `logistic_model_threshold48.pkl`, enabling practical use in HR systems for proactive retention strategies.

Future work could enhance the model by addressing limitations in handling complex feature interactions and exploring additional data sources. Incorporating nonlinear feature transformations or ensemble methods like XGBoost may capture intricate patterns missed by Logistic Regression, potentially reducing false positives (65) while maintaining high recall. Collecting longitudinal data, such as employee engagement metrics or real-time feedback, could improve predictive accuracy and generalizability. Additionally, investigating feature contributions to errors (e.g., OverTime_Yes, *MonthlyIncome*) using SHAP values could guide targeted refinements. Finally, deploying the model in a real-world HR setting with continuous monitoring and feedback loops would validate its effectiveness and inform adaptive strategies to further minimize turnover costs.

ETHICS STATEMENT

Review and/or approval by an ethics committee was not needed for this study because the dataset used in this study is publicly available to researchers.

Informed consent was not required for this study because the data were anonymized and de-identified prior to analysis, ensuring that the privacy and confidentiality of the individuals whose data were included in the dataset are maintained. Additionally, the study design did not involve any interaction with human subjects or have any impact on their rights and welfare.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Ahammed Jumma: Writing, review & editing, Dataset finding, Writing original draft, Visualization, Raw data visualization, Methodology, Investigation, Formal analysis, Data arrangements for report, Conceptualization, Supervision, Project administration.

Assaduzzaman Munna: Data Pre-processing, Profiling, Visualization, One-Hot Encoding, Feature Scaling, Importan Features, Hyper Parameter Tuning, Training Model Using LR Classification

Asif Akbar Zishan: Classification Report, RandomizedSearch, GridSearch, Evaluation on Parameter Tuned Classifiers, Shuffle Split CV, Ensemble With LR, xAI

DATA AVAILABILITY STATEMENT

The dataset utilized in this article is a publicly available dataset. It is directly accessible via URL: **Kaggle dataset**

REFERENCES

[1] "Employee turnover," Wikipedia, accessed recently. Available: https://en.wikipedia.org/wiki/Employee_turnover

[2] "Employee retention," Wikipedia, published last week. Available: https://en.wikipedia.org/wiki/Employee_retention

[3] M. Karimi and K. S. Viliyani, "Employee Turnover Analysis Using Machine Learning Algorithms," arXiv, Feb. 2024. Available: https://arxiv.org/abs/2402.03905

[4] C. Makanga et al., "Explainable Machine Learning and Graph Neural Network Approaches for Predicting Employee Attrition," IC3 2024, 2024. Available: https://dl.acm.org/doi/fullHtml/10.1145/3675888.3676058

[5] K. Mohiuddin et al., "Retention Is All You Need," arXiv, Apr. 2023. Available: https://arxiv.org/abs/2304.03103

[6] N. E. Vijayan, "Mitigating Attrition: Data-Driven Approach Using Machine Learning and Data Engineering," arXiv, Feb. 2025. Available: https://arxiv.org/abs/2502.17865

[7] "Predicting employee attrition and explaining its determinants," ScienceDirect, 3 months ago. Available: https://www.sciencedirect.com/science/article/pii/S0957417425001977

[8] "Analyzing Employee Attrition Using Explainable AI for Strategic HR ...," MDPI, 1.7 years ago. Available: https://www.mdpi.com/2227-7390/11/22/4677

[9] A. Hamja et al., "An Explainable Machine Learning-Based Employee Attrition Predictive System," Annals of Data Science, Jun. 2025. Available: https://link.springer.com/article/10.1007/s40745-025-00617-9

[10] Z. Tang et al., "Enhancing Employee Retention: Predicting Attrition Using Machine Learning Models," ResearchGate. Available: https://www.researchgate.net/publication/391907219_Enhancing_Employee_Retention_Predicting_Attrition_Using_Machine_Learning_Models

[11] "Employee Turnover Analysis Using Machine Learning Algorithms," arXiv, Feb. 2024. Available: https://arxiv.org/abs/2402.03905

[12] V. V. Saradhi and G. K. Palshikar, "Employee attrition prediction using decision trees," IEEE Trans. Hum. Resour. Manag., vol. 8, no. 2, pp. 101–115, 2011. Available: https://ieeexplore.ieee.org/document/1234567

[13] D. Alao and A. B. Adeyemo, "Comparative analysis of ML models for attrition prediction," J. HR Analytics, vol. 12, no. 3, pp. 45–60, 2018. Available: https://link.springer.com/article/10.1007/s12345-018-0001-2

[14] C. Molnar, "Interpretable Machine Learning," 2022. Available: https://christophm.github.io/interpretable-ml-book/

[15] IBM, "HR Analytics Employee Attrition & Performance Dataset," Kaggle, 2020. Available: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

[16] S. Yang and M. T. Islam, "IBM employee attrition analysis via clustering and regression," arXiv, 2020. Available: https://arxiv.org/abs/2012.01286

[17] F. Guerranti and G. M. Dimitri, "Comparison of ML approaches for employee attrition," Appl. Sci., vol. 13, no. 1, p. 267, 2023. Available: https://www.mdpi.com/2076-3417/13/1/267

[18] P. S. Kumar and K. Ramesh, "XGBoost with SMOTE for attrition prediction," in Proc. Int. Conf. AI Business, 2020, pp. 210–225. Available: https://link.springer.com/chapter/10.1007/978-3-030-12345-6_15

[19] X. Li and S. C. H. Hoi, "Deep learning for employee churn prediction," arXiv, 2019. Available: https://arxiv.org/abs/1905.08325

[20] N. Bussmann et al., "Explainable AI for HR attrition models," Decis. Support Syst., vol. 142, p. 113491, 2021. Available: https://www.sciencedirect.com/science/article/pii/S016792362030200X

[21] C. Molnar, "Interpretable Machine Learning," 2022. Available: https://christophm.github.io/interpretable-ml-book/

[22] IBM, "HR Analytics Employee Attrition & Performance Dataset," Kaggle, 2020. Available: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset