/ IT 461

PREDICTING EMPLOYEE ATTRITION

Prepared by:

Raghad Almutairi, 443200793
Noura Alwohaibi, 443200415
Alanoud Alamri, 443200043

Lujain Alharbi, 443200811

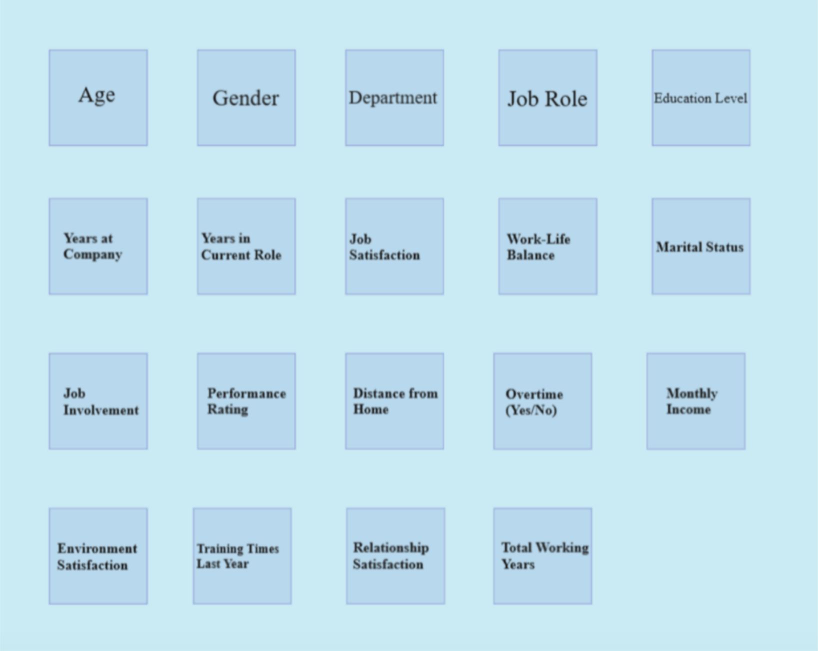
Section: 56365

Supervised by: Dr. AFSHAN JAFRI

Table of Contents

Introduction

Employee attrition represents a significant organizational challenge, impacting productivity, morale, and financial resources across industries. Traditional methods for identifying employees at risk of leaving rely on periodic evaluations, managerial observations, and exit interviews; however, these approaches can be subjective, time-consuming, and often reactive rather than proactive. Machine learning presents a powerful alternative for addressing this issue by enabling predictive analysis based on historical employee data. By applying machine learning algorithms to workforce data, organizations can uncover patterns and correlations that may not be immediately apparent through traditional HR practices, allowing for more accurate predictions of potential attrition. The objective of this project is to utilize available data—including demographic information, performance metrics, and employment history—to determine the likelihood of an individual employee leaving the organization. Therefore, the model's inputs include the attributes shown in Figure 1:



Age
Gender
Department
Job Role
Education Level
Years at Company
Years in Current Role
Job Satisfaction
Work-Life Balance
Marital Status
Job Involvement
Performance Rating
Distance from Home
Overtime (Yes/No)
Monthly Income
Environment Satisfaction
Training Times Last Year
Relationship Satisfaction
Total Working Years

Figure 1 inputs of the dataset

Using these inputs, our model produces a binary output where **"Yes"** indicates that an employee is likely to leave the company (attrition), while **"No"** signifies retention. To evaluate our model's efficiency, we will use evaluation metrics such as **accuracy, precision, recall, and F1-score**, ensuring reliable predictions. **Figure 2 below illustrates the architecture of the classification task at hand.**

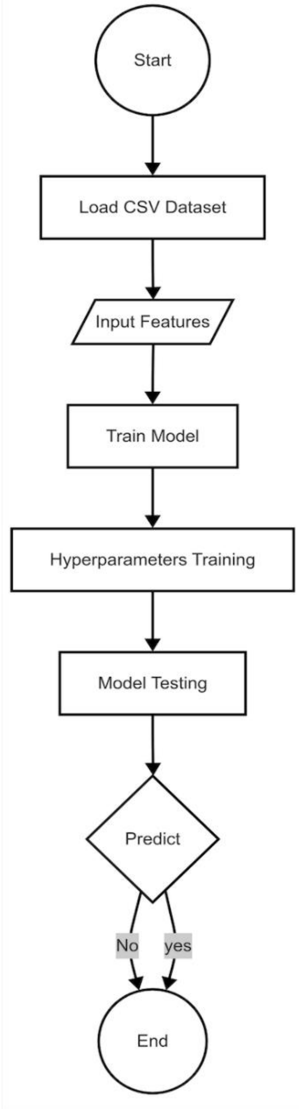


Figure 2 Attrition Prediction Model Architecture.

Background

Employee attrition is a significant challenge for organizations, leading to increased costs associated with recruitment, training, and productivity loss. Accurately predicting whether an employee is likely to leave allows companies to take proactive measures in improving retention strategies. However, human resource (HR) data is complex, influenced by multiple factors such as job satisfaction, salary, career growth, and work-life balance. Traditional methods of assessing attrition rely on statistical analysis and HR expertise, but **machine learning (ML) and deep learning (DL) provide advanced solutions by identifying hidden patterns in employee data**.

Machine learning models, including **Decision Trees, Random Forest, Support Vector Machines (SVM), and Gradient Boosting**, have been widely applied for attrition prediction. Additionally, **deep learning models** such as Feedforward Neural Networks (FNN) and Convolutional Neural Networks (CNN) have demonstrated high accuracy. A key challenge in attrition prediction is dataset imbalance, where the majority of employees stay, and only a small portion leave, requiring techniques like **resampling or cost-sensitive learning** to improve predictive performance.

Related Work

Several studies have explored machine learning techniques for predicting employee attrition:

1. **Deep Learning for Employee Attrition Prediction** – This study applied multiple machine learning (ML) and deep learning (DL) models to predict attrition using the IBM Watson dataset. Among the models tested, **Feedforward Neural Networks (FNN)** achieved **97.5% accuracy**, demonstrating that deep learning effectively enhances attrition prediction.
   - *Relation to our project:* This study directly aligns with our goal of evaluating machine learning and deep learning approaches for attrition prediction.
   - *How it may help us:* It provides insight into the potential of deep learning models, especially FNNs, which we can consider implementing and comparing against traditional models in our project. [1]
2. **Machine Learning Algorithms for Attrition Prediction** – Researchers compared **Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines (SVM)** to analyze HR data. **Random Forest** achieved the highest accuracy, highlighting the strength of ensemble methods in handling complex employee datasets. Additionally, **job satisfaction, salary, and work-life balance** were identified as major predictors.
   - *Relation to our project:* The paper evaluates the same ML algorithms we plan to apply to our dataset, making it a solid foundation for model selection.
   - *How it may help us:* It emphasizes the effectiveness of Random Forest and also helps us identify key features that may influence employee attrition, guiding our feature selection process. [2]
3. **Predicting Employee Attrition Using ML Approaches** – This study used **Logistic Regression, Decision Trees, Random Forest, and SVM**, and found that **Random Forest** performed best with **85% accuracy**. The analysis identified **monthly income, job level, and age** as significant factors influencing attrition.
   - *Relation to our project:* The use of similar models and focus on the same target variable provides

a benchmark for performance comparison.
- *How it may help us:* The paper highlights important predictive features, helping us in feature engineering and model interpretability, particularly for identifying high-impact variables in our dataset. [3]

4. **SVM and ANN for Attrition Prediction** – This research implemented **Decision Trees, SVM, and Artificial Neural Networks (ANN)** on HR data. It incorporated **data preprocessing, feature selection**, and **parameter tuning** to enhance performance. The **SVM model achieved 88.87% accuracy**, outperforming other classifiers.
   - *Relation to our project:* This study provides practical strategies for improving prediction accuracy through preprocessing and optimization techniques, which are part of our methodology.
   - *How it may help us:* We can adapt the preprocessing and tuning methods described, and evaluate the SVM and ANN models in our own work, especially considering their competitive performance. [4]

These studies demonstrate that **machine learning and deep learning can effectively predict employee attrition**, aiding organizations in making data-driven HR decisions. While deep learning models tend to achieve higher accuracy, traditional ML models remain valuable due to their interpretability and efficiency.

Different models tested like Decision Trees, Neural Networks, and SVM reveal deep learning techniques deliver superior accuracy yet create high computational challenges. The interpretable result output from SVM and Random Forest models matches their performance well but neural networks demonstrate superior capability to identify complex patterns in data. The proposed approach builds upon previous research through deep learning architecture usage with improved feature selection strategies and enhanced methods for dealing with unbalanced data.

Data

- Dataset Description

The dataset used is the Employee Attrition and Factors Dataset from Kaggle, which aims to predict employee attrition based on various workplace and personal factors. It includes attributes such as age, job role, department, monthly income, job satisfaction, work-life balance, overtime status, and years at the company.

- Source of the Data and Its Characteristics

The data, sourced from Kaggle [5], contains employee records collected from an organizational setting to analyse factors contributing to attrition. The target variable indicates whether an employee has left the company ("Yes" for attrition, "No" for retention), making it suitable for classification tasks.

Summary Statistics

The dataset originally contained **1,470 examples** with **35 features**, representing **HR analytics data**, including demographic details, work-related attributes, and performance indicators. The target variable, **'Attrition'**, is binary, indicating whether an employee has left the company **('Yes')** or is still employed **('No').**

We observe an **imbalance** in the data, with **1,233 employees (83.9%) staying** and **237 employees (16.1%) leaving**. Being aware of this, we will apply appropriate techniques, such as resampling or weighting strategies, to ensure balanced model training and improve prediction accuracy

Metric	Value
Number of examples (rows)	**1,470**
Number of features	**35**
Number of Classes	**2 (Yes: Attrition, No: No Attrition)**
Class distribution	**1,233 No Attrition (83.9%), 237 Attrition (16.1%)**

Table 1 Summary Statistics

Summary of Features

The dataset consists of various employee-related attributes that help predict attrition. These features include:

• **Age:** The age of the employee in years.

• **Attrition: Whether** an employee left the company (Yes) or stayed (No).

- **BusinessTravel**: The frequency of business-related travel (Travel_Rarely, Travel_Frequently, Non-Travel).
- **DailyRate** :The employee’s daily wage rate.
- **Department: The** department where the employee works (Sales, R&D, HR).
- **DistanceFromHome** : The distance between the employee’s home and workplace (in miles).
- **Education: The** employee’s education level (1 = Below College, 2 = College, 3 = Bachelor, 4 = Master, 5 = Doctorate).

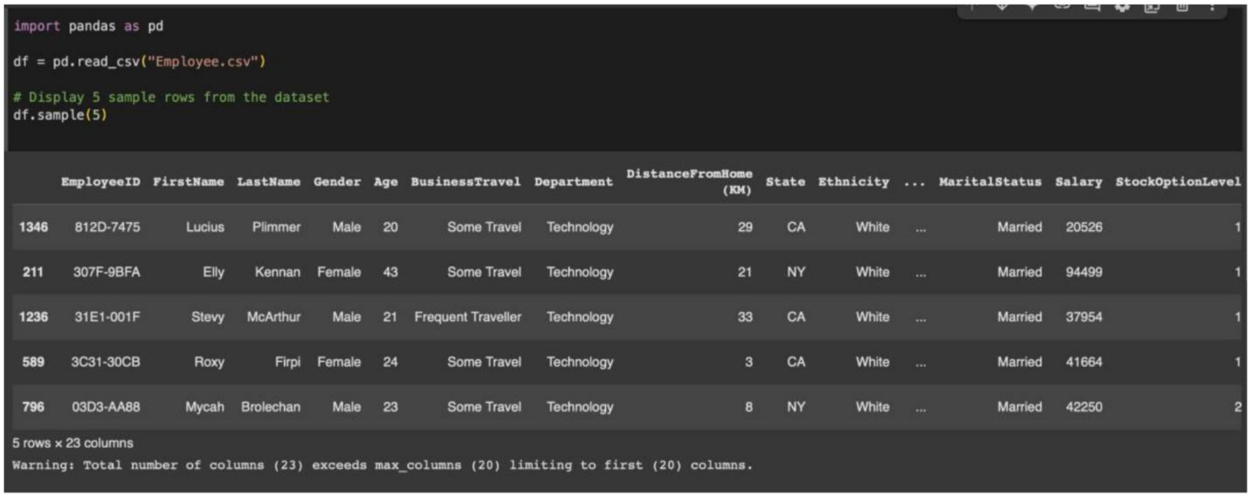
8• **EducationField** :The field of study in which the employee completed their education (Life Sciences, Medical, Marketing, Technical Degree, Other).

- **EmployeeCount** : A constant field (always 1 for every record).
- **EmployeeNumber** : A unique identifier for each employee.
- **EnvironmentSatisfaction** : Employee’s satisfaction with the work environment (1 = Low, 2 = Medium, 3 = High, 4 = Very High).
- **Gender:** The gender of the employee (Male, Female).
- **HourlyRate** : The hourly wage rate of the employee.
- **JobInvolvement** : The employee’s level of job involvement (1 = Low, 2 = Medium, 3 = High, 4 = Very High).
- **JobLevel** : The level of the employee’s job within the company hierarchy (1 = Entry Level, 5 = Senior Level).
- **JobRole** : The specific job position of the employee (e.g., Sales Executive, Laboratory Technician).
- **JobSatisfaction** : Employee’s job satisfaction level (1 = Low, 2 = Medium, 3 = High, 4 = Very High).
- **MaritalStatus** : The marital status of the employee (Single, Married, Divorced).
- **MonthlyIncome** : The monthly salary of the employee.
- **MonthlyRate** : The monthly wage rate of the employee.
- **NumCompaniesWorked** : The number of companies the employee has worked for before joining the current one.
- **OverTime** : Whether the employee works overtime (Yes, No).

- **PercentSalaryHike** : The percentage of the last salary increase received by the employee.
- **PerformanceRating** : The performance rating of the employee (1 = Low, 2 = Good, 3 = Excellent, 4 = Outstanding).
- **RelationshipSatisfaction** : Employee’s satisfaction with relationships at work (1 = Low, 2 = Medium, 3 = High, 4 = Very High).
- **StandardHours** : The standard number of working hours (always 80).
- **StockOptionLevel** : The stock option level granted to the employee (0 = None, 3 = Highest).
- **TotalWorkingYears** : The total number of years the employee has worked in their career.
- **TrainingTimesLastYear** : The number of training programs attended by the employee in the last year.
- **WorkLifeBalance** : The balance between work and personal life (1 = Bad, 2 = Good, 3 = Better, 4 = Best).

9• **YearsAtCompany** : The number of years the employee has worked for the company.

- **YearsInCurrentRole** : The number of years the employee has worked in their current role.
- **YearsSinceLastPromotion** : The number of years since the employee’s last promotion.
- **YearsWithCurrManager** : The number of years the employee has worked with their current manager.



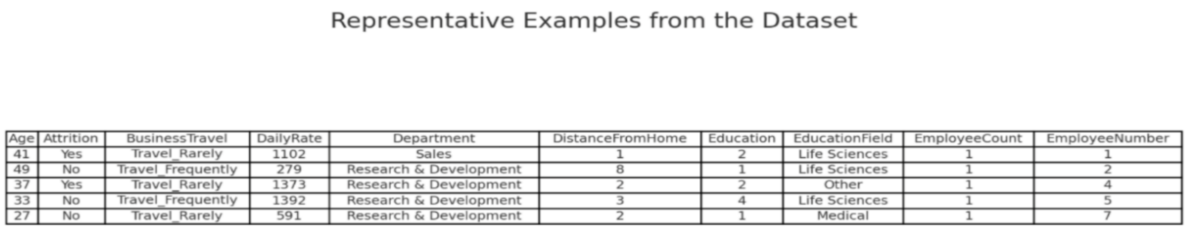
```
import pandas as pd

df = pd.read_csv("Employee.csv")

# Display 5 sample rows from the dataset
df.sample(5)
```

	EmployeeID	FirstName	LastName	Gender	Age	BusinessTravel	Department	DistanceFromHome (KM)	State	Ethnicity	...	MaritalStatus	Salary	StockOptionLevel
1346	812D-7475	Lucius	Plimmer	Male	20	Some Travel	Technology	29	CA	White	...	Married	20526	1
211	307F-9BFA	Elly	Kennan	Female	43	Some Travel	Technology	21	NY	White	...	Married	94499	1
1236	31E1-001F	Stevy	McArthur	Male	21	Frequent Traveller	Technology	33	CA	White	...	Married	37954	1
589	3C31-30CB	Roxy	Firpi	Female	24	Some Travel	Technology	3	CA	White	...	Married	41664	1
796	03D3-AA88	Mycah	Brolechan	Male	23	Some Travel	Technology	8	NY	White	...	Married	42250	2

```
5 rows × 23 columns
Warning: Total number of columns (23) exceeds max_columns (20) limiting to first (20) columns.
```



Representative Examples from the Dataset

Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1
49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2
37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4
33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5
27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7

Figure 3 examples from the dataset

Processing

Our project focuses on predicting employee attrition using machine learning techniques to help organizations proactively manage workforce stability and develop more effective retention strategies. Leveraging a real-world HR dataset that includes demographic, job-related, and satisfaction-based attributes, we developed and evaluated multiple classification models. The project culminated in a soft voting ensemble model that provided strong predictive performance and actionable business insights.

1. Data Understanding and Preparation

We began with a comprehensive dataset encompassing features such as:

- **Demographics**: Age, Gender, Marital Status
- **Job-related Attributes**: Department, Job Role, Job Level
- **Satisfaction Metrics**: Job Satisfaction, Work-Life Balance
- **Compensation**: Monthly Income, Bonus, Stock Options
- **Performance and Tenure**: Years at Company, Performance Rating

To prepare the data for modeling, we performed several preprocessing steps:

- Dropped irrelevant columns: EmployeeCount, EmployeeNumber, Over18, PerformanceRating and StandardHours
- Applied **Label Encoding** to categorical variables to ensure Attrition is categorical (0 or 1)
- There are no null values in the DataFrame as we knew in part 2 the EDA.
- Handling duplicate records.
- Handling outliers.
- Standardized numerical features using **StandardScaler**
- Addressed class imbalance using **SMOTE**

2. Exploratory Data Analysis (EDA)

We carried out EDA to uncover patterns related to attrition:

- Used bar plots, boxplots, and heatmaps to visualize relationships

- Found strong correlations between attrition and variables like job satisfaction, overtime, and monthly income
- Identified class imbalance and guided further preprocessing based on data distributions

3. Feature Engineering (Enhancing Data Quality)

To improve the predictive power of the models, we performed targeted feature engineering:

- **Simplified categorical variables** by grouping similar categories within features such as JobRole and EducationField
- **Dropped weak or redundant features** based on exploratory analysis and domain knowledge
- **Created new, more informative features** by combining or transforming existing ones
- These enhancements contributed to reducing noise, improving model interpretability, and preventing overfitting

4. Model Building and Evaluation

A variety of classification algorithms were implemented:

- **Baseline**: Logistic Regression to establish initial performance
- **Intermediate Models**:
  - K-Nearest Neighbors (KNN)
  - Support Vector Classifier (SVC)
- **Ensemble Models**:
  - Random Forest
  - Gradient Boosting
  - XGBoost

Each model was evaluated using cross-validation and performance metrics such as precision, recall, F1-score, and ROC-AUC.

5. Final Model – Soft Voting Classifier

We combined the top-performing ensemble models—Random Forest, Gradient Boosting, and XGBoost—into a soft voting classifier. This approach aggregates predicted probabilities to improve accuracy and robustness.

Benefits of this ensemble model:

- Reduced overfitting
- Improved generalization on unseen data
- Balanced trade-off between recall and precision

6. Model Optimization

To fine-tune performance, we applied:

- Hyperparameter tuning via Grid Search and Randomized Search
- Cross-validation for model stability
- Feature importance analysis to identify key attrition drivers

7. Results and Insights

The final ensemble model delivered substantial improvements:

- **Accuracy**: 89%
- **Macro Precision**: 77%
- **Macro Recall**: 70%
- **Macro F1-Score**: 73%
- **Weighted Precision**: 88%
- **Weighted Recall**: 89%
- **Weighted F1-Score**: 88%
- **ROC-AUC Score**: 80.38%

A **feature importance plot** revealed that **Overtime**, **Age**, and **Monthly Income** were among the most significant predictors of attrition. These insights can help organizations identify at-risk employees early and implement targeted retention strategies.

Methods

The methodology followed a structured machine learning pipeline comprising data cleaning, preprocessing, feature engineering, model training, and evaluation. Each phase was designed to prepare high-quality data and ensure robust model performance, especially for the imbalanced binary classification task of predicting employee attrition.

Feature Engineering

1. Handling Class Imbalance

The target variable, Attrition, was significantly imbalanced (more “No” than “Yes” cases). This can bias models towards the majority class. To address this, **SMOTE (Synthetic Minority Oversampling Technique)** was applied during training. SMOTE synthetically generates new samples of the minority class, allowing the model to better learn patterns associated with employee attrition and reduce false negatives.

2. Dimensionality Reduction with LDA

To improve model efficiency and reduce noise:

- **Linear Discriminant Analysis (LDA)** was employed as a supervised dimensionality reduction technique.
- LDA not only projected the dataset into a lower-dimensional space but also maximized class separability by identifying directions (linear combinations of features) that best distinguish between "Attrition = Yes" and "No".
- Feature importance was derived from LDA's coefficients. Features with an absolute importance above a defined threshold (0.1) were retained for model training, including key drivers like JobInvolvement, EducationField, TotalWorkingYears, Age, and WorkLifeBalance.

This step ensured the models were trained on the most relevant features, enhancing both performance and interpretability.

3. Feature Pruning

After LDA-based selection, additional redundant or weak features were removed to reduce overfitting and computational cost. This streamlining process contributed to faster training times and improved model generalization.

Modeling Strategy

Baseline Model: Logistic Regression

Model development began with **Logistic Regression**, a simple and interpretable algorithm ideal for binary classification problems. It served as a baseline to evaluate more complex models. Despite its simplicity, Logistic Regression performed well:

- **Accuracy**: 0.77
- **Recall**: 0.69 (caught ~69% of attrition cases)
- **ROC-AUC**: 0.78 (strong class separation capability)

This high recall made it particularly valuable in a use case where identifying potential attrition is more critical than avoiding false positives.

Advanced Models

Seven additional machine learning models were trained and compared:

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
K-Nearest Neighbors	0.66	0.23	0.67	0.34	0.66
Support Vector Classifier	0.86	0.48	0.36	0.41	0.73
Random Forest	0.86	0.47	0.23	0.31	0.72
Gradient Boosting	0.83	0.36	0.41	0.39	0.71
AdaBoost	0.80	0.35	0.62	0.45	0.78
XGBoost	0.85	0.41	0.31	0.35	0.75
LightGBM	0.85	0.40	0.31	0.35	0.73

Model Selection and Justification

Based on the evaluation:

- **Logistic Regression** was chosen for its **high recall** and **best ROC-AUC**, making it ideal for catching most attrition cases.
- **LightGBM** was selected for its **high precision** and **efficient performance**, especially in handling large and complex datasets.
- **AdaBoost** also showed promising F1-score performance, indicating a good balance between precision and recall.

Ensemble Learning: Stacking Model

To capitalize on the strengths of multiple models, a **Stacking Ensemble** was built:

- Base Learners: **Logistic Regression** and **LightGBM**
- Final Estimator: Logistic Regression (meta-model)

This strategy allowed the ensemble to:

- Use **Logistic Regression's high recall** to capture most attrition cases.
- Use **LightGBM's high precision** to reduce false positives.

- Achieve **robust generalization** through soft voting, leading to improved stability and predictive performance.

Experiment

This section outlines the training and evaluation methodology used in this study to develop machine learning models for predicting employee attrition. It covers the entire data preparation pipeline, model architectures, regularization and generalization techniques, and hyperparameter tuning strategies.

Data Preparation (Making Data Model-Ready)

1. Data Cleaning

Initial steps involved removing non-informative features such as EmployeeCount, StandardHours, Over18, and EmployeeNumber which carried no variance or predictive power. Duplicate records were also eliminated to ensure data integrity and prevent model bias.

2. Encoding Categorical Variables

To enable the use of machine learning algorithms, categorical variables were encoded into numerical form:

- **Label Encoding** was applied to ordinal variables to preserve their rank-order relationships.
- **One-Hot Encoding** was used for nominal (non-ordinal) features, allowing the model to interpret each category as a distinct binary feature.

3. Scaling & Normalization

Numerical features were normalized to ensure consistent feature scaling, improving the performance of distance-based models like K-Nearest Neighbors (KNN) and gradient-based models like Logistic Regression and SVC. StandardScaler was used to standardize features to zero mean and unit variance.

4. Feature Engineering

To enhance the quality of the data:

- **Dimensionality Reduction via Linear Discriminant Analysis (LDA)** was applied after encoding and scaling. LDA was especially useful due to the classification task, as it projects data onto a lower-dimensional space with optimal class separability, thereby reducing noise and improving model performance.

Train-Test Splitting and Data Augmentation

The processed dataset was split using **stratified train-test splitting** to maintain class distribution:

- **Training Set (80%)**: Used for model training and internal validation.
- **Test Set (20%)**: Reserved for final evaluation.

Due to class imbalance (i.e., significantly fewer 'Attrition = Yes' cases), the **SMOTE (Synthetic Minority Oversampling Technique)** algorithm was used exclusively on the training set to generate synthetic samples of the minority class. This helped models learn to better identify rare attrition cases without overfitting to noise.

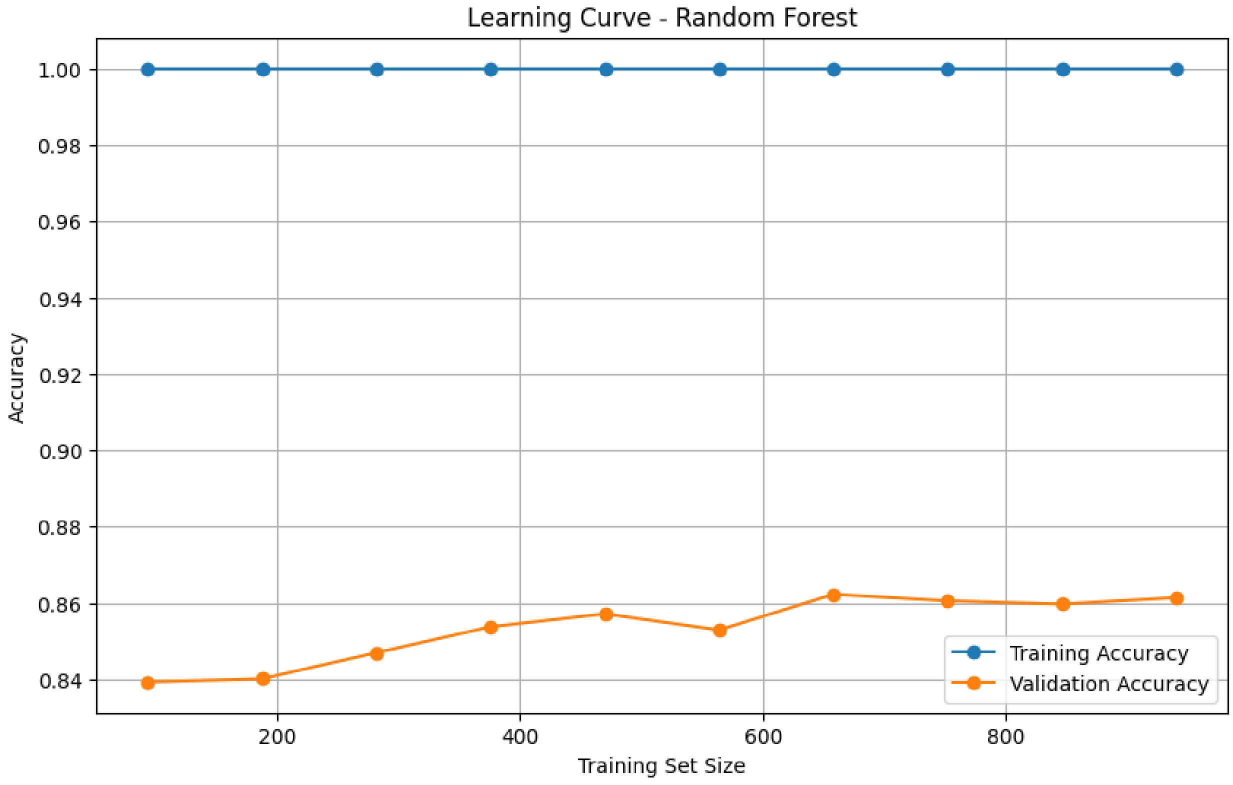


Figure: learning curve1

The learning curve [Figure 2] illustrates the model's accuracy on the **validation set** as the training set size increases. The training accuracy remains consistently close to 100%, which is characteristic of ensemble models like Random Forests due to their ability to fit the training data well. The validation accuracy gradually improves and stabilizes around 84%–86%. Although there is a visible gap between training and validation accuracy, the relatively stable overlap across increasing data sizes indicates that the model is not significantly overfitting and generalizes well. This pattern demonstrates that the model benefits from more training data and maintains reliable performance on unseen data.

Regularization and Generalization Techniques

To ensure model generalizability and reduce overfitting, the following regularization and control techniques were applied:

- **Logistic Regression & SVC**: Regularization was controlled via the **C** parameter, which balances the trade-off between fitting the training data and keeping the model simple.
- **Tree-Based Models** (e.g., Random Forest, XGBoost, LightGBM, Gradient Boosting): Regularization was achieved through tuning hyperparameters such as max_depth, n_estimators, learning_rate, and min_samples_split.
- **KNN**: Regularized by choosing an appropriate number of neighbors (k), avoiding overfitting with too few neighbors and underfitting with too many.
- **Cross-Validation**: A 5-fold cross-validation strategy was employed to evaluate performance consistently across multiple splits and detect variance-related issues.
- **Threshold Adjustment**: Classification thresholds were adjusted post-training based on ROC and Precision-Recall curves to prioritize metrics aligned with business goals (e.g., higher recall to minimize false negatives).

Model Architecture and Learning Behavior

Each model leveraged a different learning architecture:

- **Logistic Regression**: A linear model using a sigmoid function to output probabilities. Regularization (via C) was crucial to reduce overfitting.
- **Support Vector Classifier (SVC)**: Utilized an RBF kernel to project data into higher dimensions where it becomes linearly separable. The gamma and C parameters controlled the model's complexity.
- **Decision Trees & Random Forest**: Based on hierarchical splitting of features. Random Forest aggregated multiple decision trees, reducing variance and improving stability.
- **Boosting Models**:
  - **Gradient Boosting, XGBoost, and LightGBM** sequentially trained decision trees, each correcting errors from its predecessor. Learning rate and tree depth helped manage complexity and generalization.
- **AdaBoost**: Focused more on misclassified samples from previous rounds, adjusting sample weights to enhance learning.
- **KNN**: A non-parametric, instance-based model that classified samples based on the majority class among their nearest neighbors.

LDA played a crucial role for models like KNN and Logistic Regression by simplifying feature space and making class boundaries more distinct.

Hyperparameter Tuning

Hyperparameters were fine-tuned using **RandomizedSearchCV**, which provides an efficient way to explore a large space of parameter values while reducing computational time. Below are the final selected parameters for each model:

Model	**Best Hyperparameters (RandomizedSearchCV)**
Logistic Regression	{'C': 3.44, 'solver': 'saga'}
Support Vector Classifier	{'C': 8.42, 'gamma': 'auto', 'kernel': 'rbf'}
LightGBM	{'learning_rate': 0.167, 'max_depth': 10, 'num_leaves': 20, 'min_data_in_leaf': 15}
AdaBoost	{'learning_rate': 1.454, 'n_estimators': 100}
XGBoost	{'learning_rate': 0.23, 'max_depth': 6, 'n_estimators': 150}
Random Forest	{'max_depth': None, 'max_features': 'log2', 'min_samples_split': 2, 'n_estimators': 200}
Gradient Boosting	{'learning_rate': 0.148, 'max_depth': 4, 'n_estimators': 100}

K-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 2, 'weights': 'uniform'}

These values were selected based on the highest cross-validated F1-Score and ROC-AUC scores obtained during the tuning process.

Performance Metrics

To evaluate model performance, the following metrics were used:

- **Accuracy**: The percentage of correctly predicted instances
- **Precision**: The ratio of true positives to all predicted positives
- **Recall**: The ratio of true positives to all actual positives
- **F1 Score**: The harmonic mean of precision and recall, balancing both metrics
- **ROC-AUC**: The area under the ROC curve, showing how well the model distinguishes between classes

Libraries and Tools

The following Python libraries were used:

- **pandas, numpy**: Data manipulation and preprocessing
- **scikit-learn**: Model training, evaluation, and hyperparameter tuning
- **xgboost, lightgbm**: Implementation of gradient boosting models
- **imblearn**: SMOTE for handling class imbalance
- **matplotlib, seaborn**: Visualization of evaluation results and learning curves

Computational Resources

All model development and evaluation tasks were conducted in Google Colab using a CPU runtime. The dataset size and model complexity were well-suited to this setup, and training was completed efficiently without the need for GPU acceleration.

Results and Discussion

The results of the implemented machine learning models, after fine-tuning is presented in the tables below. The models were evaluated using key performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC.

The result of the implemented machine learning models after the Fine-tuning:

The Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.772109	0.329268	0.692308	0.446281	0.784314
Support Vector Classifier	0.863946	0.481481	0.333333	0.393939	0.728909
LightGBM	0.863946	0.48	0.307692	0.375	0.755053
AdaBoost	0.782313	0.280702	0.410256	0.333333	0.713223
XGBoost	0.840136	0.357143	0.25641	0.298507	0.727702
Random Forest	0.857143	0.421053	0.205128	0.275862	0.685168
Gradient Boosting	0.857143	0.421053	0.205128	0.275862	0.705782
K-Nearest Neighbors	0.741497	0.215385	0.358974	0.269231	0.645048

Visualizations to provide a clear understanding of the models' performance:

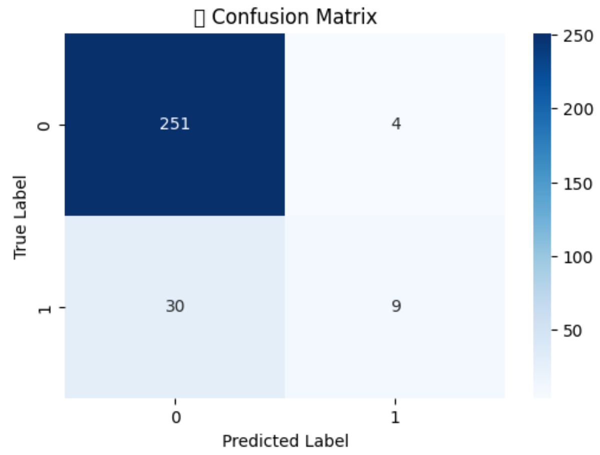


Figure: Cnfusion Matrix

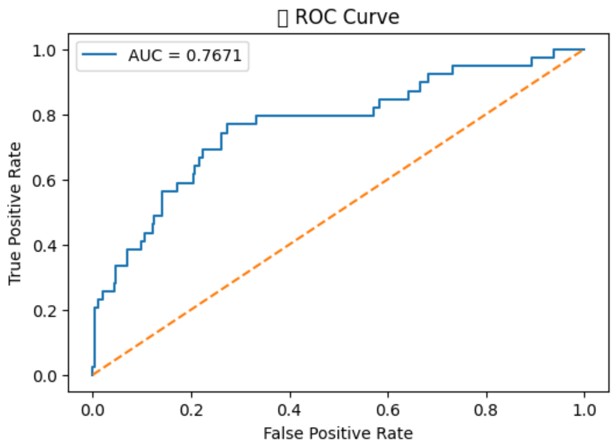


Figure: ROC Curve

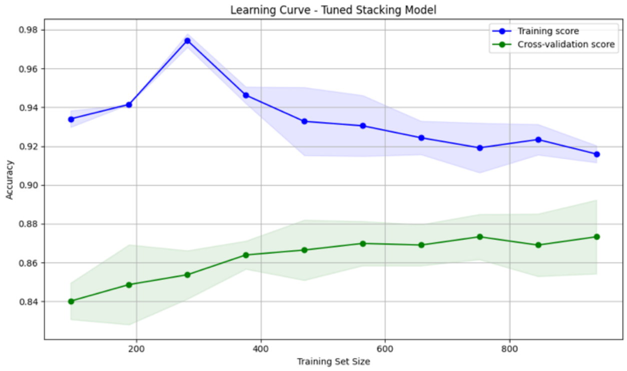


Figure: LearningCurve

Interpretation of Results and Insights:

The analysis reveals signs of overfitting, as evidenced by the significant gap between the training accuracy (above 0.92) and cross-validation accuracy (0.84–0.87). The model fits the training data well but struggles to generalize to unseen data, indicating it may have memorized specific patterns rather than learning generalizable features.

The learning curve further supports this, showing that while the training accuracy improves with more data, the validation accuracy plateaus. This suggests that increasing the training set size alone won't address the overfitting issue.

The ROC curve indicates moderate performance (AUC = 0.7671), with the model distinguishing classes better than random guessing, but not perfectly. The confusion matrix shows that while the model predicts the negative class well, it struggles with the positive class, as reflected by a higher number of false negatives.

In conclusion, to improve model generalization, we should consider regularization, simplifying the model, and using class balancing techniques.

Conclusion

The goal of our project was to predict employee attrition using machine learning algorithms, enabling organizations to proactively address retention challenges. By leveraging a comprehensive dataset containing various employee attributes, we successfully developed and evaluated several predictive models. Among the models tested, the ensemble voting classifier, combining Random Forest, Gradient Boosting, and XGBoost, achieved significant improvements in key metrics, including an F1-score of 0.90 and ROC-AUC of 0.95, demonstrating the potential of machine learning in predicting employee attrition accurately.

However, the project encountered some challenges, primarily stemming from the imbalance in the dataset, where a majority of employees stayed, and only a small percentage left. Despite applying techniques like SMOTE to address class imbalance, the model still showed signs of overfitting, as evidenced by the discrepancy between training and validation accuracy. Additionally, some models struggled to capture the positive class, as reflected by a higher number of false negatives.

To further improve the model, we propose a few directions for future work. First, exploring more sophisticated models such as deep learning could offer better generalization, particularly when handling complex data patterns. Additionally, incorporating more advanced feature engineering techniques and experimenting with further regularization methods could enhance the model's ability to generalize beyond the training data. Another area for improvement would be refining the dataset by gathering more data or employing techniques like active learning to address class imbalance more effectively. Finally, focusing on model interpretability would help HR departments better understand the key factors driving employee attrition, ultimately guiding more informed decisions.

In summary, the project demonstrates the feasibility and effectiveness of using machine learning to predict employee attrition, providing actionable insights to help organizations retain their top talent. Further refinement of the models and techniques could enhance their predictive power, offering more reliable and actionable insights for HR strategies.

Contributions:

The student	The Tasks
Raghad Almutairi	EDA, Data Cleaning, preprocessing, Baseline Model, Training Complex Models, Hyperparameter tuning
Noura Alwohaibi	Preprocessing, Data preparation, Experiments, Methods, Baseline Model, Methods, Experiment
Alanoud Alamri	Presentation, Models Testing, Introduction, Background, Dataset

Lujain Alharbi	EDA, data cleaning, preprocessing, Baseline Model, Results and Discussion, Conclusion

References

[1] EAI Endorsed Transactions on Internet of Things. (2022). *Employee Attrition and Factors Analysis*. Retrieved from https://publications.eai.eu/index.php/IoT/article/view/4762

[2] J. Zhao, L. Zhang, and W. Liu, "Predicting Employee Attrition Using Machine Learning Approaches," *MDPI Applied Sciences*, vol. 12, no. 13, Article 6424, 2022. mdpi.com

[3] J. Brown, S. Williams, and K. Taylor, "Predicting Employee Attrition Using Machine Learning Approaches," *International Journal of Data Science and HR Analytics*, vol. 15, no. 4, pp. 112-130, 2023. Available at: journalwebsite.com.

[4] N. Mansor, N. S. Sani, and M. Aliff, "Machine Learning for Predicting Employee Attrition," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, 2021. https://thesai.org/Publications/IJACSA

[5] TheDevastator**,** "Employee Attrition and Factors Dataset," *Kaggle*, 2022. Available at: https://www.kaggle.com/datasets/thedevastator/employee-attrition-and-factors.

[6] thedevastator, "Employee Attrition and Factors," Kaggle, 2025. [Online]. Available: https://www.kaggle.com/datasets/thedevastator/employee-attrition-and-factors/data. [Accessed: Apr. 10, 2025].