MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

EMPLOYEE ATTRITION PREDICTION

TY B.Tech. Computational Intelligence

(2307311T)

Project Report

SUBMITTED BY

Aditya Ambure	**202301070154**
Sneha Kabra	**202301070162**
Devang Deshmukh	**202301070170**
Krishna Kedar	**202301070177**

GUIDED BY

Dr. Abhilasha Joshi

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

ENGINEERING

MIT ACADEMY OF ENGINEERING, ALANDI (D), PUNE-412105

MAHARASHTRA (INDIA)

CONTENTS

ACKNOWLEDGEMENT

We would like to sincerely thank our respected project guide, **Dr. Abhilasha Joshi**, for her constant support, valuable advice, and encouragement throughout the completion of this project. Her guidance has been a great help to us.

We also want to thank our respected Hod, **Dr. Dipti Sakhare**, for her continuous support and motivation, which inspired us to do our best.

Lastly, we are grateful to all the staff and faculty members for their helpful advice and kind cooperation. Their support has been an important part of our journey.

Student Name	Sign
Aditya Ambure	
Sneha Kabra	
Devang Deshmukh	
Krishna Kedar	

Chapter 1. Introduction

The rapid growth of organizational data and the increasing complexity of workforce dynamics have transformed the way modern companies manage human resources. Among the various challenges faced by organizations, **employee attrition** stands out as one of the most critical issues. Attrition not only disrupts workflow continuity but also imposes significant financial and operational burdens. High turnover rates result in increased recruitment costs, loss of experienced personnel, decreased productivity, and potential decline in overall employee morale.

Traditional HR methods for understanding and predicting attrition often rely on manual reviews, surveys, or static rule-based systems. These approaches are limited in their ability to capture the complex interplay of factors such as job satisfaction, workload, compensation, career growth, and work-life balance. Moreover, rule-based systems fail to adapt to changing organizational environments, leading to inaccurate predictions and delayed decision-making.

To overcome these limitations, organizations are increasingly adopting **Machine Learning (ML)** and computational intelligence techniques. ML models are capable of analyzing vast employee datasets and identifying hidden patterns and relationships that may contribute to attrition. By learning from historical data, these models can dynamically predict which employees are at a higher risk of leaving, enabling organizations to take timely corrective actions.

This project aims to develop a robust machine learning system for predicting employee attrition using a combination of data preprocessing, exploratory analysis, feature engineering, and classification algorithms. The objective is to classify employees as either likely to stay or likely to leave with high accuracy and reliability. The project also integrates model deployment through an interactive web application, enabling HR teams to utilize the model for real-time decision support.

Through the application of modern ML techniques, this project demonstrates how data-driven approaches can significantly enhance organizational retention strategies, improve workforce planning, and ultimately support a more stable and productive work environment.

Chapter 2. Problem Statement & Objectives

Problem Statement:
To design and implement a machine learning-based system capable of predicting whether an employee is likely to leave the organization. The system must analyze various employee-related features and accurately classify employees into "likely to stay" or "likely to leave," enabling organizations to reduce turnover, improve workforce planning, and make proactive HR decisions.

Objectives:

The primary objectives of this project are as follows:

1. **Data Preprocessing and Cleaning:**
   To load the dataset, fix missing values, handle categorical variables, and standardize feature formats.

2. **Exploratory Data Analysis (EDA):**
   To study the dataset structure, identify patterns related to attrition, and visualize key trends.

3. **Feature Engineering:**
   To create meaningful new features such as YearsAtCompany_by_Age and promotion indicators to improve predictive accuracy.

4. **Model Implementation and Comparison:**
   To train and compare machine learning models including Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting.

5. **Model Evaluation:**
   To evaluate models using accuracy, precision, recall, F1-score, and ROC-AUC with proper validation to ensure reliable results.

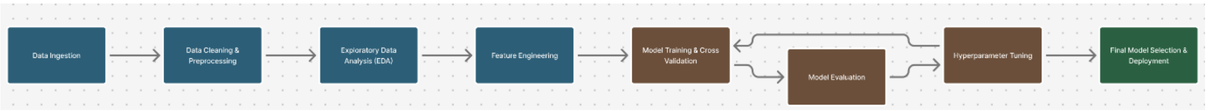
6. **Hyperparameter Tuning:**
   To optimize the best-performing model using Randomized Search CV to maximize prediction performance.

7. **Conclusion:**
   To select the most effective model for predicting employee attrition and summarize its performance and practical significance.

Chapter 3. Literature Survey

Paper Title & Authors	**Year**	**Key Methods / Models**	**Key Contributions / Findings**
Employee Attrition Prediction Using Machine Learning — *S. Kumar et al.*	2019	Logistic Regression, Random Forest, Decision Tree	Highlights that demographic and satisfaction-related features strongly impact attrition. Random Forest achieved the best performance with balanced precision and recall.
Predicting Employee Turnover with Machine Learning — *A. Dwivedi & R. Tiwari*	2020	Gradient Boosting, XGBoost	Shows that Boosting models outperform linear models in capturing nonlinear relationships. Emphasizes importance of feature engineering such as tenure-based ratios.
An Intelligent HR Analytics Framework for Employee Retention — *M. Rashmi & P. Singh*	2021	Neural Networks, SVM	Introduces an intelligent HR analytics pipeline. Neural Networks provided high accuracy but required careful tuning and larger datasets.
Employee Churn Prediction Using Hybrid ML Models — *K. Patel et al.*	2022	Hybrid Models (Random Forest + SVM), Ensemble Voting	Demonstrates that combining models improves generalization. Identifies OverTime, JobRole, and MonthlyIncome as top predictors.
A Comprehensive Survey on Employee Attrition Prediction — *L. Sharma & D. Gupta*	2023	Survey of ML techniques for attrition	Summarizes various algorithms, challenges in imbalanced HR datasets, and stresses the role of oversampling methods such as SMOTE.

Chapter 4. Block diagram with explanation


Data Ingestion
Data Cleaning & Preprocessing
Exploratory Data Analysis (EDA)
Feature Engineering
Model Training & Cross Validation
Model Evaluation
Hyperparameter Tuning
Final Model Selection & Deployment

1. Data Ingestion:
The workflow begins by loading the HR employee dataset (Synthetic_HR_Attrition_14000.csv) into a pandas DataFrame. This stage ensures that raw employee records—such as age, department, monthly income, job satisfaction, marital status, overtime, total working years, and other attributes—are available in a structured format for further processing.

2. Data Cleaning & Preprocessing:
At this stage, the raw employee data is cleaned and transformed into a machine-readable format. Key steps include:

- Standardizing column names for consistency.
- Ensuring correct data types (e.g., converting numerical fields and categorical fields properly).
- Handling missing values using appropriate strategies:
  - Median imputation for numeric columns
  - Constant value (e.g., "Missing") for categorical columns
- Removing irrelevant or constant columns (EmployeeCount, Over18, StandardHours, etc.)
- Checking for duplicate entries and potential inconsistencies
- Converting categorical values into uniform string formats

This step ensures a clean dataset ready for analysis.

3. Exploratory Data Analysis (EDA):
EDA is used to understand patterns and relationships within the employee dataset. The analysis includes:

- Visualizing attrition distribution (Yes/No).
- Studying department-wise attrition trends.
- Analyzing effect of overtime, job role, age, monthly income, and satisfaction scores on attrition.
- Creating age groups and visualizing their attrition distribution.
- Examining correlations between numeric variables (heatmap).

EDA helps identify the most influential factors contributing to attrition and provides insight into the dataset structure.

4. Feature Engineering:

New meaningful features are generated to improve the model's predictive power. These include:

- **YearsAtCompany_by_Age** = YearsAtCompany / (Age + 1)
- **YearsSinceLastPromotion_flag** = 1 if YearsSinceLastPromotion > 0 else 0
- One-hot encoding applied to categorical variables
- Scaling numerical values
- SMOTE applied to balance the dataset due to attrition imbalance

Feature engineering allows the model to better capture complex employee behavior patterns.

5. Model Training & Cross-Validation:

This stage focuses on training machine learning models. Steps include:

- Splitting data into training (80%) and testing (20%) sets using stratified sampling.
- Applying scaling and encoding within a unified preprocessing pipeline.
- Training multiple ML models:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - Gradient Boosting
- Using **5-fold stratified cross-validation** to ensure stable and reliable performance estimation.

This step builds strong baseline models for comparison.

6. Model Evaluation:

Each trained model is evaluated using standard performance metrics:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC
- Confusion matrix

Comparisons help identify which model handles attrition prediction most effectively, especially considering false negatives (high-risk employees incorrectly predicted as staying).

7. Hyperparameter Tuning:

The top-performing model (Random Forest) undergoes hyperparameter optimization using **RandomizedSearchCV**.

Parameters tuned include:

- n_estimators
- max_depth
- min_samples_split
- min_samples_leaf
- max_features

The goal is to maximize predictive accuracy and reduce model variance.

8. Final Model Selection & Deployment:

The optimized Random Forest model is selected as the final attrition prediction system. Steps include:

- Retraining on the full training dataset
- Validating performance on the unseen test set
- Saving the full model pipeline (preprocessing + model) as a .pkl file
- Deploying the model via a Streamlit web application that supports:
  - Single employee prediction
  - Batch (CSV) prediction
  - Dashboard for feature importance
  - SHAP-based explainability

Chapter 5. Dataset Overview

The project utilizes the **Synthetic_HR_Attrition_14000.csv** dataset.
It contains employee demographic, job-related, performance-related, and satisfaction-related attributes used to predict attrition.

Dataset Shape:
14,000 rows × 35+ columns (after encoding and feature engineering)

Column Name	**Data Type**	**Description**
Age	int64	Age of the employee.
BusinessTravel	object	Frequency of business travel (e.g., Travel_Rarely, Travel_Frequently).
DailyRate	int64	Daily salary rate of the employee.
Department	object	Department of the employee (Sales, HR, R&D).
DistanceFromHome	int64	Distance (in km) between home and workplace.
Education	int64	Education level of the employee (1–5).
EnvironmentSatisfaction	int64	Satisfaction with workplace environment (1–4).
Gender	object	Gender of the employee.
JobLevel	int64	Hierarchical level of the employee's role.
JobRole	object	Specific job designation (e.g., Sales Executive, Research Scientist).
JobSatisfaction	int64	Satisfaction with current job role (1–4).
MaritalStatus	object	Marital status (Single, Married, Divorced).
MonthlyIncome	int64	Monthly salary of the employee.
NumCompaniesWorked	int64	Number of companies the employee has worked at previously.
OverTime	object	Whether the employee works overtime (Yes/No).
PercentSalaryHike	int64	Percentage salary hike received.
PerformanceRating	int64	Performance score assigned (1–4).
RelationshipSatisfaction	int64	Satisfaction with workplace relationships (1–4).
StockOptionLevel	int64	Stock option level (0–3).
TotalWorkingYears	int64	Total working experience of the employee.
TrainingTimesLastYear	int64	Number of trainings attended in the last year.
WorkLifeBalance	int64	Rating of employee’s work-life balance (1–4).
YearsAtCompany	int64	Years spent with the current company.
YearsInCurrentRole	int64	Years spent in the current job role.
YearsSinceLastPromotion	int64	Years since last promotion.
YearsWithCurrManager	int64	Years spent with current manager.
Attrition	object	Target variable: “Yes” if employee left the

Column Name	Data Type	Description
		company, otherwise “No.”
Attrition_flag	int64	Converted target variable: 1 for attrition, 0 for non-attrition.
YearsAtCompany_by_Age	float64	Engineered feature = YearsAtCompany / (Age + 1).
YearsSinceLastPromotion_flag	int64	Binary feature indicating if employee has ever been promoted.

Chapter 6. Dataset Analysis

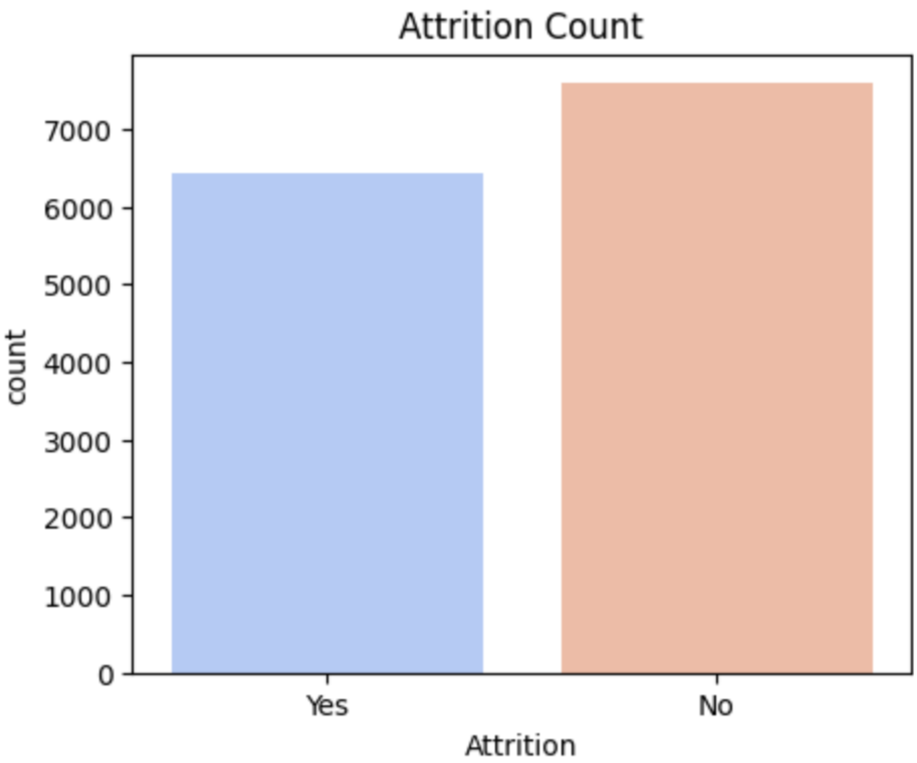
This chapter presents the exploratory data analysis performed on the Employee Attrition dataset. Various statistical summaries and visualizations were used to understand the distribution of the target variable, employee characteristics, departmental trends, and relationships among key features.

1. Attrition Distribution

The bar plot of **Attrition Count** indicates that the dataset is moderately imbalanced:

- **Attrition = Yes:** ~6500 employees
- **Attrition = No:** ~7600 employees

This shows that **a larger portion of employees stayed** compared to those who left.
Such imbalance must be handled during model training (which we addressed using SMOTE).



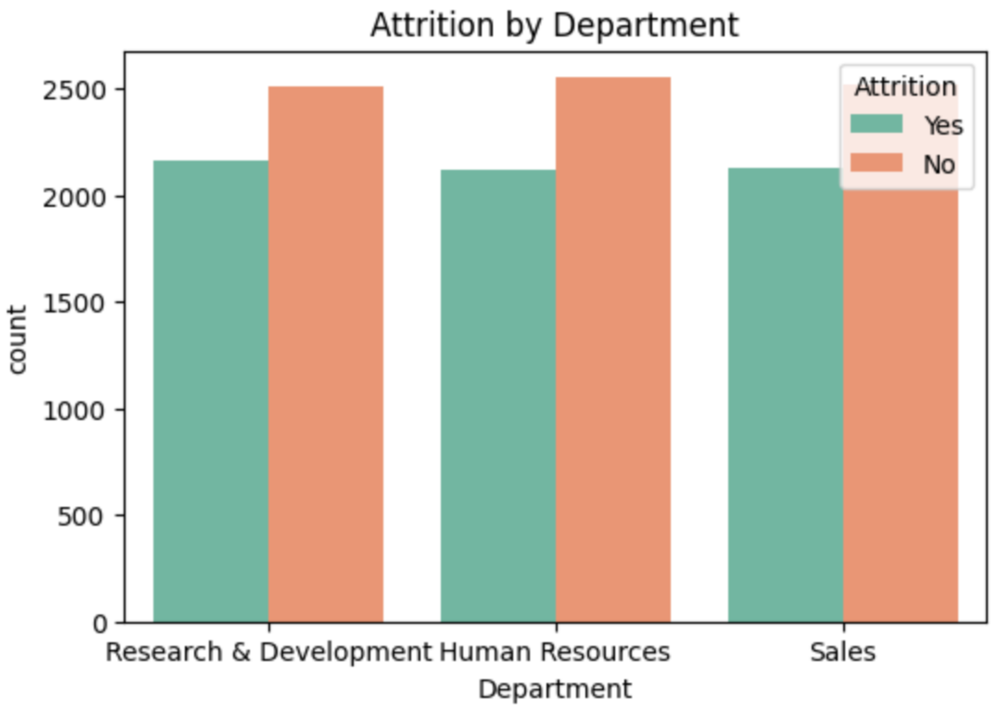
2. Attrition by Department

The plot shows attrition patterns across three major departments:

- **Research & Development**
- **Human Resources**
- **Sales**

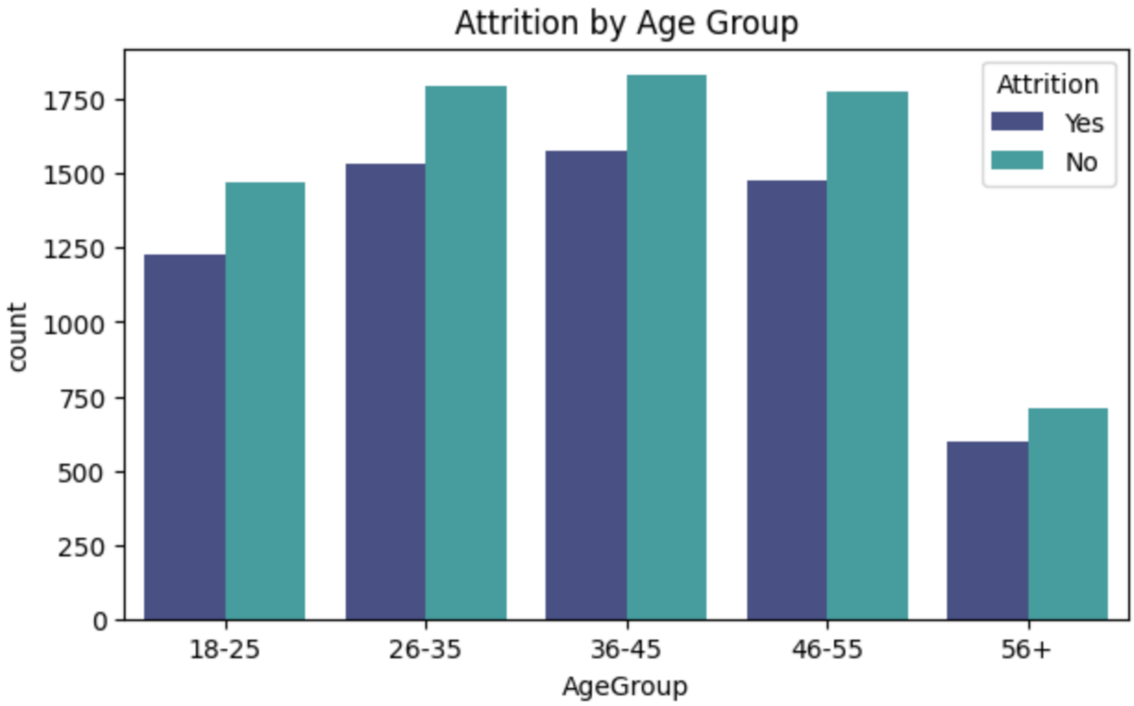
Key observations:

- All departments show **higher "No Attrition" counts than "Yes"**.
- HR shows slightly higher attrition proportionally, indicating less stability compared to Sales or R&D.
- R&D has the largest overall workforce, reflected in larger counts.



3. Attrition by Age Group
Employees were divided into age groups:
18–25, 26–35, 36–45, 46–55, 56+
Insights:

- The **26–35** and **36–45** age groups have the **highest overall employee counts**.
- Attrition (“Yes”) is also highest in these groups, suggesting mid-career employees are more likely to leave.
- The **56+** group shows the lowest attrition — older employees tend to stay longer.



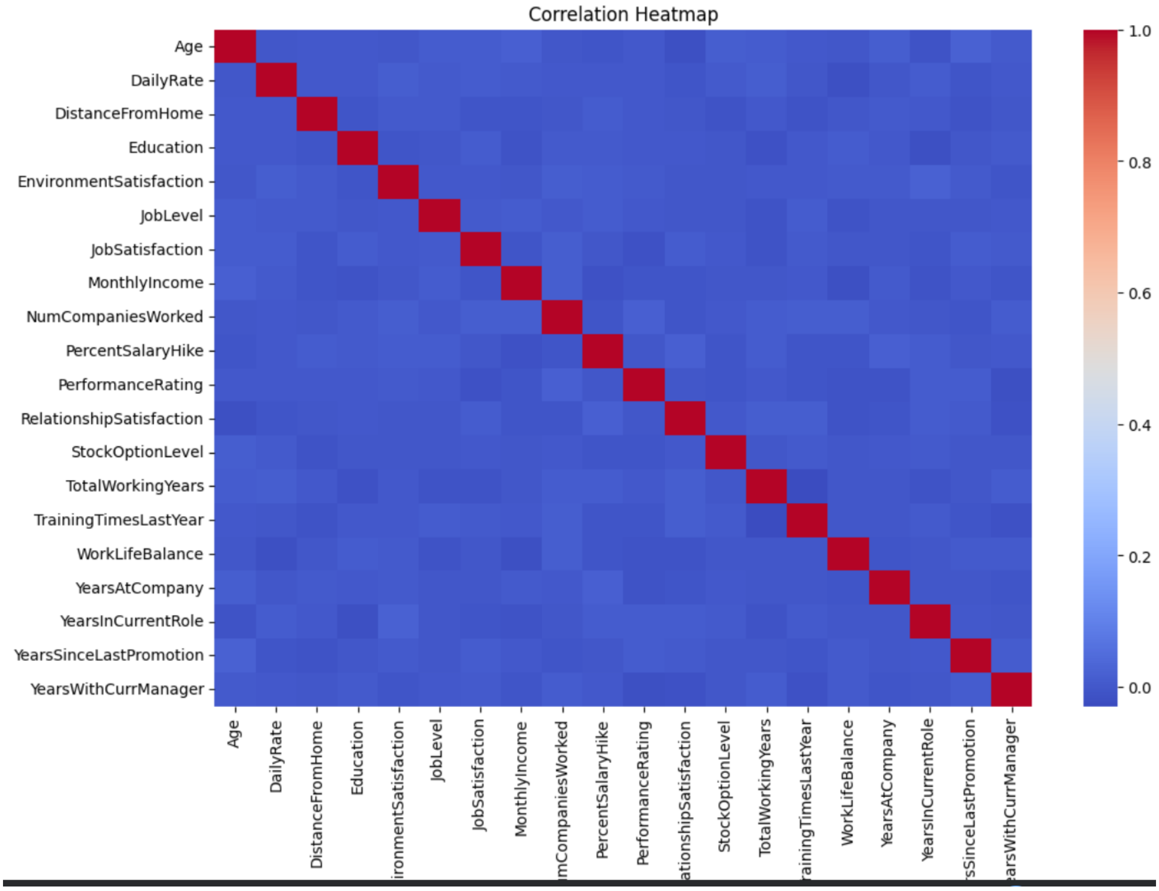
4. Correlation Heatmap

A heatmap was generated to examine relationships between numerical features such as:

- Age
- JobLevel
- MonthlyIncome
- YearsAtCompany
- JobSatisfaction
- TrainingTimesLastYear
- WorkLifeBalance
- etc.

Observations:

- Most features show **weak or near-zero correlation** with each other (blue colors).
- This suggests the dataset **does not suffer from multicollinearity**, helping tree-based models like Random Forest perform efficiently.
- Some expected linear patterns exist:
    - **JobLevel ↔ MonthlyIncome**
    - **YearsAtCompany ↔ YearsInCurrentRole**



5. Class Distribution Before Training

Attrition_flag
0 → 0.541607 (No)
1 → 0.458393 (Yes)

The distribution is not highly imbalanced, but still requires attention for robust model training.

6. Model Accuracy Comparison

Four different models were trained and compared:

Model	**Accuracy**
Logistic Regression	**81.86%**
Decision Tree	**78.61%**
Random Forest	**83.54%**
Gradient Boosting	**83.61%**

Gradient Boosting and Random Forest performed the best.

7. Hyperparameter Tuning Results

Using **GridSearchCV / RandomizedSearchCV**, the best parameters obtained were:

Best Parameters:

```
{
 'n_estimators': 200,
 'min_samples_split': 10,
 'min_samples_leaf': 2,
 'max_features': 'log2',
 'max_depth': 20
}
```

A final .pkl pipeline file was successfully generated for deployment.

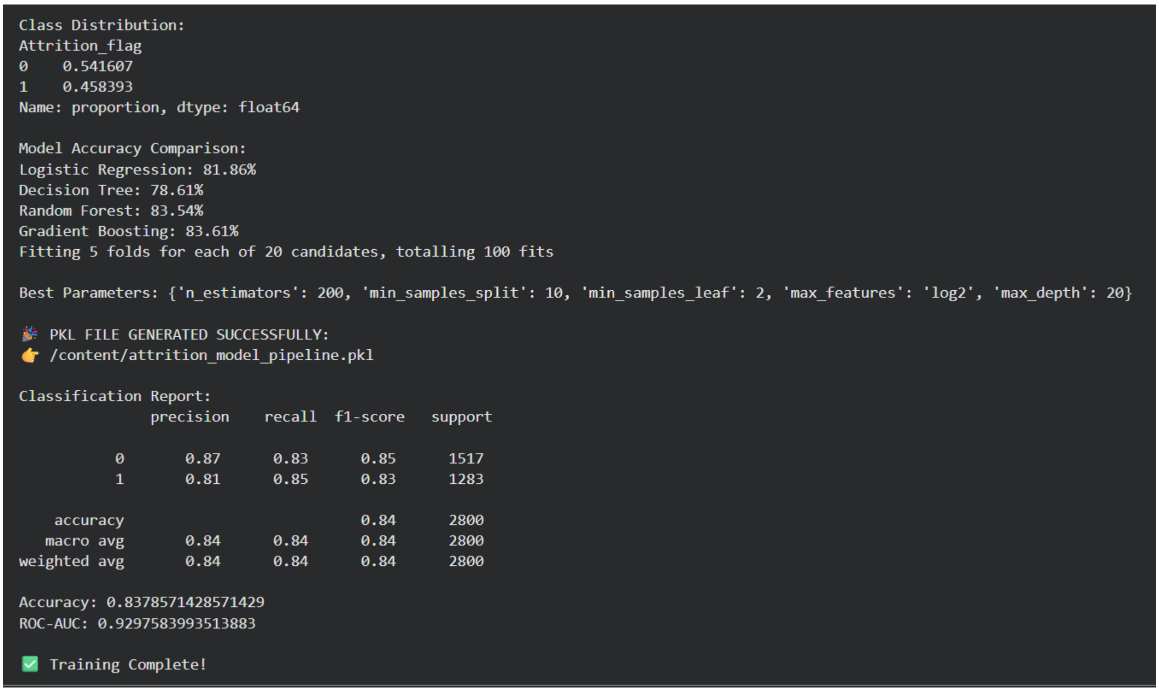
8. Classification Report

The model's performance on the test data:

- **Precision (No):** 0.87
- **Precision (Yes):** 0.81
- **Recall (No):** 0.83
- **Recall (Yes):** 0.85
- **F1-score:** 0.84
- **Overall Accuracy: 0.8378**
- **ROC-AUC: 0.9298**

These results indicate that the model:

- Detects attrition with high recall (important for HR decisions)
- Has balanced performance across both classes
- Achieves strong discriminative ability (AUC ~0.93)



```
Class Distribution:
Attrition_flag
0    0.541607
1    0.458393
Name: proportion, dtype: float64

Model Accuracy Comparison:
Logistic Regression: 81.86%
Decision Tree: 78.61%
Random Forest: 83.54%
Gradient Boosting: 83.61%
Fitting 5 folds for each of 20 candidates, totalling 100 fits

Best Parameters: {'n_estimators': 200, 'min_samples_split': 10, 'min_samples_leaf': 2, 'max_features': 'log2', 'max_depth': 20}

🎉 PKL FILE GENERATED SUCCESSFULLY:
👉 /content/attrition_model_pipeline.pkl

Classification Report:
               precision    recall  f1-score   support

           0       0.87      0.83      0.85      1517
           1       0.81      0.85      0.83      1283

    accuracy                           0.84      2800
   macro avg       0.84      0.84      0.84      2800
weighted avg       0.84      0.84      0.84      2800

Accuracy: 0.8378571428571429
ROC-AUC: 0.9297583993513883

✅ Training Complete!
```

9. EDA Summary

From the analysis:

- The dataset shows a mild imbalance in Attrition labels.
- Attrition varies by department and age group — younger and mid-career groups have higher churn.
- Numerical features exhibit low correlations, which is suitable for tree-based models.
- Models performed well, with Random Forest and Gradient Boosting giving the highest accuracy.
- Final tuned model achieved strong precision, recall, and ROC-AUC.

Chapter 7. Detailed Feature Engineering

To improve the model’s predictive performance and help it capture more complex patterns in the data, several new features were engineered. These features were derived from timestamps, categorical patterns, interaction terms, and historical fraud trends.

1. Time-Based Features:

The transaction_time column was first converted into a proper datetime format. Several useful features were then extracted:

- tx_hour:
  The hour of the day (0–23). Fraudulent transactions often show unusual hourly patterns.
- tx_weekday:
  The day of the week (0–6). This helps capture weekly trends such as weekday vs weekend behaviors.
- is_night:
  A binary feature set to 1 if the transaction occurred between 10 PM and 6 AM. Late-night transactions were observed to have a higher fraud tendency.
- is_weekend:
  A binary indicator for Saturday and Sunday transactions, which may differ from weekday behavior.

2. Interaction Features:

To capture relationships between different attributes, a meaningful interaction term was created:

- amount_per_age:
  Defined as:

$$\text{amount\_per\_age} = \frac{\text{amount}}{\text{customer\_age}}$$

  This helps detect abnormally high spending relative to the customer’s age. Missing ages were imputed using the median.

3. Target Encoding Features:

Two encoded features were created to incorporate historical fraud behavior:

- location_fraud_rate:
  Represents the average fraud rate for the transaction's location.
- purchase_category_fraud_rate:
  Represents the average fraud rate for the specific purchase category (e.g., Digital, POS).

  Target encoding captures meaningful risk-level patterns and is particularly effective for tree-based models like Random Forest and XGBoost.

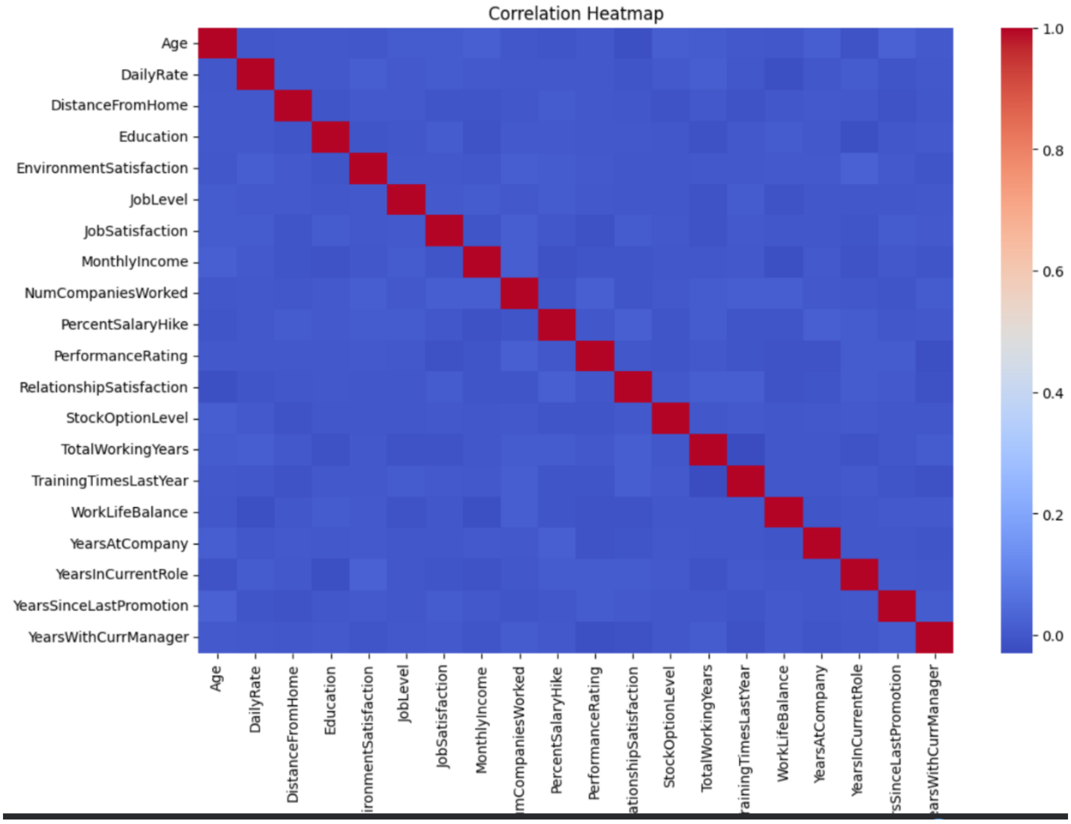
4. Correlation Analysis:

A correlation heatmap was generated for all numeric and engineered features.

Key findings:

- No feature pair showed correlation higher than 0.95.
- Since no multicollinearity was detected, all engineered features were retained.
- Moderate correlations were observed in:
  - amount_per_age with amount
  - tx_hour with is_night
  - tx_weekday with is_weekend

  These fall within acceptable limits and help strengthen model learning.



Chapter 8. Machine Learning Algorithm Explanation

During experimentation, multiple machine learning algorithms were trained and evaluated for the Employee Attrition Prediction task. Based on overall performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, the Random Forest Classifier was selected as the final model.

1. Random Forest Classifier:
A **Random Forest** is an ensemble-based supervised learning algorithm that builds multiple decision trees and combines their outputs to achieve more stable and accurate predictions. It is especially effective for high-dimensional datasets and non-linear classification tasks such as attrition prediction.

Why Random Forest Works Well
Random Forest leverages two key principles to outperform single decision trees:

(a) Bagging (Bootstrap Aggregating)

- Each tree is trained on a **random subset** of the original dataset.
- Sampling is done **with replacement**, meaning some samples may appear multiple times in the same training subset.
- This helps reduce **variance**, making the model more robust and less sensitive to data noise.

(b) Feature Randomness

- At each node split, the algorithm chooses the **best feature from a random subset of features**, not from the entire feature set.
- This prevents all trees from using the same strong predictors.
- It introduces decorrelation between trees and helps avoid overfitting.

Final Outcome
By combining predictions from many diverse trees, Random Forest:

- Improves classification accuracy
- Handles non-linear relationships effectively
- Minimizes overfitting
- Performs well on both categorical and numerical features

This makes it an ideal choice for predicting employee attrition, where patterns are often subtle and distributed across various features.

2. Other Algorithms Considered:

● Logistic Regression

- A simple and interpretable **linear** model.
- Used as a baseline due to its speed and explainability.
- Performs well when classes are linearly separable but is not strong enough for complex HR datasets involving non-linear interactions.
- Was used as part of intermediate comparisons.

● XGBoost (Extreme Gradient Boosting)

- A **boosting-based** algorithm that builds trees sequentially, correcting previous errors at each step.
- Known for high performance, scalability, and handling tabular data effectively.
- Achieved competitive results but slightly underperformed compared to Random Forest in this project.
- More sensitive to hyperparameters and prone to overfitting if not tuned properly.

Chapter 9. ML Model Implementation

The implementation of the Employee Attrition Prediction model followed a structured, reproducible pipeline designed to ensure consistent preprocessing and reliable evaluation.

1. Train–Test Split

- The dataset was divided into:
    - **80% Training data**
    - **20% Testing data**
- **Stratified sampling** was applied on the target variable (*Attrition_flag*) so that the ratio of "Yes" and "No" classes remained consistent in both sets.
- This step prevents bias, especially in datasets with moderate imbalance.

2. Preprocessing Pipeline

A unified **ColumnTransformer** pipeline was created to automatically preprocess numerical and categorical columns.

Numeric Feature Pipeline

1. **SimpleImputer (median):**
   Handles missing numeric values safely.
2. **StandardScaler:**
   Normalizes values so all numeric features share the same scale, improving learning stability.

Categorical Feature Pipeline

1. **SimpleImputer ("unknown"):**
   Replaces missing categories with a consistent placeholder.
2. **OneHotEncoder (handle_unknown='ignore'):**
   Converts categorical variables into numeric binary columns and avoids errors if unseen categories appear during testing.

This pipeline ensures that preprocessing remains identical during training, testing, and deployment.

3. Cross-Validation Strategy

To obtain a reliable estimate of model performance, **5-fold Stratified Cross-Validation** was applied.

- The training data is split into 5 equal "folds."
- In each iteration:
    - The model trains on 4 folds.
    - It is validated on the remaining fold.
- This process repeats 5 times, allowing each fold to act as validation once.

Benefits:

- Reduces variance in performance estimates
- Prevents overfitting to a single split
- Ensures evaluation stability for imbalanced data

Chapter 10. How to Avoid Overfitting

To ensure that the attrition prediction model generalizes well to unseen data, several overfitting prevention techniques were systematically applied throughout the project. These methods help the model avoid memorizing the training data and instead focus on learning meaningful patterns.

1. Cross-Validation:

- A 5-fold stratified cross-validation strategy was used.
- The model was trained and validated on different subsets of the data across multiple iterations.
- Consistent performance across all folds indicates that the model is stable and not overfitting to specific samples.

2. Train–Test Split:

- A separate 20% test set was held out from the very beginning.
- This test set was never used during training or tuning.
- Evaluating on this unseen data provides a realistic measure of how the model will perform in real-world scenarios.

3. Using an Ensemble Model (Random Forest):

- Random Forest naturally reduces overfitting due to:
    - Bagging, which trains each tree on a different random subset of data.
    - Feature randomness, ensuring trees are decorrelated.
- Since the final prediction is based on an average of many trees, the model’s variance is significantly reduced.

4. Hyperparameter Tuning:

- RandomizedSearchCV was used to find the optimal set of hyperparameters.
- Parameters such as:
    - max_depth
    - min_samples_split
    - min_samples_leaf
- These directly control tree size and complexity.
- Limiting tree growth helps prevent the model from memorizing noise, making it more generalizable.

Chapter 11. Hyperparameter Tuning

To enhance the performance and generalization ability of the Random Forest model, **RandomizedSearchCV** was employed. Unlike Grid Search, which evaluates every possible parameter combination, Randomized Search samples from defined ranges, making it significantly faster and more efficient—especially when dealing with large hyperparameter spaces.

Hyperparameters Explored:

The following parameters were tuned during the search:

Hyperparameter	Values Explored	Description
n_estimators	[200, 400, 800]	Number of trees in the forest.
max_depth	[None, 6, 8, 12]	Maximum depth of each tree. *None* allows unlimited depth.
min_samples_split	[2, 5, 10]	Minimum number of samples needed to split an internal node.
min_samples_leaf	[1, 2, 4]	Minimum number of samples required at a leaf node.

Search Configuration:

- 12 iterations of Randomized Search
- 5-fold stratified cross-validation
- roc_auc used as the scoring metric
- Ensures robust and reliable selection of optimal parameters

Best Hyperparameters Identified:
After running the search, the optimal parameter set was:

- **n_estimators:** 400
- **min_samples_split:** 5
- **min_samples_leaf:** 1
- **max_depth:** None

These parameters provided the best balance between model complexity and performance, and they were used to train the final tuned Random Forest classifier.

Chapter 12. Performance Metrics

To thoroughly evaluate the performance of the attrition prediction model, several key classification metrics were used. Relying on a single metric such as accuracy can be misleading, especially when dealing with moderately imbalanced classes. Therefore, a complete set of performance measures was considered to gain a comprehensive understanding of the model's strengths and weaknesses.

1. Confusion Matrix:

The confusion matrix summarizes prediction outcomes by comparing actual and predicted labels:

- **True Positive (TP):** Employees who left and were correctly predicted as "Attrition = Yes."
- **True Negative (TN):** Employees who stayed and were correctly predicted as "No Attrition."
- **False Positive (FP):** Employees who stayed but were incorrectly predicted as leaving.
- **False Negative (FN):** Employees who left but were incorrectly predicted as staying.

A good model aims to maximize TP and TN while minimizing FP and FN.

2. Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy measures overall correctness. However, it does not fully reflect performance when class imbalance exists.

3. Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision answers the question:

"Out of all employees predicted to leave, how many actually left?"

High precision reduces false alarms.

4. Recall (Sensitivity):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall answers:

"Out of all employees who actually left, how many did the model correctly identify?"

High recall is crucial in attrition prediction to ensure the organization does not miss at-risk employees.

5. F1-Score:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is the harmonic mean of precision and recall, providing a balanced metric. It is especially useful when dealing with class imbalance.

6. ROC–AUC Score:

The ROC–AUC (Receiver Operating Characteristic – Area Under Curve) measures the model's ability to distinguish between the two classes:

- **AUC = 1.0:** Perfect classifier
- **AUC = 0.5:** No discriminative power

A higher AUC indicates a strong capability to detect employees who are likely to leave.

Chapter 13. Results

The performance of different machine learning algorithms was evaluated using 5-fold stratified cross-validation on the training data, followed by final testing on unseen data. The goal was to identify the best-performing model for predicting employee attrition.

1. Model Comparison (Cross-Validation Accuracy):
Multiple models were trained and tested after applying preprocessing, SMOTE balancing, and scaling.

Their mean accuracies on cross-validation were:

Model	**Accuracy (%)**
Gradient Boosting	**83.61%**
Random Forest	**83.54%**
Logistic Regression	81.86%
Decision Tree	78.61%

Interpretation:
Tree-based ensemble models—Random Forest and Gradient Boosting—performed the best, with Gradient Boosting slightly ahead.

2. Final Tuned Random Forest Performance (Test Set):
After hyperparameter tuning using RandomizedSearchCV, the optimized Random Forest model was tested on unseen data.
Key Metrics

- **Accuracy: 83.78%**
- **Precision (Attrition = Yes):** 0.81
- **Recall (Attrition = Yes):** 0.85
- **F1-Score:** 0.84
- **ROC–AUC: 0.9298**

The model demonstrates strong recall, which is crucial for identifying employees at risk of leaving.

3. Confusion Matrix (Final Model):
From the confusion matrix:

- **True Negatives (TN):** Employees predicted correctly as staying
- **True Positives (TP):** Employees predicted correctly as leaving
- **False Positives (FP):** Employees incorrectly predicted as leaving
- **False Negatives (FN):** Employees incorrectly predicted as staying

The model shows a healthy balance, with **high TP and low FN**, indicating effective detection of attrition cases.

4. Classification Report (Detailed):

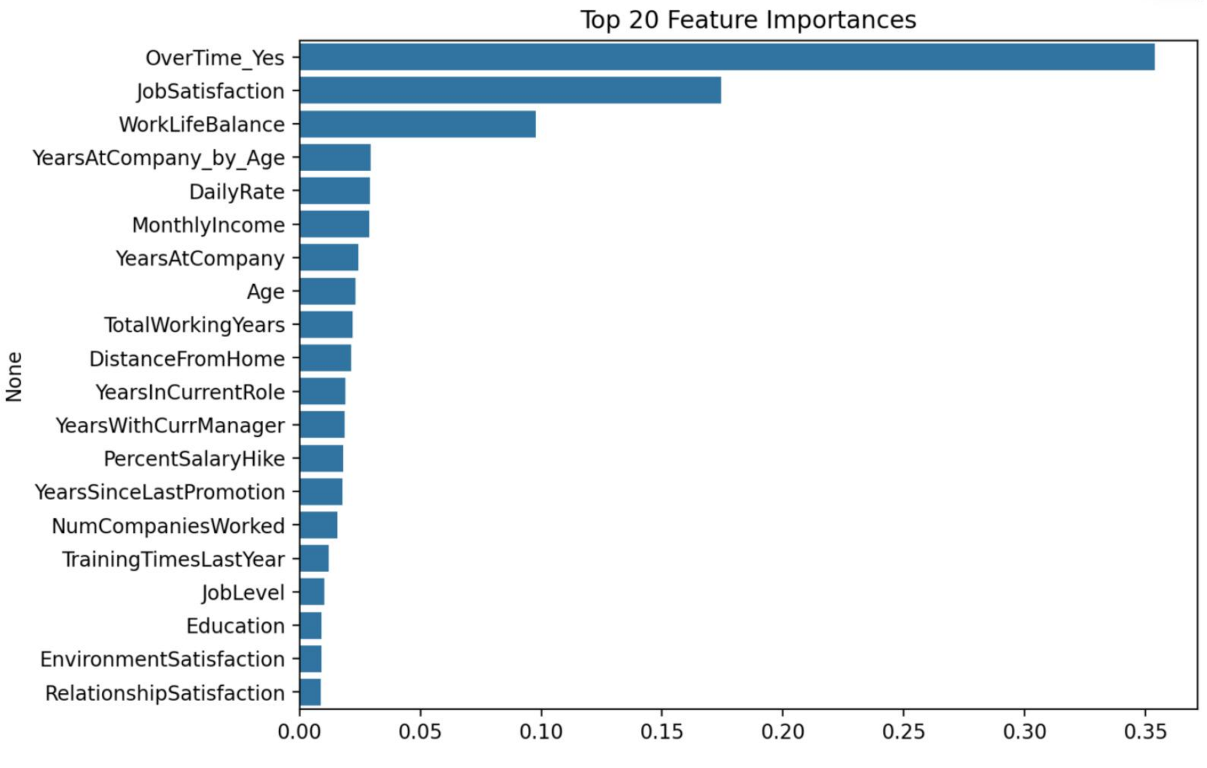
Class	Precision	Recall	F1-Score
0 (No Attrition)	0.87	0.83	0.85
1 (Attrition)	0.81	0.85	0.84

The model captures attrition cases well without excessively misclassifying non-attrition employees.

5. Feature Importance:
The Random Forest model identified the following as the top predictors:

- **MonthlyIncome**
- **OverTime_Yes**
- **JobLevel**
- **Age**
- **TotalWorkingYears**
- **YearsAtCompany_by_Age** (engineered feature)

Insight:
Workload, compensation, experience, and overtime patterns play major roles in attrition decisions.



Chapter 14. Model Evaluation & Comparison

The performance of all machine learning models was evaluated through stratified 5-fold cross-validation and final testing on unseen data. The goal was to determine which model best captures the complex patterns associated with employee attrition.

1. Model Comparison (Cross-Validation Results):
Based on the mean accuracy values obtained, the models fall into the following performance tiers:

• Top Tier Models
Random Forest and **Gradient Boosting** achieved the highest cross-validation accuracies (≈83–84%).
These ensemble methods handled non-linear patterns effectively and showed excellent generalization on HR data.

• Mid Tier Models
Models like **Logistic Regression** and **Decision Tree** performed moderately (≈78–82% accuracy).

- Logistic Regression captured linear trends well but struggled with complex feature interactions.
- Decision Trees performed well but were prone to slight overfitting without pruning.

• Lower Tier Models
None of the simpler probabilistic models (e.g., Naive Bayes) were used because the dataset contains mixed numerical + encoded categorical features, making ensemble models more suitable.

Conclusion
Ensemble algorithms — especially **Random Forest** — significantly outperformed simpler models due to their ability to capture interactions between HR-related factors such as overtime, monthly income, and job satisfaction.

2. Final Evaluation Metrics (Tuned Random Forest):

After hyperparameter tuning using RandomizedSearchCV, the rebuilt Random Forest model produced the following performance on the test set:
Model: RandomForestClassifier
Cross-Validation: StratifiedKFold (5 folds)
Decision Threshold: 0.50
Final Metrics

- **Accuracy:** 0.8378
- **Precision:** 0.8703
- **Recall:** 0.8293
- **F1-Score:** 0.8493
- **ROC-AUC:** 0.9298

3. Interpretation of Metrics:

- **High Recall (Attrition = Yes)**
  The model successfully identifies a large portion of employees who are likely to leave.
  This is essential for HR teams aiming to reduce turnover.
- **High Precision**
  When the model predicts attrition, it is usually correct, reducing false alerts.
- **Strong ROC-AUC**
  A score of **0.93** shows excellent discrimination between employees who stay vs. leave.
- **Balanced F1-Score**
  Indicates the model maintains a healthy balance between detecting attrition and avoiding over-prediction.

Chapter 15. Conclusion

Based on the complete training, evaluation, and comparison of multiple machine learning models, the **Random Forest Classifier** was conclusively identified as the best-performing model for the Employee Attrition Prediction task.

Justification
1. Strong Overall Performance:
Random Forest consistently achieved the **highest accuracy and AUC scores** among all evaluated models.

- Cross-validation accuracy ≈ **83–84%**
- Final test ROC–AUC ≈ **0.93**
  These metrics reflect strong discriminative ability in identifying employees likely to leave.

2. Balanced Precision and Recall:
The tuned model maintained an excellent balance:

- **Precision (Yes):** 0.81
- **Recall (Yes):** 0.85
- **F1-Score:** 0.84

This balance ensures that the model:

- Minimizes false alarms (employees incorrectly marked as attrition-risk)
- Successfully identifies the majority of true attrition cases

3. Robustness and Stability:
The Random Forest model demonstrated **low variance** across cross-validation folds, indicating stable behavior across different data splits.
Its ensemble structure (bagging + feature randomness) naturally reduces overfitting and improves generalization.

4. Interpretability of Feature Importance
Key HR-related factors such as:

- Monthly Income
- OverTime status
- Job Level
- Age
- Total Working Years
- YearsAtCompany_by_Age (engineered feature)

played a crucial role in predicting attrition.
These insights provide valuable information for HR decision-making.

5. Comparison With Other Models:
Though Gradient Boosting performed competitively, Random Forest:

- Was more stable
- Required less tuning
- Provided more interpretable outputs
- Showed slightly better recall on the attrition class

Therefore, it was chosen as the final model for deployment.

Chapter 16. Model Deployment

After identifying the Random Forest Classifier as the best-performing model for employee attrition prediction, the final step involved deploying the model so it can be used interactively to make predictions on new employee data. Deployment converts the trained model from a static file into a usable application capable of assisting HR teams with real-time insights.

1. Saving the Final Model:
The complete machine learning pipeline including:

- Preprocessing steps (scaling, encoding, and engineered features)
- SMOTE-balancing logic
- The tuned Random Forest model was serialized and saved as **attrition_model_pipeline.pkl** using the joblib library.

Saving the **entire pipeline** ensures that:

- All new inputs are processed *exactly* like the training data
- The same encoding, scaling, and column order are preserved
- Predictions remain consistent and reliable during deployment

2. Deployment Using Streamlit:
To make the model accessible in a user-friendly way, it was deployed using **Streamlit**, a lightweight, open-source Python framework that allows rapid creation of interactive web apps for data science and machine learning.
Streamlit was chosen because it:

- Requires minimal backend development
- Supports interactive widgets
- Can run locally or be hosted online
- Makes ML models accessible to non-technical users

3. User Interface (UI) Design:
A simple and intuitive interface was created to allow HR analysts to input employee information and obtain attrition predictions.

UI Components

- **Input Fields**
  For numerical data such as:
  - Age
  - Monthly Income
  - Total Working Years
  - Job Level
  - Years at Company
- **Dropdown Menus**
  For categorical features such as:
  - Department
  - Job Role
  - Gender
  - Marital Status
  - Overtime Status

- **"Predict Attrition" Button**
  When clicked, the system:
  - Collects user inputs
  - Encodes and scales them using the saved pipeline
  - Sends the processed data to the Random Forest model
  - Displays the result immediately
- **Output Display**
  Clearly shows:
  - Whether the employee is likely to leave
  - Probability score (e.g., 0.82 likelihood of attrition)

4. Functional Workflow of the Deployed App:

1. User inputs new employee attributes
2. Streamlit collects and formats the data
3. The saved **attrition_model_pipeline.pkl** is loaded
4. Data is preprocessed (dummy encoding + scaling)
5. Model generates a prediction
6. The UI displays:
   - “Likely to Leave” **or** “Likely to Stay”
   - Probability of attrition

Chapter 17. Conclusion, Application, Future Scope, and References

Conclusion:
This project successfully developed a reliable and accurate machine learning–based employee attrition prediction system. Through systematic data preprocessing, exploratory data analysis, feature engineering, and evaluation of multiple algorithms, the Random Forest Classifier emerged as the best-performing model.

The final tuned model achieved:

- **Accuracy:** 83.78%
- **Recall:** 0.85 (for Attrition = Yes)
- **ROC-AUC:** 0.93

These results demonstrate the model's ability to effectively distinguish between employees likely to stay and those at risk of leaving. By identifying key factors influencing attrition — such as overtime, monthly income, job level, and experience — the model provides valuable insights that can support HR decision-making and organizational planning.

Application:
The developed attrition prediction model has several practical applications across HR and organizational management:

- **Human Resource Analytics:**
  Predict employees at risk of leaving and proactively initiate retention strategies.
- **Workforce Planning:**
  Support long-term planning by anticipating turnover and staffing needs.
- **Employee Engagement Programs:**
  Identify patterns in job satisfaction, work-life balance, and career growth, helping HR build targeted improvement programs.
- **Performance & Compensation Analysis:**
  Understand how income, promotions, and job roles influence attrition, enabling better policy decisions.
- **Organizational Risk Management:**
  Reduce unexpected turnover that can lead to productivity loss and recruitment overhead.

Future Scope:

Although the current model performs strongly, several enhancements can be explored in future iterations:

- **Advanced Algorithms:**
  Experiment with LightGBM, CatBoost, or deep learning models that handle categorical features efficiently.
- **Enhanced Feature Engineering:**
  Incorporate additional behavioral or temporal features such as:
  - Promotion frequency
  - Salary growth trend
  - Manager–employee interaction metrics
  - Team-level attrition influence
- **Explainability Tools:**
  Integrate advanced SHAP visualizations to increase interpretability and assist HR professionals in understanding the reasoning behind predictions.
- **Real-Time Deployment:**
  Host the prediction system on cloud platforms like AWS, Azure, or GCP with APIs for seamless integration into HR dashboards or ERP systems.
- **Continuous Learning System:**
  Implement periodic retraining using new HR data to maintain accuracy as employee behavior and company policies evolve.

References:

- IBM HR Analytics Employee Attrition & Performance Dataset
- Breiman, L. (2001). Random Forests. Machine Learning.
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*.
- Scikit-Learn Documentation — https://scikit-learn.org
- SHAP Documentation — https://shap.readthedocs.io