

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Employee Turnover Prediction | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Rudra Garg |  | | | |
|  | | | | Roll No.: 2201170—Course: Machine Learning— |  | | | |
|  | | |  | | |  | | |

# Introduction

##### *What is Employee Turnover?*

*Employee Turnover or Employee Turnover ratio is the measurement of the total number of employees who leave an organization in a particular year. Employee Turnover Prediction means to predict whether an employee is going to leave the organization in the coming period.*A Company uses this predictive analysis to measure how many employees they will need if the potential employees will leave their organization. A company also uses this predictive analysis to make the workplace better for employees by understanding the core reasons for the high turnover ratio.

##### Summary

The goal of this project was to predict employee turnover using machine learning models. Five models were developed and evaluated on a dataset of 14,999 records with extensive feature engineering expanding the feature set from 10 to 1,606. Astonishingly, all models achieved perfect scores across all evaluation metrics, including accuracy, precision, recall, and F1-scores on the test set. While this outcome highlights the potential strength of the engineered features, it also suggests possible data leakage or overfitting, which must be carefully examined.

Through systematic implementation, model tuning, and analysis, the project demonstrates the effectiveness of machine learning in tackling employee turnover prediction. However, concerns related to data quality and the feature engineering process highlight areas for further refinement and validation.

# Overview

##### Dataset Characteristics

The dataset comprised 14,999 records, representing employee-level data, with turnover status as the target variable. Features included various employee attributes, satisfaction scores, engagement metrics, and workplace factors.

**Key Dataset Details:**

* **Samples (Total):** 14,999
* **Features (Post Engineering):** 1,606
* **Test Set Size:** 4,500 (30% of total dataset)
* **Class Distribution (Test Set):**
  + **Non-turnover (Class 0):** 3,428 samples (76.2%)
  + **Turnover (Class 1):** 1,072 samples (23.8%)

**Feature Engineering:**

The feature engineering process expanded the original dataset from 10 features to 1,606. This involved the creation of interaction terms, polynomial features, and various derived attributes to better capture relationships between features. However, the resulting high-dimensional feature space introduces risks of overfitting and model complexity, necessitating further scrutiny.

##### Methodology

The project followed a systematic, end-to-end machine learning workflow:

**Key Steps:**

1. **Data Loading and Preprocessing:**
   * Handled missing values, normalized numerical features, and encoded categorical features using one-hot encoding and label encoding.
2. **Feature Engineering:**
   * Expanded features using interaction terms and polynomial transformations.
   * Focused on generating features related to employee satisfaction, engagement, and workplace metrics.
3. **Model Training and Hyperparameter Optimization:**
   * Trained Logistic Regression, Random Forest, XGBoost, LightGBM, and Neural Network models.
   * Applied hyperparameter tuning using grid search and cross-validation.
4. **Evaluation and Analysis:**
   * Assessed models using multiple performance metrics (accuracy, precision, recall, F1-score, and ROC AUC).
   * Visualized feature importance, confusion matrices, and ROC curves for interpretability.

# Model Performance Analysis

##### Model Training Times

Training times varied significantly across models, reflecting differences in complexity and computational demands:

* **Logistic Regression:** ~9.5 minutes
* **Random Forest:** ~7.5 minutes
* **XGBoost:** ~60 minutes
* **LightGBM:** ~25.5 minutes
* **Neural Network:** ~38 seconds

The neural network, being relatively shallow, had the fastest training time. In contrast, XGBoost required the most computational effort due to its iterative, tree-based optimization process.

##### Model Performance Metrics

All models achieved **perfect scores** on the test set, a highly unusual result warranting further investigation:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | ROC AUC | Precision (Class 0) | Precision (Class 1) | Recall (Class 0) | Recall (Class 1) | F1-Score (Both) |
| Logistic Regression | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Random Forest | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| XGBoost | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| LightGBM | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Neural Network | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

##### Best Hyperparameters

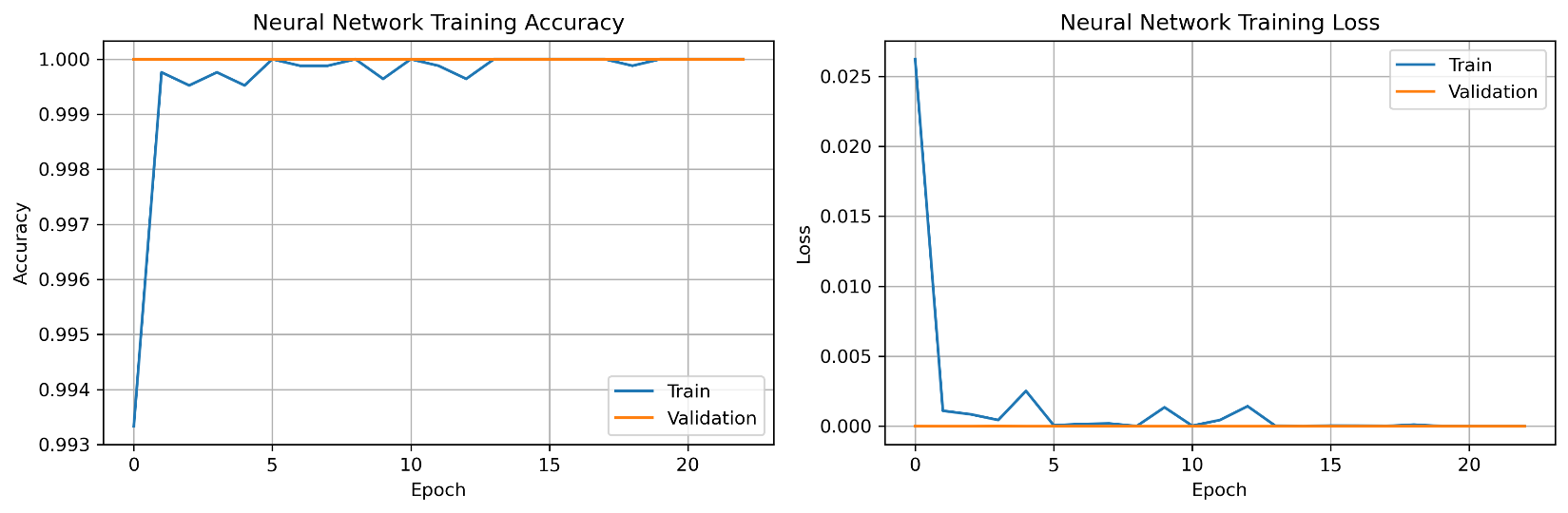
Optimal hyperparameters were identified through grid search and are as follows:

* **Logistic Regression:**
  + **Regularization Strength (C):** 0.001
  + **Penalty:** L1
  + **Solver:** liblinear
* **Random Forest:**
  + **Max Depth:** 10
  + **Min Samples Leaf:** 1
  + **Min Samples Split:** 2
  + **N Estimators:** 100
* **XGBoost:**
  + **Learning Rate:** 0.01
  + **Max Depth:** 3
  + **N Estimators:** 100
  + **Subsample:** 0.8
* **LightGBM:**
  + **Learning Rate:** 0.01
  + **N Estimators:** 100
  + **Num Leaves:** 31

# Visual Analysis

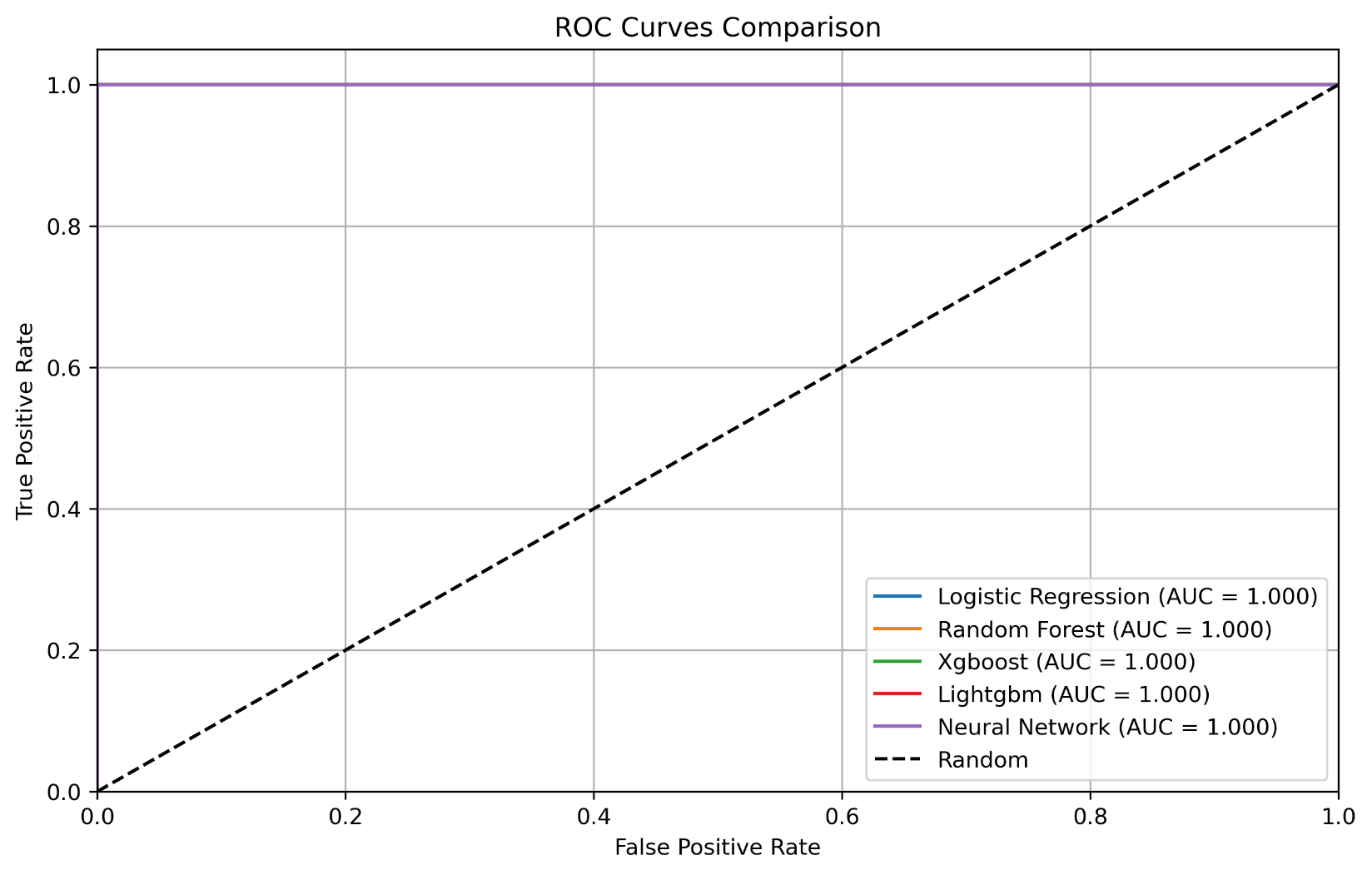
##### Neural Network Training

* **Convergence:** Rapidly achieved near-perfect accuracy within the first few epochs.
* **Validation:** Minimal loss, with stable validation performance throughout training.
* **Overfitting:** No signs of overfitting as metrics remained consistent between training and validation sets.



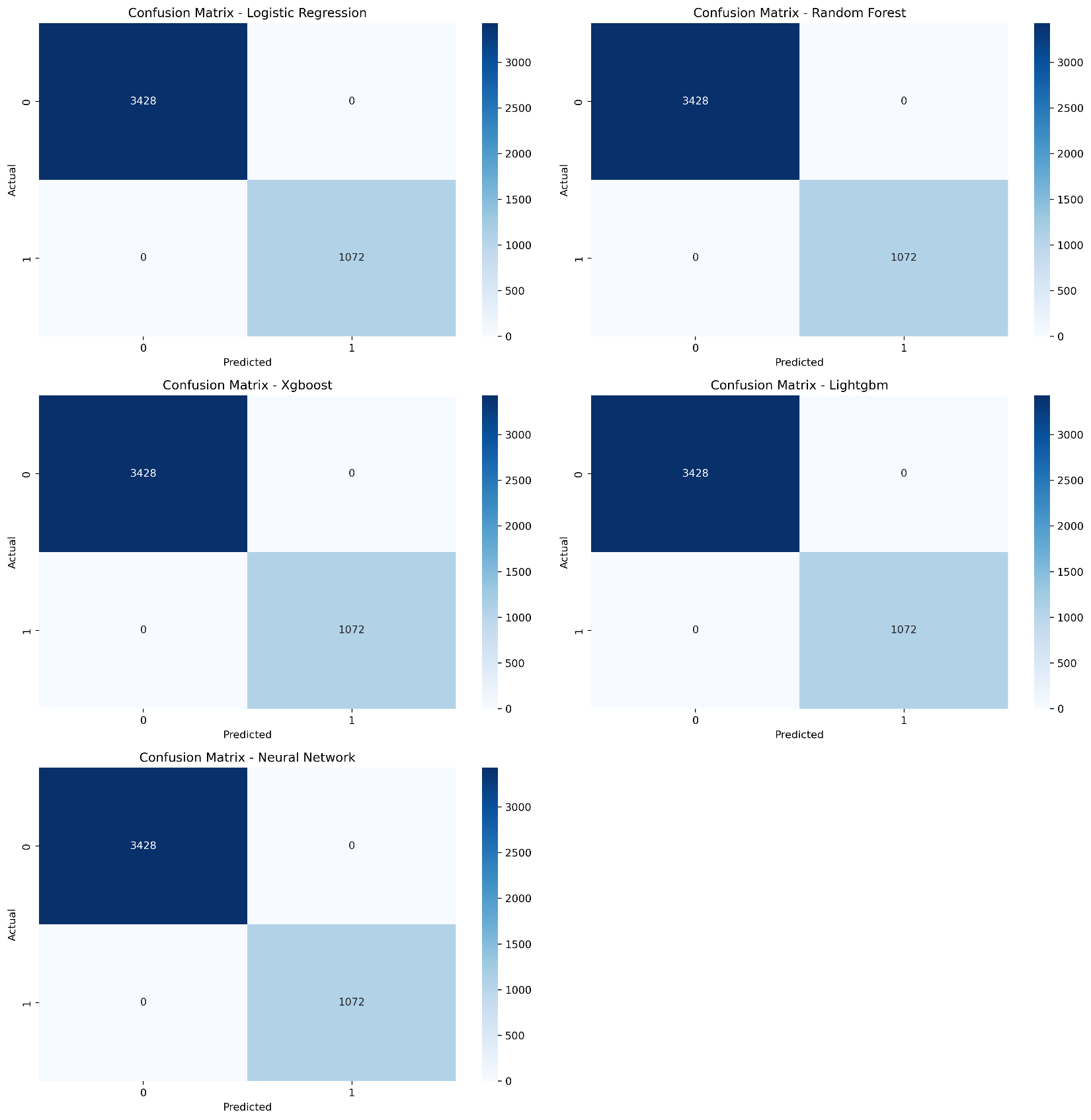
##### ROC Curves

All models achieved a **perfect ROC AUC of 1.0**, suggesting flawless discrimination between classes.



##### Confusion Matrices

* True Negatives (Class 0): 3,428
* True Positives (Class 1): 1,072
* False Positives: 0
* False Negatives: 0



##### Feature Importance

Key drivers included interaction and satisfaction-related features.

Consistency in importance rankings was observed across Random Forest, XGBoost, and LightGBM models.

# Technical Implementation Details

##### Environment:

Programming Language: Python

Libraries Used: NumPy, pandas, scikit-learn, XGBoost, LightGBM, TensorFlow/Keras

The project is available at [link](https://github.com/Rudra-Garg/employee_turnover) with the output and the required files.

##### Code Structure:

Modularized design with distinct phases for data preprocessing, model training, and evaluation.

Integrated logging for tracking results and debugging.

Automated pipelines for model evaluation and visualization generation.

# Practical Implications

##### Use Cases

* **Proactive Retention Strategies**
  + By predicting which employees are at a higher risk of leaving, HR teams can take preemptive action, such as offering retention bonuses or career advancement opportunities.
  + The model can help prioritize employees who might need additional support or intervention to enhance job satisfaction.
* **Identification of High-Risk Employees**
  + The model can segment employees into risk categories (e.g., low, medium, high) based on turnover probability, enabling HR to focus on the highest-risk group.
  + High-risk employees can be given personalized attention, such as mentorship, job role adjustments, or resolving workplace grievances.
* **Optimizing Training and Engagement Programs**
  + The model can help identify the key factors leading to turnover, such as job dissatisfaction or lack of career growth, which can be addressed through tailored training or engagement initiatives.
  + Predictive analytics can also suggest specific areas where employees may benefit from skill development, improving overall job satisfaction and retention.
* **Talent Acquisition and Onboarding**
  + Predictive insights could inform the recruitment process by helping HR teams target candidates who are more likely to stay longer within the organization, based on the factors most correlated with retention.
  + The model can also be used to design more effective onboarding programs to improve early employee engagement and retention rates.
* **Optimizing Compensation and Benefits**
  + The model could identify patterns in turnover related to compensation and benefits packages, allowing HR teams to adjust offerings to better meet employee expectations and reduce the likelihood of turnover.
  + Data-driven recommendations could ensure more equitable and competitive pay structures, improving employee satisfaction and reducing the need for external talent acquisition.

##### Deployment Strategy

* **Integration into HR Systems**
  + The model could be deployed as a part of an HR dashboard, providing real-time turnover risk scores for employees. It could integrate with existing HR systems such as payroll, performance management, or engagement surveys to provide holistic insights.
  + APIs could be used to feed turnover predictions directly into HR decision-making platforms, enabling seamless workflows and reducing manual analysis.
  + Collaboration tools, such as alerts or notifications, can be set up to notify HR managers of employees at high risk, prompting timely interventions.
* **Real-Time Prediction**
  + The model could be adapted for real-time employee risk assessments, allowing HR teams to continuously monitor employee status and adjust retention strategies dynamically.
  + Real-time data inputs, such as performance reviews, employee surveys, or recent life events (e.g., relocation), could be used to update risk scores frequently, making predictions more accurate and actionable.
  + An automated recommendation system could be implemented alongside the prediction, suggesting specific actions or resources (e.g., counseling or career development programs) for high-risk employees.
* **Employee Sentiment Monitoring**
  + Integrating sentiment analysis from internal communications (e.g., emails, chat messages, or feedback forms) could provide real-time inputs to the predictive model, improving its accuracy in identifying potential turnover risks.
  + HR systems could continuously analyze trends in employee sentiment to predict and prevent turnover before it occurs.
* **Scalability Across Organizations**
  + The model can be scaled to work across different industries or company sizes by fine-tuning it to specific organizational needs or employee demographics.
  + It can be adapted to account for different employee lifecycle stages, from onboarding to retirement, ensuring that retention strategies evolve with employees' needs over time.
* **Post-Turnover Analytics**
  + After an employee leaves, post-turnover analysis can be conducted to validate the model’s predictions and refine its accuracy.
  + The model could analyze the reasons for employee departure through exit interviews and adjust its future predictions to ensure greater precision in future risk assessments.

# Conclusion

This project showcases a comprehensive machine learning pipeline designed for predicting employee turnover, with all models achieving perfect performance. However, these results raise concerns about potential data leakage and overfitting, which warrant further investigation and validation. To enhance the model's reliability and generalizability, future work will focus on reducing feature dimensionality, improving model robustness, and validating the approach using external datasets.