PREDICTING EMPLOYEE ATTRITION

IT 461 | 56365

OVERVIEW

- Introduction
- Related works
- Data
- Methods
- Experiments
- Results and Discussions
- Conclusion
- Live testing

INTRODUCTION

Employee attrition poses major challenges for organizations, affecting productivity and finances. Traditional methods for identifying at-risk employees are often subjective and reactive. Machine learning offers a proactive solution by analyzing historical employee data to uncover patterns and predict attrition more accurately. This project aims to use demographic, performance, and job history data to assess the likelihood of an employee leaving.

RELATED WORKS

1. Deep Learning for Employee Attrition Prediction

- Method: ML + DL (FNN reached 97.5% accuracy)
- Takeaway: Deep learning (esp. FNNs) shows high potential
- Use for us: Consider implementing FNNs for comparison

2. ML Algorithms for Attrition Prediction

- Models: Logistic Regression, Decision Tree, RF, SVM
- Outcome: Random Forest performed best
- Use for us: Strong case for ensemble methods; key features like job satisfaction, salary

RELATED WORKS

3. Predicting Employee Attrition Using ML

- Models: Similar ML models (RF = 85% accuracy)
- Insights: Monthly income, age, job level were most important
- Use for us: Guides our feature engineering

4. SVM and ANN for Attrition Prediction

- Models: SVM, ANN, Decision Trees
- Result: SVM performed best (88.87% accuracy)
- Use for us: Valuable preprocessing & tuning methods to adopt

RELATED WORKS

Key Insights from Related Work:

ML & DL are both effective for attrition prediction

- Deep Learning (e.g., FNNs) offers high accuracy
- Traditional ML (e.g., SVM, RF) is more interpretable and efficient
- SVM & RF balance performance with interpretability
- Neural Networks uncover complex patterns, but require more computation
- Our approach builds on prior work by:
  - Improving feature selection
  - Enhancing performance on imbalanced data
  - Leveraging deep learning architecture

DATASET

- **Dataset Description:**

The dataset is sourced from **Kaggle** and includes HR records used to predict employee attrition. Each row represents a different employee, and columns contain attributes such as demographics, job satisfaction, and compensation. It contains **1,470** samples and **35** features.

- **Target Variable:**

Attrition: Indicates if the employee left the company **("Yes")** or stayed **("No")**, suitable for binary classification.

DATA PROCESSING & PREPARATION

Key Steps:

Data Included: Demographics, job roles, compensation, and satisfaction metrics

Preprocessing:

- Removed irrelevant features: (EmployeeCount, EmployeeNumber, Over18, PerformanceRating and StandardHours)
- Encoded categories: to ensure Attrition is categorical (0 or 1)
- Handled duplicates/outliers
- Scaled numerics with StandardScaler
- Applied SMOTE for class imbalance

EDA Insights:

- Found strong correlations between **attrition** and variables like **job satisfaction**, **overtime**, and **monthly income**
- Identified class imbalance

DATASET

Feature Engineering (Enhancing Data Quality)

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

- **Simplified categorical variables**
- **Dropped weak or redundant features**
- **Created new, more informative features**
- These enhancements contributed to reducing noise, improving model interpretability, and preventing overfitting

METHODS-FEATURE ENGINEERING

1. Label Encoding

All categorical variables were transformed into numerical format using Label Encoding, ensuring compatibility with machine learning models that require numerical input.

2. Dimensionality Reduction using Linear Discriminant Analysis (LDA)

We applied LDA with 1 component to identify and retain the most discriminative features for the binary classification task (Attrition: Yes/No). This helped to:

- Reduce feature space dimensionality
- Improve training efficiency
- Highlight features with the highest discriminatory power

METHODS-FEATURE ENGINEERING

3. Feature Selection Based on Importance

Feature importance was calculated using the absolute values of the LDA coefficients. We selected only features with an importance value greater than 0.1, resulting in a refined set of high-impact features such as:

- Job Involvement
- Education Field (Other, Medical, Life Sciences)
- Total Working Years
- Age
- Environment Satisfaction

METHODS-MACHINE LEARNING MODELS

1- Baseline Model: Logistic Regression

- Why used: Simple, interpretable, and ideal for binary classification.
- Performance:
  - Accuracy: 0.77
  - Recall: 0.69
  - ROC-AUC: 0.78

- Justification: High recall made it useful for identifying most attrition cases, which is crucial in real HR scenarios where missing an at-risk employee is more costly than a false positive.

METHODS-MACHINE LEARNING MODELS

2- Advanced Models Evaluated

📊 Model Performance Summary:

	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

- **Justification:**

Best Candidates for Stacking:

we typically want diverse yet strong models, ideally one with:

🔁 High Recall — Logistic Regression (to catch "Attrition" cases)

🧠 High Precision — SVC or LightGBM (to reduce false positives)

🎯 Good overall ROC-AUC — LightGBM & Logistic Regression

METHODS-MACHINE LEARNING MODELS

3- Ensemble Model: Stacking

- Why Stacking? Combines strengths of multiple models.
- Base Learners: Logistic Regression (recall-focused), LightGBM (precision-focused)
- Meta-Learner: Logistic Regression
- Benefits:
  - Balances high recall and high precision
  - Improves generalization and stability
  - Reduces overfitting by combining model perspectives

EXPERIMENTS

1-Experiment Workflow:

- Preprocessing: Label Encoding, LDA-based Feature Selection
- Selected Features: Job Involvement, Education Field, Total Working Years, Age, Environment Satisfaction,etc
- Dataset Split: 80% Train | 20% Test
- Models Used: Logistic Regression, KNN, SVM, RF, GB, AdaBoost, XGBoost, LightGBM
- Final Model: Stacking Ensemble (LogReg + LightGBM → LogReg)

EXPERIMENTS

2-Hyperparameter Tuning & Threshold Adjustment:

- Tuning with RandomizedSearchCV & GridSearchCV
- Final Best Params:
  - LogReg: C = 10
  - LightGBM: learning_rate = 0.01, n_estimators = 200
  - Final Estimator: C = 10

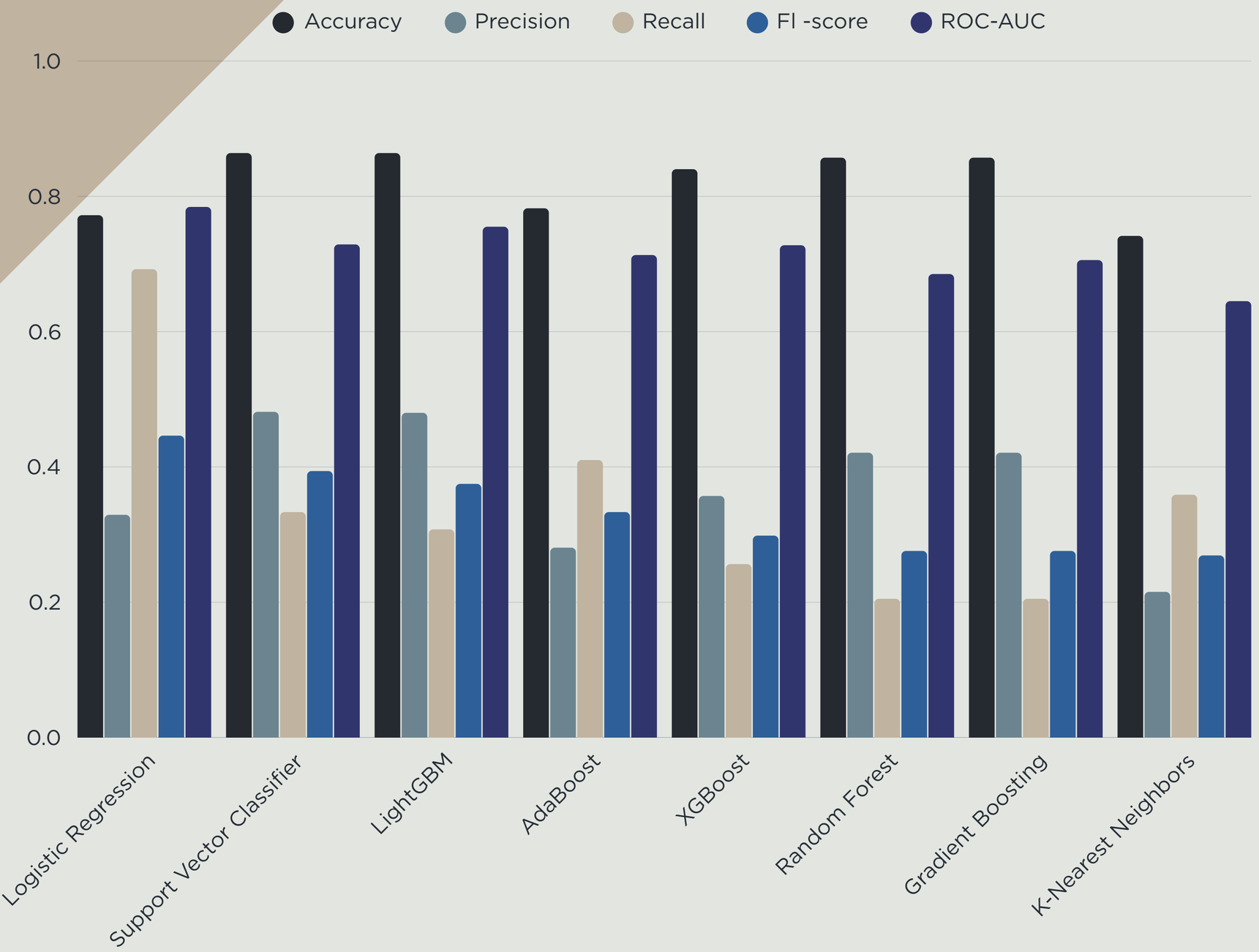
- Class Imbalance Handling:
  - Adjusted threshold to 0.2–0.4
  - Improved recall and F1-score for minority class

EXPERIMENTS

3-Final Model Evaluation

- Accuracy: 0.89
- Precision (Class 1): 0.63
- Recall (Class 1): 0.44
- F1-Score (Class 1): 0.52
- ROC-AUC: 0.8038
- Conclusion: Ensemble outperformed all individual models

RESULTS



- **Best Accuracy**: SVM & LightGBM (~86.4%)
- **Best Recall**: Logistic Regression (69.2%)
- **Best F1-score**: Logistic Regression
- **Weakest Model**: K-Nearest Neighbors (low recall & precision)

TUNED MODEL – RESULTS

Model Used:
Stacking Classifier
(Base Models: Logistic Regression & LightGBM | Final Estimator: Logistic Regression)
Classification Report:

- Class 0 (Non-Attrition):
  - Precision: 0.92 | Recall: 0.96 | F1-Score: 0.94
- Class 1 (Attrition):
  - Precision: 0.63 | Recall: 0.44 | F1-Score: 0.52

TUNED MODEL – RESULTS

Performance Ratings:

- Accuracy: Excellent (89%)
- Precision (Macro): Good (77%)
- Recall (Macro): Good (70%)
- Weighted Avg Precision: Excellent (88%)

CONCLUSION & INSIGHTS

- The Stacking Classifier demonstrated strong overall performance, with 89% accuracy on the test set.
- It performs exceptionally well in identifying non-attrition cases (Class 0) with high precision and recall.
- For attrition cases (Class 1), recall improved to 0.44 and precision reached 0.63 — a meaningful step toward minimizing false negatives.
- ROC-AUC of 0.8038 shows the model distinguishes well between classes, despite class imbalance.

CONCLUSION & INSIGHTS

Challenges:

- **Class Imbalance:** A significant skew toward non-attrition cases was addressed using SMOTE to balance the dataset.

Future Directions

- NexImprove Class 1 metrics (recall and F1) further by exploring:
  - More granular feature engineering
  - Cost-sensitive learning
  - Additional ensemble techniques

LIVE TESTING

✅ Predictions from the 3 models on 5 test samples:

	True Label	Logistic Prediction	LightGBM Prediction	Stacking Prediction
464	0	0	0	0
49	0	0	0	0
332	0	0	0	0
1114	0	1	0	0
886	0	0	0	0

LIVE TESTING

- All models performed correctly on 4 out of 5.
- Logistic regression misclassified sample 1114 — predicted 1 instead of 0.
- LightGBM and stacking were both correct.

Thank You

For your attention

Prepared by:

Raghad Almutairi, 443200793
Noura Alwohaibi, 443200415
Alanoud Alamri, 443200043
Lujain Alharbi, 443200811