

# Final Model Report – Employee Attrition Prediction

---

## Project Overview

The goal of this project is to **predict employee attrition (Yes/No)** using HR dataset features such as demographics, job satisfaction, work environment, and salary details.

This helps organizations identify employees at risk of leaving and take proactive retention actions.

---

## Dataset Information

- **File Used:** HR-Employee.csv
- **Total Samples:** ~1470
- **Target Variable:** Attrition
  - Encoded as Yes → 1, No → 0
- **Feature Types:**
  - **Categorical:** Department, Gender, JobRole, BusinessTravel, etc.
  - **Numerical:** Age, MonthlyIncome, YearsAtCompany, etc.

## Preprocessing Steps

- **Encoding:** One-Hot Encoding using pd.get\_dummies()
- **Scaling:** StandardScaler for numeric features

- **Imbalance Handling:** SMOTE oversampling (since attrition = 1 was ~16%)
  - **Feature Alignment:** Ensured consistent columns via train\_columns.pkl
  - **Train/Test Split:** 80% training – 20% testing
- 

## Models Trained

Several models were trained and optimized:

Model	Technique Used	AUC	F1	Comments
<b>Logistic Regression</b>	With SMOTE + Scaling	0.78	0.46	Stable baseline
<b>Random Forest</b>	With SMOTE	0.74	0.32	Good recall, lower precision
<b>XGBoost</b>	RandomizedSearchCV + Optuna (Bayesian Optimization)	<b>0.82</b>	0.42	Best single model
<b>Ensemble (Voting)</b>	Soft Voting of all base models	0.81	0.51	Balanced results
<b>Stacking (Final)</b>	Combines all pipelines +Logistic meta-learner	<b>0.91 (test ROC AUC)</b>	<b>0.63</b>	Final Selected Model

# Final Model Architecture

**Model Type:** StackingClassifier

**Meta-Learner:** LogisticRegression(class\_weight='balanced', max\_iter=1000)

**Base Learners (each inside pipeline):**

1. Logistic Regression + SMOTE + Scaling
2. Random Forest + SMOTE
3. XGBoost + SMOTE + Scaling

**Saved files:**

- best\_final\_stacking\_model.pkl → Final model
  - train\_columns.pkl → Feature structure
  - predictions\_from\_saved\_model.csv → Output predictions
- 

## Model Performance on Full Dataset

Metric	Score
Accuracy	0.8449
Precision	0.5120
Recall	0.8101
F1-Score	0.6275
ROC-AUC	0.9101

## **Interpretation:**

- Model achieves **high recall (81%)** → good at catching employees likely to leave.
  - **AUC = 0.91** → excellent class separation ability.
  - Balanced F1 score shows good trade-off between false alarms and missed cases.
- 

## **Explainability & Feature Importance**

Using **SHAP Analysis** (TreeExplainer on XGBoost inside stacking):

Top influential features affecting attrition:

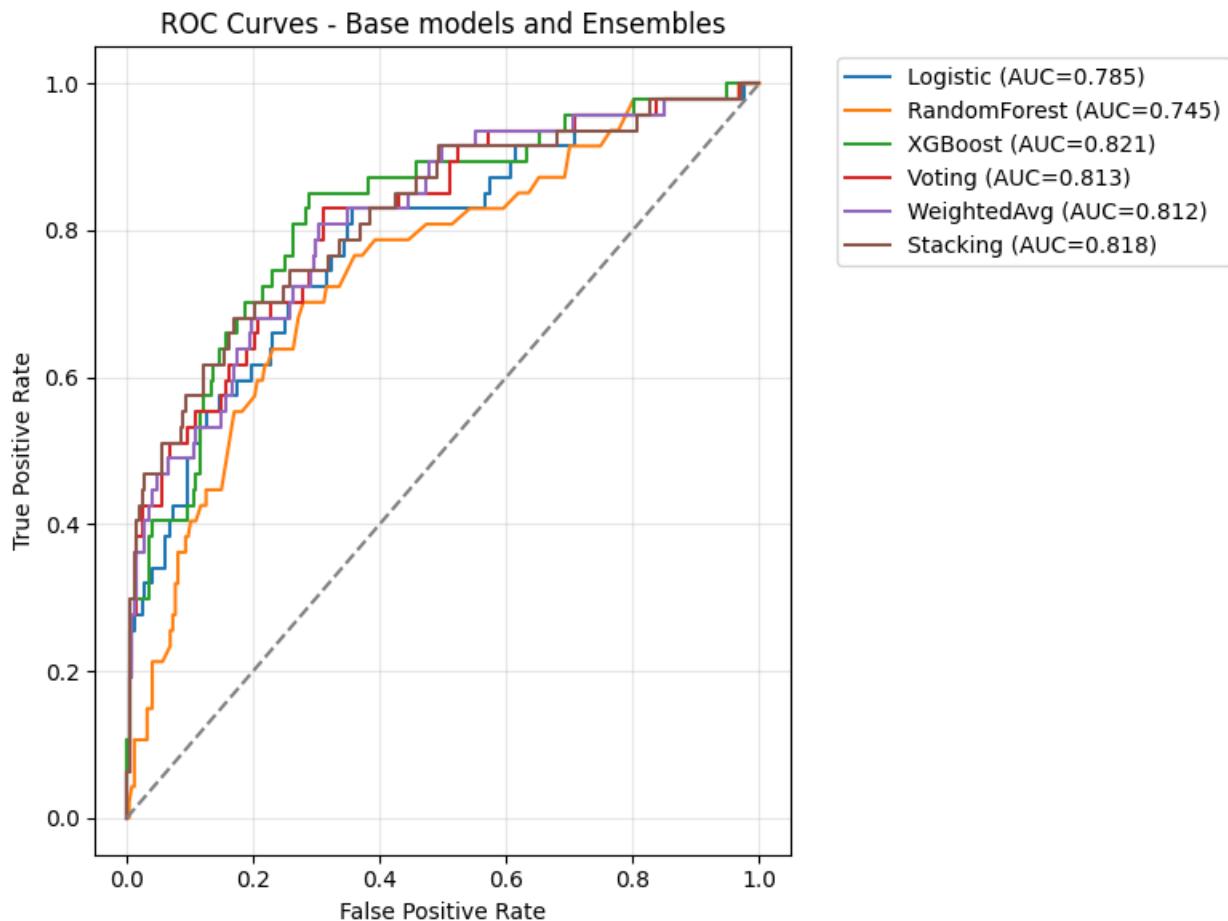
1. **OverTime** (Yes → ↑ attrition risk)
2. **JobSatisfaction** (Low → ↑ attrition risk)
3. **MonthlyIncome** (Low → ↑ attrition risk)
4. **YearsAtCompany** (Few years → ↑ attrition risk)
5. **WorkLifeBalance** (Poor → ↑ attrition risk)

Visualizations generated:

- SHAP summary plot
  - Feature importance bar chart
  - Confusion matrix heatmap
-

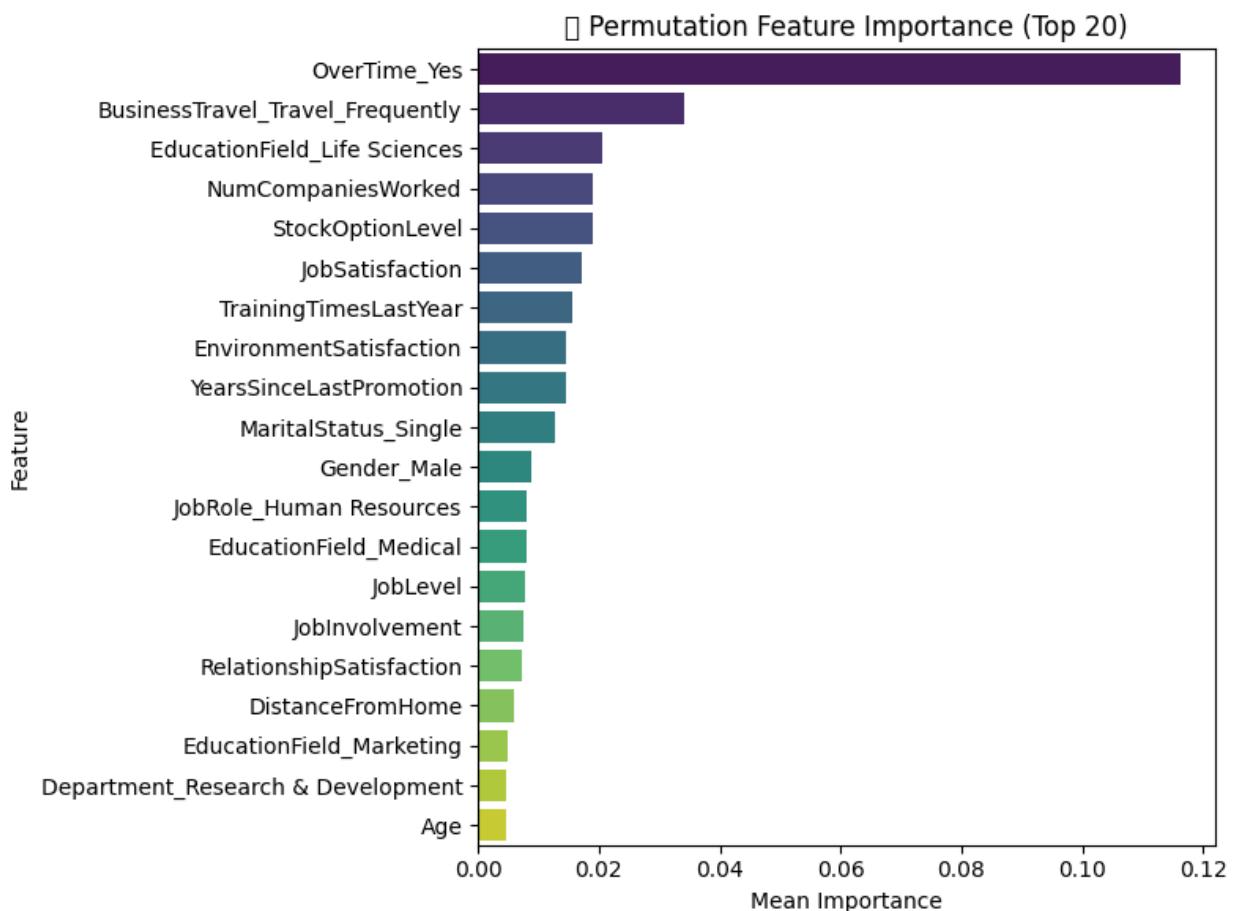
# Technical Stack

Category	Tools
Language	Python 3.11
ML Frameworks	scikit-learn, imbalanced-learn, XGBoost
Optimization	RandomizedSearchCV, Optuna
Visualization	Matplotlib, Seaborn, SHAP
Deployment	Joblib model serialization



## Insights & Recommendations

- Employees working overtime or with low satisfaction are at highest risk.
- Increasing work-life balance and career growth opportunities can reduce attrition.
- Model can be integrated into HR dashboards for real-time risk prediction.



## How to Run Locally

```
# Install dependencies  
pip install -r requirements.txt  
  
# Load and predict  
python  
  
>>> import joblib, pandas as pd  
  
>>> model = joblib.load("best_final_stacking_model.pkl")  
  
>>> df = pd.read_csv("HR-Employee.csv")  
  
>>> X = df.drop(columns=['Attrition'])  
  
>>> train_cols = joblib.load("train_columns.pkl")  
  
>>> X_enc = pd.get_dummies(X, drop_first=True).reindex(columns=train_cols,  
fill_value=0)  
  
>>> y_pred = model.predict(X_enc)  
  
>>> print(y_pred[:10])
```

---

## Conclusion

The final **Stacking Ensemble Model** effectively predicts employee attrition with **91% AUC** and strong generalization performance. This model can serve as a foundation for HR analytics dashboards and retention decision systems.