ML Report: Job Change Prediction

Summary of Findings

This analysis compares various classification models (Random Forest, Gradient Boosting, Logistic Regression, etc.) on datasets treated with different sampling strategies - specifically baseline (no balancing) and SMOTE (Synthetic Minority Oversampling Technique). SMOTE significantly improved model performance, increasing average ROC-AUC from 0.7395 to 0.8536. Among all models, Random Forest trained on SMOTE-enhanced data emerged as the best performer, achieving a mean ROC-AUC of 0.9297 and F1-Score of 0.8603 across cross-validation. This combination not only handles class imbalance effectively but also balances precision and recall, making it suitable for identifying at-risk employees who may change jobs. The final validation set performance slightly dipped (ROC-AUC 0.7974, F1 0.5534), likely due to natural variance or overfitting mitigation.

Metric Explanations and Selection Rationale

* Accuracy = (TP + TN) / (TP + TN + FP + FN) - Not ideal for imbalanced data.
* Precision = TP / (TP + FP) - Indicates how many predicted positives are true.
* Recall = TP / (TP + FN) - Measures how many actual positives are correctly predicted.
* F1-Score = 2 * (Precision * Recall) / (Precision + Recall) - Balances both precision and recall.
* ROC-AUC - Measures ability to distinguish between classes across thresholds.

For this task, F1-Score and ROC-AUC are most suitable: F1 captures a good trade-off between precision and recall, while ROC-AUC assesses overall model separability.

Model & Dataset Recommendation

Best Model: Random Forest
Best Dataset: SMOTE-treated (oversampled) data

Random Forest with SMOTE achieved the highest performance with ROC-AUC = 0.9297 and F1 = 0.8603. It handles non-linearities, class imbalance, and provides interpretability through SHAP.

Deployment Strategy

Deploy the model as a microservice using FastAPI or Flask in a Docker container. Preprocessing steps must be serialized and consistently applied. Host the service on GCP Cloud Run or AWS Lambda.

Expose the model as a REST API for integration into HR dashboards. Use thresholds to flag employees at risk and SHAP for explanations. Regularly monitor and retrain to address drift.