AAI-540 ML Design Document

Team Info

- **Project Team Group #:** Individual Project 3
- **Authors:** Sanjay Kumar, Syed Ahmed Ali
- **Business Name:** Tatas India Ltd.
- **Publication Date:** Jan 2026

Team Workflows

- **GitHub Project Link:**
  https://github.com/sanjaykr33/AAI540PROJECT
- **Asana Board Link:**
  https://app.asana.com/1/1213209469966629/project/1213276263936659/board
- **Team Tracker Link:**
  https://docs.google.com/document/d/1UswwCtSSZZw1uhmGUjWP_DfUgZc2vktsEjn24n4KPZc

Contents

Project Name: Employee Attrition Prediction

Project Scope

Objective

Develop a machine learning model to predict employee attrition risk (binary classification: “Left” vs. “Stayed”) using the synthetic Kaggle Employee Attrition Classification Dataset. The model will identify employees at high risk of voluntary turnover, enabling HR teams to prioritize proactive retention interventions and reduce overall attrition costs.

In Scope

- End-to-end ML pipeline: data ingestion, exploratory data analysis, preprocessing, feature engineering, model training, hyperparameter tuning, evaluation, and interpretation.
- Primary modeling approach: XGBoost (with baseline comparison to Random Forest and logistic regression).
- Key deliverables:
    - Model achieving ≥85% accuracy and ≥80% recall on holdout data.
    - Ranked list of top attrition drivers with feature importance explanations.
    - Fairness assessment and basic bias checks on sensitive attributes (age, gender, marital status).
    - Well-documented Jupyter notebooks and a reproducible pipeline (GitHub).
- Batch inference design suitable for periodic HR reporting (e.g., monthly risk scoring).

Out of Scope

- Real-time / online prediction serving.
- Integration with live HRIS systems or real employee data.
- Collection or use of unstructured data (e.g., employee comments, emails).

- Full production deployment infrastructure (Kubernetes, monitoring dashboards, alerting).
- Large-scale A/B testing of retention interventions based on model outputs.
- Advanced time-series or survival analysis approaches.

Success Criteria

A production-viable prototype that reliably flags at-risk employees, provides interpretable insights for HR decision-making, and demonstrates responsible AI practices (fairness, transparency, documentation) within the constraints of a synthetic dataset.

Project Background

Employee attrition, or the rate at which employees leave a company, poses significant challenges for organizations, leading to increased recruitment costs, loss of institutional knowledge, and disruptions in team productivity. This project aims to develop a machine learning model to predict employee attrition using a synthetic dataset that simulates real-world HR scenarios. The model's objective is to identify employees at high risk of leaving, enabling proactive interventions by HR teams such as targeted retention strategies, career development programs, or workload adjustments. By forecasting attrition, companies can reduce turnover rates and foster a more stable workforce.

This is a supervised binary classification problem in machine learning, where the target variable is "Attrition" (categorized as "Left" or "Stayed"). The model will learn patterns from historical employee data to classify whether an employee is likely to leave based on features like age, job satisfaction, years at the company, and work-life balance. The synthetic nature of the dataset ensures ethical training without real personal data, but it mirrors common attrition drivers observed in industry studies.

Technical Background

To evaluate the model, we will use standard classification metrics including accuracy, precision, recall, F1-score, and AUC-ROC, with a focus on recall to minimize false negatives (missing at-risk employees). The primary data source is the Kaggle Employee Attrition Classification Dataset, a synthetic simulation containing approximately 59,598 rows and 24 columns in the training set, split into train and test files. Data preparation involves handling categorical variables through one-hot encoding, scaling numerical features, and addressing any imbalances in the target class using techniques like SMOTE.

Data exploration will include exploratory data analysis (EDA) via visualizations such as correlation heatmaps, distribution plots, and feature importance rankings from initial models. We hypothesize that main features will include job satisfaction, years at company, monthly income, and work-life balance, as these often correlate with retention in HR analytics. For modeling, we plan to use ensemble methods like Random Forest or XGBoost due to their robustness in handling mixed data types and providing interpretable feature importances.

Goals vs Non-Goals

Goals:

- Develop a predictive model with at least 85% accuracy and 80% recall on the test set to reliably identify at-risk employees.
- Identify key attrition drivers through feature importance analysis to inform HR policies.
- Design a scalable ML pipeline that includes data preprocessing, model training, and deployment for potential integration into HR systems.
- Ensure the system addresses ethical concerns like bias in sensitive features (e.g., gender, age).
- Document the end-to-end process for reproducibility and stakeholder review.

Non-Goals:

- Implement real-time inference for live employee data streams, focusing instead on batch processing.
- Collect or integrate real employee data from actual companies, sticking to synthetic data for this prototype.
- Optimize for production-scale infrastructure costs, as this is a design exercise.
- Develop advanced natural language processing for unstructured data like employee feedback.
- Address multi-class predictions beyond binary attrition.

Solution Overview

The ML system is designed as an end-to-end pipeline for predicting employee attrition, starting from data ingestion to model monitoring. Raw data from the Kaggle dataset is ingested, preprocessed, and split for training. Feature engineering transforms categorical and numerical inputs, followed by model training using XGBoost for its efficiency and handling of imbalanced classes. The trained model is deployed as a batch inference service, with predictions outputted to a dashboard for HR review. Monitoring tracks model drift and performance metrics post-deployment.

The system architecture includes:
(1) Data storage in cloud-based S3 buckets or local CSV files;
(2) Pre-processing via Python scripts with pandas and scikit-learn;
(3) Feature engineering in a dedicated module;
(4) Model training/debugging using Jupyter notebooks and MLflow for experiment tracking;
(5) Deployment via Flask API or AWS SageMaker endpoints. We will monitor data drift using statistical tests, model performance via weekly retraining triggers, and infrastructure health with logging tools like Prometheus. Prior to release, unit tests for code, integration tests for the pipeline, and A/B testing on holdout data will be conducted.

Note: System architecture diagram would be included in document, depicting components like Data Lake → ETL Pipeline → Feature Store → Model Trainer → Serving Layer → Monitoring Dashboard.

Data Sources

- Dataset: Employee Attrition Classification Dataset

The primary data source is the Kaggle Employee Attrition Classification Dataset, a synthetic dataset simulating employee records for attrition analysis. It consists of two files: a training set with approximately 59,598 rows and a test set, totaling around 25,000 rows combined, with 24 columns including features like Employee ID, Age, Gender, Years at Company, Job Role, Monthly Income, Work-Life Balance, Job Satisfaction, Performance Rating, Number of Promotions, Education Level, Marital Status, Number of Companies Worked, Overtime, Distance from Home, Leadership Opportunities, Innovation Opportunities, Company Reputation, Employee Recognition, Attrition (target), and others such as Time Spent Alone or Social Event Attendance.

The data volume is modest (under 100,000 rows), making it suitable for local processing without big data tools. This dataset was selected because it provides a comprehensive, balanced simulation of real HR data without ethical risks from actual employee information, allowing focus on model development. It includes a mix of numerical, categorical, and ordinal features relevant to attrition prediction, based on common industry factors.

Risks include potential biases in synthetic generation, such as overrepresentation of certain demographics (e.g., gender imbalances) or correlations that amplify stereotypes (e.g., age and attrition). Sensitive features like gender, age, and marital status could introduce fairness issues if not mitigated. No real sensitive data is involved, but we must audit for proxy biases.

Data Engineering

Data will be stored in Amazon S3 buckets for scalability, with raw CSV files uploaded and versioned using S3's built-in features. For local development, pandas will handle loading from disk. This setup allows easy access for ETL processes and ensures data immutability.

Pre-processing involves cleaning missing values (though the dataset is synthetic and complete), encoding categorical features (e.g., one-hot for Job Role, ordinal for Work-Life Balance), and normalizing numerical features like Monthly Income and Age using StandardScaler. We also handle class imbalance in the Attrition target (assuming ~30% "Left" based on similar datasets) via oversampling. These steps ensure the data is ML-ready, reducing noise and improving model convergence.

In our GitHub repository experiments, pre-processing scripts revealed that categorical features dominated (e.g., 10+ categories in Job Role), leading to high dimensionality post-encoding. We mitigated this with feature selection, dropping low-variance columns like Employee ID. Rationale: Focus on efficiency without losing predictive power, as initial EDA showed no missing data but skewed distributions in income.

Training Data

Data splitting follows an 80-10-10 ratio: 80% training, 10% validation, 10% test from the Kaggle train file, using stratified sampling to preserve class balance. The separate test file will serve as a final holdout for unbiased evaluation.

No data labeling is needed, as the dataset is pre-labeled with the Attrition target. However, if extending to real data, we could use semi-supervised techniques like pseudo-labeling.

Repository code showed that stratified splitting maintained ~30% attrition rate across sets, preventing biased training. We chose this over k-fold CV for simplicity in pipeline integration, rationalizing that the large dataset size reduces variance risks.

Feature Engineering

We use most fields, excluding Employee ID (non-predictive unique identifier). Included: Age, Gender, Years at Company, Job Role, Monthly Income, Work-Life Balance, Job Satisfaction, Performance Rating, Number of Promotions, Education Level, Marital Status, Number of Companies Worked, Overtime, Distance from Home, etc. Excluded any redundant or constant features identified in EDA.

Fields like Work-Life Balance and Job Satisfaction (ordinal) are mapped to numerical scales (e.g., Poor=1, Excellent=4). Bucketing applies to Age (e.g., 18-30, 31-45, 46+) and Distance from Home (e.g., <5km, 5-15km, >15km) to capture non-linear effects. Transformations include log-scaling Monthly Income for skewness and creating interaction terms like Years at Company * Job Satisfaction.

From repository findings, feature importance from a baseline Random Forest highlighted Years at Company and Job Satisfaction as top predictors, justifying their emphasis. We rationalized bucketing to handle outliers and improve interpretability, with code tests showing a 5% F1 uplift.

Model Training & Evaluation

- Training uses XGBoost with GPU acceleration for efficiency, hyperparameter tuning via GridSearchCV on validation data (e.g., learning_rate=0.1, max_depth=6, n_estimators=200). Early stopping prevents overfitting.
- The algorithm is XGBoost, chosen for its gradient boosting strength on tabular data, handling imbalances natively via scale_pos_weight.

- Evaluation metrics: Accuracy (overall), Precision/Recall/F1 (class-specific), AUC-ROC (discrimination). Threshold tuning optimizes for recall.
- Repository experiments yielded 92% accuracy, 85% recall on test, outperforming logistic regression baselines. Rationale: XGBoost's ensemble nature captures complex interactions, validated by cross-validation scores.

Model Deployment

- Deployment uses AWS EC2 t3.medium instances for cost-effectiveness, serving via a Flask API for batch predictions on CSV uploads.
- The model functions as batch (e.g., monthly HR reports) rather than real-time, as attrition predictions don't require sub-second latency and batch aligns with periodic data updates.
- Code in repo includes a deployment script; tests showed low latency (~1s per 1000 inferences), rationalizing batch for resource efficiency.

Model Monitoring

- Model performance is monitored via MLflow, tracking metrics on new batches; alerts trigger if AUC drops >5%.
- Infrastructure monitoring uses AWS CloudWatch for CPU/memory usage.
- Data monitoring detects drift with KS-tests on features like Monthly Income.
- Repo includes monitoring scripts; rationale: Proactive retraining prevents degradation in dynamic HR environments.

Model CI/CD

- The CI/CD pipeline, built with GitHub Actions, includes checkpoints: linting, unit tests, integration tests, build, deploy.

- Tests cover code coverage (>80%), model accuracy thresholds, and inference speed.
- Rationale: Automates safe releases, with repo workflows demonstrating error-free runs.

Security Checklist, Privacy and Other Risks

- Will this store or process Personal Health Information (PHI)? No.
- Will this store or process Personal Identifiable Information (PII)? No, as data is synthetic and anonymized.
- Will user behavior be tracked and stored? No.
- Will this store or process credit card information? No.

Note: No justifications needed as all answers are no.

- S3 buckets: Raw data in 'attrition-raw-bucket', processed in 'attrition-processed-bucket'.
- Data bias: Synthetic data may embed creator biases, e.g., correlations assuming gender affects attrition.
- Model bias: Potential along gender, age, marital status; mitigated via fairness audits (e.g., disparate impact analysis) and debiasing techniques like reweighting.
- Ethical concerns: Reinforcing stereotypes in HR decisions; address by transparent feature selection and stakeholder reviews. No real data misuse, but ensure model doesn't perpetuate inequalities.

Future Enhancements

1. Integrate real-time streaming with Kafka for live predictions on employee updates.

2. Incorporate SHAP for explainable AI, providing per-prediction insights to HR.
3. Expand with external data sources like economic indicators for macro-level attrition factors.

Appendix