Customer Churn Prediction using AWS

This document provides implementation-level details and cloud infrastructure context for the churn risk modeling system described in the main README.

High-Level System Architecture

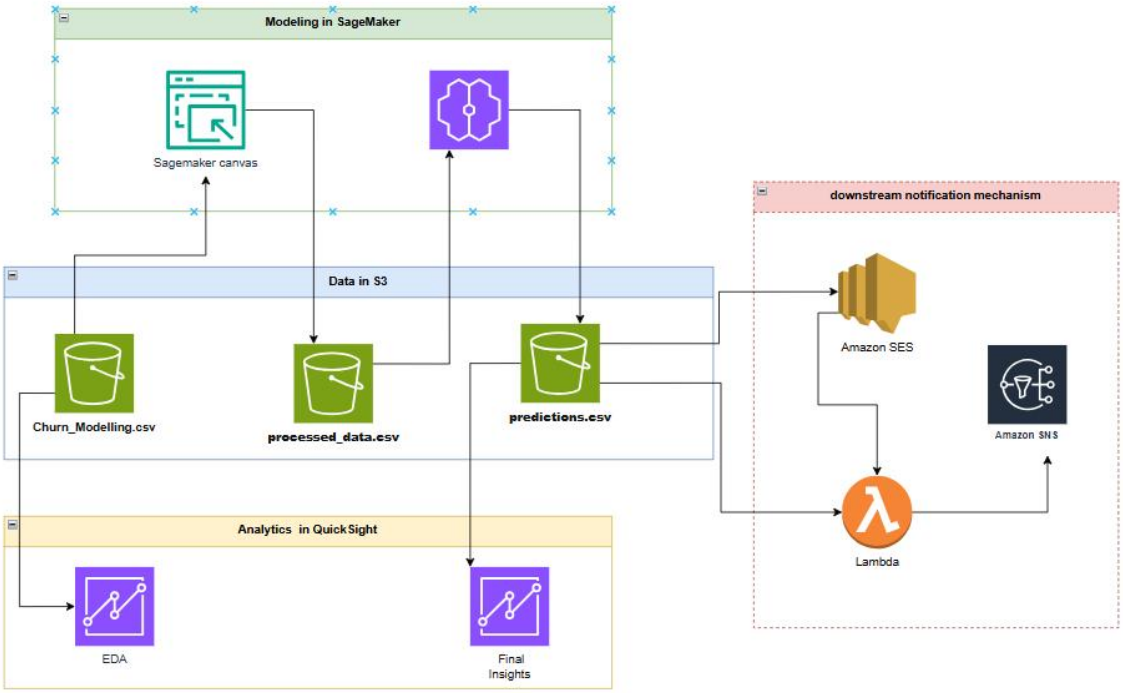


Figure 1: Architecture diagram

Core AWS Services Used

1. **AWS S3:** Data storage and management

2. **AWS SageMaker:** Data cleaning, preprocessing, model training and deployment

3. **AWS QuickSight:** Data visualization and dashboards

4. **AWS SNS:** Event-driven notifications

5. **AWS Lambda:** Automating custom emails

6. **AWS SES:** Configuring recipients and delivering emails

Dataset Overview

The dataset used in this project, sourced from Kaggle (Dataset), provides comprehensive information for customer churn prediction. It consists of **10,000 rows** and **14 columns**, capturing a variety of customer attributes relevant to predicting churn behavior. Below is a detailed description of the columns:

1. **RowNumber**: Sequential index of the rows, representing each customer.
2. **CustomerId**: Unique identifier for each customer.
3. **Surname**: Last name of the customer.
4. **CreditScore**: Credit score of the customer.
5. **Geography**: Country of residence (e.g., France, Spain, Germany).
6. **Gender**: Gender of the customer (Male/Female).
7. **Age**: Age of the customer.
8. **Tenure**: Number of years the customer has been with the bank.
9. **Balance**: Account balance of the customer.
10. **NumOfProducts**: Number of products the customer is using.
11. **HasCrCard**: Indicates if the customer has a credit card (1: Yes, 0: No).
12. **IsActiveMember**: Indicates if the customer is an active member (1: Yes, 0: No).
13. **EstimatedSalary**: Estimated annual salary of the customer.
14. **Exited**: Target variable indicating if the customer has churned (1: Yes, 0: No).

Key Features

- **Geography** and **Gender** provide demographic insights.
- **CreditScore**, **Age**, and **Balance** are important financial indicators.
- **NumOfProducts**, **HasCrCard**, and **IsActiveMember** reflect customer engagement and usage patterns.
- **Exited** serves as the primary label for churn prediction.

AWS S3 - Data Storage

Amazon S3 is used as the system of record for all data artifacts produced by the churn modeling workflow. This includes raw inputs, processed feature datasets, and model prediction outputs.

Using S3 as a centralized storage layer allows clear separation between data ingestion, modeling, and consumption. Raw and transformed datasets are versioned and stored independently, enabling reproducibility of experiments and clean hand-offs to downstream systems such as dashboards and alerting workflows.

Model outputs are written back to S3 in tabular form, allowing them to be consumed without direct dependency on notebooks or model code.

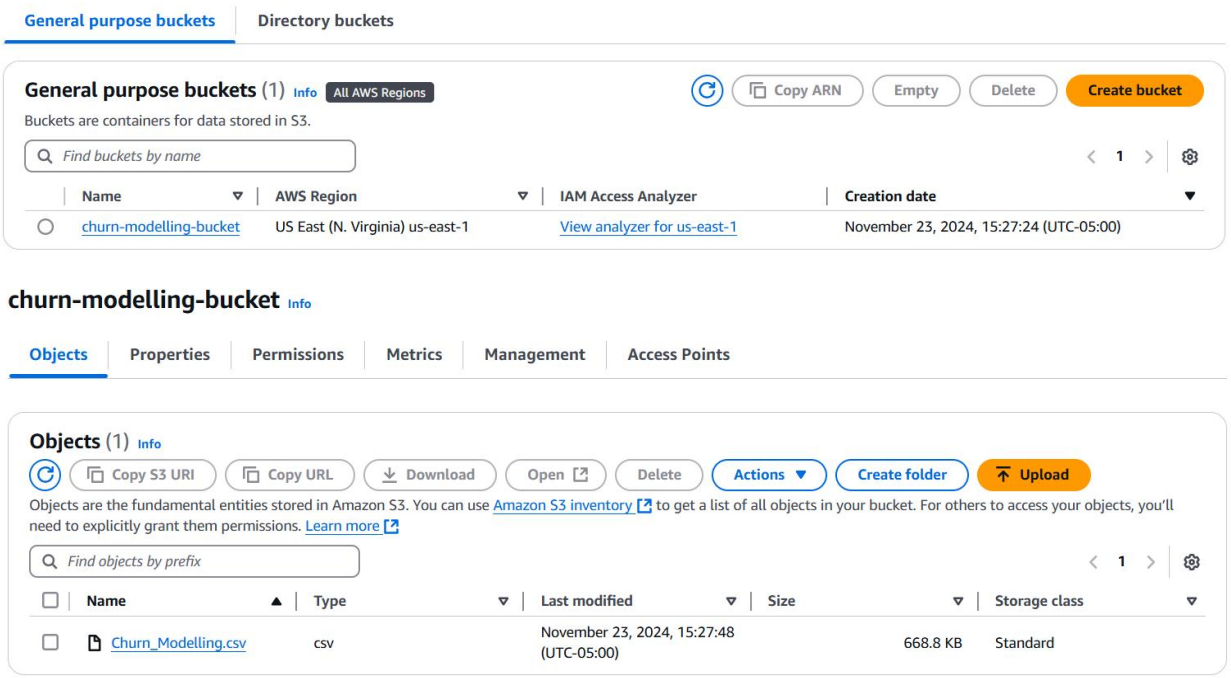


Figure 2: S3 bucket with "Churn_Modelling.csv"

AWS Quicksight - Exploratory Data Analysis (EDA)

Exploratory analysis was conducted using Amazon QuickSight to surface behavioral patterns associated with customer churn. Dashboards were built directly on top of data stored in S3, enabling interactive analysis without duplicating datasets or embedding logic in notebooks.

The analysis focused on understanding how churn rates vary across customer demographics, engagement levels, tenure, and balance tiers. Particular attention was

paid to identifying segments where churn risk meaningfully diverges from the population average, as these segments are more actionable than individual-level predictions.

Insights from this stage informed feature selection, segmentation logic, and downstream interpretation of model outputs rather than serving as static reports.

Figure 3: EDA in QuickSight

AWS SageMaker - Data Preprocessing and Feature Engineering

Data preprocessing and feature engineering were performed in SageMaker-managed notebook environments to ensure consistent execution and scalable compute.

Categorical variables were encoded into machine-readable formats, and numeric features were normalized where scale differences could impact model behavior. Feature transformations were applied selectively, with the goal of improving model stability rather than maximizing transformation complexity.

Skewed distributions were examined during preprocessing. For example, age exhibited right-skewness, and a log-based transformation was applied to reduce extreme influence and improve model learning behavior.

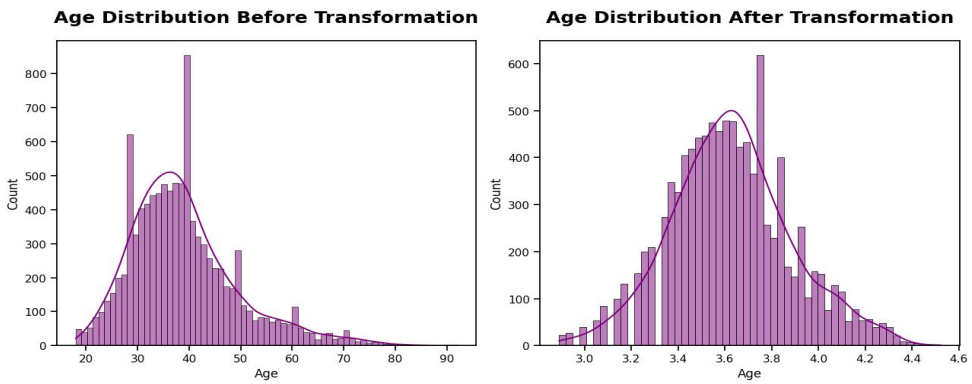


Figure 4: Age distribution before and after transformation

The preprocessing pipeline produces a clean, model-ready dataset that is exported back to S3 for downstream training and evaluation.

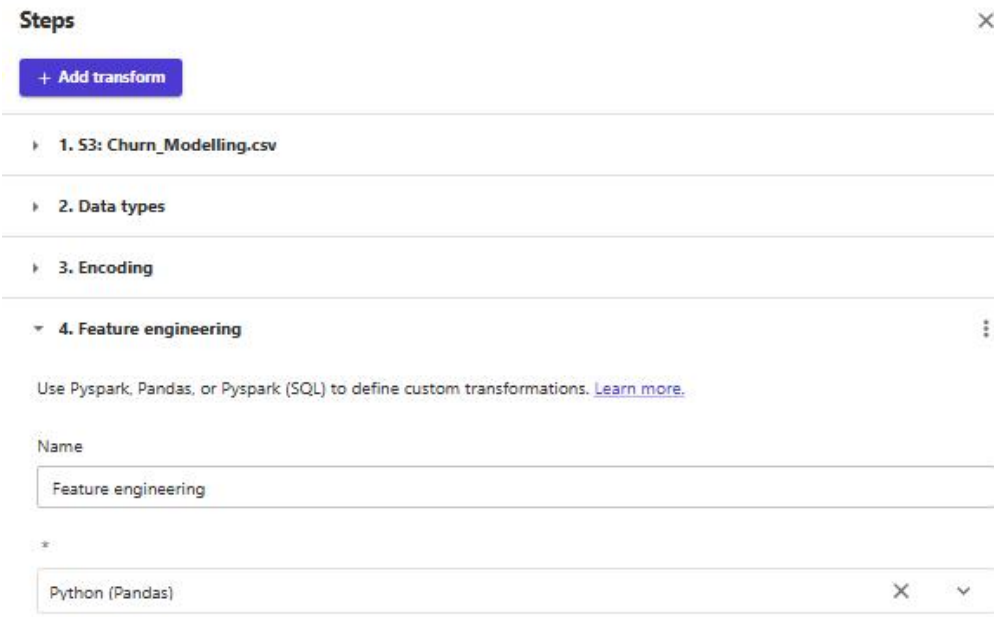


Figure 5: Data transformation in SagaMaker Canvas

Machine Learning Evaluation Results

Customer churn is modeled as a binary classification problem with a moderately imbalanced target distribution. Because both false negatives (missed at-risk

customers) and false positives (unnecessary interventions) carry cost, model evaluation emphasized balance rather than raw accuracy.

Multiple baseline and tree-based models were evaluated, including Logistic Regression, Support Vector Machines, K-Nearest Neighbors, Decision Trees, and Random Forests. Stratified train, validation, and test splits were used to preserve label balance across datasets.

F1-score was selected as the primary evaluation metric to balance precision and recall under class imbalance. Among the evaluated models, Random Forest provided the strongest overall tradeoff between performance, robustness, and interpretability, achieving an F1-score of approximately 0.83 on validation data and 0.82 on held-out test data.

A comparison of the F1-Scores obtained with each of the classification algorithms are as follows:

Algorithm	F1 Score
Logistic Regression	0.76
SVM	0.82
KNN	0.78
Decision Tree	0.82
Random Forest	0.83

Table 1: Algorithms with F1 scores

Event Driven Email Notifications

An optional event-driven workflow demonstrates how aggregate churn risk signals can be surfaced when model outputs change meaningfully. The intent is to illustrate how predictions can participate in downstream signaling rather than to prescribe email-based delivery as a production solution.

In practice, dashboard-driven workflows or integration with operational systems would typically be preferred to avoid alert fatigue.

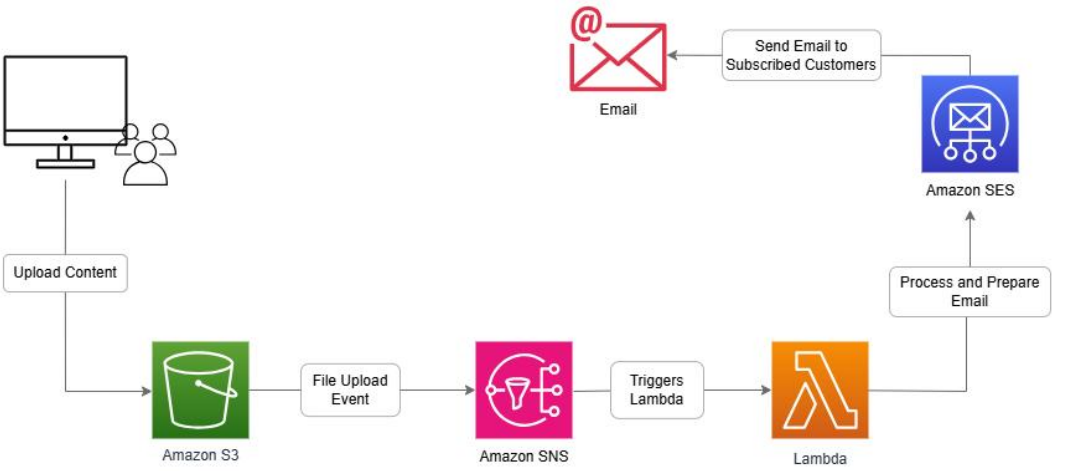
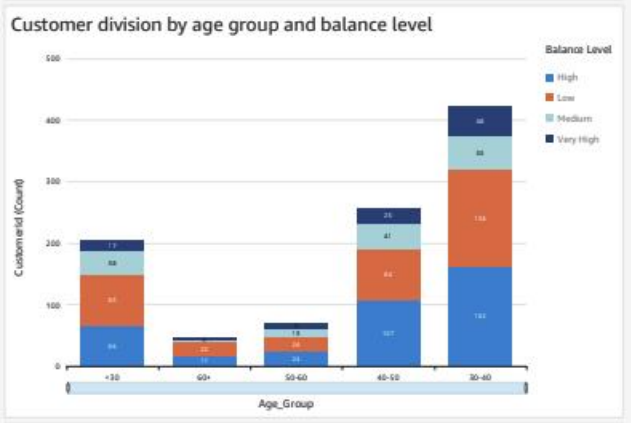


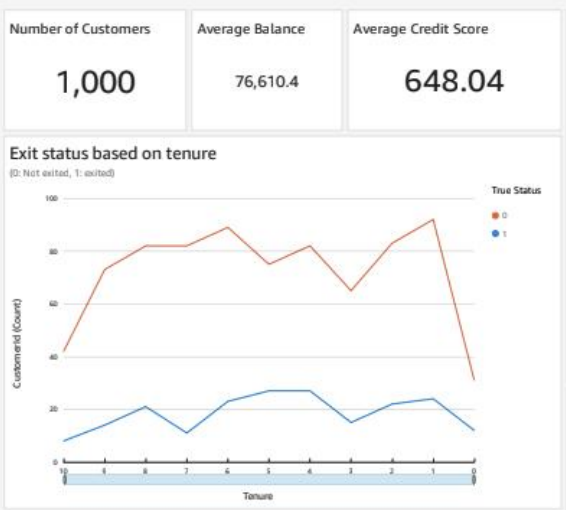
Figure 6: Architecture for notification service

AWS Quicksight - Data Visualization and Analysis

Model outputs are consumed through interactive QuickSight dashboards designed for non-technical stakeholders. Dashboards focus on segment-level churn risk, engagement patterns, and balance tiers rather than individual predictions.

This consumption layer enables stakeholders to explore churn dynamics, validate assumptions, and prioritize intervention strategies without interacting directly with raw data or model internals.





Final model predictions are exported to S3 as structured datasets rather than retained as in-memory notebook outputs. This design enables reuse of predictions across dashboards, analyses, and downstream workflows without coupling consumers to the modeling environment.

Treating predictions as first-class data artifacts allows longitudinal analysis, segment-level monitoring, and integration with systems that do not directly interact with machine learning code.

Design Tradeoffs & Limitations

1. The dataset is static and observational; insights are directional rather than causal
2. Threshold selection depends on business cost tolerance and intervention capacity
3. The system prioritizes interpretability and stability over marginal performance gains
4. Alerting is demonstrated at an aggregate level to avoid per-customer noise
5. Real-time inference and drift monitoring are intentionally out of scope