Machine Learning Deployment Architecture Plan

Capstone Project: Customer Churn Prediction

1. System Overview

The deployment architecture is designed to transition our customer churn prediction model from prototype to production. The Random Forest classifier, which achieved an ROC AUC score of 0.841 during development, will be used to ensure reliability, performance, and maintainability. The system will process incoming customer data through a structured pipeline that handles preprocessing, prediction, and monitoring while maintaining data integrity throughout the lifecycle.

2. Core System Components

Data Storage and Management

All customer data resides in AWS S3 buckets organized with date based usaage for efficient storage and retrieval. Feature sets and models are stored separately from raw data to maintain reproducibility.

Model Serving Infrastructure

The prediction service is built on AWS SageMaker, which allows for infrastructure and deploying machine learning models. We use SageMaker's builtin Scikit learn to allow for training and testing environments.

API Integration Layer

External applications interact with the system through an API Gateway that routes requests to a preprocessing Lambda function. This Lambda performs critical validation and transformation tasks, applying the same StandardScaler and OneHotEncoder operations used during model training.

3. Model Lifecycle Management

Training and Retraining Process

The model undergoes scheduled retraining monthly, with additional triggers based on performance metrics and data drift detection. When retraining initiates, the system collects fresh production data and combines it with historical datasets stored in S3. Feature engineering replicates the original preprocessing pipeline exactly to maintain consistency. Hyperparameter tuning occurs through SageMaker's automated optimization, exploring configurations around tree depth and ensemble size.

Evaluation and Deployment

New model versions must achieve at least 80% ROC AUC on a test set before progressing to deployment. The model can then be used for full production deployment.

4. Monitoring and Maintenance Framework

Performance Tracking

The monitoring system collects three categories of metrics: infrastructure health indicators like latency and throughput, model quality measures including weekly AUC calculations, and data consistency checks that compare live feature distributions against training baselines. CloudWatch dashboards aggregate these metrics for operational visibility while automated alerts notify engineers of anomalies.

Alert Management

Critical failures trigger PagerDuty alerts, performance issues notify Slack channels, while minor concerns create Jira tickets for later review.

Cost Management

We optimize costs using spot instances for 70% savings, auto-scaling endpoints, and S3 lifecycle policies, with a $165 monthly baseline for 100K predictions.

Implementation Timeline

The first two weeks establish core infrastructure, weeks 3 to 4 gradually roll out to production, followed by ongoing monthly reviews and optimizations.