Machine Learning Product Design

Product: Real-Time Customer Churn Prediction System

Authors: Emmanuel Kirui Barkacha, Audrey Djiosseu Tiodo and Chist Sagombaye

Date: 2025-10-17

1. BACKGROUND	2. VALUE PROPOSITION	3. OBJECTIVES	4. SOLUTION	5. FEASIBILITY
User: Telecommunications company. **Goal:** Predict customer churn in real time and understand why a customer is likely to churn(explainability). **Pain:** High customer churn rate leading to revenue loss.	**Product:** Real time churn prediction system. **Alleviates:** Enables proactive retention through timely alerts. **Advantages:** Improves customer retention and reduces revenue loss.	• Develop a churn prediction model with high accuracy. • Implement explainability and transparency. • Deploy real time inference pipeline. • Enable visualization dashboard for churn insights. • Establish automated monitoring and retraining.	**Core Features:** Data pipeline and preprocessing, model training, real-time inference API, monitoring dashboard. **Integration:** Integrate with a user interface and database via REST APIs for easier access and management. Also integration with Prometheus and Grafana dashboards for monitoring. **Alternatives:** Supports batch inference for scheduled predictions when real-time is not required. **Constraints:** low-latency performance for real-time predictions. **Out-of-Scope:** Role-specific business workflows and advanced AI models (e.g., reinforcement learning).	• **Data:** Kaggle datasets available; production data can be simulated. • **Processing:** Supports data balancing and preprocessing for model reliability. • **Infrastructure:** Azure cloud provides required storage, compute and container resources. • **Team:** Skilled in machine learning, backend, frontend and DevOps for full implementation.

6. DATA & SOURCES	7. LABELING	8. METRICS	9. EVALUATION	10. MODELING
Training Data: Uses Kaggle's Telco Customer Churn dataset. Data is split into training, validation, and testing sets, with sampling techniques (e.g., stratified sampling, SMOTE) to address class imbalance. **Production Data:** Ingested via secure APIs in batch or real-time modes. Data quality and trust ensured through schema validation and automated monitoring.	• The dataset includes a predefined "Churn" label (Yes/No). • The label is converted to binary (1 = churned, 0 = retained). • Key features include demographics, account details (tenure, contract type, payment method), and service usage metrics. • Categorical data is encoded, numerical data is scaled, and class imbalance is addressed using stratified sampling or class weighting.	• Accuracy, Precision, Recall, F1-score, ROC-AUC.	**Offline:** Evaluate models using an 80/20 train-test split with metrics like AUC, Precision, Recall, and F1-score. Compare baselines (Logistic Regression, Random Forest, XGBoost) to select the best model. **Online:** Simulate real-time evaluation with rolling windows against reference data. Use tools like Evidently AI to detect feature and prediction drift, triggering alerts (i.e if AUC drops by more than 5% or drift is statistically significant.	Iterative experimentation with Logistic Regression, Random Forest, and XGBoost. Use feature selection, hyperparameter tuning, and cross-validation. Initial baselines built using simple rule-based methods before ML models.
11. INFERENCE	**12. FEEDBACK**	**13. PROJECT**		
Real-time (online) inference through REST API for instant churn prediction. Batch inference for periodic churn reports and analytics dashboards.	Continuous model monitoring via Prometheus and Grafana. Human-in-the-loop feedback to review and correct false predictions. Automated drift detection triggers retraining pipeline.	**Team:** Skilled in machine learning, backend, frontend and DevOps. **Key deliverables:** ML model, deployed API, functional frontend, and CI/CD pipeline. **Timeline:** Project scheduled for 3 weeks duration.		

System design flow diagram