Friday 17 October 2025

MLOPS Product Design of Real-Time Customer Churn Prediction System project

Lecturer: Vangelis Oden - Technology Lead (Kera)
Assistant: Natalija Mitic, Ismael Coulibaly - AI/ML Engineer (Kera)

Groupe 3 : Papa, Mahamat, Elhadji

AMMI 2024/2025

Background: 3

Value proposition: 3

Objectives 3

Solutions 4

Feasibility 4

Data 4

Metrics 7

Evaluation 7

Modeling 7

Inference 8

Feedback 8

Project 8

Background:

- **Users**: Telecom company (marketing managers, data analysts, customer service representatives).
- **Pains** : high customer churn rate, high cost for acquiring new customers, lack of ability to anticipate and retain at-risk customers ,direct impact on company revenue.
- **Goal** : Develop a **real-time churn prediction system** that enables the company to reduce losses (revenue, customers), turn churn into opportunity (better customer experience), improve customer experience and satisfaction by understanding churn drivers.

Value proposition:

- **Product** : A real-time customer churn prediction system
- **Alleviates** : Identify potential customer churn and enable proactive marketing and retention strategies.
- **Advantages**: reduce customer churn rate, improve customer retention through targeted marketing actions.

Objectives

- Reduce the churn rate of high-risk customers
- Proactively identify 'at-risk' customers for targeted retention actions
- Optimize retention budgets by focusing on customers most likely to respond

Solutions

- Develop a real time pipeline to predict churn probability per each customer
- Build ETL Pipeline for Data and applying feature Engineering to Identify Most Impactful features
- ML Task: Binary Classification (Churn or No Churn)
- Input Data :Customer demographic, service, and contract data
- Output : Churn Probability score for each customer
- Model Candidates : Random Forest, XGBoost, Neural Network…
- Integration **:** Integrate Model to Churn Management System
- Action: Notify the sales team or launch a customer retention campaign.
- Monitoring: Model drift detection, A/B testing on retention rate, latency tracking.
- Constraint: Streaming Data, model Explainability
- Out of Scope: Churn reason explanation

Feasibility

- Data : Data available on kaggleTools: Open-source (MLflow, DVC,FastAPI), Cloud: Azure for deployment and scalability.
- Infrastructure: The system uses Airflow to orchestrate data ingestion, model training, deployment, and monitoring. Data from the Telco Churn dataset is versioned with DVC, models are tracked with MLflow, and deployed through FastAPI in Docker, automated by GitHub Actions. Evidently AI monitors data drift and triggers retraining via Airflow to maintain model performance.
- Team : Mathematician, DevSecOps, Software Engineer.
- Budget : Open source tools (MLflow, DVC, Fast API) and Azure for cloud deployment.

Data

Data source**:** Kaggle Telco Customer Churn dataset

Data Volume: ~7000 records initially: raw data contains 7043 rows (customers) and 21 columns (features)

Key components:

Demographics**:** gender, age group, partner, dependents.

Services**:** phone, internet, online security, backups, device protection, tech support, streaming TV/movies.

Contract**:** tenure, contract type, payment method, paperless billing, monthly and total charges.

- **Training :**

  Stratified sampling to maintain class balance,SMOTE for synthetic oversampling of minority class,feature engineering to enhance predictive power.

  Other considerations:

  Feature engineering

- **Production:**

  Access to real-time streams of data

- **Labeling**: “Churn” column is binary (Yes/No) — already labeled.

Assumptions

Churn is clearly defined as customers whose Churn = "Yes".

The label is accurate and timely, meaning that once a customer leaves, it is immediately reflected in the dataset.

Customer data (e.g., tenure, monthly charges, contract type) is consistent and reliable across all records.

All relevant behavioral and demographic variables are available and complete.

The churn label reflects the customer’s true decision (not technical or billing errors).

Reality

In real-world telecom operations, churn is not always detected immediately, there can be a delay between customer departure and system update, you won’t have the label immediately after a customer leaves.

Some customers pause their service or change their plan, which may look like churn but isn’t.

Data quality issues exist: missing values, incorrect contract information, or inconsistent billing records.

Behavioral data like service usage may not be as detailed or as clean in production as in the static dataset.

The static dataset does not include timestamped events, making it hard to model churn timing in real time.

The dataset assumes all features are available at prediction time, which is not always true in live systems.

Decisions

The labeling logic in production should define churn as “no active contract for X consecutive days”.

A delay-aware labeling mechanism can be built to handle lag between the churn event and label confirmation.

Use data validation DAGs in Airflow to detect missing or delayed churn labels.

Build a feedback loop to recheck churned users who later return (to prevent false positives).

Introduce a staging dataset to separate “pending churn” from “confirmed churn”.

Metrics

- **PR-AUC** measures the overall quality of predictions in an imbalanced context.

- **Recall** captures the *sensitivity* to churners.
- **Precision** captures the *cost-efficiency* of interventions.
- **F1-score** summarizes Recall and Precision in one interpretable metric.

Evaluation

- **Offline**: Use Validation and Test Dataset to Ensure that the model is reliable using Techniques(Cross Validation) and Metrics (F1,Precision…)
- **Online**: A/B testing is conducted to measure the model’s real impact on business indicators.In parallel, Continuous Monitoring is integrated into the MLOps pipeline to track the model’s operational performance and the quality of data in production.

Modeling

End to end utility: Obtain a first baseline and track the progress of subsequent iterations

Manual before ML: Start with simple rule-based to identify customers at risk using the dataset key features.

Augment vs automate: First aims to assist marketing teams in identifying at-risk customers, by providing them with a churn probability score, before considering full automation of retention actions

Internal vs external: The first versions of the model remain internal, used to test and adjust performance without direct exposure to the end user, in accordance with the “internal vs external” principle. These models include Random Forest, XGBoost, Neural Network which are deployed through FastAPI.

Thorough: The entire process is conducted in a rigorous and exhaustive manner, carefully testing the code (unit test,integration test), testing on data (schema validation, anomaly detection) and robustness of the models while comparing several approaches to select the one offering the best balance between performance, interpretability and operational stability.

Inference

- **Batch inference:** Batch scoring (weekly/monthly) for inactive users
- **Online inference:** Real-time REST API scoring for individual customers

Feedback

- **Human in the loop:** Agents can provide qualitative feedback on the relevance of the recommendations, the accuracy of the scores, or the identified customer segments.
- **Automated feedback:**
  Prediction results vs. reality
  Logs and technical metrics
  Monitoring of business KPI with respect to metrics
  Drift monitoring (Data / Concept Drift)

Project

- **Deliverables:**

  Complete codebase with documentation
  CI/CD pipeline configuration
  Deployed model with accessible API endpoint
  Live monitoring dashboard

Technical report covering architecture, design decisions, and results
Presentation with Demo

- **Team Members**:

  Data Scientists / Data Engineers:
  Data exploration, pipeline construction, and feature engineering.

  Ensure scalability, reliability, and compliance with data governance.

  Develop and evaluate predictive algorithms.

  Implement anonymization, encryption, and integrity checks.

  Manage data lineage and access control (RBAC).

  ML Engineers / MLOps Specialists:
  Automate the ML lifecycle with CI/CD pipelines.

  Manage model deployment, versioning, and retraining workflows.

  Secure model endpoints (HTTPS, encrypted communication).

  Monitor predictions and performance in production.

  Software Engineers:
  Integrate model outputs into production applications.

  Develop APIs, dashboards, and microservices for business workflows.

  Secure API endpoints (RBAC, tokens).

  Log API requests and errors for observability.

  Product Owner / Project Manager:

  Coordinate teams, define priorities, manage timelines, and ensure measurable business outcomes.

- **Project Timelines**:
  Week 1: ML Product Design, Architecture System and Data Understanding
  Week 2: Data Preparation, Modeling - Offline Evaluation, Build API System
  Week 3: Deployment - Online Evaluation - Retraining and Inference