Team Members:

1. Furkan Kırteke 150240704

2. Burak Koçoğlu 150220738

3. Defne Yıldırım 150230727

1. Problem Definition

In the telecommunications industry, customer retention is a critical business metric, as the cost of acquiring a new customer is significantly higher than retaining an existing one. However, customer behavior is volatile; a model trained on data from 2023 may fail to accurately predict churn behaviors in 2025 due to changing market trends, pricing models, or competitor offers.

The Problem: Most churn prediction solutions are static "one-off" models that degrade in performance overtime (model decay) and lack the infrastructure to detect when they becomeunreliable. **Stakeholders:** Marketing teams (to target at-risk users), Customer Success Managers (to prioritize retention efforts), and Business Executives (forecasting revenue). **Goal**: We aim to build a deployable, real-me churn prediction service that not only predicts which customers will leave but also automatically alerts stakeholders when the incoming data distribution shifts (data drift), ensuring the system remains reliable without constant manual checking.

2. Motivation & Course Fit

This project is designed to move beyond "model training" and focus on the Machine Learning Systems (MLOps) lifecycle as defined in the course objectives. It addresses the following course themes:

Data to Deployment: We will move from a static CSV dataset to a live REST API, containerized with Docker for reproducibility.

Monitoring & Maintenance: A core component of our project is implementing a monitoring layer (using Evidently AI or Prometheus) to detect "Data Drift." This directly addresses the "Monitoring & maintenance" item in the Project Design Checklist.

Scalability & Reliability: By decoupling the training pipeline from the inference service, we demonstrate a microservices architecture pattern suitable for scalable deployment.

Trade-offs: We will analyze the trade-off between model complexity (e.g., Deep Learning vs. XGBoost) and inference latency (latency requirements for real-me customer support

dashboards).

3. Existing Work

Customer churn prediction is a well-researched domain in Data Science.

• **Literature**: Standard approaches utilize Logistic Regression, Random Forests, or Gradient Boosting on historical usage data. Studies often focus heavily on hyperparameter tuning to maximize accuracy (AUC-ROC).

• **Gaps:** Existing academic projects o en stop at the "Jupyter Notebook" phase. They rarely address the opera onal challenges: How is the model served? How do we know if the model is hallucinating due to new data patterns? How do we handle model versioning?

• **Our Contribution:** We bridge this gap by building a Continuous Training (CT) pipeline. Unlike standard implementations, our system will simulate "production traffic" to demonstrate how the system reacts to data drift and triggers alerts, focusing on the system reliability rather than just raw accuracy.

4. Planned Approach

We plan to execute this project in four operational phases :

• **Phase 1: Data Pipeline & Experimentation**

o **Dataset:** We will use the Telco Customer Churn Dataset (Kaggle), which contains customer demographics, services, and tenure.

o **Feature Engineering:** Develop a reproducible preprocessing pipeline (handling categorical variables, scaling) using Scikit-learn pipelines.

o **Experiment Tracking:** Use **MLflow** to log experiments, tracking parameters and metrics to select the best baseline model (likely XGBoost or Random Forest).

Phase 2: Deployment & Inference

• **Model Serving:** Wrap the best model in a FastAPI application to provide real- me predictions (/predict endpoint).

• **Containerization:** Dockerize the training script and the API to ensure the system runs reliably on any machine (solving the "it works on my machine" problem).

Phase 3: Monitoring & Drift Detection

• **Drift Simulation:** We will artificially induce drift (e.g., change the age distribution of input data) to simulate a changing market.

• **Monitoring:** Integrate Evidently AI to monitor data distribution changes and model

performance degradation. The system will generate a "Health Report" dashboard.

Phase 4: Integra on & Demo

• Build a simple Streamlit UI for the "Business Stakeholder" to view churn predictions and system health status.

• Finalize the "Ethical Considerations" report regarding bias in demographic data (e.g., is the model biased against senior citizens?).

5. References

1. Kaggle. (n.d.). *Telco Customer Churn. Retrieved from* https://www.kaggle.com/datasets/blastchar/telco-customer-churn

2. Alla, S., & Adari, S. K. (2021). *Beginning MLOps with MLflow: Deploy, Train, and Manage Machine Learning Models. Apress.*

3. Breck, E., et al. (2017*). The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction*. IEEE Interna onal Conference on Big Data.

4. Evidently AI. (2024). *Open-Source Machine Learning Monitoring*. https://www.evidentlyai.com/