**Assignment submission – part of** [**DUKE AI PM**](https://www.coursera.org/specializations/ai-product-management-duke)

**PREDICTING CUSTOMER CHURN FOR OTT PLATFORM USING VIEWING BEHAVIOUR AND ENGAGEMENT DATA**

**Problem Statement**  
Design a machine learning model to predict which users are likely to cancel their subscription within the next 30 days, based on their watch history, time spent, and engagement behavior.

**Why ML?**  
ML enables scalable personalization, continuous prediction, and automation — all critical in dynamic platforms like OTT.  
Traditional heuristics (e.g., time spent or frequency of usage) are static and cannot capture subtle changes in user intent or preference. A churn prediction model can dynamically learn behavior patterns, detect early signals of drop-off, and enable the business to take preemptive action — via better content recommendations, pricing nudges, or re-engagement campaigns.

A regularly retrained ML model can adapt quickly to user behavior shifts — weekly retraining ensures relevance. This means ML isn’t just about automation, but staying ahead of customer intent.

**2. CRISP-DM: Business Understanding**

Goal  
Predict the likelihood of a user churning in the next month. The model output will support the product and strategy teams in targeting high-risk users for retention actions.

Success Definition  
Business success is defined as reducing the overall monthly churn rate and improving customer lifetime value.  
Key metric: Reduce churn by at least 30% in 6 months post-deployment.  
Model output: A churn probability score per user.  
Model outcome: Revenue retention, improved targeting, and reduced acquisition pressure.

Constraints & Considerations

* Heuristics like "low engagement" aren't enough — churn can also be driven by pricing, competition, or shifts in user interest.
* Must account for concept drift — a user might still use the platform but be ready to leave due to other external factors (like price hikes).

Potentially Relevant Features

* Watch time trends
* Frequency of platform visits
* Recommendation engagement (click-throughs)
* Region and content language
* Recent price changes or promotions
* Presence of new competing content/services

**3. Data Understanding**

Data can be sourced from:

* Internal user analytics dashboards (session logs, watch history)
* Marketing reports (new releases, campaigns)
* Churn user insights (exit surveys, usage drop patterns)

Initial steps:

* Validate and clean the data
* Handle missing values
* Analyze distributions and feature correlations

**4. Data Preparation**

* Split data into training and testing sets
* Engineer features like:
  + “Browsing time” = Session time – Watch time
  + Engagement diversity = Number of unique genres watched
* Normalize time-based variables if needed
* Ensure balance across churn/non-churn users to avoid model bias

**5. Modeling**

* Start with interpretable models like Decision Trees or Random Forests
* Weekly retraining to adapt to user trend shifts
* Include fallback heuristics to support edge cases

**6. Evaluation**

* Validate with standard classification metrics (precision, recall, F1)
* Track false positives/negatives in real usage scenarios
* Compare model results with existing heuristics

**7. Deployment**

* Deployed via cloud-based API or internal dashboard
* Latency isn’t a major concern (not real-time), so batch predictions are viable
* Ensure regular feedback loop into training set

**Solution Validation Plan**

The model will be piloted on a subset of users and predictions compared with actual churn events.  
Evaluation includes:

* Comparison with current heuristic methods
* A/B testing for interventions on predicted churners
* Business metric tracking (retention uplift, campaign ROI)

**ML System Design Decisions**

* Latency tolerance: High — predictions used in batch
* Deployment infra: Cloud — no privacy risks; suitable for web dashboards
* Retraining: Weekly with a drift detection mechanism

**Risks in Production**

* Training-serving skew: Misaligned features in prod vs. training — mitigated via schema enforcement
* Data drift: Users may leave sessions open, inflating session time — solve with activity filters
* Concept drift: E.g., churn due to price hike — model may not detect until after impact; requires business context
* Imbalanced data: Needs stratified sampling or class weighting