

2) Check-in — Proyecto Final DL

Predicción de Churn en KKBOX (WSDM Cup 2018)

i

Fecha:

Miércoles 3 de noviembre de 2025

Equipo:

Sebastián Bolívar — David Rojo

Objetivo:

Predecir `is_churn` usando señales transaccionales y de engagement; motivación principal: capturar dinámica temporal del uso (secuencias) más allá de promedios.

1) Avance (lo logrado hasta hoy)

- **Pipeline de datos en Kaggle listo:** descompresión de archivos `.7z`, carga eficiente con reducción de memoria y liberación de RAM (`gc`), integración de tablas `train_v2`, `members_v3`, `transactions_v2`, `user_logs_v2`.
- **Baseline tabular (una fila por usuario)** con **970,960** usuarios y **19 features** (demografía + última transacción + agregados de logs).
 - **Churn rate: 8.99%** (desbalance moderado).
- **Modelos baseline entrenados:**
 - **Logistic Regression:** ROC-AUC **0.934**, PR-AUC **0.780**
 - **XGBoost (mejor baseline):** ROC-AUC **0.945**, PR-AUC **0.829**
- **Ablaciones** para identificar qué fuente de señal domina (transacciones vs logs vs members) y aislar la dificultad real del problema.

2) Retos (lo más importante)

- **Dominancia del bloque transaccional:** el modelo aprende gran parte de la señal con variables de pago/cancelación, lo que puede ocultar el valor real del engagement.
- **“Trampa del promedio” en logs:** agregados globales de `user_logs_v2` (sumas/promedios) muestran baja capacidad predictiva; sugiere que la señal relevante está en **recencia/tendencias** (temporal).
- **Riesgo metodológico a vigilar (hacia la entrega final):** asegurar que las features usen solo información disponible **antes** del punto de predicción (control de leakage temporal).

3) Primeros resultados (métricas + evidencia)

Split: 80/20 estratificado (misma tasa de churn en train/valid).

Métrica principal: PR-AUC (por desbalance).

Baseline principal (XGBoost):

- **PR-AUC: 0.8286**
- **ROC-AUC: 0.9451**

Tabla — Ablaciones (XGBoost)

Experimento	PR-AUC	ROC-AUC	churn_rate	Interpretación
Solo transacciones	0.7738	0.9365	0.0899	Señal dominante del baseline
Solo logs (agregados)	0.1158	0.5674	0.0899	Agregados casi no capturan churn
Solo members	0.2091	0.7068	0.0899	Señal demográfica moderada
Full – sin <code>is_cancel</code>	0.6799	0.8690	0.0899	<code>is_cancel</code> es feature clave
Full – sin <code>is_auto_renew</code>	0.8243	0.9437	0.0899	Señal redundante/compensada
Full – sin <code>payment_method_id</code>	0.8243	0.9424	0.0899	Señal redundante/compensada
Subset <code>auto_renew=1</code>	0.6357	0.8904	0.0383	Caso difícil: churn menos “obvio”

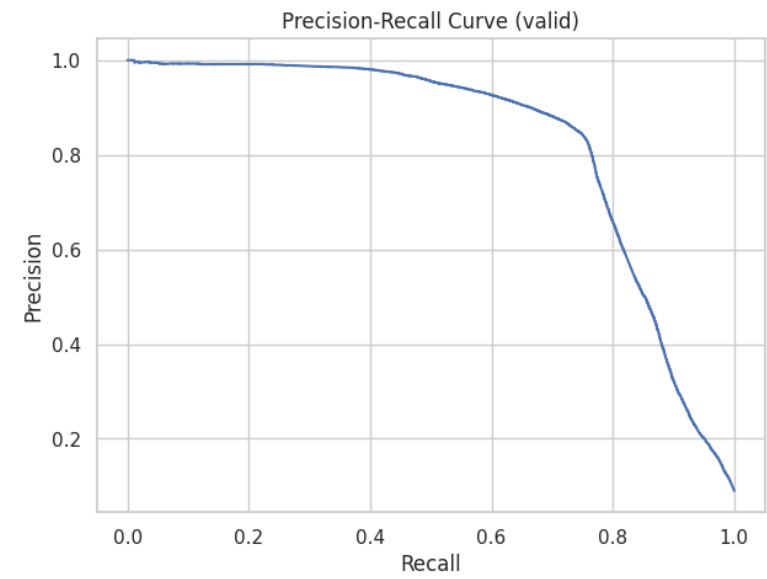
Conclusión de resultados:

- El baseline tabular funciona muy bien, pero **principalmente por transacciones**.
- Los logs agregados fallan → refuerza la hipótesis del proyecto: **la señal de engagement útil es temporal**, y debe modelarse como secuencia (GRU/Transformer).

- El subconjunto `auto_renew=1` hace visible el reto real (PR-AUC baja a 0.636), donde un modelo secuencial podría aportar más valor.

4) Figuras

1. Curva Precision–Recall del baseline XGBoost (validación)



La curva Precision–Recall (valid) muestra un desempeño muy alto: para recall bajos y medios (hasta ~0.6–0.7) la precisión se mantiene cercana a 0.9–1.0, lo que indica que el modelo identifica un grupo grande de churners “claros” con muy pocos falsos positivos. Además, se observa una “rodilla” alrededor de recall ~0.75–0.8, a partir de la cual la precisión cae más rápido: para capturar casi todos los churners el modelo debe aceptar muchos falsos positivos, por lo que un umbral cercano a esa rodilla es un punto operativo razonable.

Este resultado corresponde al escenario **full features**, donde se usan simultáneamente variables demográficas, transaccionales y agregados de logs; en particular, las features transaccionales aportan gran parte de la señal y elevan el rendimiento global. Precisamente por eso, el siguiente paso es profundizar en subpoblaciones donde el churn es menos “obvio” (p.ej., usuarios con `is_auto_renew=1`, donde el churn rate baja y el PR-AUC cae), para estudiar mejor qué aporta el comportamiento temporal y motivar el uso de modelos secuenciales (GRU/Transformer) enfocados en tendencias/recencia de engagement.

5) Diferencias frente al plan

- Se priorizó un **baseline tabular robusto + ablaciones** antes de construir tensores secuenciales completos, para:
 1. obtener una línea base fuerte y defendible,
 2. identificar dónde está la señal real, y
 3. definir mejor el objetivo del modelo secuencial (en especial `auto_renew=1`).

6) Próximos pasos (concretos)

1. **Feature engineering temporal (puente al modelo secuencial):** recencia/tendencia en logs (últimos 7/14/30 días, ratios de caída, días sin actividad, etc.).
2. **Construcción de secuencias T×F** (p.ej., T=30 días) con máscara para días faltantes.
3. Entrenar **GRU/LSTM** como baseline secuencial y luego **Transformer Encoder**.
4. Evaluación focalizada en:
 - PR-AUC global,
 - PR-AUC en el subset `auto_renew=1`,
 - y ablaciones por ventana temporal (T=30 vs T=60).