UNIVERSIDAD
DE ANTIOQUIA
1803

Predicción de Churn en plataforma de audio: Comparativo entre Modelos Tabulares (XGBoost) y Arquitecturas Secuenciales Profundas (GRU con Atención)

Sebastian Bolivar Vanegas - David Santiago Rojo Castrillón
sebastian.bolivar1@udea.edu.co - david.rojo1@udea.edu.co
Facultad de Ingenieria

***Resumen*—Este estudio evalúa estrategias de predicción de abandono (churn) en la plataforma KKBox, comparando la eficacia de la ingeniería de características tradicional (XGBoost) frente al aprendizaje profundo secuencial (GRU Híbrida). Mediante una arquitectura de datos dual (estática + dinámica) y una validación temporal estricta, los resultados demostraron la clara superioridad del modelo XGBoost (PR-AUC 0.8214) sobre la red neuronal (0.6120). Se concluye que, ante ventanas de observación cortas y alta dispersión de datos (sparsity), los resúmenes estadísticos explícitos ofrecen mayor robustez y capacidad de generalización frente al data drift que las arquitecturas complejas de Deep Learning.**

***Palabras Clave*—Predicción de Abandono (Churn), Aprendizaje Profundo (Deep Learning), Series Temporales, Ingeniería de Características, XGBoost, GRU, Mecanismos de Atención, Datos Dispersos.**

I. Introducción.

El abandono de clientes (Churn) rara vez es un evento repentino; suele ser el desenlace de una degradación gradual del engagement que los modelos tradicionales, basados en promedios o "fotos estáticas", fallan en detectar a tiempo.
Este proyecto aborda la predicción de churn en la plataforma KKBOX cambiando el paradigma: en lugar de aplanar el comportamiento del usuario en agregados mensuales, analizamos la interacción completa de su actividad diaria. Nuestra hipótesis central es que la secuencia temporal de sus métricas de consumo (la variación diaria en la duración de sesiones, la frecuencia de canciones saltadas vs. completadas y la recurrencia de uso) contiene señales de "fatiga de usuarioïnvisibles para los métodos convencionales..
.
Para validar esto, se empleará una estrategia incremental. Se inicia con un Baseline Tabular (XGBoost) utilizando variables estáticas y promedios de actividad. Posteriormente, avanzaremos hacia arquitecturas de Deep Learning Híbridas: una GRU con Attention Pooling y finalmente un Transformer Encoder. Estos modelos no mirarán el promedio de uso, sino la trayectoria del comportamiento, buscando identificar caídas sutiles o patrones de desconexión que anticipen la no renovación de la suscripción.
.

II. Trabajo relacionado:

La predicción de la deserción de clientes (churn prediction) se ha consolidado como un pilar estratégico en la gestión de relaciones con clientes (CRM). No se trata únicamente de una métrica de retención, sino de un barómetro de la satisfacción del usuario. Históricamente, este desafío ha evolucionado desde enfoques puramente transaccionales hacia modelos conductuales capaces de interpretar la "narrativa"de uso de un suscriptor.

De la "sección Estática.ª la Ingeniería de Características
Tradicionalmente, la literatura ha abordado este problema mediante modelos de aprendizaje automático clásico, tales como la Regresión Logística, Random Forest y, más recientemente, máquinas de Gradient Boosting (como XGBoost o LightGBM). Estos enfoques dependen intrínsecamente de una rigurosa ingeniería de características manual. Se basan en la creación de ïnstantáneas.º resúmenes del usuario, utilizando marcos como RFM (Recencia, Frecuencia, Monetización) y estadísticas agregadas (promedios de uso, desviaciones estándar).
Si bien estudios previos demuestran que estos modelos son extremadamente robustos y eficientes en datos tabulares estructurados, presentan una limitación conceptual crítica: al promediar el comportamiento, pierden la dimensión temporal. Un usuario que disminuye gradualmente su consumo día

tras día puede tener el mismo "promedio mensual"que uno estable, pero su riesgo de abandono es radicalmente distinto. Los modelos estáticos tienden a ignorar esta degradación secuencial del servicio que a menudo actúa como preludio silencioso del abandono.

El enfoque hacia el Deep Learning Secuencial

En respuesta a esta limitación, el estado del arte ha migrado hacia el Deep Learning, tratando el historial de logs del usuario no como una tabla, sino como una serie de tiempo o una secuencia, análogo al procesamiento de lenguaje natural. Arquitecturas como las Redes Neuronales Recurrentes (RNN), y específicamente sus variantes con memoria a largo plazo (LSTMs) y Unidades Recurrentes Cerradas (GRUs), han ganado tracción.

Investigaciones recientes en plataformas de streaming (música y video) sugieren que estas redes son capaces de capturar patrones latentes y micro-comportamientos —como un aumento repentino en el skipping de canciones o sesiones cada vez más cortas— que escapan a los modelos tradicionales. La premisa es que el abandono es un proceso, no un evento puntual, y las RNNs están diseñadas para modelar esa evolución.

Mecanismos de Atención y Arquitecturas Híbridas

Un avance crucial en esta línea es la incorporación de mecanismos de atención (Attention Mechanisms). En lugar de tratar todos los días del historial con la misma importancia, la .ªtención"permite al modelo aprender qué días o eventos específicos (por ejemplo, un fallo en la reproducción o un fin de semana sin actividad) pesan más en la decisión de cancelar. Esto no solo mejora la precisión, sino que aporta una capa de interpretabilidad, permitiendo identificar los "momentos de la verdad.ªen la experiencia del cliente.

Este trabajo se inserta en esta corriente de vanguardia, pero con una perspectiva integradora. Reconocemos que el comportamiento secuencial no lo es todo; el contexto contractual (planes, auto-renovación) es igualmente decisivo. Por ello, proponemos una arquitectura híbrida que busca lo mejor de ambos mundos: la solidez de las variables estáticas y transaccionales, combinada con la sensibilidad de una GRU para detectar los matices temporales de la desconexión del usuario.

III. DATOS:

Para el desarrollo de este estudio, se utilizaron los conjuntos de datos proporcionados por la competición pública WSDM - KKBox Churn Prediction Challenge. La complejidad del problema requirió la integración de cuatro fuentes de información heterogéneas, unificadas bajo el identificador único de usuario (msno). El resultado es una arquitectura de datos dual que combina un perfil estático del suscriptor con una representación dinámica de su comportamiento reciente.

1. **Definición de la Variable Objetivo (Y)**
   La variable dependiente, proveniente del archivo train_v2.csv, modela el abandono como un evento binario basado en la ventana de renovación:
   * Y=1 (Churn): El usuario no registró una renovación de suscripción dentro de los 30 días posteriores a su fecha de expiración.
   * Y=0 (Renovación): El usuario extendió su servicio exitosamente dentro de la ventana permitida.

2. **Fuentes de Información y Características.**
   A. Dimensiones Estáticas (Contexto del Usuario)
   Estas variables proporcionan la "fotografía"del usuario en el momento del corte y provienen de dos fuentes:

   * Demográficas (members_v3.csv): Atributos personales como ciudad, género, edad (bd) y método de registro (registered_via).

   * Transaccionales (transactions_v2.csv): Dado que un usuario puede tener múltiples transacciones históricas, se aplicó una lógica de selección para retener únicamente la última transacción válida antes de la fecha de expiración. De aquí se extrajeron predictores críticos como la activación de renovación automática (is_auto_renew), el método de pago (payment_method_id), el precio de lista del plan (plan_list_price) y si hubo cancelaciones previas (is_cancel).

   B. Dimensiones Dinámicas (Secuencias de Comportamiento))
   La fuente más rica de información reside en los registros diarios de actividad (user_logs_v2.csv). Para capturar la evolución del consumo, se construyó una ventana de observación temporal (T) de 30 días previos a la fecha de expiración de la membresía (membership_expire_date). Para cada día t dentro de esta ventana, se generó un vector de características (X_t) que distingue entre diferentes tipos de interacción:
   * **Engagement y Satisfacción:** Medido a través de num_100 (canciones escuchadas en su totalidad) y total_secs (tiempo total de escucha diaria).
   * **Desinterés y Fricción:** Capturado mediante num_25 (canciones saltadas antes de completar el 25 % de su duración), un indicador clave de comportamiento de skipping.
   * **Exploración:** Cuantificada por num_unq (número de canciones únicas reproducidas), que denota la variedad del consumo.
   * **Métricas auxiliares:** Conteos intermedios de progreso (num_50, num_75, num_985).

3. **Preprocesamiento y Limpieza** Para garantizar la calidad de los datos de entrada a los modelos, se aplicaron las siguientes transformaciones:

   Filtrado Temporal: Se eliminaron registros con fechas de transacción o expiración atípicas (fuera del rango lógico 2015-2017) para mantener la coherencia histórica.

Normalización de Secuencias: Dado el sesgo positivo en los conteos de reproducción (donde algunos usuarios escuchan cientos de canciones y otros ninguna), se aplicó una transformación logarítmica (log1p) seguida de un escalado estándar.

Gestión de Datos Faltantes: Para los días sin actividad registrada dentro de la ventana de 30 días, se utilizaron máscaras de secuencia, permitiendo que los modelos distingan entre un valor çero"por falta de datos y un çero"por inactividad real.

IV. METODOLOGÍA:

La estrategia metodológica del estudio se diseñó para realizar una evaluación comparativa rigurosa entre dos paradigmas de modelado: la ingeniería de características tradicional sobre modelos de Gradient Boosting frente al aprendizaje de representación secuencial mediante Deep Learning. El objetivo central fue determinar si la complejidad computacional de las redes neuronales aporta una ganancia predictiva significativa sobre las técnicas estáticas establecidas en la industria.

Diseño Experimental y Validación Temporal

Para simular con fidelidad un entorno de producción real y mitigar el riesgo de fuga de información futura (look-ahead bias), se descartó la validación cruzada aleatoria en favor de una partición temporal estricta. El conjunto de datos se dividió estableciendo el 25 de marzo de 2017 como fecha de corte: los usuarios cuya suscripción expiraba antes de esta fecha conformaron el conjunto de entrenamiento, mientras que aquellos con expiración posterior fueron asignados al conjunto de validación.

Esta segmentación cronológica reveló un desafío crítico para la generalización de los modelos: un desplazamiento significativo en la distribución de la variable objetivo, conocido técnicamente como data drift. Se observó una discrepancia severa en las tasas de abandono, registrándose un 63.4 % en el conjunto de entrenamiento frente a un drástico descenso al 32.7 % en el conjunto de validación. Este desbalance implica que los modelos no solo debían aprender patrones de comportamiento, sino también demostrar robustez ante cambios estructurales en la población de usuarios entre principios y finales de mes.

Arquitecturas de Modelado

Como modelo de control y línea base, se implementó el algoritmo XGBoost, ampliamente reconocido por su eficacia en datos tabulares. Dado que este algoritmo no procesa nativamente secuencias temporales, se aplicó un proceso de ingeniería de características explícita para çolapsar"la ventana de observación de 30 días en un vector de atributos estáticos. Este procedimiento transformó la serie temporal en descriptores estadísticos agregados —tales como la media, varianza, pendiente de la actividad lineal y tasas de recencia— bajo la hipótesis de que estos resúmenes capturan suficientemente la tendencia de degradación del servicio sin necesidad de procesar la secuencia día a día.

En contraste, el modelo experimental çhallengerçonsistió en una Arquitectura Híbrida de Deep Learning. Este modelo ingiere directamente tensores tridimensionales de tamaño (N,30,F), permitiendo un aprendizaje de extremo a extremo. La arquitectura integra una unidad recurrente GRU (Gated Recurrent Unit) para modelar las dependencias temporales a corto plazo, seguida de un mecanismo de Attention Pooling. Este último componente es crucial, ya que permite a la red ponderar dinámicamente la relevancia de cada día, aprendiendo a identificar "momentos de la verdad.^{es}pecíficos en el historial. Finalmente, la representación vectorial extraída de la secuencia se concatena con las variables estáticas demográficas y transaccionales antes de la clasificación, fusionando así el contexto histórico con el comportamiento reciente del usuario.

V. RESULTADOS:

El análisis cuantitativo de los experimentos arrojó resultados concluyentes que contradicen la hipótesis inicial sobre la superioridad de la complejidad arquitectónica. En este escenario específico, caracterizado por ventanas temporales cortas y datos heterogéneos, el enfoque tradicional basado en ingeniería de características demostró una eficacia significativamente mayor que el modelado secuencial profundo.

Rendimiento Cuantitativo Comparado

El modelo de control (XGBoost) estableció una línea base excepcionalmente robusta, alcanzando un PR-AUC de 0.8214 y un ROC-AUC de 0.8677. Un hallazgo notable fue la rápida convergencia del algoritmo, que maximizó su rendimiento en torno a las 100 iteraciones; esto indica que las características agregadas manualmente contenían una señal predictiva de alta densidad, permitiendo al modelo discriminar clases con mínimo esfuerzo computacional.

Por el contrario, la arquitectura experimental (GRU + Atención) mostró un desempeño inferior, registrando un PR-AUC de 0.6120 y un ROC-AUC de 0.7079. Las curvas de aprendizaje de la red neuronal exhibieron inestabilidad, evidenciando dificultades para extraer patrones generalizables de los datos secuenciales crudos.

Análisis Cualitativo de las Limitaciones.

La disección del bajo rendimiento del modelo de Deep Learning permitió identificar tres factores técnicos determinantes: En primer lugar, la dispersión de los datos (Sparsity) jugó un rol crítico. Se observó que los usuarios mantienen actividad en la plataforma, en promedio, solo 9 de los 30 días de la ventana de observación. Como consecuencia, los tensores de entrada alimentados a la GRU presentaban una prevalencia excesiva de valores cero. Esta falta de densidad informativa dificultó que la red recurrente aprendiera dependencias temporales sutiles, ya que la señal de comportamiento real se encontraba diluida en grandes periodos de inactividad.

En segundo lugar, se detectó una ineficacia en la integración de variables estáticas. Un subanálisis sobre el segmento de usuarios con renovación automática activada (is_auto_renew=1) reveló que la GRU operó con una precisión cercana al azar (PR-AUC 0.53) en este grupo. Esto sugiere que la arquitectura compleja, al priorizar el procesamiento de la secuencia diaria, terminó .ªhogando"la señal determinística y contractual de la auto-renovación entre el ruido estocástico de los logs diarios. El modelo XGBoost, al recibir esta variable de forma explícita y directa, pudo capitalizarla sin interferencias. Finalmente, el desplazamiento de datos (Data Drift) derivado de la partición temporal penalizó desproporcionadamente al modelo secuencial. La discrepancia estructural entre las tasas de abandono del conjunto de entrenamiento (63 %) y validación (33 %) provocó que la red neuronal aprendiera patrones excesivamente "pesimistas". Al enfrentarse a la población de validación —usuarios de fin de mes con comportamientos de retención más estables—, el modelo careció de la flexibilidad necesaria para ajustar sus predicciones, evidenciando una menor capacidad de generalización frente a cambios distributivos en comparación con el enfoque basado en árboles.

VI. Ética

El desarrollo y despliegue de sistemas de inteligencia artificial para la predicción de abandono (churn prediction) trasciende la mera optimización métrica; conlleva responsabilidades éticas inherentes que deben regir tanto el diseño del algoritmo como su aplicación comercial.

En primera instancia, la protección de la privacidad constituyó un requisito no funcional crítico. Dado que el estudio involucra el análisis de hábitos de consumo cultural —datos sensibles que pueden revelar preferencias personales y rasgos de comportamiento—, se aplicaron protocolos estrictos de anonimización. Todos los identificadores únicos de usuario (msno) fueron sometidos a procesos de hashing irreversibles antes del procesamiento. Esta técnica de seudonimización garantiza que los patrones conductuales analizados permanezcan disociados de la identidad civil de los individuos, asegurando la confidencialidad de los sujetos de estudio.

Simultáneamente, se abordó la problemática del sesgo algorítmico y la equidad. Existe un riesgo latente de que los modelos de aprendizaje automático perpetúen inequidades al penalizar injustamente a grupos demográficos con menor huella digital o patrones de uso esporádicos. Un modelo entrenado predominantemente con üsuarios intensivos"podría interpretar erróneamente el comportamiento natural de usuarios de mayor edad o menor alfabetización digital como señales de abandono. Por tanto, es imperativo que las fases de validación incluyan auditorías de equidad para asegurar que las tasas de error (falsos positivos/negativos) se mantengan invariantes a través de atributos sensibles como género y edad, evitando así un impacto dispar.

Finalmente, desde una perspectiva teleológica sobre el uso de la predicción, se establece una distinción ética fundamental. El objetivo legítimo de estos sistemas debe ser la mejora de la experiencia del usuario a través de la personalización y la oferta de valor añadido. Sin embargo, éticamente es inadmisible la instrumentalización de estas predicciones para el diseño de "patrones oscuros"(Dark Patterns). El conocimiento sobre la probabilidad de fuga de un cliente no debe utilizarse para introducir fricción artificial en los procesos de baja ni para manipular la toma de decisiones del consumidor, sino para diagnosticar y reparar las brechas en la calidad del servicio ofrecido.

VII. División de trabajo:

El presente proyecto fue ejecutado bajo una modalidad de trabajo integral y colaborativo, donde ambos integrantes del equipo participaron activamente en todas las fases del ciclo de vida de los datos. Lejos de una segmentación estanca de tareas, se adoptó un enfoque de co-autoría tanto en la implementación técnica como en el análisis teórico, asegurando que cada decisión de diseño fuese producto del consenso y la discusión conjunta. La distribución de responsabilidades se estructuró de la siguiente manera:

Diseño Experimental y Conceptualización (Conjunto): La definición de la variable objetivo, la estrategia de partición temporal (temporal split) y la selección de las arquitecturas a evaluar (XGBoost vs. GRU) fueron acordadas en sesiones de planificación conjunta. Ambos integrantes validaron la pertinencia de las métricas de evaluación (PR-AUC) frente al desbalance de clases detectado.

Ingeniería de Datos y Preprocesamiento (Conjunto):

La construcción del pipeline de datos, que implicó la unificación de fuentes heterogéneas y la limpieza de fechas, se realizó mediante sesiones de programación en pares (pair programming). Ambos colaboraron en la resolución de los desafíos de alineación temporal y en la definición de la lógica para la creación de las variables agregadas y los tensores secuenciales.

Desarrollo y Modelado (Conjunto):

La implementación del código en Python fue un esfuerzo compartido. Mientras se iteraba sobre los hiperparámetros del modelo XGBoost, simultáneamente se depuraba la arquitectura de la red neuronal en PyTorch. La interpretación de los errores de entrenamiento y el ajuste de los mecanismos de atención se realizaron mediante revisión cruzada de código para asegurar la robustez de los algoritmos.

Análisis de Resultados y Redacción (Conjunto): La interpretación de las métricas y el diagnóstico de las limitaciones (como el impacto de la dispersión de datos y el data drift) fueron fruto de un análisis dialéctico entre ambos miembros. La redacción del presente informe refleja esta visión unificada, sintetizando las observaciones y conclusiones

alcanzadas por el equipo en su totalidad.

VIII. Reflexión:

Este proyecto ha servido como un caso de estudio empírico crítico para contrastar la promesa teórica del Deep Learning con la realidad pragmática del análisis de datos transaccionales. Más allá de las métricas de rendimiento, la investigación arroja lecciones fundamentales sobre la naturaleza del modelado predictivo en entornos industriales.

1. **1. La Paradoja de la Complejidad**
   La lección más contundente es que la complejidad arquitectónica no es garante de superioridad predictiva. En escenarios restringidos por ventanas temporales cortas (T=30) y caracterizados por una alta dispersión de datos (sparsity), la ingeniería de características explícita demostró ser un enfoque más eficiente y robusto. Los resúmenes estadísticos actuaron como una forma de çonocimiento experto inyectado", permitiendo al modelo XGBoost tomar decisiones informadas sobre señales consolidadas. Por el contrario, el aprendizaje de representación de extremo a extremo (end-to-end) de la GRU luchó por extraer señal del ruido en secuencias donde el silencio (inactividad) era la norma, reafirmando el principio de parsimonia en la ciencia de datos.

2. **Supremacía de la Calidad de Datos sobre el Algoritmo**
   Se corroboró que la curaduría y alineación de los datos tienen un impacto marginal mayor que la elección de la familia algorítmica. La desalineación temporal detectada entre los registros de actividad (logs) y las fechas de expiración generó vacíos de información que afectaron a los modelos de manera asimétrica. Mientras que los modelos basados en árboles mostraron resiliencia ante estos "huecos"históricos, la red neuronal resultó desproporcionadamente penalizada, colapsando ante la falta de continuidad secuencial. Esto subraya que la sofisticación del modelo nunca puede compensar deficiencias estructurales en el pipeline de datos.

3. **Trabajo Futuro: Hacia la Explicabilidad y el Contexto Extendido.**
   Mirando hacia el futuro, la evolución de este sistema no debe centrarse en "predecir mejor", sino en .[en]tender mejor". Se propone la transición hacia arquitecturas basadas en Transformers (como Self-Attention), lo cual permitiría extender la ventana de observación a 3-6 meses sin incurrir en los problemas de memoria de las RNNs, capturando así la estacionalidad a largo plazo.
   Más importante aún, el objetivo final debe ser transformar el modelo de una çaja negra"predictiva a una herramienta de diagnóstico de retención. Mediante la visualización de los pesos de atención, es posible informar al negocio no solo quién está en riesgo, sino cuándo y por qué se rompió el vínculo con el usuario, operacionalizando la inteligencia artificial como un motor de estrategia proactiva.

IX. Fuentes

[1] WSDM Cup, "KKBox's Churn Prediction Challenge: Can you predict when a subscriber will churn?,"Kaggle, 2017. [En línea]. Disponible: https://www.kaggle.com/c/kkbox-churn-prediction-challenge.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System,ïn Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794. [En línea]. Disponible: https://doi.org/10.1145/2939672.2939785

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, .[Em]pirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,.[a]rXiv preprint arXiv:1412.3555, 2014. [En línea]. Disponible: https://arxiv.org/abs/1412.3555

[4] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?,ïn Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 507-520. [En línea]. Disponible: https://arxiv.org/abs/2207.08815

[5] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, .[A] comparison of machine learning techniques for customer churn prediction,"Simulation Modelling Practice and Theory, vol. 55, pp. 1–9, 2015. [En línea]. Disponible: https://doi.org/10.1016/j.simpat.2015.03.003

[6] A. Vaswani et al., .[A]ttention Is All You Need,ïn Advances in Neural Information Processing Systems, vol. 30, 2017. [En línea]. Disponible: https://arxiv.org/abs/1706.03762

[7] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special Issue on Learning from Imbalanced Data Sets,"SIGKDD Explorations, vol. 6, no. 1, pp. 1–6, 2004. [En línea]. Disponible: https://dl.acm.org/doi/10.1145/1007730.1007733

[8] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey,"Philosophical Transactions of the Royal Society A, vol. 379, no. 2194, 2021. [En línea]. Disponible: https://doi.org/10.1098/rsta.2020.0209