Tecnología Digital VI: Inteligencia Artificial

Olivier Saint-Nom, Valentina de Diego, Lucas Piñera

Trabajo Práctico 2

Este trabajo se basa en la creación de un sistema de predicción de la probabilidad de compra de productos anunciados por usuarios en un retailer en línea. Utilizamos técnicas de aprendizaje automático en un conjunto de datos que incluye información sobre la interacción de los usuarios con anuncios, algunos resultando en compras. A lo largo de este informe, detallaremos los enfoques y estrategias empleadas para la creación del modelo de aprendizaje automático, mencionando también los desafíos enfrentados y las problemáticas que tuvimos que solucionar.

Análisis Exploratorio de Datos

El conjunto de datos contiene el comportamiento de los usuarios en un comercio electrónico y se basa en predecir la conversión de impresiones en órdenes de compra. La columna "`conversion`" indica si una impresión resultó en una compra. Algunos de los atributos más importantes incluyen la disponibilidad de Mercado Pago, la cantidad de stock disponible, el precio promedio de venta, la categoría del artículo, etc. El objetivo es predecir la conversión en las filas donde "`ROW_ID`" no está ausente.

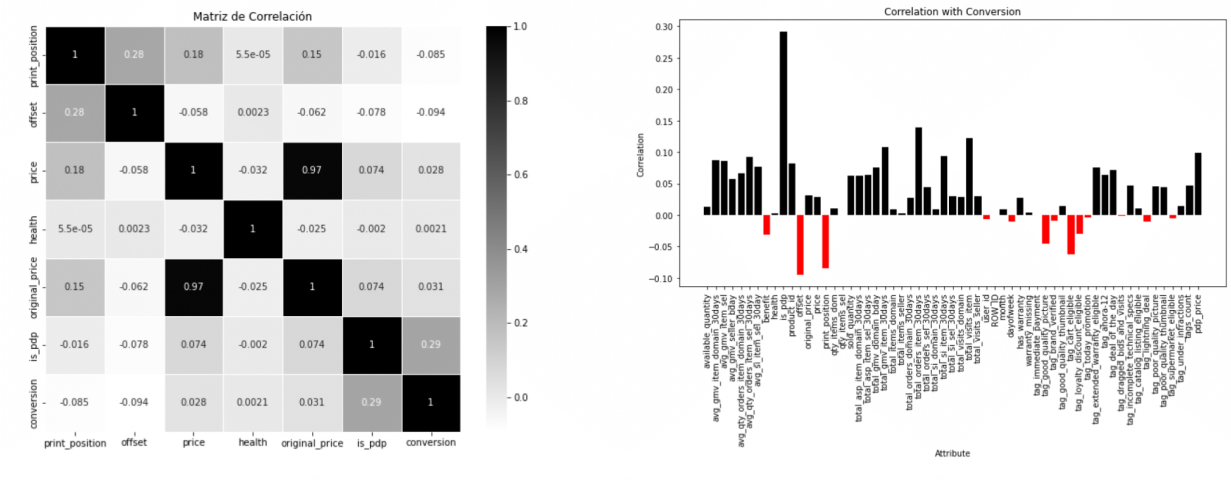

Figura 1 - Matriz de correlación , Figura 2- correlación entre atributos con conversión

La decisión de qué atributos retener y cuáles eliminar de un conjunto de datos fue una parte crítica del proceso de preparación previo a construir nuestro modelo de aprendizaje automático. Al principio del proceso, decidimos eliminar algunos atributos ya que complejizaban el modelo en instancias tempranas, como números de identificación, variables categóricas, atributos con muchos datos faltantes y con poca correlación. Entre estos se encontraban algunos como `deal_print_id`, `full_name`, `user_id`, `site_id`. Luego, a medida que íbamos construyendo el modelo, varios de los atributos que eliminamos pudieron ser reinsertados en el conjunto de datos y procesados correctamente, como `date`, `platform`, `listing_type_id`, y otros.

Variables Utilizadas

En primer lugar, aplicamos la codificación One-Hot a varias variables categóricas, como el tipo de listado, el tipo logístico, la plataforma de venta, el mes y el día de la semana. Esto nos permitió representar estas características como múltiples columnas binarias, lo que facilitó su inclusión en el modelo. Además, para agregar variables categóricas útiles en la predicción, implementamos embeddings de Word2Vec para las variables "`category_id`" y "`title`", lo que permitió al modelo capturar relaciones semánticas entre categorías y palabras en los títulos. Luego quisimos ver relaciones entre algunas variables categóricas, y comenzamos explorando la opción de agregar nuevas *polynomial features* a partir de atributos como "`print_position`", "`price`", "`health`", "`offset`", "`original_price`" e "`is_pdp`". La nueva variable "`price * is_pdp`" fue la más beneficiosa para el modelo, captando la interacción entre ellas. Otras combinaciones también fueron incorporadas pero no mostraron una mejora igual de notoria.

Además, incorporamos *tags* para mejorar la capacidad predictiva del modelo. Primero realizamos un procesamiento de las etiquetas, lo cual nos permitió crear nuevas características binarias que indicaban la presencia (registrado como "`tags_count`") o ausencia de tags específicos. Al incluir estas características, logramos que el modelo aprenda patrones y relaciones entre los tags y la conversión de impresiones en órdenes de compra.

Conjunto de Validación

Elegimos dividir el conjunto de datos mediante el método de holdout set, ya que computacionalmente es el mejor para casos en los que el conjunto de datos es suficientemente grande. Al dividir los datos en un conjunto de entrenamiento y de evaluación, podemos entrenar nuestro modelo en una parte de los datos y evaluar su rendimiento en la otra. Esto nos permite obtener una estimación fiable del rendimiento del modelo en datos no vistos, lo cual es esencial para tener un modelo preciso. Si bien intentamos con la técnica k-fold cross-validation, es computacionalmente costosa y no nos termina brindando una mejor predicción.

Cuando decidimos cómo dividir nuestros datos en conjuntos de entrenamiento y validación, hicimos un análisis manual evaluando diferentes porcentajes, desde un 10% hasta un 30% de datos para validación. Finalmente optamos por la división estándar recomendada del 20/80 para evitar tanto un modelo sobreajustado como uno subajustado, logrando un equilibrio adecuado. Esto lo logramos utilizando la columna "`ROW_ID`", donde las filas con "`ROW_ID`" presentes se reservaron como nuestro conjunto de evaluación, mientras que las filas sin "`ROW_ID`" se utilizaron para entrenar el modelo.

Al principio, cuando estábamos explorando los datos, aplicamos la codificación one-hot a las categorías antes de dividir nuestros datos en conjuntos de entrenamiento y validación. Sabíamos que esto podría causar un problema de *data leakage*, pero lo hicimos de todas formas para obtener mejores resultados desde el principio. A medida que nuestro modelo mejoró, en lugar de aplicar one-hot encoding antes de dividir los datos, primero dividimos los datos, y luego aplicamos cualquier transformación o ajuste necesario. Esto nos aseguró que haya menos filtración de información de un conjunto al otro.

Modelos Predictivos y Búsqueda de Hiperparámetros

En nuestro proceso de búsqueda y selección de hiperparámetros, primero aplicamos un enfoque *greedy* para explorar el espacio de hiperparámetros y encontrar un rango inicial en donde buscar. También usamos lo que vimos en la práctica para reconocer estos rangos iniciales de valores útiles. Luego, decidimos usar el proceso de optimización de hiperparámetros con la biblioteca Hyperopt. Esto incluía optimizar `max_depth`, `learning_rate`, `gamma`, `reg_lambda`, `subsample`, `min_child_weight`, `colsample_bytree`, y `n_estimators`. El proceso se basó en generar una función objetivo (*Función de pérdida* = 1 - AUC-ROC) que entrena el modelo utilizando los hiperparámetros candidatos y evalúa el área bajo la curva ROC. Luego, minimizamos la pérdida (con la función `fmin` de Hyperopt) para optimizar estos hiperparametros a través del proceso de evaluación por bayesiano. Finalmente, tomamos las mejores combinaciones identificadas por Hyperopt y las devolvimos al enfoque greedy, lo cual nos permitió seleccionar el modelo que obtuvo el mejor rendimiento en términos del AUC-ROC.

En cuanto a los algoritmos predictivos que fuimos usando en la exploración, empezamos implementando un Random Forest en los datos. Luego, una vez que teníamos un modelo más completo y estábamos en busca de mejora, hicimos una transición hacia el uso de XGBoost. XGBoost ofrece capacidades para capturar patrones más complejos en los datos y proporciona herramientas más avanzadas, lo cual nos dio mejoras rápidamente en nuestras predicciones.

Importancia de Atributos

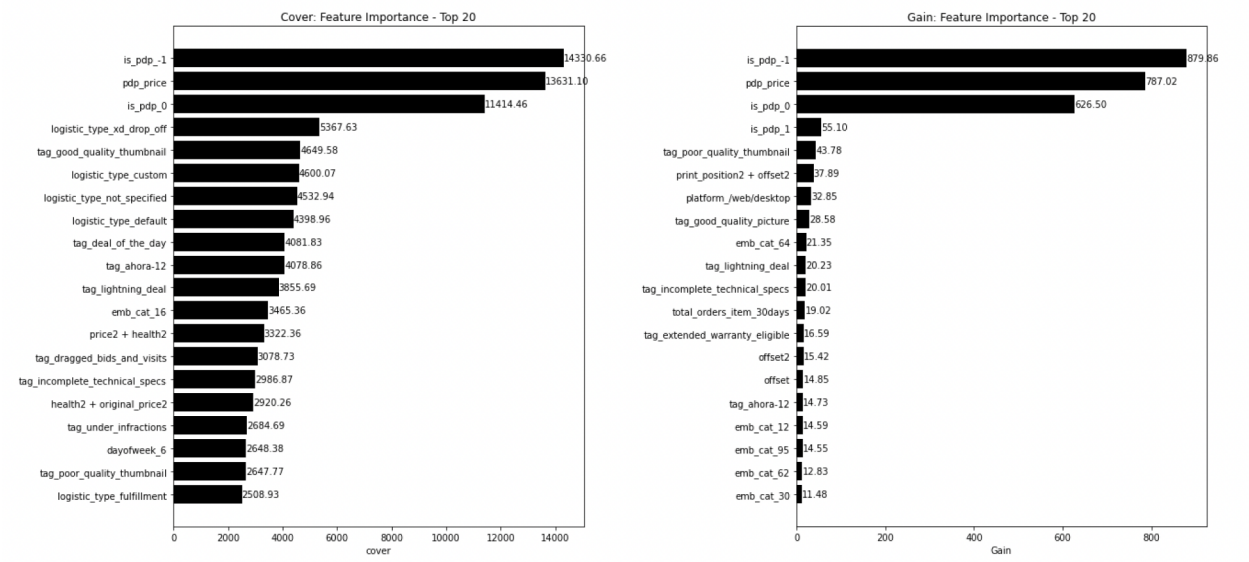

Figura 3: Feature importance (Cover) , Figure 4: Feature importance (Gain)

En el análisis de importancia de atributos utilizando el modelo XGBoost visualizado en las figuras 3 y 4, pudimos ver qué características influyen más en la predicción del modelo. Entre los atributos más destacados, encontramos "`pdp_price`", "`is_pdp`", "`logistic_type_custom`" y "`tag_good_quality_thumbnail`".

Estos resultados son importantes en la vida real para quienes diseñan anuncios de venta. Por ejemplo, la importancia de "`pdp_price`" sugiere que el precio de un producto en su página de destino es un factor crítico en la decisión de compra de los usuarios. Recomendariamos asegurarse de que los precios sean competitivos y atractivos en las páginas de destino podría mejorar la conversión. La presencia de "`is_pdp`" también resalta la importancia de tener páginas de destino buenas y atractivas que motiven a los usuarios a entrar y comprar. "`Logistic_type_custom`" puede indicar que ofrecer opciones de envío personalizadas podría ser clave para la estrategia de venta. "`tag_good_quality_thumbnail`" nos dice que la las imágenes de alta calidad y atractivas en los anuncios es muy importante y aumentar las posibilidades de conversión. También encontramos que los títulos y la categoría son aspectos relevantes (en menor escala), por lo que aconsejamos hacer una investigación previa de cuáles son las categorías más frecuentes y las palabras “trending” para captar una mayor demanda.

Conclusiones

En este trabajo, exploramos el comportamiento de los usuarios en el proceso de compra y aplicamos conceptos aprendidos en clase. Nuestra metodología consistió en un enfoque iterativo, donde incorporamos herramientas y estrategias a medida que observamos mejoras en el desempeño. Iniciamos con un modelo simple, enfocado en pocos atributos y evitando la complejidad. Además, exploramos alternativas en la ingeniería de datos para incorporar información de manera efectiva en nuestro modelo. El salto más significativo se produjo al introducir `XGBoost` sumado a la constante optimización de hiperparametros con `Hyperopt`, lo que marcó gran cambio en los resultados.

En cuanto a las debilidades en el análisis, para empezar podríamos haber aislado ciertos factores para evaluar su impacto individual en el modelo. También podríamos haber aplicado una ingeniería de atributos más exhaustiva y crear más atributos que capturaran mejor los patrones de comportamiento. Por otro lado, somos conscientes que dejamos un data leakage en los títulos de los anuncios, lo cual también podría ser un área de mejora. También podríamos haber abordado de manera más efectiva el manejo de valores faltantes (estrategias de imputación de datos) y el balanceo de clases (técnicas de sobremuestreo o submuestreo para igualar el número de ejemplos en ambas clases) para mejorar la calidad del modelo. Estas son áreas en las que podríamos haber mejorado la robustez y la precisión de nuestras predicciones.