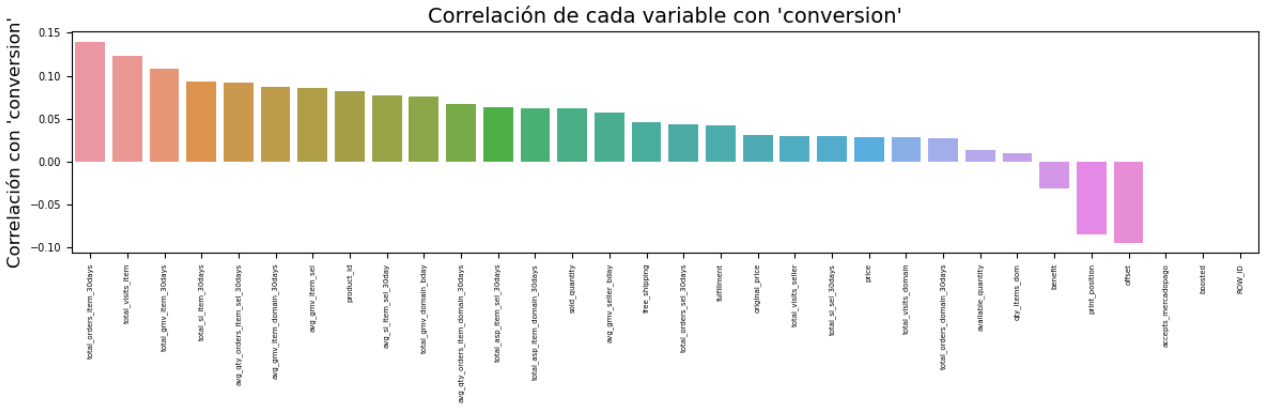
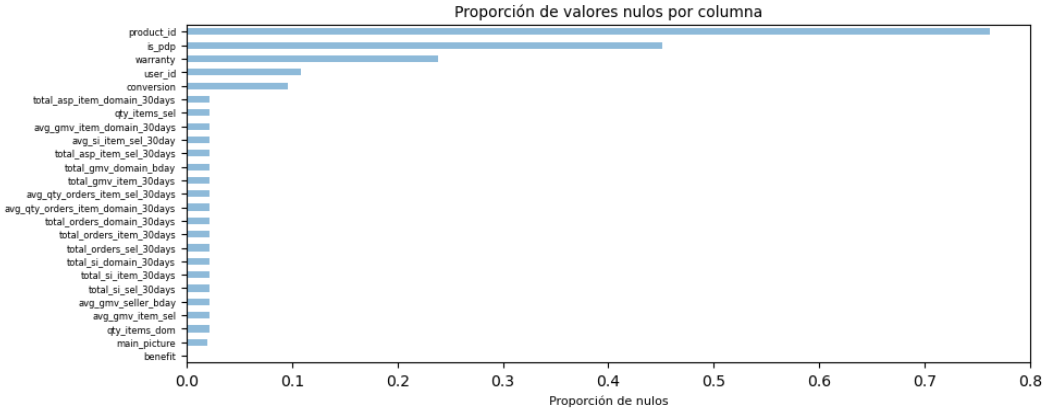
TD VI TP II: Inteligencia Artificial

Análisis Exploratorio y Feature engineering

Correlación de variables iniciales:
Con el fin de comenzar el análisis exploratorio de variables se optó por analizar la correlación de las variables (sin nulos) con respecto a la variable a predecir *conversión*. A partir de esta figura podemos tener una noción previa de las variables y sus importancias.



La presencia de valores nulos en el conjunto de datos es notable. Por ello, es esencial examinar la proporción de datos faltantes en cada variable. Esta revisión nos permitirá diseñar estrategias adecuadas para su manejo en las siguientes etapas del análisis.



Eliminación de variables inválidas:
La descripción del dataset determinaba algunas de las variables como inválidas y por lo tanto se decidieron eliminar de la base las variables *benefit*, *decimals*, *etl_version* y *rn*.

Análisis de nulos:
Es relevante mencionar que identificamos 4,365 registros con numerosos valores nulos en nuestro dataset. Aunque los modelos que empleamos gestionan eficazmente los valores nulos, optamos por explorar la imputación de estos mediante la media o la mediana. Sin embargo, esta estrategia no derivó en mejoras significativas, por lo que finalmente decidimos permitir que el modelo maneje los valores nulos directamente.

La variable *is_pdp* tiene una gran proporción de nulos, por lo que se decide hacer OHE separando is_pdp_true, is_pdp_false e is_pdp_nan ya que los valores que presentan la columna de is_pdp_nan NUNCA realizan la conversión como se muestra en la tabla.

conversion	0.0	1.0
is_pdp		
False	82360	16532
True	391	212
nan	81266	0

Análisis de variables numéricas y booleanas:
Se procede a eliminar las variables *accepts_mercadopago* y *boosted* con un único valor posible, porque lo único que aportan al modelo es tiempo computacional, ¡que no queremos!

Además, la variable *total_gmv_domain_bday* es una transformación lineal de la variable *total_gmv_domain_30days*, por lo que también se elimina de la base.

Por otro lado, se agregó la variable *discount* producto de una combinación de las variables *original_price* y *price* ya que logra encapsular información relevante sobre los descuentos aplicados, además de ser más fácil de interpretar. Se eliminaron las variables originales con el fin de prevenir la multicolinealidad, reducir dimensionalidad y acelerar el análisis disminuyendo el tiempo computacional.

Análisis de variables categóricas:

Se procede a eliminar la variable *site_id* con un único valor posible, porque lo único que aportan al modelo es tiempo computacional, ¡que no queremos!

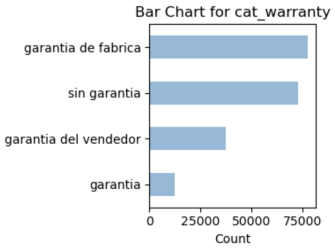
La variable *main_picture* representa la imagen del producto, un factor que intuitivamente se percibe como crucial en la decisión de compra. Inicialmente, contemplamos la idea de diseñar una variable booleana que indicara la presencia o ausencia de una imagen. Sin embargo, al evaluar las correlaciones, nos dimos cuenta de que los valores nulos en *main_picture* no necesariamente se debían a productos sin imágenes, sino que podrían surgir de fallos en la recopilación de datos. Al explorar la plataforma de Mercado Libre, confirmamos que no se permite publicar productos sin imágenes, invalidando nuestra hipótesis inicial. Además, aunque intentamos diferentes técnicas de codificación, no logramos mejorar el rendimiento del modelo. Por ello, decidimos excluir main_picture como atributo en nuestro modelo final.

Continuando con el análisis, identificamos que la variable *platform* se puede dividir en si el usuario está desde la web o mobile. Por lo que se originó la nueva variable *platform_web* que identifica si la impresión fue hecha desde la web o no.

Análisis de variables de texto:

Para continuar con el análisis notamos que la variable *warranty* es de tipo objeto pero en realidad está detallando si tiene garantía (cuanto tiempo) o no. Por lo tanto, intentaremos modificar esa variable y analizaremos si la garantía de los productos tiene o no una correlación con la conversión. Primero se formatea para intentar sacar la mayor cantidad de duplicados:

1. Convertir a Minúsculas
2. Eliminar tildes y caracteres especiales
3. Eliminar espacios dobles
4. Eliminar espacios al principio y al final de la oración.
5. Categorizar la variable en 4 valores que se muestran en el gráfico

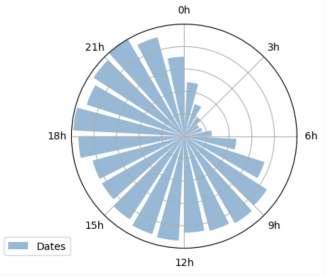


En el caso de los *tags*, son etiquetas que se le asignan al producto dentro de un rango de opciones. Por los que decidimos hacer OHE con las diferentes etiquetas.

Análisis de variables temporales:

Las variables temporales *date* y *print_server_timestamp* pueden influir mucho en el comportamiento del consumidor. Se debe evaluar y tener en cuenta el momento en el que están visitando la página ya que puede determinar si se realiza o no la conversión. Es por eso que se realiza un encoding de estas variables.

Al fusionar las funciones del seno y coseno, es posible caracterizar una variable de naturaleza cíclica. Este método se adopta para asegurarse de que las variables conserven su naturaleza cíclica en lugar de una interpretación lineal. Tomando como ejemplo las horas del día: la hora 00 y la hora 23, aunque parecen distantes en una escala lineal, en realidad están próximas cíclicamente. Al realizar un "encoding" mediante el cálculo de sus respectivos valores de seno y coseno, se preserva esta relación cíclica. Este análisis se aplicará a las horas, días de la semana y días del mes. No se considerará la variable de los meses dado que solo contamos con datos de dos meses.



Encoding de variables:

Realizamos distintas técnicas de Encoding en las variables categóricas. El tipo de codificación a aplicar ('OHE', 'LE', 'OE', 'FE').
La mejor combinacion de tecnicas se vio al utilizar One Hot Encoding con las variables: 'listing_type_id', *'logistic_type'*, 'platform', 'cat_warranty', 'is_pdp' y Frecuency Encoding con las variables: *'category_id'*, 'domain_id', 'item_id', 'uid' y 'deal_print_id'.

Conjunto de Validación

Elegimos la rama del *K-Fold Cross-Validation* en lugar de otros enfoques debido a sus ventajas en la evaluación de modelos. Este proporciona varias rondas de entrenamiento y evaluación, aportando una estimación más robusta del rendimiento del modelo. Además, permite utilizar todos los datos tanto para entrenamiento como para evaluación, maximizando así el aprovechamiento de los mismos.

Posteriormente, dentro de la rama del K-Fold, optamos por *Stratified K-Fold*. El conjunto de validación estratificado se crea para asegurar una distribución proporcional de las clases originales en los pliegues, lo que garantiza evaluaciones justas y representativas del modelo. Dado que en nuestro caso trabajamos con un conjunto de datos desequilibrado, el Stratified K-Fold ayudó a obtener una estimación más precisa de la capacidad de generalización, para evitar sesgos en la evaluación debido al desbalance de clases.

Modelos Predictivos

Tras concluir el análisis exploratorio de datos y decidir cómo dividir el conjunto de validación, avanzamos hacia la experimentación y configuración de modelos. Diversos modelos y técnicas para la afinación de hiperparámetros fueron empleados a lo largo del proceso. Los modelos que exploramos incluyen XGBoost, Random Forest y LGBM, mientras que para la optimización de hiperparámetros recurrimos a técnicas como Random Search, Grid Search y Hyperopt.

XGBoost

Inicialmente, durante las primeras etapas del análisis exploratorio, nos inclinamos por el modelo XGBoost junto con Random Search. Esta combinación nos proporcionó una performance de 0.88977 en el tablero público de Kaggle y 0.89269 en el privado. Lamentablemente, perdimos el registro de su rendimiento en nuestro conjunto de validación debido a sucesivas pruebas.

A medida que avanzábamos en la ingeniería de características y codificación, esperábamos mejoras en el rendimiento. Sin embargo, para nuestra sorpresa, la performance no estaba mejorando.

Experimentos con técnicas como word2vec aplicadas a *title*, *tags* y *name*, la imputación de valores faltantes mediante la media o mediana, y ciertos enfoques de codificación de variables no parecían incrementar la efectividad de nuestro modelo.

Random Forest

Dada la falta de avance en la precisión con XGBoost, exploramos el modelo Random Forest utilizando Hyperopt para la sintonización de hiperparámetros. No obstante, al no observar mejoras tangibles, decidimos abandonar este enfoque rápidamente.

LightGBM

Frente a los desafíos con la ingeniería de variables categóricas y ante la cercanía de la fecha límite de la competencia, nos inclinamos hacia LightGBM. Este modelo es ampliamente reconocido en las competencias de Kaggle, y es especialmente útil en contextos donde la eficiencia en tiempo y memoria es esencial, ya sea por el volumen de datos o por limitaciones de recursos. Para mejorar su adaptación a nuestro conjunto de datos desbalanceado, configuramos el modelo con los siguientes parámetros:

- class_weight='balanced'
- scale_pos_weight = sum(y_train == 0) / sum(y_train == 1)
- boost_from_average=False

Con este modelo, logramos una mejora en la performance, pero aún sentíamos que podíamos alcanzar más.

Una vez concluida la competencia, nos propusimos alcanzar nuevas metas y nos embarcamos en una introspección para determinar qué nos impedía superar el 0.89 en el marcador público de Kaggle. Pronto identificamos que habíamos subestimado variables como uid y deal_print_id. Aunque en un primer momento parecían irrelevantes para la predicción, más tarde comprendimos que sí lo eran.

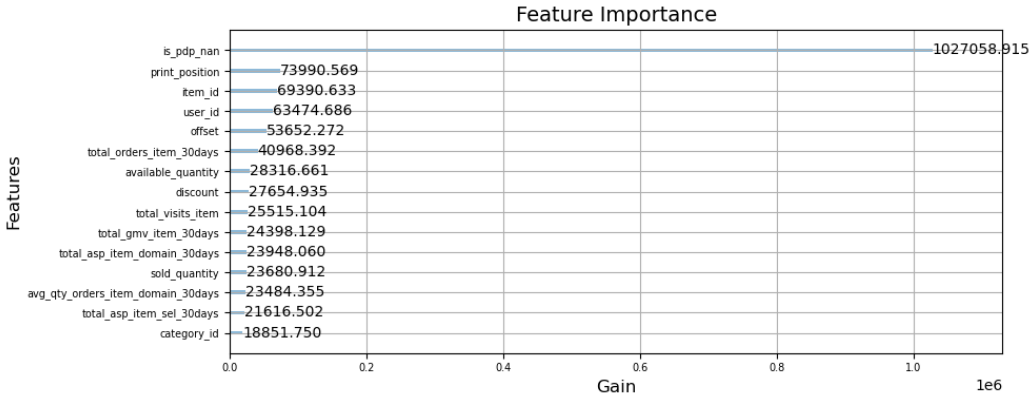
XGBoost con Grid Search: Con el fin de mejorar nuestro Score probamos XGBoost con Grid Search y early stopping de parámetros alrededor de los valores de los best params de lo que nos dio con Random Search. El modelo nos llevó a un gran score de 0.89216 en el tablero público de Kaggle y 0.89587 en el privado.

LightGBM: Al experimentar con el modelo LightGBM, incorporando estas mejoras, obtuvimos una performance de 0.89019 en el marcador público de Kaggle y un 0.8961 en el privado. Estos resultados nos hubieran ubicado en el tercer puesto si hubiéramos presentado nuestras conclusiones a tiempo.

Aunque nos pesa reconocer que un desliz en el análisis limitó nuestra performance, la competencia nos brindó valiosas lecciones y estamos ansiosos por enfrentar nuevos desafíos en futuras competiciones.

Importancia de Atributos

La importancia de atributos de LightGBM es la siguiente:



Es evidente la relevancia de la variable *is_pdp_nan* en nuestro modelo. Tal observación concuerda con análisis previos, donde identificamos que todos los registros con valores nulos en *is_pdp* no culminan en una conversión.
Además, las variables *print_position* y *offset* también destacan. Cuando un producto está listado en posiciones iniciales, se percibe como más relevante, quizá incluso asociado con calidad o popularidad. Esta disposición inicial puede actuar como un indicador de confiabilidad, llevando a los usuarios a creer que es una elección preferida por otros.
Sumado a esto, navegar a través de múltiples páginas y evaluar un gran número de productos puede ser agotador para los usuarios. Por lo tanto, muchos optan por seleccionar productos de las primeras páginas para reducir la carga cognitiva.
Es esencial también reconocer la trascendencia de variables como *is_pdp_nan* y *discount* en el modelo. Sin el adecuado procesamiento y las optimizaciones que aplicamos, estas variables, entre otras, posiblemente no tendrían el peso que tienen ahora en la predicción. La ingeniería de características fue, sin duda, un paso decisivo para realzar la efectividad de nuestro modelo.

Consejo a diseñador

Cuando se trata de publicar un anuncio eficaz en este reconocido sitio de compras en línea, ciertos elementos pueden marcar una notable diferencia.

En primer lugar, la ubicación de su producto en la página es crucial. Así como un artículo en una vitrina de tienda física capta más atención si está al frente, en el mundo digital, la visibilidad en las primeras páginas aumenta considerablemente las posibilidades de venta. Por ende, es vital que se esfuerce por asegurarse de que su producto sea visible en las páginas iniciales y no quede relegado a lugares más distantes de la vista del usuario.
En segundo lugar, un atractivo descuento puede actuar como un potente imán para los compradores. Los datos indican que ofertas más generosas tienden a tener mayores tasas de conversión. Considerando la naturaleza competitiva del mercado en línea, un descuento significativo podría ser el diferenciador que impulse al usuario a elegir su producto por encima de otros.

Finalmente, es benéfico trabajar en la popularidad de su producto. Un artículo con numerosas ventas y pedidos sugiere a los clientes potenciales que es deseable y confiable. Esta percepción positiva, respaldada por otros compradores, puede ser un fuerte incentivo para que nuevos clientes se decidan por su producto.

En el curso de nuestro análisis, identificamos un área de incertidumbre. La razón detrás de los valores nulos en la variable *is_pdp* permanece enigmática, lo que nos impide formular una hipótesis robusta sobre por qué todos estos valores nulos no culminan en una conversión. Esta falta de claridad limita nuestra capacidad para proporcionar recomendaciones precisas a nuestro diseñador sobre cómo incrementar la probabilidad de conversión del producto.