Minería de Datos – Trabajo Práctico
Master in Management + Analytics, Escuela de Negocios,
Universidad Torcuato Di Tella

Contact Prediction Challenge

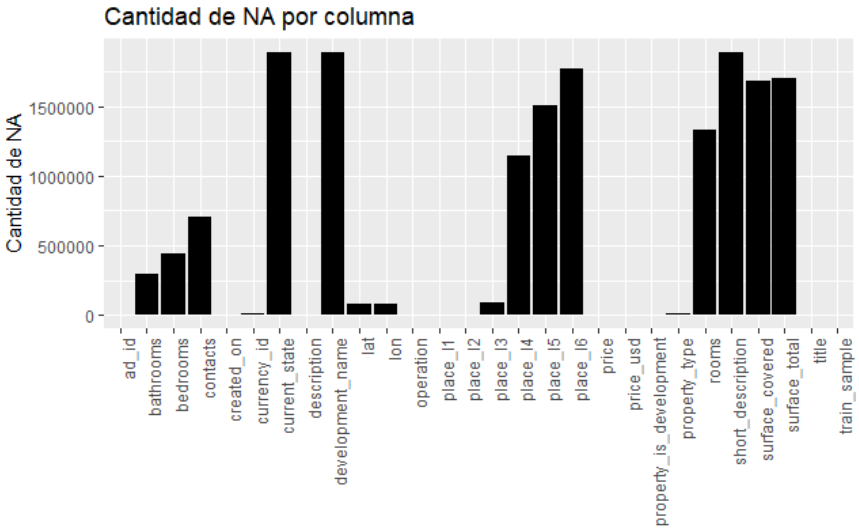
Informe del desarrollo del modelo de predicción para las publicaciones de Julio, Agosto y Septiembre de 2023.

- Alumnos: Mauro Bertini y Lucas Veteikis
- Sección 1
- Nombre del equipo: Properatis

1. Análisis exploratorio de datos

Valores nulos

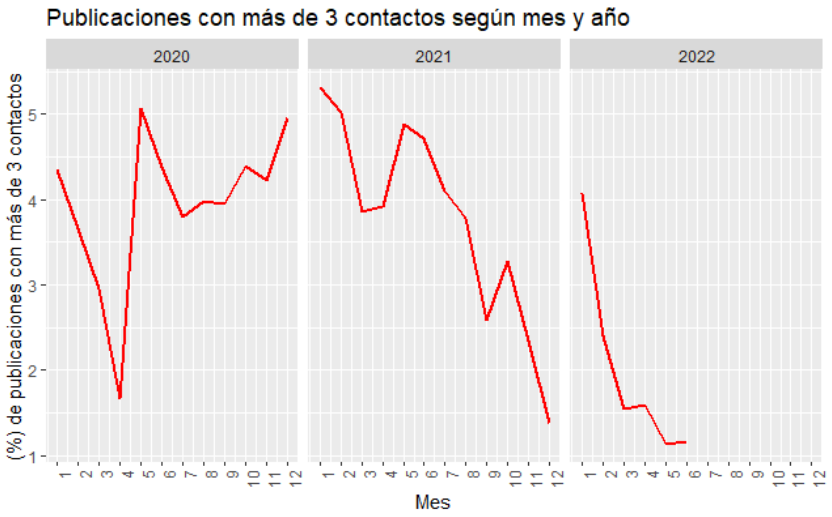
Para comenzar con el análisis exploratorio de datos, empezamos analizando la cantidad de valores nulos por columna. Esta información resulta útil para evitar ciertas variables en modelos de aprendizaje automático que no aceptan valores nulos, como podría ser Kmeans. También es importante tener en cuenta la cantidad de valores nulos a la hora de crear nuevas variables, ya que si usamos variables para ingeniería de atributos que ya de por si tienen muchos valores nulos, esta característica se verá replicada en las nuevas variables y afectará a la calidad de estas.



En este sentido, las variables “current_state”, “development_name”, “place_l6” y “short_description” son las que mayor cantidad de valores nulos tienen. Estas variables se deberían evitar para la creación de nuevas, o para algunas de ellas se podría encontrar alguna forma de imputar los valores faltantes.

Factor temporal

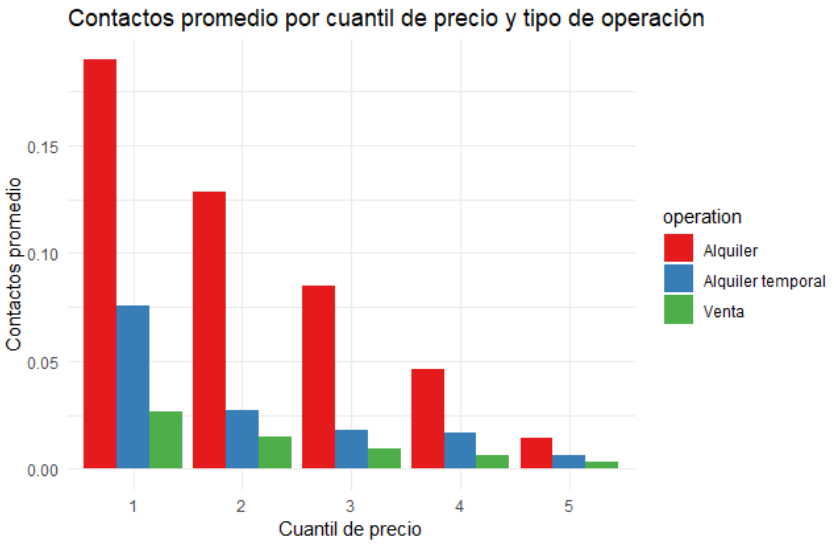
Por los cambios en el macroentorno, desde el primer momento intuimos que el factor temporal podría influir, entonces, la pregunta que intentamos buscar respuesta es, ¿Vale la pena considerar todos los datos a la hora de modelar o conviene solo tomar una parte?. Esta pregunta la respondimos probando distintos modelos, pero también realizamos un análisis exploratorio del factor temporal y cómo influía en la variable de respuesta.



Como se puede observar, las proporciones de publicaciones con más de 3 contactos en cada mes no siguen una tendencia clara a lo largo de los años bajo evaluación. Sin ir más lejos, el mes de mayo tiene valores máximos en 2020 y 2021, pero en 2022 tiene un valor mínimo. Estas diferencias se repiten a lo largo de los meses, lo que podría sugerir que los datos previos a cierta fecha podrían afectar al modelo en vez de mejorarlo, lo que nos hizo contemplar la posibilidad de trabajar con modelos solo con los meses más recientes.

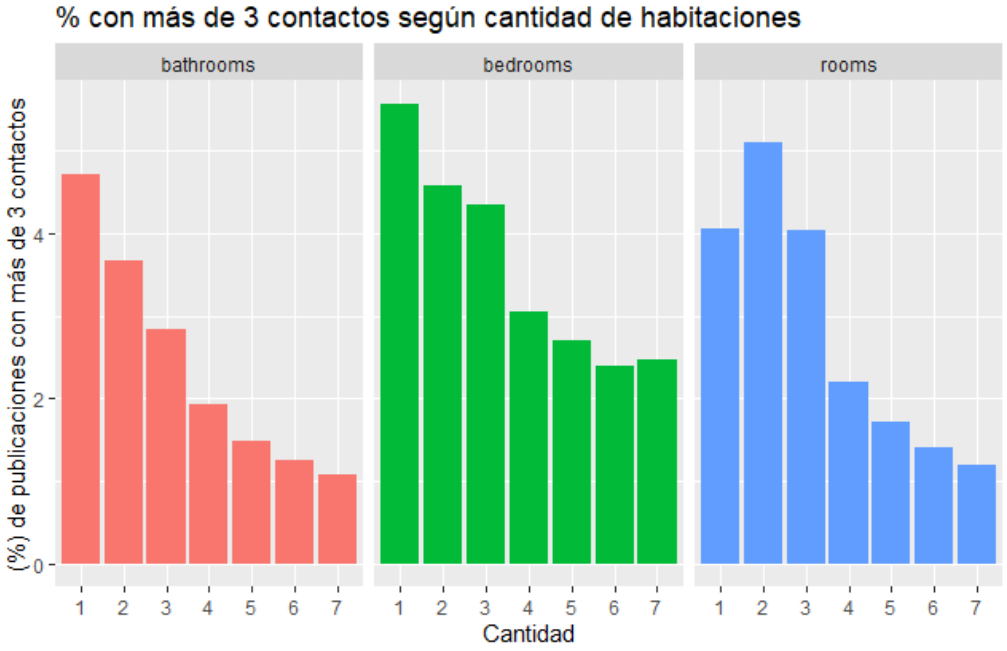
<u>Variables que inciden en la variable de respuesta</u>

- Precio en dólares



Como se puede observar en el gráfico, el precio resulta una variable determinante para determinar si una publicación recibirá o no más de 3 contactos, siendo las publicaciones más baratas las más contactadas con mucha diferencia. También es interesante notar como la diferencia entre cuantiles de precio disminuye según el tipo de operación, indicándonos que la demanda es más sensible al precio cuando se trata de un alquiler, y menos cuando se trata sobre una venta.

- Cantidad de habitaciones y baños



Se ve una distribución similar a lo largo de las tres variables, donde a mayor cantidad de habitaciones, dormitorios o baños menor es el porcentaje de publicaciones con más de 3 contactos. Esto también nos da el indicio de que estas tres variables podrían ser buenas predictoras de si una publicación va a tener más o menos de 3 contactos.

2. Selección de variables/Ingeniería de atributos probada

En un principio, se seleccionaron todas las variables para hacer el análisis exploratorio de datos realizado en el punto anterior, a partir del mismo, se tomaron diferentes decisiones:

Modificaciones o eliminación de columnas del dataset

- La columna **“ad_id”**, que tiene poder predictivo por su relación temporal (las publicaciones tienen mayor ad_id a medida que pasa el tiempo), fue convertida a numérica y conservada para el modelo. Se la convirtió a numérica ya que si se hubiese hecho OHE (One Hot Encoding) sobre esta variable como un factor, se corre riesgo de overfitting.
- Las columnas **“description** y **title”**, por su naturaleza de ser únicas para cada publicación, no podrían ser pasadas a OHE, entre otras cosas, por riesgo de overfitting. Para estas columnas, se crearon variables dummies que marque 1 si la descripción o el título tenían cierta palabra, y 0 si no la tenían. Posteriormente, se eliminaron ambas variables.
- Las variables **“current_state”**, **“development_name”** y **“short_description”** fueron eliminadas por la cantidad de valores nulos que contenian. A su vez, se analizaron para evaluar su capacidad de transformación en otras variables predictivas pero su contenido no traía información rica para el análisis.
- Las variables **“place_l5”** y **“place_l6”** tienen mayores valores nulos dado que su especificidad parece hacer referencias a barrios privados o sitios más exclusivos.

Bajo esa premisa, podría ser una variable predictora interesante, por lo que se modificaron para que tomaran una variable binaria (1 y 0) y poder captarla en el modelo.

Creación de nuevas variables

- Se crearon nuevas variables a partir del precio, las habitaciones y las superficies, tales como "**price_per_room**", "**price_per_surf_cov**", "**bathroom_proportion**" o "**bedroom_proportion**", que son ratios de variables ya presentes en el dataset.
- Se realizó una transformación logarítmica sobre el precio en dólares, dejando la variable de precio original en el dataset e incluyendo la nueva "**log_price_usd**"
- Se creó una variable relacionada con el precio, que representa el ratio entre el precio en usd y el precio promedio de la localización geográfica (l1, l2 o l3). Si este valor fuese >1, significa que la propiedad es más cara que el promedio de su localización.
- Se crearon variables dummy a partir de la presencia de ciertas palabras en la descripción o título de la publicación, tales como "**parrilla**", "**jardin**" o "**playa**".

Variables creadas desestimadas

- En los intentos donde utilizamos la base de datos completa, intentamos agregar una variable dummy llamada **pandemia**, que tenga 1 en las publicaciones que fueron hechas en épocas de cuarentena estricta, como lo puede ser marzo a mayo de 2020. Esta variable no tuvo el efecto esperado y fue desestimada una vez que decidimos recortar la base de datos.

3. Validación de modelos

Se tomó la decisión de utilizar AUC como medida de performance del modelo dado que performa mejor con datasets desbalanceados (como es el caso de este data set). Las decisiones en la subida de los modelos a Kaggle se tomaron en base a la mejora de esta medida en validación.

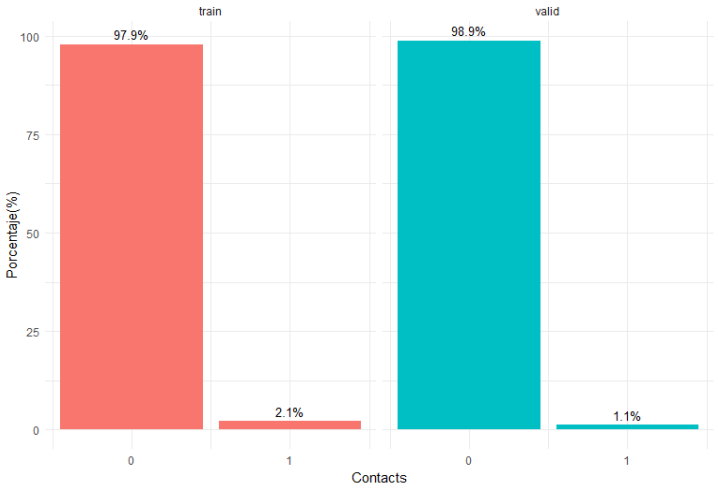
Para la construcción de un conjunto de validación apropiado, tomamos la decisión de filtrar el dataset para que nuestro conjunto de validación sean los últimos meses, ya que creemos que es importante medir que nuestro modelo prediga bien las publicaciones más recientes, que son las más parecidas sobre las que luego el modelo se medirá en el evaluation set.
Esta idea surgió gracias a la disparidad que tenían los resultados de nuestros modelos sobre los resultados que se mostraban en Kaggle. Nos dimos cuenta de que no nos servía de nada una métrica que nos indique que un modelo performa bien sobre datos viejos, que son muy dispares a los meses que se medirán en la evaluación (julio a septiembre de 2022).
A partir de esto, detectamos que nuestros errores en la generación de los sets de validación de los modelos anteriores generó que se tomaron dos decisiones que con el devenir de la cursada se probaron no tan efectivas:

- Realizar un muestreo aleatorio de nuestro dataset para definir el set de validación : Esto agregaba variabilidad e información de datos anteriores (en los casos en los que tomábamos el dataset este punto se acentuaba) dando como resultado un set

de validación que no reflejaba una comparación representativa con lo que serían los meses siguientes.

- Tomar la mejora en AUC como un factor relevante ante ese muestreo: Nuestros modelos que arrastraban el error en el set de validación mostraron consistentemente una mejora en AUC que no se refrendaba en la performance en Kaggle.

Un último factor que podría generar esa diferencia era que Kaggle solo hacía una muestra del 30% de los resultados para asignar el puntaje, pero esto era un factor fuera de nuestro control.
A partir de todo este análisis, concluimos que la mejor forma de armar el conjunto de validación era tomando los últimos dos meses del dataset (Mayo y Junio 2022). Siguiendo este conjunto de validación, la proporción de la clase de validación es similar a los datos de entrenamiento del modelo.



4. Algoritmos usados

El proceso de selección de algoritmos se realizó en base al avance de las clases, subiendo en complejidad a medida del avance de la cursada:

Naive Bayes

Se realizó un intento utilizando Naive Bayes con un archivo de Argentina de enero de 2020, dividiéndolo en train y validation set, y utilizando un suavizado de 10 como hiperparámetro. Lógicamente, este intento se utilizó a modo de prueba y no se buscó predecir el set original de la manera más efectiva, aún así la performance alcanzó un 75% de precisión.

Árboles de decisión

Para el primer submit como modelo básico, se tomó una muestra de los 3 meses anteriores, asumiendo que los valores más recientes tendrían un mayor poder predictivo, y para armar el conjunto de validación se utilizó el método de k-fold cross-validation, probando 3, 5 y 10 folds, obteniendo la mejor performance con 10 folds. El modelo se entrenó con las variables predeterminadas del dataset y según la métrica ROC, resultó en una predicción con un puntaje del 0.92 en Kaggle.

Xgboost

Se realizaron múltiples intentos para mejorar la performance del modelo XGBoost. Los primeros intentos involucraron optimización manual de hiperparámetros, agregando variables y ajustando el muestreo de datos. Un intento que fue el mejor con hiperparámetros ajustados manualmente, fue el resultado de utilizar sólo datos de mayo y junio de 2022.

Se realizaron pruebas adicionales con diferentes combinaciones de meses y años, pero las conclusiones indicaron que los datos más recientes producen mejores resultados que los anteriores, posiblemente debido a la influencia de la pandemia en la economía durante 2020 y 2021 o alguna otra variable externa o de negocio que explique la diferencia que hay entre los datos de los distintos años.

Por último se empleó una búsqueda aleatoria (random search) para optimizar hiperparámetros. El mejor modelo resultó de un xgboost con este tipo de búsqueda de hiperparámetros. Estos últimos modelos fueron los que tuvieron una tarea de ingeniería de atributos más fuerte, factor que también influye en que sean los mejores. El esquema para llevar a cabo un random search lo más eficiente posible fue:

- Primero, establecimos unos hiperparametros muy amplios y entrenamos 100 xgboost.
- Posteriormente, evaluamos los resultados del modelo y nos quedamos con el mínimo y el máximo de los hiperparametros que superaron cierto rendimiento en validation y estuvieron por debajo de cierto rendimiento de training (para eliminar aquellos que overfittearon). Para realizar esto se definió la función "**get_new_min_max"**, disponible en el archivo de funciones.
- Con estos nuevos mínimos y máximos "óptimos" de cada hiperparámetro, entrenamos otros 20 modelos.
- Seleccionamos el mejor de estos últimos 20 modelos, y lo entrenamos con todos los datos para posteriormente predecir.

Finalmente, el mejor modelo fue con los siguientes hiperparametros: nround: 271, max_depth: 9, ETA: 0.1199, gamma: 4.5335, colsample_bytree: 0.3221, min_child_weight: 11.7117, subsample: 0.8261.

5. Tiempo asignado

La distribución del tiempo total utilizado se repartió de la siguiente manera:

- Análisis exploratorio de datos = Se utilizó el 15% del tiempo dedicado al trabajo práctico con la intención de entender la composición de los datos y entender aquellas variables que indicaran insights de su capacidad predictiva
- Ingeniería de atributos = Se dedicó un 35% del tiempo empleado. Se profundizó en este punto luego de entrar en una meseta en la performance en selección de hiperparámetros. Allí se crearon los ratios, las variables dummies con palabras claves y la transformación logarítmica del precio que probaron mejorar la performance del modelo.
- Selección y validación de modelos = se utilizó el 50% del tiempo con el objetivo principal de probar distintas combinaciones de atributos e hiperparámetros que mejoraran la performance en la competencia. También se ocupó el tiempo en encontrar el set de validación óptimo luego de encontrar inconsistencias en la subida de los datos en Kaggle.