Predicción de precios de inmuebles en C.A.B.A mediante técnicas de aprendizaje automático

Trabajo Final - Bases de datos masivas - UNLu

Rapaport, Mariano
marianorapaport@gmail.com

Normand, Agustín
normandagustin@gmail.com

13 de febrero de 2023

1. Resumen

Los precios del mercado inmobiliario de la Ciudad de Buenos Aires son variados y surgen como consecuencia de múltiples factores, presentando el desafío de determinar qué valor resulta adecuado para un inmueble. En este trabajo se recolectaron datos de inmuebles de la Ciudad de Buenos Aires publicados en el sitio MercadoLibre[1] y se enriquecieron con fuentes externas para entrenar modelos de aprendizaje automático que permitan estimar los precios de las propiedades determinando las características más importantes para este fin. Se realizó el ajuste de árboles de decisión con diferentes técnicas y configuraciones, con (1) un enfoque de clasificación capaz de determinar el rango de precios, obtenido mediante *clustering*, en que encasilla un inmueble con una precisión entre 60 % y 66 %, y (2) un enfoque de regresión apto para estimar el valor de una propiedad con un error absoluto medio porcentual de 13,8 %. Tomando este último como modelo de referencia por su buen desempeño, las características que en mayor medida explican el valor de un inmueble son el área total y cubierta de una propiedad, seguido de la cantidad de baños, ambientes, y dormitorios de la misma.

2. Introducción

La adquisición de un inmueble representa una inversión significativa y un hito en la vida de la mayoría de los individuos, debido a la significancia de la compra y a los altos importes que se deben afrontar. Con tan solo efectuar una rápida búsqueda por inmobiliarias o sitios de publicación de inmuebles, es fácil determinar la diversidad de características y precios de la oferta. Algunas diferencias en las características de los inmuebles hacen que parezca obvia la incidencia sobre el precio. Sin embargo, existen muchas variables que un vendedor puede considerar a la hora de definir el precio de su propiedad, por lo que ciertas conclusiones pueden resultar premeditadas. Algunas de estas variables son inherentes al inmueble, otras están relacionadas con su ubicación geográfica, o bien basadas en referencias de propiedades similares, la necesidad y apuro del vendedor, entre otras.

En definitiva, el problema de determinar el precio de un inmueble no es uno sencillo. Debe considerarse también que - como todo mercado - sufre fluctuaciones al alza y a la baja que se ven afectadas por el contexto macroeconómico del país. En el caso de Argentina, las crisis económicas frecuentes agregan complejidad adicional, y esto profundiza la dificultad del análisis en caso de considerar datos históricos.

Respecto a la importancia de una correcta estimación, se encuentra la perspectiva del vendedor que desea obtener el beneficio correspondiente, y la del comprador que no quiere abonar más que el precio justo. Además, una buena estimación brinda la posibilidad de detectar si un inmueble posee una valuación adecuada, un sobreprecio, o si se trata de una oportunidad de inversión rentable por su bajo precio.

Por lo tanto, habiendo expuesto algunas dimensiones de la dificultad del problema, es necesario mencionar que existen ciertas concepciones, tal vez acertadas, sobre los valores de referencia de un inmueble. En especial, se asocia la ubicación y el tamaño de una vivienda como factores determinantes de su precio. [7] De hecho, existen índices acotados que reflejan el valor por metro cuadrado de los inmuebles por zona o barrios [11]. En este trabajo se podrá verificar la validez de estas afirmaciones.

Existen algunos *datasets* de inmuebles en venta en Buenos Aires que se podrían utilizar como punto de partida para un trabajo de estimación de precios, pero su cobertura puede estar sesgada

[1] https://www.mercadolibre.com.ar/

por tratarse de empresas dedicadas al mercado inmobiliario. Esta limitación deberá ser sorteada para poder llevar a cabo el proceso de manera adecuada.

2.1. Trabajos Relacionados

Existen investigaciones previas para la estimación de precios inmobiliarios basadas en aprendizaje automático entre las que pueden mencionarse la llevada a cabo en Brasil sobre un *dataset* de 12 millones de registros extraído de Properati[2], conformado por 24 atributos del inmueble, enriquecido por los autores con la *API* de Google Maps[3] y datos del Instituto de estadísticas y geografía Brasileño[4]. Entre los pasos de preprocesamiento a destacar, puede mencionarse la aplicación de una escala logarítmica sobre el precio y la identificación de *outliers* como aquellas observaciones que excedían el percentil 99. Los modelos ajustados fueron árboles de regresión y redes neuronales recurrentes, mediante una arquitectura ensamblada. Los resultados demuestran que el número de habitaciones, el área cubierta y el área total son las variables más significativas en la determinación del precio. [1] Además, en Colombia se condujo una investigación utilizando técnicas de *machine learning* para el análisis de regresión de precios de inmuebles recolectados a partir de avisos de viviendas usadas entre 2016 y mediados de 2018, conformado por alrededor de 61 mil observaciones y 18 variables. Los autores utilizaron árboles de regresión y compararon los resultados de la utilización de todos los atributos contra la preselección mediante una estrategia de muestreo incremental con *resampling*, que descarta aquellos atributos que no considera relevantes, quedándose con los pocos considerados "vitales". A su vez, utilizaron análisis de componentes principales para resumir características afines, tal como las variables físicas del inmueble (área construida, número de baños y ambientes, habitaciones) y las variables vinculadas a comodidades adicionales o *amenities*. La métrica de evaluación elegida para evaluarlos fue R2, y el mejor modelo evaluado sobre el 30 % del dataset obtuvo un valor cercano al 82 %. Entre los atributos a destacar, los autores resaltan la importancia del estrato socioeconómico del barrio de la propiedad, el precio del metro cuadrado en la zona y la combinación de características que conforman al tamaño del inmueble.[6] En Argentina se llevó a cabo una investigación sobre la valuación de la tierra en la Ciudad de San Francisco, mediante un *dataset* de 174 observaciones y utilizando 22 variables continuas relacionadas a distancias (barrios cercanos, centro de la ciudad, espacios verdes) y de la urbanización del entorno. Utilizan árboles de clasificación y regresión, junto a *bagging* para solventar el tamaño reducido del conjunto de datos, obteniendo un error relativo promedio en valor absoluto igual al 20 %. **??**

Por otro lado, la estimación de precios inmobiliarios ha sido abordada en el campo de las ciencias económicas, donde predomina la idea de considerar la rentabilidad que puede generar el inmueble a partir de la explotación del mismo mediante alquileres, un enfoque sobre tasas de descuento ajustadas por riesgo, o bien funciones de utilidad respecto a las preferencias frente al riesgo del individuo [4]. Otra parte de la bibliografía se centra en la construcción de índices históricos que permitan evaluar la fluctuación de los precios en los inmuebles a lo largo del tiempo y basados en estimadores de precios de regresión hedónica[8][5], que ponderan las características inherentes a la propiedad, las del vecindario, ubicación, zonificación y equipamiento de cada inmueble. Esta sección de la bibliografía introduce conceptos que permiten entender qué características influyen a la hora de determinar el precio. Sin embargo, no involucran modelos de aprendizaje automático para realizar la tarea de predicción. Por ejemplo, en la Ciudad de Buenos Aires, se desarrolló un trabajo que utiliza dichas técnicas y se limita a 12 barrios de la ciudad debido al volumen de información. Entre las conclusiones, informan que las características físicas de los inmuebles son

[2]https://www.properati.com.ar/

[3]https://developers.google.com/maps/documentation/places/web-service/overview?hl=es-419

[4]https://www.ibge.gov.br/

más fuertes que las de localización. [8]

2.2. Objetivos

El objetivo general de este trabajo es predecir el precio de inmuebles de la Ciudad Autónoma de Buenos Aires siguiendo un proceso de descubrimiento de conocimiento. Para esto se definieron tres objetivos específicos:

- Crear un *dataset* con inmuebles en oferta en la Ciudad de Buenos Aires.
- Enriquecer el *dataset* con ingeniería de *features*.
- Ajustar modelos de predicción de precios a partir de las características que describen un inmueble.

El primer objetivo permite tener una vista recurrente y actualizada de los datos de interés, además de sortear la restricción de cobertura sesgada de los *datasets* existentes planteada anteriormente. El segundo contempla la expansión de los datos para mejorar su calidad y posibilidades de explotación; finalmente el tercero contempla la tarea de descubrimiento de conocimiento que enmarca este trabajo.

3. Materiales y Métodos

En esta sección se describen las principales etapas del trabajo y que sustentan los objetivos formulados. La primera de ellas se asocia a la construcción del *dataset*, la incorporación de diversas fuentes y el tratamiento de datos llevado a cabo, y la segunda a los ajustes de los modelos, técnicas incorporadas y evaluación de las configuraciones.

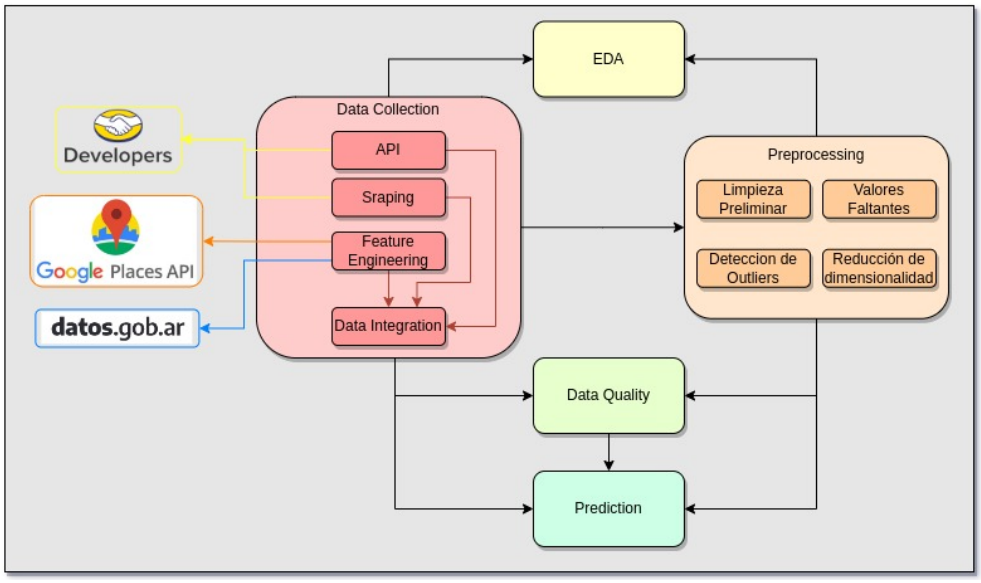

Figura 1: Esquema de la metodología

3.1. Construcción del *dataset*

Contar con una base de datos de inmuebles no es común, pero en los últimos años la web se ha convertido en el medio de publicación de *e-commerce* por excelencia [2], y los inmuebles no son la excepción. El acceso público a esta información y su masividad brinda la posibilidad de consolidar una base de datos que permita una posterior explotación. Una estrategia para hacerlo es segmentarlo por etapas, donde la extracción de información de la web es el inicio, y mediante la utilización de herramientas automatizadas se conforma el *dataset*. En este trabajo, tomando este enfoque, se creó un *scraper* para la extracción de datos de inmuebles de uno de los sitios de publicación más populares del país: MercadoLibre[5].

3.1.1. Recopilación de los datos

En conformidad con el primer objetivo del trabajo, es necesario aplicar diferentes técnicas para recuperar datos de fuentes diversas, almacenarlos e integrarlos. En esta sección se detallan los orígenes y métodos de recuperación aplicados.

3.1.1.1. Consumo de la *API* La *API* del sitio comentado anteriormente brinda la forma más sencilla para consultar los datos de forma programática y expone un *endpoint* para acceder a las publicaciones con ciertos criterios de búsqueda. Sin embargo, presenta algunas limitaciones que se detallan en el Anexo A. Como resultado de la recuperación de los datos, se obtuvieron 308.017 registros, cuyas ubicaciones se observan en la figura 2. De estos, se conservaron 82.277 registros que son los pertenecientes a la Ciudad Autónoma de Buenos Aires. Procesar este volumen de información excede las capacidades de una computadora personal, lo que denota que el problema abordado se trata de una tarea de *Big Data*.

Los 27 atributos de la primera versión del *dataset* se observan en la tabla 1.

Tabla 1: Conformación final de atributos a partir de la *API*.

Lista de atributos	
Atributo	**Tipo de dato**
rooms, *full_bathrooms*, *bedrooms*, *seller_sales*, *seller_handling_time*, *seller_id*, *seller_city* , *seller_state*, *real_estate_agency*, *seller_cancelations*, *seller_claims*	*Integer*
operation, *property_type*	*Enum*
covered_area, *total_area*, *item_condition*, *currency_id*, *neighborhood*, *state*, *city*, *permalink*, *title*	*String*
has_air_conditioning, *has_telephone_line*, *with_virtual_tour*	*Boolean*
price, *longitude*, *latitude*	*Float*

3.1.1.2. Scraping HTML La *API* devuelve atributos útiles para comenzar el *dataset*, pero hay otros que resultan interesantes y no pueden obtenerse por este medio. El método tradicional de acceso a una publicación en MercadoLibre[6] mediante un navegador *web*, ofrece - en la mayoría de los casos - información relevante tal como la descripción de la publicación, una tabla de características, la cantidad de fotos y algunas tablas con atributos adicionales. Para poder explotar estos datos, es necesario profundizar en el procesamiento. En el Anexo B se detallan los pasos llevados a cabo, y en la figura 3 un esquema de la estrategia utilizada.

[5] https://www.mercadolibre.com.ar/
[6] https://www.mercadolibre.com.ar/

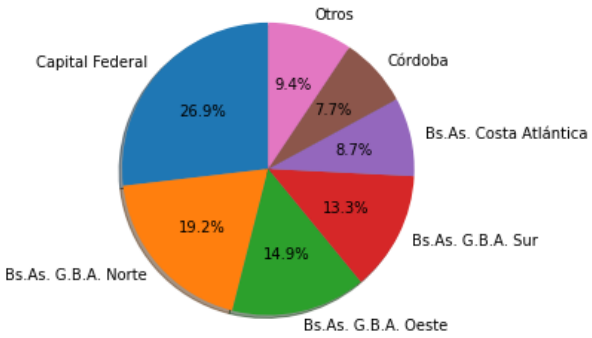

Figura 2: Ubicación de los inmuebles

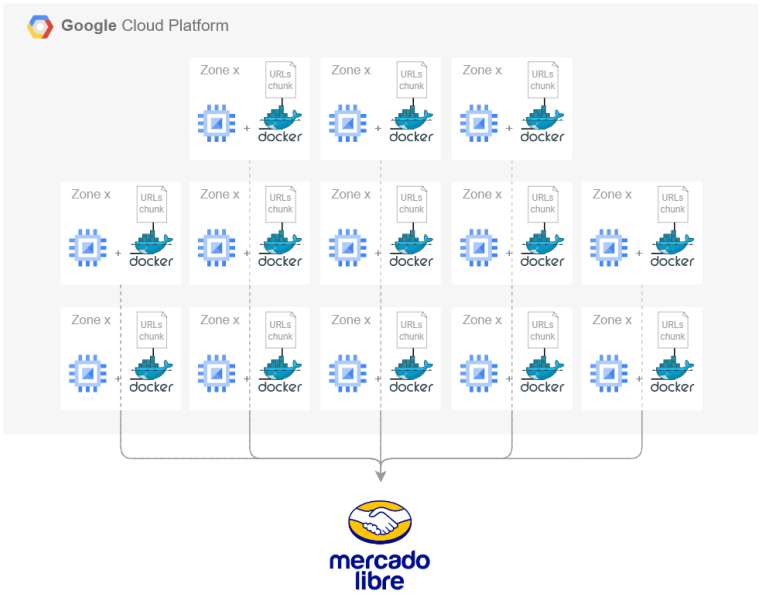

Figura 3: Infraestructura en *GCP* desplegada para el procesamiento

Los atributos resultantes fueron agregados al *dataset*, y se presentan en la tabla 2. Junto con las *features* de tipo *booleanas* que fueron extraídas de la descripción no estructurada, mediante el uso de expresiones regulares y sinónimos.

3.1.1.3. Ingeniería de *features* Los datos consolidados hasta el momento fueron recuperados de una única fuente mediante los recursos disponibles tanto en la *API* como en el sitio de MercadoLibre[7]. Sin embargo, aplicando cierto criterio, conocimiento del dominio y creatividad, pueden generarse nuevos atributos que no se encuentren en la publicación.

Una dirección para enriquecer el *dataset* consiste en agregar los lugares cercanos a cada inmueble en base a la información obtenida de la *API* de Google Places[8]. Para poder recuperarlos, se utilizaron las coordenadas de cada inmueble y se definió un rango de distancia de 500 metros con los establecimiento aledaños a incluir. El procedimiento, dificultades encontradas y su resolución se encuentra en el Anexo C.

[7]https://www.mercadolibre.com.ar/

[8]https://developers.google.com/maps/documentation/places/web-service/overview?hl=es-419

Tabla 2: Atributos derivados de la tabla de características.

Lista de atributos	
Atributo	**Tipo de dato**
Ambientes, Dormitorios, Baños, Cocheras, Cantidad de pisos, Departamentos por piso, Número de piso de la unidad, Bodegas,	*Integer*
Superficie total, Superficie cubierta, Disposición, Antigüedad, Expensas, Superficie de terreno	*String*
Tipo de departamento, Orientación, Tipo de casa	*Enum*
has pool, *has terrace*, *has jacuzzi*, *has washer*, *has gym*, *has cochera*, *has air conditioning*, *has underfloor heating*, *has elevator*, *needs recycling*, *has amenities*, *is recycled*, *has security*, *has parquet floor*, *has service dependency*, *is luminous*, *was revaluated*, *allows pets*	*Boolean*

Finalmente, los atributos agregados al *dataset* se observan en la tabla 3, donde el valor de cada uno es un número que representa la cantidad de lugares cercanos de la categoría en cuestión:

Tabla 3: Atributos derivados de la *API* de *Google Places*.

Lista de atributos	
Atributo	**Tipo de dato**
food_and_drinks_stores, *spa*, *transport_facilities*, *parking*, *school_universities*, *airports*, *hospitals*, *retail_stores*, *laundry*, *gym*, *pet_services*, *car_services*, *culture_and_entertainment*, *house_services*, *public_forces*, *banks_atms*, *green_spaces*, *funerary_services*	*Integer*

3.1.1.4. ***Real Neighbourhood*** Algunos inmuebles tienen un barrio definido diferente al que pertenecen por sus coordenadas, por lo que se realizó un cruce de información con respecto a los limites geográficos de cada barrio, agregando un nuevo atributo al *dataset*, con el valor real del mismo. Para el detalle de la resolución, acudir al Anexo D.

3.1.1.5. Barrios populares cercanos Un factor que puede resultar determinante a la hora de lograr el objetivo del modelo, es la distancia de un inmueble al asentamiento o barrio popular más cercano, dado que muchas veces esto caracteriza la zona y puede ser considerado por los compradores.

Para agregar este atributo la metodología fue similar a la del apartado anterior: se consideró la proximidad de un inmueble a un asentamiento como la distancia entre la latitud-longitud del inmueble y el punto más cercano del barrio popular. Para determinar esta distancia se realizan cálculos entre un punto del plano y el polígono correspondiente a cada barrio, almacenando tal valor en el nuevo atributo. Ciertos asentamientos informados son habitados por muy pocas familias, tratándose de casas superpobladas que no fueron consideradas dentro de la cercanía, por no tratarse realmente de barrios populares.

3.1.1.6. ***Crime Number*** Se incorporó el número de comuna al que pertenece cada inmueble para poder obtener la tasa de criminalidad asociada a su zona. Concretamente, el valor utilizado corresponde a los crímenes promedio desde 2018 hasta 2020.

3.1.2. Entendiendo el *dataset*

Luego de la recolección de distintas fuentes, se conformó un *dataset* con inmuebles de todo el país. Luego de acotarlo a los de Capital Federal, se analizó en mayor detalle qué ocurre en el resto de las variables para lograr un mayor entendimiento de los datos, poniendo el foco en: valores faltantes, rangos de las variables, presencia de ruido, medidas de tendencia central, distribución de la variable objetivo y en especial, la correlación del resto con ella, en pos de detectar una relación directa o inversa muy fuerte con el precio, que permita una sencilla predicción.

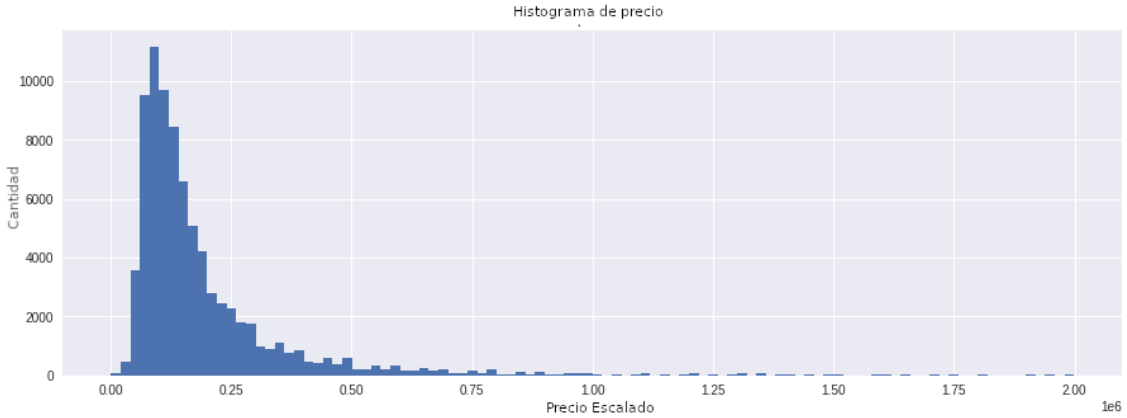

Figura 4: Distribución precios USD

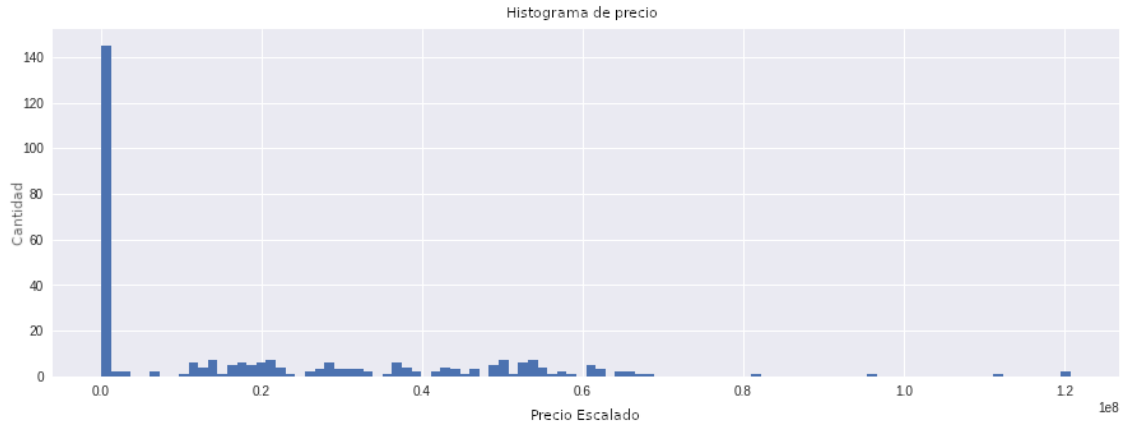

Figura 5: Distribución precios ARS

3.1.2.1. Distribución de precios En la figura 4 se observa cómo se comporta la variable para aquellos inmuebles publicados en USD. Se puede ver que el sesgo es hacia la derecha, y el 50 % de los valores está concentrado entre los 94000 USD (primer cuartil) y 220000 USD (tercer cuartil). Las propiedades que superan el millón de dólares están en torno al 1.5 % del total. El gráfico se ha limitado a 2 millones de USD para facilitar la visualización, pero el precio máximo encontrado es de 29500000 USD.

A su vez, se ve cómo se comporta el precio para aquellas publicaciones en ARS en la figura 5. Cabe destacar que hay alrededor de 300 publicaciones en pesos, entre las que se distinguen dos

grandes grupos. Por un lado, aproximadamente 140 inmuebles a la izquierda del gráfico con un precio menor a 650000 ARS, que muy probablemente sean publicaciones mal catalogadas, cuyo valor corresponde a USD. Por el otro, se encuentran publicaciones que superan los diez millones de pesos, donde el valor es acorde y la moneda es correcta.

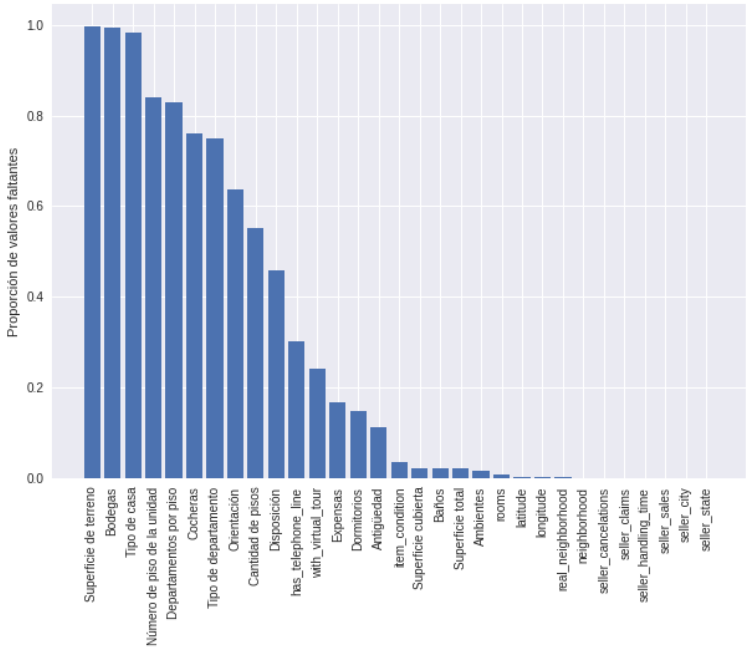

Figura 6: Proporción de valores faltantes

3.1.2.2. Valores faltantes Una porción de los atributos de las publicaciones son opcionales. Como consecuencia, y sabiendo además que los usuarios no dan importancia a todos por igual, es necesario entender cuáles de estos presentan faltantes y que tal vez, convenga descartar. En la figura 6 se observa que Superficie de Terreno, Bodegas y Tipo de casa superan el 95 % de valores faltantes. A su vez, el número de piso de la unidad, la cantidad de departamentos por piso, las cocheras y tipo de departamento son atributos con aproximadamente 80 % de faltantes. Los atributos restantes tienen como máximo la mitad de sus valores ausentes, a excepción de la orientación que supera el 60 %.

3.1.2.3. Variables correlacionadas al precio Luego de establecer el coeficiente de correlación de *Pearson* con cada una de las variables de los datos recolectados, no fue posible identificar un atributo relacionado al precio con un grado mayor a 0.5 a excepción de la Superficie del Terreno, que posee la mayor cantidad de faltantes y por ende no es fiable. Los baños, la cantidad de ambientes del inmueble y los dormitorios son los atributos que mayor correlación presentan con la variable objetivo.

3.1.2.4. Atributos con valores ruidosos A partir del análisis de valores máximos y mínimos, es posible detectar los atributos con ruido que requerirán un posterior tratamiento:

- Latitud y longitud: coordenadas que exceden la República Argentina.
- Precio: valor máximo de 1460000000 de ARS.

- *Bedrooms*: valor mínimo -1 y máximo 11111111.
- *Full bathrooms*: valor mínimo -4 y valor máximo 1111111.
- *Rooms*: valor mínimo -5 y valor máximo 111111.
- *Covered area*: valor máximo 145000.
- *Total area*: valor máximo 4580000.
- Cocheras: valor máximo 20000.
- Cantidad de pisos: valor máximo 14123355.
- Antiguedad: valor máximo 1111.
- Expensas: valor máximo 81002142.

3.2. Preprocesamiento de datos

El *dataset* que se conformó contaba con datos en el mismo formato que en la fuente, por lo que - en algunos casos - los valores contenían ruido, estaban incompletos, eran redundantes, tenían valores extremos y requerían operaciones tales como la normalización y discretización. En conjunto, este grupo de actividades conformaron la etapa de preprocesamiento donde se mejoró la calidad de los datos.

3.2.1. Limpieza de datos

Se pueden agrupar las tareas de limpieza de datos realizadas en tres grupos: transformaciones, eliminación de registros y eliminación de ruido.

3.2.1.1. Transformaciones Los atributos numéricos listados a continuación estaban representados como *string* en el *dataset* e incluían la medida en que se encuentran expresados dentro del valor de cada registro. A su vez, un mismo atributo tenía diferentes medidas (por ejemplo, metros o hectáreas para la superficie total). Para el precio del inmueble ocurrió algo similar, pero en este la medida no estaba dentro del atributo (que es numérico) sino que era *currency_id*, un atributo adicional. Con esto en consideración, se unificaron las medidas para los siguientes atributos:

- *total_area*: ha a metros cuadrados
- Expensas: USD a ARS
- Superficie Total: ha a metros cuadrados
- Superficie Cubierta: ha a metros cuadrados
- Superficie de Terreno: ha a metros cuadrados
- Precio: ARS a USD

Luego de la transformación, la presencia de la medida dentro del atributo ya no resultaba conveniente para poder manipular los datos, por lo que se eliminó y los valores fueron convertidos a números.

En el caso del precio se presentaron dos inconvenientes. Por un lado, los inmuebles que erróneamente colocaron al peso argentino como moneda cuando el valor definido es en dólares y por otro lado, el tipo de cambio a considerar para la transformación. Sobre el primero de estos, la conversión fue realizada para aquellos registros cuyo valor en pesos superaba 1000000. Esto se ve justificado por la distribución de la figura 5, donde la porción de la izquierda se trata de inmuebles con un valor en pesos tan bajo que probablemente estén valuados en dólares y mal publicados en cuanto a definición de la moneda. Respecto al tipo de cambio, se consideró 1 USD = 210 ARS siendo esta la conversión no oficial a la fecha de recuperación de los datos.

Todos los atributos expresados en término de "Si" o "No" fueron transformados a booleanos.

Por último, respecto a la Antigüedad de algunos inmuebles (expresada en años) se asumió que aquellos superiores a 1800 indican el año de construcción y no la cantidad de años de la propiedad. Por lo tanto, se calculó la diferencia entre el año en curso y el valor indicado, homogeneizando así la antigüedad.

3.2.1.2. Eliminación de registros Algunas publicaciones "contaminan.[el] subconjunto de interés del *dataset*. Específicamente los terrenos, emprendimientos, pozos, hoteles, *hostels*, casas multifamiliares, locales y edificios con ventas en *block*: la naturaleza de estos ẗipos de propiedadës muy distinta a la de un departamento o casa unifamiliar y se comercializa de otra forma. Por lo tanto, estos registros fueron eliminados del *dataset*.

Además, se definió el rango de precios de inmuebles a conservar, acotándolo a los rangos de mayor presencia de inmuebles, dado por el límite inferior de 35 mil USD, y el superior de 410 mil USD. El porcentaje de registros eliminados con dicha acción no fue significativo.

3.2.1.3. Eliminación de ruido Todos los atributos numéricos recolectados, a excepción de los incorporados mediante fuentes externas, tienen valores ruidosos que exceden un rango válido dentro del dominio de los mismos y probablemente se deban a errores humanos.

Para detectarlos, se calcularon los valores por debajo y por encima del percentil 1 y 99 respectivamente en la distribución de cada atributo. Para poder establecer estas medidas, se hizo una separación entre las dos grandes categorías de inmuebles que conforman el *dataset*: casas y departamentos. El objetivo es comparar inmuebles similares, dado que atributos como la cantidad de pisos tienen un dominio distinto para uno y otro.

Para remover estos valores sin perder información del *dataset*, se llevó a cabo un reemplazo por nulos manteniendo el registro en el conjunto, sabiendo que posteriormente la imputación de datos faltantes los reemplazará por valores dentro de un rango válido.

3.2.2. Valores faltantes

Se puede observar en la figura 6, que el *dataset* tiene valores faltantes para varios de sus atributos, por lo que se imputaron utilizando diferentes criterios, que se pueden consultar en el Anexo E.

3.2.3. Construcción de la variable objetivo

Para la creación de la variable objetivo del modelo de clasificación, una primera aproximación para definir categorías de precios puede consistir en aplicar un criterio subjetivo entre inmuebles

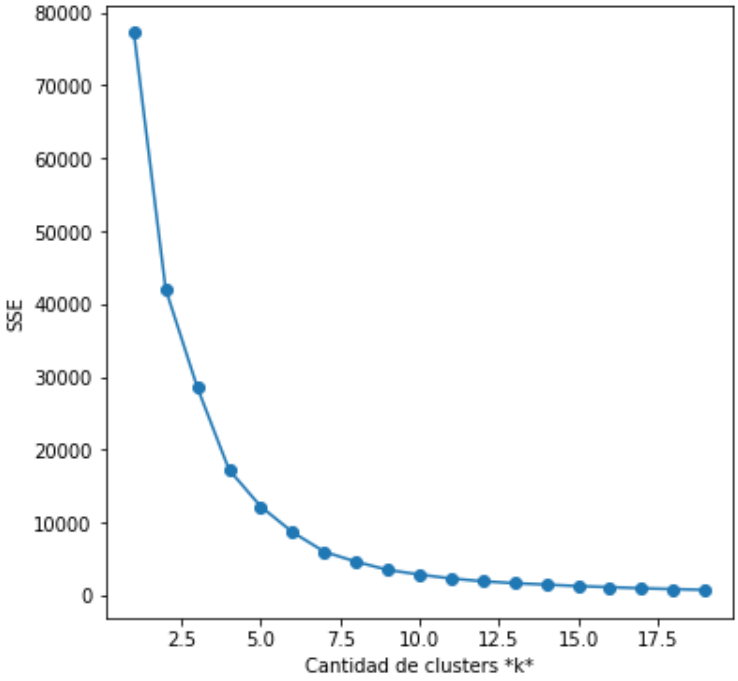

Figura 7: Similitud de precios entre propiedades de cada grupo

"baratos", "intermedios" *y* "caros", cuyos límites pueden variar entre persona y persona. Se realizaron experimentos con estos grupos, pero dado que esto carece de fundamento teórico y por ende se dificulta definir tales límites, se llevó a cabo una agrupación de inmuebles mediante *k-medias* [9], tomando 7 como valor de k a partir de la técnica de *Elbow*. En la figura 7 se puede ver que es allí donde la curva comienza a aplanarse.

En la tabla 4 se observan las estadísticas de los grupos obtenidos.

Tabla 4: Estadísticas de los *Clústers* generados

Cantidad	**Media**	**Mínimo**	**Máximo**
8378	63779.74	35295.0	77423.0
13522	91100.37	77500.0	106300.0
5651	121574.00	106400.0	138700.0
14442	155951.25	138770.0	178886.0
11077	202019.52	178999.0	233942.0
3392	266008.01	234000.0	309653.0
10174	353491.81	309811.0	409765.0

3.3. Ajuste de los modelos de aprendizaje automático

Dividiendo el conjunto de datos en set de entrenamiento y de validación, con proporción 70-30 respectivamente, se utilizaron las siguientes técnicas de aprendizaje automático.

3.3.1. *Random Forest Classifier*, *Cross Validation* y *Grid Search*

Utilizando un enfoque de clasificación, basándose en un árbol de decisión que, a partir de las variables incorporadas al modelo, indique el rango de precios al que la propiedad pertenece. Siendo la variable objetivo los rangos de precios creados anteriormente y utilizando este conjunto de técnicas, se experimentó sobre un conjunto de datos de entrenamiento correspondiente al 70 % del *dataset*, reservando el restante para probar sobre datos no vistos por el modelo. Sobre el primero se aplicó la técnica de *Grid Search*[9] para encontrar el conjunto de hiperparámetros óptimos para el entrenamiento de *RandomForestClassifiers* [3] de la librería *Scikit-learn*[10]. Para estos, los parámetros variados son *n_estimators* entre 100 y 500, con un paso de 50; *max_features* entre '*sqrt*' y '*log2*'; *max_depth* entre 4 y 9 con un paso de 1 y *criterion* entre *gini* y *entropy*;

3.3.2. *Catboost*

Mediante esta herramienta[10], se incorporaron técnicas de potenciación del gradiente (*gradient boosting*) [11], para construir un árbol de decisión nuevo utilizando los rangos de precios de la tabla 4.

3.3.3. *Random Forest Regressor*

Los resultados de la clasificación no fueron satisfactorios, y sin poder encontrar una clara separación de categorías para la variable objetivo, se evidenció la necesidad de incorporar técnicas de regresión a los experimentos. Mediante *Random Forest* [3] y *Cross Validation* [12], se utilizaron distintos enfoques para la predicción: la construcción de un único modelo para el *dataset* completo, con proporción 80-20 para entrenamiento y *testing*; la construcción de un conjunto de modelos donde cada uno está enfocado a un barrio del *dataset*; la anidación adicional de la cantidad de ambientes del inmueble a los modelos previamente construidos por barrio, en los casos que la cantidad de registros lo permitiera.

3.3.4. Selección de *features*

Al haber detectado aquellos atributos que los modelos consideran más importantes, y mediante el uso de *Lasso* [13], se dio lugar a una nueva construcción que solo incluyera dichas features. Con esto, se intentó reducir el ruido que pudieran haber causado atributos de menor importancia y que las técnicas utilizadas tal vez incorporaron sin éxito. A su vez, se ejecutaron pruebas combinando atributos tales como los lugares cercanos mediante una sumatoria.

3.3.5. Discretización de atributos

Algunos experimentos llevados a cabo involucran la transformación de atributos continuos a discretos. Entre estas, la discretización del atributo expensas.

3.4. Métricas de evaluación

A continuación se describirán las métricas utilizadas para comparar el comportamiento de los modelos y así entender cuál resulta más apropiado para la tarea de predicción.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[10] https://scikit-learn.org/stable/index.html
[11] https://en.wikipedia.org/wiki/Gradient_boosting

3.4.1. MAE - *Mean absolute error*

Es una métrica que acumula las diferencias entre las predicciones y el valor real de la variable objetivo, dividiendo esta suma por la cantidad de observaciones. De esta manera, permite entender qué tanto se equivoca un modelo, dando el mismo peso a errores pequeños y grandes.

3.4.2. R2

R2 (también conocido como coeficiente de determinación) mide la variación que se explica por un modelo de regresión. R2 de un modelo de regresión es positivo si la predicción del modelo es mejor que una predicción que es solo la media de los valores de 'y' ya disponibles; de lo contrario, es negativo.

3.4.3. R2 ajustado

Para un modelo de regresión múltiple, R-cuadrado aumenta o permanece igual a medida que se agregan nuevas *features* al modelo, incluso si las nuevas *features* agregadas son independientes de la variable objetivo y no agregan ningún valor al poder predictivo del modelo. El R-cuadrado ajustado elimina este inconveniente del R-cuadrado. Solo aumenta si la *feature* recién agregada mejora el poder de predicción del modelo. Agregar variables independientes e irrelevantes a un modelo de regresión da como resultado una disminución en el R-cuadrado ajustado.

3.4.4. Porcentaje de error relativo

El rango de precios de los inmuebles considerados es bastante amplio, por lo que utilizar una métrica de error absoluta para entender el error puede llevar a la mala interpretación de la calidad de un modelo. En otras palabras, conocer que el error es de 20000 USD no es lo mismo para un inmueble que vale 40000 USD que para otro de 500000 USD. Como consecuencia, se incorporó una nueva métrica sencilla a la evaluación en función al precio del inmueble.

Básicamente, el error relativo conforma la relación entre el error de la predicción del modelo para un inmueble y el precio real del mismo. Por ejemplo, para una propiedad publicada en 100000 USD, una predicción de 110000 USD tiene un error relativo del 10 %. Para un inmueble del doble de valor, predecir 210000 USD representa un error del 5 %, pero si se tratara de uno de 50000 USD, equivocarse por 10000 USD se traduce en un error del 20 %.

4. Resultados obtenidos

El modelo construido con *Catboost* alcanza una precisión de 61.9 %. En la figura 8 se presenta la matriz de confusión obtenida. En los rangos con una proporción suficiente de registros el modelo predice correctamente la mayoría de las observaciones. En el octavo y décimo rango, por el contrario, la cantidad de registros es muy acotada y el modelo no se desempeña de forma adecuada. El mayor inconveniente del enfoque de clasificación está dado por la cercanía entre los rangos centrales. Si se observa la diagonal de la matriz, se comprueba que el modelo tiende a categorizar erróneamente el 30 % de las observaciones de cada clase, distribuyéndolas en las contiguas. Teniendo en cuenta este fenómeno y la naturaleza de la variable, se evidencia la necesidad de utilizar otro enfoque que no requiera la definición de categorías.

El árbol de decisión obtenido en la iteración 26.026 con los parámetros correspondientes se encuentra anexo a este informe. Puede verse que las características más importantes a la hora de

determinar el precio de un inmueble son el área cubierta, el costo de las expensas, el barrio, la comuna y la cantidad de pisos del edificio.

Valores Predichos	Rango 1	1498	506	34	5	1	0	0
	Rango 2	359	1859	575	82	6	0	1
	Rango 3	33	548	1590	458	53	1	0
	Rango 4	1	72	517	1279	367	26	4
	Rango 5	0	7	57	373	968	215	14
	Rango 6	0	1	3	45	328	591	136
	Rango 7	0	0	1	4	24	210	476
		Rango 1	Rango 2	Rango 3	Rango 4	Rango 5	Rango 6	Rango 7
		Valores reales						

Figura 8: Matriz de confusión *Catboost*

Por otro lado, mediante *random forest classifier* la precisión obtenida es del 64,02 %. La matriz de la figura 9 presenta los resultados de la predicción del modelo sobre el 30 % no vistos anteriormente.

Valores Predichos	Rango 1	1505	497	38	3	1	0	0
	Rango 2	347	1955	506	70	3	0	1
	Rango 3	31	541	1664	402	44	1	0
	Rango 4	2	75	516	1338	316	17	2
	Rango 5	0	4	64	364	995	191	16
	Rango 6	0	0	5	60	315	601	123
	Rango 7	0	1	1	8	38	204	463
		Rango 1	Rango 2	Rango 3	Rango 4	Rango 5	Rango 6	Rango 7
		Valores reales						

Figura 9: Matriz de confusión *Random Forest*

Utilizando *random forest regressor* con el enfoque único y todos los atributos del *dataset*, dado el siguiente *grid*:

- *n_estimators*: 1800, 1850, 1900, 1950, 2000, 2050, 2100, 2150, 2200
- *criterion*: *squared_error*
- *max_features*: *sqrt*
- *min_samples_split*: 2, 3, 4, 5, 6, 10, 50, 100, 500
- *min_samples_leaf*: 1, 2, 3, 4, 5, 10, 50, 100, 500

La combinación de hiperparámetros con mejor desempeño:

- *criterion*: *squared_error*
- *max_features*: *sqrt*
- *min_samples_leaf*: 1
- *min_samples_split*: 3

- *n_estimators*: 1900

Alcanza un MAE de 18882.14 USD, r2 ajustado igual a 0.877 y el error relativo medio de 13.9 %. Los experimentos manteniendo los 11 atributos que *Lasso* identifica como más importantes no logran mejorar estos valores.

Al evaluar el modelo separado por barrios, la mejor combinación de hiperparámetros (consultar Anexo F), obtenida del siguiente *grid*:

- *n_estimators*: 1500, 1750, 1900, 2000, 2100, 2200
- *max_features*: *sqrt*
- *min_samples_split*: 2,3,4,5
- *min_samples_leaf*: 1,2,3,4
- *random_state*: 18

Logró alcanzar un MAE igual a 18650.4 USD, r2 ajustado igual a 0.874 y un porcentaje de error relativo medio de 13.8 %.

Al agregar la cantidad de ambientes a la anidación, con la combinación de hiperparámetros:

- *n_estimators*: 2000
- *max_features*: *sqrt*
- *min_samples_split*: 2
- *min_samples_leaf*: 1
- *random_state*: 18

El MAE asciende a 20176.62 USD, r2 ajustado baja a 0.83 y el error relativo medio de 14.6 %.

Todos estos experimentos representan una mejora sobre la ejecución inicial del modelo eliminando valores faltantes, cuyo RMSE era de 31.700 USD.

Un patrón encontrado en la búsqueda de los mejores hiperparámetros, es que ningún modelo obtuvo el mejor desempeño, con valores *min_samples_split* y *min_samples_leaf* diferentes a 1, 2 o 3. Por este motivo, para encontrar los mejores modelos en cada uno de los casos no hubo oportunidad para alterar más parámetros que *n_estimators*.

Las pruebas llevadas a cabo discretizando atributos, no han demostrado mejorar el error del modelo.

Analizando en mayor detalle el modelo separado por barrios y al concentrarse en el error relativo, es posible confirmar la robustez del mismo para la predicción de precios. Al observar el histograma de la figura 10, cuyo eje x representa el porcentaje de error, se comprueba que el modelo es capaz de predecir el valor del inmueble con un error menor al 10 % del precio real del mismo en más de la mitad de los casos. A partir del tercer cuartil de la distribución, se comprueba que el 75 % de las predicciones tienen un error menor al 20 %. Esto significa que tres de cada cuatro veces el modelo hará una predicción con un error menor al 20 %.

Llevando estos valores a un caso concreto, puede suponerse un inmueble de 150000 USD. Los resultados obtenidos demuestran que la mitad de las predicciones indicarán que su valor está entre 135000 USD y 165000 USD, una de cada cuatro que tiene un valor inferior a 120000 USD o superior a 180000 USD, y una de cada tres veces aproximadamente se equivocará por tan solo 7500 USD.

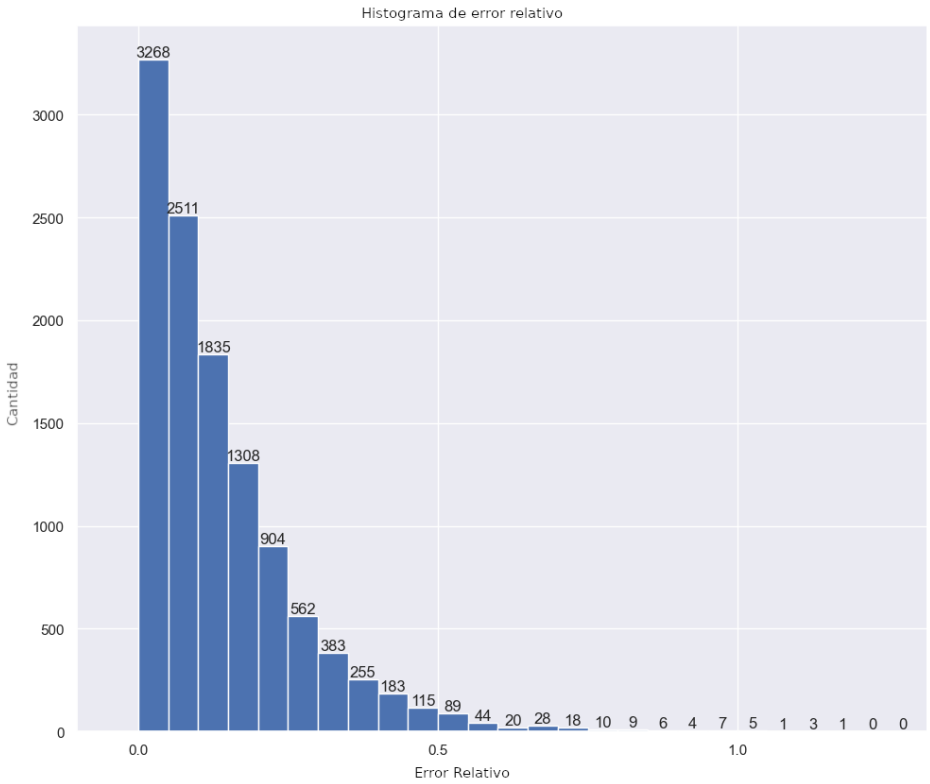

Figura 10: Histograma de error relativo para el mejor modelo obtenido

Por otro lado, respecto a la tendencia de la predicción, se detecta que el modelo tiende a sobrevalorar los inmuebles al superar el 20 % de error, dado que los casos en que predice un valor mayor al real suceden el 72 % de las veces. A su vez, por encima del 75 % de error, no hay registros en que el modelo prediga un valor menor al real.

5. Discusión

Construir un *dataset* es una tarea laboriosa en donde surgen problemas de naturaleza muy variada que obligan a encontrar soluciones en diferentes campos y a tomar decisiones que impactan en el resultado final. A su vez, la elección de los atributos trae consigo una carga subjetiva respecto a la utilidad de los mismos, sabiendo que parte del trabajo realizado puede no servir para el modelo de predicción a construir, pero se alinea con los dos primeros objetivos de esta investigación y puede resultar útil para aquellas futuras investigaciones que utilicen el *dataset* conformado. La construcción del mismo puede ser iterativa, incorporando nueva información mediante recursos ya recuperados, como los archivos *HTML*, o bien incorporando nuevas fuentes de datos, dado que la estructura desarrollada y la infraestructura distribuida sirven de soporte para tal fin. Este trabajo, sin embargo, se limita a los atributos expuestos en la sección 3.1 por considerarlos una base suficiente para los objetivos.

Luego del trabajo de recolección, se obtuvieron más de 80 mil publicaciones de inmuebles en la Ciudad Autónoma de Buenos Aires a partir de uno de los sitios de *e-commerce* más populares del país y se lograron ajustar varios modelos de predicción sobre el precio de una propiedad en base a sus características. Para ello, se incorporó información propia a las publicaciones indicadas por

los autores de las mismas, debiendo enfrentar el desafío de procesar grandes cantidades de datos. Dado que esto requiere de un poder de cómputo mayor al de una computadora personal, se adoptó una infraestructura superior. A su vez, se incorporaron atributos desde orígenes adicionales para enriquecer el conjunto de datos de estudio. Luego de llevar a cabo acciones de preprocesamiento para mitigar la presencia de ruido y valores faltantes, a la vez de entender los datos disponibles a través de técnicas de análisis exploratorio, se ajustaron múltiples modelos con comportamiento similar y diferente grado de especificidad que logran estimar el valor de un inmueble con un error absoluto de 18 mil USD aproximadamente, pero que en más de la mitad de los casos tendrá un error relativo inferior al 10 % respecto al real del inmueble.

Se analizaron las propiedades con mayor error relativo, encontrando casos donde, evaluándolo por sus características y en correspondencia con inmuebles similares en venta actualmente, el precio predicho por el modelo resultaba más adecuado que el indicado por el vendedor, siendo este último mucho menor. Tratándose de los inmuebles con mayor error relativo, se pone en tela de juicio si estos casos son los peores errores del modelo, o sus mejores aciertos. Una alternativa es que sean inmuebles con características que exceden a las presentes en este *dataset*, tales como inmuebles con una deuda elevada o propiedades cercanas a un proyecto de obra de una cárcel. La otra alternativa, sin embargo, es que se trate de oportunidades de compra.

6. Trabajos futuros

Luego de llevar a cabo el trabajo, se encontraron ciertas direcciones que presentan una oportunidad de profundizar la investigación para aumentar la riqueza del *dataset*, mejorar la precisión de la predicción o ajustar nuevos modelos. Entre ellas, se ha evaluado la posibilidad de consumir una *API* con las direcciones de los inmuebles para obtener la latitud-longitud en los que sean faltantes, como también utilizarla para todos los inmuebles en pos de detectar discrepancias entre la dirección informada y la real. Ciertas particularidades de las publicaciones, tales como la falsedad de algunos datos o la presencia y/o ausencia de otros, dan a suponer que no todas son fiables. De hecho, se encontraron registros del dataset sin los datos fundamentales como las imágenes del inmueble o la descripción del mismo, que generalmente se relacionan a publicaciones que degradan el conjunto de datos. Generar una herramienta que permita identificar estos registros mediante diferentes validaciones y que aplique una puntuación a cada uno, podría ser de gran valor para mejorar la estimación. Asimismo, restan oportunidades de discretización de atributos tal como el crimen promedio, al igual que la exploración en mayor profundidad de técnicas de *clustering* y con ello, la posible utilización de componentes principales para reducir la dimensionalidad de los datos. Sobre el enfoque de clasificación de precios hay múltiples direcciones por explorar, especialmente en la distinción de inmuebles en el límite entre categorías contiguas, dado que en ellos los modelos ajustados presentaron el mayor error. Por otro lado, evitar el recorte de inmuebles por precios, manteniendo la desproporción y solventando la desigualdad en la cantidad de muestras para cada categoría con técnicas de balanceo tal como *SMOTE*.

Referencias

[1] Bruno Afonso y col. “Housing Prices Prediction with a Deep Learning and Random Forest Ensemble”. En: (sep. de 2019).

[2] Ricardo Baeza-Yates y Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology behind Search*. 2nd. USA: Addison-Wesley Publishing Company, 2011. ISBN: 9780321416919.

[3] Leo Breiman. “Random forests”. En: *Machine learning* 45 (2001), págs. 5-32.

[4] Etelvina Chavez, Gastón S. Milanesi y Gabriela Pesce. “Valuación inmobiliaria en Argentina: propuesta de diferentes modelos. XL Jornadas Nacionales de Administración Financiera (modalidad virtual).” En: (2020).

[5] E. R. Domínguez Prost. “Construcción de un índice de precios inmobiliario para la Ciudad Autónoma de Buenos Aires a partir de datos espaciales”. En: (ago. de 2019).

[6] Juan Carlos Correa-Morales Favián González-Echavarría Jorge Iván Pérez-Rave. “A machine learning approach to big data regression analysis of real estate prices for inferential and predictive purposes, Journal of Property Research”. En: *Journal of Property Research* (2019).

[7] Luis A. Castro1 y Luis-Felipe Rodriguez1 Laura P. Lopez-Arredondo1 Cynthia B. Perez2. “Estudio sobre la Percepción de los Factores Involucrados en la Estimación de Precios de Viviendas: El Caso de Cajeme”. En: (sep. de 2018).

[8] Sonia Mabel León. “Determinación de precios inmobiliarios en CABA y efectos de política de transporte: modelos espaciales y evaluación de impacto”. En: (ago. de 2016).

[9] J MacQueen. “Classification and analysis of multivariate observations”. En: *5th Berkeley Symp. Math. Statist. Probability*. University of California Los Angeles LA USA. 1967, págs. 281-297.

[10] Liudmila Prokhorenkova y col. “CatBoost: unbiased boosting with categorical features”. En: *Advances in neural information processing systems* 31 (2018).

[11] UCEMA y RE/MAX. “Estudio sobre precios reales por m2 de departamentos en C.A.B.A.” En: (ago. de 2022).

[12] Mervyn Stone. “Cross-validatory choice and assessment of statistical predictions”. En: *Journal of the royal statistical society: Series B (Methodological)* 36.2 (1974), págs. 111-133.

[13] Robert Tibshirani. “Regression shrinkage and selection via the lasso”. En: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), págs. 267-288.

A. Anexo: Consumo *API* MercadoLibre

La *API* utiliza un enfoque de paginación limitado a 10.000 resultados en total para un recurso determinado. Esto presenta un inconveniente, y para poder obtener los inmuebles publicados es necesario sortear esta limitación. Para entender el problema, observar que dada una petición determinada que identifica un recurso, la *API* solo permitirá obtener 50 registros en la respuesta, y al iterar por cada página de resultados, se llegará como máximo al registro 10.000. De esta forma, quedarían por fuera una gran cantidad de inmuebles que podrían incorporarse al *dataset*.

De esta forma, surge el primer problema para construir la fuente de datos: no basta con realizar una única petición al servidor solicitando inmuebles, sino que requiere una estrategia para mitigar esta restricción.

Afortunadamente, las consultas a la *API* permiten incorporar filtros de búsqueda que reducen el total de coincidencias, y por ende, mantenerse por debajo del límite mencionado. De esta forma, se ejecutaron llamadas más específicas utilizando las categorías disponibles y excluyentes entre sí. Cuantos más filtros se incorporan, iterando por cada valor posible y generando conjunciones entre distintas opciones, aumenta la cantidad de consultas a realizar y se reducen los posibles resultados que devuelven. Gracias a esto, fue posible obtener los 308.017 registros que el sitio ofrecía a la fecha de ejecución de las consultas.

A continuación, se deben procesar las respuestas recolectadas. Debido al volumen de información manipulado, es necesario contar con mayor capacidad de cómputo y memoria de trabajo que la que suele encontrarse en una computadora personal. Esto evidencia que el problema abordado se trata de una tarea de *Big Data*. En el marco de este trabajo, se empleó una instancia de 4 *cores* y 30gb de RAM en *Google Cloud Platform*.

Los nombres de algunos atributos recuperados de la *API* denotan que es necesario realizar una limpieza de ciertos registros que podrían aportar ruido al conjunto de datos. “Fecha de entrega” o “Nombre del emprendimiento” por ejemplo, son indicadores de que el inmueble no es una vivienda existente, sino que se trata de ventas en pozo o emprendimientos. Asímismo, el atributo “Número de oficinasïndica la presencia de inmuebles publicados con un fin no residencial. Las publicaciones que encasillan dentro de estos grupos fueron descartadas y tales atributos eliminados. De esta manera, la primera conformación del *dataset* cuenta con 308.017 registros, representando inmuebles en venta en la República Argentina y otros países, junto a 27 atributos para cada uno de ellos. Persiguiendo una mayor precisión, buscando acotar la búsqueda y simplificando el modelo de aprendizaje automático, se eliminaron los registros que no pertenecen a tal ciudad, resultando en un *dataset* de 82277 registros.

B. Anexo: Scraper MercadoLibre

De la lista de inmuebles obtenida de la *API*, se cuenta con el enlace de cada publicación (atributo *permalink*), por lo que utilizando Python es posible obtener el HTML de la misma y mediante librerías de terceros se accedió a los atributos con la información buscada.

Al no ser una interfaz de programación y realizar sucesivas peticiones HTTP a MercadoLibre, el servidor impone una restricción: si estas son efectuadas de forma muy rápida, dejan de ser respondidas. Además, la demora en descargar la respuesta del servidor no es menor, dado que a diferencia de la *API* el contenido obtenido es mayor y por ende, su tamaño también.

Efectuando estas peticiones desde un único nodo local, el tiempo necesario impide recuperar los datos de forma práctica, alcanzando varios días de demora. La tarea no puede paralelizarse utilizando *multithreading*, porque para MercadoLibre todas las peticiones se originan desde la misma IP pública. Por lo tanto, ni siquiera distintas computadoras dentro de una misma red podrían ser una configuración adecuada.

Aunque se podrían haber utilizado proxies, se optó por otra alternativa que permita también facilitar también el procesamiento distribuido de los datos.

Para resolverlo en un orden temporal menor, fue necesario realizar las peticiones desde un grupo de IPs públicas distintas. Para obtenerlas se decidió utilizar un servicio de procesamiento en la nube, concretamente, *Google Cloud Platform*. En este último se desplegaron 13 instancias, de 13 zonas diferentes, con la finalidad de paralelizar las consultas ejecutando un *script* Python *dockerizado* esquematizado en la figura 3 del documento.

Las instancias trabajan de manera sincronizada para repartir la cantidad de recursos (URLs) que cada una debe consultar, calculándolo como la división del total entre el número de *workers*. La tarea de obtener el HTML no es trivial, dado que existen dos casos a contemplar respecto a la respuesta:

- Puede ser un desafío JavaScript
- Puede ser una página anterior a la publicación, en carácter de *banner* promocional con pocos datos del inmueble y un hipervínculo a la publicación en sí misma.

Es necesario implementar la lógica pertinente en cada uno de los *workers* para mitigar dichos problemas.

El archivo HTML es almacenado completo en un *bucket* de *GCP* para permitir un posterior procesamiento, sin necesidad de realizar nuevamente las peticiones en caso de que interese incorporar nueva información al *dataset* presente en dichos archivos. Para tener una dimensión de la cantidad de datos disponibles, el tamaño total de la colección sin comprimir alcanza los 130gb.

En una primera iteración sobre los archivos se obtuvo la tabla de características, dado que la misma contenía atributos que resultan importantes para valorizar los inmuebles. De igual manera, se realizó *parsing* sobre la descripción para obtener información valiosa, con la dificultad agregada de no poseer estructura. Y por último, se contabilizó la cantidad de imágenes de la publicación.

Esta etapa del procesamiento, cabe mencionar, también se llevó a cabo de manera distribuida, dado lo impráctico de procesar 130gb de archivos en un único nodo.

C. Anexo: Consumo *API* Google Places

La *API* de Google tiene una limitación en el tiempo entre peticiones, por lo que fue necesario realizar un procesamiento distribuido para que el tiempo de ejecución sea aceptable, almacenando cada resultado parcial en un archivo, e introduciendo una demora entre peticiones sucesivas.

La respuesta del servicio es una lista de establecimientos con cierta información sobre ellos en donde se destaca la categoría a la que pertenece. La *API* provee un amplio dominio de categorías especificas, por lo que se realizó una agrupación mediante un mapeo de las mismas hacia nuevas categorías.

En primer lugar, se definieron nuevas agrupaciones:

```
categories = {
    "food_and_drinks_stores" : ["restaurant", "bar", "cafe", "meal_takeaway",
    "meal_delivery"],
    "spa": ["spa"],
    "transport_facilities": ["bus_station", "transit_station", "train_station",
    "subway_station"],
    "parking": ["parking"],
    "school_universities": ["school", "primary_school", "secondary_school", "university"],
    "airports": ["airport"],
    "hospitals": ["hospital", "doctor"],
    "retail_stores": ["supermarket", "drugstore",  "convenience_store"],
    "laundry": ["laundry"],
    "gym": ["gym"],
    "pet_services": ["pet_store", "veterinary_care"],
    "car_services": ["gas_station", "car_repair", "car_wash"],
    "culture_and_entertainment": ["movie_theater", "movie_rental", "night_club",
    "casino", "shopping_mall", "museum", "library",  "aquarium", "book_store",
    "bowling_alley", "art_gallery"],
    "house_services": ["electrician", "moving_company", "painter", "plumber",
    "locksmith"],
    "public_forces": ["police", "fire_station"],
    "banks_atms": ["bank", "atm"],
    "green_spaces": ["park", "zoo", "campground"],
    "funerary_services": ["cemetery", "funeral_home"]
}
```

Luego, se definió una función que dado un permalink, la lista de establecimientos cercanos recuperados desde la *API* de Google Places, una serie de acumuladores y los mapeos de categorías creadas, determina la cantidad de lugares de interés cercanos a un inmueble.

```
def find_close_by_places(permalink, establishments,
categories_count, types_category, categories):

  # Inicializar una cuenta parcial de cada categoria en 0, para este registro
  row_count = {}
  for category in categories:
```

```
  row_count[category] = 0

# Por cada resultado, determinar la categoría a la que pertenece,
recorriendo las categorías disponibles
for entry in establishments :
  # Obtener los types del entry
  types = entry['types']
  # Para cada type, lo busco dentro de categories y sumo 1 en ese row_count
  for a_type in types:
    try:
      row_count[types_category[a_type]] += 1
    except:
      continue      # Since some types are not considered, we ignore the missing key

categories_count['permalink'].append(permalink)
for key in row_count:
  categories_count[key].append(row_count[key])
```

D. Anexo: BA Open Data

El problema de verificar si un inmueble pertenece a cierto barrio, se puede transpolar a determinar si un punto (x,y) pertenece a un polígono. Dado que las coordenadas del punto son conocidas, solo restaría obtener los vértices de los polígonos correspondientes a los limites de los barrios de Buenos Aires. Estos fueron obtenidos de la pagina oficial del Gobierno [12].

Para cada inmueble, se realizó el procesamiento pertinente, iterando sobre los polígonos, hasta encontrar al que contiene el punto. De esta forma, se encontró el barrio al que pertenece el inmueble según sus coordenadas.

MercadoLibre tiene subdivisiones de los barrios que no son reconocidas oficialmente, por ejemplo, el barrio de Palermo está dividido en Palermo Soho, Palermo Hollyhood, etc. Todos estos casos se corresponden con el único barrio de Palermo en el *dataset* de limites geográficos. Por este motivo, se conservarán ambos atributos.

Se encontraron casos adicionales en los cuales no se pudo obtener el *real neighbourhood*, dando cuenta de que se trataba de inmuebles cuyas coordenadas excedían los limites de Capital Federal, incluso los de Argentina, encontrando una serie de publicaciones que requerirán un tratamiento de limpieza.

```
from shapely.geometry import Polygon

def get_neighborhood(row, neighborhood_polygons,
    included_cities = ['Capital Federal'], excluded_neighborhoods = []):
  if ((row['city'] in included_cities) and  not (row['neighborhood']
  in excluded_neighborhoods)):
    try:
      for neighborhood_name in neighborhood_polygons:
        if (is_point_in_polygon(row['latitude'], row['longitude'],
        neighborhood_polygons[neighborhood_name])):
          return neighborhood_name
    except Exception as e:
      print(e)
  return ""

# Function to enrich the dataset with real neighborhood values, only for CABA properties
def enrich_dataset_with_real_caba_neighborhood(dataset, caba_neighborhoods):
  # Load once each neighbourhood
  neighborhood_polygons = {}
  for pol in caba_neighborhoods:
    neighborhood_polygons[pol['properties']['BARRIO'].capitalize()] =
        Polygon(pol['geometry']['coordinates'][0])
    dataset['real_neighborhood'] =
        dataset.apply(lambda row: get_neighborhood(row, neighborhood_polygons), axis=1)
```

Luego de cargar los límites geográficos de cada barrio de la Ciudad de Buenos Aires, la lógica implementada para determinar el barrio real de cada propiedad en base a sus coordenadas se reduce a las dos funciones previas, donde una de ellas itera por el *dataset* aplicando la segunda función,

[12] https://data.buenosaires.gob.ar/dataset/

que toma las coordenadas de la propiedad y recorre los polígonos de los barrios verificando en cuál de ellos se encuentra el punto.

E. Valores faltantes

E.1. Tipo de Casa y Tipo de departamento

Estos atributos son propios a los dos tipos de inmuebles que posee el *dataset*. Son mutuamente excluyentes y por ende el porcentaje de faltantes es alto. Todos aquellos correspondientes a casas poseen nulo el tipo de departamento, y viceversa. Por este motivo se creó un atributo adicional que los unifique.

Ahora bien, esta nueva *feature* aún contenía valores faltantes. Existiendo el atributo *property type* en el *dataset*, se podría optar por usar su valor para la imputación; no obstante, las categorías del mismo no son tan ricas como las de Tipo de Casa y Tipo de Departamento, por lo que se utilizó como último recurso. Previo a ello, se procesó el título de la publicación en pos de detectar la tipificación del inmueble, siendo común encontrar esta información en dicha sección. Esta alternativa permite mantener el dominio del nuevo atributo similar al de los anteriores.

E.2. Cocheras

El procedimiento consistió en procesar el título de las publicaciones en búsqueda de la mención de los términos cochera o estacionamiento. En caso de no encontrarlo, se optó por utilizar el atributo has_cochera que, como se ha visto, utiliza el mismo criterio pero derivado de la descripción. Por último, en los casos restantes donde no se logró definir el valor, se decidió asumir que, al tratarse de una característica valiosa en el inmueble, la ausencia de ella implica que la propiedad no cuenta con la misma, entendiendo que no es común indicar aspectos negativos en la publicación .

E.3. *Neighborhood*

Para realizar la imputación de *neighborhood* se utilizó el valor del atributo *real_neighborhood*.

E.4. Latitud y Longitud

Los registros con valores faltantes fueron eliminados del *dataset*, dado que no es posible validar si la ubicación indicada por el vendedor es real.

E.5. Cantidad de pisos

Para este atributo se llevaron a cabo agregaciones de dos naturalezas distintas. Por un lado, se consideró que las casas y los departamentos poseen una cantidad de pisos con distribuciones distintas y por el otro, que ciertas zonas de la Ciudad se caracterizan por edificios de alturas similares, incluso existiendo restricciones de construcción[13]. Por consiguiente, se agrupó por tipo de propiedad y barrio para realizar una imputación por la media.

E.6. Disposición

Hay ciertas disposiciones que son más valiosas que otras, debido a su diferencia en el confort y apariencia del inmueble. Dicho en otras palabras, muy probablemente un departamento con vista al frente posea más valor que uno con vista interna. Como consecuencia, es común encontrarnos

[13]https://buenosaires.gob.ar/desarrollourbano/noticias/las-10-claves-para-entender-que-cambia-en-buenos-aires-con-los-nuevos

con publicaciones que indiquen su orientación cuando esta es al frente, pero otras que oculten una disposición interna, similar a lo visto en el atributo de cochera.

Teniendo en cuenta lo dicho anteriormente la imputación debería tener cierto sesgo, en el que la probabilidad de que la disposición sea al frente sea mucho menor que la de interna, específicamente, su inversa. El mismo criterio se toma para el resto de los posibles valores del atributo. De esta manera, a la hora de imputar, en la mayor parte de los casos se aplicará el valor interno y en orden descendente, los restantes.

E.7. Expensas

Los departamentos suelen tener expensas asociadas y las casas no. Esto no se verifica en la totalidad de los casos, pero a los fines del trabajo no debería tener un impacto alto. Por lo tanto, en las casas los valores faltantes de expensas se imputaron por 0. Para departamentos, por otro lado, la imputación se realizó usando la media por barrio.

E.8. Ambientes y *rooms*

Ambos atributos hacen referencia a una misma característica del inmueble, pero como rooms es un valor obtenido desde la *API*, fue el utilizado por defecto. Sin embargo, en algunos casos este no está presente y la imputación puede hacerse con el valor correspondiente en el atributo Ambientes.

Ahora bien, suele ocurrir que - especialmente en departamentos - la cantidad de ambientes tiene una relación con la cantidad de dormitorios. Esta información es útil en casos donde no están rooms ni Ambientes. Se analizó esto en el *dataset* y se determinó que los departamentos con hasta 5 dormitorios, suelen tener N+1 ambientes, por lo que se realizó la imputación de esta forma. En el resto de los casos, se reemplazó los valores nulos con la media agrupada por precio.

E.9. Dormitorios y *bedrooms*

Nuevamente el atributo preferido fue el que se obtuvo de la *API* frente a la discrepancia de valores. A su vez, esto sirvió para imputar aquellos registros con dormitorios en nulo. Sin embargo, cuando *bedrooms* también tenía un valor nulo, se tomó el siguiente criterio:

Si ambientes es 1, *bedrooms* será 1. Si ambientes es menor o igual a 4, *bedrooms* será una unidad menor. Si no, *bedrooms* será la media de ambientes agrupada por precio decrementada en 1.

E.10. Antigüedad

Una porción de los faltantes tienen una condición "Nuevo", por lo que pasan a tener una antigüedad igual a 0.

Para el resto, considerando el fenómeno que suele ocurrir en la ciudad de Buenos Aires respecto a construcciones de cierto barrio que datan de la misma fecha, la imputación fue por la media de antigüedad agrupada por barrio.

E.11. *Item condition*

Así como es posible derivar antigüedad a partir de *item condition*, la análoga también al considerar que una antigüedad mayor a 0 implica que el inmueble es "Usado". De lo contrario, es "Nuevo".

E.12. Superficie Cubierta, *Covered Area*, Superficie Total, *Total Area*

Cuando se trata de los pares de atributos Superficie Cubierta y *Covered_Area*, como también, Superficie Total y *Total_Area*, son preferibles los valores que provienen de la *API*.

Sin embargo, para los casos que dichos pares difieran entre sí y haya una amplia divergencia, el valor final es el promedio entre ellos.

Por otro lado, para los valores faltantes de dichos atributos, para realizar la imputación, se utilizó la media, agrupada por *property_type*, *neighborhood*, *discretized_price*, y Antigüedad_discretizada.

E.13. Baños y *Full_Bathrooms*

Tratándose nuevamente de dos atributos con la misma semántica, se prefirió el de la *API* si está presente, es decir, *full_bathrooms*. Sabiendo que la cantidad de baños de un inmueble es proporcional a la cantidad de ambientes, se aprovechó esta característica y para los valores faltantes, se realizó una imputación por la media agrupada por tal atributo.

E.14. *Seller ID*

Podría argumentarse que imputar un ID no tiene sentido, pero mantener el atributo posibilita a realizar 2 experimentos: agrupar publicaciones por vendedor y determinar características comunes, o bien generar relaciones a partir de la antigüedad de pertenencia al sitio del vendedor.

Por lo tanto, para completar los valores nulos, se incrementó en 1 el valor máximo, por cada registro faltante.

E.15. *Seller_sales*, *Seller_cancelations*, *Seller_handling_time*, *Seller_claims*, *Seller_state* y *seller_city*

Cuando se trata de los atributos *Seller_sales*, *Seller_cancelations*, *Seller_handling_time*, *Seller_claims*, se realizó una imputación de valores faltantes a partir de la media general del *dataset* en cada uno de ellos. Son todos valores discretos, por lo que se realizó un redondeo antes de asignarlos. Por ultimo, para *Seller_state* y *seller_city* se imputó por la moda.

F. Mejores parámetros modelo separado por barrios

Neighborhood 20 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 4 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 7 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1750, 'random_state': 18

Neighborhood 1 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2200, 'random_state': 18

Neighborhood 2 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 46 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2200, 'random_state': 18

Neighborhood 12 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2200, 'random_state': 18

Neighborhood 27 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2200, 'random_state': 18

Neighborhood 19 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 35 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 10 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 29 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18

Neighborhood 31 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18

Neighborhood 14 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 36 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18

Neighborhood 37 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2100, 'random_state': 18

Neighborhood 16 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 40 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18

Neighborhood 30 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 8 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18

Neighborhood 22 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18

Neighborhood 3 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1900, 'random_state': 18

Neighborhood 28 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1900, 'random_state': 18

Neighborhood 6 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 2200, 'random_state': 18
Neighborhood 11 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1900, 'random_state': 18
Neighborhood 17 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18
Neighborhood 15 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1750, 'random_state': 18
Neighborhood 42 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18
Neighborhood 41 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1750, 'random_state': 18
Neighborhood 9 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2100, 'random_state': 18
Neighborhood 32 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18
Neighborhood 38 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2200, 'random_state': 18
Neighborhood 33 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1900, 'random_state': 18
Neighborhood 39 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18
Neighborhood 13 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18
Neighborhood 45 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18
Neighborhood 5 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 2000, 'random_state': 18
Neighborhood 24 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 2100, 'random_state': 18
Neighborhood 21 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18
Neighborhood 23 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2200, 'random_state': 18
Neighborhood 25 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18
Neighborhood 26 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 3, 'n_estimators': 2200, 'random_state': 18
Neighborhood 0 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18
Neighborhood 18 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1900, 'random_state': 18
Neighborhood 34 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 2000, 'random_state': 18
Neighborhood 43 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1500, 'random_state': 18
Neighborhood 44 best params: 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1900, 'random_state': 18