INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA
ESCUELA DE GESTIÓN Y TECNOLOGÍA

Modelo de Optimización de Precios para Anfitriones de Airbnb

AUTORES

Fiorellino, Delfina (Leg. N° 60465)

Pruden, Valentina (Leg. N° 60769)

Vidal, Rosario (Leg. N° 60369)

DOCENTES Y TUTORES

Rodríguez Varela, Juan Pablo

Gonzáles Rodríguez, Rubén Darío

Brottier, Ignacio

PROYECTO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE LICENCIADO/A EN ANALITICA EMPRESARIA Y SOCIAL

BUENOS AIRES

PRIMER CUATRIMESTRE, 2023

ÍNDICE

Objetivo del Proyecto

Marcelo Marcone se dedica a administrar 4 propiedades en el barrio de Belgrano en Buenos Aires a través de la plataforma de **Airbnb**. Si bien su negocio es rentable, actualmente tiene la dificultad de definir precios óptimos para sus propiedades: si fija precios demasiado *altos*, es posible que no se atraigan suficientes huéspedes y pierda oportunidades de ingresos mientras que si fija precios demasiado *bajos*, puede terminar perdiendo la oportunidad de obtener más dinero por los alquileres.

Hasta el día de hoy, Marcelo define precios en base a su experiencia personal como anfitrión, habiendo observado a lo largo de los años qué precios funcionan para cada uno de sus alojamientos en cada periodo del año. Debido a esto, Marcelo tiene el objetivo de mejorar su negocio fijando precios óptimos para sus departamentos tomando decisiones con más fundamento que su experiencia; quiere fijar precios en función de la demanda de la competencia, que a su vez se ve afectada por eventos que tienen lugar en el barrio o en la ciudad (por ejemplo, maratones o conciertos) en fechas particulares.

En este contexto, el objetivo del proyecto es brindarle a Marcelo una herramienta para que logre posicionarse de la mejor manera posible en un mercado altamente competitivo como Airbnb, ayudando a definir precios óptimos para el alquiler de sus propiedades. Mediante la clusterización de las barrios y el análisis de la demanda, se desarrollará un **modelo de predicción de cantidad de días de ocupación en una semana determinada** en función de diferentes parámetros, como: precios promedio por noche, huéspedes, tipo de estadía, rating promedio entre otros. Este modelo a su vez considerará la demanda de alquileres a lo largo del año. De esta manera Marcelo podrá encontrar el equilibrio entre precios competitivos y rentables.

Medición de Valor y KPIs a Impactar

Para medir el éxito del proyecto y su impacto en el negocio, se pueden utilizar los siguientes KPIs:

- **Diferencia de ingresos totales:** indica el valor económico generado por el proyecto al mostrar cuánto varía el ingreso total del anfitrión cuando determina precios utilizando el modelo.

$$Dif.\ de\ ingresos\ totales\ [mes\ i]\ =\ ICH\ [mes\ i]\ -\ ISH[mes\ i]$$

$$ICH\ [mes\ i]\colon\ Ingresos\ del\ mes\ i\ con\ el\ uso\ de\ la\ herramienta$$
$$ISH\ [mes\ i]\colon\ Ingresos\ del\ mes\ i\ sin\ el\ uso\ de\ la\ herramienta$$
$$1 \leq i \leq 12$$

- **Diferencia de porcentaje de ocupación:** indica cuánto varía la ocupación o demanda de cada propiedad al ajustar su precio. ¿Cómo cambia la ocupación de las propiedades al establecer precios óptimos?

$$Dif.\ de\ porcentaje\ de\ ocupación\ [mes\ i]\ =\ PCH\ [mes\ i]\ -\ PSH[mes\ i]$$

$$PCH\ [mes\ i]\colon\ Porcentaje\ de\ ocupación\ del\ mes\ i\ con\ el\ uso\ de\ la\ herramienta$$
$$PSH\ \ [mes\ i]\colon\ \ Porcentaje\ de\ ocupación\ del\ mes\ i\ sin\ el\ uso\ de\ la\ herramienta$$
$$1 \leq i \leq 12$$

- **Precio promedio por noche:** muestra si el anfitrión está obteniendo un precio justo por su propiedad y si los precios son competitivos en comparación con otros alojamientos similares en la misma área (barrio, cantidad de huéspedes admitidos, comodidades, rating).

$$Dif.\ de\ precio\ promedio\ por\ noche\ [mes\ i]\ =\ PNCH\ [mes\ i]\ -\ PNSH[mes\ i]$$

$$PNCH\ [mes\ i]:\ Precio\ promedio\ por\ noche\ del\ mes\ i\ con\ el\ uso\ de\ la\ herramienta$$
$$PNSH\ \ [mes\ i]:\ \ Precio\ promedio\ por\ noche\ del\ mes\ i\ sin\ el\ uso\ de\ la\ herramienta$$
$$1 \leq i \leq 12$$

- **Reseñas y puntuaciones:** mide la satisfacción de los huéspedes después de la implementación del modelo de predicción de precios. Un aumento en las puntuaciones y comentarios positivos, o una mayor proporción de estos con respecto a los negativos, indicaría que el anfitrión está proporcionando precios acordes a las características y la demanda de la propiedad, brindando una experiencia satisfactoria para los huéspedes.
- **Cantidad de visitas:** Airbnb registra la cantidad de veces que una persona hace "click" en un alojamiento. Una mayor cantidad de "clicks" en un alojamiento es indicador de un crecimiento en su demanda, que puede explicarse por la definición de un precio óptimo para la propiedad.
- **Cantidad de mensajes:** la cantidad de mensajes que el anfitrión recibe es indicador de la demanda de sus alojamientos. Si aumenta la cantidad de mensajes, puede significar que el anfitrión definió un precio óptimo para la propiedad.
- **Coeficiente de determinación (R^2):** es un estadístico usado en el contexto de un modelo estadístico cuyo principal propósito es predecir futuros resultados o probar una hipótesis.
- **Raíz del error cuadrático medio (RMSE):** mide la precisión del modelo de regresión, se utiliza para medir la diferencia entre el valor predicho y el real.

Entregables y Outputs del Proyecto

El entregable principal del proyecto será una herramienta basada en un **modelo de predicción de cantidad de días de ocupación en una semana determinada** que, entrenado con datos históricos de reservas de propiedades de Airbnb, devuelva la cantidad de días de ocupación de cada propiedad en determinadas semanas mediante diferentes parámetros. La complejización del modelo no sólo comprende el entrenamiento con reservas históricas, sino que para predecir la cantidad de días de ocupación también se considerarán características de las propiedades (cantidad de cuartos, cantidad de baños, barrio, etc). De esta manera, Marcelo podrá establecer precios tomando el riesgo de ocupación que él desee.

Además de la herramienta descripta, se entregará este informe al cliente con el análisis del trabajo desarrollado. Este informe es de valor siendo que analizará los factores influyentes en la demanda y precio de un alojamiento, lo que puede llevar a que Marcelo tome mejores decisiones y conozca los insights del proyecto.

Investigación de Metodologías y Publicaciones Académicas Relevantes

Se encontraron algunas publicaciones que pueden ser de utilidad para un mayor entendimiento y desarrollo del trabajo:

- *"What do guests value most in Airbnb accommodations? An application of the hedonic pricing approach"* de Dogru, T., & Pekin, O. (2017): Este artículo indaga sobre los determinantes de precios en servicios como Airbnb y los aspectos que los huéspedes más valoran cuando van a hospedarse a través de la plataforma. Entre los más importantes, y los que motivan a pagar más por un alquiler, se encuentran: el espacio y privacidad, experiencias únicas y limpieza.

- *"Use of dynamic pricing strategies by Airbnb hosts"* de Gibbs, C., Guttentag, D., Gretzel, U. Yao, L. & Morton, J. (2018): Este artículo analiza las estrategias dinámicas de precios usadas por los anfitriones de Airbnb y compara este comportamiento con el de los hoteles tradicionales. El análisis demuestra que el uso de estrategias de variación de precios es limitado entre los anfitriones, quienes podrían beneficiarse de adoptar otras estrategias de fijación de precios.

- *"A Machine Learning Model for Occupancy Rates and Demand Forecasting in the Hospitality Industry"* de Caicedo-Torres, W. & Payares, F. (2016): El forecasting de la tasa de ocupación es un paso indispensable en el proceso de toma de decisiones de los planificadores y administradores de hoteles, dando lugar a políticas de fijación de precios más eficientes. Este artículo investiga el desarrollo de modelos de machine learning para predecir las tasas de ocupación y la demanda en la industria hotelera. Para esto se probaron diferentes algoritmos como Kernel Ridge Regression, Multilayer Perceptron y Ridge Regression, concluyendo que este último fue el más eficaz.

Abordaje del problema

El problema de Marcelo se abordará utilizando datos publicados por Inside Airbnb.

Para el acertado análisis de los datos es necesario un correcto abordaje de la limpieza de los mismos. Es por ello que analizaremos los valores faltantes y outliers de estos, modificándolos en el grado que consideremos pertinente.

Luego se realizarán correlaciones y tests estadísticos para entender la relación entre las variables y se clusterizarán los barrios en una cantidad de grupos conveniente para la posterior predicción de la ocupación utilizando el algoritmo de K-Medias.

Se probarán los siguientes algoritmos predictivos:

- Random forest
- XGBoost
- Red neuronal

Se compararán estos modelos y se seleccionará aquel que tenga mejor capacidad de predicción en base al error de predicción de cada uno. El modelo final se integrará a una herramienta en la que Marcelo podrá ingresar diferentes parámetros (tamaño y comodidades del alojamiento, tipo de alquiler, precio, entre otros) para obtener la cantidad de días que se espera que su alojamiento esté ocupado en determinada semana.

Plan de Trabajo

Para el desarrollo de la solución planteada se seguirá una serie de pasos:

1. **Recolección y reestructuración de datos** (2 días). Si bien los conjuntos de datos a utilizar ya se encuentran disponibles para su descarga, uno de ellos es actualizado trimestralmente y requiere de ser combinado con sus versiones anteriores.

2. **Limpieza de datos** (3 días)**.** Una vez hecho esto y descargados los datasets, se analizará la relevancia de los *valores faltantes* y *outliers*, modificándolos o eliminándolos en los casos que consideremos pertinentes.

3. **Análisis de datos** (8 días)**.** Se estudiará el comportamiento y estructura de los datos.

   Algunas preguntas a abordar en esta instancia son:
   - ¿Qué características (precio, comodidades, competencia, demanda, etc) tienen las propiedades según el barrio?
   - ¿Cómo varían las reservas de las propiedades en determinadas fechas o por temporada?

   Para responder estas preguntas se realizarán los siguientes análisis:
   a. **Análisis *univariado* y *multivariado* de las variables**. Mientras en el primer análisis se estudiará la distribución de las variables individuales, en el segundo se buscará añadir complejidad introduciendo *correlaciones*, *patrones* y *tests estadísticos* para evaluar la relación entre variables.
   b. **Clustering de barrios.** Se segmenta a los barrios dadas tres características: el precio por noche por persona, su desviación y demanda histórica.

4. **Modelar la proyección de Parámetros vs. Ocupación** (10 días)**.** Una vez entendido el comportamiento de las variables y, en particular, su comportamiento para cada grupo de alojamientos, se construirán diferentes modelos para calcular la cantidad de días de ocupación de una determinada propiedad para cada semana del año, en base a diversos parámetros. Es importante que dichos modelos contemplen la estacionalidad de la demanda.

5. **Validación de resultados** (4 días)**.** La predicción de los modelos será validada utilizando datos históricos de reservas de Airbnb. El modelo seleccionado será aquel con mayor R^2 y menor RMSE.

6. **Integración de la herramienta** (7 días)**.** El modelo seleccionado en la etapa anterior será integrado a una herramienta para el uso personal del anfitrión, Marcelo.

7. **Presentación del proyecto al cliente y ajustes de herramienta** (7 días)**.** La herramienta desarrollada será presentada a Marcelo para su evaluación final. El feedback del cliente llevará a perfeccionar la herramienta y hacer los últimos ajustes.

8. **Presentación del proyecto a peer group** (7 días). A partir de los ajustes realizados en la herramienta se creará una presentación que se compartirá con un grupo de compañeros de la materia.

9. **Rearmado de informe** (14 días). Se actualizará el informe de acuerdo al trabajo realizado, para dar cohesión. Se tomarán las sugerencias y se realizarán las modificaciones recomendadas en la medida de lo posible para dar cierre al proyecto.

Para la realización de las tareas, se seguirá una metodología Agile, lo que permitirá adaptarse a los cambios en los requisitos del proyecto y reducir los riesgos. Si bien se define una estructura de fechas fijas en el diagrama de Gantt, los sprints representan un proceso iterativo de retroalimentación sobre las tareas, permitiendo volver atrás a hacer correcciones sobre el trabajo realizado. Es por esto que tampoco se especifica todo el detalle de las tareas, ya que están sujetas a cambios. Esto permite una mayor flexibilidad y adaptabilidad a medida que se avanza en el proyecto y responde a los cambios en el camino.

Bajo esté enfoque, la planificación de esté proyecto consiste en 4 sprints: 3 de 2 semanas y uno de 3. A continuación se muestra el diagrama de Gantt, en el que además de las tareas se pueden ver las fechas de los entregables a desarrollar:

	Etapa	Fecha inicio	Fecha Fin	Duración
Sprint 1				
1	Recolección y reestructuración de datos	3/04/2023	4/04/2023	2 días
2	Limpieza de datos	6/04/2023	8/04/2023	3 días
3	Análisis de datos	9/04/2023	16/04/2023	8 días
Segundo entregable		**17/4/2023**	**17/4/2023**	
Sprint 2				
4	Modelar la proyección de Precio vs. Ocupación	17/04/2023	26/04/2023	10 días
5	Validación de resultados	27/04/2023	30/04/2023	4 días
Tercer entregable		**8/5/2023**	**8/5/2023**	
Sprint 3				
6	Integración de la herramienta	1/04/2023	7/04/2023	7 días
7	Presentación del proyecto al cliente y ajustes de herramienta	8/05/2023	14/5/2023	7 días
Cuarto entregable		**22/5/2023**	**22/5/2023**	
Sprint 3				
8	Presentación del proyecto al peer group	15/05/2023	21/05/2023	7 días
9	Rearmado de informe	22/05/2023	4/06/2023	14 días
Quinto entregable		**5/6/2023**	**5/6/2023**	

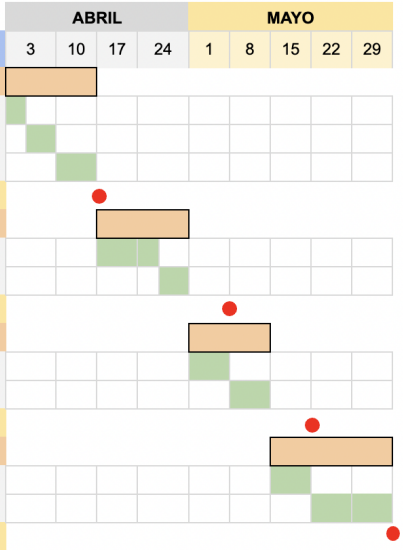


Fig 1. *Diagrama de Gantt del proyecto*

Datos a Utilizar

Se analizarán los datasets provistos por **Inside Airbnb**, un proyecto creado con el objetivo de cuantificar y entender el impacto de Airbnb en las comunidades. Los datasets a utilizar son: **calendar** y **listings**, que tienen su última actualización el 23 de marzo de 2023.

- **calendar:** tiene el detalle de la disponibilidad de cada departamento desde el día en que se levantan los datos hasta los 365 días que le siguen. Además, cuenta con el detalle de precio por noche y cantidad mínima de días que el anfitrión habilita una reserva, entre otros.

  Frecuencia de actualización: trimestral.
  Última actualización: marzo de 2023.

Limitaciones:

- El dataset muestra la disponibilidad de los alojamientos a futuro, por lo que no se registra la efectivización de una reserva que puede ser cancelada luego de la actualización de los datos.

- **listings:** detalla las características de cada alojamiento de la ciudad registrado en Airbnb. Entre algunas de ellas se pueden encontrar: el barrio, la cantidad de personas que aloja, las comodidades, calificaciones de servicio y datos del anfitrión.

  Frecuencia de actualización: trimestral.
  Última actualización: marzo de 2023.

Premisas y Simplificaciones

Debido a que el dataset no permite ver si existieron cancelaciones en las reservas se asume que no hay cancelaciones dentro de los 3 meses. Por ende, ya que **calendar** se actualiza trimestralmente no hay cancelaciones a considerar.

Herramientas a Utilizar

El análisis de los datos se llevará adelante utilizando el lenguaje de programación R. Además, se podrían usar otras herramientas como Eviews para el análisis econométrico como series de tiempo y Power BI para la visualización y presentación de resultados.

Análisis

Para resolver el problema de Marcelo se utilizarán 2 datasets: **listings** y **calendar**.

Listings

El dataset de **listings** cuenta con **22.713 observaciones** y **75 variables**, en la Tabla 1 se puede observar los primeros seis registros, es decir 6 alojamientos, con las 6 primeras variables de la base:

	id	listing_url	scrape_id	last_scraped	source	name
1	130424	https://www.airbnb.com/rooms/130424	2.023033e+13	2023-03-30	city scrape	Palermo Hollywood Studio Apt - 4th floor - 33 m2
2	131145	https://www.airbnb.com/rooms/131145	2.023033e+13	2023-03-29	city scrape	DONDE VIVI (1) Rooms for Rent .
3	132298	https://www.airbnb.com/rooms/132298	2.023033e+13	2023-03-30	city scrape	Urban Escape ★ Studio with great views
4	317509	https://www.airbnb.com/rooms/317509	2.023033e+13	2023-03-29	city scrape	Comfortable house in Buenos Aires
5	133043	https://www.airbnb.com/rooms/133043	2.023033e+13	2023-03-29	city scrape	Great Studio - Roof top Pool
6	11508	https://www.airbnb.com/rooms/11508	2.023033e+13	2023-03-30	city scrape	Amazing Luxurious Apt-Palermo Soho

Tabla 1. *Dataset* ***listings***.

Estas 22.713 observaciones representan cada uno de los alojamientos que se encuentran en la Ciudad de Buenos Aires. Y la vastedad de las variables que se encuentran en este dataset se debe a la exhaustiva descripción que hay sobre cada uno de estos alojamientos en cuestión.

En el caso de este proyecto, no estaremos trabajando con las 75 variables ya que gran parte de ellas no proporcionan ninguna información útil para el alcance de los objetivos. Por ejemplo, las variables: *listing_url*, *picture_url*, *last_scraped*, *host_has_profile_pic*, entre otras, no darían ningún análisis estadísticamente significativo a la hora de predecir la probabilidad de alquiler de un alojamiento determinado en una semana particular o realizar clusters de los barrios. Es por esto que se redujo el dataset de **listings** a 37 variables.

Grupo	Variable	Descripción	Tipo de Dato
Barrio	neighborhood_overview	La descripción que hace el host sobre el barrio	String
	neighborhood_cleansed	El barrio donde está ubicada la propiedad. Tiene en cuenta la latitud y longitud de la misma.	String
Host	host_id	Identificador único de Airbnb para el host	Int
	host_since	La fecha en la cual el host fue ingresado. Para hosts que eran antes huéspedes, esta fecha puede representar el día en el cual se registraron como huéspedes.	Date
	host_is_superhost	Si el host es categorizado como superhost.	Boolean
	host_identity_verified	Si el host fue verificado por AirBnb	Boolean
Propiedad	id	Identificador único de Airbnb para la propiedad.	Int
	description	Detallada descripción del alojamiento	String
	latitude	Latitud	Numeric
	longitude	Longitud	Numeric
	property_type	Tipo de propiedad	String
	room_type	Tipo de cuarto. Puede pertenecer a las siguientes categorías (Entire place, Private rooms, Shared rooms)	String
	accommodates	Cuantas personas entran	Numeric
	bedrooms	Número de cuartos	Numeric
	amenities	Comodidades que tiene el alojamiento	String
	price	Precio por noche	Character
	minimum_nights	Noches mínimas para reservar	Numeric
	maximum_nights	Noches máximas para reservar	Numeric
	has_availability	Si está disponible	Boolean
	availability_30	La cantidad de días en el cual el alojamiento estará disponible en los próximos 30 días.	Numeric
	availability_60	La cantidad de días en el cual el alojamiento estará disponible en los próximos 60 días.	Numeric

availability_90	La cantidad de días en el cual el alojamiento está disponible en los próximos 90 días.	Numeric
availability_365	La cantidad de días en el cual el alojamiento está disponible en los próximos 365 días.	Numeric
number_of_reviews	Cantidad de reviews	Numeric
number_of_reviews_ltm	Cantidad de reviews en los últimos 12 meses	Numeric
number_of_reviews_l30d	Cantidad de reviews en los últimos 30 días.	Numeric
review_scores_rating	Calificación promedio que tiene el alojamiento teniendo en cuenta todas las otras calificaciones.	Numeric
review_scores_accuracy	Calificación promedio que tiene el alojamiento teniendo en cuenta que tan verídico es el alojamiento en las fotos vs la vida real.	Numeric
review_scores_cleanliness	Calificación promedio que tiene el alojamiento teniendo en cuenta la limpieza.	Numeric
review_scores_checkin	Calificación promedio que tiene el alojamiento teniendo en cuenta el check in.	Numeric
review_scores_communication	Calificación promedio que tiene el alojamiento teniendo en cuenta la comunicación.	Numeric
review_scores_location	Calificación promedio que tiene el alojamiento teniendo en cuenta la ubicación.	Numeric
review_scores_value	Calificación promedio que tiene el alojamiento teniendo en cuenta el valor.	Numeric
instant_bookable	Si es reservable al instante	Boolean
calculated_host_listings_count	Cantidad de alojamientos que un mismo host tiene	Numeric
bathrooms	Número de baños	Numeric

***Tabla 2**. Descripción de variables de **listings**.*

Análisis de valores faltantes

Teniendo ahora un mejor conocimiento de las variables que serán de ayuda para lograr los objetivos del trabajo, se procede a analizar los valores faltantes que se encuentran en cada una de las variables. Para esto se confeccionó una tabla en donde por un lado se presentan las variables y por el otro la cantidad de valores faltantes que tienen.

A continuación se puede observar esta tabla teniendo en cuenta sólo las variables que tienen valores faltantes:

Variable	Cantidad
description	561

neighborhood_overview	9.943
bathrooms_text	25
bedrooms	3.058
beds	232
review_scores_rating	4.122
review_scores_accuracy	4.202
review_scores_cleanliness	4.202
review_scores_checkin	4.202
review_scores_communication	4.201
review_scores_location	4.201
review_scores_value	4.202

***Tabla 3**. Cantidad de valores faltantes por variable.*

Se comenzó analizando por qué es que las variables: *review_scores_value, review_scores_location, review_scores_communication, review_scores_checkin, review_scores_cleanliness, review_scores_accuracy, review_scores_rating*, tienen un número casi idéntico de valores faltantes. Ante esto se descubrió que esto se debía a que estos alojamientos nunca habían recibido reseñas, por lo que no iban a tener ningún tipo de calificación en cuanto a esto.

Una de las maneras en las cuales se pueden tratar los valores faltantes es eliminandolos del dataset. Pero debido a que quitando todos estos registros se eliminaría casi el 20% de los alojamiento, se decidió imputarlos por el valor de la media o mediana de la variable dependiendo del resultado de hacer un test de Kolmogorov-Smirnov.

En está línea, el primer pasó fue realizar el test de Kolmogorov-Smirnov para entender si cada una de estas variables seguían una distribución normal o no. El motivo por el cual se requiere saber esto es para decidir si reemplazar los valores faltantes por la media o por la mediana de la muestra. Si el *valor-p* del test es menor a 0,05, se rechaza la hipótesis nula y se concluye que la variable no tiene una distribución normal.

Cada una de estas variables dio un *valor-p* menor a 0,05, por lo que cada uno de estos valores faltantes fue cambiado por la mediana de su respectiva variable. La mediana se calculó a partir del comportamiento que tienen los alojamientos según el mismo tipo de cuarto (*room_type*) y el barrio (*neighbourhood_cleansed*) en donde se encuentra. Este mismo tratamiento fue realizado con las variables de *bedroom* & *bathroom*.

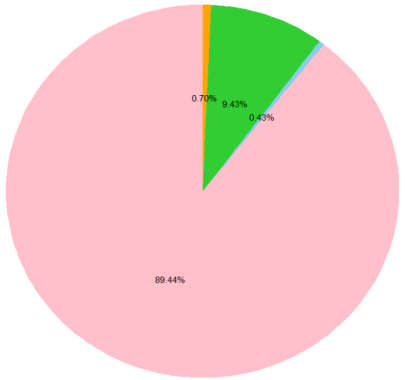
Por otro lado, se decidió eliminar la variable *beds* y suponer que habría la misma cantidad de camas que personas que se pueden acomodar en el alojamiento, donde este valor se puede encontrar en la variable *accommodates*.

Luego a las variables *description* y *neighborhood_overview* no les fue realizado ningún tipo de tratamiento, ya que estos datos faltantes no eran necesarios cambiarlos ni eliminarlos, debido a que el análisis que se les haría es una nube de palabras. Esto se debe a que es necesario entender más que nada que es lo que los hosts consideran importante a la hora de remarcar qué es lo que sus alojamientos tienen y cómo es su barrio.

Análisis de las variables categóricas y continuas

Una vez teniendo la limpieza hecha de los valores faltantes de la base de datos de **listings**, se continuó analizando las variables categóricas y la frecuencia en la cual cada uno de sus datos aparecen.

En primer lugar se realizó un pie chart de la variable *room_type*. Esto para ver más que nada como es la distribución de los 4 posibles grupos en el cual un alojamiento puede pertenecer. Esta información se puede visualizar a continuación.



room_type <chr>	n <int>
Entire home/apt	20282
Hotel room	98
Private room	2139
Shared room	158

Fig 2. *Pie Chart y Tabla de room_type.*

A partir de lo anterior se observa que casi la totalidad de alojamientos registrados en **listings** son apartamentos y casas. Esto tiene sentido siendo que las personas suelen alquilar un apartamento o una casa cuando están viajando o cuando se están mudando.

En cuanto a la variable categórica de *neighbourhood_cleansed*, se realizó un gráfico de frecuencia de los 10 barrios que tienen la mayor cantidad de alojamientos, como se puede ver a continuación. No se incluyeron todos los barrios, que llegaban a una totalidad de 44, ya que la visualización del gráfico sería poco clara. En este caso el barrio más popular sería Palermo, y Belgrano, barrio en donde se encuentran los 4 alojamientos de Marcelo, es el cuarto barrio con mayores alojamientos en Airbnb.

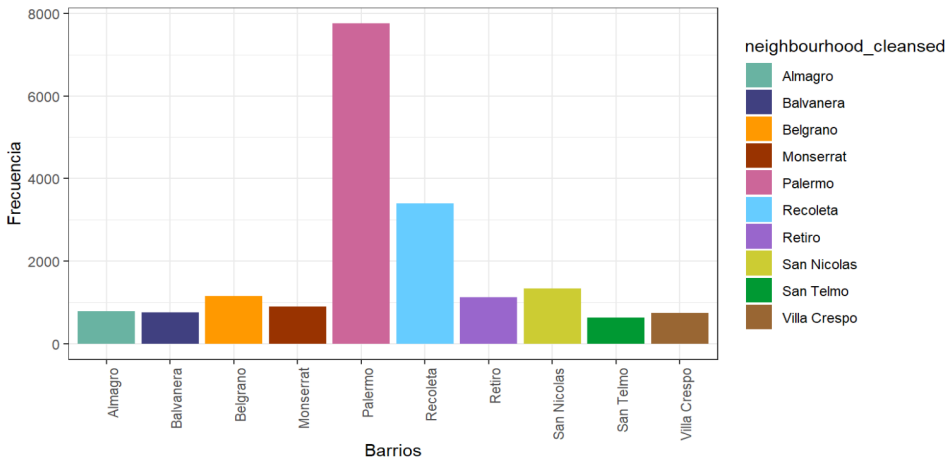


Fig 3. *Gráfico de frecuencias Barrios.*

Siguiendo ahora con la variable *property_type*, se realizó otro gráfico de frecuencias, como se puede observar en la Figura 4. En este caso más de la mitad de los alojamientos son una unidad de alquiler completa, categoría en la cual los alojamientos de Marcelo entran. El resto de tipo de propiedades se distribuye de una manera muy similar cuando se habla de cantidades.

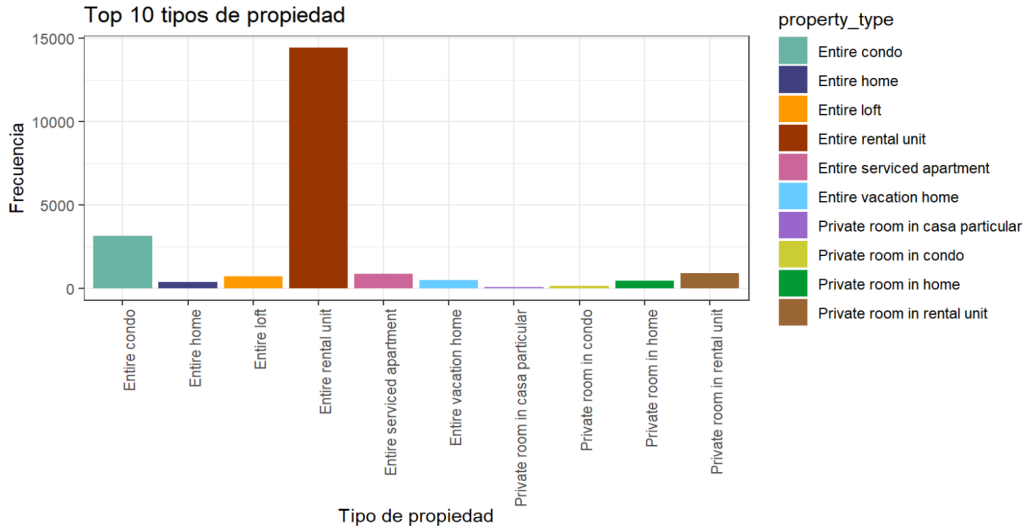


Fig 4. *Gráfico de frecuencias de tipo de propiedad.*

Se continuó analizando cuántos de los anfitriones son “Super Hosts”. Para que lo sean, estos deben cumplir ciertas condiciones, y por lo general se le otorga este título a aquellos anfitriones que son más experimentados, hospitalarios y mejor valorados. Estos anfitriones reciben un distintivo especial en su anuncio y su perfil que les otorga una mayor visibilidad en la plataforma. En este caso muy pocos anfitriones, casi un tercio de ellos, son considerados “Super Hosts” Afortunadamente Marcelo entra en esta categoría por lo que esto le permite tener una ventaja frente a sus competidores.

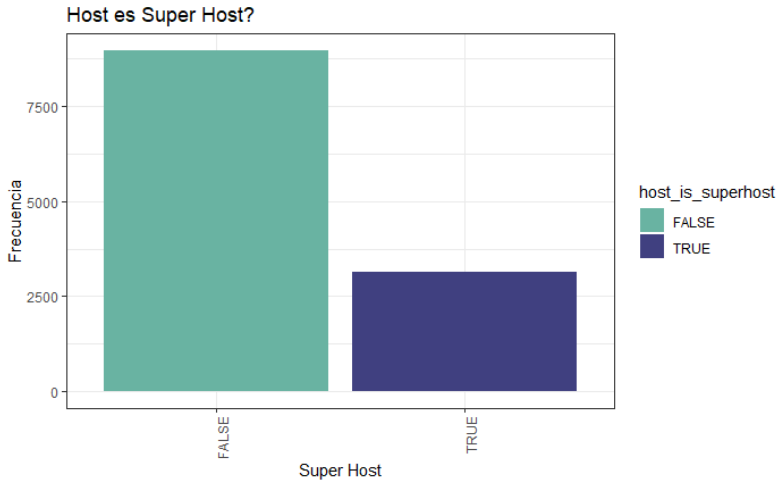


Fig 5. *Gráfico de frecuencias de SuperHost*

Luego se analizó la variable de si el alojamiento puede ser alquilado al instante o no. Esto permite que los huéspedes reserven el alojamiento al instante en las fechas disponibles para que los anfitriones no tengan que revisar y aceptar cada solicitud de reservación por separado. En este caso muy pocos de ellos tienen habilitada esta opción.

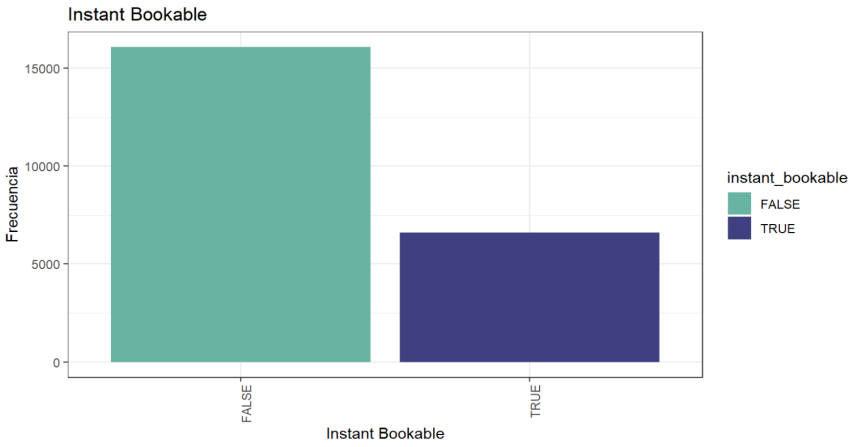


Fig 6. *Gráfico de frecuencias de tipo de reservación inmediata*

Por último, se realizó un análisis para las variables *description* y *amenities*. En primer lugar, la variable *description* consiste en un texto escrito por el anfitrión, quien muchas veces presenta información redundante sobre el alojamiento, como la cantidad de cuartos, baños o el barrio del mismo. Por otro lado, la variable *amenities* especifica las comodidades del alojamiento, tales como si tiene wifi, horno, vajilla o televisor.

Ya que para alcanzar el objetivo del presente trabajo es de gran importancia entender el comportamiento de la competencia en Airbnb, se decidió hacer 2 nubes de palabras para visualizar la importancia de algunas características y comodidades de los alojamientos. En este caso, ambas variables dan a entender qué palabras utilizan los anfitriones para llamar la atención de posibles huéspedes hacia sus alojamientos.

Se puede observar como para la variable *description*, muchos anfitriones consideran que es importante mencionar el espacio de su alojamiento. A su vez se resaltan palabras como “palermo”, “apartment” y “departamento”, haciendo referencia al barrio con mayor cantidad de alojamientos y tipos de propiedad más ofertados.

Por otro lado, las comodidades que más se mencionan en los alojamientos son “agua caliente”, “cocina”, “freezer”, “wifi”, “parking”, entre otros. Esto da a entender qué características son las más comunes, por lo que la falta de ellas puede repercutir en bajas reservas.

Fig 7. *Nube de palabras de description*

Fig 8. *Nube de palabras de amenities*

Ya que gran parte de este proyecto y la herramienta que se confeccionara girará en torno a la variable precio, a continuación se analiza dicha variable, y como la misma se distribuye y se comporta cuando está relacionada a los barrios, tipo de propiedad, tipo de cuarto y si el host es o no superhost.

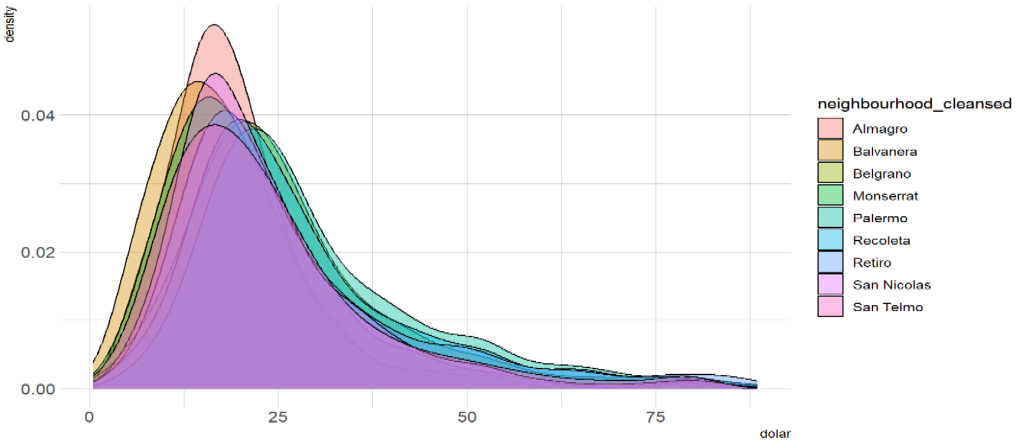


Fig 9. *Gráfico de densidad de dólares por noche para los 10 barrios con más alojamientos.*

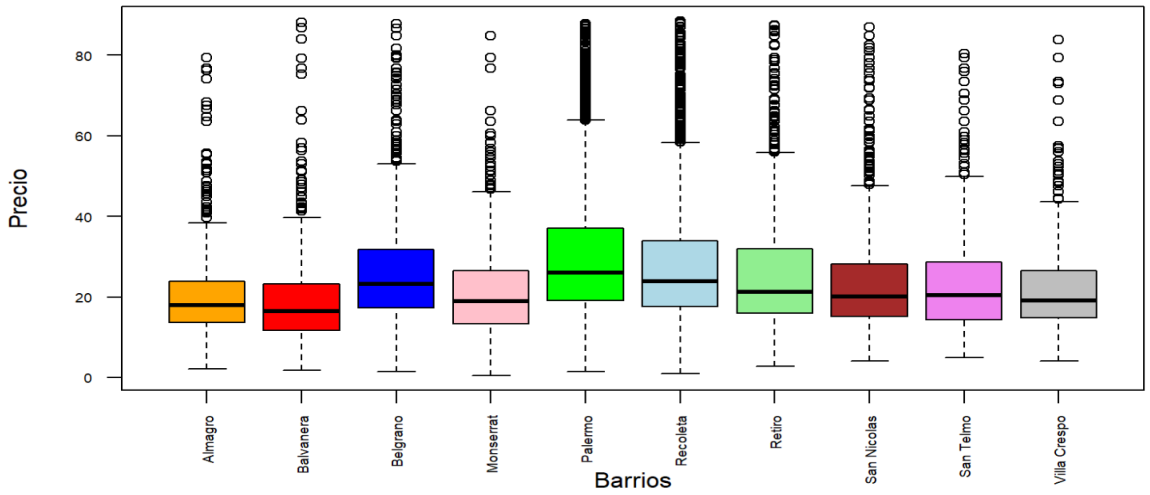


Fig 10. *Box plot de dólares por noche para los 10 barrios con más alojamientos.*

Como se puede observar en este gráfico de densidad y Box plot, la mayor cantidad de alojamientos de cada barrio tiene un precio por noche rondando los casi 20 dólares.

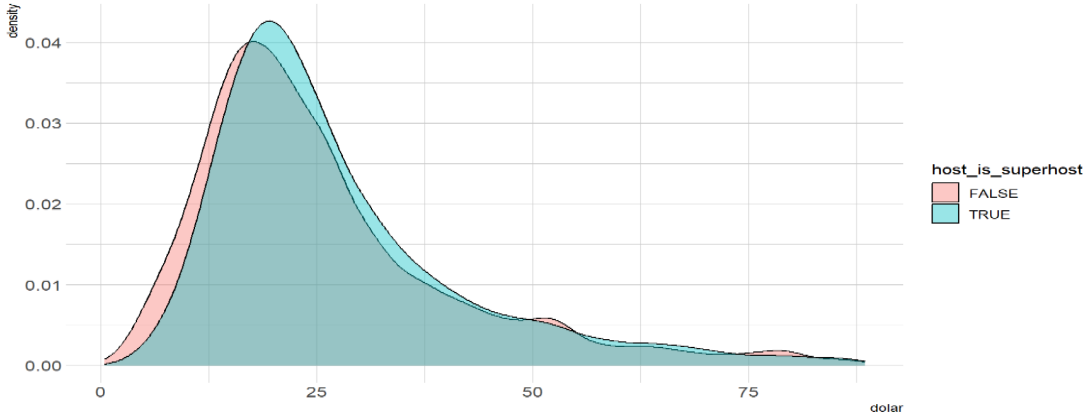


Fig 11. *Gráfico de densidad de dólares por noche por tipo de host (Super Host o no).*

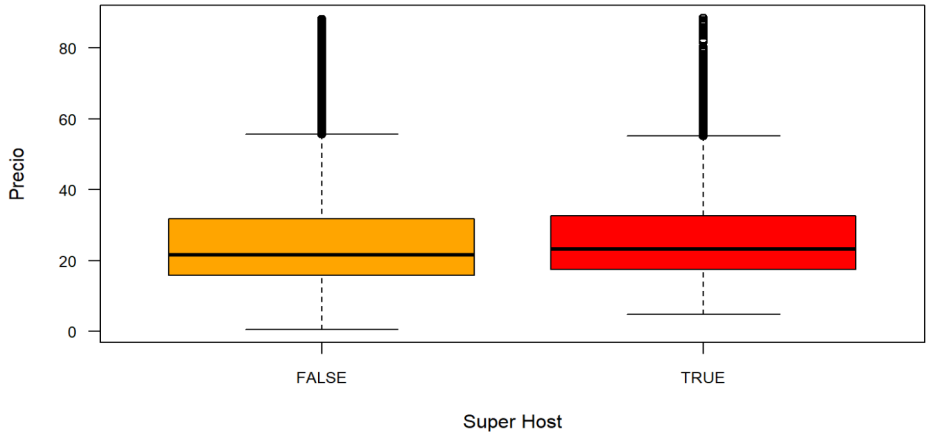


Fig 12. *Box plot de dólares por noche por tipo de host (Super Host o no).*

Continuando con la variable de *host_is_superhost*, se observa en las Figura 10 y 11 cómo los precios de estos dos grupos no varía en gran medida. Pero lo que sí se puede observar es cómo la mayor cantidad de alojamientos con el menor precio pertenecen a aquellos listings donde su anfitrión no es superhost.

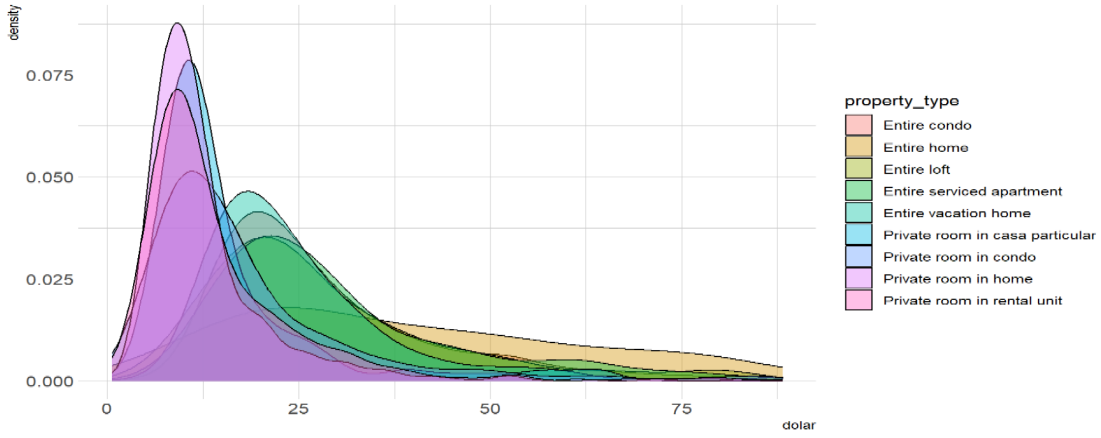


Fig 13. *Gráfico de densidad de dólares por noche por tipo de propiedad (top 10 con más alojamientos).*

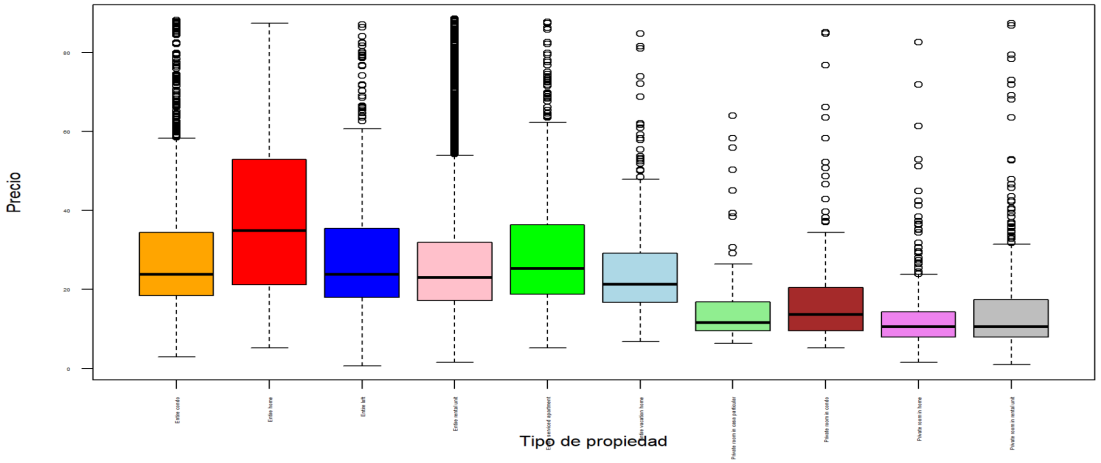


Fig 14. *Box plot de dólares por noche por tipo de propiedad (top 10 con más alojamientos).*

Luego, para *property_type* la densidad de precios de cada uno de estos tipos de alojamientos varía con una mayor notoriedad. Esto se puede deber a la vasta diferencia de alojamientos de un mismo tipo. Así como por ejemplo hay casas con 3 pisos mientras que hay otras de 1.

Por otro lado, como se observa en el Boxplot, la distribución del precio por noche es similar entre Shared y Private room, con una media de alrededor de 10 dólares, mientras que para Entire home/apt el precio promedio se encuentra alrededor de los 20 dólares.

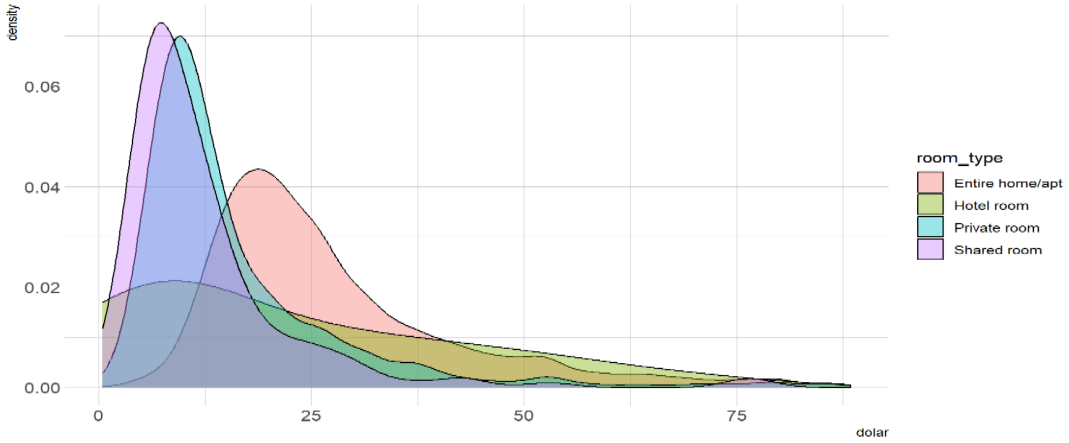


Fig 15. *Gráfico de densidad de dólares por noche por tipo de cuarto.*

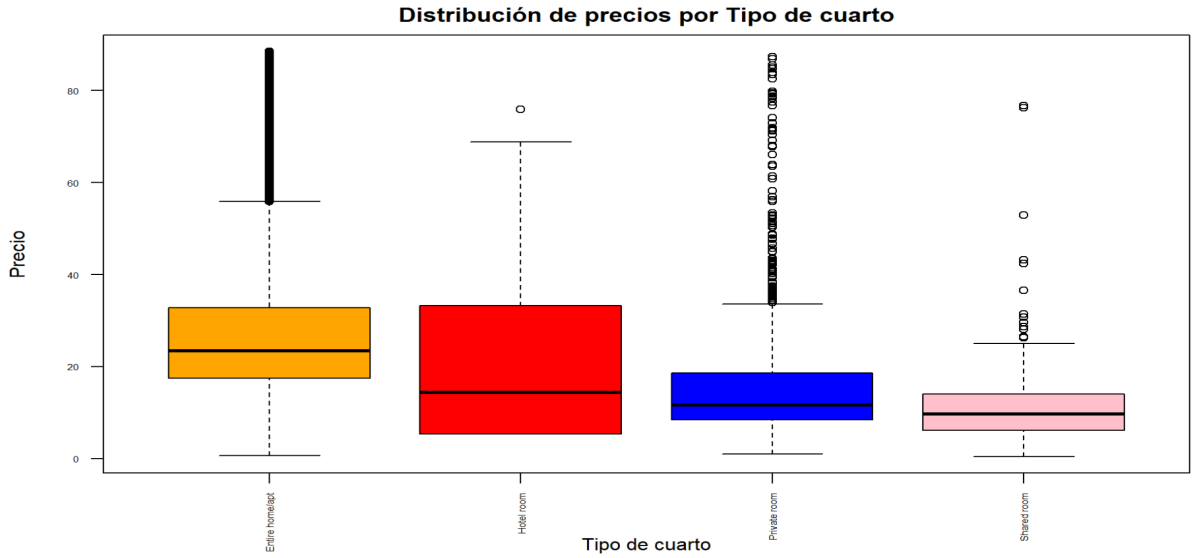


Fig 16. *Box plot de dólares por noche por tipo de cuarto.*

Por último, la variable de *room_type* tiene un comportamiento similar al de *property_type*, ya que cada categoría tiene una conducta diferente. Sin embargo, como se puede observar en la Figura 14, el precio promedio de cada cuarto es casi idéntico.

Viendo los comportamientos de cada uno de estos gráficos se ve cómo, a pesar de estar agrupados por diferentes clases, la mayor cantidad de alojamientos ronda en un precio por noche de casi 20 dólares.

Análisis de valores outliers

Se realizará un análisis de las variables numéricas y los valores atípicos que se pueden encontrar en cada una de ellas. Estos valores pueden afectar significativamente la precisión y la fiabilidad de los resultados de análisis y modelos estadísticos, predictivos y de aprendizaje automático, por lo que comprender su comportamiento y correcto tratamiento en cada caso particular es importante para evitar llegar a predicciones erróneas y a decisiones incorrectas.
En primer lugar se analizó la **variable *precio***, que es fundamental para los fines predictivos del trabajo. Observando los estadísticos y el Box plot para la variable como se puede ver en la Tabla 4 y la Figura 15, se identifican valores atípicos. En particular hay un departamento para 3 personas cuyo precio por noche es de 20 millones 500 mil pesos. Se analizó la posibilidad de que los valores estén mal cargados en nuestra base de datos, pero son los mismos precios que figuran en la plataforma de Airbnb.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
175	6390	8974	15471	13453	20500432

Tabla 4. *Estadísticas precio*

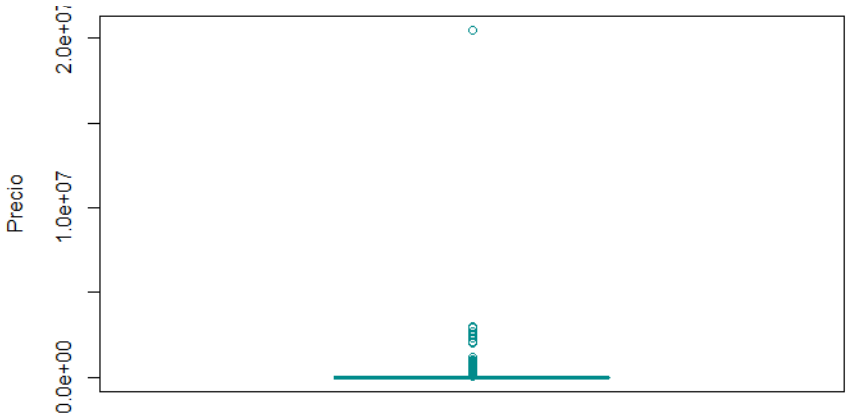


Fig 17. *Box plot de precio.*

Para continuar con el tratamiento de los outliers se realizó un análisis del precio por barrio y por tipo de propiedad. En la Figura 18 se pueden observar una gráfica acotada en el eje x con los boxplots para los 10 barrios con más alojamientos (el 82% de las propiedades de caba se encuentran en estos barrios). En los barrios de Palermo, Recoleta y Belgrano las propiedades en general tienen precios más altos. Por otra parte en los barrios de Palermo, Recoleta y San Telmo, es donde se encuentran los valores atípicos más elevados.

En el caso del tipo de habitación, como era de esperarse, los cuartos de hotel y las casas o departamentos enteros son los más caros, y estos últimos junto con los cuartos privados son los que presentan mayor distribución en sus precios. Esto se puede observar en la Figura 19, acotada en el eje x.

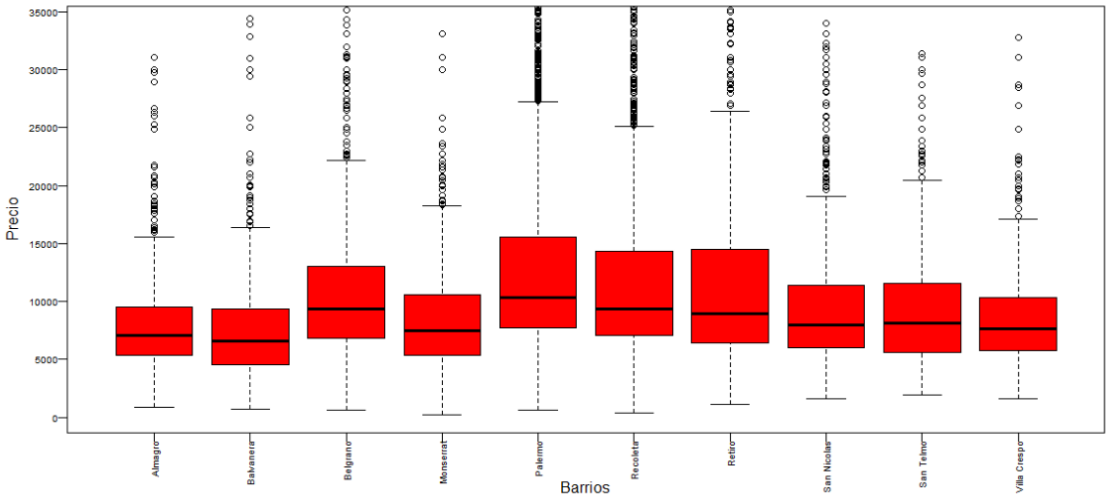


Fig 18. *Box plot de precio por barrio.*

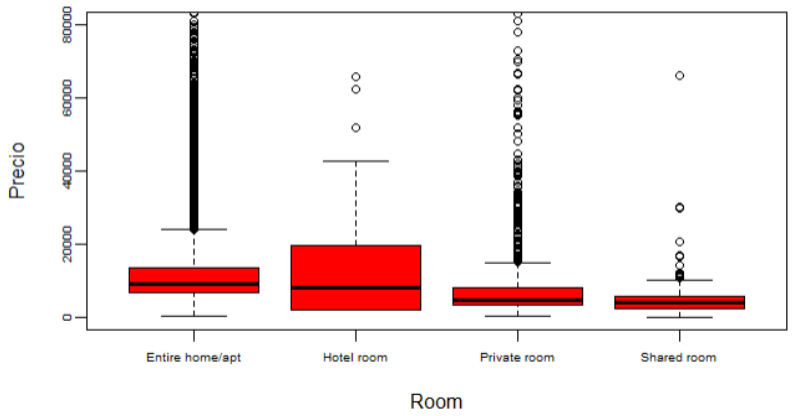


Fig 19. *Box plot de precio por tipo de cuarto.*

Del total de las 22.677 propiedades hay 882, es decir un 3.8%, cuyo precio es mayor a $34.642 *(tercer cuartil + 3 * rango intercuartílico)* considerándose valor extremo. Se realizó un análisis para observar la proporción de estos valores que se encuentran en cada barrio al igual que por tipo de propiedad. En Belgrano por ejemplo hay 36 de las 882 propiedades, que parece mucho si se quisieran eliminar los registros, pero en proporción con el total de propiedades que hay en el Belgrano estás 36 representan solo un 3%. Para el resto de los barrios esta proporción no supera el 5% (observar tabla en Anexo 2). En cuanto a los tipos de propiedad sucede lo mismo; para "Private room", "Shared room" y "Entire home/apt" estos valores extremos representan menos de un 4.5% en cada caso. Bajo estos fundamentos, se decidió eliminar los 882 registros de la base de datos. A continuación se observan los estadísticos del precio una vez eliminados los valores extremos:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
175	6241	8697	10322	12425	34611

Tabla 5. *Estadísticas precio sin valores atípicos.*

Para la **variable *maximum_nights***, se encontraron solo 3 propiedades con valores atípicos. Luego de analizar el comportamiento de la variable para el resto de las propiedades, se decidió reemplazar estos valores (1825 y 99.999) por 1.125 noches que es la cantidad de noches máxima que alcanza el resto de las propiedades.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.0	90.0	365.0	526.4	1125.0	1125.0

Tabla 6. *Estadísticas maximum_nights.*

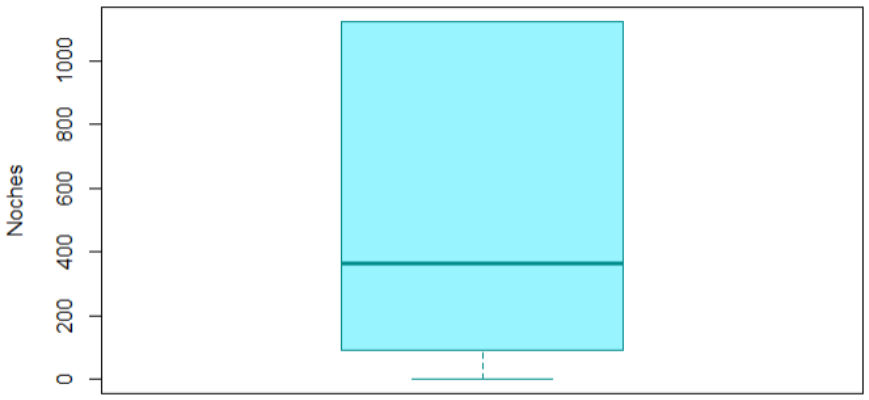


Fig 20. *Box plot de maximum_nights.*

Se observan 7 propiedades con un máximo de noches igual a 1. Esto lleva a crear una nueva columna llamada "dif" que contiene para cada registro la diferencia entre *maximum_nights* y *minimum_nights*. Se observó que hay 61 propiedades para las cuales la diferencia es 0; es decir que el mínimo y máximo de noches que se puede alquilar una propiedad es la misma. Esto sólo podría tener sentido para 4 registros cuyos valores eran 1 para ambas variables ya que se trata de casos particulares como pasar una noche en una carpa compartida en un camping. Luego, se pudo reemplazar el mínimo o máximo de 10 propiedades observando el comportamiento de estas variables en propiedades similares del mismo host si es que este tenía más de una. Para las restantes 47 propiedades se reemplazó el mínimo de noches por la mediana de *minimum_nights* igual a 3 o por el máximo de noches por la mediana de *maximum_nights* igual a 365.

En cuanto a la **variable *minimum_nights*** se encontraron 1.657 valores atípicos extremos, que superan las 14 noches, pero luego de verificar que esto tuviera sentido en relación con la cantidad máxima de noches que se puede alquilar la propiedad, se decidió no modificar estos valores por el momento ya que se debe a alojamientos únicamente de larga estadía que representan un porcentaje mucho menor que los alojamientos que permiten estadías cortas. En el boxplot en la Figura 21 se puede observar la distribución de la variable cortando del gráfico los valores extremos.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	2.000	3.000	6.445	5.000	1000.000

Tabla 7. *Estadísticas minimum_nights.*

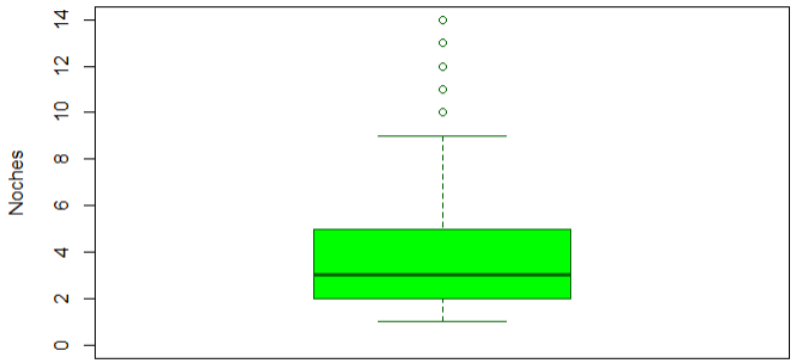


Fig 21. *Box plot de minimum_nights.*

En la Tabla 8 y la Figura 22 a continuación se presentan los estadísticos y los Box plots para las **variables *bedrooms*, *bathrooms* y *accommodates*.** Tanto para la cantidad de cuartos como de baños por propiedad, vemos un promedio de 1,2 y una mediana de 1. Para la cantidad de huéspedes, el promedio es de 2,7 y la mediana de 2. Sin embargo, para las 3 variables se observan valores atípicos altos. En cuanto a la cantidad de habitaciones y de baños, luego de realizar un análisis por tipo de habitación (Anexos 3, 4 y 5) y de la descripción de la propiedad observamos que los valores más extremos como es el caso de 35 cuartos y 22 baños se deben a que hay algunos registros que corresponden a habitaciones de un mismo host que se encuentran en edificios, bed and breakfast, o hostels con múltiples opciones para alquilar dentro de los mismos. Por lo que en muchos casos se cuenta la cantidad de cuartos privados ("Private rooms" y algunos "Hotel rooms") y baños totales del establecimiento. El resto de los valores extremos de estas variables corresponden a alquileres de casas o departamentos enteros, lo cual tiene sentido. En cuanto a los accommodates sucede lo mismo; o se debe al total que entra en el bed and breakfast, por ejemplo, o, a una casa grande como es el caso en el que se da el máximo de huéspedes de la base que es igual a 16.

	mean	mediana	min	25%	50%	75%	max
bedrooms	1.269	1	1	1	1	1	35
bathrooms	1.220	1	0	1	1	1	22
accommodates	2.772	2	1	2	2	4	16

Tabla 8. *Estadísticas de bedrooms, bathrooms y accommodates.*

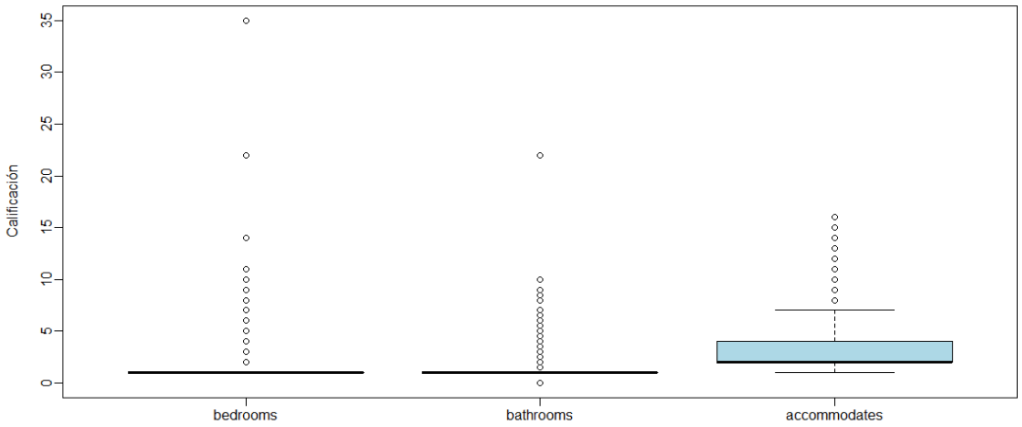


Fig 22. *Box plots de bedrooms, bathrooms y accommodates.*

Además se realizó un análisis de correlación entre la cantidad de cuartos y de baños, y entre la cantidad de cuartos y de accommodates. En ambos casos se indicó un valor de correlación de alrededor de 0,6, lo que indica que existe una relación positiva entre las variables, es decir, que a medida que la cantidad de cuartos aumenta, también lo hace la cantidad de baños y de huéspedes permitidos.

Luego de este análisis, se decidió no imputar estos valores ya que no se consideran incorrectos y su imputación implicaría realizar muchas suposiciones sin una fuerte fundamentación. Sin embargo, se considerará para lo que sigue del trabajo, trabajar únicamente con las propiedades de la misma categoría que las de Marcelo (departamentos enteros y casas).

Por otra parte se analizó la distribución para las **variables relacionadas con los puntajes recibidos** en las reseñas de las propiedades.

	mean	mediana	sd	min	25%	50%	75%	max
review_scores_rating	4.769	4.86	0.438	0	4.74	4.86	5.00	5
review_scores_accuracy	4.829	4.91	0.321	1	4.80	4.91	5.00	5
review_scores_cleanliness	4.739	4.83	0.369	1	4.67	4.83	4.96	5
review_scores_checkin	4.882	4.96	0.288	1	4.88	4.96	5.00	5
review_scores_communication	4.878	4.96	0.296	1	4.88	4.96	5.00	5
review_scores_location	4.882	4.96	0.250	1	4.86	4.96	5.00	5
review_scores_value	4.732	4.81	0.362	1	4.67	4.81	4.93	5

Tabla 9. *Estadísticas de review_scores_*.*

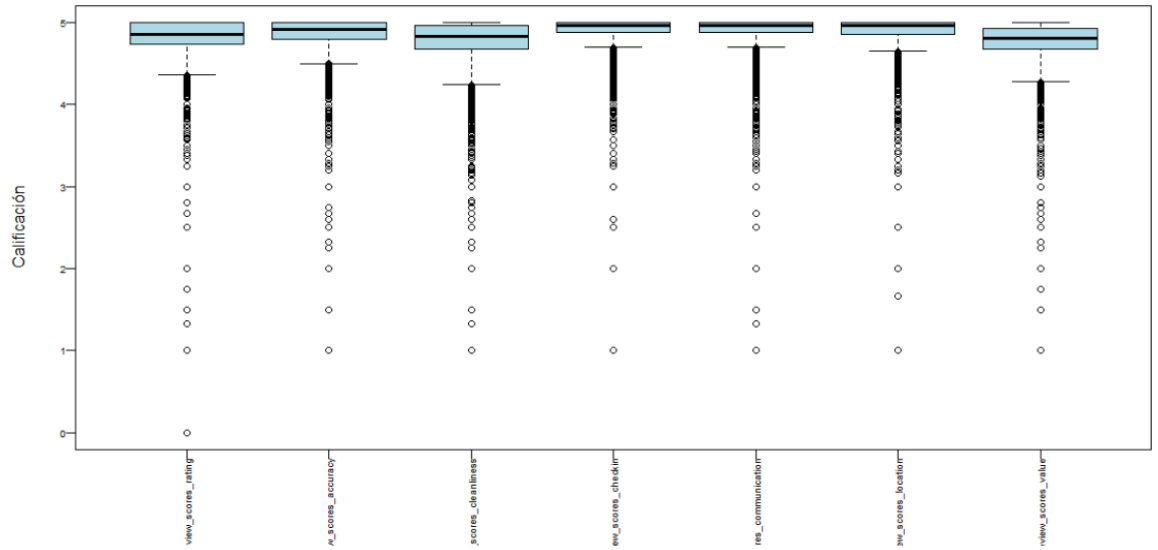


Fig 23. *Box plots de review_scores_*.*

En general el puntaje promedio recibido para cada categoría (limpieza, ubicación, etc.) es de alrededor de 4,8. Para *reviews_scores_rating* que pondera el puntaje general recibido para la propiedad en las distintas categorías, sólo el 5% de las propiedades enlistadas tienen una calificación menor a 4,33 (lo que se considera valor atípico). Analizando las propiedades de este subgrupo se encuentra que el valor promedio de reseñas recibidas es de 7, mientras que la mediana es de 3, ya que son propiedades con baja ocupación; el promedio de días en que las propiedades están disponibles en el próximo año es de 231 días. Los estadísticos generales de estas variables se pueden ver en el Anexo 6.

Análisis de correlación

En cuanto al análisis de correlación, se consideraron todas las variables numéricas. El análisis se realiza para entender su comportamiento y cómo se influyen unas a las otras. Ayuda a identificar interdependencia y multicolinealidad en los datos, lo que es útil para entender el comportamiento futuro de las variables, y tomar decisiones informadas basadas en los resultados del análisis. También ayuda a identificar variables irrelevantes o redundantes, por lo cual servirá para simplificar el dataset y mejorar la eficacia de los modelos.

A continuación se muestra la matriz de correlación de las variables numéricas del dataset. Se observa que la variable *price* tiene mayor correlación positiva con las variables *bedrooms* y *accommodates*. En el Anexo 7 se puede observar una tabla con los valores de la correlación del precio con cada una de las variables, y en el Anexo 8 se muestra una matriz de correlación más acotada en la que como resultado de esta primera matriz de correlación no demostraron una correlación significativa con el precio.

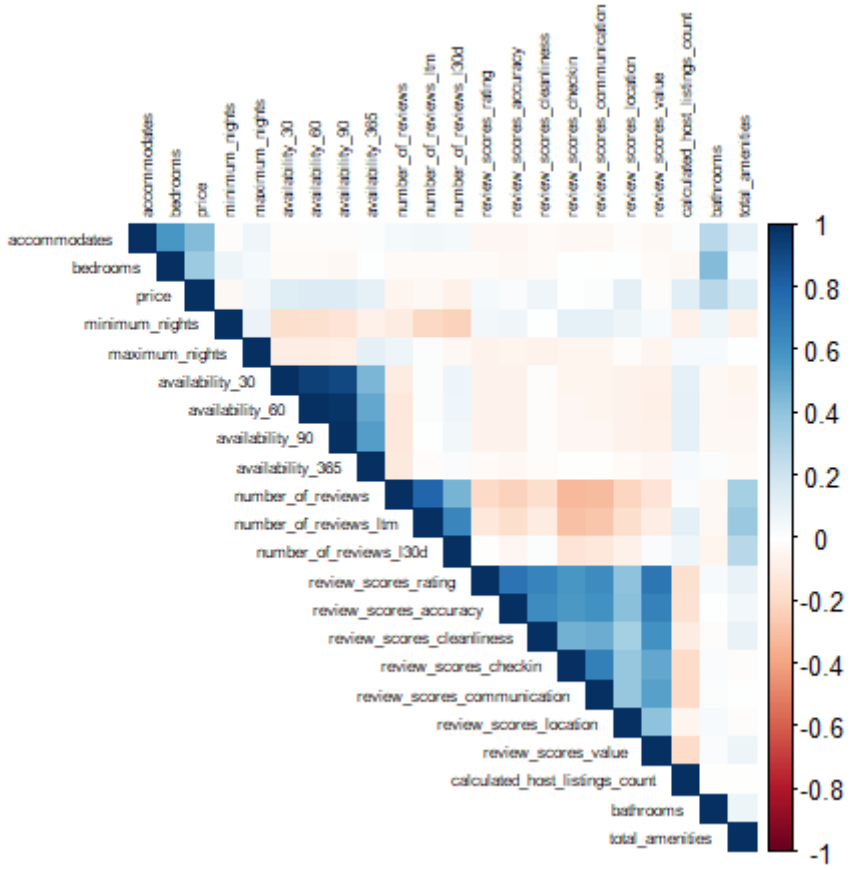


Fig 24. *Matriz de correlación de variables numéricas.*

Calendar

El dataset de **calendar** cuenta con **13.344.091 observaciones** y **11 variables**, resumidas en la Tabla 10 que se presenta a continuación:

	listing_id	date	available	price	adjusted_price	minimum_nights	maximum_nights	price_num	adjusted_price_num	dolar_price	dolar_adjusted_price
1	130424	2023-03-30	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102
2	130424	2023-03-31	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102
3	130424	2023-04-01	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102
4	130424	2023-04-02	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102
5	130424	2023-04-03	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102
6	130424	2023-04-04	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102
7	130424	2023-04-05	FALSE	$10,401.00	$10,401.00	7	90	10401	10401	26.60102	26.60102

Tabla 10. *Dataset* ***calendar****.*

El dataset presentado resulta de una transformación de 4 archivos encontrados en el sitio web de Inside Airbnb, los cuales fueron combinados para obtener una versión completa y actualizada de las reservas “a un año” de los alojamientos de Airbnb, desde junio de 2022 hasta junio de 2023. Cada uno

de esos 4 archivos registraba las reservas realizadas "a un año" en cada alojamiento por fecha, especificando el precio por noche y la cantidad mínima y máxima de noches que un huésped estaba habilitado a quedarse.

Para facilitar el posterior análisis de datos, se modificó el tipo de dato de las variables *listing_id*, *date* y *available*, y se construyeron nuevas variables a partir de las originales. A continuación se describen las variables (aquellas resaltadas en amarillo corresponden a las construidas):

Variable	Descripción	Tipo de dato
listing_id	ID del alojamiento.	Numeric
date	Fecha del calendario.	Date
available	Indica si el alojamiento estaba o no disponible el día en que se scrapearon los datos de Airbnb.	Boolean
price	Precio del alojamiento por noche registrado para la fecha (**date**) el día en que se scrapearon los datos de Airbnb.	Character
adjusted_price		Character
minimum_nights	El mínimo de noches habilitado para una reserva realizada en la fecha (**date**).	Integer
maximum_nights	El mínimo de noches habilitado para una reserva realizada en la fecha (**date**).	Integer
price_num	El valor de **price** en tipo de dato num.	Numeric
adjusted_price_num	El valor de **adjusted_price** en tipo de dato num.	Numeric
dolar_price	El valor de **price** en dólares, al cambio del dólar blue del día en que se scrapearon los datos de Airbnb.	Numeric
dolar_adjusted_price	El valor de **adjusted_price** en dólares, al cambio del dólar blue del día en que se scrapearon los datos de Airbnb. Los valores del dólar blue se recuperaron del registro del dólar informal histórico de Ámbito.	Numeric

Tabla 11. *Descripción de variables de* ***calendar****.*

Análisis de valores faltantes

La presencia de valores faltantes en el dataset se evaluó inicialmente en cada uno de los 4 conjuntos de datos que componen **calendar,** por separado. Se encontraron valores faltantes en 2 conjuntos de datos: en uno de ellos en las variables de *minimum_nights* y *maximum_nights* y en el otro en *price* y *adjusted_price* (directamente reflejados en *price_num, adjusted_price_num*).

Para las variables *minimum_nights* y *maximum_nights* se encontraron **6 observaciones** con valores faltantes, para un total de **6 alojamientos**. Se consideró que una imputación atinada sería reemplazar los valores faltantes por la **media de *minimum_nights* y *maximum_nights* de cada listing correspondiente**. Es decir, si determinado alojamiento tiene establecido un valor de *minimum_nights*

de 3 noches en promedio para el resto de las fechas, se tomará dicho valor para reemplazar las fechas en dónde no hay registro de mínimo de noches.

Para las variables *price* y *adjusted_price* se encontraron **184 observaciones** con valores faltantes, para un total de **2 alojamientos**. Si bien se intentó seguir el mismo criterio de imputación que para *minimum_nights* y *maximum_nights*, se encontró que no había registro de precios para ninguna otra fecha de esos 2 alojamientos. Por ende, la mejor alternativa de imputación fue reemplazarlos por el precio por noche registrado en el dataset de **listings**.

Análisis de valores outliers

La demanda de alojamientos se estudiará a partir de la *availability* registrada en cada fecha, pero además se relacionará con el precio por noche, por fecha. Es por esto que es importante revisar los valores de las variables de precio por noche de los alojamientos. Los anfitriones de Airbnb fijan un precio por noche para todas las fechas futuras y en cualquier momento pueden actualizarlo, por ejemplo si se aproxima la alta temporada o hay un evento importante en la zona.

La variable a través de la cual se va a analizar el precio de alquiler de los alojamientos es *dolar_price*, que tiene el valor más representativo de los alquileres a lo largo del tiempo. Previo a revisar los valores atípicos de *dolar_price* en **calendar**, se excluyeron registros correspondientes a alojamientos que se excluyeron en el análisis del dataset de **listings** por ser valores extremos en la variable de *price*.

summary de *dolar_price*	
Min	0,45 dólares
1st. Q	16,42 dólares
Median	23,83 dólares
Mean	43,49 dólares
3rd Q	35,71 dólares
Max	54.464 dólares

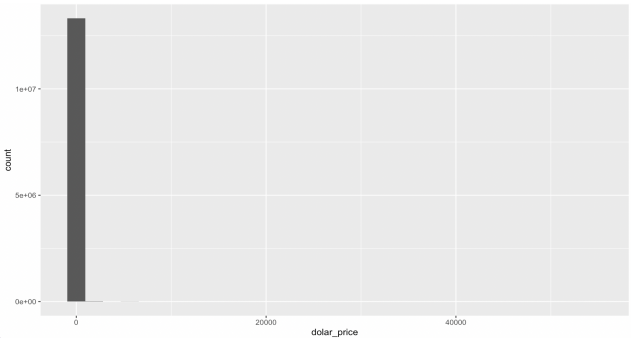


Fig 25. *Estadísticos descriptivos e histograma de dolar_price.*

Los estadísticos de la variable evidencian la presencia de valores atípicos aún habiendo excluido los alojamientos con valores extremos mencionados. Como se puede observar, hay observaciones con precios por noche de hasta 54.464 dólares y, en contraste de 0,45 dólares por noche, lo cual no es representativo para la mayoría de los alojamientos.

Al analizar la distribución de *dolar_price* en mayor profundidad se descubrió que:

a) Solo el **1% de las observaciones de calendar** tienen precios mayores a 251 dólares por noche. Esto corresponde a 2,5% de los alojamientos registrados en **calendar**.
b) Sólo el **5% de las observaciones de calendar** tienen precios menores a 9 dólares por noche. Esto corresponde al 8,9% de los alojamientos registrados en **calendar**.

Si bien hay una gran diferencia de precio entre alojamientos que cobran 9 dólares la noche y alojamientos que cobran 251 dólares, características propias de las propiedades y épocas del año pueden hacer que los precios sean más o menos altos. Es por esto que se decidió eliminar de la base la

información sobre todos los alojamientos que estén registrados con precios menores a 9 dólares y mayores a 251, eliminando un **11,4%** de los alojamientos de **calendar**.

Realizando está limpieza, se observa de mejor manera la distribución de *dolar_price* en el siguiente histograma:

summary de *dolar_price*	
Min	9 dólares
1st Q	18 dólares
Median	24 dólares
Mean	32 dólares
3rd Q	37 dólares
Max	250 dólares

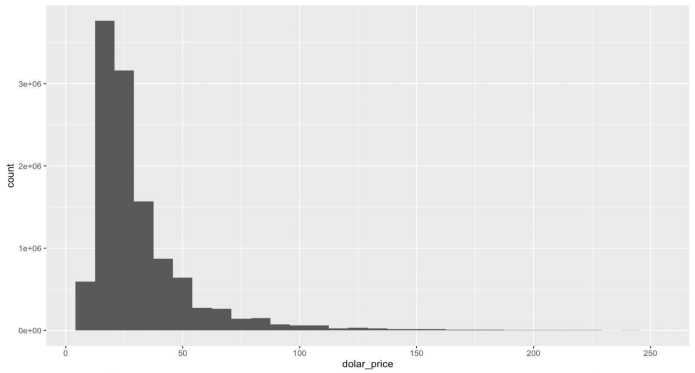


Fig 26. *Estadísticos descriptivos e histograma de dolar_price luego de la eliminación de valores atípicos.*

Series de tiempo

Como se describió anteriormente, para el desarrollo del modelo predictivo de cantidad de días de ocupación en una semana es necesario entender cómo varía la demanda de los alojamientos a lo largo del tiempo. Es por esto que con los datos obtenidos de **calendar** se realizaron gráficos de series de tiempo utilizando la variable de *availability* para estimar el porcentaje de ocupación de las propiedades.

Es importante hacer énfasis en que este análisis se basa en las reservas efectuadas para hacer una estimación de la ocupación, sin considerar si se efectivizó dicha ocupación. En esta línea, la nueva variable de *porcentaje de ocupación* se calcula como la suma de alojamientos reservados por fecha sobre el total de alojamientos registrados en esa fecha. El resultado de graficar el promedio del *porcentaje de ocupación* **por mes** se muestra a continuación:

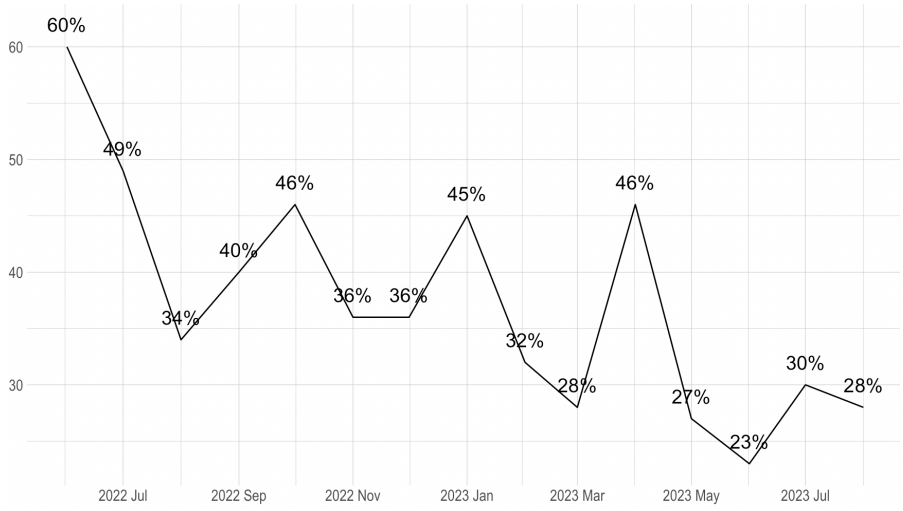


Fig 27. *Porcentaje de ocupación de alojamientos por mes.*

Si bien este gráfico muestra altas y bajas en la ocupación de los alojamientos registrados en **calendar**, se dificulta encontrar un patrón específico. A fin de comprender de mejor manera las tendencias en la ocupación se realizó el análisis de ocupación **por semana**, ya que tal vez existía algún patrón escondido en ese nivel de detalle. A continuación se muestran 3 gráficos que muestran el porcentaje de ocupación semanal por barrio (para los 3 barrios con más alojamientos: Palermo, Recoleta y San Nicolás):

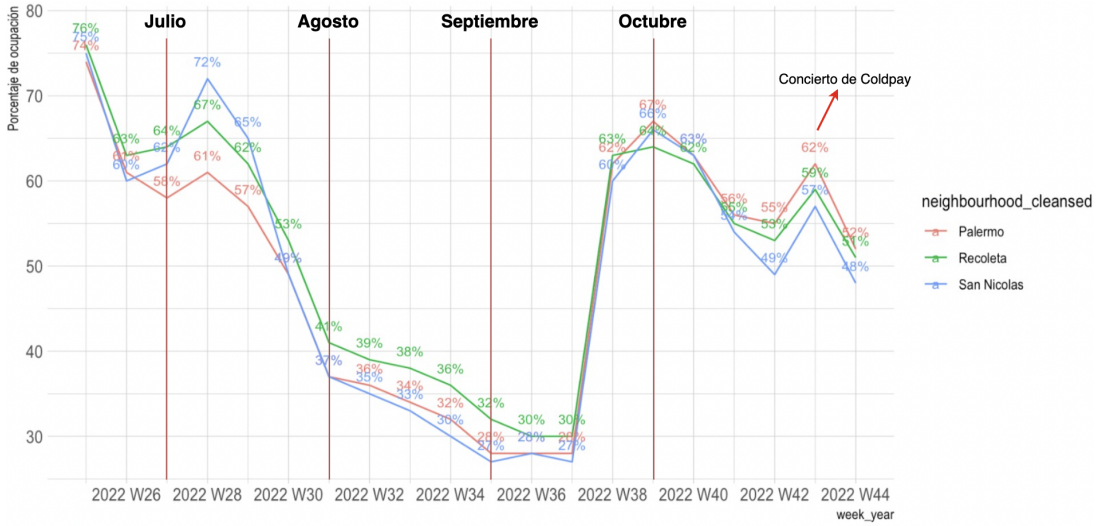


Fig 28. *Porcentaje de ocupación de alojamientos por barrio y semana* ***junio 2022 a octubre 2022***

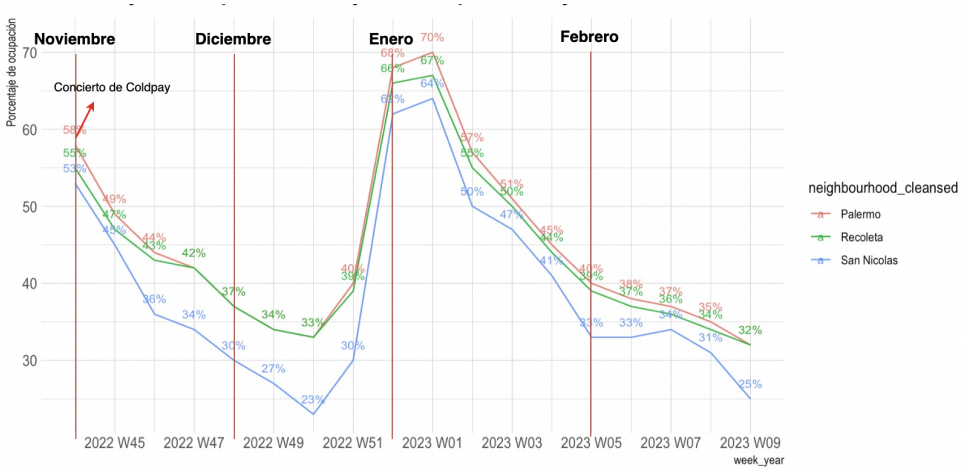


Fig 29. *Porcentaje de ocupación de alojamientos por barrio y semana* ***noviembre 2022 a febrero 2023***

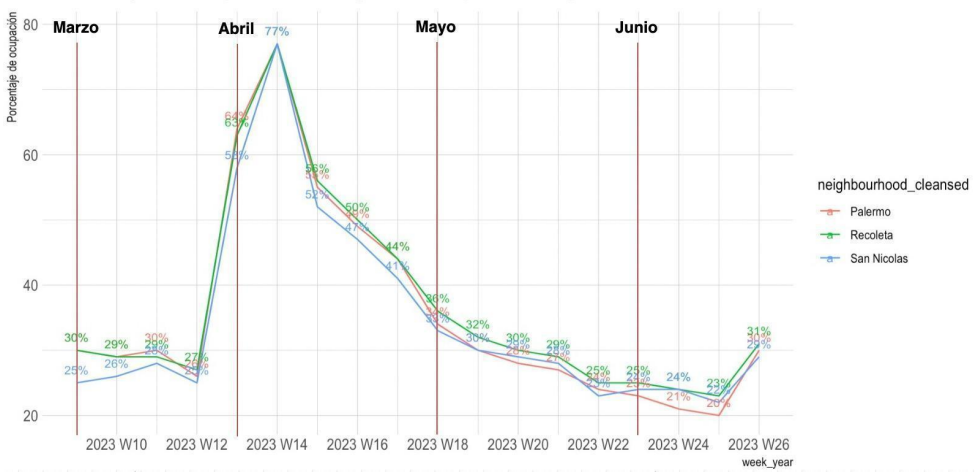


Fig 30. *Porcentaje de ocupación de alojamientos por barrio y semana* ***marzo 2023 a junio 2023***

Los gráficos de promedio de *porcentaje de ocupación* desagregados por barrio y semana permiten observar determinados comportamientos en la demanda:

- La ocupación de los alojamientos parece tener un comportamiento similar por barrio, al menos para Palermo, Recoleta y San Nicolás.
- Los meses en que hubo más ocupación fueron:

    - **Julio de 2022:** tiene un pico de ocupación promedio de **67%** en la segunda semana, lo que es lógico considerando que muchas personas se toman vacaciones en esas fechas. A partir de ese pico la ocupación decrece 28 puntos porcentuales, llegando a una ocupación promedio de **39%** en la primera semana de agosto.

    - **Octubre de 2022:** si bien septiembre comienza con una ocupación promedio de **29%**, la demanda crece abruptamente para su anteúltima semana, llegando a octubre con una ocupación promedio de **66%**. ¿Estará este crecimiento relacionado al feriado del 7 de octubre destinado a fines turísticos? Semanas más tarde también puede observarse un pico de ocupación, que puede relacionarse a los conciertos de Coldplay que cayeron en fines de semana, lo que pudo llevar a que mucha gente viaje a CABA para asistir.

    - **Enero de 2023:** la ocupación venía en caída desde noviembre, hasta que en la primera semana de enero se remontó. Las dos primeras semanas de enero son destacables, logrando un promedio de ocupación de **65%** y **67%**, respectivamente.

    - **Abril de 2023:** la segunda semana de abril destaca con un promedio de ocupación de **77%**. Este pico repentino puede estar relacionado a la Semana Santa, que fue del domingo 2 de abril al sábado 8 de abril.

Con este análisis podemos concluir que la demanda se ve influenciada por eventos externos, feriados y fechas festivas en la ocupación de los alojamientos. No se puede confirmar una estacionalidad dado que sólo hay datos de 1 año.

Relación de precio con ocupación

A fin de ver cómo se relaciona la variable de *dolar_price* con la ocupación se observó la distribución de los precios por semana, para 3 rangos de ocupación (**baja** si el alojamiento estuvo 2 o menos días ocupado, **media** si el alojamiento estuvo ocupado entre 3 y 5 días y **alta** si el alojamiento estuvo ocupado 6 o 7 días.

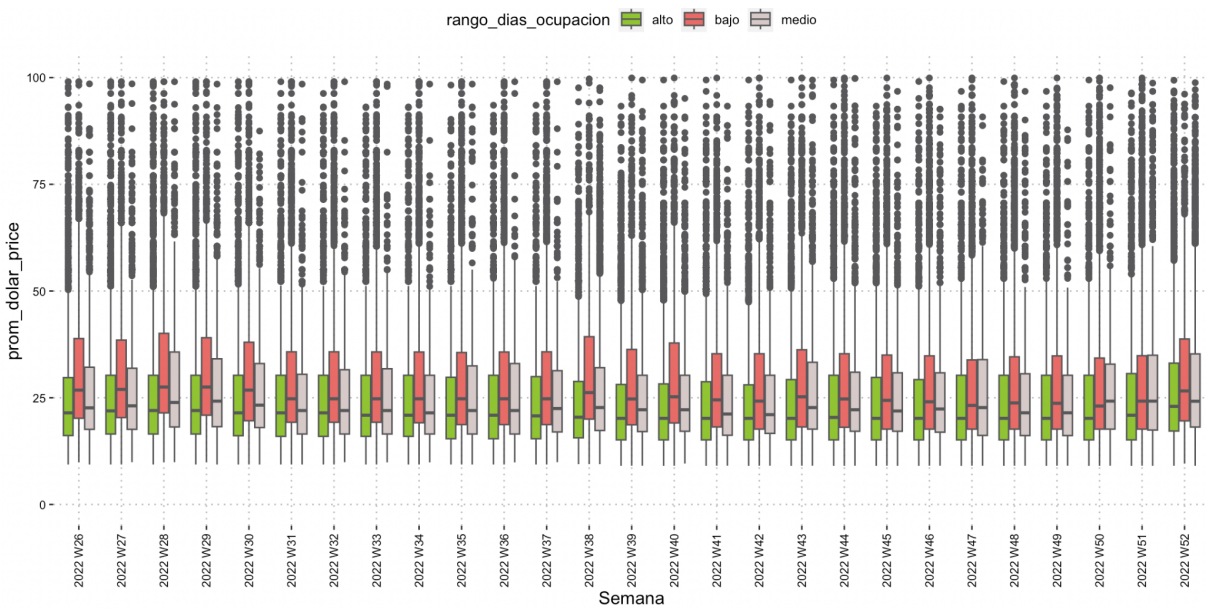


Fig 31. *Distribución de dólares por noche por semana.*

El gráfico presentado confirma la relación esperada entre precio y ocupación: que a menores precios tiene sentido una mayor ocupación de los alojamientos.

Hipótesis

- Más alto el precio por noche, menos reservas realizadas en el alojamiento por año.
- Los apartamentos y casas son los alojamientos que más demanda tienen.
- Los potenciales huéspedes prefieren alojamientos situados en el barrio de Palermo.
- Hay más potenciales reservas en la época donde hay eventos transcurriendo en la ciudad y en la provincia de Buenos Aires.
- Un precio competitivo resultará en más reservas y mayor rentabilidad.

Enfoque de Solución

Ya teniendo la información limpia y lista para analizar, el próximo paso a seguir es la experimentación y el desarrollo de la solución. La herramienta a desarrollar en el presente trabajo requiere entender la incidencia de un conjunto complejo de variables en la ocupación de alojamientos de Airbnb. Existen muchas variables que repercuten sobre la ocupación de un alojamiento de Airbnb; la época del año, el precio por noche, las características del departamento, barrio donde se encuentra, entre otras.

En esta sección, se buscará encontrar un modelo que determine, para una semana del año en particular, la cantidad de días que un alojamiento va a estar alquilado a partir de un conjunto de

variables. Para poder realizar esto, se investigaron modelos de machine learning, tanto de clasificación como de regresión, para poder así predecir los días de ocupación con mayor exactitud. Esto se hará a partir de una partición de la base en dos: entrenamiento y testeo. Una vez seleccionado el modelo, se desarrollará una herramienta que lo integre y pueda ser utilizada por un anfitrión de Airbnb, un usuario como Marcelo. Para llegar al producto final, la herramienta, seguirá una serie de pasos:

1. ***Diseño de la base de datos con la que el modelo predictivo será entrenado***. Como se observó en secciones anteriores, la base de datos **calendar** cuenta con información *diaria* sobre la ocupación de los alojamientos. Debido a que la herramienta a construir hará predicciones semanales se debe rediseñar la base de datos.
2. ***Selección de variables y feature engineering.*** En línea con lo estudiado en el EDA, se tomarán y construirán las variables más influyentes en la ocupación semanal de los alojamientos.
3. ***Desarrollo del modelo.*** Se investigarán modelos de regresión para predecir el porcentaje de ocupación semanal. Esta fase comprende el tuneo de hiper parámetros correspondiente a cada modelo en particular.
4. ***Integración a la herramienta.*** Como se mencionó anteriormente, la herramienta consiste de un modelo predictivo. Este se encuentra en una página web donde el usuario ingresará diversos parámetros como el precio por noche que desea aplicar, la semana del año que desea analizar, el mes, el tipo de estadía (corta, media1, media2 y larga) en la que un huésped se puede alojar, cuántos huéspedes se pueden quedar, el puntaje promedio del alojamiento y cuál es el porcentaje de ocupación que tuvo el alojamiento el año anterior.

Metodologías a implementar

Debido a que las variables a utilizar en estos modelos predictivos de machine learning serán tanto numéricas como categóricas, las metodologías seleccionadas para el aprendizaje supervisado deben adaptarse a este requerimiento.

Antes de emplear los algoritmos predictivos, se decidió aplicar el método de agrupamiento de K-Medias para clasificar los 44 barrios incluidos en las bases de datos. Incluir la totalidad de estos barrios variables de input al modelo predictivo podría no ser la mejor alternativa, ya que probablemente haya barrios que se comporten de forma similar en cuanto a la oferta y demanda de alquileres de Airbnb. Es por esto que se decidió clusterizar para resumir la información en grupos de barrios y consolidar la información relevante.

El algoritmo empleado agrupa los datos en k clusters, donde k es un número elegido por el usuario, y busca una partición óptima de los datos para minimizar la distancia euclidiana entre cada punto y el centroide del cluster al que pertenece. Así, cada alojamiento es asignado al cluster cuyo valor medio es más cercano. Este enfoque fue elegido para clasificar los barrios en grupos con características similares.

Para hacer el clustering se utilizaron las variables: precio por noche por persona, desviación estándar del precio por noche por persona y porcentaje de ocupación del alojamiento en el último año. Estas fueron seleccionadas en base a su relevancia para la segmentación, ya que busca enfocarse únicamente en los aspectos clave y evitar introducir ruido o redundancia en esta primera instancia del análisis. Esto permite una exploración inicial de los patrones y segmentos más relevantes, para luego complementar este análisis al utilizar modelos predictivos en los que se puedan incorporar más

variables, brindando al usuario la posibilidad de elegir diferentes características al realizar predicciones personalizadas.

El número de clusters, que en este caso es 5, fue establecido utilizando el método Silhouette que permite encontrar el valor *k* óptimo. Una vez habiendo clasificado cada uno de estos alojamientos en uno de estos 5 grupos, se estudió el comportamiento de los mismos.

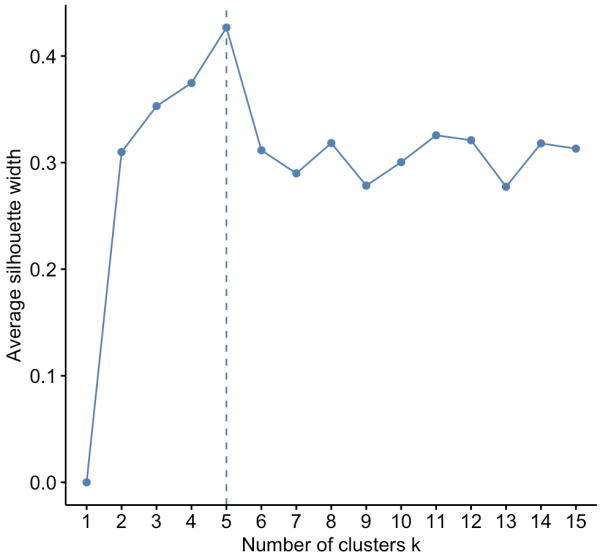


Fig 32. *Método silhouette para encontrar el k óptimo.*

Como se puede observar en el *Anexo 10,* en el que se detalla a qué cluster pertenece cada barrio, el cluster 1 está conformado únicamente por el barrio de Puerto Madero. Esto era de esperarse ya que este es uno de los barrios con mayor poder adquisitivo de Buenos Aires, por lo que tiene sentido que se diferenciara en mayor medida del resto. Por otro lado, los barrios más turísticos y habituados de la ciudad confirman el cluster 2: Belgrano, Palermo, Flores, Nuñez, Constitución y Villa Urquiza, entre otros.

cluster	precio_por_persona	precio_por_persona_sd	porcentaje
<int>	<dbl>	<dbl>	<dbl>
1	20.3	10.1	43.0
2	10.5	5.86	45.1
3	8.16	4.30	38.8
4	8.70	1.50	17.2
5	8.20	5.23	58.2

Tabla 12. *Variables de cada cluster.*

Insights:

- Cluster 1: conformado por Puerto Madero, presenta el precio por persona y la variabilidad del mismo más altos.
- Cluster 2: en este cluster se encuentran barrios cuyo precio por persona y porcentaje de ocupación en el último año tienen valores medios. Aproximadamente el 93% de las propiedades de todo CABA se encuentra en este cluster, incluyendo los cuatro alojamientos de Marcelo.

- Cluster 3: este clusters presenta un precio por persona bajo y un porcentaje de ocupación medio.
- Cluster 4: al igual que el anterior en este grupo de barrios el precio se mantiene bajo, pero además cuenta con la menor variabilidad del mismo y el menor porcentaje de ocupación, siendo este de 17% en el último año
- Cluster 5: este cluster está compuesto por los barrios que mayor porcentaje de ocupación tuvieron y con un precio promedio por persona bajo.

Si bien inicialmente se pensó el clustering como una manera de generar nuevas variables para el modelo, ya que el cluster 2 contiene el 93% de los datos se decidió que en lo que resta del proyecto se trabajará únicamente con las propiedades de este grupo, descartando el 7% restante.

Modelos predictivos:
Se trabajarán con cuatro modelos de machine learning para predecir la cantidad de días de una semana en que un alojamiento de Airbnb estará ocupado.

➢ *Regresión lineal*: es una herramienta que permite estudiar la relación entre dos o más variables, y predecir el valor de una a partir de las otras. En este caso la relación sería entre una variable de respuesta (por ejemplo, el número de días ocupados en una semana) y varias variables predictoras (por ejemplo, el precio por noche, la ubicación, las características de la propiedad, etc.)

➢ *Random forest*: Los bosques aleatorios son una técnica de modelado predictivo que consiste en la construcción de múltiples árboles de decisión, los cuales se ajustan de manera paralela y se combinan para generar una predicción final más precisa y estable que la de un solo árbol. Este modelo reduce el sesgo y la varianza en el modelo, lo que conduce a una mejor generalización y capacidad de predicción en datos nuevos.

➢ XGBoost: Este método es iterativo y se basa en dar más importancia a las observaciones que son difíciles de clasificar. Asimismo, se generan nuevos árboles para llegar a una predicción final. Una de las ventajas de esta técnica es que el proceso se realiza en paralelo, lo que permite una mayor rapidez y un buen desempeño general. Además, es altamente adaptable a diferentes situaciones y contextos.

➢ *Red neuronal*: Las redes neuronales son un tipo de modelo de aprendizaje profundo que se basan en la estructura y el funcionamiento del cerebro humano para modelar relaciones complejas entre las variables predictoras y la variable de respuesta.

Para este modelo fue necesario **estandarizar** los datos de entrenamiento y testeo (excluyendo las variables *dummies*) para asegurar que cada variable contribuya proporcionalmente al proceso de entrenamiento.

Seteo de experimentación

Cada uno de estos modelos predictivos será entrenado con las mismas variables: precio promedio por noche, cuantas personas caben en este, porcentaje de ocupación por año, semana del año, temporada, mes ocupado alto, tipo de estadía, barrio y puntaje promedio del alojamiento. En este caso cada una de

estas variables serán las mismas que se ingresarán como parámetros en la herramienta. Más adelante se profundizará sobre la selección de las variables y sus definiciones.

Para poder hacer todo esto, y como se explicó previamente, se particionó la base de datos en 2: entrenamiento y testeo. El primer conjunto consistirá en el 80% de los datos de la base original mientras que el conjunto de testeo estará compuesto por el 20% restante. En este caso se fijó una *seed* de 123.

Se estudiará cada uno de los modelos anteriormente mencionados y se trabajará con aquel que predice con mayor certeza cuál es el porcentaje de días que un alojamiento estará ocupado en una determinada semana. Para poder hacer esto se evaluarán los estadísticos que permiten cuantificar el desempeño del modelo, estos son:

- *El coeficiente de determinación* R^2: es una medida estadística que indica cuánta varianza de la variable de respuesta puede ser explicada por las variables predictoras en un modelo de regresión. Proporciona una medida de la calidad de ajuste del modelo y se utiliza para evaluar qué tan bien se ajusta el modelo a los datos observados. Un valor de R^2 cercano a 1 indica que el modelo explica bien la variabilidad de los datos, mientras que un valor cercano a 0 indica que el modelo no es adecuado para explicar los datos.

- *Root Mean Squared Error (RMSE)*: mide cuán bien un modelo se ajusta a los datos. Básicamente, se calcula la raíz cuadrada de la media de los errores al cuadrado. Un RMSE más bajo indica que el modelo se ajusta mejor a los datos, mientras que un RMSE más alto indica que el modelo tiene más errores en sus predicciones.

Desarrollo de la solución

Como fue anticipado, para el desarrollo del entregable final del proyecto se debe seguir una serie de pasos:

1. Diseño de la base de datos con la que el modelo predictivo será entrenado

Para desarrollar el modelo predictivo se cuenta con las bases de **listings** y **calendar** previamente analizadas. Uniéndolas se obtiene un conjunto que resume la disponibilidad de cada alojamiento por día, sumado a sus características. Por ejemplo, para el alojamiento de id = 130424, entre el 30 de marzo de 2023 y el 8 de abril de 2024 se resume la información en la siguiente tabla:

Variables de calendar

Variables de listings

listing_id	date	dolar_price	available	minimum_nights	maximum_nights	id	room_type	review_scores_rating	week_year
130424	2023-03-30	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W13
130424	2023-03-31	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W13
130424	2023-04-01	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W13
130424	2023-04-02	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W13
130424	2023-04-03	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W14
130424	2023-04-04	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W14
130424	2023-04-05	26.60102	FALSE	7	90	130424	Entire home/apt	4.89	2023 W14
130424	2023-04-06	26.60102	TRUE	7	90	130424	Entire home/apt	4.89	2023 W14
130424	2023-04-07	26.60102	TRUE	7	90	130424	Entire home/apt	4.89	2023 W14
130424	2023-04-08	26.60102	TRUE	7	90	130424	Entire home/apt	4.89	2023 W14

Tabla 13. *Resumen de información para el alojamiento con id = 130424.*

Como el objetivo del modelo predictivo no es predecir si un alojamiento estará o no ocupado en determinado día, la variable a predecir no puede ser *available*. Es por esto que se decidió agrupar la información sobre la ocupación de los alojamientos por semana del año, promediando variables de precio, mes, estadía mínima y estadía máxima, y contando la cantidad de días que el alojamiento aparece o no disponible. Así se obtuvo el dataset que se resume a continuación:

listing_id	week_year	prom_dolar_price	prom_min_nights	prom_max_nights	room_type	review_scores_rating	dias_ocupado	dias_disponible
130424	2022 W31	22.00446	7	90	Entire home/apt	4.89	0	7
130424	2023 W07	25.17143	7	90	Entire home/apt	4.89	0	7
130424	2023 W04	25.17143	7	90	Entire home/apt	4.89	0	7
130424	2022 W40	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2022 W46	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2022 W42	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2022 W50	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2023 W21	26.60102	7	90	Entire home/apt	4.89	0	7
130424	2022 W49	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2022 W47	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2023 W02	25.17143	7	90	Entire home/apt	4.89	0	7
130424	2023 W25	26.60102	7	90	Entire home/apt	4.89	0	7
130424	2022 W51	20.17422	7	90	Entire home/apt	4.89	0	7
130424	2022 W27	22.00446	7	90	Entire home/apt	4.89	0	7
130424	2022 W38	20.95861	7	90	Entire home/apt	4.89	4	3

Tabla 14. *Resumen de información para el alojamiento con id = 130424.*

Se tomaron las observaciones con semanas de 7 días y los alojamientos de *room_type* = “Entire home/apt” (comprenden casi el 90% de la totalidad).

2. Selección de variables y feature engineering.

Una vez rediseñada la base de datos, se seleccionaron las variables a tomar en cuenta para entrenar el modelo.

1. *week* y *prom_dolar_price:* se toman en consideración debido a su alta relación con la ocupación de los alojamientos, como se observó en el EDA.

2. *prom_min_nights* y *prom_max_nights*: no se observó una relación clara entre estas variables y la ocupación de los alojamientos. Se decidió construir la variable *tipo_estadia* que resume ambas variables en las siguientes categorías:
    - **corta** si el huésped se puede quedar hasta 5 días.
    - **media 1** si el huésped se puede quedar entre 5 días y 3 semanas.
    - **media 2** si el huésped se puede quedar entre 3 semanas y 3 meses.
    - **larga** si el huésped se puede quedar más de 3 meses.

3. *accommodates*: la cantidad de huéspedes que caben en un alojamiento es indicador de su tamaño, por lo que está relacionado a la fácil ocupación o no del mismo.

4. *review_scores_rating:* altos ratings para los alojamientos pueden llevar a una más fácil ocupación de los mismos si los precios son acordes.

5. *percent_booked:* el porcentaje de ocupación del alojamiento en el año, es un indicador de cuán bien posicionado está el alojamiento en relación a la competencia.

6. *mes_ocupado_alto*: dentro de las cuatro semanas que se encuentran dentro de un mes, en caso que los alojamientos estén más de 4 días ocupados cada semana, los mismo entran en la categoría "Alto", caso contrario entrarían en la categoría "Bajo". Una vez contando con esta nueva característica, si es que la suma de los alojamientos que se encuentran bajo la categoría "Alto" es mayor a los que se encuentran en la categoría "Bajo", el mes del cual se estaría tratando contaría como un mes ocupado alto.

7. *temporada*: esta variable representa aquellos periodos de tiempo, tanto festivos como de vacaciones, que hay en el año. Se considera que una semana es de temporada "alta" cuando la mayoría de sus días caen dentro de alguno de estos rangos de fechas:
    - Semana Santa
    - Primera quincena de enero
    - Primera quincena de diciembre
    - Primera quincena de julio

8. *neighbourhood_cleansed*: los barrios del cluster 2, al que pertenecen los departamentos de Marcelo.

Uno de los problemas que pueden llegar a aparecer a la hora de realizar estos modelos es la multicolinealidad. Esto sucede cuando hay una alta correlación entre dos o más variables predictoras. Esto no solo lleva a que la precisión y confiabilidad de los resultados encontrados del modelo disminuya sino que también dificulta la interpretación de los efectos que tiene cada variable en la variable de respuesta.

Se entiende que hay multicolinealidad cuando el valor del factor de inflación de la varianza (VIF) es mayor a 5 o 10. Es por esto que, durante en análisis de cada modelo, se utilizó la función VIF() para ver si existe o no esta alta correlación.

Haciendo esto se noto como la variable "*room_type*" daba un valor extremadamente alto. Ante esto se decidió quitarla de los modelos.

Cabe aclarar que casi el 90% de los alojamientos tienen como característica: entire home/apt dentro de la variable "room_type", por lo que no pareció pertinente su inclusión en los modelos.

3. Desarrollo del modelo

Se entrenaron 3 modelos con las variables seleccionadas:

a. XGBoost
b. Random Forest

c. Red Neuronal

Los modelos presentados fueron perfeccionados a través del tuneo de sus hiperparámetros. Para cada tipo de modelo se seleccionó la mejor combinación de los mismos, definida a través de su R^2 y RMSE al evaluar sobre el conjunto de testeo. En primer lugar, el R^2 se utilizará para evaluar la proporción de la varianza de *dias_ocupado* que es explicada por las variables independientes. Por otro lado, el RMSE se utilizará para evaluar el rendimiento de los modelos, siendo que es una medida que magnifica el error promedio de la predicción para cada modelo en las unidades de la variable a predecir, en este caso días.

El R^2 es calculado con esta función:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

El RMSE es calculado con esta función:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

y_i = valor real de dias_ocupado para la observación i.

$\hat{y}_i$ = valor predicho de dias_ocupado para la observación i.

$\bar{y}_i$ = media de dias_ocupado.

A continuación se describe el proceso de tuneo de los hiperparámetros para cada uno de los modelos.

XGBoost:

El modelo de XGBoost se construyó utilizando la librería de Python *xgboost*:

```
from xgboost import XGBRegressor
model = XGBRegressor(n_estimators=n_estimators_trial,max_depth=max_depth_trial)
```

El tuneo de los hiperparámetros se realizó utilizando la función *FunctionBestParamsXGB*, que construye todos los modelos que surgen de la combinación de los hiperparámetros definidos. Los hiperparámetros evaluados para el XGBoost fueron:

- ***n_estimators****:* la cantidad de árboles en el bosque.

  Valores: [50, 366, 683, 1000]

- ***max_depth***: la profundidad máxima de los árboles del bosque.

Valores: [10, 15, 20]

Para el modelo de XGBoost una buena manera de iniciar el tuneo de hiperparámetros es evaluar distintas estructuras del bosque a través de variar *n_estimators* y *max_depth*. Si bien hay más hiperparámetros que se pueden modificar para un mayor perfeccionamiento del modelo (*learning_rate, alpha*, *lambda*, *gamma,* etc) sólo se consideraron estos 2 ya que el entrenamiento de modelos con más hiperparámetros es muy costosa en tiempo y memoria de cómputo.

Dados los hiperparámetros y sus valores se obtiene un total de 12 modelos diferentes a entrenar. A continuación se muestran la función *FunctionBestParamsXGB* y los resultados obtenidos tras evaluar los 12 modelos.

```
def FunctionFindBestParamsXGB(X_train, y_train, X_test, y_test):

    # Defino los hiperparámetros a evaluar
    n_estimators = [int(x) for x in np.linspace(start = 50, stop = 1000, num = 4)]
    max_depth = [10,15,20]

    SearchResultsData=pd.DataFrame(columns=['TrialNumber', 'Parameters','MSE','R2'])

    # Inicializo los trials
    TrialNumber=0
    for n_estimators_trial in n_estimators:
        for max_depth_trial in max_depth:
            TrialNumber+=1

            # Creo el XGBoost
            model = XGBRegressor(n_estimators=n_estimators_trial,max_depth=max_depth_trial)

            # Entreno el modelo
            model.fit(X_train, y_train)

            # Hago las predicciones
            y_pred = model.predict(X_test)

            # Evalúo el modelo
            MSE = mean_squared_error(y_test, y_pred)
            R_squared = r2_score(y_test, y_pred)

            # Imprimo los resultados de la iteración actual
            print(TrialNumber, 'Parameters:','n_estimators:', n_estimators_trial,'-', 'max_depth:',max_depth_trial)

            # Agrego la información del modelo a la tabla de modelos
            SearchResultsData=SearchResultsData.append(pd.DataFrame(data=[[TrialNumber, str(n_estimators_trial)+'-'+
                                                                           str(max_depth_trial), MSE, R_squared]],
                                                                    columns=['TrialNumber', 'Parameters','MSE','R2'] ))
    return(SearchResultsData)
```

Fig 34. *Función FunctionBestParamsXGB.*

Al correr el siguiente código se obtienen los estadísticos de los modelos construidos observados en la Tabla 15, en donde el primer valor de la columna Parameters corresponde a *n_estimators* y el segundo a *max_depth*:

```
modelosXGB=FunctionFindBestParamsXGB(X_train_nb, y_train_nb, X_test_nb, y_test_nb)
```

```
modelosXGB['RMSE'] = np.sqrt(modelosXGB['MSE'])
modelosXGB['TrialNumber', 'Parameters', 'MSE','RMSE','R2']
```

	TrialNumber	Parameters	MSE	RMSE	R2
0	1	50-10	4.750738	2.179619	0.534042
0	2	50-15	3.603566	1.898306	0.646558
0	3	50-20	3.146459	1.773826	0.691392
0	4	366-10	3.363894	1.834092	0.670066
0	5	366-15	2.746890	1.657374	0.730582
0	6	366-20	2.969717	1.723287	0.708727
0	7	683-10	3.027491	1.739969	0.703060
0	8	683-15	2.723251	1.650228	0.732901
0	9	683-20	2.969586	1.723249	0.708740
0	10	1000-10	2.875820	1.695824	0.717936
0	11	1000-15	2.721393	1.649665	0.733083
0	12	1000-20	2.969586	1.723249	0.708740

Tabla 15. *Estadísticos de cada modelo de XGBoost al evaluar sobre el conjunto de testeo.*

La Figura 33 muestra como en la mayoría de los modelos una profundidad de 15 es la mejor dado el mismo número de árboles en el bosque. Es destacable cómo mejora el R^2 con una profundidad de 15 si se compara con los otros valores.

Por otro lado, el mayor perfeccionamiento del modelo se da cuando se pasa de un XGBoost de 50 árboles a uno de 366. Entre estos modelos se da el mayor mejoramiento tanto del R^2 como el RMSE:

- El R^2 pasa de 69,1% a 73,1%.
- El RMSE pasa de 1,77 días a 1,66 días.

Una vez dado esté salto la mejora del modelo no se vuelve tan significativa a medida que se aumenta la cantidad de árboles en el bosque.

Debido a esto, se considera que el mejor modelo es el del *TrialNumber* = 5:

- ***n_estimators* = 366**
- ***max_depth = 15***
- ***R^2 = 73,1%***
- ***RMSE = 1,66 días***

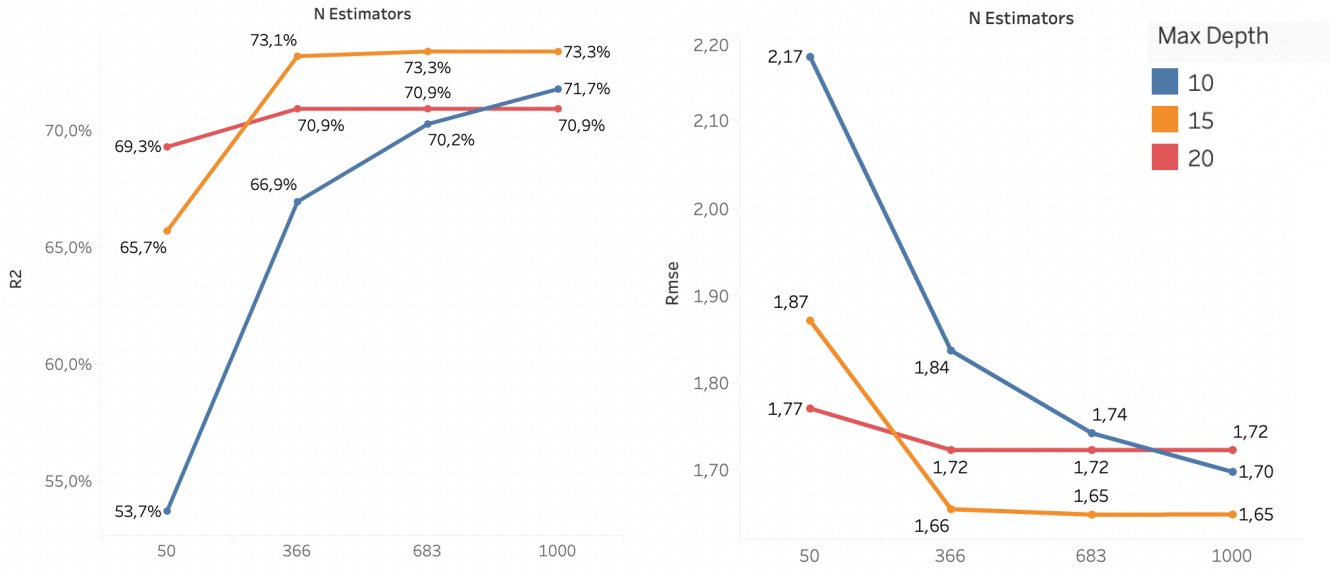


Fig 33. *Estadísticos del XGBoost en función de n_estimators y max_depth.*

Una vez seleccionado el mejor modelo de XGBoost se graficó la importancia de las variables:

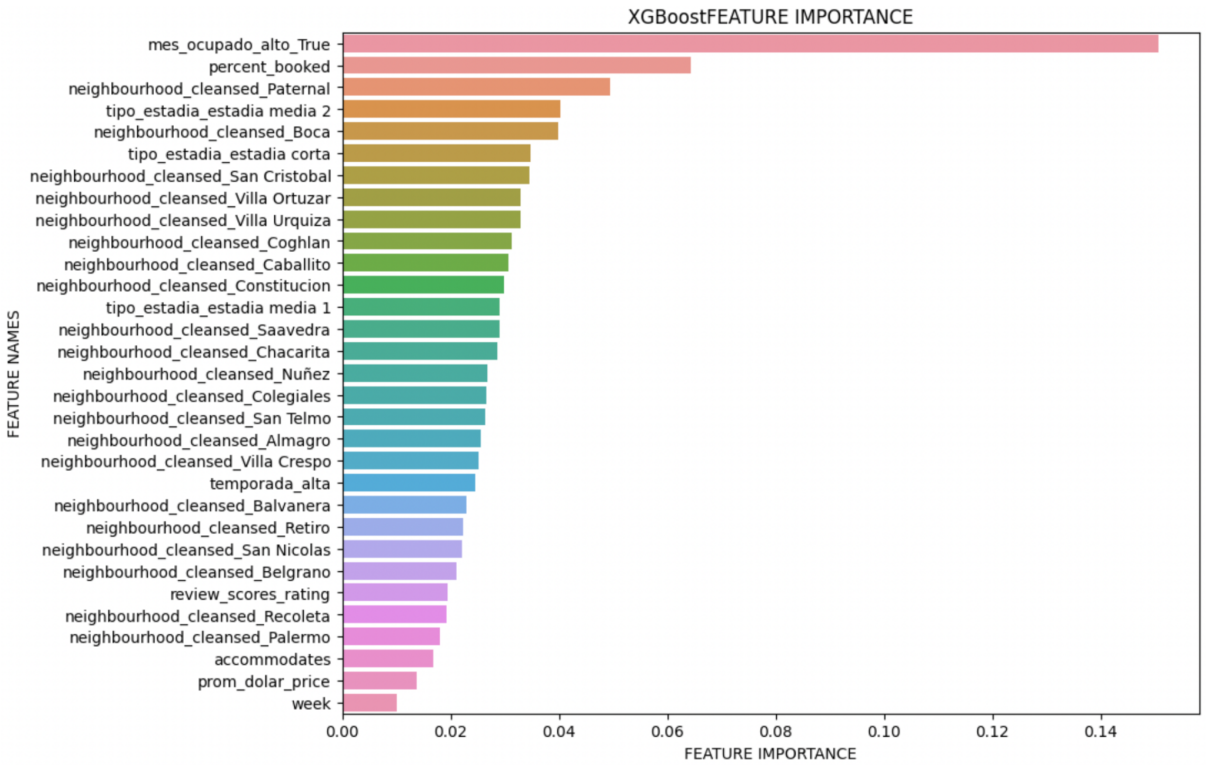


Fig 33. *Feature importance de XGBoost.*

Random Forest:

El modelo de Random Forest se construyó utilizando la librería de Python *Scikit-Learn*:

```
from sklearn.ensemble import RandomForestRegressor
```

```
model = RandomForestRegressor(n_estimators=n_estimators_trial,max_features=max_features_trial)
```

El tuneo de los hiperparámetros se realizó utilizando la función *FunctionBestParamsRF*, que construye todos los modelos que surgen de la combinación de los hiperparámetros definidos. Los hiperparámetros evaluados para el Random Forest fueron:

- ***n_estimators****:* la cantidad de árboles en el bosque.

  Valores: [50, 200, 350, 500, 650, 800]

- ***max_features***: la cantidad de variables a evaluar cuando se busca el mejor "split". Si *max_features* = 'auto', se utiliza como máximo todas las variables para la construcción de cada árbol mientras que si *max_features* = 'sqrt' se utiliza como máximo √(*cantidad de variables*) variables para la construcción de cada árbol

  Valores: ['auto', 'sqrt']

Tal como describe la documentación de esta librería, para el modelo de Random Forest los hiperparámetros más importantes son *n_estimators* y *n_features*. Si bien hay más hiperparámetros que se pueden modificar (*max_depth*, *min_samples_split, min_samples_leaf*, *bootstrap*, etc) sólo se consideraron estos 2 ya que, al igual que el modelo de XGBoost, el entrenamiento de modelos con más hiperparámetros es más costosa en tiempo y memoria de cómputo.

Dados los hiperparámetros y sus valores se obtiene un total de 12 modelos diferentes a entrenar. A continuación se muestran la función *FunctionBestParamsRF* y los resultados obtenidos tras evaluar los 12 modelos de Random Forest.

```python
def FunctionFindBestParamsRF(X_train, y_train, X_test, y_test):

    # Defino los hiperparámetros a evaluar
    n_estimators = [int(x) for x in np.linspace(start = 50, stop = 800, num = 6)]
    max_features = ['auto', 'sqrt']

    SearchResultsData=pd.DataFrame(columns=['TrialNumber', 'Parameters','MSE','R2'])

    # Inicializo los trials
    TrialNumber=0
    for n_estimators_trial in n_estimators:
        for max_features_trial in max_features:
            TrialNumber+=1

            # Creo el RANDOM FOREST
            model = RandomForestRegressor(n_estimators=n_estimators_trial,max_features=max_features_trial)

            # Entreno el modelo
            model.fit(X_train, y_train)

            # Hago las predicciones
            y_pred = model.predict(X_test)

            # Evalúo el modelo
            MSE = mean_squared_error(y_test, y_pred)
            R_squared = r2_score(y_test, y_pred)

            # Imprimo los resultados de la iteración actual
            print(TrialNumber, 'Parameters:','n_estimators:', n_estimators_trial,'-', 'max_features:',max_features_trial)

            # Agrego la información del modelo a la tabla de modelos
            SearchResultsData=SearchResultsData.append(pd.DataFrame(data=[[TrialNumber, str(n_estimators_trial)+'-'+
                                                                           str(max_features_trial), MSE, R_squared]],
                                                                    columns=['TrialNumber', 'Parameters','MSE','R2'] ))
    return(SearchResultsData)
```

Fig 34. *Función FunctionBestParamsRF.*

Al correr el siguiente código se obtienen los estadísticos de los modelos construidos observados en la Tabla 13, en donde el primer valor de la columna Parameters corresponde a *n_estimators* y el segundo a *max_features*:

```python
modelosRF=FunctionFindBestParamsRF(X_train_nb, y_train_nb, X_test_nb, y_test_nb)
```

```python
modelosRF['RMSE'] = np.sqrt(modelosRF['MSE'])
modelosRF['TrialNumber', 'Parameters', 'MSE','RMSE','R2']
```

	TrialNumber	Parameters	R2	MSE	RMSE
0	1	50-auto	0.678567	3.279526	1.810946
0	2	50-sqrt	0.670230	3.364589	1.834282
0	3	200-auto	0.683376	3.230464	1.797349
0	4	200-sqrt	0.678651	3.278675	1.810711
0	5	350-auto	0.684503	3.218964	1.794147
0	6	350-sqrt	0.680268	3.262172	1.806148
0	7	500-auto	0.684588	3.218092	1.793904
0	8	500-sqrt	0.679998	3.264923	1.806910
0	9	650-auto	0.684880	3.215118	1.793075
0	10	650-sqrt	0.680508	3.259729	1.805472
0	11	800-auto	0.684978	3.214122	1.792797
0	12	800-sqrt	0.680556	3.259235	1.805335

Tabla 13. *Estadísticos de cada modelo de Random Forest al evaluar sobre el conjunto de testeo.*

Los gráficos de la Figuras 34 muestran que ante la misma cantidad de árboles en el bosque resulta mejor modelo aquel que toma como máximo todas las variables para la construcción de los árboles (*max_features = 'auto'*) que aquel que toma como máximo √(*cantidad de variables*). Esto se cumple tanto para la métrica R^2 como para el RMSE.

Observando los modelos que utilizan *max_features = 'auto'* se concluye que hay una diferencia significativa en el R^2 y el RMSE cuando los bosques pasan de tener 50 árboles a 200 árboles. A partir de ese punto la mejora de los estadísticos es muy sutil. Si se aumenta la cantidad de árboles de 200 a 800 el R^2 sólo mejora en 0,0015 puntos porcentuales y el RMSE en 0,004 días.

Debido a esto, se considera que el mejor modelo es el del *TrialNumber* = 3:

- ***n_estimators* = 200**
- ***max_features* = 'auto'**
- ***R^2 = 68%***
- ***RMSE = 1,8 días de ocupación***

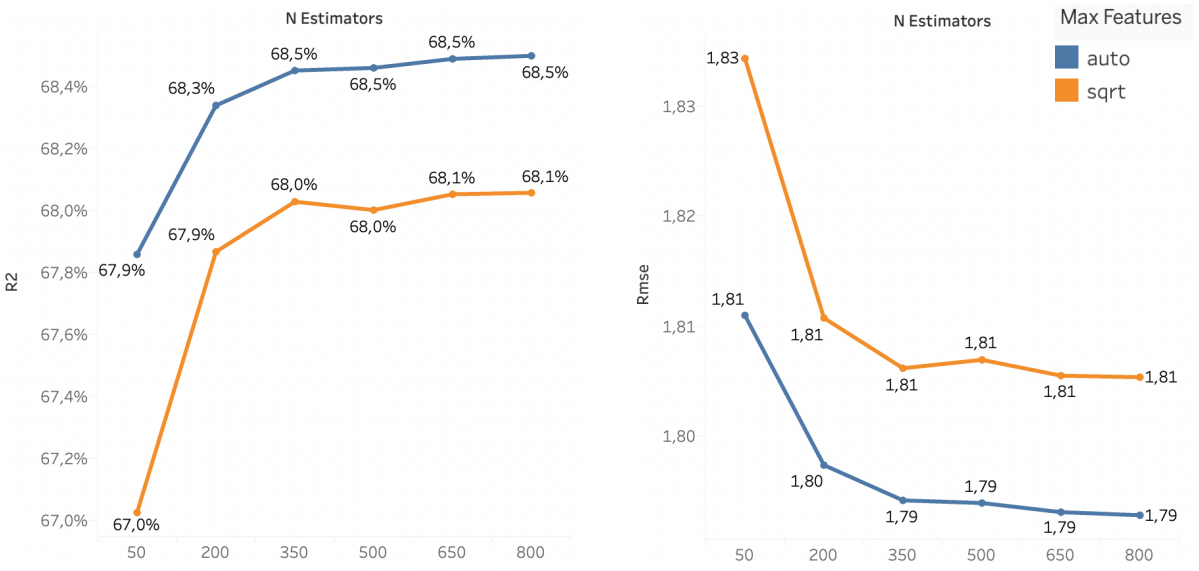


Fig 34. *Estadísticos del Random Forest en función de n_estimators y max_features.*

Una vez seleccionado el mejor modelo de Random Forest se graficó la importancia de las variables:

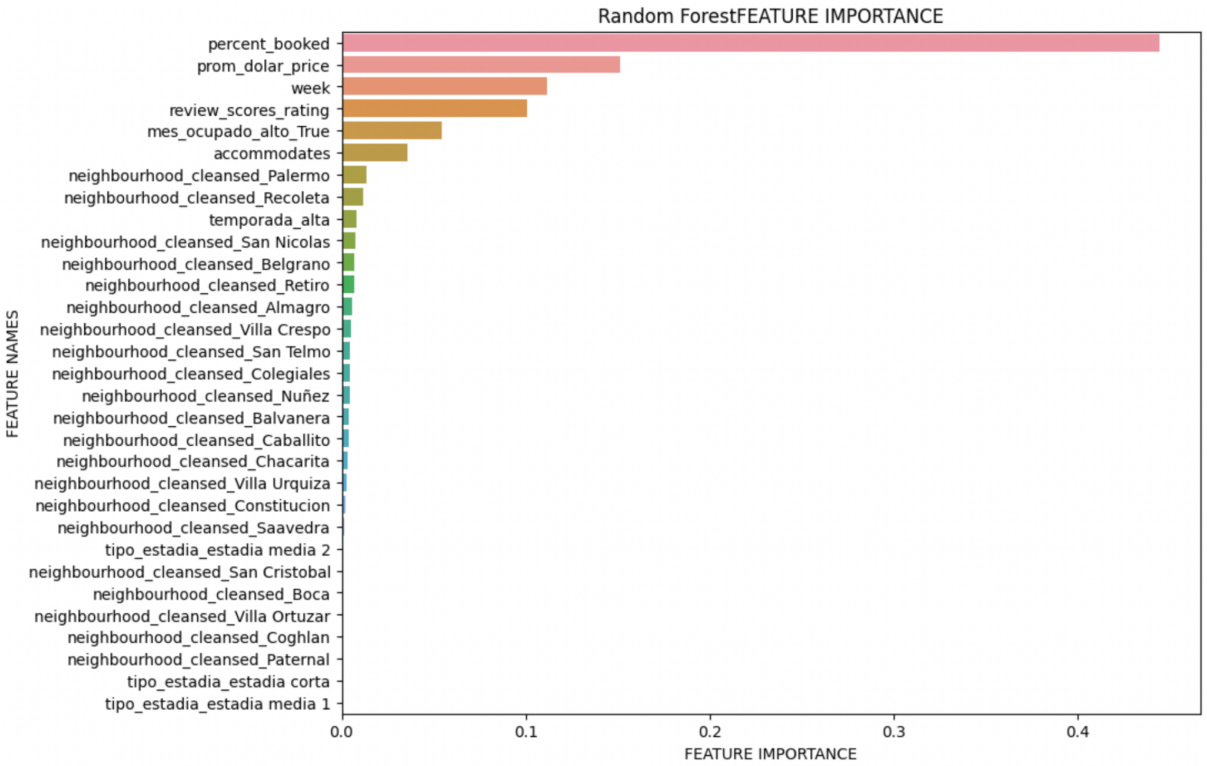


Fig 35. *Feature importance de Random Forest.*

Red Neuronal:

El modelo de Red Neuronal se construyó utilizando la librería de Python *Keras*:

```
from keras.models import Sequential
from keras.layers import Dense
```

```
# Creo la RED NEURONAL
model = Sequential()

# Defino las capas de la RED NEURONAL
model.add(Dense(units=30, input_dim=X_train.shape[1], kernel_initializer='normal', activation='relu'))
model.add(Dense(units=30, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))

# Compilo el modelo
model.compile(loss='mean_squared_error', optimizer='adam')
```

Se definieron 3 capas de neuronas: la primera con una función de activación *relu*, y las últimas dos con una función de activación *normal*. Se definieron 30 neuronas ocultas. Este valor se eligió tomando la "regla" presentada por Jeff Heaton, data scientist y autor de *Introduction to Neural Networks for Java*, que menciona varios métodos para definir la cantidad de neuronas ocultas, entre los que se encuentra el siguiente:

El número de neuronas en cada capa oculta debería estar entre el tamaño la capa de input (en este caso 34) y el tamaño de la capa de output (en este caso 1).

El tuneo de los hiperparámetros se realizó utilizando la función *FunctionBestParamsNN*, que construye todos los modelos que surgen de la combinación de los hiperparámetros definidos. Los hiperparámetros evaluados para la Red Neuronal fueron:

- ***batch_size**:* la cantidad de observaciones utilizadas en cada iteración para la actualización de los pesos del entrenamiento de la red.

  Valores: [20, 30]

- ***epochs***: la cantidad de veces que el conjunto de entrenamiento atraviesa el proceso de entrenamiento. Es la cantidad de iteraciones completas realizadas en el entrenamiento de la red.

  Valores: [10, 20, 30, 40]

Dados los hiperparámetros y sus valores se obtiene un total de 8 modelos diferentes a entrenar. A continuación se muestran la función *FunctionBestParamsNN* y los resultados obtenidos obtenidos tras evaluar los 8 modelos de Red Neuronal.

```
def FunctionFindBestParamsNN(X_train, y_train, X_test, y_test, units):

    # Defino los hiperparámetros a evaluar
    batch_size = [20,30]
    epoch  = [10,20,30,40]
    SearchResultsData=pd.DataFrame(columns=['TrialNumber', 'Parameters','MSE','R2'])

    # Inicializo los trials
    TrialNumber=0
    for batch_size_trial in batch_size:
        for epochs_trial in epoch:
            print("Batch size: ",batch_size_trial," - Epoch: ",epochs_trial)
            TrialNumber+=1

            # Creo la RED NEURONAL
            model = Sequential()

            # Defino las capas de la RED NEURONAL
            model.add(Dense(units=units, input_dim=X_train.shape[1], kernel_initializer='normal', activation='relu'))
            model.add(Dense(units=units, kernel_initializer='normal', activation='relu'))
            model.add(Dense(1, kernel_initializer='normal'))

            # Compilo el modelo
            model.compile(loss='mean_squared_error', optimizer='adam')

            # Entreno el modelo
            model.fit(X_train, y_train ,batch_size = batch_size_trial, epochs = epochs_trial, verbose=1)

            # Hago las predicciones
            y_pred = model.predict(X_test)

            # Evalúo el modelo
            MSE = mean_squared_error(y_test, y_pred)
            R_squared = r2_score(y_test, y_pred)

            # Imprimo los resultados de la iteración actual
            print(TrialNumber, 'Parameters:','batch_size:', batch_size_trial,'-', 'epochs:',epochs_trial)

            # Agrego la información del modelo a la tabla de modelos
            SearchResultsData=SearchResultsData.append(pd.DataFrame(data=[[TrialNumber, str(batch_size_trial)+
                                                                           '-'+str(epochs_trial), MSE, R_squared]],
                                                                    columns=['TrialNumber', 'Parameters','MSE','R2'] ))

    return(SearchResultsData)
```

Fig 36. *Función FunctionBestParamsNN.*

Al correr el siguiente código se obtienen los estadísticos de los modelos construidos observados en la Tabla 14, en donde el primer valor de la columna Parameters corresponde a *batch_size* y el segundo a *epochs*.

```
modelosNN=FunctionFindBestParamsNN(X_train_nb_scale, y_train_nb, X_test_nb_scale, y_test_nb,30)
```

```
modelosNN['RMSE'] = np.sqrt(modelosNN['MSE'])
modelosNN[['TrialNumber', 'Parameters', 'MSE','RMSE','R2']]
```

TrialNumber	Parameters	MSE	RMSE	R2
1	20-10	6.007373	2.450994	0.411205
2	30-10	5.956292	2.440552	0.416212
3	20-20	5.918878	2.432874	0.419879
4	30-20	5.915855	2.432253	0.420175
5	20-30	5.954514	2.440187	0.416386
6	30-30	5.809075	2.410202	0.430641
7	20-40	5.831535	2.414857	0.428440
8	30-40	5.814099	2.411244	0.430149

Tabla 14. *Estadísticos de cada modelo de Red Neuronal al evaluar sobre el conjunto de testeo.*

Observando los gráficos que se encuentran en la Figura 37, se considera que el mejor modelo es el del *TrialNumber* = 6:

- ***epochs* = 30**
- ***batch_size* = 30**
- ***R^2 = 43,1%***
- ***RMSE = 2,41 días***

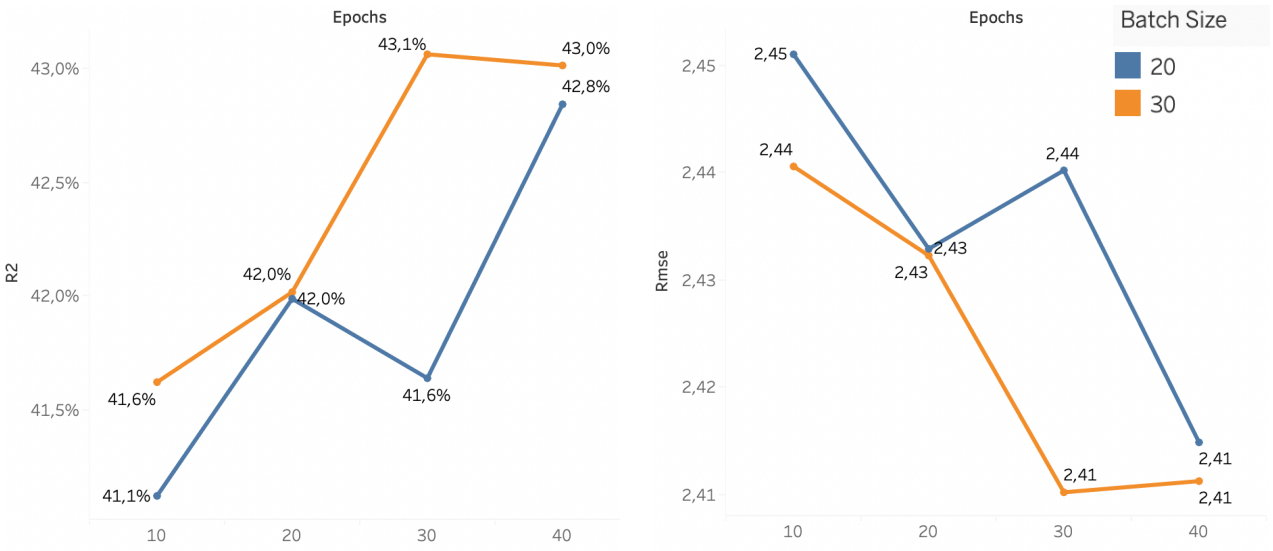


Fig 37. *Estadísticos del Red Neuronal en función de epochs y batch_size.*

A continuación se puede observar una tabla comparativa de cada uno de los modelos seleccionados, con su respectivo R^2 y RMSE, evaluados sobre el conjunto de testeo:

Modelo	R^2	RMSE
XGBoost	73%	1,66
Random forest	68%	1,81
Red neuronal	43%	2,41

Tabla 15. *Comparación de modelos seleccionados*

Si bien el modelo de XGBoost tiene el R^2 más alto y el RMSE más bajo, se decidió seleccionar el modelo de Random Forest para el desarrollo de la herramienta. Mientras que el modelo de XGBoost variables como *prom_dolar_price*, *accommodates* o *week* no tienen alto grado de importancia, en el RandomForest están entre las de mayor peso para la predicción. Estas variables se consideran fundamentales para entender el comportamiento de los alquileres de Airbnb, por lo que se decidió priorizar la interpretabilidad del modelo a coste de peores métricas de predicción, que igualmente cumplen con las expectativas del trabajo.

Análisis de sensibilidad

Se llevó a cabo un análisis de sensibilidad con el objetivo de examinar la distribución del error, es decir, cómo los errores se distribuyen alrededor de las predicciones del modelo. Además, se buscó identificar posibles patrones o grupos de valores en los cuales el modelo presentó mayores errores. Para esto último se examinaron los errores en función de diferentes variables o características relevantes del conjunto de datos como los barrios y el precio.

En cuanto a la distribución del error, en la figura a continuación se puede apreciar como los errores se distribuyen de manera simétrica alrededor del cero, donde la mayoría de los errores se concentran cerca de cero y disminuyen a medida que se alejan.

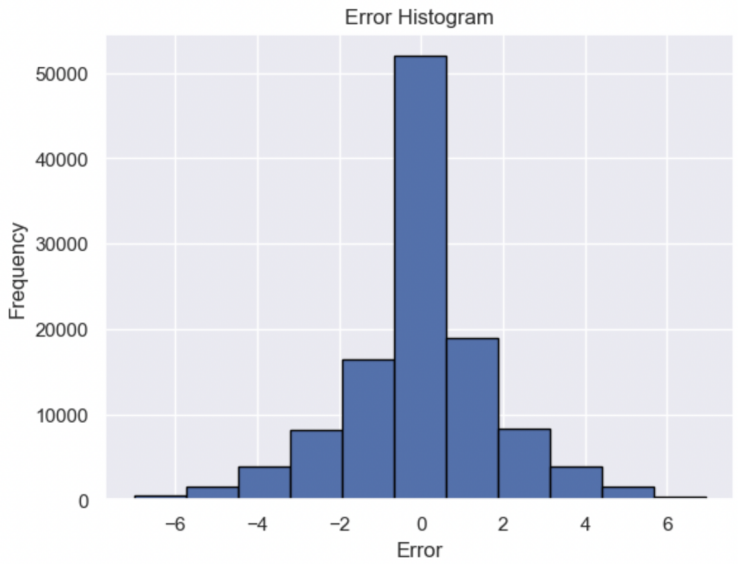


Fig 42. *Distribución del error.*

Esto indica que el valor predicho tiende a ser muy cercano o casi igual al valor real de la cantidad de días que estaría ocupada una propiedad en una semana dada. Los valores negativos y positivos por su parte se distribuyen de manera similar, y además con una media, mediana y moda iguales a 0 como muestra la tabla 17, se puede decir que se trata de una distribución simétrica.

Media	Mediana	Moda	Varianza	Desvío estándar
0	0	0	3,2	1,8

Tabla 17. *Métricas de la distribución.*

En cuanto al análisis de los posibles grupos de error, en primer lugar se examinó el error en función de los barrios en los que se encuentran los alojamientos del conjunto de testeo. Para esto se filtro el error haciendo énfasis en los errores graves, considerados como aquellos cuya diferencia entre el valor predicho y el valor real es mayor o igual a 2, o, menor o igual a -2, y se observó la cantidad de propiedades con este tipo de error que hay por barrio. Esto se puede visualizar en la figura a continuación:

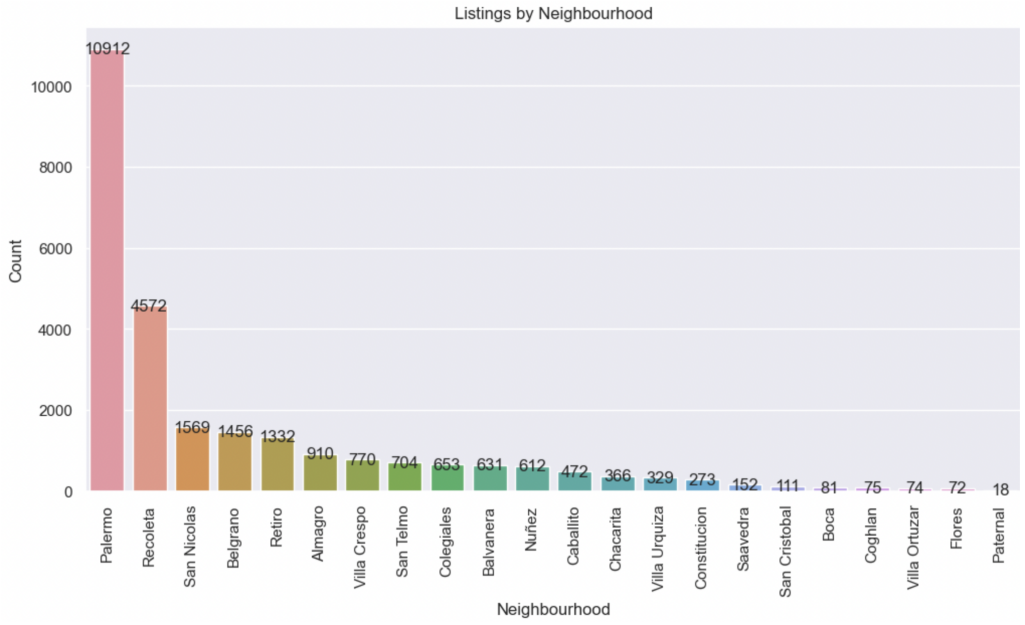


Fig 43. *Distribución de los barrios para los errores graves.*

Se puede observar que la mayor cantidad de errores graves se encuentran en el barrio de Palermo, con 10.912 errores, seguido por Recoleta, San Nicolás y Belgrano. Sin embargo, esto puede deberse a que en estos barrios se encuentran la mayor cantidad de propiedades, como se puede observar en la figura a continuación en la que se visualiza la cantidad total de propiedades por barrio del conjunto de testeo:

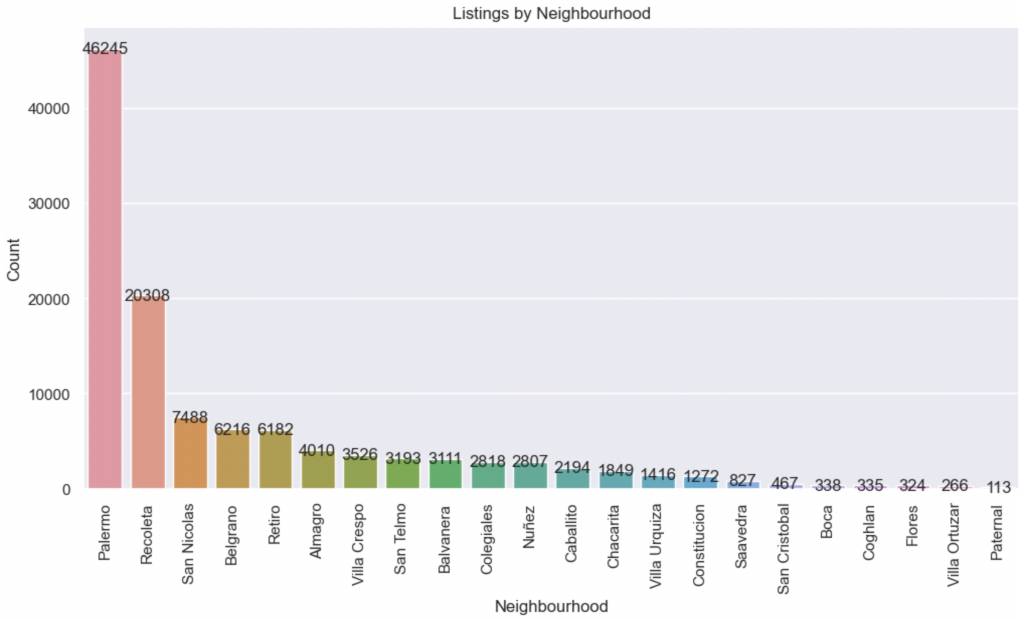


Fig 44. *Distribución de los barrios - test set.*

Se puede apreciar como la cantidad de errores graves encontrados por barrio está relacionada con la cantidad de propiedades que se encuentran en cada barrio, y que el orden de los mismos se mantiene igual, con leves excepciones, en ambos casos. Dado esto, para realizar comparaciones apropiadas se analizó la proporción de propiedades con errores graves que hay en un barrio sobre el total de propiedades de ese mismo barrio. Esta proporción se puede ver en la última columna de la siguiente tabla:

Neighbourhood	Count_x	Count_y	porc
Palermo	46245	10912	0.235961
Recoleta	20308	4572	0.225133
San Nicolas	7488	1569	0.209535
Belgrano	6216	1456	0.234234
Retiro	6182	1332	0.215464
Almagro	4010	910	0.226933
Villa Crespo	3526	770	0.218378
San Telmo	3193	704	0.220482
Balvanera	3111	631	0.202829
Colegiales	2818	653	0.231725
Nuñez	2807	612	0.218026
Caballito	2194	472	0.215132
Chacarita	1849	366	0.197945
Villa Urquiza	1416	329	0.232345
Constitucion	1272	273	0.214623
Saavedra	827	152	0.183797
San Cristobal	467	111	0.237687
Boca	338	81	0.239645
Coghlan	335	75	0.223881
Flores	324	72	0.222222
Villa Ortuzar	266	74	0.278195
Paternal	113	18	0.159292

Tabla 18. *Comparación de barrios.*

En la gran mayoría de los barrios, se obtuvo que alrededor del 21% de las propiedades del mismo presentan errores graves. Los únicos dos barrios cuya proporción se aleja más de 2 o 3 puntos porcentuales de este último valor son Villa Ortuzar con un 27% y Paternal con un 15%, y para el resto de los barrios se puede decir que el error se da de igual manera en cada uno de ellos.

Del mismo modo, se analizó la distribución de los errores graves en función del precio por noche (en dólares). Esto se puede observar en la figura 45 a continuación, donde se aprecia que los valores están sesgados hacia la izquierda, indicando que el error se da con más frecuencia en propiedades cuyo precio es más bajo, especialmente para precios de alrededor de 25 dólares la noche. Sin embargo, si se corrobora con la figura 46 en la que se aprecia la distribución del precio de todas las propiedades del conjunto de testeo, se puede apreciar que la distribución es la misma. Esto quiere decir que la frecuencia de errores graves en cada nivel de precios es proporcional a la cantidad de propiedades del conjunto que ofrecen estos precios.

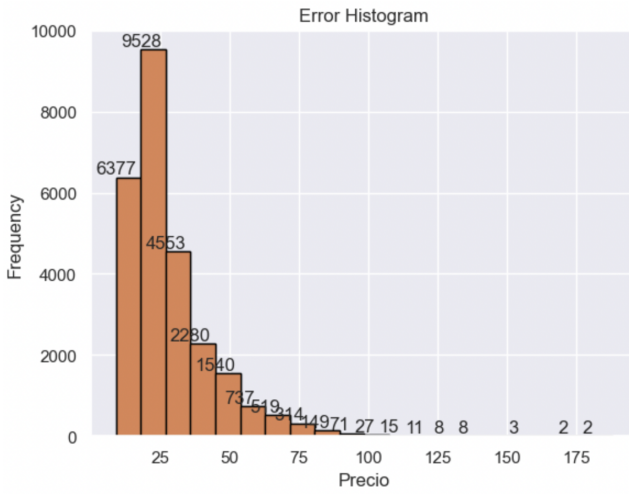


Fig 45. *Distribución precio - errores graves.*

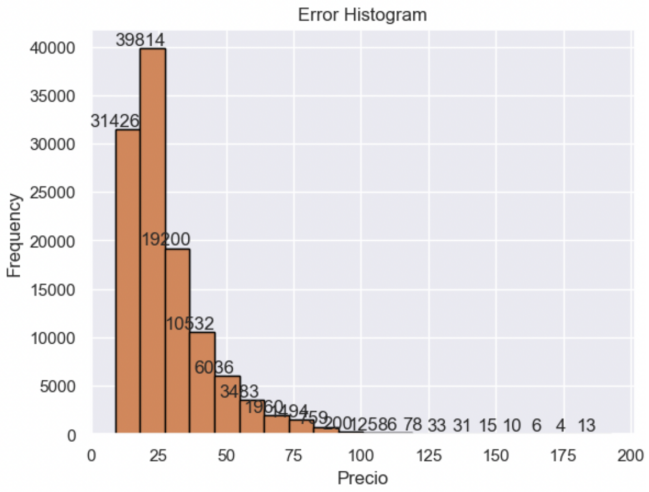


Fig 46. *Distribución precio - test set.*

Se puede decir entonces que, tanto para el precio como para las diferentes variables analizadas, la frecuencia del error se da de manera correspondiente con la frecuencia de los valores que toman cada una de estas variables en el conjunto de datos en cuestión. Si bien se identificaron algunos barrios para los cuales el error se daba con mayor o menor frecuencia, ninguno de estos eran casos muy significativos. Por lo tanto, se podría decir que no se identifican grupos muy fuertes que vayan a inducir al error en mayor o menor medida. Por otra parte, gracias al análisis de sensibilidad se obtuvo información valiosa sobre la distribución de los errores generados por el modelo, reconociendo una distribución simétrica alrededor del cero.

4. Integración a la herramienta

Habiendo establecido qué modelo se utilizará para la predicción, el próximo paso fue crear una función llamada predict que al ingresar valores para las variables con las que se corre el modelo, devuelve la predicción sobre la cantidad de días que estará ocupada una propiedad con las características o valores ingresados en la semana ingresada. A continuación se puede observar parte de el código de la misma:

```python
def predict():

    # Crear un diccionario con los valores de la nueva propiedad
    new_prop = {
        'week': int(request.form['week']),
        'prom_dolar_price': float(request.form['prom_dolar_price']),
        'percent_booked': float(request.form['percent_booked']),
        'accommodates': int(request.form['accommodates']),
        'review_scores_rating': float(request.form['review_scores_rating']),
        'temporada_alta': int(request.form['temporada_alta']),
        'tipo_estadia_estadia corta': int(request.form['tipo_estadia_estadia corta']),
        'tipo_estadia_estadia media 1': int(request.form['tipo_estadia_estadia media 1']),
        'tipo_estadia_estadia media 2': int(request.form['tipo_estadia_estadia media 2']),
        'mes_ocupado_alto_True': int(request.form['mes_ocupado_alto_True']),
        'Almagro': int(request.form['Almagro']),
        'Balvanera': int(request.form['Balvanera']),
        'Barracas': int(request.form['Barracas']),
        'Belgrano': int(request.form['Belgrano']),
        'Boca': int(request.form['Boca']),
        'Caballito': int(request.form['Caballito']),
        [...] # Resto de los barrios
    }

    # Crear un DataFrame con los valores de la nueva propiedad
    new_data = pd.DataFrame(new_prop, index=[0])

    # Asegurarse de que el tipo de datos de las columnas coincida con el modelo entrenado
    new_data['week'] = new_data['week'].astype(int)
    new_data['prom_dolar_price'] = new_data['prom_dolar_price'].astype(float)
    new_data['percent_booked'] = new_data['percent_booked'].astype(float)
    new_data['accommodates'] = new_data['accommodates'].astype(int)
    new_data['review_scores_rating'] = new_data['review_scores_rating'].astype(float)


    # Utilizar el modelo para hacer la predicción
    prediction = model.predict(new_data)

    return render_template('result.html', prediction=prediction)
```

Fig 38. *Función 1 herramienta*

Como se mencionó anteriormente, la predicción es calculada a partir del modelo de machine learning Random Forest entrenado, que como se puede observar en la figura anterior se lo llama dentro de la función para que prediga con los nuevos datos ingresados.

Para crear un formulario en una página web en donde el usuario pueda ingresar los valores que se le pasaran a la función y obtener la predicción (herramienta que se le presentará a Marcelo) fue necesario utilizar herramientas adicionales como HTML y un entorno de desarrollo web como "Flask" en Python, que permite manejar las solicitudes y respuestas del usuario.

De esta manera, en primer lugar se instaló el paquete *Flask* en el entorno de Python. Una vez hecho esto se creó un archivo llamado "app.py" donde se cargó el modelo entrenado y se corrió la función en la que se creó el nuevo data frame llamado "new_data" que va a almacenar los valores ingresados por el usuario (precio por noche, semana del año, barrio, cantidad de personas que el alojamiento acomoda, etc), para luego pasarlos al modelo para la predicción.

En paralelo se crearon dos archivos HTML dentro de una carpeta llamada "Templates" ubicada en el mismo directorio. Uno de estos archivos se llama "index.html", que contiene el formulario "Formulario de Predicción" donde el usuario va a ingresar los valores para realizar el pronóstico de días ocupados en una semana determinada.

El otro archivo se llama "result.html" que es el responsable de mostrar el resultado de la predicción. A continuación se pueden observar ambos archivos de texto:

templates > result.html > ...

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Resultado de la Predicción</title>
5  </head>
6  <body>
7      <h1>Resultado de la Predicción</h1>
8      <p>La predicción es: {{ prediction }}</p>
9  </body>
10 </html>
```

Fig 39. *Archivo 'result.html"*

templates > index.html > html > body > form

```html
<!DOCTYPE html>
<html>
<head>
    <title>Formulario de Predicción</title>
</head>
<body>
    <h1>Formulario de Predicción</h1>
    <form action="/predict" method="post">
        <!-- Campos de entrada existentes -->
        <label for="week">Semana:</label>
        <input type="number" id="week" name="week" required><br><br>

        <label for="prom_dolar_price">Precio Promedio Dólar:</label>
        <input type="number" step="0.01" id="prom_dolar_price" name="prom_dolar_price" required><br><br>

        <label for="percent_booked">Porcentaje Reservado:</label>
        <input type="number" step="0.01" id="percent_booked" name="percent_booked" required><br><br>

        <label for="accommodates">Capacidad de Alojamiento:</label>
        <input type="number" id="accommodates" name="accommodates" required><br><br>

        <label for="review_scores_rating">Review scores rating:</label>
        <input type="number" step="0.01" id="review_scores_rating" name="review_scores_rating" required><br><br>

        <label for="temporada_alta">Temporada Alta:</label>
        <select id="temporada_alta" name="temporada_alta" required>
            <option value="0">No</option>
            <option value="1">Sí</option>
        </select><br><br>
```

```
            [...] # Resto de las variables

            <input type="submit" value="Predict">
        </form>
    </body>
    </html>
```

Fig 39. *Archivo 'index.html"*

Presentación herramienta

Una vez que se ejecuta el archivo "app.py" en Python y Marcelo accede al link privado http://localhost:5000, en el mismo se plasmara el formulario para ingresar los datos con los que se realizará la predicción; resultado que se podrá visualizar una vez que Marcelo apriete el botón de Predict.

A continuación se puede observar un pantallazo de lo que sería la herramienta que se le entregará a Marcelo. En este caso hay 9 parámetros que el usuario tendrá que ingresar a su gusto para poder recibir la predicción de cuantos días de la semana elegida su alojamiento va a estar ocupado. Una vez seleccionadas estas características, el usuario presiona el botón "predict" y obtiene el resultado de la predicción.

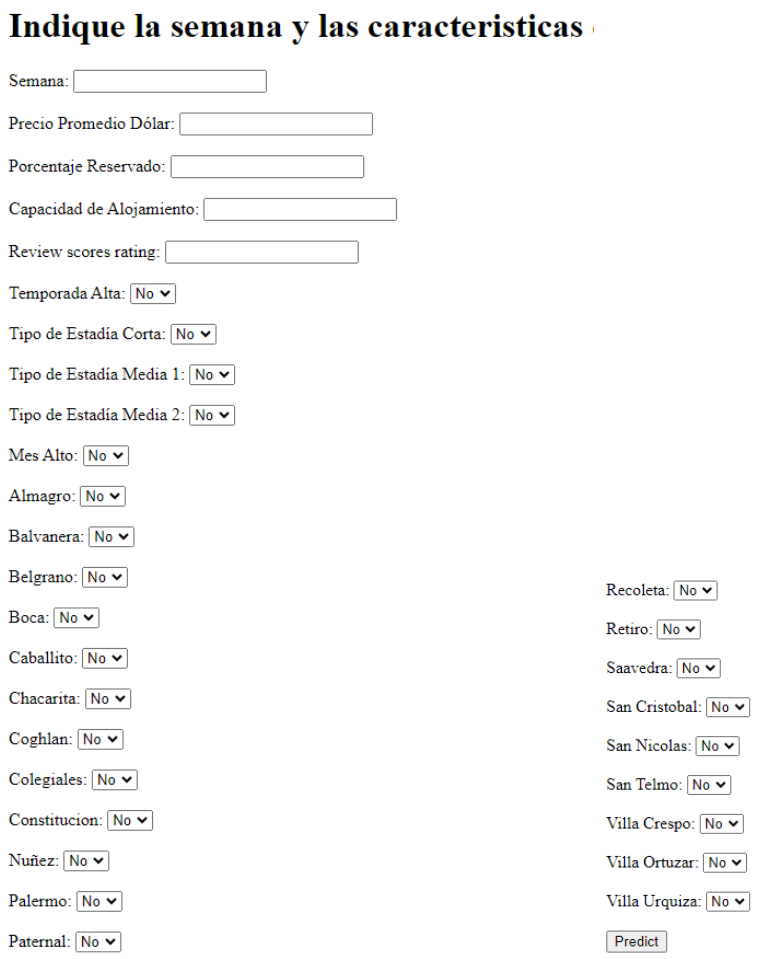


Fig 41. *Captura de pantalla de la aplicación.*

Para tener un mejor entendimiento de la aplicación, a continuación se adjunta un link para observar una demostración del funcionamiento de la herramienta:

- *https://www.loom.com/share/362e59a6a6b1457d960fc13f346bcd19*

A modo de ejemplo, en el video se puede observar como para un departamento para 2 personas en Belgrano, de estadía media, con un porcentaje de ocupación promedio de 42% en el último año y un rating igual a 4, en temporada alta, va a ser de 4 días en la semana 5 por un precio igual a 21 dólares la noche. Cuando este precio se aumenta a 45 dólares, los días de ocupación bajan a 2.

Business Case

La herramienta construida propone que el usuario pueda poner otros precios por noche para evaluar sus alojamientos en AirBnb y así mejorar su ganancia. Mientras los precios sean más bajos, probablemente lo lleven a tener el alojamiento más días ocupados por semana, y los precios más altos probablemente lo lleven a estar más desocupado en condiciones habituales. Cualquiera de esas alternativas puede significar una mayor ganancia.

Se evaluó el potencial de la herramienta para mejorar la ganancia de Marcelo en Airbnb. Si Marcelo hubiese tenido a disposición la herramienta predictiva, ¿cuál sería la diferencia en su ganancia? Para

estimar esta diferencia en la ganancia, se tomó 1 año de las reservas de los departamentos de Marcelo y se realizó una predicción de la *cantidad de días de ocupación* para 5 precios:

- El precio sugerido por Airbnb (el precio al que se alquiló el departamento).
- 10% más que el precio sugerido por Airbnb.
- 10% menos que el precio sugerido por Airbnb.
- 20% más que el precio sugerido por Airbnb.
- 20% menos que el precio sugerido por Airbnb.

A partir de lo anterior se calculó la ganancia esperada a cada valor de precio ([precio]*[predicción de días de ocupación para ese precio]) y se almacenó la mayor ganancia de cada semana, para poder compararla con la ganancia real de dicha semana. A continuación se presenta una tabla que detalla la diferencia entre la ganancia predicha y la ganancia real para cada alojamiento, para la cantidad de semanas consideradas de cada uno:

	Alojamiento 1	**Alojamiento** 2	**Alojamiento** 3	**Alojamiento** 4
Cantidad de semanas	52	52	52	25
Diferencia de ganancia total (dólares)	$396	$1017	$1568	$99
Promedio de diferencia de ganancia por semana (dólares)	$8	$20	$30	$4

Tabla 16. *Métricas de la distribución.*

Los resultados fueron que la diferencia de ganancia estimada para los 4 departamentos de Marcelo para un año hubiese sido de aproximadamente **3.080 dólares**, con un con una diferencia de ganancia promedio de **17 dólares por semana por alojamiento**. Para más detalle sobre el proceso referirse al Anexo.

Conclusión

El modelo predictivo construido para estimar la ocupación de un departamento de Airbnb ha demostrado ser efectivo, con un bajo RMSE de 1,8. Utilizando características relevantes del departamento y el precio por noche, el modelo pudo predecir con precisión la cantidad de días de ocupación. Además, se estima que el anfitrión habría obtenido una ganancia adicional de 3080 dólares si hubiera utilizado este modelo el año pasado.

A pesar de estos resultados prometedores, es importante destacar que el modelo aún tiene margen de mejora. Existe evidencia de sobreajuste, ya que el R2 de entrenamiento es significativamente más alto que el de testeo. Esto indica que el modelo se ajusta demasiado a los datos de entrenamiento y puede tener dificultades para generalizar a nuevos datos.

Se concluye que la implementación de un modelo predictivo en el contexto de alquileres de Airbnb ha demostrado su utilidad al proporcionar la posibilidad de variar el precio así como

otras características de una propiedad permitiendo observar el comportamiento consecuente de la ocupación de la misma y maximizar las ganancias. Sin embargo, se deben abordar las diferentes limitantes del modelo para mejorar aún más la capacidad de predicción y de generalización del mismo.

Mediante las predicciones que la aplicación arrojó sobre cuántos días va a estar ocupado un alojamiento en la semana, notamos como una de las hipótesis que fue planteada al comienzo de la investigación no necesariamente se cumplía.
La hipótesis en cuestión era: *Más alto el precio por noche, menos reservas realizadas en el alojamiento por año*.

En este caso advertimos cómo este comportamiento no siempre es cierto y sucede, ya que esta afirmación no tiene en cuenta factores externos, como: eventos, feriados, temporada vacacional, etc.

Próximos pasos

Contando con la implementación de la aplicación y ya habiendo entregado el producto a Marcelo, hay ciertas funcionalidades y herramientas que serán incorporadas en el futuro en el mismo.

En este caso, en un principio, y a lo largo del tiempo, se ajustará el modelo a medida que se cuenten con los nuevos datos que son cargados en la página de Inside Airbnb. El propósito de esto es que las predicciones que la aplicación realiza esten cada vez más atinadas y cerca de la realidad, lo que generará más confianza en Marcelo a la hora de usar la aplicación ya que al fin y al cabo podrá elegir el precio por noche que más le convenga.

Por otro lado, una de las incorporaciones que se le hará a esta herramienta, va a ser un gráfico. Este gráfico contará con un precio mínimo y un precio máximo que Marcelo está dispuesto a ofrecer en una semana determinada.

Mediante las predicciones de ocupación que la aplicación generará con estos precios, el gráfico plasmará una curva que muestra cómo es que será la ocupación en esa semana determinada a cada precio.

Con esta ayuda visual Marcelo podrá entender de mejor manera cuales son las opciones que tiene y elegirá qué camino desea tomar.

Junto a este gráfico se le brindará a Marcelo aquel precio por noche que le convendría ofrecer en una semana para generar mayor rentabilidad.
Esta rentabilidad podrá ser calculada mediante la multiplicación del precio por noche por la cantidad de días que se predice que ese alojamiento va a estar ocupado en x semana.

Habiendo calculado la rentabilidad para todos los precios posibles entre el mínimo y máximo que Marcelo escogió, se plasmará aquel con mayor rentabilidad.

La aplicación fue creada principalmente para asistir a Marcelo con la incertidumbre con la que él cuenta sobre que precio poner por noche. Esto no quita que esta herramienta pueda ser utilizada por cualquier otro usuario de Airbnb.

Es por esto que, una vez que las funcionalidades previamente mencionadas estén activas en la aplicación, y se haya realizado el ajuste necesario en el modelo para que exista una diferencia casi inexistente entre los valores predichos de ocupación y los que realmente sucedieron, buscaremos expandir la aplicación y su uso en el resto del mundo.

Anexo l

	neighbourhood_cleansed	mean	median	max	min	n
1	Palermo	15518.265	10354.0	2070751	600	7753
2	Recoleta	16620.800	9361.0	3000000	400	3403
3	San Nicolas	10914.783	7987.0	325108	1600	1337
4	Belgrano	12257.935	9318.0	207075	600	1161
5	Retiro	16098.199	8950.0	2300811	1100	1123
6	Monserrat	15984.486	7455.0	2050043	175	902
7	Almagro	12927.521	7101.0	1046350	850	789
8	Balvanera	9583.587	6575.0	421770	693	756
9	Villa Crespo	10977.000	7662.0	639999	1600	740
10	San Telmo	20736.598	8090.5	2532232	1950	634

	room_type	mean	median	max	min	n
1	Private room	11079.35	4763	724763	400	2139
2	Entire home/apt	15680.01	9318	20500432	260	20282
3	Shared room	25269.45	3967	2070544	175	158
4	Hotel room	52378.61	8109	2532232	2071	98

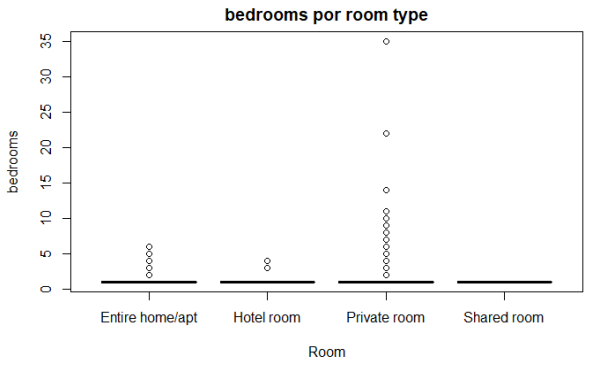
Anexo 1 *- Tablas de precio por Barrio y Tipo de cuarto*

neighbourhood_cleansed	n	Cantidad	porc
Palermo	352	7753	4.5401780
Recoleta	144	3403	4.2315604
San Nicolas	31	1337	2.3186238
Belgrano	36	1161	3.1007752
Retiro	74	1123	6.5894924
Monserrat	21	902	2.3281596
Almagro	25	789	3.1685678
Balvanera	10	756	1.3227513
Villa Crespo	18	740	2.4324324
San Telmo	16	634	2.5236593
Colegiales	23	535	4.2990654
Nuñez	6	491	1.2219959
Caballito	8	436	1.8348624
Chacarita	13	362	3.5911602
Constitucion	8	298	2.6845638

room_type	n	Cantidad	porc
Entire home/apt	770	20282	3.796470
Hotel room	13	98	13.265306
Private room	94	2139	4.394577
Shared room	5	158	3.164557

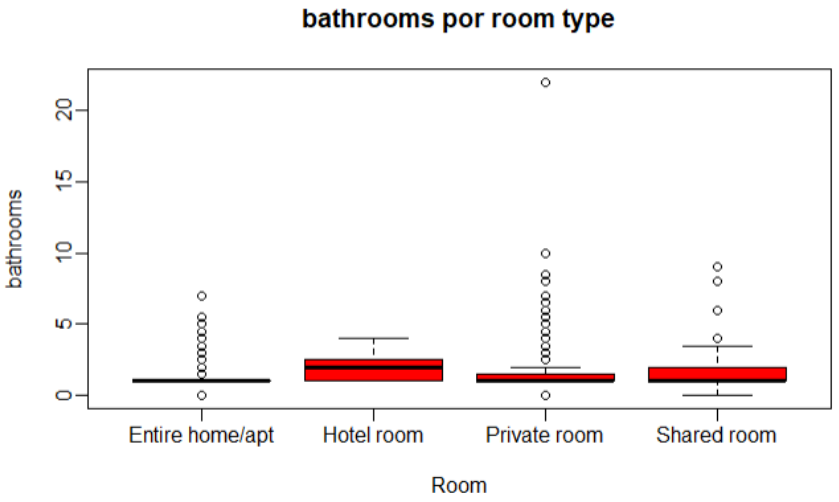
Anexo 2 *- Proporción de valores extremos por barrio y tipo de cuarto:*

	room_type	mean	median	max	min	n
1	Shared room	1.000000	1	1	1	153
2	Hotel room	1.082353	1	4	1	85
3	Entire home/apt	1.263581	1	6	1	19512
4	Private room	1.348166	1	35	1	2045

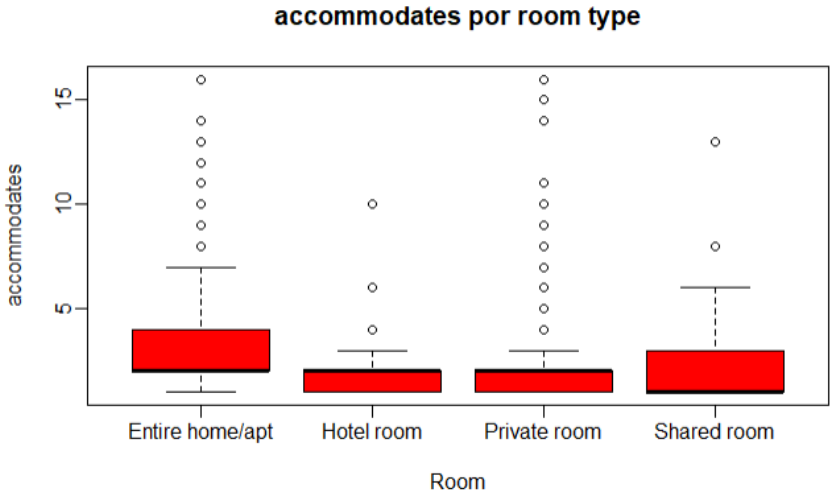


Anexo 3 *- Bedrooms por room type*

	room_type	mean	median	max	min	n
1	Hotel room	1.735294	2	4	1	85
2	Entire home/apt	1.181606	1	7	0	19512
3	Shared room	1.689542	1	9	0	153
4	Private room	1.528606	1	22	0	2045



Anexo 4 *- Bathrooms por room type*



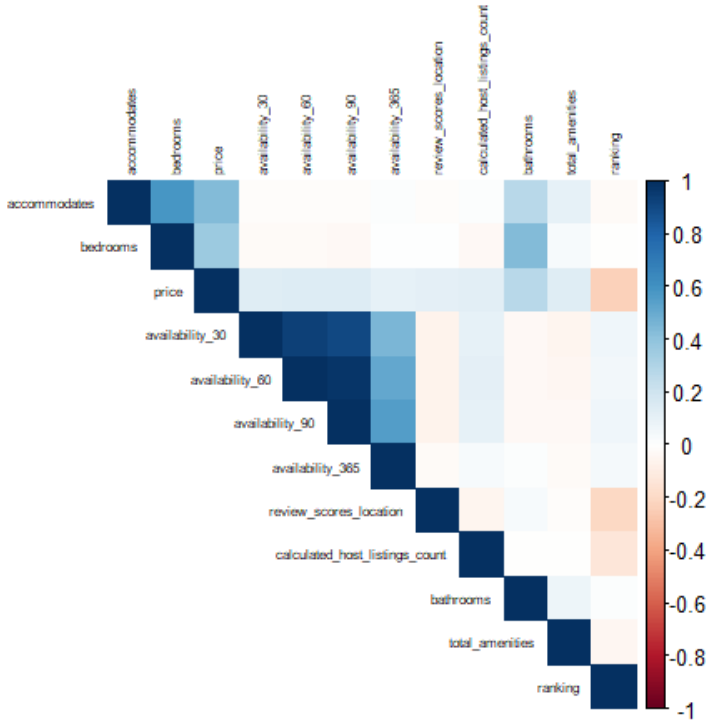
Anexo 5 *- Accommodates por room type.*

	mean	mediana	sd	min	25%	50%	75%	max
availability_30	12.232	12	10.553	0	0	12	22	30
availability_60	32.436	37	20.875	0	13	37	51	60
availability_90	55.275	65	30.253	0	33	65	81	90
availability_365	219.097	245	125.655	0	89	245	343	365
number_of_reviews	22.448	8	39.071	0	1	8	26	637
number_of_reviews_ltm	9.490	4	13.509	0	0	4	13	252
number_of_reviews_l30d	1.018	0	1.685	0	0	0	2	44

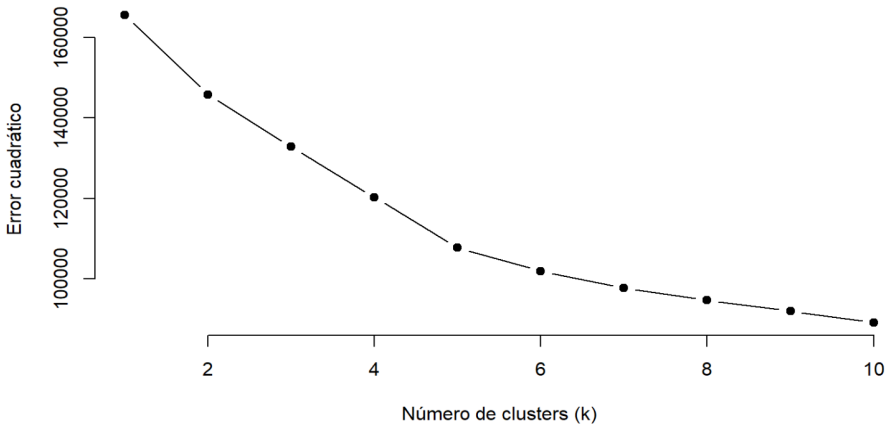
Anexo 6 *- Estadísticos de otras variables cuantitativas*

	matriz_cor[, c("price")]
accommodates	0.435824710
bedrooms	0.363048412
price	1.000000000
minimum_nights	-0.030004282
maximum_nights	0.056906290
availability_30	0.135692045
availability_60	0.141510976
availability_90	0.142326141
availability_365	0.106374821
number_of_reviews	-0.059582948
number_of_reviews_ltm	-0.033322296
number_of_reviews_l30d	-0.075459893
review_scores_rating	0.041645628
review_scores_accuracy	0.029625342
review_scores_cleanliness	0.064490710
review_scores_checkin	0.007518541
review_scores_communication	0.016473254
review_scores_location	0.113013956
review_scores_value	-0.018825161
calculated_host_listings_count	0.124910210
bathrooms	0.271562188
total_amenities	0.138077463

Anexo 7 *- Tabla de correlaciones para todas las variables con precio*



Anexo 8 *- Matriz de correlación para variables reducidas*



Anexo 9 *- Método del codo*

	neighbourhood_cleansed	cluster
1	Puerto Madero	1
2	Almagro	2
3	Balvanera	2
4	Belgrano	2
5	Boca	2
6	Caballito	2
7	Chacarita	2
8	Coghlan	2
9	Colegiales	2
10	Constitucion	2
11	Flores	2
12	Nuñez	2
13	Palermo	2
14	Paternal	2
15	Recoleta	2
16	Retiro	2
17	Saavedra	2
18	San Cristobal	2
19	San Nicolas	2
20	San Telmo	2
21	Villa Crespo	2
22	Villa Ortuzar	2
23	Villa Urquiza	2
24	Agronomia	3
25	Barracas	3
26	Boedo	3
27	Floresta	3
28	Mataderos	3
29	Monserrat	3
30	Monte Castro	3
31	Nueva Pompeya	3
32	Parque Chacabuco	3
33	Parque Chas	3
34	Parque Patricios	3
35	Villa Del Parque	3
36	Villa Devoto	3
37	Villa Pueyrredon	3
38	Villa Santa Rita	3
39	Velez Sarsfield	4
40	Liniers	5
41	Parque Avellaneda	5
42	Villa Gral. Mitre	5
43	Villa Luro	5

Anexo 10 *- Barrios por cluster*

```python
# Predicciones de precio

y_pred_depto1_orig = modelo.predict(X_dpto1_orig)
y_pred_depto1_mas_10 = modelo.predict(X_dpto1_mas_10)
y_pred_depto1_mas_20 = modelo.predict(X_dpto1_mas_20)
y_pred_depto1_menos_10 = modelo.predict(X_dpto1_menos_10)
y_pred_depto1_menos_20 = modelo.predict(X_dpto1_menos_20)
```

```python
predicciones_dpto1 = pd.DataFrame({'dias_ocupado_orig_pred': y_pred_depto1_orig,
                                   'dias_ocupado_mas_10': y_pred_depto1_mas_10,
                                   'dias_ocupado_mas_20': y_pred_depto1_mas_20,
                                   'dias_ocupado_menos_10': y_pred_depto1_menos_10,
                                   'dias_ocupado_menos_20': y_pred_depto1_menos_20})
```

✓ 0.0s

```python
predicciones_dpto1.head()
```

✓ 0.0s

	dias_ocupado_orig_pred	dias_ocupado_mas_10	dias_ocupado_mas_20	dias_ocupado_menos_10	dias_ocupado_menos_20
0	6.015	4.650	4.535	6.270	6.385
1	5.085	4.810	4.790	5.130	5.630
2	3.960	4.270	3.875	3.640	3.955
3	5.465	4.275	4.250	4.485	4.570
4	4.135	4.460	4.250	4.620	4.640

```python
dpto1_final = pd.merge(dpto1.reset_index(), predicciones_dpto1, left_index=True, right_index=True)

# Calculo la ganancia esperada a cada precio
dpto1_final['ganancia_orig_pred'] = dpto1_final['prom_dolar_price']*dpto1_final['dias_ocupado_orig_pred']
dpto1_final['ganancia_mas_10%'] = dpto1_final['price_mas_10%']*dpto1_final['dias_ocupado_mas_10']
dpto1_final['ganancia_mas_10%'] = dpto1_final['price_mas_10%']*dpto1_final['dias_ocupado_mas_10']
dpto1_final['ganancia_mas_20%'] = dpto1_final['price_mas_20%']*dpto1_final['dias_ocupado_mas_20']
dpto1_final['ganancia_menos_10%'] = dpto1_final['price_menos_10%']*dpto1_final['dias_ocupado_menos_10']
dpto1_final['ganancia_menos_20%'] = dpto1_final['price_menos_20%']*dpto1_final['dias_ocupado_menos_20']

# Me quedo con la mayor ganancia
dpto1_final['ganancia_predict']=dpto1_final[['ganancia_orig_pred','ganancia_mas_10%',
                                             'ganancia_mas_20%', 'ganancia_menos_10%',
                                             'ganancia_menos_20%']].max(axis=1)
```

✓ 0.0s

```python
# ganancia original del depto1
ganancia_original1 = dpto1_final.ganancia_original.sum()

# ganancia predicha del depto1
ganancia_pred1 = dpto1_final.ganancia_predict.sum()

# diferencia
diferencia1 = ganancia_pred1-ganancia_original1
diferencia1
```

✓ 0.0s

```
395.65870923269085
```

```python
# cuanto gana de diferencia en promedio por semana si usa el modelo?
dpto1_final['diferencia'] = dpto1_final['ganancia_predict']-dpto1_final['ganancia_original']
dpto1_final.diferencia.mean()
```

✓ 0.0s

```
7.608821331397893
```

Anexo 11 *- Proceso de encontrar la diferencia de ganancia esperada para el Departamento 1.*

```
dptos = pd.concat([dpto1_final, dpto2_final, dpto3_final, dpto4_final])
```
✓ 0.0s

```
# ganancia original de los deptos
ganancia_original = dptos.ganancia_original.sum()

# ganancia predicha de los deptos
ganancia_pred = dptos.ganancia_predict.sum()

# diferencia
diferencia = ganancia_pred-ganancia_original
diferencia
```
✓ 0.0s

```
3079.8314947372055
```

Anexo 12 *- Diferencia de ganancia total esperada.*

```
# cuanto gana de diferencia en promedio por semana si usa el modelo?
dptos['diferencia'] = dptos['ganancia_predict']-dptos['ganancia_original']
dptos.diferencia.mean()
```
✓ 0.0s

```
17.015643617332653
```

Anexo 13 *- Diferencia de ganancia total promedio por semana esperada.*

Anexo ll

El/los autor/es firmante/s autoriza/n al Instituto Tecnológico de Buenos Aires (ITBA) a poner a disposición del público la obra detallada en el presente documento, a solo fin de divulgación de la producción científico-académica de la Universidad. El trabajo será de consulta libre y gratuita en el Repositorio Institucional ITBA, a través de Internet, y en todos aquellos repositorios digitales en los que participe la Universidad. Esta autorización representa la cesión al ITBA, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de los derechos de reproducción, distribución y comunicación pública por cualquier medio o soporte de la obra. Asimismo, se autoriza la transformación de la obra, sin producir cambios en el contenido, siempre que sea necesaria para permitir su preservación y uso en formato electrónico, incluyendo la realización de copias digitales y migraciones de formato necesarios para la seguridad, resguardo y preservación a largo plazo de la misma.

1. Datos del/os Autor/es

Apellido y Nombre: Fiorellino Delfina
DNI: 42587269
Legajo: 60465
E-mail: dfiorellino@itba.edu.ar

Apellido y Nombre: Pruden Valentina
DNI: 42911852
Legajo: 60769
E-mail: vpruden@itba.edu.ar

Apellido y Nombre: Vidal Maria del Rosario
DNI: 42816056
Legajo: 60369
E-mail: marvidal@itba.edu.ar

2. Datos de la obra

Título completo del trabajo: Vadero

Palabras clave: AirBnb, ocupación, precio

Carrera: Analitica Empresarial y Social

3. Tipo de obra:

− Proyecto Final de Grado [x]

4. Autorizo la publicación del:

Texto completo [x]

Dentro de los 6 meses posteriores a su aprobación/presentación [x]

El período de confidencialidad o el secreto del trámite finaliza el: .....

El/los autor/es declara/n que la autorización realizada no infringe derechos de terceros y libera/n al ITBA de todo tipo de responsabilidad (sea civil, administrativa o penal) que pudiera surgir frente a cualquier reclamo o demanda referida a la obra por parte de terceros, asumiendo dicha responsabilidad de forma exclusiva. Acepta/n y toma/n conocimiento de que en caso que la obra sea inédita perderá la condición de tal con su publicación en la web.

Lugar y fecha: Buenos Aires, Argentina; 28 de agosto del 2023

Firma y aclaración del/os autor/es

Vidal, Maria del Rosario

Pruden, Valentina

Fiorellino, Delfina

A ser completado por el Departamento de Grado /Posgrado/Doctorado:

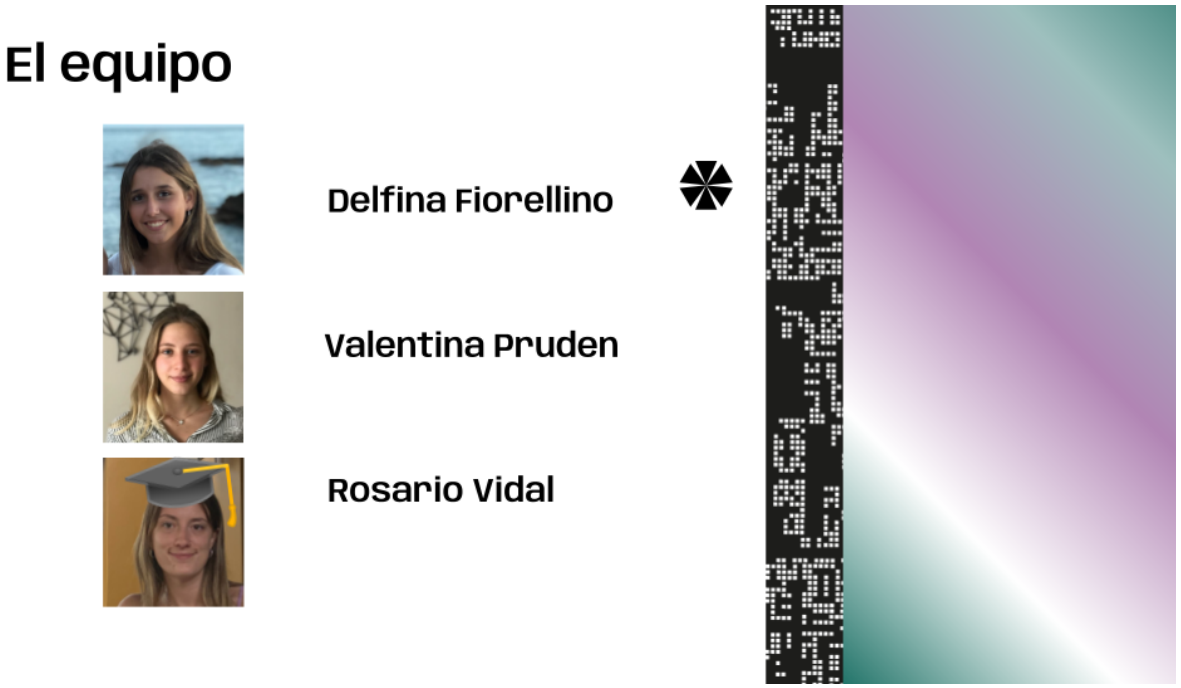
Nro. de Acta: … Calificación: 6,50

Jurado (Apellido, Nombre):
Rodríguez Varela, Juan Pablo
Brottier, Ignacio

Firma y sello
Fecha de defensa/aprobación: 10/07/2023 (Director de Departamento)

Anexo lll

Presentacion final:
https://docs.google.com/presentation/d/1s-dGmCTEr-yIK-oIZ3sHhh7B8SdG_2jiho-NbcxbOwM/edit?usp=sharing

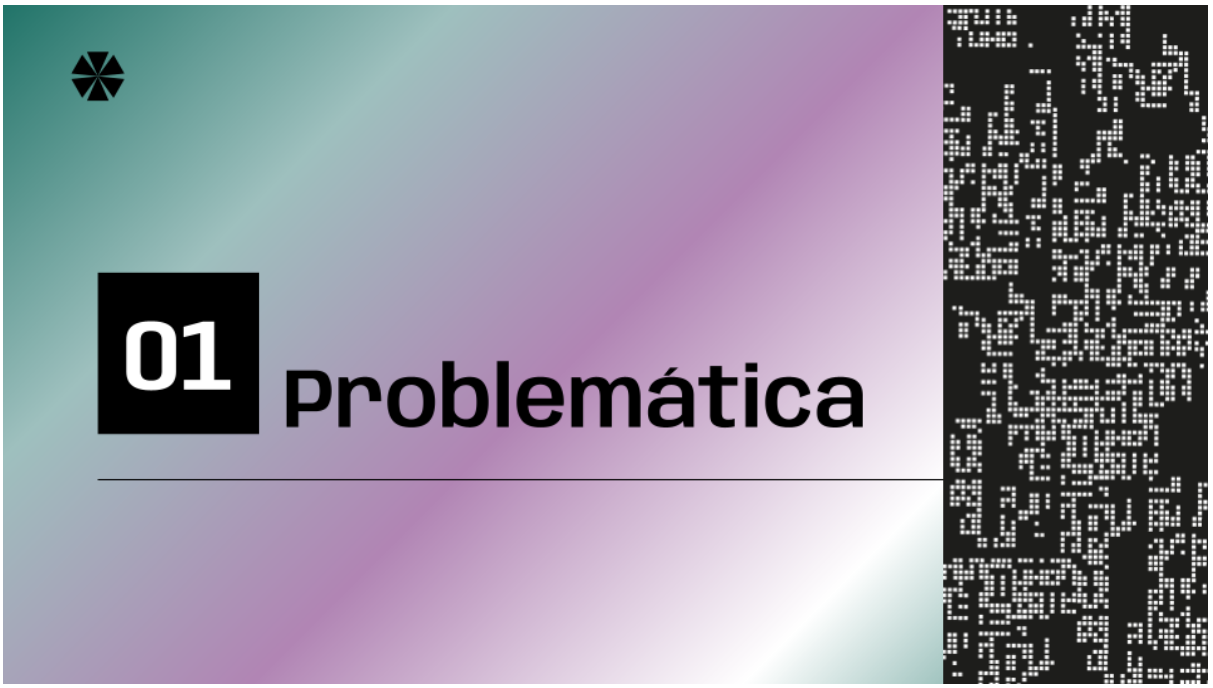


CALENDARIO

01 Problemática

02 Solución

03 Pasos a futuro



Anfitriones reportan una disminución de reservaciones en Airbnb, pero ¿qué dicen los huéspedes?

"La quiebra de Airbnb está llegando"

Airbnb: cómo ahorré $205 USD – trucos casi inmorales

¿Cuánto cobra Airbnb a los anfitriones?

- Cantidad de departamentos: 4
- Localidad de los departamentos: Belgrano
- Inicio actividad: 2017
- ***Precio elegido: el que estipula AirBnb***

Marcelo

Solución 02

Nuestra propuesta

PROBLEMA

Marcelo fija precios por noche en base a los sugeridos

SOLUCIÓN

Predecir cantidad de días que un alojamiento va a estar ocupado en una semana del año

RESULTADO

Selección del mejor precio por noche

DATOS

- Listings
- Calendar (período de 1 año)

ETL

EDA

AI

APP

DATOS

- Listings
- Calendar (periodo de 1 año)

ETL

EDA

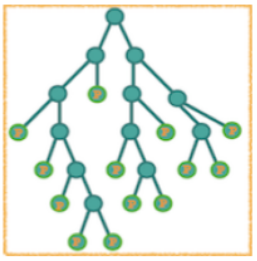
AI

APP

AI

Clustering

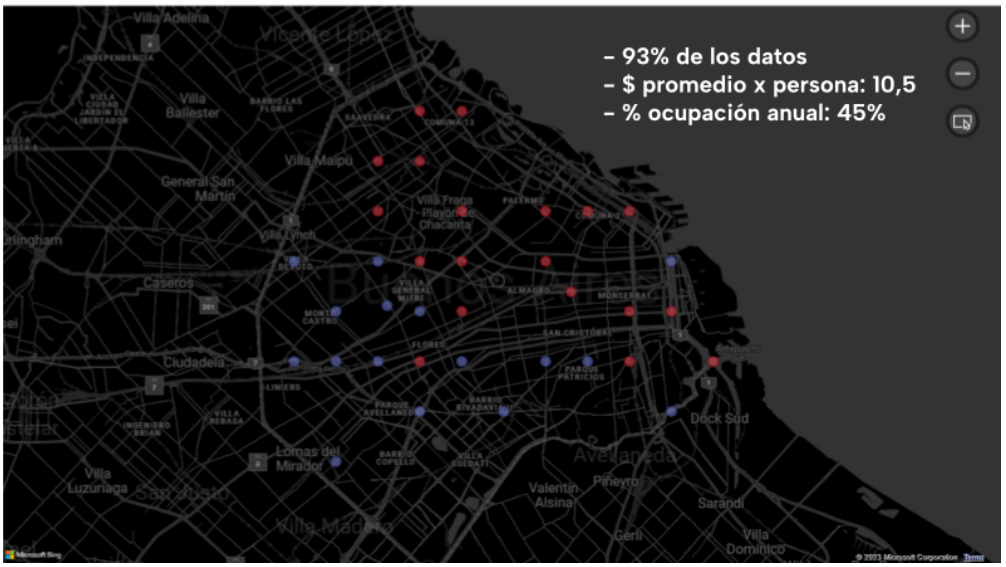
Metodología utilizada para agrupar y acotar los barrios de interés.



Random Forest

Predecir días de ocupación de alojamientos en distintas semanas del año.

Clustering



Random Forest

1. Porcentaje de ocupación en el último año
2. Precio promedio por noche
3. Semana
4. Puntaje de reseñas
5. Mes de alta ocupación
6. Huéspedes
7. Temporada alta
8. Barrio
9. Tipo de estadía

Cómo se evaluó el modelo

Una medida estadística que indica qué tan bien se ajusta un modelo a los datos observados.

R^2

68%

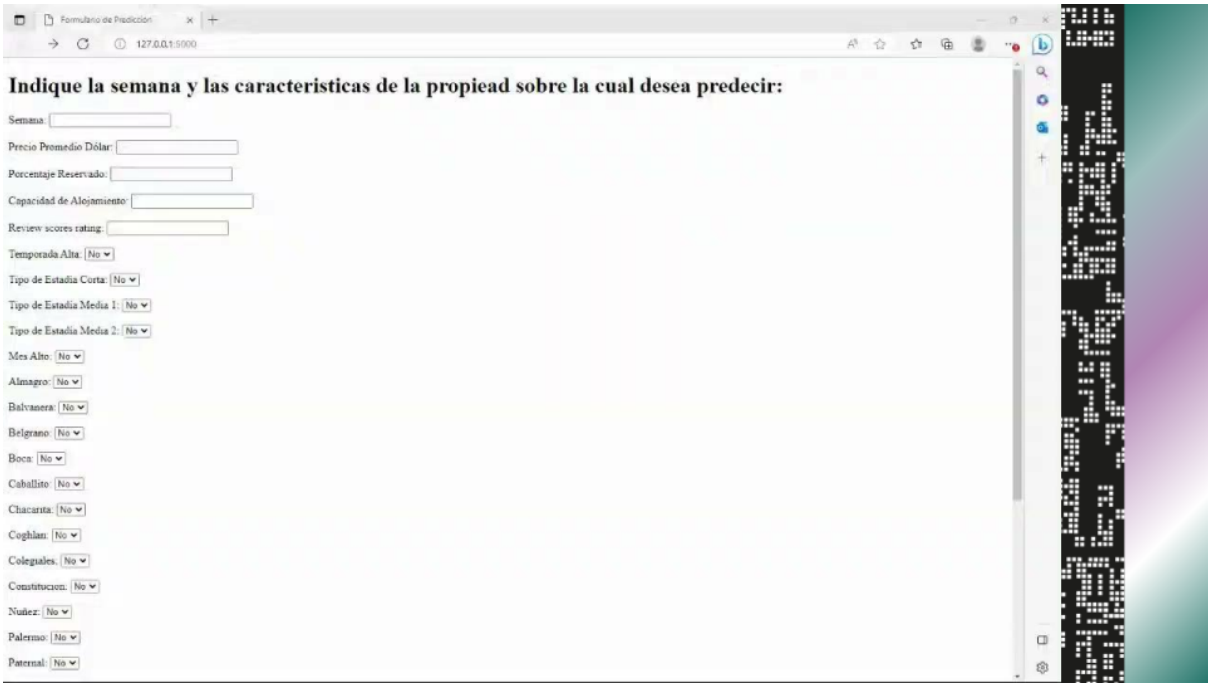
Una medida estadística que indica la raíz del error cuadrático medio.

RMSE

1,81

¡Acceso rápido y fácil al alcance de tu mano!

¿CÓMO?



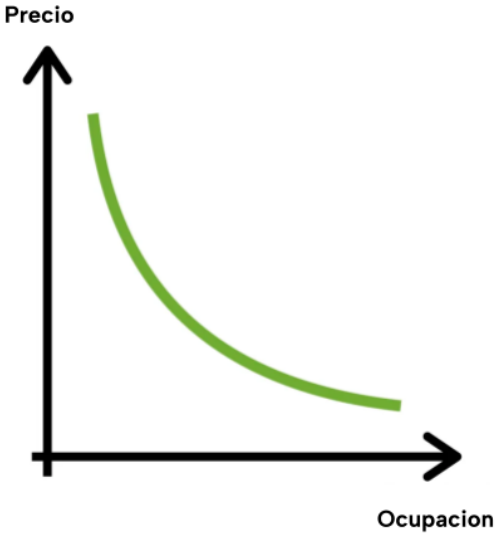
¿Cuál es el Impacto?

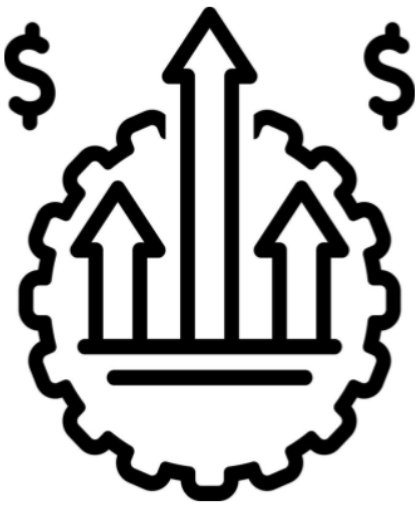
	Alojamiento 1	Alojamiento 2	Alojamiento 3	Alojamiento 4
Cantidad de semanas	52	52	52	25
Diferencia de ganancia total (dólares)	$396	$1017	$1568	$99
Promedio de diferencia de ganancia por semana (dólares)	$8	$20	$30	$4

Ganancia esperada con la aplicación último año - Ganancia real último año = $3080

03 Próximos pasos

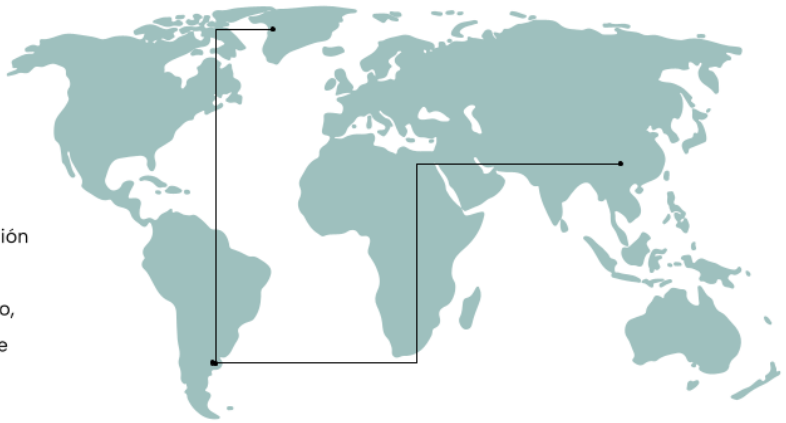
Nuevas funcionalidades





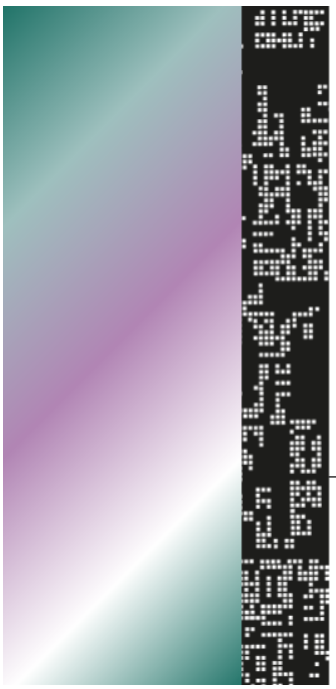
Precio optimo

Globalización



Finalizada la implementación de la aplicación a los departamentos de Marcelo, expandir su uso al resto de usuarios de AirBnb.

Eventos



Gracias!

¿Preguntas?

Bibliografía

- Martins, D. (23 de marzo de 2023). *XGBoost: A complete guide to fine-tune and optimize your model*. Medium. https://towardsdatascience.com/xgboost-fine-tune-and-optimize-your-model-23d996fab663
- Matthew Stewart, P. (10 de febrero de 2023). *Simple guide to hyperparameter tuning in Neural Networks*. Medium. https://towardsdatascience.com/simple-guide-to-hyperparameter-tuning-in-neural-networks-3fe03dad8594
- Krishnan, S. (8 de septiembre de 2021). *How do determine the number of layers and neurons in the hidden layer?*. Medium. https://medium.com/geekculture/introduction-to-neural-network-2f8b8221fbd3