Trabajo Práctico N° 1 - Grupo 02

Indice

Introducción

El presente trabajo se centra en el análisis y la predicción de precios de propiedades inmobiliarias en la Ciudad Autónoma de Buenos Aires (CABA). Utilizando un dataset extenso con 460154 registros y 20 columnas, se busca desarrollar modelos predictivos que permitan estimar el precio de venta de propiedades basándose en diversas características.

El análisis comienza con una filtración inicial del dataset, enfocándose exclusivamente en las propiedades ubicadas en CABA, que son de tipo casa, PH (propiedad horizontal) o departamento, y que están a la venta en USD. Esta filtración reduce el número de registros a 94249, lo que representa una disminución significativa en el tamaño del conjunto de datos original.

Una vez filtrado el dataset, se realiza un análisis exploratorio de datos para comprender mejor las características y relaciones dentro de los datos. Esta fase incluye la identificación de outliers, el manejo de valores nulos y la eliminación de columnas irrelevantes. Además, se visualizan las distribuciones de las variables más importantes y se analiza la correlación entre diferentes atributos.

El siguiente paso consiste en dividir el dataset en conjuntos de entrenamiento y prueba, utilizando una proporción de 80% y 20% respectivamente, con el objetivo de construir y evaluar modelos predictivos. El análisis se complementa con técnicas de preprocesamiento de datos, como la imputación de valores faltantes y la normalización de variables, para asegurar que los datos estén en la mejor forma posible para el entrenamiento de los modelos.

El trabajo culmina con la construcción de modelos de machine learning para la clasificación y regresión, con el propósito de predecir el precio de las propiedades y entender mejor los factores que influyen en su valor en el mercado inmobiliario de CABA.

Desarrollo

Análisis Exploratorio

Inicialmente el dataset tiene un total de 20 columnas y 460154 registros, los cuales representan propiedades en venta o en alquiler. Dichas propiedades tienen todos sus datos relevantes como ubicación y tipo de propiedad, precio, entre otros.
Para este análisis se tomaron solamente las propiedades que se ubican en CABA, son de tipo casa, ph o departamento y están a la venta en USD.

Una vez que el dataset fue filtrado con los criterios mencionados este pasó a tener un total de 94249 registros (disminuyó casi por un factor de 5).

Luego del filtrado de datos de interés y de llevar a cabo una exploración inicial de la cual se hablará en esta sección.

Exploracion Inicial

En esta etapa del análisis se hizo un vistazo de los datos para encontrar correlaciones entre atributos, ver la forma de los mismos con medidas de resumen, y una primera visualización de estos (utilizando el dataset filtrado).

Primeramente se observaron distintas medidas a modo de resumen de las variables cuantitativas del dataset, donde se puede ver por ejemplo la desviación estándar de la latitud y longitud la cual es mínima ya que todas las propiedades están en CABA, así como también se pueden identificar outliers (aunque estos serán analizados más profundamente en la sección de datos atípicos) en los atributos de “rooms” y “bedrooms” ya que la moda es 3 y 1 respectivamente pero tienen valores máximos de 36 y 32, entre otras observaciones.

Luego se indagaron los valores posibles y la frecuencia de los mismos para cada variable cualitativa (se utilizaron visualizaciones de barras para mostrar estos datos). Es en esta parte que se descubrió que las variables “place_l5” y “place_l6” tienen 0 valores posibles es decir que son columnas nulas, que hacer con estas se determinará más adelante.

También en esta etapa se hizo un análisis de cuáles columnas son irrelevantes para el propósito de este trabajo y cuáles no. En base a este análisis se tomó la decisión de dejar de contar con las siguientes columnas:

- Place_l2 (ya que siempre será CABA)
- Place_l5/l6 (100% nulas)
- Operation (todos son Venta)
- Property_currency (todos son USD)
- Property_title (no tiene importancia)

Se exploró la distribución de las variables más relevantes del dataset a través de visualizaciones univariadas. Estas son, un histograma de distribución del precio, un gráfico de barras que muestra cuales son los puntos del año con más ventas y dos boxplots para las variables “room” y “bedroom” (donde se pueden ver los outliers previamente identificados).

Finalmente se le dio cierre a la sección introductoria de exploración inicial haciendo un análisis de la correlación entre diferentes variables. Para empezar este análisis se usó un pairplot entre algunas variables seleccionadas, este muestra algunas relaciones entre variables como también muestra la completa independencia entre algunas otras variables. Se analizan más profundamente las relaciones entre superficie total y superficie cubierta, precio y cantidad de ambientes/habitaciones, y entre “rooms” y “bedrooms”, ya que son las que a simple vista parecen tener relación; Para este análisis se calculó la covarianza y correlación, y se obtuvo que la correlación más alta está entre “rooms” y “bedrooms”, seguida de la correlación entre surface_total y surface_covered, y las más bajas entre precio y superficie cubierta/total.

Preprocesamiento de Datos

Esta parte está comprendida por dos pilares, el análisis de los datos nulos/faltantes y el análisis de los datos atípicos/outliers. A lo largo de este análisis se irá modificando el dataset con el que se está trabajando de manera tal que sea más útil a la hora de entrenar un modelo predictivo del precio de las propiedades. Estas modificaciones serán, imputación de datos, modificación de registros por datos mal ingresados, y de ser necesario, eliminación de algunos registros.

Datos faltantes

En esta sección, se llevó a cabo un análisis exhaustivo de los datos faltantes en el conjunto de datos. Inicialmente, se identificaron las variables (columnas) que contenían valores nulos.

Se observó que las variables place_l6 y place_l5 no contienen datos en absoluto, lo que las hace irrelevantes para el conjunto de datos. Asimismo, la variable place_l4 presenta un 96% de datos nulos, lo que la sitúa en la misma categoría que las variables anteriores. Por lo tanto, se tomó la decisión de eliminarlas tanto del conjunto de entrenamiento como del de prueba.

Posteriormente, se analizaron las demás variables que presentaban valores nulos, como property_bedrooms, property_surface_total, latitud, longitud, property_surface_covered, property_rooms y place_l3. Se hicieron visualizaciones de la cantidad de valores nulos y no nulos en cada una de ellas. Además, se investigó si el precio de las propiedades también contenía datos nulos.

Se abordaron los datos faltantes o mal ingresados en el dataset comenzando con la imputación Cold Deck para la variable place_l3, que contiene información sobre los barrios de las propiedades. Para completar esta información fue de utilidad el archivo barrios.csv proporcionado por el Gobierno de la Ciudad de Buenos Aires.

Para las demás variables, se realizó un análisis cuantitativo de los datos. Dado que la presencia de valores negativos o cero en estas variables carece de sentido, fueron asignados como valores NaN.

Además, se verificó la presencia de datos duplicados en el dataset y se confirmó que no había ninguno.

Luego, se aplicó la imputación MICE (Multiple Imputation by Chained Equations) a las variables Property_surface_covered, Property_rooms, Property_bedrooms y Property_surface_total. Esta técnica permitió completar los datos faltantes en el conjunto de entrenamiento.

Finalmente, se compararon las distribuciones del dataset antes y después de la imputación. Se identificaron los valores que presentaban la máxima cantidad de datos para cada variable que fue completada y se analizaron los gráficos de distribución resultantes.

Valores atipicos

ANÁLISIS UNIVARIADO

Se buscó los valores atípicos en estas variables:

- property_bedrooms
- property_rooms
- property_price
- property_surface_covered
- property_surface_total

Se buscó valores atípicos según el índice de rango intercuartil y se graficó con boxplots con el campo en general.

De estos, se identificaron outliers extremos mediante la observación de puntos muy aislados en los boxplots de las variables analizadas. Estos puntos, al estar significativamente separados del resto de los datos en el gráfico, indicaron valores que pudieron influir de manera considerable en los resultados.

Al realizar la inspección de los datos con estas características, se identificaron registros que presentaban información incoherente o improbable en el contexto de la venta de inmuebles y/o el diseño de una propiedad habitable, sugiriendo posibles errores de ingreso de datos o inconsistencias en la recolección de información. Algunos ejemplos de estos errores y su correspondiente corrección y análisis:

Propiedades con > 40 habitaciones	Se eliminaron los registros.
Propiedades con > 15 habitaciones	En los casos donde el número de ambientes era < 10, se observó un error decimal, corregido.
Propiedades con precio > 8.000.000	Se buscó referencias en sitios de venta de propiedades (Zonaprop, Remax, Argenprop, Properati). No se observaron propiedades de esas dimensiones y/o en esos barrios con ese precio en dólares estadounidenses. Realizar una conversión (en el caso de haber un error y ser un precio expresado en ARS, lo cual sería raro en el mercado) o realizar una reducción de un décimo, no parecerían adecuar los precios. No se realizaron modificaciones.
Superficie cubierta < 10m2	Si bien según el código urbanístico de la Ciudad de Buenos Aires no se permiten departamentos tan pequeños, esto puede infringirse. Bajo estos supuestos, y viendo los registros que aparecen, se reemplazó la superficie cubierta por la superficie total.
Superficie cubierta > 15000	Se realizó corrección de unidad (división por 1000)
Superficie cubierta > 4000	Se realizó corrección de unidad (división por 100)
Superficie cubierta > 1100	En los casos donde el número de habitaciones estaba dentro de los valores normales (<= 5), se realizó una corrección logarítmica.

A su vez, considerando la naturaleza del problema, se realizaron boxplots según barrios. Al observar estos boxplots se notó la diferencia que existía al realizar esta simple consideración. Si se ponen estos datos en contexto, todos saben que no serían los mismos los resultados del análisis si, por ejemplo, solo se limitaran a analizar las zonas sur de la Ciudad de Buenos Aires. Desde las medianas en tamaños de superficie cubierta, hasta los precios y el tiempo de publicación, cada barrio tiene tendencias específicas que, si bien colaboran en el resultado final, influyen mucho en el precio. Teniendo en cuenta que el objetivo final de este problema es poder predecir el precio de propiedades que serán lanzadas en venta en esta ciudad, no se quisieron eliminar outliers que realmente no lo eran si se tenía en cuenta el barrio en el que se encontraban.

Teniendo en cuenta esto, se determinaron para cada variable límites que fueran outliers extremos para todos los barrios. Se realizó una corrección logarítmica sobre

las variables (únicamente) que superaron esos límites, los cuales se detallan a continuación:

Variable	**Limite**
property_bedrooms	6
property_rooms	10
property_surface_total	3000

Se procedió a analizar el Z Score, normal y modificado, utilizando como límite de z-score 3.5 y 5. Los límites no fueron lo suficientemente claros para determinar qué valores eran realmente atípicos y necesitaban ser modificados. Ante la falta de claridad en los resultados, se optó por no realizar cambios en los datos debido a la preocupación de no alterar los datos de manera innecesaria sin una comprensión clara de cuáles valores son verdaderamente atípicos. Se postergó cualquier modificación al análisis multivariado más exhaustivo que permitiera comprender mejor cómo podrían influir en el conjunto estos datos.

Es cierto que en lo que llamaron “análisis univariado” muchas veces se tomaron en cuenta otras variables antes de realizar una modificación, pero siempre fue con las herramientas de análisis univariado. Se dejó el análisis LOF / isolation forest para lo que no es tan visible a simple vista.

ANÁLISIS MULTIVARIADO

En primer lugar, se realizó un análisis de isolation forest. Revisando las anomalías y viendo los scatterplot que resultaron de éste modelo, en su mayoría parecerían seguir las tendencias de los registros no considerados anómalos pero con valores levemente inferiores/superiores a la mediana.. Parecen ser anomalías genuinas y decidimos mantenerlas por contexto. Sabemos que existe un riesgo inherente en mantener los datos como están pero de detectarse en el entrenamiento que los resultados no son los indicados se revisará esta sección.

LOF mostró algunos problemas para identificar outliers en análisis multivariados al no filtrarse los datos según tipo o barrio. En el caso específico de contraponer (property_surface_total, property_surface_covered) y (property_price,

property_surface_total), al inspeccionar las anomalías determinadas por LOF, se encuentran anomalías extrañas que se normalizan con MinMax para poder analizarlas nuevamente. Luego de la normalización se vuelve a analizar con LOF pero no se encuentran mejoras notables en los resultados.

Se estableció n_neighbors = 30 para el caso general y n=10 para los casos por tipo de propiedad. Para ciertas variables, había un tipo de propiedad que resaltaba en cantidad de outliers con respecto a la media general (por ejemplo, las casas tenían precios demasiado altos, los PHs con respecto a la superficie total, etc). Igualmente, concluyeron que eran casos razonables por la naturaleza del problema y no se descartaron estos registros.

Visualizaciones

La visualización de datos es el proceso de representar información de manera gráfica y visual para facilitar su comprensión y análisis. Se trata de transformar datos numéricos, textuales o cualitativos en elementos visuales, como gráficos, diagramas o mapas, que puedan ser interpretados fácilmente por los usuarios.

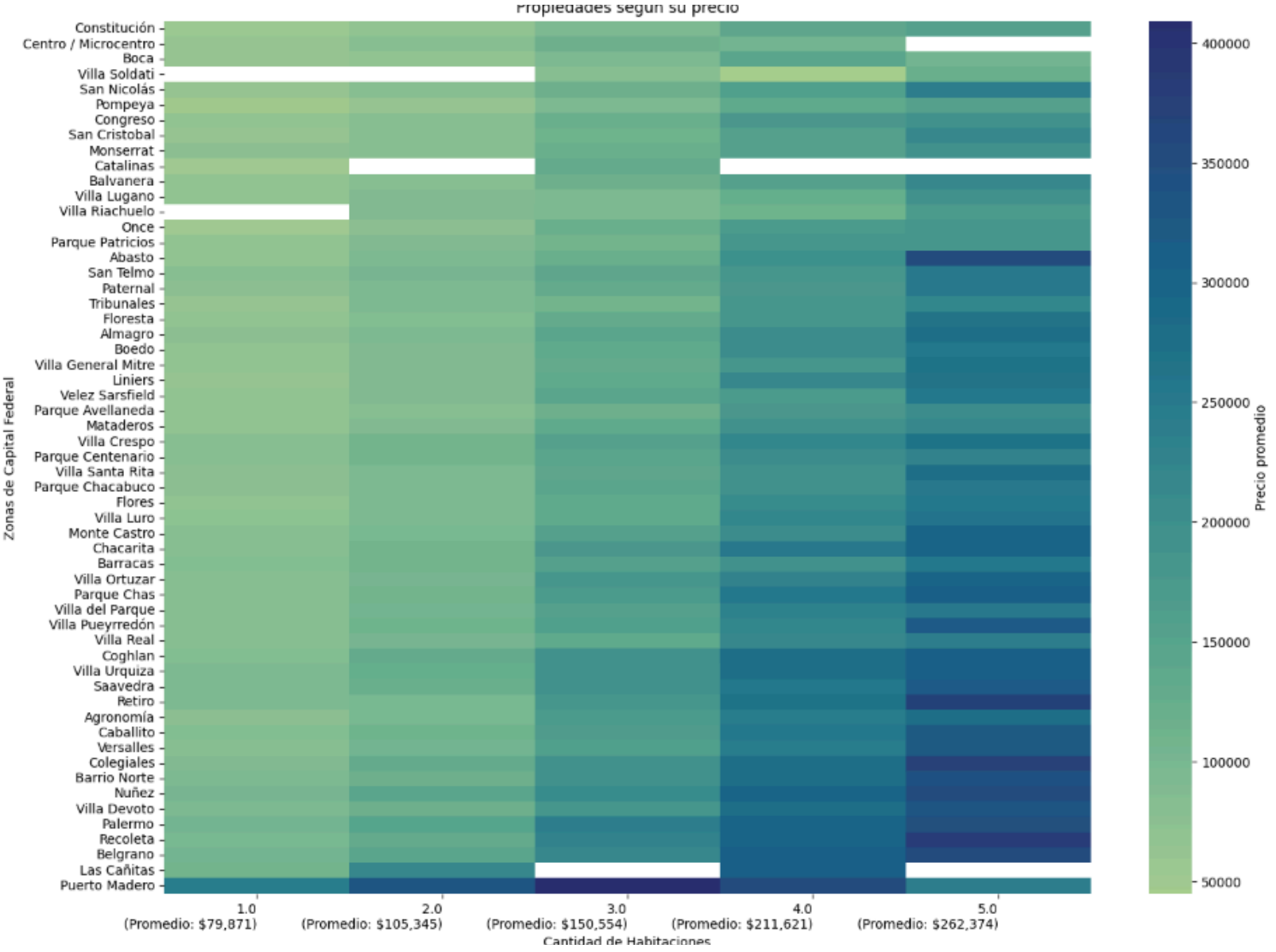
Se hicieron visualizaciones por varias razones:

Comprensión de los datos: Las visualizaciones nos permiten comprender rápidamente grandes volúmenes de datos y detectar patrones, tendencias o relaciones que podrían pasar desapercibidos en una tabla de números.

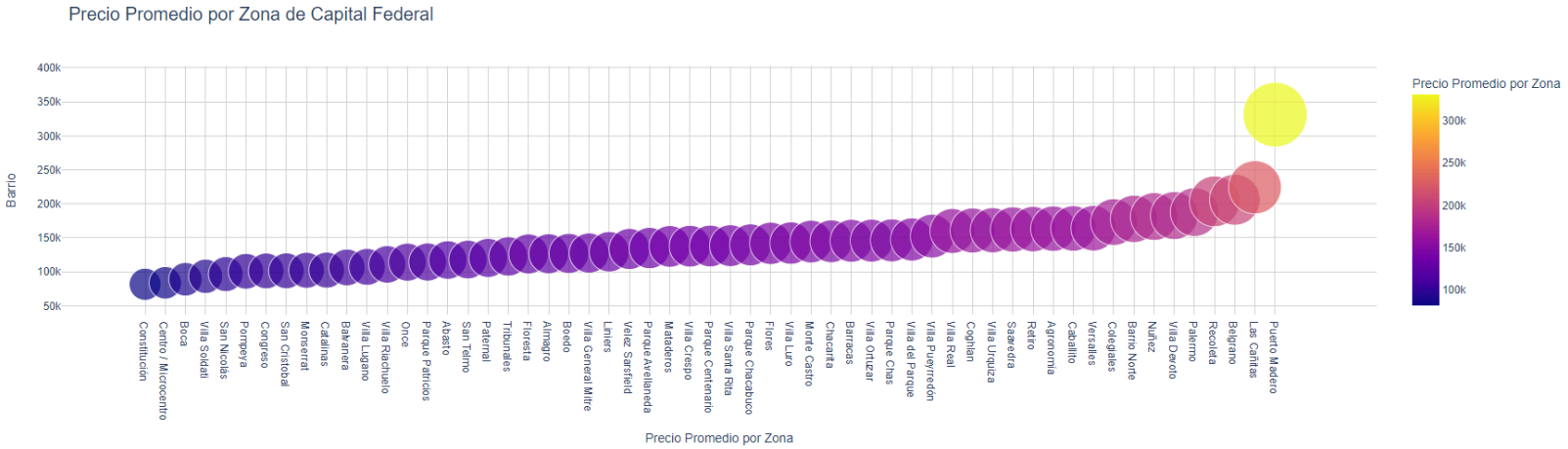
Comunicación efectiva: Las visualizaciones son una herramienta poderosa para comunicar información de manera efectiva a una audiencia. Son más intuitivas y fáciles de entender que los datos en bruto, lo que las hace ideales para presentaciones, informes o publicaciones.

Toma de decisiones: Las visualizaciones ayudan a respaldar la toma de decisiones informadas al proporcionar una representación clara y concisa de la información relevante. Permiten identificar oportunidades, riesgos o áreas de mejora.

Exploración de datos: Las visualizaciones nos permiten explorar datos desde diferentes perspectivas y niveles de detalle. Podemos interactuar con gráficos interactivos para profundizar en los detalles o ampliar nuestra comprensión.

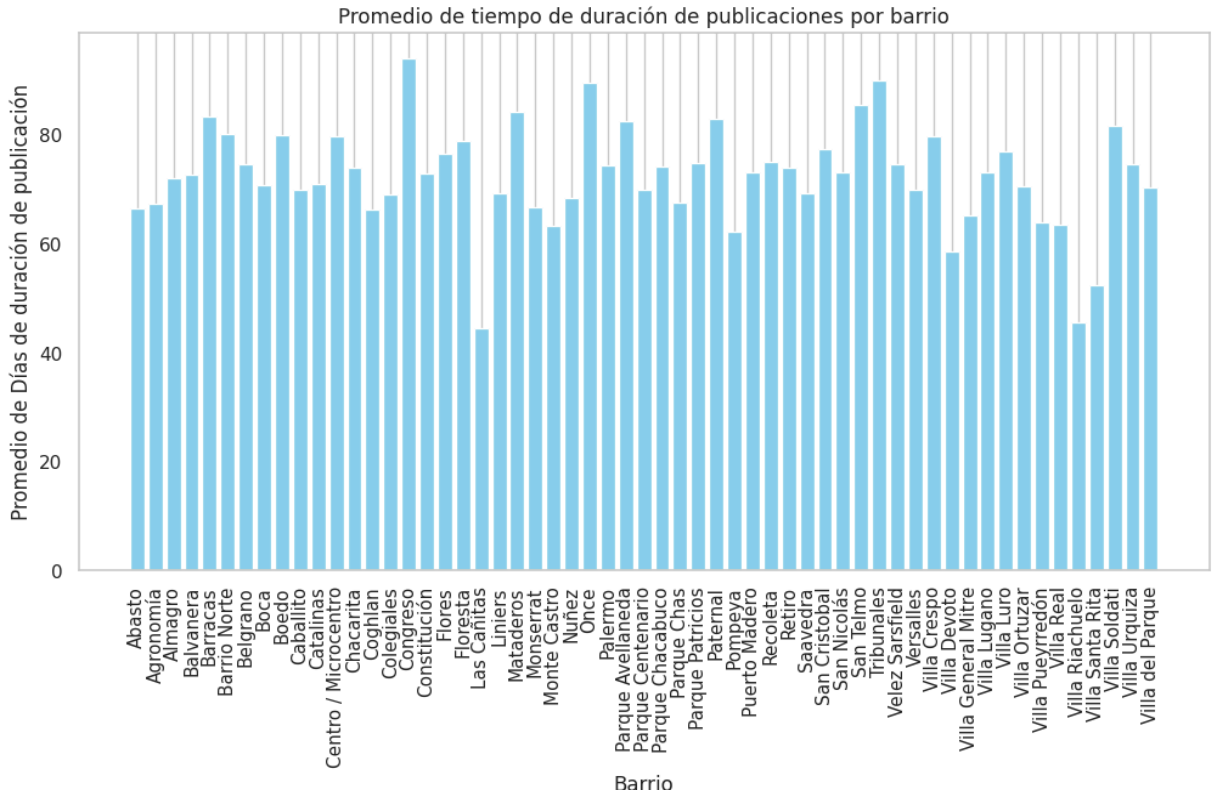

Propiedades según su precio
Zonas de Capital Federal
Constitución
Centro / Microcentro
Boca
Villa Soldati
San Nicolás
Pompeya
Congreso
San Cristobal
Monserrat
Catalinas
Balvanera
Villa Lugano
Villa Riachuelo
Once
Parque Patricios
Abasto
San Telmo
Paternal
Tribunales
Floresta
Almagro
Boedo
Villa General Mitre
Liniers
Velez Sarsfield
Parque Avellaneda
Mataderos
Villa Crespo
Parque Centenario
Villa Santa Rita
Parque Chacabuco
Flores
Villa Luro
Monte Castro
Chacarita
Barracas
Villa Ortuzar
Parque Chas
Villa del Parque
Villa Pueyrredón
Villa Real
Coghlan
Villa Urquiza
Saavedra
Retiro
Agronomía
Caballito
Versalles
Colegiales
Barrio Norte
Nuñez
Villa Devoto
Palermo
Recoleta
Belgrano
Las Cañitas
Puerto Madero
1.0
(Promedio: $79,871)
2.0
(Promedio: $105,345)
3.0
(Promedio: $150,554)
4.0
(Promedio: $211,621)
5.0
(Promedio: $262,374)
Cantidad de Habitaciones
Precio promedio
400000
350000
300000
250000
200000
150000
100000
50000

La visualización presentada fue un mapa de calor (heatmap) que mostraba el precio promedio de las propiedades en función de la cantidad de habitaciones y su ubicación en diferentes zonas de la Ciudad Autónoma de Buenos Aires (CABA). Al observar el heatmap, fue posible identificar patrones en los precios según la ubicación y el tamaño de la propiedad. Por ejemplo, había áreas donde las propiedades con más habitaciones tendían a tener precios más altos, mientras que en otras zonas los precios eran más uniformes independientemente del tamaño de la propiedad. La visualización proporcionó una manera efectiva de entender cómo se relacionaban el precio promedio de las propiedades, la cantidad de habitaciones y la ubicación en diferentes áreas de la Ciudad Autónoma de Buenos Aires. Ayudó a los interesados a tomar decisiones informadas sobre la compra o inversión en propiedades en la ciudad. Cabe destacar que los precios estaban ordenados, lo que facilitó la percepción de los degradés de colores en el heatmap. Sin embargo, es importante señalar que los valores nulos ("los espacios en blanco en la visualización") aún persistieron en el conjunto de datos, ya que esta sección fue previa a la sección de datos faltantes.



Esta visualización mostró un gráfico de burbujas agrupadas que representaba el precio promedio de las propiedades en cada barrio de la zona de Capital Federal. Cada burbuja representaba un barrio, y el tamaño de la burbuja indicaba el precio promedio de las propiedades en ese barrio. Los colores de las burbujas indican diferentes rangos de precios para facilitar la identificación visual: los tonos más oscuros representaban barrios con precios promedio más bajos, mientras que los tonos más claros representaban barrios con precios promedio más altos. El gráfico de burbujas agrupadas era interactivo, lo que permitió a los usuarios explorar los datos con mayor detalle al pasar el cursor sobre las burbujas para ver información adicional, como el nombre del barrio y el precio promedio. Esta representación visual

proporcionó una manera intuitiva de comprender la distribución de precios en los diferentes barrios de la zona de Capital Federal, después de haberse realizado todas las correcciones sugeridas por el corrector, llegando a la conclusión de que era una variación de un heatmap.



Esta visualización presentó un gráfico de barras que mostraba el promedio de tiempo de duración de las publicaciones por barrio en la Ciudad Autónoma de Buenos Aires (CABA). Cada barra en el gráfico representaba un barrio específico, y su altura indicaba el promedio de días que las publicaciones permanecieron activas en ese barrio. La visualización permitió comparar fácilmente la duración promedio de las publicaciones entre diferentes barrios de la ciudad. Las barras más altas indicaron que las publicaciones en esos barrios tendieron a permanecer activas por más tiempo, mientras que las barras más cortas indicaron una duración promedio más corta de las publicaciones. Los barrios con barras más altas podrían indicar áreas de la ciudad donde la demanda de propiedades era más alta o donde las propiedades tendían a permanecer en el mercado por períodos más prolongados. Esto pudo haber sido útil para comprender las dinámicas del mercado inmobiliario en diferentes áreas de la ciudad.

Clustering

Análisis de la tendencia al clustering del dataset

Primeramente se eliminaron variables que no sirven para el clustering como id, variables de fechas y luego se observó la agrupación natural de los datos mediante gráficos y se procesaron los datos haciendo una regresión logística de la columna property_type, encodeando los barrios y normalizando las variables numéricas por medio de Min-Max.

Estimación de la cantidad apropiada de grupos

Primeramente se utilizó K-means varias veces con diferentes cantidades de grupos guardando el SSD (sum of squared distances) de cada uno. Este ssd fue utilizado para analizar la cantidad óptima de grupos mediante el método del codo (Elbow method). Dicho método dio como resultado que el mejor K estaría entre 2 y 5, y para elegir el mejor de estos es que se calcula el silhouette score de cada uno y se lo grafica. El K con el mayor score fue K=3 con un total de 0.632, sin embargo los datos no estaban bien balanceados y había un cluster con muchas más observaciones que el otro. Es por esto que al evaluar una tabla con la información obtenida:

- k = 2, score: 0.59, division: 48k/23k
- k = 3, score: 0.632, division: 33k/28k/10k
- k = 4, score: 0.62, division: 24k/21k/16k/9k
- k = 5, score: 0.629, division: 24k/21k/16k/6k/2k

Se decidió como mejor opción el K=4, y es con el que se trabajó.

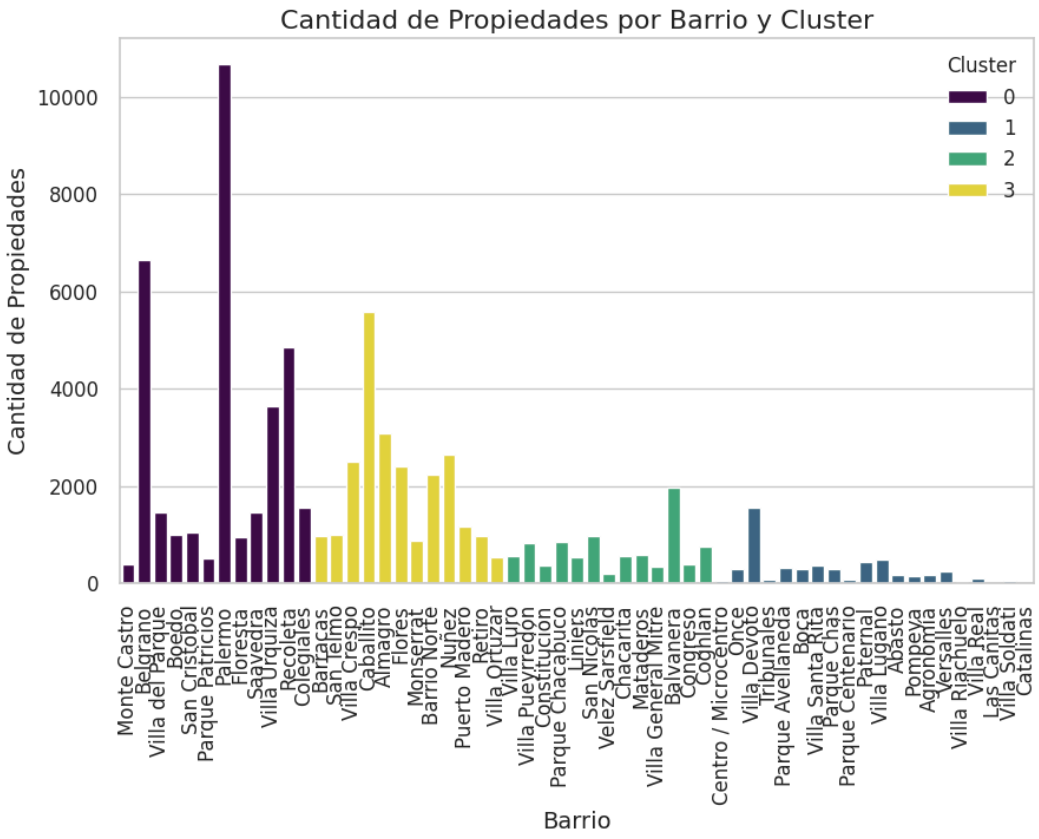
Análisis de grupos

Se examinaron a detalle todas las variables utilizando distintas visualizaciones y mostrando cómo cada una impactó a la creación de los clusters. Más específicamente se observó si la agrupación sucedió:

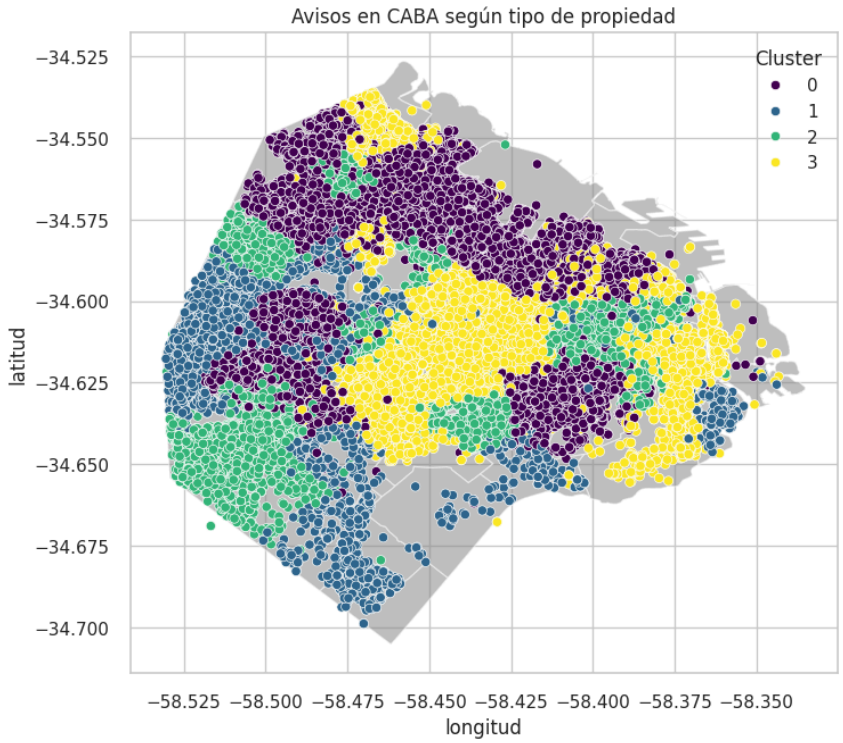
- Por cercania geografica
- Por barrio
- Por tipo de propiedad
- Por precio

- Por cantidad de habitaciones/ambientes
- Por superficie

Se concluyó que hay una clara división geográfica entre clusters, y esta se dio debido a los barrios. Cada barrio pertenece a un y solo un cluster, es decir no hay intersecciones, lo que quiere decir que es una agrupación perfecta en función de dicha variable (ver el gráfico de abajo). Adicionalmente se intentó ver si alguna otra variable también tuvo algún grado de influencia pero dicha hipótesis se descartó luego de analizar las visualizaciones de las otras variables.



Gráficos de los clusters en el mapa de CABA



Análisis con tres grupos

En esta sección se hizo otro análisis de grupos pero esta vez utilizando el algoritmo K-means con un K=3. A su vez esta vez se eliminó la columna de barrios ya que de lo contrario este análisis terminaba dando un resultado muy parecido al anterior.

Primeramente luego de clusterizar se hizo el mismo gráfico mostrado en el anterior análisis, pero esta vez se observó que ya no hay agrupación geográfica alguna. Por lo tanto se procedió a analizar variable por variable el impacto de cada una, y en base a esto se encontró que nuevamente ocurrió una agrupación perfecta pero esta vez en base al tipo de propiedades, es decir, se crearon tres clusters cada uno conteniendo el 100% de cada tipo de propiedad y sin intersecciones.

Entonces se concluye que esta clusterización no fue muy útil ya que las propiedades ya estaban 'agrupadas' por su tipo previamente.

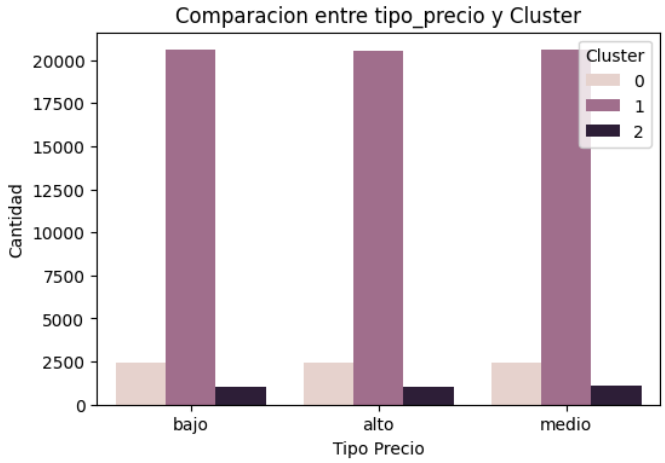
Clasificación

Esta sección se abre haciendo una construcción un target dataset, para el cual se creó una nueva feature llamada “tipo_precio” que se distribuye en “bajo”, “medio” o “alto”. Para la asignación de una categoría de precio a cada propiedad se evaluaron tres alternativas que tienen en cuenta el precio por metro cuadrado (pxm2) de las propiedades. Estas fueron:

- División del pxm2 equitativa entre las tres categorías.
- División de pxm2 de 25% - 50% - 25% entre las tres categorías.
- División de pxm2 de 25% - 50% - 25% relativo a cada tipo de propiedad, entre las tres categorías.
- División de pxm2 equitativa relativa a cada tipo de propiedad, entre las tres categorías.

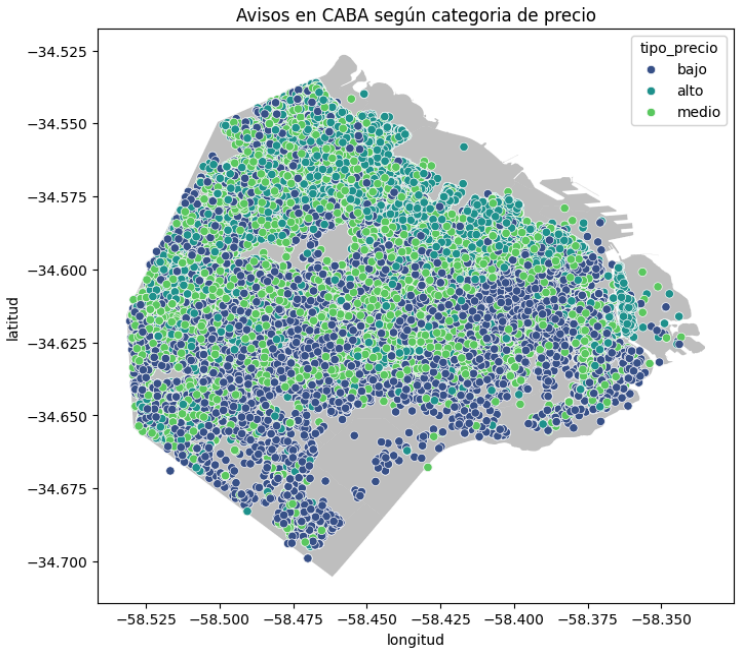
Se decidió que el mejor método fue el tercero ya que tiene una distribución más justa (no simplemente arbitrariamente equitativa) y a su vez tiene en cuenta que por ejemplo puede ocurrir que un precio X sea barato para un PH pero caro para un Departamento. A su vez utilizamos la división equitativa para que todos los tipos de precio tengan la misma representación en el dataset.

Luego se compararon las variables de tipo precio con los clusters mencionados anteriormente cuando el K=3, y se ve cómo se distribuyen los tipos de propiedad en cada categoría de precio.



- Cluster 0: PHs
- Cluster 1: Departamentos
- Cluster 2: Casas

Finalmente antes de comenzar con la sección de entrenamiento y predicción, se observó mediante un gráfico en el mapa de CABA que hay mayor densidad de precios baratos en el sur de la ciudad, y este aumenta a medida que se va más al norte, siendo que en el borde norte hay más densidad de precios altos.



Ingeniería de características

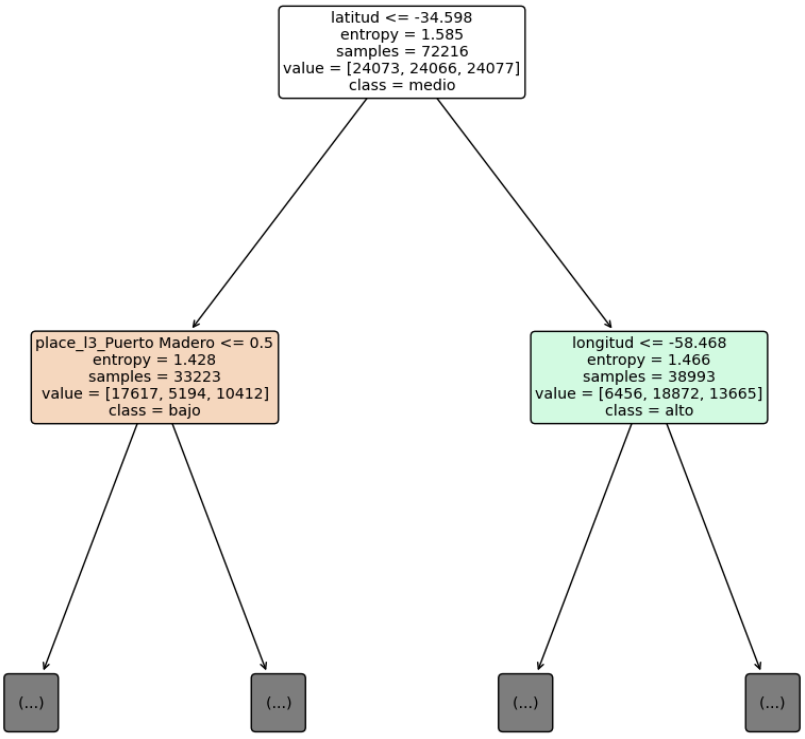
En esta sección se eliminaron las columnas que den información sobre el precio de las propiedades ya que se quiere predecir la categoría de precio a partir del resto de columnas. También se eliminaron todas las columnas consideradas irrelevantes como ID, fechas y una de las columnas con una alta correlación entre sí (por ejemplo, rooms y bedrooms). Luego se procedió a encodear las variables cualitativas (para place_l3 y property_type se utilizó one hot encoding y tipo_precio se representó con un número 0, 1, 2).

Concluye este proceso con un dataset de 65 columnas, place_l3 (one hot encoded), property_type (one hot encoded), property_rooms, property_surface_total, tipo_precio adecuadamente procesadas.

a. Construcción del modelo

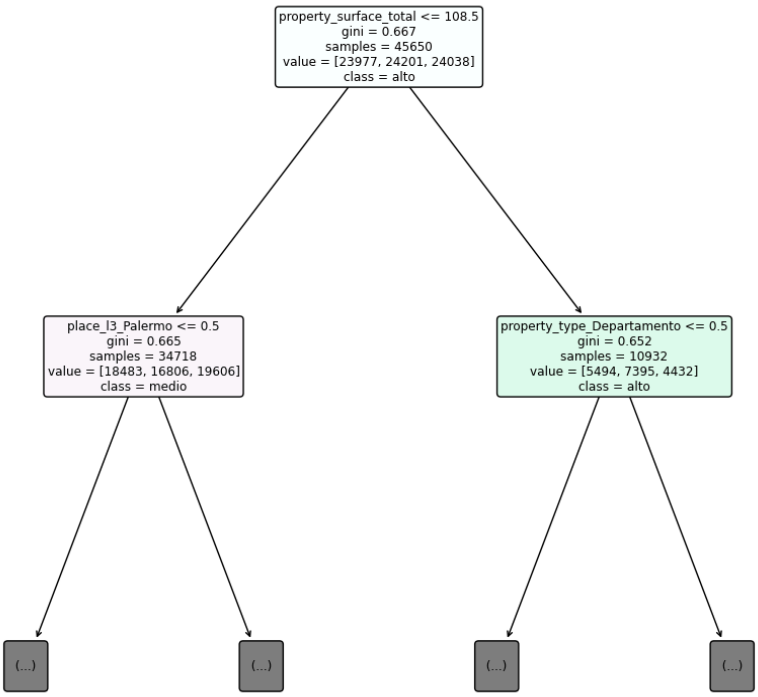
Arbol de Decision (clasificador)

		Rango	**Mejor parámetro**
Hiperparámetros	criterion	Gini, entropy	entropy
	ccp_alpha	linspace(0,0.05,10)	0.0
	max_depth	1 a 10	7
K-fold Cross Validation	5 folds		
Métrica	f1_score	average=macro	



Random Forest

		Rango	**Mejor parámetro**
Hiperparámetros	criterion	Gini, entropy	gini
	min_samples_split	5 a 50 con salto 5	20
	n_estimators	10 a 150 con paso 10	130
	max_depth	2 a 25	14
	ccp_alpha	linspace(0,0.05,10)	0.0
K-fold Cross Validation	5 folds		
Métrica	f1_score	average=macro	



XGBoost (clasificador)

		Rango	**Mejor parámetro**
Hiperparámetros	learning_rate	linspace(0.05,0.5,50)	0.42653061224489797
	subsample	linspace(0.5,1,20)	0.8157894736842105
	gamma	0,1,2	0
	lambda	0,1,2	1
	alpha	0,1,2	0
	n_estimators	10 a 100 con paso 10	90
	max_depth	2 a 10	6
K-fold Cross Validation	5 folds		
Métrica	f1_score	average=macro	

b. Cuadro de Resultados

Medidas de rendimiento en el conjunto de TEST

Modelo	**F1-Test**	**Precision Test**	**Recall Test**	**Accuracy Test**
Árbol de Decisión	0.40	0.40	0.40	0.40
Random Forest	0.41	0.39	0.41	0.39
XGBoost	0.40	0.40	0.40	0.40

Medidas de rendimiento en el conjunto de TRAIN:

Modelo	**F1-Test**	**Precision Test**	**Recall Test**	**Accuracy Test**
Arbol de Decision	0.57	0.57	0.58	0.58
Random Forest	0.64	0.65	0.64	0.65
XGBoost	0.70	0.70	0.70	0.70

Comparación y Análisis en **Árbol de Decisión**:
El rendimiento del Árbol de Decisión disminuye significativamente del conjunto de entrenamiento al conjunto de prueba. Esto nos sugiere que el modelo podría estar en overfitting al conjunto de entrenamiento

Comparación y Análisis en **Random Forest**:
El Random Forest muestra una reducción en el rendimiento al pasar del entrenamiento a la prueba, pero la caída es menos pronunciada comparada con el Árbol de Decisión, aunque sigue siendo notable. Sin embargo este modelo tiene un tiempo de entrenamiento mucho mayor a sus contrincantes (siendo de aproximadamente media hora, versus los otros que tardan algunos minutos).

Comparación y Análisis en **XGBoost**:
El rendimiento de XGBoost también disminuye drásticamente en el conjunto de prueba, similar al Árbol de Decisión, lo que nos puede indicar overfitting.

En síntesis, todos los modelos muestran signos de sobreajuste, especialmente XGBoost y el Árbol de Decisión, ya que sus rendimientos caen significativamente al pasar del conjunto de entrenamiento al de prueba. El Random Forest tiene una mejor capacidad de generalización comparado con los otros dos modelos, aunque su rendimiento también cae, la caída es menos pronunciada. Los modelos aún podrían mejorarse aplicando más técnicas de regularización, mayor validación cruzada, o ajuste de hiperparámetros para mejorar su rendimiento en el conjunto de prueba.

c. Elección del modelo

Una vez obtenidos los resultados de las métricas presentadas en la tabla, podemos concluir que el mejor modelo para clasificar el tipo de precio de las propiedades sería XGBoost.
Razones:

- Mejor Rendimiento en Métricas Clave: XGBoost tiene las puntuaciones más altas en todas las métricas evaluadas (F1-Test, Precision Test, Recall Test y Accuracy Test). Esto sugiere que XGBoost es capaz de capturar mejor las relaciones en los datos y generalizar de manera más efectiva en comparación con los otros modelos.
- Equilibrio entre Precisión y Recall: Las métricas de precisión y recall son iguales (0.70), lo que indica un buen equilibrio entre la capacidad del modelo para identificar correctamente las instancias positivas y su capacidad para evitar falsos negativos. Este equilibrio es crucial en aplicaciones prácticas donde ambos aspectos son importantes.
- Consistencia en el Desempeño: XGBoost muestra un desempeño consistente en todas las métricas, lo que indica que es menos probable que el modelo sufra de problemas de sobreajuste(Overfitting) o subajuste(Underfitting) en comparación con otros modelos como el Árbol de Decisión.
- Robustez y Flexibilidad: XGBoost es conocido por ser un algoritmo robusto y flexible, capaz de manejar datos complejos y de alta dimensionalidad. También ofrece varias técnicas de regularización que ayudan a prevenir el sobreajuste, lo que es una ventaja significativa en modelos predictivos.

Conclusión:
Debido a su rendimiento superior en todas las métricas evaluadas y sus características intrínsecas de robustez y flexibilidad, recomendamos utilizar el modelo **XGBoost para clasificar el tipo de precio de las propiedades.**

Regresión

Ingeniería de características

En esta sección se aplica la ingeniería de características anteriormente utilizada para la parte de Clasificación con la diferencia de que esta vez se hace un encodeo numérico en lugar de one hot (es decir se asigna un número a cada valor, como 0, 1, 2, etc) para place_l3 y property_type para trabajar con menor dimensionalidad. Adicionalmente se normalizaron los datos con MinMax. También en lugar del tipo de precio vamos a predecir el precio per se de la propiedad con los siguientes modelos de regresión.

a. Construcción del modelo

Para cada modelo dividimos los dataset en x e y . Siendo x los parámetros a entrenar e y los parámetros a predecir. Este criterio se usa para los tres modelos.
Realizamos la validación cruzada para probar el modelo en diferentes conjuntos de datos para asegurarnos de que funcione bien en general y no sólo con un conjunto específico. Luego utilizamos la función make_scorer para medir qué tan bien predice el modelo.

A partir de la grilla de parámetros probamos diferentes configuraciones para el modelo y realizamos la búsqueda aleatoria de los parámetros. Probamos varias combinaciones de los parámetros anteriores para encontrar cuál funciona mejor. Esto se hace de manera aleatoria para cubrir más posibilidades sin probar todas las combinaciones posibles. Por último entrenamos el modelo con los datos de entrenamiento, obteniendo el mejor resultado que nos muestra qué tan bien funciona el modelo con la mejor combinación de parámetros.

Luego de realizar las pruebas, identificamos cuáles fueron los mejores parámetros que a partir de ellos, se busca tener el mejor regresor según el modelo elegido.
Por último vamos a ver las métricas que arrojan este mejor regresor con los datos de test y train.

KNN

Aquí veremos los hiperparámetros para K-Nearest Neighbors y la configuración empleada para optimizar y evaluar el modelo.

		Rango/Opciones	**Mejor parámetro**
Hiperparámetros	n_neighbors	5 a 25	21
	weights	Distance, uniform	distance
	algorithm	ball_tree, kd_tree, brute	brute
	metric	Euclidean ,manhattan, chebyshev	euclidean
K-fold Cross Validation	10 folds		
Métrica	r2_score		

- N_neighbors: este parámetro determina el número de vecinos más cercanos que el modelo considerará para hacer una predicción. En este caso, se probó un rango de 5 a 25 vecinos y se encontró que 21 vecinos es el valor que mejor funcionó.

- Weights: este parámetro define cómo se ponderan los vecinos en una predicción. Con distance, los vecinos más cercanos tienen mayor influencia, mientras que con uniform, todos los vecinos tienen igual peso. Finalmente se determinó que ponderar por distance proporciona mejores resultados.

- Algorithm: este parámetro especifica el algoritmo utilizado para computar los vecinos más cercanos. Ball_tree y kd_tree son estructuras de datos especializadas para búsquedas eficientes, mientras que brute significa realizar la búsqueda a fuerza bruta. En este caso ganó brute.

- Metric: Este parámetro define la métrica de distancia utilizada para encontrar a los vecinos más cercanos. La métrica euclidiana fue la que mejor rendimiento ofreció.

- La validación cruzada con 10 pliegues significa que el dataset se divide en 10 subconjuntos (folds). En cada iteración, uno de los pliegues se utiliza como conjunto de prueba, mientras que los otros nueve se utilizan para entrenar el modelo. Este proceso se repite 10 veces, asegurando que cada pliegue se utilice una vez como conjunto de prueba. Este método ayuda a evaluar la estabilidad y generalización del modelo. Esta validación se repite en todos los modelos.

Entonces, se configuraron y probaron varios hiperparámetros del modelo KNN para encontrar la mejor combinación que optimiza el rendimiento predictivo. El modelo resultante utiliza 21 vecinos, pondera los vecinos por distancia, emplea el algoritmo de búsqueda exhaustiva y utiliza la métrica euclidiana para calcular las distancias. La validación cruzada con 10 pliegues se utilizó para evaluar de manera robusta el rendimiento del modelo y asegurar que funcione bien en diferentes subconjuntos de datos.

XGBoost (regresor)
Aquí veremos los hiperparámetros para XGBoost y la configuración empleada para optimizar y evaluar el modelo.

		Rango/Opciones	**Mejor parámetro**
Hiperparámetros	learning_rate	linspace(0.05,0.5,50)	0.5
	max_depth	2 a 10	3
	subsample	linspace(0.5,1,20)	0.7105263157894737
	gamma	0,1,2	0
	lambda	0,1,2	0
	alpha	0,1,2	0
	n_estimators	10 a 150 con paso 10	120
K-fold Cross Validation	5 folds		
Métrica	r2_score		

- Learning rate: El learning rate controla la contribución de cada árbol al modelo y es utilizado para evitar sobreajustes. Un valor más bajo requiere más árboles en el modelo. El valor óptimo fue 0.5.

- Max_depth: es la máxima profundidad de cada árbol. Un valor más alto permite al modelo capturar relaciones más complejas en los datos, pero también puede llevar a sobreajustes. En este caso llegamos a un valor óptimo de profundidad 3.

- Subsample: es la proporción de muestras utilizadas para entrenar cada árbol. Valores más bajos pueden evitar sobreajustes, mientras que valores más altos pueden hacer que el modelo sea más robusto. En este caso el valor óptimo fue 0.71.

- Gamma, lambda y alpha: Parámetros de regularización utilizados para controlar la complejidad del modelo y prevenir sobreajustes. En este caso, se utilizó el valor predeterminado de 0 para estos parámetros.

- N_estimators: El número de árboles a utilizar en el modelo. Más árboles pueden mejorar la precisión del modelo, pero también aumentan el tiempo de entrenamiento.

Entonces, los hiperparámetros se ajustaron para optimizar el rendimiento del modelo XGBoost, con especial atención a la tasa de aprendizaje, la profundidad máxima del árbol y el número de estimadores. La validación cruzada ayudó a garantizar que el modelo sea robusto y generalice bien a datos no vistos.

Árbol de decisión (regresor)

Aquí veremos los hiperparámetros para Árbol de decisión de regresión y la configuración empleada para optimizar y evaluar el modelo.

		Rango/Opciones	**Mejor parámetro**
Hiperparámetros	criterion	squared_error, friedman_mse, absolute_error	squared_error
	min_samples_leaf	5 a 10	5
	min_samples_split	2, 4, 10, 12, 16	4
	splitter	Random, best	best
K-fold Cross Validation	5 folds		
Métrica	r2_score		

- Criterion: El criterio es utilizado para medir la calidad de una división. En este caso, se utiliza el error cuadrático medio.

- Min_sample_leaf: es el número mínimo de muestras requeridas para estar en un nodo hoja. Un valor más alto puede evitar el sobreajuste, pero también puede hacer que el modelo sea menos flexible. El valor óptimo resulta ser 4.

- Min_samples_split: es el número mínimo de muestras requeridas para dividir un nodo interno. Este parámetro controla la profundidad de los árboles generados. En este caso se obtiene como valor óptimo las 5 muestras requeridas.

- Splitter: es la estrategia utilizada para elegir la división de cada nodo. En este caso, se elige la mejor división posible.

Entonces, el modelo de árbol de decisión se ajustó con diferentes hiperparámetros para optimizar su rendimiento en la tarea de regresión. Se exploraron diversas combinaciones de criterios de división, número mínimo de muestra en nodos hojas y

nodos internos, y estrategias de división. La validación cruzada proporciona una evaluación imparcial del rendimiento del modelo en diferentes conjuntos de datos, lo que ayuda a garantizar su capacidad de generalización.

En resumen, cada modelo fue ajustado mediante una combinación de búsqueda aleatoria y validación cruzada, asegurando no solo un buen rendimiento en el conjunto de datos de entrenamiento, sino también una sólida capacidad de generalización. Los resultados indican que los modelos optimizados están bien preparados para realizar predicciones precisas de precios de propiedades.

b. Cuadro de Resultados

- **MSE (Mean Square Error)**: Es la media de los errores al cuadrado entre los valores predichos por el modelo y los valores reales.
- **RMSE (Root Mean Squared Error)**: Raíz cuadrada del MSE, en la misma unidad que el original, por lo que facilita la identificación del error promedio.
- **R2-Score (Coeficiente de determinación)**: Proporción de variabilidad en los datos de salida. Un valor de R2 más cercano a 1 indica que el modelo explica bien la variabilidad de los datos.

Medidas de rendimiento en el conjunto de TEST

Modelo	**MSE**	**RMSE**	**R2-Score**
KNN	0.000131575307048860 9	0.011470628014579713	0.37963183068352946
XGBoost	8.873311150515407e-05	0.009419825449824113	0.5816297208277482
Árbol de decisión (Regresión)	9.027193895245698e-05	0.009501154611543 64	0.574374259390575

Medidas de rendimiento en el conjunto de TRAIN

Modelo	**MSE**	**RMSE**	**R2-Score**
KNN	2.2743972777593e-05	0.00476906414064573	0.8914397864266198
XGBoost	5.227970868486065e-05	0.007230470848074879	0.7504615224489726
Árbol de decisión (Regresión)	3.943778225825102e-05	0.006279950816547135	0.8117578618879576

En base a las métricas podemos ver:

MSE:
En el conjunto de entrenamiento, tanto en KNN como en XGB el MSE es muy bajo, lo que indica un ajuste cercano de los modelos a los datos de entrenamiento. En cambio en Árbol de Decisión tenemos un valor más alto.
En el conjunto de prueba, el MSE más bajo se observa para el modelo de Árbol, seguido por KNN y XGB.

RMSE:
Los valores de RMSE en el conjunto de entrenamiento son bajos para KNN y XGB, mientras que para el modelo de Árbol son más altos.
En el conjunto de prueba, el modelo de Árbol tiene el RMSE más bajo, lo que sugiere que es el modelo que mejor generaliza en datos nuevos y no vistos.

R2:
En el conjunto de entrenamiento, tanto KNN como XGB tienen un alto R2, lo que indica una buena capacidad para explicar la variabilidad de los datos. Sin embargo, el modelo de Árbol tiene un valor muy bajo de R2.
En el conjunto de prueba, el R2 más alto se observa para el modelo de Árbol, seguido por XGB y luego KNN. Esto sugiere que el modelo de Árbol tiene el mejor rendimiento en la explicación de la variabilidad de los datos de prueba.
En resumen, basándonos en estas métricas, parece que el modelo de Árbol de Decisión tiene un mejor rendimiento en el conjunto de prueba, mientras que los modelos KNN y XGB parecen tener un rendimiento comparable en el conjunto de entrenamiento.

c. Elección del modelo

El modelo a elegir sería el XGBoost o el árbol de decisión, ya que ambos tuvieron una performance parecida en datos nuevos para ellos y uno no tarda mucho más que el otro en entrenarse aunque, lógicamente xg boost tarda un poco más ya que se trata de un modelo más complejo con ensambles y boosteo.

Conclusión

Tras realizar el análisis de las viviendas de la Ciudad Autónoma de Buenos Aires, pudimos reconocer patrones que se corresponden a la realidad según nuestro criterio.

Durante la etapa inicial de exploración, detectamos deficiencias en la carga de datos, especialmente en la precisión de la información sobre el barrio en relación con la longitud y latitud de algunas viviendas. Para abordar esta problemática, implementamos técnicas que nos permitieron completar los datos faltantes sin necesidad de descartar registros, lo cual nos brindó un conjunto de datos robusto para nuestro análisis.

Al realizar el análisis de valores atípicos, observamos que, en una perspectiva univariada, no había una cantidad significativa de extremos, lo cual resultó ser aceptable dada la cantidad de datos disponibles. Sin embargo, al examinar los datos de manera multivariada, identificamos una mayor presencia de valores atípicos en los departamentos, particularmente en lo que respecta al precio, la superficie cubierta y la superficie total.

Respecto al clustering se puede observar que devolvió patrones distintos en la distribución de propiedades. Aunque inicialmente se observó una agrupación geográfica perfecta basada en los barrios, el análisis más detallado mostró que esta agrupación no era significativa en términos de otras variables relevantes como el precio o la superficie. Además, al realizar un segundo análisis con un número diferente de grupos y sin considerar la variable de barrios, se descubrió que las propiedades se clasificaban de forma natural según su categoría, insinuando que el clustering no aportaba datos significativos en esta situación particular. Entonces, mientras el clustering parecía bueno desde un inicio para identificar patrones en los datos, los resultados finales indicaron que otros enfoques analíticos podrían ser más adecuados para comprender la distribución y características de las propiedades en la ciudad.

Con respecto a los modelos de clasificación, al entrenar y probar los modelos de árbol de decisión, random forest y XG Boost, no obtuvimos las métricas esperadas. Para lograr mejores resultados, se probaron diferentes modificaciones en el preprocesamiento de datos, pero no se pudo lograr grandes mejoras. La mayoría de estas modificaciones no se ven volcadas en la entrega. Entre ellas contamos:

- Control de los barrios imputados a través de Cold Deck.
- Control y reemplazo de error en asignación de barrio a través de latitud-longitud.
- Análisis de outliers según precio por metro cuadrado y tipo de propiedad, reduciendo los registros que en caso de Casas y PHs tuvieran un precio mayor a $4000 por m2, y $7000 en caso de Departamentos.
- Análisis de outliers según precio por metro cuadrado, eliminando registros que superan la distancia estipulada por la regla de oro de Z-Score.
- Eliminar registros para aquellos barrios con muy poca representación. Se probó eliminando barrios con menos de 1000, 500 y 100 propiedades.

A su vez se probó con diferentes ajustes de hiperparámetros, y combinación de campos a considerar, sobretodo en el caso del árbol de decisión y el modelo de random forest, donde se evaluó posibilidades donde solo se tomara la latitud y longitud y no el barrio y viceversa. También se probaron variedad de profundidades máximas para los árboles y de rangos de estimación. Estos cambios tampoco produjeron resultados satisfactorios. Sin embargo, al observar las matrices de confusión, algunos sets de parámetros, si bien no mostraban grandes avances a nivel de promedio de métricas, lograban predecir mucho mejor ciertos tipos de precio.
En éstos experimentos se reiteraron resultados en los que predominaba la importancia de ciertas variables en específico, sobre todo aquellas relacionadas a patrones geográficos, a la hora de determinar el posible precio de una propiedad.

Se probaron tres modelos de regresión: K-Nearest Neighbors (KNN), XGBoost, y Árbol de Decisión y se evaluaron sus desempeños utilizando métricas como MSE, RMSE y R2-Score. Al igual que en clasificación, nos hubiera gustado obtener mejores resultados en la performance de los modelos, razón por la que se apuntó a una mejora en el preprocesamiento del dataset, para evitar datos ruidosos, insuficientes o no representativos.

Tiempo dedicado

Integrante	Tarea	Prom. Hs Semana
Aramayo Carolina	Tareas del checkpoint 1 y 2 Regresión creación de modelos y análisis de las métricas Reporte Grupal	~4
Utrera Maximo Damian	Tareas del checkpoint 1 y 2 Ingeniería de características Cross validation (modelos de clasificacion y regresion) Ajuste de hiperparametros (Modelos de clasificacion y regresion) Modelo RandomForest y XGBoost Graficos de métricas e importancia de atributos Reporte Grupal	~18
Villalba Ana Daniela	Tareas del checkpoint 1 y 2 Arbol de decisión Ajuste de hiperparametros (clasificación) Reporte grupal	~10
Fiorilo Roy	Tareas del checkpoint 1 y 2 Finalización de análisis de tres clusters Construcción del target (clasificación) Reporte grupal	~8