Taller # 3
Big Data y Machine Learning

Yenny Castillo , Fabián Vidal , Andrey Rincón , Carlos Alape

Link del repositorio: GitHub

1. Introducción

El objetivo de este taller es construir nuestro modelo predictivo capaz de estimar con alta precisión los precios de oferta de los inmuebles en el barrio Chapinero de Bogotá, empleando información proveniente de Properati complementada con múltiples fuentes externas. La relevancia de este ejercicio no es únicamente académica: casos recientes, como las pérdidas millonarias de Zillow por el uso de modelos automatizados deficientemente validados, evidencian la importancia de contar con metodologías robustas que eviten sesgos sistemáticos y sobreajustes en mercados con fuerte heterogeneidad espacial. En este contexto, combinamos un enfoque hedonista tradicional con técnicas modernas de aprendizaje de máquina, junto con un esquema de validación espacial diseñado para mitigar la filtración geográfica y mejorar la capacidad de generalización del modelo final.

Para el estudio utilizamos dos bases —`train` y `test`— a las cuales integramos diversas fuentes externas (Datos Abiertos de Bogotá y Base Wikipedia) que enriquecen la caracterización estructural y espacial de cada inmueble. Entre ellas se encuentran datos de localización geográfica (localidad, UPZ y barrio), estrato socioeconómico obtenido mediante *shapefiles* oficiales, indicadores de criminalidad por localidad, variables textuales depuradas de títulos y descripciones de los anuncios, y medidas de accesibilidad urbana derivadas de OpenStreetMap (distancias a parques, hospitales, CAI, universidades, supermercados, transporte y vías principales). El proceso incluye limpieza exhaustiva, estandarización de textos, consolidación del área útil, imputación de valores faltantes y construcción de nuevas variables. Este flujo nos permite obtener un conjunto de datos consistente, amplio y representativo, capaz de capturar tanto atributos estructurales como características del entorno urbano inmediato.

A partir de esta base enriquecida estimamos múltiples modelos predictivos pertenecientes a distintas familias de algoritmos: regresiones penalizadas (`Ridge`, `Elastic Net`), `CART`, `Random Forest`, `XGBoost`, `GBM`, `Redes Neuronales` y `SuperLearner`. El modelo con mejor desempeño fue un `XGBoost` entrenado mediante una estrategia de doble validación —5-fold aleatoria inicial seguida de una validación espacial por localidades— que nos permitió capturar relaciones no lineales, interacciones de alto orden y variaciones geográficas persistentes. Este modelo obtuvo el menor MAE en la competencia de Kaggle, superando ampliamente el desempeño de los demás enfoques y demostrando su capacidad para generalizar en un entorno urbano heterogéneo.

El análisis de importancia de variables muestra que el precio está altamente determinado por el área del inmueble y sus transformaciones logarítmicas, la ubicación espacial (latitud y longitud), las interacciones entre área y estrato, y atributos estructurales como el número de habitaciones y baños. Estas variables aparecen sistemáticamente como las más influyentes bajo los criterios de *Gain* y *SHAP*, lo que confirma que el mercado inmobiliario bogotano presenta relaciones no lineales y dependencias cruzadas entre características físicas, socioeconómicas y espaciales. La capacidad de nuestro modelo para capturar estas interacciones explica su desempeño superior.

2. Análisis descriptivo de los datos

2.1. Adecuación de los datos

La base de datos proviene de anuncios inmobiliarios de Properati para Bogotá, organizados en dos conjuntos: *train*, que incluye el precio de oferta, y *test*, que carece del precio y se emplea para generar las predicciones finales. La unidad de análisis es cada inmueble, identificado por `property_id`, con información estructural (área, baños, habitaciones), temporal (mes y año) y espacial (coordenadas). Esta estructura es compatible con un enfoque hedónico, donde el precio depende de características internas y del entorno urbano. Además, estas bases incluyen, en texto, dos columnas (*title* y *description*) que contienen información sobre las características propias de cada inmueble.

Limpieza y normalización: Todos los textos se pasaron a minúsculas, sin tildes ni signos de puntuación. A partir de esta depuración se generaron versiones normalizadas del título y la descripción, lo que permitió extraer información adicional sobre el tipo de propiedad, presencia de garaje o terraza, piso del apartamento, número de baños y posibles menciones explícitas a barrios.

Construcción del área: El área de los inmuebles se obtuvo combinando: (i) el área originalmente reportada en la variable $surface\_total$ y (ii) los campos vacíos (NAs) se completan extrayendo el área reportada dentro de la descripción del inmueble. Posteriormente, corregimos los valores atípicos en el área reportada teniendo en cuenta que pueden ser errores humanos de digitación, por ejemplo: valores mayores a $2000m^2$ fueron divididos entre 100 y se eliminaron filas con áreas no plausibles (menores a 15 m^2). A partir de esto se definió la variable final, `area_m2`, utilizada en la construcción de todos los modelos.

Georreferenciación: Con información extraída de Wikipedia, web scraping a través del paquete *rvest*, obtuvimos las localidades, UPZs y barrios de Bogotá. Estos datos nos permitieron identificar los nombres de los barrios que están contenidos en el texto de las variables *title* y *description*, y realizamos un match con las correspondientes UPZs y localidades de cada inmueble. El nombre de los barrios está reportado en la variable $barrio\_oficial$ y los casos donde no fue posible encontrar el barrio dejamos un NA.

Para obtener el estrato socieconómico de los inmuebles usamos dos archivos shapefile, disponibles en la página Datos Abiertos Bogotá, donde el primer shapefile contiene información de manzanas con estratos y el segundo integra todos los barrios de Bogotá. Como no existe un id numérico que permita unir estos dos shapefile, fue necesario hacer un $left\_join()$ geográfico con los polígonos reportados en ambas bases, y obtener una base que consolide estas tres variables. Dado que una manzana, o un barrio, pueden contener múltiples estratos, fue necesario usar la moda para agrupar los datos en un único estrato.

Integración de fuentes externas

Se añadieron variables externas relevantes para capturar el entorno urbano:

- **Criminalidad por localidad**: niveles y variación porcentual anual de: homicidios y hurtos reportados a personas y residencias. Los datos de seguridad están reportados a nivel de Localidad y año, y esta información se asignó a cada uno de los inmuebles con base en su ubicación. La fuente de estos datos es la Secretaría de Seguridad de Bogotá.
- **Estrato socioeconómico**: obtenido a partir de información geoespacial de barrios.
- **Localidad , UPZs y barrios**: se asignaron de acuerdo con la ubicación geográfica del inmueble tomando como referencia los archivos shapefile de Datos Abiertos Bogotá y la tabla de Wikipedia.
- **Amenidades urbanas (OSM)**: distancias a parques, hospitales, colegios, supermercados, estaciones de transporte, universidades, CAI, avenidas principales y centros comerciales. Las distancias se

calcularon mediante la función `st_distance` y estos datos geográficos fueron obtenidos de la base de *Open Street Maps*.

Variables derivadas del texto

Construimos variables dummy como indicadores de apartaestudios, casa, apartamento, garaje y terraza. Para obtener el número de baños y habitaciones que no fueron reportados en la base original, realizamos una extracción de texto, de la variable *description*, en la que detectamos si existe un token (i.e baño, bano, cuarto, habitación, etc) y buscamos un entero, o character, que esté ubicado a su respectivo lado izquierdo para finalmente generar una variable que contenga la información de las variables originales complementada con los baños y habitaciones reportados en las descripciones de los inmuebles.

Datos faltantes y consistencia

Luego de la construccción y limpieza de las variables, todavía existen datos faltantes que no se pudieron extraer de las variables de texto originales ni tampoco de las fuentes externas empleadas.

- **Baños del inmueble**: para finalizar la imputación de la variable de baños totales (*banos_tot*) aplicamos la siguiente regla: para los apartamentos con tres o más habitaciones se asignaron dos baños y para menos de tres habitaciones se asignó un solo baño. Para las casas con más de un piso se adjudicaron 2 baños y en caso contrario un solo baño.
- **Tipo de propiedad**: para evitar discrepancias entre la variable que reporta el tipo de propiedad (*property_type*) y la descripción del inmueble, creamos la variables *property_type_2* donde se extrae esta información y en caso de no coincidencia dejamos la categoría que reporta la descripción. Los NAs sobrantes son imputados con la moda de esta misma variable.
- **Localidad y UPZs**: los inmuebles a los cuales no fue posible detectarles el barrio, y por ende tampoco Localidad o UPZ, se les asignó la localidad a partir de la latitud y longitud suministrada en la base y se hizo match con los datos geográficos de los shapefile de Datos Abiertos Bogotá.
- **Numero de pisos de una casa**: extraemos de la descripción las expresiones alfabéticas o numéricas asociadas a esta variable. Los NAs los reemplazamos con el valor de 1 piso para la casa.
- **Numero de pisos de un apartamento**: extraemos de la descripción las expresiones alfabéticas o numéricas asociadas a esta variable. Los NAs los reemplazamos con el valor de 1 piso para el apartamento.

Muestra final

Finalmente se construyeron dos bases: `df_def_entrenamiento` y `df_def_testeo`, que contienen características estructurales, variables textuales normalizadas, estrato socioeconómico, localidad, UPZ, barrio, variables de criminalidad, dummys de características del tipo del inmueble y valores agregados (i.e. garaje y/o terraza) y medidas de accesibilidad urbana como distancia entre inmuebles y sitios estratégicos para predecir el precio. Ambas bases tienen la misma estructura, lo que permite estimar modelos y aplicarlos directamente en el conjunto de prueba.

2.2. Análisis descriptivo de los datos

Una vez finalizado el proceso de limpieza, integración de fuentes externas y construcción de variables, se realizó un análisis descriptivo para caracterizar la variación de los inmuebles y del entorno urbano en Bogotá. Este ejercicio se basa en las estadísticas de las bases *Train* y *Test*, así como en mapas de precios y de accesibilidad, y permite entender la heterogeneidad observada en la muestra de anuncios formales y

anticipar el comportamiento del modelo al capturar diferencias estructurales, espaciales y socioeconómicas.

Estadísticas descriptivas variables numéricas: Las estadísticas descriptivas de las variables numéricas (Tabla 3) muestran patrones consistentes entre las bases *Train* y *Test*. El área promedio de los inmuebles se ubica entre 125 y 130 m^2, mientras que la mediana ronda los 109 m^2. Esto indica una distribución asimétrica, donde existe un segmento reducido de viviendas con áreas considerablemente superiores al promedio. El número de habitaciones y baños refleja la configuración típica de la oferta residencial formal en Bogotá, donde los inmuebles publicados suelen corresponder a apartamentos de tres habitaciones y dos baños. Esta estructura es coherente con estadísticas reportadas por firmas inmobiliarias locales, que muestran que la mayoría de viviendas en estratos medios y altos siguen este formato estándar de distribución interna.

En cuanto a criminalidad, se observan contrastes marcados entre sectores. Por ejemplo, las localidades del centro y occidente presentan niveles anuales de hurtos y homicidios mayores, mientras que zonas del norte muestran valores menores y variaciones porcentuales más estables. Aun así, las tasas de variación anual —representadas en las variables `perc_variacion_homicidios`, `perc_variacion_hurto_re` y `perc_variacion_hurto_pe` permanecen en rangos plausibles y alineados con las cifras oficiales reportadas por la Secretaría de Seguridad.

Las distancias a amenidades urbanas exhiben una dispersión amplia. Esto es esperable en una ciudad como Bogotá, donde la ubicación determina fuertemente el acceso a parques, centros comerciales, hospitales, universidades y transporte. La variabilidad es visible en la mediana y los valores extremos: en algunas zonas la distancia al parque más cercano es inferior a 200 metros, mientras que en otras supera con facilidad el kilómetro. Esta heterogeneidad espacial se aprecia con mayor claridad en el Mapa 1 , que muestra la mediana del precio por m^2 por localidad. Las zonas del corredor centro–norte (Chapinero, Usaquén, Teusaquillo) concentran los valores más altos, mientras que localidades del sur exhiben precios sustancialmente inferiores. Este patrón es consistente con las diferencias observadas en estrato socioeconómico, accesibilidad a equipamientos urbanos y niveles de criminalidad.

De manera complementaria, el Mapa 2 presenta la mediana de la distancia entre los inmuebles y el parque más cercano. Las localidades con mayor oferta de espacio público —como Suba, Usaquén y Engativá— muestran distancias más cortas, mientras que sectores con densidad elevada o menor disponibilidad de parques presentan distancias sustancialmente mayores. Esto respalda la dispersión observada en la Tabla 3 y evidencia variaciones significativas en la accesibilidad urbana dentro de la ciudad.

Estadísticas descriptivas variables categóricas: En cuanto a las variables categóricas (Tabla 4), la mayoría de registros corresponden a apartamentos, lo que resulta coherente con un mercado predominantemente urbano. Las dummies construidas a partir del texto confirman que una fracción importante de los inmuebles cuenta con garaje y que aproximadamente un tercio tiene terraza, mientras que las categorías modales de cada variable permiten identificar de forma clara cuál es la configuración más frecuente de cada atributo en la muestra. La comparación entre *Train* y *Test* muestra distribuciones muy similares en estas proporciones, lo que sugiere que la partición original de Properati no introduce sesgos relevantes en la composición del conjunto de prueba en términos de tipo de inmueble y características principales.

Diccionario de variables

Finalmente, el diccionario de variables (Tabla 2) resume la definición y la fuente de cada una de las variables utilizadas en el modelo, incluyendo variables originales de Properati, indicadores derivados del texto de los anuncios, medidas de accesibilidad urbana construidas a partir de OpenStreetMap, variables de seguridad ciudadana y transformaciones específicas para la modelación (logaritmos e interacciones).

Este conjunto amplio pero consistente permite capturar tanto atributos estructurales como características del entorno, elementos centrales en un enfoque hedónico de precios. En conjunto, las tablas descriptivas muestran que las bases de datos están balanceadas, sin valores no plausibles y con variación suficiente para permitir la estimación robusta del modelo predictivo.

3. Modelos y resultados

3.1. Selección del modelo y entrenamiento

En el modelo de mejor desempeño, estimado con XGBoost sobre el logaritmo del precio por metro cuadrado, se asume que el precio del inmueble P_i está determinado por una función flexible de las principales características físicas y del entorno. La relación puede escribirse como

$$P_i = f\big(\text{area\_m2\_final}_i,\, \log(\text{area\_m2\_final}_i + 1),\, \log(\text{distancia\_parque}_i + 1),\, \text{lon}_i,\, \text{bedrooms}_i,\, \mathbf{Z}_i\big) + u_i,$$

donde u_i es el término de error y $\mathbf{Z}_i$ recoge el resto de covariables incluidas en la matriz de diseño (tipo de propiedad, estrato, localidad, UPZ, año y mes, interacciones, entre otras). El algoritmo se entrena sobre la variable dependiente

$$y_i = \log\left(1 + \frac{P_i}{\text{area\_m2\_final}_i}\right),$$

es decir, el logaritmo del precio por metro cuadrado.

Tal como se observa en la Gráfica 1 y la Gráfica 2, tanto la medida de importancia basada en *gain* como la basada en valores SHAP coinciden en destacar el mismo grupo de covariables como las más influyentes en el desempeño del modelo. En particular, las cinco variables con mayor contribución son: (i) `area_m2_final`, que mide el área total construida del inmueble; (ii) `log_area`, correspondiente al logaritmo del área, que permite capturar rendimientos decrecientes del tamaño sobre el precio; (iii) `log_distancia_parque`, que recoge el logaritmo de la distancia al parque más cercano y aproxima el acceso a zonas verdes; (iv) la coordenada `lon`, que captura patrones espaciales de precios a lo largo del eje oriente–occidente de la ciudad; y (v) `bedrooms`, que representa el número de habitaciones. Estas variables combinan atributos estructurales del inmueble (tamaño y número de cuartos) con características de localización y entorno urbano, y son las que el modelo identifica como más relevantes para explicar la variación en el precio de oferta de la vivienda en Bogotá.

Un modelo *XGBoost* consiste en el ensamblaje de varios árboles de decisión en donde se construye, de forma secuencial, muchos árboles pequeños en el que cada nuevo árbol va corrigiendo los errores del conjunto anterior. Al finalizar la ejecución, se obtiene una predicción que es la suma de las contribuciones de los árboles escaladas por una tasa de aprendizaje (η).

La variable seleccionada para el modelo *XGBoost* fue el logaritmo natural de la división del precio por el m^2 de los inmuebles, calculada a partir de la variable `area_m2_final`. Esta transformación del precio nos permite reducir los posibles problemas del efecto multiplicativo del precio (entre mayor sea el área del inmueble su valor va a aumentar) ya que si se divide por el área estamos calculando cuánto cuesta cada m^2 y se reduce la dependencia directa del tamaño de la propiedad. El logaritmo natural, al que sumamos una unidad para evitar valores iguales a cero, reduce la influencia de los *outliers* provenientes de los datos, estabiliza la varianza de los precios y permite que los árboles, y sus *splits*, capturen las relaciones multiplicativas entre precio y área. Lo anterior implica una mejora en el aprendizaje del *XGBoost* porque hay una reducción en la heterocedasticidad del modelo, con la división precio por m^2, una menor presencia de datos atípicos y patrones más suaves que ayudan a un mejor ajuste de las relaciones no lineales.

Al concluir la estimación, se hace la reconversión del precio a través de una corrección por escala para evitar los problemas del sesgo de retransformación. En este caso multiplicamos el área por la función exponencial de $\hat{y}$ menos 1 y una escala que calculamos a partir de un conjunto de validación: $\widehat{\text{precio}} = \text{área} \times \big(\exp(\widehat{y}) - 1\big) \times \text{escala}$

3.1.1. Elección y ajuste de hiperparámetros

Se realizó una búsqueda en dos etapas sobre los hiperparámetros principales del *XGBoost*. En la etapa inicial (*coarse*) se probaron hasta 30 combinaciones mediante validación cruzada (VC) 5-fold aleatoria, con `nrounds`=120 y `early stopping` de 50 rondas. El entrenamiento usa `reg:squarederror` sobre y y se detiene por MAE en el espacio *log* m^2; sin embargo, la selección de combinaciones se hizo por MAE en **precio** tras reconvertir y aplicar la escala robusta. Las seis mejores pasaron a una segunda etapa (*refine*), donde se revalidaron con **VC espacial** (folds por Localidad/UPZ; en su ausencia, `k`-means sobre coordenadas `lon/lat`), con más rondas (`nrounds` hasta 1500) y `early stopping` de 100. Finalmente, elegimos la configuración y el número óptimo de árboles por MAE en **precio**.

- Tasa de aprendizaje: $\eta \in \{0{,}03,\ 0{,}05,\ 0{,}1\}$.
- Profundidad máxima del árbol: `max_depth` $\in \{4,\ 6,\ 8\}$.
- Mínimo peso por hoja: `min_child_weight` $\in \{1,\ 5,\ 10\}$.
- Muestreo por árbol: `subsample` $\in \{0{,}6,\ 0{,}8\}$ (filas) y `colsample_bytree` $\in \{0{,}6,\ 0{,}8\}$ (columnas).
- Regularización: `lambda` (L2) $\in \{0,\ 1,\ 10\}$ y `alpha` (L1) $\in \{0,\ 1\}$.

3.1.2. Estrategia de validación

En nuestro modelo *XGBoost* utilizamos una validación cruzada en dos etapas para estimar la generalización del modelo y seleccionar hiperparámetros: en primer lugar, usamos una validación cruzada regular de 5-fold aleatoria (coarse) que nos permitió explorar rápidamente muchas combinaciones con un costo computacional moderado, además en esta fase se usa early stopping y se compara cada configuración con su MAE en precio tras reconvertir las predicciones out-of-fold y aplicar una escala robusta. Esto permite descartar rápidamente configuraciones pobres y poder concentrarnos en una vecindad prometedora, manteniendo reproducibilidad con la semilla fija y evitando leakage al generar las OOF por fold.

Posteriormente, aplicamos una validación cruzada espacial (refine) que construye folds por zonas (Localidad), reduciendo la pérdida geográfica y ofreciendo una evaluación más realista; aquí se permiten más rondas y un early stopping más laxo, y se vuelve a seleccionar por MAE en precio. En ambas etapas se utilizan predicciones out-of-fold y la métrica interna de parada es el MAE en el espacio del logaritmo natural del m^2. Con esta doble validación, los hiperparámetros elegidos equilibraron el rendimiento global, la robustez espacial, y el número óptimo de rondas se fija con el mejor punto de early stopping de la segunda etapa antes de volver a entrenar el modelo final en todo el conjunto.

Al comparar los resultados obtenidos mediante validación cruzada regular y validación cruzada espacial, observamos una diferencia clara en el comportamiento del modelo. La validación cruzada regular tiende a reportar un MAE más bajo, debido a que el modelo entrena y valida sobre observaciones espacialmente cercanas, lo que facilita la captura de patrones locales muy específicos y genera un riesgo elevado de *leakage* espacial. En consecuencia, los hiperparámetros seleccionados bajo este esquema ofrecen un desempeño aparentemente superior, pero en realidad no generalizan bien cuando el modelo se evalúa fuera de muestra. En contraste, la validación espacial, en donde separamos los datos por localidad, produce un MAE más alto pero más representativo del error real de generalización fuera de muestra.

Tabla 1: Comparación de desempeño por método de validación.

Método	**MAE**
Validación regular (aleatoria)	113,650,599
Validación espacial	149,799,535

3.1.3. Decisiones metodológicas adicionales

Para mejorar la generalización y la estabilidad del modelo tomamos varias decisiones adicionales al ajuste de los hiperparámetros: se consolidó e imputó el área de los inmuebles, se aplicaron transformaciones logarítmicas e interacciones, se redujo la cardinalidad de categorías, se construyó una matriz de diseño consistente y se corrigieron las predicciones con una escala robusta y clipping; la validación espacial con predicciones OOF permitió estimar esa escala sin pérdida de localidades y seleccionar el modelo por MAE en precio, dando lugar a una mejora de la generalización geográfica.

3.2. Análisis comparativo

OLS con Ridge: El modelo Ridge es una regresión lineal penalizada con L2 que minimiza el error cuadrático más un término $\lambda||\beta||^2$ que encoje los coeficientes, reduce la varianza, estabiliza la multicolinealidad, pero mantiene una relación lineal entre la variable dependiente y las independientes. El hiperparámetro del modelo es λ y se fija con validación cruzada espacial de 5-folds optimizando el MAE sobre el precio. La penalización del modelo Ridge suele introducir sesgo que evita la captura de interacciones complejas y estructuras locales, razón por la que el MAE final de este modelo es mayor que el XGBoost final con menor puntaje en Kaggle.

CART: El modelo CART es un árbol de regresión que parte recursivamente el espacio de predictores para minimizar el error cuadrático dentro de cada hoja. Su complejidad se controla con hiperparámetros como *cp* (penalización de complejidad), que fueron fijados manualmente (*cp*=0.01, minsplit=40, minbucket=15, maxdepth=10) y se usa validación cruzada espacial para generar predicciones sobre y, que luego se retransformaron a precio. Al ser un único árbol sin búsqueda de hiperparámetros, CART suele tener una mayor varianza y menor capacidad para capturar relaciones suaves e interacciones complejas en todo el espacio, por lo que su MAE final tiende a ser mayor que el del modelo XGBoost.

Elastic Net: Se ajustó un modelo Elastic Net para manejar alta dimensionalidad y colinealidad entre predictores. La validación cruzada implementada fue estrictamente espacial, donde cada fold corresponde a uno de los cinco clusters geográficos obtenidos a partir de las coordenadas de latitud y longitud. Se exploró una grilla de valores de α entre 0.1 y 0.9, junto con múltiples valores de λ. El mejor desempeño se alcanzó con un grado de regularización predominantemente *ridge* ($\alpha = 0{,}1$) y un parámetro de penalización $\lambda_{\text{mín}} \approx 0{,}0167$, que produjo el menor error cuadrático medio espacial (cv_mse_min = 0,318).A pesar de su estabilidad y capacidad para manejar colinealidad, el Elastic Net presentó uno de los desempeños predictivos más débiles del conjunto de modelos, lo cual es coherente con su naturaleza lineal frente a las relaciones marcadamente no lineales presentes en el mercado inmobiliario de Bogotá.

Random Forest con CV espacial: Se implementó un modelo Random Forest calibrado mediante validación cruzada estrictamente espacial, explorando una grilla de hiperparámetros que variaba el número de árboles, la profundidad efectiva controlada mediante `mtry` y el tamaño mínimo de nodo. La mejor configuración seleccionada correspondió a un bosque de 500 árboles, con `mtry` = 15 y un `min.node.size` = 5, obtenidos a partir del esquema de entrenamiento en cuatro zonas geográficas y validación en la quinta, lo que permitió identificar una estructura adecuada para capturar la complejidad de las relaciones espaciales y no lineales presentes en el mercado inmobiliario.

XGBoost II: El modelo XGBoost se entrenó con una partición interna 80/20 y *early stopping* aplicado sobre el MAE. La especificación utilizada incluyó los hiperparámetros `eta = 0.05`, `max_depth = 6`, `subsample = 0.8`, `colsample_bytree = 0.8` y `min_child_weight = 5`, configurando un modelo con una capacidad moderada para capturar relaciones no lineales sin incurrir en sobreajuste. El proceso de validación interna determinó un número óptimo de **2000 árboles** (`nrounds = 2000`). En términos comparativos, este modelo se posicionó muy cerca del modelo con mejor desempeño en Kaggle, aunque ligeramente por debajo, lo cual se explica por las restricciones impuestas en la profundidad y en el peso mínimo de nodo —hiperparámetros que favorecen la estabilidad y generalización, pero limitan parcialmente la capacidad de capturar interacciones altamente complejas presentes en el mercado inmobiliario.

Random Forest: El modelo de Random Forest se entrenó con 800 árboles, mtry = sqrt(p),min.node.size = 5 y sample.fraction = 0.8. En el test interno obtuvo un MAE de 146 millones, mientras que en Kaggle el error aumentó a 247 millones, evidenciando una generalización más débil que la observada en la partición local 70/30 utilizada durante el entrenamiento. Frente a este comportamiento, el modelo XGBoost presentó un MAE de 186 millones en Kaggle, el menor entre todas las propuestas. La diferencia en desempeño se explica principalmente por la estrategia de validación: mientras el Random Forest se basó en una división aleatoria simple, XGBoost implementó una validación cruzada en dos etapas (5-fold aleatoria *coarse* seguida de una validación espacial *refine* por Localidades), incorporando además *early stopping* y evaluación del MAE tanto en $\log(m^2)$ como en el precio reconvertido. Esto permitió a XGBoost ajustar interacciones complejas, reducir *leakage* geográfico y seleccionar hiperparámetros mucho más robustos.

Gradient Boosting Machine (GBM) : Para el modelo GBM se evaluaron configuraciones con `n.trees = {300, 600, 1000}`, profundidades de 3 y 5, tasas de aprendizaje de 0.01 y 0.005, y tamaños mínimos de nodo de 10 y 20. El modelo óptimo correspondió a la configuración de 1000 árboles, profundidad 5, *shrinkage* 0.01 y 20 observaciones mínimas por nodo. Su MAE interno fue de 152 millones, mientras que en Kaggle obtuvo un MAE mayor, de 222 millones. A diferencia del enfoque de doble validación utilizado en XGBoost, el GBM se entrenó exclusivamente con validación cruzada aleatoria estándar, sin un componente espacial que capturara la heterogeneidad geográfica del mercado inmobiliario. El refinamiento espacial de XGBoost, junto con un *early stopping* más laxo en la etapa *refine* y la evaluación del MAE en $\log(m^2)$ al seleccionar el número óptimo de rondas, le permitió modelar mejor las variaciones no lineales y los patrones espaciales ausentes en el GBM estándar.

Red neuronal MLP: El modelo de red neuronal implementado fue un `MLP` de una sola capa oculta, entrenado sobre la matriz numérica de predictores generada en el preprocesamiento. Todas las variables fueron transformadas e imputadas para evitar inconsistencias. La arquitectura incluyó 20 neuronas ocultas, un `decay = 0.001` y 300 iteraciones, buscando equilibrar aprendizaje y sobreajuste. El entrenamiento se realizó con una partición 80/20 y se evaluó mediante `RMSE`, `MAE`, $\mathtt{R}^2$ y `MAPE`. A diferencia de los métodos basados en árboles, este `MLP` depende de relaciones suaves y su baja profundidad limita la captura de interacciones no lineales, lo que resultó en un desempeño inferior y mayor sensibilidad a la escala de los precios. Sin validación espacial estricta ni un ajuste exhaustivo de hiperparámetros, la red no alcanzó el rendimiento de modelos más robustos como `Random Forest` o `XGBoost`.

SuperLearner: El SuperLearner combinó los algoritmos `SL.mean`, `SL.lm`, `SL.glm`, `SL.ridge`, `SL.glmnet`, `SL.ranger` y `SL.gbm`, aunque bajo hiperparámetros conservadores —por ejemplo, `SL.ranger` con 300 árboles y `SL.gbm` con solo 100 árboles y profundidad 2— y empleando un conjunto reducido de predictores debido a restricciones computacionales. Internamente obtuvo un MAE de 141 millones, pero su desempeño en Kaggle obtuvo un MAE de 260 millones. Este modelo no incorporó validación espacial, utilizó predictores limitados y no realizó una búsqueda exhaustiva de hiperparámetros en los *learners* base. A pesar de la potencia teórica del SuperLearner, su configuración resultó insuficiente para igualar el rendimiento de XGBoost, cuyo diseño metodológico más cuidadoso alcanzó el mejor desempeño absoluto.

3.3. Comparación del modelo con mejor rendimiento y las nueve propuestas

La Tabla 2 muestra que el modelo con mejor desempeño en la competencia de Kaggle fue el `XGBoost (ganador)`, con un MAE de 186 millones, superando de manera consistente a las otras nueve mejores propuestas evaluadas por el equipo. Este resultado se explica por una combinación de decisiones metodológicas que optimizaron su capacidad de generalización: (i) una especificación centrada en predecir el logaritmo del precio por metro cuadrado, lo que redujo la heterocedasticidad y estabilizó la escala del objetivo; (ii) una estrategia de validación en dos etapas —5-fold aleatoria *coarse* seguida de validación espacial por Localidad *refine*— que mitigó el riesgo de *leakage* geográfico; y (iii) un ajuste cuidadoso de hiperparámetros, combinando tasas de aprendizaje bajas (`eta` entre 0.03 y 0.05), profundidades moderadas de árbol (`max_depth` = 6), muestreo por filas y columnas (`subsample = 0.8`, `colsample_bytree = 0.8`) y regularización L1/L2. Este diseño permitió capturar relaciones no lineales, interacciones entre variables estructurales e indicadores espaciales, así como gradientes suaves en la distribución de precios.

Al contrastarlo con los otros nueve modelos, se observa que las diferencias de rendimiento tienen explicaciones sistemáticas. En primer lugar, varios modelos como OLS-Ridge, Elastic Net y CART presentan una capacidad limitada para modelar interacciones y no linealidades, conduciendo a errores sustancialmente mayores en Kaggle. Por ejemplo, el Ridge y el Elastic Net —penalizaciones lineales— no logran capturar las interacciones dominantes del mercado inmobiliario que sí emergen en el XGBoost, como `area m2 final`, `log area`, `lon`, `bedrooms` y las interacciones `area × estrato` o `bed × banos`. Estas variables, identificadas como las más importantes mediante Gain y SHAP, permiten modelar rendimientos marginales decrecientes del tamaño del inmueble, gradientes espaciales oriente–occidente y complementariedades entre características internas del hogar, ninguno de los cuales es capturable mediante modelos lineales.

En segundo lugar, modelos de ensamble como `Random Forest` o `Gradient Boosting Machine` presentaron desempeños competitivos pero inferiores. Aunque estos algoritmos capturan no linealidades, sus estrategias de entrenamiento fueron menos rigurosas: en particular, la ausencia de validación espacial en Gradient Boosting Machine y la dependencia de particiones aleatorias redujo su capacidad para generalizar a zonas geográficamente aisladas del conjunto de entrenamiento. Como se observa en la Tabla 2, estos modelos tienden a tener un MAE interno relativamente bajo, pero su error en Kaggle aumenta de manera notable, evidenciando sobreajuste a patrones locales no representativos. Además, configuraciones como `mtry` fijos o profundidades limitadas restringen la habilidad del Random Forest para capturar la estructura compleja del mercado inmobiliario bogotano.

En tercer lugar, la red neuronal `MLP` exhibió el peor desempeño de los nueve modelos comparados (MAE Kaggle = 340 millones). Aunque conceptualmente apta para capturar no linealidades, la arquitectura empleada —una sola capa oculta con 20 neuronas y `decay = 0.001`— resulta insuficiente para aprender relaciones altamente complejas en presencia de heterogeneidad espacial. Además, la falta de validación espacial produció un sesgo optimista en la evaluación interna y un deterioro significativo al exponerse a nuevas localidades en Kaggle. Finalmente, el SuperLearner, pese a su potencia teórica, sufrió por el uso de un conjunto restringido de predictores y la ausencia de ajuste exhaustivo de hiperparámetros, resultando en un MAE de 260 millones.

En conjunto, la comparación evidencia que la superioridad del XGBoost ganador no se debe solo a su algoritmo, sino a un diseño metodológico integral: ingeniería de variables amplia, incorporación de fuentes externas (OSM, criminalidad, estrato), transformaciones robustas, manejo de interacciones complejas, doble validación con componente espacial y selección del número de rondas mediante early stopping. Estos elementos permitieron capturar los determinantes estructurales y espaciales identificados en los anexos (área, ubicación, accesibilidad urbana, interacciones), ofreciendo un modelo capaz de generalizar de manera superior frente a las nueve propuestas restantes.

3.4. Discusión de las variables del modelo

Para encontrar las variables que más contribuyeron al modelo vamos a tomar dos metodologías: la ganancia (*Gain*) del *XGBoost* y la media del valor absoluto (SHAP) para rankear el top 10 de variables. Con *Gain* medimos cuánto reduce la pérdida cada split de la variable estudiada y se suma esa ganancia a través de todos los árboles del modelo para construir el ranking, mientras que el SHAP descompone la predicción en contribuciones por variable y se toma la media del valor absoluto de cada contribución en todo el conjunto.

En la **Tabla 6** y la **Gráfica 1** podemos ver que la variable que más aporta a la ganancia del modelo *XGBoost* es el área en metros cuadrados de los bienes inmuebles de Bogotá. Además, le siguen otras variables como la longitud, el logaritmo del área, el número de habitaciones y baños, pero también tenemos interacciones como el área con estratos o las habitaciones con los baños. Por otra parte, en la **Gráfica 2** y la **Tabla 7**, el método SHAP sigue mostrando la prevalencia del área como la principal variable explicativa del modelo junto con la longitud, número de habitaciones, estrato, logaritmo natural del área o la interacción entre el número de habitaciones y baños.

En particular, el resultado final nos indica que el área total de un inmueble (`area_m2_final`) es la mejor variable predictora, tanto con *Gain* como *SHAP*, para nuestro modelo *XGBoost*. La prevalencia de `area_m2_final` es un insumo valioso para futuros proyectos relacionados con el mercado inmobiliario de Bogotá, ya que la obtención del área total de una casa o apartamento va a ser muy importante en el momento de predecir su respectivo precio.

4. Conclusiones y recomendaciones

El propósito central de este taller fue responder una pregunta fundamental: *¿qué modelo de predicción permite estimar con mayor precisión el precio de oferta de los inmuebles en Chapinero, a partir de características estructurales, espaciales, socioeconómicas y de accesibilidad urbana?* Para ello se construyó una base de datos enriquecida mediante fuentes externas y se entrenó un conjunto amplio de algoritmos clásicos y modernos, evaluados tanto en validación interna como en el escenario competitivo de Kaggle.

El modelo con mejor rendimiento fue `XGBoost` con validación espacial. Este modelo obtuvo uno de los mejores desempeños en la competencia de Kaggle, lo que se explica por su capacidad para capturar relaciones no lineales, interacciones complejas entre variables y patrones espaciales persistentes dentro de Chapinero. El XGBoost se apoya en un conjunto reducido pero altamente informativo de predictores: el área final y sus transformaciones logarítmicas, la ubicación espacial (latitud y longitud), el estrato socioeconómico, y características estructurales como el número de habitaciones y baños. Además, las interacciones entre área y estrato, así como múltiples medidas de accesibilidad urbana, contribuyeron a mejorar la capacidad explicativa del algoritmo.

El modelo final ofrece una herramienta robusta para estimar el valor de mercado de viviendas en Chapinero, gracias a la validación espacial que evita el sobreajuste y permite un desempeño más realista en zonas no observadas. Sin embargo, todavía existen oportunidades de mejora, como incorporar técnicas avanzadas de análisis de texto, refinar hiperparámetros y utilizar métricas espaciales adicionales. Además, dado que factores como la criminalidad, la movilidad y la oferta de servicios cambian con frecuencia en Bogotá, el modelo debe actualizarse y reentrenarse periódicamente; de lo contrario, su precisión disminuirá con el tiempo. Para una start-up que busca basar sus decisiones de compra en predicciones confiables, mantener una validación espacial y mecanismos de drift detection es fundamental para reducir riesgos y asegurar estimaciones consistentes..

Referencias

Properati. (2025). *Base de datos de anuncios inmobiliarios para Bogotá*. Recuperado de https://www.properati.com.co/

OpenStreetMap contributors. (2025). *OpenStreetMap*. Recuperado de https://www.openstreetmap.org/

Alcaldía Mayor de Bogotá D.C., & Secretaría Distrital de Planeación. (2025). *Datos Abiertos Bogotá: Información geográfica y socioeconómica de la ciudad*. Recuperado de https://datosabiertos.bogota.gov.co/

Wikipedia. (2025). *Barrios, localidades y UPZ de Bogotá*. Recuperado de https://es.wikipedia.org/wiki/Anexo:Barrios_de_Bogot%C3%A1

Slate. (2021). *Zillow quemó $381 millones pagando de más por casas. Espectacular*. Recuperado de https://slate-com.translate.goog/technology/2021/11/zillow-house-flipping-failure-awesome.html?_x_tr_sl=en&_x_tr_tl=es

5. Anexos

Tabla 2: Diccionario de variables utilizadas en el modelo final de predicción.

Variable	**Descripción**	**Fuente**
price	Precio publicado del inmueble (COP).	Properati
area_m2_final	Área consolidada entre columnas originales.	Properati / EDA propio
bedrooms	Número de habitaciones.	Properati
banos_tot	Total de baños.	Properati / EDA propio
n_pisos_num	Número de pisos del inmueble.	Texto del anuncio
piso_numerico	Piso en el que está el inmueble.	Texto del anuncio
estrato_num	Estrato socioeconómico.	Properati
dummy_apartaestudio	1 si es apartaestudio.	Texto del anuncio
dummy_apartamento	1 si es apartamento.	Texto del anuncio
dummy_casa	1 si es casa.	Texto del anuncio
dummy_garaje	1 si tiene garaje.	Texto del anuncio
dummy_terraza	1 si tiene terraza.	Texto del anuncio
property_type	Tipo de inmueble reportado.	Properati
property_type2	Tipo depurado mediante procesamiento de texto.	Texto del anuncio
lat	Latitud del inmueble.	Properati (georreferenciación)
lon	Longitud del inmueble.	Properati (georreferenciación)
LocNombre	Localidad del inmueble.	Wikipedia / Procesamiento propio
upz	UPZ asociada.	Wikipedia / Procesamiento propio
num_homicidios_anual	Total anual de homicidios en el sector.	Datos de seguridad ciudadana
perc_variacion_homicidios	Variación porcentual anual de homicidios.	Datos de seguridad ciudadana
num_hurto_re_anual	Hurtos a residencia al año.	Datos de seguridad ciudadana
perc_variacion_hurto_re	Variación porcentual de hurtos a residencia.	Datos de seguridad ciudadana
num_hurto_pe_anual	Hurtos a personas al año.	Datos de seguridad ciudadana
perc_variacion_hurto_pe	Variación porcentual de hurtos a personas.	Datos de seguridad ciudadana
area_parque	Área del parque más cercano.	OpenStreetMap
log_distancia_parque	Log-distancia al parque más cercano.	OpenStreetMap
log_distancia_gimnasio	Log-distancia a gimnasio cercano.	OpenStreetMap
log_distancia_hospital	Log-distancia al hospital más cercano.	OpenStreetMap
log_distancia_colegio	Log-distancia al colegio cercano.	OpenStreetMap
log_distancia_supermercado	Log-distancia al supermercado.	OpenStreetMap
log_distancia_universidad	Log-distancia a universidad cercana.	OpenStreetMap
log_distancia_bus	Log-distancia al paradero de bus.	OpenStreetMap
log_distancia_cai	Log-distancia al CAI cercano.	OpenStreetMap
log_distancia_avenida_principal	Log-distancia a vía principal.	OpenStreetMap
log_distancia_centrocomercial	Log-distancia a centro comercial.	OpenStreetMap
month	Mes del anuncio.	Properati
year	Año del anuncio.	Properati
log_area	Logaritmo del área del inmueble.	Transformación del modelo
log_area_parque	Logaritmo del área del parque.	Transformación del modelo
area_x_estrato	Interacción entre área y estrato.	Transformación del modelo
bed_x_banos	Interacción entre habitaciones y baños.	Transformación del modelo
lat_x_lon	Interacción entre coordenadas.	Transformación del modelo

Nota: La tabla resume todas las variables efectivamente utilizadas en el modelo final XGBoost, incluyendo variables originales de Properati, variables externas construidas a partir de OpenStreetMap y datos de seguridad ciudadana, y variables generadas mediante ingeniería de características.

Tabla 3: Variables numéricas en las bases *Train* y *Test*

Base	Variable	n	Media	Mediana
Train	area_m2	38644	131	109
Train	surface_covered	38644	113	108
Train	bedrooms	38644	3	3
Train	banos_tot	38644	3	2
Train	lat	38644	5	5
Train	lon	38644	-74	-74
Train	n_pisos_num	38644	1	1
Train	piso_numerico	38644	2	1
Train	num_homicidios_anual	38644	47	37
Train	perc_variacion_homicidios	38644	0.06	-0.08
Train	num_hurto_re_anual	38644	677	661
Train	perc_variacion_hurto_re	38644	-0.09	-0.08
Train	num_hurto_pe_anual	38644	7909	7734
Train	perc_variacion_hurto_pe	38644	0.08	0.23
Train	distancia_avenida_principal	38644	324	242
Train	distancia_bus	38644	783	618
Train	distancia_cai	38644	979	956
Train	distancia_colegio	38644	583	514
Train	distancia_gimnasio	38644	986	878
Train	distancia_hospital	38644	943	884
Train	distancia_parque	38644	196	165
Train	area_parque	38644	8032	3201
Train	distancia_supermercado	38644	488	438
Train	distancia_universidad	38644	1075	975
Train	distancia_centrocomercial	38644	722	680
Train	area_m2_final	38644	133	110
Train	estrato_num	38644	4	5
Test	area_m2	10286	126	109
Test	surface_covered	10286	113	108
Test	bedrooms	10286	2	2
Test	banos_tot	10286	3	2
Test	lat	10286	5	5
Test	lon	10286	-74	-74
Test	n_pisos_num	10286	1	1
Test	piso_numerico	10286	2	1
Test	num_homicidios_anual	10286	22	13
Test	perc_variacion_homicidios	10286	0.09	-0.08
Test	num_hurto_re_anual	10286	450	366
Test	perc_variacion_hurto_re	10286	-0.09	-0.12
Test	num_hurto_pe_anual	10286	8006	7734
Test	perc_variacion_hurto_pe	10286	0.14	0.23
Test	distancia_avenida_principal	10286	344	189
Test	distancia_bus	10286	1052	1039
Test	distancia_cai	10286	589	556
Test	distancia_colegio	10286	431	392
Test	distancia_gimnasio	10286	910	872
Test	distancia_hospital	10286	848	837
Test	distancia_parque	10286	265	200
Test	area_parque	10286	5593	1417
Test	distancia_supermercado	10286	612	561
Test	distancia_universidad	10286	530	448
Test	distancia_centrocomercial	10286	884	821
Test	area_m2_final	10286	129	110
Test	estrato_num	10286	5	6

Fuente: Cálculos propios con base en la información de Properati para Bogotá y variables externas obtenidas de OpenStreetMap y del contenido de los anuncios inmobiliarios.

Tabla 4: Variables categóricas en las bases *Train* y *Test*.

Base	Variable	Categoría	n	Proporción (%)
Train	dummy_apartaestudio	0	38134	98.70
Train	dummy_apartaestudio	1	510	1.30
Train	dummy_apartamento	0	9501	24.60
Train	dummy_apartamento	1	29143	75.40
Train	dummy_casa	0	29653	76.70
Train	dummy_casa	1	8991	23.30
Train	dummy_garaje	0	11774	30.50
Train	dummy_garaje	1	26870	69.50
Train	dummy_terraza	0	27109	70.20
Train	dummy_terraza	1	11535	29.80
Train	property_type	Apartamento	29177	75.50
Train	property_type	Casa	9467	24.50
Train	property_type2	Apartamento	29642	76.70
Train	property_type2	Casa	9002	23.30
Test	dummy_apartaestudio	0	10047	97.70
Test	dummy_apartaestudio	1	239	2.30
Test	dummy_apartamento	0	578	5.60
Test	dummy_apartamento	1	9708	94.40
Test	dummy_casa	0	9947	96.70
Test	dummy_casa	1	339	3.30
Test	dummy_garaje	0	3153	30.70
Test	dummy_garaje	1	7133	69.30
Test	dummy_terraza	0	7000	68.10
Test	dummy_terraza	1	3286	31.90
Test	property_type	Apartamento	10012	97.30
Test	property_type	Casa	274	2.70
Test	property_type2	Apartamento	9947	96.70
Test	property_type2	Casa	339	3.30

Fuente: Cálculos propios con base en información de Properati para Bogotá.
Variables externas obtenidas de OpenStreetMap y del contenido de los anuncios inmobiliarios.

Tabla 5: Comparación de modelos: métricas internas y puntaje en Kaggle.

Modelo general	**MAE (test)**	**RMSE (test)**	**Puntaje Kaggle (MAE)**
Redes neuronales	245,729,670	309,030,826	340,115,955
OLS-Ridge	205,171,801	287,731,542	265,571,420
CART	205,171,801	287,731,542	248,255,420
Random Forest	146,052,767	193,585,019	247,039,507
Gradient Boosting Machine	152,493,262	212,582,468	222,057,769
SuperLearner	141,946,030	195,582,202	260,196,139
Elastic Net	172,837,162	251,996,502	254,519,525
Random Forest con CV espacial	101,228,886	162,352,357	218,810,463
XGBoost II	98,704,429	155,788,519	188,895,296
XGBoost (ganador)	149,832,428	219,633,506	186,141,249

Tabla 6: Top 10 variables por Gain (XGBoost).

Ranking	Variable	Gain	Cover	Frequency	MeanAbsSHAP
1	area_m2_final	0.2694	0.0474	0.0367	0.1434
2	log_area	0.1265	0.0243	0.0149	0.0380
3	area_m2	0.0803	0.0188	0.0265	0.0054
4	lon	0.0590	0.0567	0.0448	0.0582
5	bedrooms	0.0359	0.0275	0.0243	0.0394
6	banos_tot	0.0345	0.0343	0.0202	0.0424

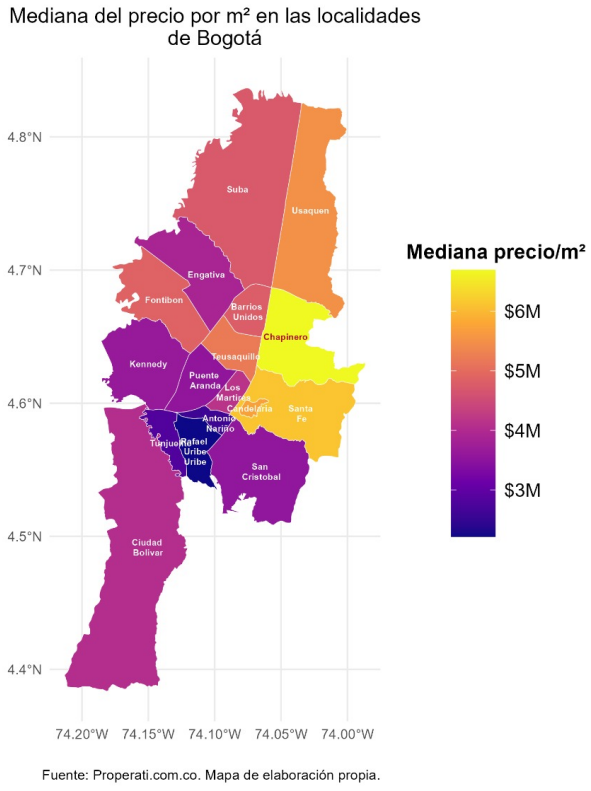
Ranking	Variable	Gain	Cover	Frequency	MeanAbsSHAP
7	area_x_estrato	0.0328	0.0285	0.0367	0.0111
8	estrato_num	0.0287	0.0123	0.0110	0.0332
9	lat	0.0276	0.0351	0.0423	0.0269
10	bed_x_banos	0.0275	0.0317	0.0294	0.0382

Fuente: Cálculos propios con base en Properati para Bogotá. Tabla de elaboración propia.

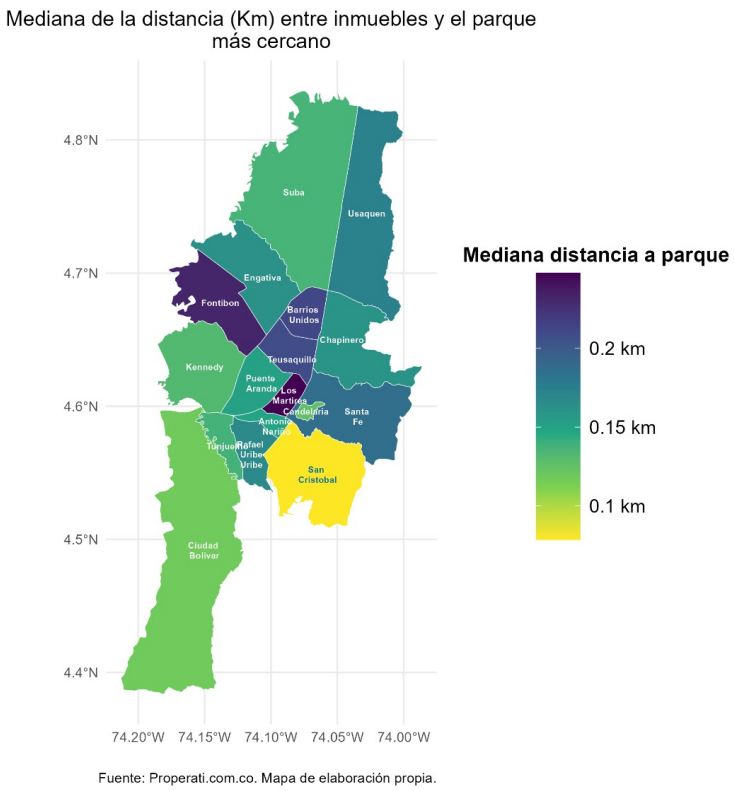
Tabla 7: Top 10 variables por importancia SHAP (media del valor absoluto).

Rank	Feature	MeanAbsSHAP	Gain	Cover	Frequency
1	area_m2_final	0.1434	0.2694	0.0474	0.0367
2	lon	0.0582	0.0590	0.0567	0.0448
3	banos_tot	0.0424	0.0345	0.0343	0.0202
4	lat_x_lon	0.0423	0.0221	0.0401	0.0323
5	bedrooms	0.0394	0.0359	0.0275	0.0243
6	bed_x_banos	0.0382	0.0275	0.0317	0.0294
7	log_area	0.0380	0.1265	0.0243	0.0149
8	estrato_num	0.0332	0.0287	0.0123	0.0110
9	dummy_terraza	0.0291	0.0078	0.0109	0.0066
10	lat	0.0269	0.0276	0.0351	0.0423

Fuente: Cálculos propios con base en Properati para Bogotá. Tabla de elaboración propia.

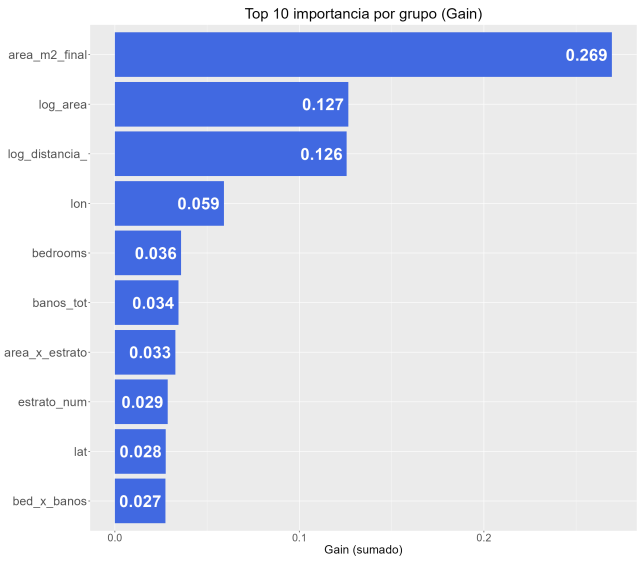


Mapa 1: Mediana del precio por m^2 en las localidades de Bogotá

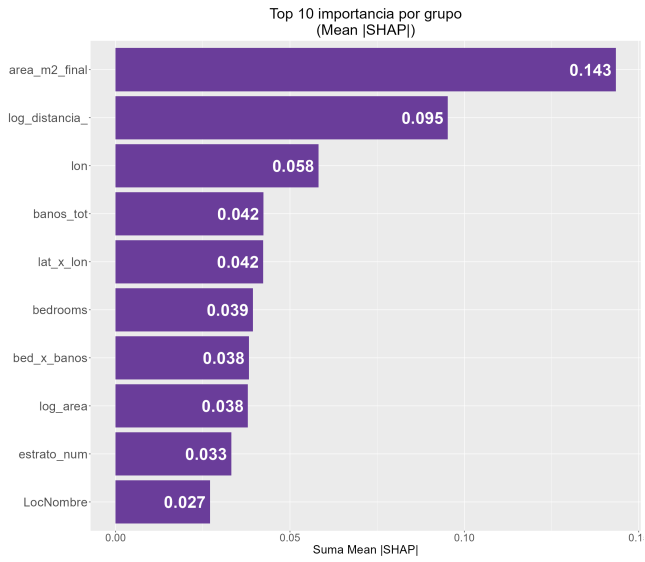


Mapa 2: Mediana de la distancia (km) entre inmuebles y el parque más cercano

Gráfica 1

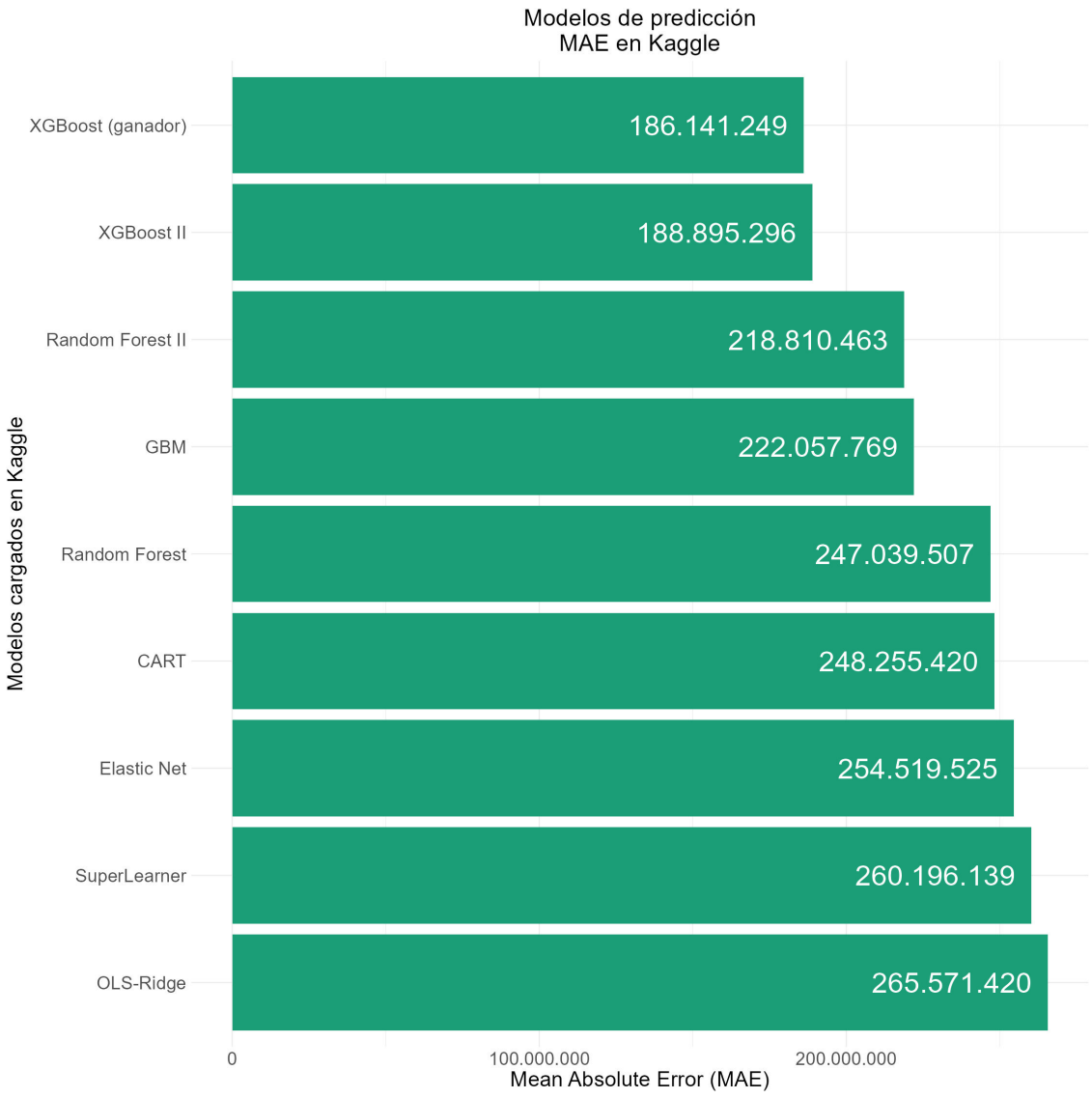


Gráfica 2



Fuente: Cálculos propios con base en Properati para Bogotá. Gráficas de elaboración propia.

Mapa 3: **Gráfica 3**



Fuente: Cálculos propios con base en datos de Properati para Bogotá. Gráfica de elaboración propia.