¿Hogar dulce hogar?: una metodología de machine learning para escoger vivienda en Chapinero

Mariana Correa, Rodrigo Iriarte, Marcel Montesdeoca, y Juan E. Moncada

Link repositorio: https://github.com/jmoncadal/taller_3

1. Introducción

El pronóstico de precios de vivienda es un reto en Bogotá, particularmente porque el mercado inmobiliario de la ciudad muestra una fuerte segmentación y una alta heterogeneidad, lo que complica la capacidad de predecir los precios con precisión. Estudios previos como el de Rincón y Robledo (2016) refuerzan la idea anterior, al poner en evidencia las problemáticas del mercado de vivienda bogotano debido a la insuficiencia espacial, el déficit cualitativo y, más importante aún, la alta segregación social. De manera similar, un factor que dificulta el ejercicio de pronóstico de los precios de vivienda en Colombia fue tratado por Cárdenas Rubio et al. (2019), quienes buscan solucionar las deficiencias en la disponibilidad y análisis de los precios de alquiler y venta de las viviendas en Colombia, subrayando la falta de datos útiles para pronosticar el precio de vivienda en la capital.

Con el propósito de aportar una solución, este documento propone un ejercicio de pronóstico usando distintas metodologías de Machine Learning para pronosticar el precio de vivienda en Chapinero. Para ello, se emplea un conjunto de datos de más de 38.600 observaciones con información estructural del inmueble, variables geográficas derivadas de OpenStreetMap, mediciones de accesibilidad, características del entorno urbano y variables de texto provenientes de la descripción del inmueble. La pertinencia de este conjunto de variables radica en su capacidad para capturar tanto atributos observables como elementos del contexto urbano que tradicionalmente afectan el precio de la vivienda.

La conclusión del estudio es que el mejor modelo de pronóstico de precios de vivienda para Bogotá es un Superlearner, el cual se caracterizó por otorgarle un alto peso a los modelos basados en árboles de decisión. El error absoluto promedio de este modelo fue de 223 millones de pesos, dato que, si bien se considera elevado, es una primera buena aproximación donde hay mucho potencial de mejora. En conjunto, los resultados evidencian que las metodologías no lineales y los esquemas de validación espacial son especialmente efectivos para modelar mercados inmobiliarios urbanos complejos, característica destacable del contexto bogotano y su alta heterogeniedad urbanística. Además, la incorporación de variables geográficas y de texto aporta mejoras sustantivas en la capacidad predictiva de los modelos.

2. Datos

Para llevar a cabo los distintos modelos de Machine Learning se utilizó una base de aproximadamente 38.650 observaciones sin duplicados tomada de Properati, un buscador de anuncios clasificados para venta y alquiler de inmuebles que funciona en Sudamérica. Para el alcance de este estudio, se escogió una muestra que reunía únicamente los apartamentos y casas en venta en Bogotá desde 2019 hasta 2021. Los datos extraídos resumen características físicas acerca de las viviendas en venta en Bogotá tales como: cantidad de cuartos, baños, superficie construida y tipo de vivienda. Además, se incluyeron variables socioeconómicas publicadas por la alcaldía de Bogotá en su portal de datos abiertos como: el estrato de la manzana donde se ubica el inmueble y la localidad de pertenencia. También, se optó por incluir datos de OpenStreetMap como la proximidad a amenidades como: parques, cafés, transporte público, y características adicionales de la zona como alumbrado público y si la vivienda se ubicaba en una zona residencial. Para obtener los datos espaciales, se descargaron los elementos geográficos de OpenStreetMap y para algunas variables creadas como cantidad de cafés en un área de 500 metros se tuvieron que transformar las unidades de grados a metros. En este orden de ideas, se crearon dos tipos de variables espaciales: de distancia y de cantidad de establecimientos dentro de un área. Las variables de distancia creadas fueron: distancia al café más cercano y distancia a estación de transporte. Las variables de cantidad en un área son: cantidad de cafés en 500 metros y cantidad de lámparas en 200 metros para ver que tan iluminada es la zona.

Finalmente, se creó una variable binaria que toma el valor de 1 si el inmueble está en una zona residencial y 0 de lo contrario.

Estas variables fueron seleccionadas a partir de una minuciosa revisión de literatura, donde se encontró que trabajos como el de Toloza-Delgado et al. (2021) hallan evidencia de que el estrato, la condición de entrega y el estado constructivo afectan el precio de manera lineal; mientras que el área, las distancias a parques, vías y estaciones de transporte público presentan resultados no lineales. También, autores como Azcarate-Romero et al. (2025), refuerzan la idea de que la cantidad de cuartos y la edad de la propiedad influyen positivamente en los precios de las viviendas, mientras que el estrato afecta negativamente el precio. Sin embargo, debido a la falta de datos útiles para pronosticar los precios de vivienda —siendo este un reto para las metodologías de machine learning, tal y como detallan Cárdenas Rubio et al. (2019)— se decidió explorar una metodología adicional, la cual buscaba explotar las descripciones de las viviendas para hallar información relevante no especificada dentro de las variables iniciales.

Así, mediante análisis de texto de las descripciones de los inmuebles detalladas en Properati, se incluyeron variables como si la vivienda contaba con terraza, parqueadero, seguridad privada, depósitos, cocina y sala o comedor, entre otras. Para ello, se diseñó un pipeline lingüístico basado en spaCy (modelo es_core_news_lg), lo que permitió realizar tokenización, lematización y clasificación morfosintáctica de más de dos millones de tokens provenientes de títulos y descripciones. Este procesamiento se complementó con regex especializadas para detectar patrones numéricos (“3 baños”, “dos alcobas”, “piso 7”, “120 m2”, etc.) y con diccionarios semánticos construidos ad hoc para identificar entidades relevantes (habitaciones, baños, terrazas, balcones, depósitos, proximidad a comercio, vigilancia, entre otros). Posteriormente, los tokens fueron organizados en una estructura ordenada, lo que permitió aplicar reglas de extracción basadas en contexto, como identificar números que preceden o siguen a una palabra clave. Para variables con potencial ambigüedad (p. ej., número de habitaciones), se combinó la información explícita del dataset con la información derivada del texto mediante un esquema de consistencia y plausibilidad. Finalmente, una vez construidas por separado la base estructural produciendo una base de entrenamiento (38.644 observaciones y 97 variables) y otra de prueba (10.286 observaciones y 96 variables). Estas bases robustas no solo contienen predictores tradicionalmente usados en modelos hedónicos, sino también atributos textuales derivados de descripciones inmobiliarias, lo cual permite capturar matices que reducen el error de pronóstico.

Desafortunadamente, el 47 % de las observaciones de cuartos, baños y piezas eran faltantes, sin contar con que casi el 80 % de valores faltantes para la superficie del apartamento no se tenían, siendo esto un punto débil, especialmente porque la literatura enseña que entre más cuartos y superficie construida el precio de la vivienda aumenta. Para solucionar este error, se decidió imputar los valores faltantes de los cuartos, baños y piezas. Para esto, se escogió agrupar las viviendas que tuvieran características físicas similares (como localidad, estrato, tipo de vivienda, año, mes) y se promedió la cantidad de habitaciones, baños, piezas y superficie que tenían. De esta manera, las viviendas que compartieran características físicas y geoespaciales similares, se les imputaría el promedio de cuartos que viviendas parecidas a estas tenían. Se reconoce que esta imputación puede ser un punto débil del manejo de datos, especialmente porque es una simplificación de la realidad. No obstante, se considera que es plausible pensar que apartamentos con características físicas similares como zona de ubicación, tipo de vivienda, fecha de construcción y localidad tengan características físicas comparables. Un argumento a favor de esta metodología de imputación es que la regulación actual sobre la construcción de viviendas establece unos lineamientos claros sobre el tipo y características de las construcciones que se pueden hacer dentro de cada localidad dadas las características establecidas en el Plan de Ordenamiento Territorial (POT). Por ejemplo, la regulación limita características como: área mínima habitable y estándar de metros cuadrados. Entonces, la posibilidad de que viviendas que comparten características como localidad de pertenencia y tipo de vivienda difieran grandemente en su cantidad de cuartos, baños y piezas es baja.

Para finalizar el tratamiento de la base, se trataron los outliers. Al graficar el precio por metro cuadrado, una de las variables creadas por su relevancia en la determinación del precio de la vivienda, se notaba que habían casas que presentaban valores altos relativos al resto de valores. Como las zonas en las que se ubicaban esos valores extremos (localidad, estrato y localidad) no empataban con el valor

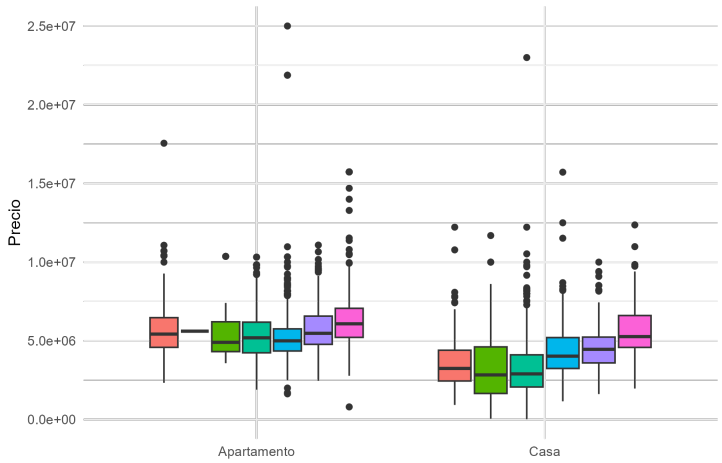
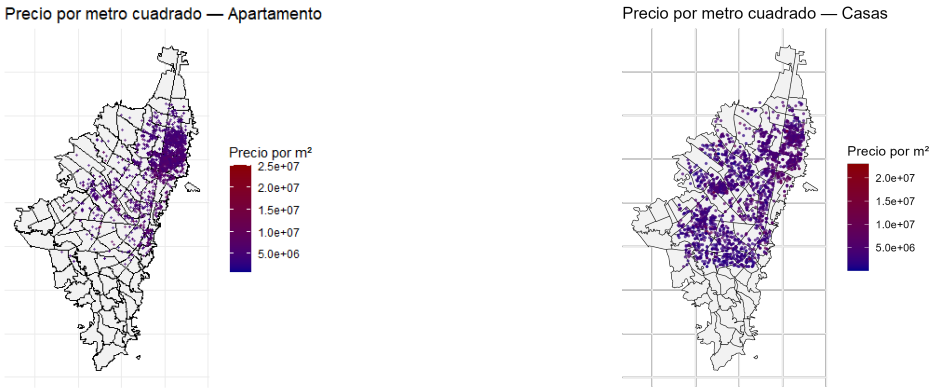


Figura 1: Distribución de precios por estrato y tipo de vivienda

de viviendas de similares características, se asumió que el valor estaba errado y se procedió a eliminar esas observaciones. Una vez se tuvo la base completa, se procedió a hacer un análisis descriptivo sobre las variables que se tenían. Para el objetivo del estudio, resultó interesante explorar la distribución de los precios por metro cuadrado por tipo de vivienda en Bogotá. Esta información se presenta en la Figura 1. Se aprecia que hay una alta heterogeneidad de precios en todos los estratos, mostrando valores atípicos en cada uno de los estratos evaluados. Además, los precios por metro cuadrado en apartamentos tienden a ser más altos para todos los estratos, menos el 6, donde presentan una distribucón similar. Esta alta heterogeneidad puede presentar una dificultad para el pronóstico de los precios, pues hay variables no observables que pueden estar afectando la estimación de los precios.

Además, al ver la concentración de las viviendas en Bogotá, se puede apreciar que hay también una alta heterogeneidad. Los apartamentos tienden a concentrarse hacie el nororiente de la ciudad, mientras que las casas se presentan a lo largo de toda la ciudad. Como se busca predecir Chapinero, puede ser útil enfocarse en los determinantes de precio para los apartamento, pues las áreas aledañas a esta localidad concentran más viviendas de tipo apartamento.



(a) Apartamentos (b) Casas

Figura 2: Precio por metro cuadrado en Bogotá según tipo vivienda

3. Modelos y resultados

3.1. Validación cruzada espacial

Con el fin de poder buscar la mejor especificación validación cruzada, se realizó un ejercicio. Este buscaba comparar de manera sistemática la validación cruzada tradicional (aleatoria) con esquemas de validación cruzada espacial en un contexto de precios de vivienda en Bogotá, donde la correlación espacial es relevante. Para ello se estimó un modelo lineal ordinario (OLS) con una especificación fija y se aplicaron tres esquemas de partición espacial sin buffer: por localidades y mediante celdas de grilla asignadas de manera aleatoria. El supuesto de partida, de acuerdo con la literatura de machine learning espacial, es que las observaciones no son independientes dada la correlación espacial por lo que una validación cruzada aleatoria sería inútil, e incluso dañina.

Los resultados muestran diferencias claras entre esquemas. Con validación espacial por localidades (18 localidades, excluyendo algunas con poca información o no relevante), el error absoluto medio (MAE) se ubicó alrededor de COP 223 millones, mientras que por localidad fue cercano a COP 220 millones y con grilla aleatoria descendió a aproximadamente COP 204 millones. Aunque el MAE más bajo del grilla podría interpretarse superficialmente como mejor desempeño, este resultado es consistente con demostraciones que señalan: a mayor correlación espacial, es más optimista el error dentro de la muestra; algo que no necesariamente se cumple out of sample. Es decir, en realidad es indicativo de sobre ajuste. A partir de esta evidencia, para la mayoría de las estimaciones de aquí en adelante se usa la validación cruzada por localidades. En otras palabras, las celdas de grilla aleatorias conducen a métricas engañosamente optimistas; la validación por localidad ofrece un corte más granular y un MAE ligeramente menor que el de localidades, pero al costo de potencialmente mantener una mayor dependencia espacial entre pliegues. Por todo lo anterior, la validación cruzada espacial por localidades es el método principal de validación cruzada, y se recomienda complementarla en trabajos futuros con buffers espaciales para reforzar la independencia espacial que intenta alcanzar.

3.2. Mejor Modelo (SuperLearner)

El mejor modelo fue un SuperLearner, caracterizado por tener una especificación más compleja. En total, se utilizaron 94 variables que resumían características físicas del inmueble como la cantidad de baños, piezas, cuartos, y si contaba con facilidades como terrazas, garajes, comedor y depósitos, entre otros. También, se incluyeron variables que describían el estado de la vivienda como el estrato, el precio del catastro y la localidad. El SuperLearner obtuvo un resultado dentro de la muestra de 223 millones de pesos, siendo una mejoría frente a lo obtenido por el mejor modelo anterior. Los pesos del SuperLearner se concentraron en los modelos de árboles de decisión, especialmente en el Random Forest. Como se discutirá más adelante, se le otorga principal importancia a estos modelos de árboles de decisión porque logran explotar, mejor que los otros modelos, las relaciones no lineales entre los datos.

3.2.1. Importancia de las variables

En el mejor modelo, las variables más determinantes para el SuperLearner son, sobre todo, las características internas del inmueble y un proxy fuerte de su valor base. `bathrooms_final` es, con diferencia, la variable más importante, por encima de `bedrooms_final` y de las dummies de estrato, lo que sugiere que el número de baños discrimina mejor la "calidad" y el segmento de mercado que el número de habitaciones. Un baño adicional suele ir asociado a mayor área útil, mejor diseño y un público objetivo de mayor ingreso, por lo que el modelo lo utiliza como un buen separador de precios altos y bajos. En los primeros lugares también aparece `precio_catastro_prom`, que captura buena parte de la información estructural y de localización que no es explícita en el set de variables: en la práctica, el SuperLearner parece está "ajustando" el avalúo catastral hacia el precio de mercado. A esto se suma `property_typeCasa`, que indica que, a igualdad de otras variables, las casas tienen una prima o un descuento sistemático frente a los apartamentos, algo esperable en una localidad con alta densidad y donde las casas suelen ubicarse en zonas muy específicas y de alto valor.

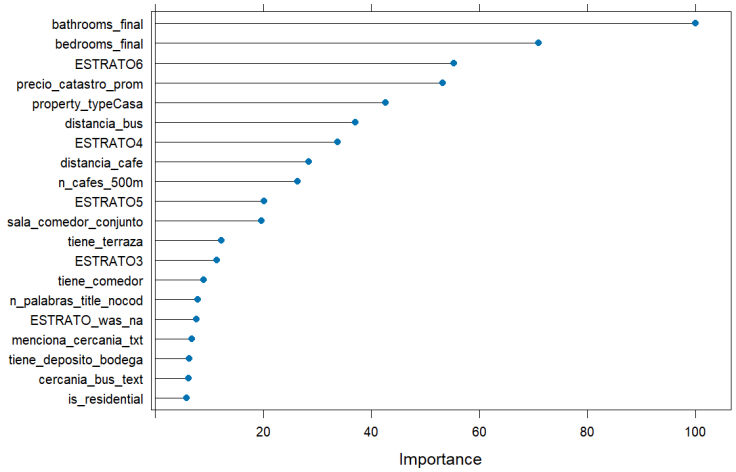


Figura 3: Importancia de variables

Un segundo grupo de variables relevantes combina la estratificación socioeconómica y la accesibilidad urbana. Dentro de los estratos, `ESTRATO6` aparece con mucha más importancia que `ESTRATO4`, `ESTRATO5` y `ESTRATO4`, lo cual indica que el modelo distingue con claridad el segmento más costoso, mientras que los demás estratos aportan información, pero de forma más marginal. En el bloque de accesibilidad, `distancia_bus`, `distancia_cafe` y `n_cafes_500m` tienen una importancia intermedia: el modelo recoge que la cercanía a paraderos de bus y a amenidades urbanas tipo cafés sí se capitaliza en el precio, pero no al nivel de las características internas o del estrato. La variable `ESTRATO_was_na` recuerda que incluso la "falta de dato" contiene información: los predios donde el estrato no estaba reportado parecen pertenecer a un submercado particular (por ejemplo, proyectos nuevos, usos mixtos o fichas mal clasificadas) con un patrón de precios distinto, que el algoritmo aprende a explotar.

Finalmente, en la parte baja del ranking aparecen variables que refinan la predicción, más que definir grandes saltos de precio: `sala_comedor_conjunto`, `tiene_terraza`, `tiene_comedor` y dummies textuales como `menciona_cercania_txt`, `cercania_bus_text` o el conteo de palabras del título. Su importancia indica que el modelo sí extrae señal de cómo se describe el inmueble en el anuncio: los vendedores tienden a enfatizar la cercanía al transporte y a servicios en las viviendas más atractivas, y esa información se traduce en mejor capacidad predictiva. Que `is_residential` esté al final sugiere que la base ya está muy concentrada en uso residencial y la variable apenas ayuda a separar algunos outliers comerciales. Desde el punto de vista de machine learning, esta jerarquía de importancias implica que, para simplificar el modelo, se sugiere conservar las variables de baños, habitaciones, catastro, tipo de inmueble, estrato y un puñado de distancias/amenidades y apenas se perdería precisión.

3.3. Modelo Regresión Lineal

Para el primer modelo, se utilizaron 23 predictores como: si el inmueble cuenta con espacios del tipo terraza, garaje, sala comedor conjunto, cocina integral, vigilancia y variables espaciales que se crearon del estilo distancia a estación de transporte y número de cafés en 500 metros, entre otras. Si bien un punto a favor de los modelos lineales es su baja varianza, característica que resulta útil para los pronósticos, su sobresimplificación de la relación entre los datos puede redundar en un pronóstico por lo general impreciso. Por ejemplo, autores como Toloza-Delgado et al. (2021), encuentran que variables como la condición de entrega y el estado constructivo afectan el precio de manera lineal, mientras que el área, las distancias a parques, vías y estaciones de Transmilenio presentan resultados afectan de manera no lineal.

El MAE estimado mediante validación cruzada espacial sobre el logaritmo del precio fue 0.337. Este error está expresado en unidades de log-precio, por lo que no es directamente comparable con el MAE

en nivel utilizado por Kaggle. El MAE en la muestra test, reportado por la competencia de Kaggle, fue de 300.707.935, un error bastante alto. El R^2 refleja que el modelo explica aproximadamente el 14 % de la variación del log-precio, lo cual es un valor muy bajo, enseñando que el modelo no recoge bien la relación entre los datos. Esto sugiere que el modelo lineal no es el más apropiado para predecir el precio de la vivienda, puesto que comparado con los otros modelos, ofrece el segundo peor MAE. Esto puede ser debido a la falta de no linealidades, las cuales se exploran en otros modelos más complejos como redes neuronales o árboles de decisión.

3.4. Modelo Elastic Net

Si bien el modelo lineal no ofreció la mejor aproximación, es útil ver cómo los modelos de regularización se comportan, a pesar de tener una especificación lineal similar a la del modelo anterior. Los modelos de regularización pueden tener un pronóstico más acertado. Como lo mencionan James et al. (2013), los modelos de regularización tienden a tener unos pronósticos acertados, puesto que la penalización ayuda a reducir la varianza de estos modelos relevante en un problema con tantas variables. Para el proceso de selección de las variables se escogieron 54 variables de acuerdo con la teoría y revisión de literatura, luego se hizo un forward selection y quedaron 23 variables; no obstante en este proceso se eliminaron algunas variables importantes por lo que se le adicionaron algunas variables consideradas importantes a nivel teórico. Al final, las variables utilizadas en el modelo incluyen características estructurales del inmueble (como el estrato socioeconómico, el tipo de propiedad, la presencia de terraza, balcón, garaje o sala-comedor), calidad del interior del inmueble (como cocina integral, remodelación o distribución abierta), variables de texto (como menciones a cercanía o descripciones específicas del inmueble), así como variables espaciales construidas a partir de OpenStreetMap, entre ellas la cantidad de cafés y lámparas en el entorno, indicadores de zona residencial y medidas de accesibilidad al transporte público, como la distancia y cercanía a paraderos de bus.

El Elastic Net, al ser una combinación de Ridge y Lasso, cuenta con dos parámetros para ser estimados: la penalización Lambda y Alpha. Así, se utilizó una grilla que combinaba distintos valores de Alpha y Lambda para dar con la mejor estimación minimizando el MAE. Entonces, se escogió un Alpha entre 0 y 1 que aumentara de a 0.05 unidades. De esta manera, después de evaluar por validación cruzada espacial por localidad, donde cada fold deja por fuera una localidad, el mejor alpha fue 0.5, logrando un balance entre Ridge y Lasso. Asimismo, al usar folds geográficos, se reduce la correlación espacial entre cada localidad. Así, el Elastic Net seleccionado representa el equilibrio entre complejidad y generalización espacial, capturando de manera más robusta las relaciones en los datos del precio de la vivienda frente al modelo lineal. Los resultados se resumen en la Tabla **??**. Este modelo, si bien presentó una mejora sustancial con respecto al modelo lineal, el error sigue siendo bastante grande, ubicándose en 287 millones de pesos.

3.5. Modelo CART

Una buena primera aproximación a la documentada no linealidad de los datos son los árboles de decisión, cuya estructura permite ajustarse a relaciones entre los datos más complejas. James et al. (2013) mencionan que los modelos de árboles pueden superar los pronósticos de los modelos lineales si la relación entre los datos es altamente no lineal y compleja. Sin embargo, un punto en contra de estos modelos es su alto sobreajuste, por lo que sus pronósticos pueden ser errados también. A pesar de esto, se procedió a estimar un árbol para predecir el precio y documentar si ofrecen una mejor predicción que los modelos de especificación lineal con la misma base del punto anterior, pero agregando variables de distancias a amenidades. La estimación si bien dio un gran resultado dentro de la muestra, fuera de la muestra tuvo uno de los peores comportamientos de todos los modelos probados. Así, se refuerza la idea de que los árboles sencillos dentro de muestra logran una predicción acertada, pero su sobreajuste concluye en pésimos resultados fuera de muestra. Concretamente, su error dentro de muestra fue de 158 millones, pero fuera de muestra fue de 254 millones. Sin embargo, se nota el beneficio de contar con metodologías que explotan la no linealidad de los datos, pues su error fuera de muestra presentó una mejoría con respecto a los modelos lineales.

3.6. Modelo Random Forest

Para continuar con los modelos no lineales y de árboles de decisión, se optó por desarrollar dos modelos Random Forest. En este algoritmo, se construyen varios árboles sobre una muestra bootstrap del conjunto de entrenamiento y se considera, en cada división, un subconjunto aleatorio de predictores para determinar cada partición. Así mismo, se estarían "descorrelacionando" los árboles (a diferencia de Bagging) y se reduce la varianza sin aumentar significativamente el sesgo. De esta manera, el modelo gana estabilidad y generalización, al evitar que todos los árboles dependan de los mismos predictores dominantes.

Para el primer Random Forest que se desarrolló, se incluyeron 32 predictores, dentro de los cuales hay linealidades, no linealidades e interacciones para mejorar la precisión. Como variable a estimar se utilizó el logaritmo del precio del inmueble. Esto se hizo debido a que la distribución del precio de los inmuebles está heterogénea en toda la muestra y dentro de los estratos, lo que genera errores más grandes. De esta manera, se buscó normalizar la muestra al reducir la asimetría de la distribución y reducir la influencia de outliers. Con respecto al entrenamiento del modelo, se llevó validación cruzada espacial por localidades. En este proceso, cada localidad se dejó por fuera en una iteración, usando todas las demás localidades para entrenar y reservando únicamente la localidad excluida para validar. El proceso se repitió tantas veces como localidades existían, garantizando que cada una funcionara como conjunto de validación una vez. Ahora bien, para el ajuste de hiperparámetros se especificaron 100 árboles; en el número de variables en cada partición se llevó a cabo una grilla y el resultado fueron 10 variables en cada partición del árbol; para el número mínimo de observaciones en cada nodo terminal se hizo por medio de una grilla que resultó en 30 como número mínimo de observaciones en cada nodo; finalmente el splitrule utilizado fue "variance"para reducir más la varianza en cada partición.

Para el segundo Random Forest (mejor que el anterior), se utilizó el precio como variable a estimar y los mismos predictores del modelo CART expuesto anteriormente. Se llevó a cabo una validación cruzada espacial por localidades, al igual que en el modelo anterior de random forest. Para el ajuste de hiperparámetros se especificaron 1000 árboles; en el número de variables en cada partición fueron 6 variables; para el número mínimo de observaciones fueron 40; finalmente se utilizó la impureza para mejorar la precisión de los árboles.

Los resultados de ambos modelos se presentan a continuación:

Cuadro 1: Resultados Modelos Random Forest

Parámetro	**RF 1**	**RF 2**
ntree	100	1000
mtry	10	6
min.node.size	30	40

Al comparar los dos modelos Random Forest, se observa un desempeño superior del segundo modelo (RF2) frente al primero (RF1). Aunque RF1 fue entrenado sobre el logaritmo del precio y obtuvo un MAE de 0.298 en validación cruzada espacial, su desempeño en la competencia de Kaggle fue considerablemente menor, alcanzando un MAE de 266.632.734 al volver al nivel del precio. Este deterioro en la precisión puede ser explicado por el sesgo introducido al retransformar el logaritmo a nivel, lo cual afecta especialmente inmuebles con precios altos. Por otro lado, RF2 fue entrenado directamente en nivel, logrando un error fuera de muestra sustancialmente menor, con un MAE de 229.437.438. Además, los hiperparámetros más conservadores del RF2 (como un número mayor de árboles, un min.node.size más grande y un mtry más bajo) favorecieron una reducción de la varianza y una mejor generalización espacial bajo validación cruzada por localidades. En conjunto, los resultados muestran que el RF2 captura de manera más efectiva las no linealidades e interacciones en los datos y constituye el modelo de Random Forest con mejor desempeño predictivo en Kaggle.

3.7. Modelo XGBoost

Ante el buen desempeño de los Random Forest, se exploró la posibilidad de incluir un modelo de XGBoost, el cual combina los árboles con métodos de regularización para penalizar los modelos con alta complejidad, reduciendo así su sobreajuste. Para llevarlo a cabo, se utilizaron 32 predictores, donde, a diferencia de los otros modelos, se agregaron variables relacionadas con distancias como: distancias a centros comerciales y supermercados, entre otras. Esta decisión se tomó porque los modelos de Random Forest tendían a sobreajustarse, limitando el poder predictivo del modelo. Así, para mejorar la predicción, se decidió limitar la cantidad de variables que se estaban tomando. En cambio, como el XGBoost penaliza la complejidad de los modelos, se pensó en agregar estas nuevas variables tanto para aprovechar la no linealidad, como la penalización por sobreajuste al incluir más variables. Para hallar los hiperparámetros óptimos de los modelos, se aplicó una grilla que modificara la cantidad de rondas, la profundidad de cada árbol, y los hiperparámetros de penalización. Al final, los resultados de la estimación dieron los siguientes resultados:

Cuadro 2: Resultados Modelos XGBoost

Parámetro	**XGBoost**
nrounds	250
max_depth	4
eta	0.05
gamma	0.5
min_child_weight	10
subsample	0.4

3.8. Redes Neuronales

Continuando con modelos más complejos se desarrollaron 2 modelos de redes neuronales. El primero de ellos está compuesto por una capa de entrada, una capa oculta con activación de ReLU, ya que mejora la estabilidad del gradiente y captura no linealidades; y una capa de salida. Este tipo de red permite capturar relaciones altamente no lineales entre las variables explicativas y el precio, sin necesidad de especificar interacciones o transformaciones complejas. El conjunto de predictores incluye 17 variables seleccionadas por relevancia teórica y empírica las cuales son: variables estructurales del inmueble (como estrato, vigilancia, remodelación, tipo de propiedad, número de habitaciones y baños), características del entorno (como residencial, cantidad de cafés y iluminación de la zona).

Para el ajuste de hiperparámetros se implementó una validación cruzada espacial por localidades, garantizando que el modelo aprenda a generalizar de manera espacial, evitando que la red neuronal se aprenda las idiosincracias de una localidad en específico. La selección del número óptimo de neuronas se realizó a través de una grilla y se encontró que lo óptimo eran 21 neuronas. Finalmente, el modelo se entrenó con el optimizador Adam, utilizando 30 épocas y el tamaño del batch de 64. Este modelo fue la aproximación al teorema visto en clase que anunciaba que con un modelo de una capa se podrían capturar las idiosincracias de las funciones.

La segunda red neuronal se hizo con el objetivo de reducir el MAE en la muestra mejorando los errores del modelo anterior. Se utilizaron los mismos predictores y se agregaron más variables como el área del inmueble, el tipo de cocina, las habitaciones, distancia a cafés, entre otras, terminando con 24 predictores. Para el ajuste de hiperparámetros también se hizo a través de la validación cruzada espacial. De esta manera, se hizo una búsqueda en grilla sobre número de capas ocultas y el número de neuronas por capa, resultando en 8 y 18 respectivamente. Por otro lado, se añadió early-stopping con un nivel de tolerancia =3 para que el entrenamiento se detenga cuando el MAE en el conjunto de validación no mejora durante tres épocas consecutivas. De esta manera, se evita continuar entrenando una red que ya no está aprendiendo patrones nuevos y se reduce el riesgo de sobreajuste. Además, al activar la opción *restore_best_weights = TRUE*, los pesos finales del modelo corresponden a la época con el mejor desempeño en validación. No sobra mencionar que, el modelo también se entrenó con el optimizador Adam, utilizando 60 épocas y el

tamaño del batch de 64.

La siguiente tabla muestra los resultados de ambos modelos:

Cuadro 3: Resultados de las redes neuronales

Parámetro	**NN 1**	**NN 2**
Capas ocultas	1	8
Neuronas por capa	21	18
Función de activación	ReLU	ReLU
Optimizador	Adam	Adam
Learning rate	0,001	0,0005
Batch size	64	64
Epoch	30	60
Early stopping	No	Sí (patience = 3)

Al comparar el desempeño de las dos redes neuronales, se observa una mejora sustancial en el segundo modelo tanto en el ajuste dentro de muestra como en la capacidad de generalización. Mientras la NN 1 obtuvo un MAE in-sample de 654.520.853 y un MAE en Kaggle de 873.550.194 COP, la NN 2 redujo estos valores a 159.432.656 y 255.310.794, respectivamente. También se puede decir que, la brecha entre el error en la muestra de entrenamiento y el error fuera de muestra se redujo en la NN 2, evidenciando un menor sobreajuste. Esta mejora puede ser explicada debido a la red más profunda, con 8 capas ocultas y 18 neuronas por capa, que permite capturar no linealidades más complejas en las relaciones espaciales y estructurales del precio de vivienda. Por otra parte, la inclusión de variables adicionales como superficies del inmueble o distancias a cafés puede mejorar la capacidad predictiva del modelo. Finalmente, añadir el early stopping con tolerancia de 3 épocas junto con un learning rate más bajo, estabiliza el entrenamiento y previene el sobreajuste. De esta manera, la NN 2 supera a la NN 1 y muestra un comportamiento más robusto y generalizable para predecir precios de vivienda en Chapinero.

3.9. Superlearners

Por último, se quiso explorar un superlearner, donde se combinaran todos los modelos anteriores y el resultado fuese un pronóstico, potencialmente, más acertado. Con este propósito, se hicieron 2 modelos. Uno de ellos fue el mejor modelo y se explicó anteriormente. En este apartado se profundizará sobre el superlearner cuyo resultado no fue el mejor. Este modelo combina varios modelos base mediante mínimos cuadrados no negativos (NNLS). El Superlearner combina predicciones "out-of-fold" de cada modelo base, es decir, predicciones obtenidas sobre observaciones que no fueron usadas durante su entrenamiento. Estas predicciones permiten estimar el desempeño real de cada modelo sin sesgo de sobreajuste. Con ellas, el metalearner (a través de NNLS) estima pesos que minimizan el error de validación. En consecuencia, el superlearner genera una predicción final seleccionando automáticamente la mezcla óptima entre los learners individuales.

Para este superlearner, los pesos reportados fueron: Random Forest: 0.696, GLM lineal: 0.304 y 0 para los demás. Esto muestra que el desempeño óptimo se logra combinando el random forest con un componente lineal (GLM). Esta combinación de random forest (baja varianza) y un GLM (bajo sesgo) tiene sentido ya que captura relaciones mayormente no lineales con un componente lineal (por ejemplo estrato). El superlearner obtuvo un MAE in-sample de 164.446.525 y un MAE en Kaggle de 262.589.948, con una brecha entre ambos valores un poco alta, lo que puede sugerir sobreajuste. El desempeño de este modelo es competitivo y muy cercano al de la mejor red neuronal 2, quedando ligeramente por detrás en la competencia de Kaggle.

3.10. Análisis comparativo

Comprender las métricas de desempeño es fundamental para interpretar la capacidad predictiva de los modelos. Ese es el propósito del Cuadro 3 que muestra el MAE de cada modelo:

Cuadro 4: Resultados generales de desempeño de los modelos

Modelo	**MAE in-sample**	**MAE Kaggle**
Superlearner 2 (BEST)	124.104.381	223.571.326
Regresión lineal	0,3373*	300.707.935
Elastic Net	245.772.907	287.891.344
CART	158.531.931	254.626.606
Random Forest 1	0,2983*	266.632.734
Random Forest 2	132.522.834	229.437.438
XGBoost	184.768.188	224.562.574
Neural Network 1	654.520.853	873.550.194
Neural Network 2	159.432.656	255.310.794
Superlearner 1	164.446.525	262.589.948

* en escala logarítmica

Al comparar el desempeño de los modelos, se observa una diferencia fundamental entre los métodos lineales y los no lineales. Los modelos basados en árboles de decisión muestran desempeños sustancialmente mejores que la regresión lineal y el Elastic Net, confirmando que la relación entre las características del inmueble, las variables geográficas y el precio es no lineal. El modelo Random Forest 2 supera con al Random Forest 1, lo que indica que configuraciones más profundas y con mayor capacidad predictiva capturan mejor la heterogeneidad espacial. En la misma línea, XGBoost obtiene uno de los mejores desempeños en la competencia de Kaggle.

Por otro lado, el desempeño de las redes neuronales revela un contraste importante. La NN1 muestra un sobreajuste, con un salto entre el MAE in-sample y en Kaggle. Sin embargo, la NN2 (con mayor profundidad, más variables y regularización mediante early stopping) logra reducir de forma notable tanto el error en la muestra de entrenamiento como el error en Kaggle, también con un buen desempeño. El Super Learner 1, basado en una combinación de Random Forest y GLM, obtiene un buen desempeño y cercano al de NN2; sin embargo, no alcanza la precisión de XGBoost. Los resultados sugieren que la combinación óptima de modelos no lineales mejora la generalización espacial. Es por esto que, el análisis evidencia que los métodos no lineales y, especialmente, los modelos de Random Forest y XGBoost, ofrecen el mejor poder predictivo para estimar precios inmobiliarios en un contexto urbano complejo como Bogotá.

4. Conclusión

En general, predecir los precios de vivienda en Chapinero es un objetivo difícil. Particularmente, la alta heterogeneidad de los datos, sumado a la compleja relación urbanística de Bogotá, impide que las tareas de imputación, estimación y predicción de los precios sean sencillas. Concretamente, la atipicidad y alta variabilidad de los precios por metro cuadrado por estrato y tipo de vivienda sugieren que hay características no observables difíciles de generalizar que afectan fuertemente los precios de venta de viviendas en Bogotá. Sumado a esto, la falta de datos útiles para pronosticar los precios de vivienda, documentado en la literatura, agregan complejidad a la labor de predicción.

De todos los modelos realizados, destaca el desempeño de: el segundo SuperLearner, el XGBoost y el Random Forest. Se considera que la complejidad de estos modelos, sumado a su capacidad para recoger relaciones no lineales entre los datos fueron fundamentales para su mejor desempeño. En cambio, los peores modelos fueron: la primer red neuronal, la regresión lineal, y el Elastic Net. Nuevamente, la especificación lineal de los últimos dos modelos impide que sean un buen predictor de precio de vivienda en Chapinero, aunque destaca el aporte de los métodos de regularización para mejorar los pronósticos. El mejor modelo de todos fue el segundo Super Learner, cuyo error fuera de muestra es de 223 millones de pesos. Su comparativo alto desempeño fue atribuido a unas mejores especificaciones del modelo, junto con la adición de variables que resumían características físicas de los inmuebles y alrededores. Además, el alto peso dado a los modelos de Random Forest y XGBoost, sugiere que son los mejores modelos

para recoger no linealidades y predecir dadas las variables con las que es trabajó. Las variables de mayor importancia, como era de esperar, eran aquellas que se relacionaban con las características del inmueble como baños y cuartos, además de otras documentadas por las características como el estrato y la distancias a amenidades.

4.0.1. Limitaciones, aprendizajes y oportunidades futuras

En lo que respecta a las limitaciones, los resultados mostraron que las principales restricciones provinieron de la ingeniería de datos. La eliminación involuntaria de variables geoespaciales relevantes, la imputación poco estructurada y la falta de un pipeline reproducible redujeron la calidad de la base y limitaron el rendimiento de los modelos. En este sentido, durante el proceso se perdió información relevante, se reconstruyeron capas espaciales varias veces y la imputación necesitó varios ajustes. El equipo no logró anticipar la complejidad del problema, tanto a nivel computacional como teórico. Estos factores impidieron implementar herramientas más avanzadas, como buffers espaciales, técnicas de text mining basadas en Machine Learning o modelos geográficos específicos.

A pesar de ello, vale la pena resaltar que sí existieron aprendizajes significativos. El ejercicio permitió comprender con estremecedora claridad cómo las formas funcionales de cada algoritmo afectan su capacidad para capturar no linealidades, que demostraron ser notablemente diferentes entre modelos. En línea con lo anterior, se dieron aprendizajes valiosos en la optimización de hiperparámetros y su impresionante poder in-sample. Se resalta especialmente mediante el uso de grillas compuestas. Otra lección relevante que, a riesgo de sonar poco técnica, es profundamente formativa, fue la de aprender a gestionar la frustración: detrás de cada modelo hubo múltiples iteraciones, ajustes y combinaciones de variables. Entender que este esfuerzo no siempre se traduce en un buen score, pero que la clave es seguir adelante, constituye una enseñanza. Teniendo en cuenta lo anterior, el proyecto, aun con sus restricciones, produjo resultados relevantes e interesantes, además de valiosas reflexiones.

Para fortalecer el proyecto se propone un pipeline y gobierno de datos con validaciones sistemáticas que eviten pérdida de información, protocolos de imputación más avanzados (como MICE o random forest imputer) comparados técnicamente para decisiones informadas, trazabilidad completa y versionamiento, además de un trabajo organizado en distintos scripts y documentos para garantizar orden y reproducibilidad. Asimismo, se plantea la ampliación de fuentes de información mediante un inventario exhaustivo (OSM, Catastro, IDECA, POT, movilidad, comercio, ruido, crimen), el uso de modelos de machine learning para extracción y clasificación de texto, y un enfoque colaborativo que aproveche el conocimiento comunitario y del equipo académico. Finalmente, se propone un plan metodológico robusto con tareas estructuradas y fechas de corte ambiciosas, validación espacial con buffers desde el inicio y comparación sistemática entre modelos espaciales y no espaciales. La experiencia y sus resultados confirmaron que la calidad del proceso determina la calidad del modelo: un pipeline sólido, fuentes amplias y una planificación rigurosa son esenciales. Pero también hay componentes humanos como lo son la gestión emocional, resiliencia, trabajo en equipo. Ambos aspectos se deben reconocerse como parte integral del desarrollo de este tipo de trabajos.

Referencias

Azcarate-Romero, J., Toloza-Delgado, J., Cruz Gutierrez, N., & Mahecha, P. (2025). Urban spatial analysis of the profitability of housing rental in Bogotá. *International journal of housing markets and analysis*.

Cárdenas Rubio, J. A., Chaux Guzmán, F. J., & Otero, J. (2019). Una base de datos de precios y características de vivienda en Colombia con información de Internet. *Revista de economia del Rosario*, *22*(1), 25-99.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning : with Applications in R* (1st ed. 2013.). Springer New York.

Rincón, M. C., & Robledo, J. C. (2016). Análisis de la política de vivienda en Bogotá: un enfoque desde la oferta y la demanda /An analysis of housing policy in Bogotá: a supply and demand perspective/Análise da política habitacional em Bogotá: um enfoque a partir da oferta e da demanda. *Revista finanzas y política económica*, *8*(1), 105-.

Toloza-Delgado, J., Melo-Martínez, O., & Azcarate-Romero, J. (2021). Determinants of New Housing Prices in Bogotá for 2019: an Approach Through a Semiparametric Spatial Regression Model. *Ingeniería y ciencia (Medellín, Colombia)*, *17*(34), 23-.