Universidad de los Andes

Big Data y Machine Learning

Ignacio Sarmiento-Barbieri

Problem set 3 - Making Money with ML?

Repositorio: https://github.com/GeorgeWton1986/T3_BDML

Elaborado por:
Laura Sarif Rivera Sanabria
Jorge Eliecer Viafara Morales
Nicolas Jacome Velasco
Zaira Alejandra Garcia
Bernal

27 de mayo de 2025

1. Introducción

La valoración de inmuebles ha sido objeto de estudio desde mediados del siglo XX. Investigaciones pioneras como las de Bailey, Richard y Hugh (1963) analizaron el comportamiento del mercado inmobiliario en función de variables descriptivas como la calidad y la localización de las viviendas, mientras que Alonso (1964), citado en Linneman (1982), introdujo la teoría del Bid Rent, que explica cómo los precios de las propiedades varían según su distancia al centro económico de una ciudad. Estos estudios sentaron las bases para enfoques más estructurados como el modelo de precios hedónicos propuesto por Rosen (1974), el cual plantea que los bienes diferenciados, como las viviendas, pueden representarse como una combinación de atributos que determinan su valor de mercado. La literatura contemporánea ha expandido esta visión incorporando dimensiones espaciales y socioeconómicas (Glaeser et al., 2014; Brueckner, 2011), mientras que en Colombia investigaciones como las de Morales et al. (2013) y Revollo (2009) han mostrado la relevancia de las características estructurales, urbanas y sociales en la valoración del suelo.

En este contexto, el objetivo de este trabajo es predecir el precio de venta de viviendas en la localidad de Chapinero, Bogotá, mediante la aplicación de diversos algoritmos avanzados de aprendizaje automático. Para ello, se utiliza una base de datos proveniente de Properati, complementada con variables geográficas respaldadas por la literatura especializada y con atributos derivados del procesamiento de texto de las descripciones de las propiedades. Tras evaluar el desempeño de los modelos con MAE, se concluye que el algoritmo de Random Forest ofrece la mayor precisión en la predicción de precios de venta, posicionándose como la alternativa más robusta para este tipo de análisis.

2. Datos

La metodología para la construcción de la muestra se presentará conforme su desarollo en R-studio.

1. Configuración del entorno de trabajo: Se estableció el entorno de trabajo en R, cargando los paquetes necesarios mediante el uso de las funciones `require("pacman")` y `p_load()`, lo que permitió gestionar de forma eficiente las herramientas requeridas para la manipulación de datos, análisis espacial, procesamiento de texto y modelado predictivo.

2. Imputación de valores faltantes y tratamiento de valores atípicos: Se realizó la imputación de valores faltantes en las variables `area_privada`, `area_total`, `habitaciones` y `baños`, utilizando la mediana para variables continuas y la moda para variables categóricas. Asimismo, se identificaron y trataron los valores atípicos en la variable `precio`, aplicando un recorte en el percentil 99 % con el fin de mitigar el efecto de precios extremos que podrían sesgar los modelos.

3. Extracción de variables desde texto: A partir del procesamiento de las descripciones textuales de las propiedades, se generaron variables binarias que indican la presencia o ausencia de ciertas características relevantes, como `parqueadero`, `terraza`, `gimnasio`, `lavandería` y `ascensor`. Estas variables permiten capturar información que no estaba explícitamente estructurada en el conjunto original.

4. Incorporación de variables espaciales: Se calcularon distancias geográficas desde cada propiedad hacia puntos de interés para las personas como centros comerciales, CAI, paraderos de buses y avenidas principales, utilizando coordenadas geográficas y funciones de distancia del paquete `geosphere`. Estas variables permiten introducir la dimensión espacial dentro del modelo predictivo.

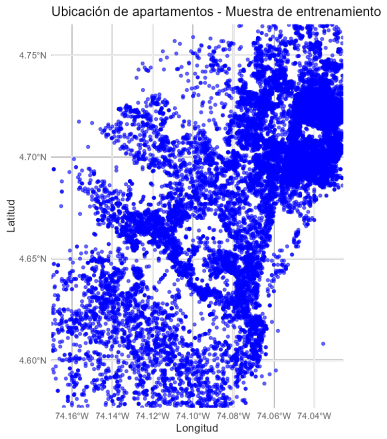

Figura 1: Mapa con viviendas de Bogotá

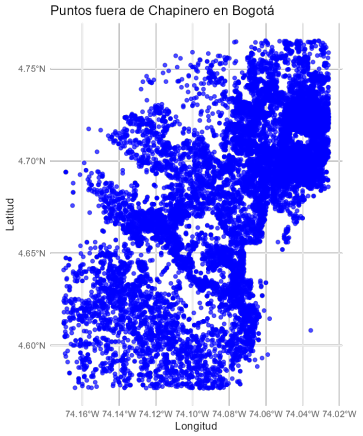

Figura 2: Mapa de viviendas de Bogotá sin Chapinero

Ahora, se presenta un análisis exploratorio del conjunto de datos utilizado para modelar el precio de propiedades residenciales en Bogotá. El objetivo es comprender cómo las distintas variables se relacionan con el valor de mercado y qué implicaciones tienen para la construcción del modelo predictivo.

Cuadro 1: Estadísticas descriptivas de las variables

#	Variable	Media	Mediana	Min	Max	Des.st
1	price	647	550	300	1,650	300,259
2	year	2020.29	2020	2019	2021	0.76
3	distancia_cai	719.28	683.32	1.53	2167.46	365.44
4	distancia_mall	584.62	516.94	1.21	3807.57	362.85
5	distancia_bus	208.02	182.49	0	2668.31	131.27
6	distancia_avenida	411.40	284.09	0	3239.26	412.76
7	binaria_parking	0.69	1	0	1	0.46
8	binaria_terrace	0.39	0	0	1	0.49
9	binaria_gym	0.21	0	0	1	0.41
10	binaria_laundry	0.23	0	0	1	0.42
11	binaria_elevator	0.15	0	0	1	0.36

El precio (cifras en miles)promedio de las viviendas es de aproximadamente 647 millones de pesos colombianos, con una mediana de 550 millones, lo que indica una ligera asimetría hacia precios más altos. Esta diferencia sugiere la existencia de propiedades de lujo que elevan el promedio. Además, la alta desviación estándar (300 millones) confirma una marcada heterogeneidad en los precios del mercado inmobiliario, lo que refuerza la necesidad de incluir variables explicativas que capturen su contexto urbano.

Las variables relacionadas con accesibilidad geográfica permiten evaluar el entorno de cada propiedad. La distancia a estaciones de transporte público, centros comerciales, avenidas principales y puestos de policía (CAI) varía considerablemente entre viviendas. Esto refleja la diversidad urbana de Bogotá, donde coexisten zonas céntricas, bien conectadas y seguras, con áreas periféricas más aisladas. En términos prácticos, estas variables nos permiten aproximar la conectividad y la percepción de seguridad, dos factores clave en la valorización inmobiliaria. Por ejemplo, la cercanía a una avenida principal puede ser deseable por su accesibilidad, pero también puede

generar ruido o contaminación, lo cual exige modelar estos efectos de forma no lineal o interactiva.

En cuanto a las características estructurales, se incorporaron variables binarias que capturan la presencia de ciertos atributos del inmueble. Aproximadamente el 69 % de las propiedades cuenta con parqueadero, lo que indica que este es un atributo relativamente común, pero aún diferenciador. Por otro lado, solo 15 % dispone de ascensor, lo cual podría señalar que la mayoría de las construcciones son de baja altura, o que se encuentran en zonas donde este tipo de infraestructura no es común. La presencia de terraza, gimnasio, lavandería y otras amenidades también muestra una amplia variación, lo que puede reflejar diferencias socioeconómicas entre sectores, o distintos tipos de desarrollos (por ejemplo, conjuntos cerrados frente a edificios tradicionales).

El año de construcción tiene una media cercana a 2020, con poca variabilidad, ya que el rango va de 2019 a 2021. Esto sugiere que la muestra se compone principalmente de edificaciones recientes, lo cual puede minimizar los efectos de depreciación estructural, pero también introduce un sesgo hacia inmuebles nuevos o en proceso de comercialización.

En conjunto, el diseño del dataset busca capturar tanto la calidad intrínseca de los inmuebles (a través de amenidades y año de construcción) como su ubicación relativa dentro de la ciudad (por medio de distancias a puntos de interés). Esta combinación es coherente con la teoría hedónica de precios, la cual postula que el valor de una propiedad se explica por el conjunto de sus atributos individuales y su entorno.

Este análisis evidencia que las variables seleccionadas no son meros indicadores descriptivos, sino piezas clave para entender la lógica del mercado inmobiliario urbano. Su inclusión está respaldada tanto por fundamentos económicos como por un conocimiento práctico de las dinámicas residenciales en Bogotá, lo que fortalece la solidez del modelo predictivo.

3. Modelo y resultados

Esta sección describe cómo fue el proceso de predicción de precios de vivienda para Chapinero, Bogotá con Random Forest como el mejor modelo a partir del resultado del MAE. En los anexos se encuentra el detalle del entrenamiento de seis algoritmos diferentes para la estimación de precios como: Elastic Net, Cart, Boosting, Redes Neuronales, Super Learners y Regresión lineal.

Para entrenar el modelo Random Forest, se implementaron dos estrategias complementarias: un entrenamiento tradicional con validación tipo out-of-bag (OOB) y una validación cruzada espacial, cada una con criterios diferenciados para la selección de hiperparámetros.

En el entrenamiento tradicional, se utilizó el motor `ranger` a través de la función `train()` del paquete `caret`, con validación interna basada en OOB. Esta técnica permite estimar el error de generalización sin necesidad de particionar explícitamente los datos, aprovechando el muestreo con reemplazo propio del Random Forest. Para este enfoque, se definió un grid de hiperparámetros restringido: `mtry = 5`, que determina el número de variables candidatas en cada división del árbol. Este valor se considera razonable en el contextos actual con un número moderado de 12 predictores, lo cual promueve la diversidad entre árboles y reduce la correlación dentro del bosque; `splitrule = variance`, siendo adecuada para la regresión, ya que minimiza la varianza dentro de los nodos hijos; y `min.node.size = {2, 3, 4}`, es posible controlar la profundidad de los árboles, de esta forma se confirma que los valores más bajos permiten capturar mayor detalle de los datos, aunque con un mayor riesgo de sobreajuste.

La validación cruzada espacial se llevó a cabo mediante el paquete `spatialsample`, el cual permite construir pliegues geográficamente excluyentes. Primero, se preparó la base de entre-

namiento excluyendo el identificador `property_id`. Luego, se construyó una receta de preprocesamiento que normalizó las variables continuas espaciales (`distancia_cai`, `distancia_mall`, `distancia_bus`, `distancia_avenida`), se crearon variables ficticias para predictores categóricos y se eliminó predictores con varianza cero.

El modelo de Random Forest se definió usando el motor `ranger` dentro del framework `parsnip`, estableciendo `mtry` y `min_n` como hiperparámetros a sintonizar, junto con un total de 500 árboles. El grid de hiperparámetros se generó con `dials`, explorando combinaciones entre 2 y 10 para ambos parámetros. A continuación, se creó un *workflow* uniendo la receta y el modelo.

Para la validación espacial, se unió la base de entrenamiento con su geometría y se aplicó la función `spatial_block_cv()` con cinco pliegues. Este procedimiento garantiza que cada pliegue contenga observaciones de una región geográfica diferente, reforzando la independencia espacial entre entrenamiento y prueba.

La sintonización del modelo se realizó mediante `tune_grid()` usando `MAE` como métrica de evaluación. Los mejores hiperparámetros se seleccionaron con `select_best()` y se aplicaron al *workflow* final, que luego fue ajustado a toda la base de entrenamiento. Finalmente, se evaluó la importancia de variables mediante el paquete `vip`, lo que permitió identificar los predictores más relevantes para el modelo.

La función estimada para el mejor modelo puede expresarse de la siguiente manera:

$$\begin{aligned}\text{Precio}_i = f(&\text{Año}_i, \text{TipoPropiedad}_i, \text{DistCAI}_i, \text{DistCentroComercial}_i,\\ &\text{DistParaderoBus}_i, \text{DistAvenida}_i, \text{Parqueadero}_i,\\ &\text{Terraza}_i, \text{Gimnasio}_i, \text{Lavandería}_i, \text{Ascensor}_i) + u_i\end{aligned}$$

donde $f(\cdot)$ representa la función de entrenamiento por el modelo Random Forest para capturar la relación entre las variables explicativas y el precio de venta de las viviendas.

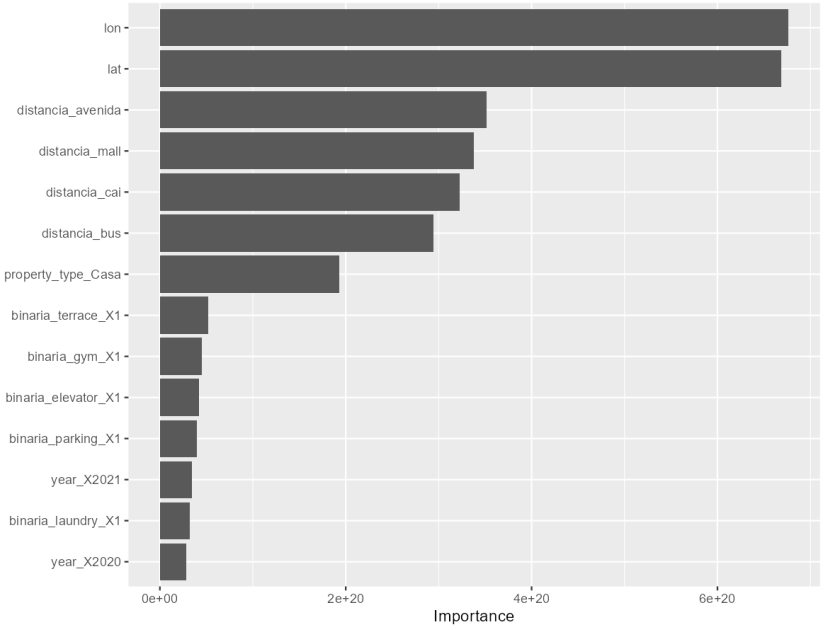

Figura 3: Importancia de las variables en Random Forest

El análisis de importancia de variables, extraído del modelo Random Forest ajustado con validación cruzada espacial, revela que las variables más influyentes en la predicción del precio

de venta son de naturaleza tanto estructural como contextual. En primer lugar, `dist_avenida` y `dist_CentroComercial` destacan por su fuerte relación inversa con el precio, lo que sugiere que una mayor cercanía a vías principales y centros comerciales incrementa significativamente el valor de la propiedad, en línea con la teoría hedonística del precio.

Asimismo, características obtenidas de la descripción de venta de los inmuebles como `binaria_parking` y `binaria_elevator` también muestran alta importancia relativa, indicando que las propiedades con parqueadero y ascensor tienden a tener precios superiores. Entre las variables estructurales, el `año` de la propiedad y su `tipo` (`property_type`) aportan información clave para discriminar precios según antigüedad y categoría. Este resultado empírico respalda la relevancia de combinar variables tradicionales, derivadas de texto y espaciales para capturar adecuadamente las dinámicas del mercado inmobiliario en un contexto urbano como Chapinero en Bogotá.

Sin embargo, la función presentada anteriormente presenta limitaciones en la granularidad de sus datos, al considerar únicamente las amenidades de la propiedad y omitir variables intrínsecas cruciales como el número de habitaciones, baños o las dimensiones de las áreas. La elección metodológica de entrenar los modelos sin estas variables se sustentó en la necesidad de validar una hipótesis fundamental: si la variabilidad en el precio de la vivienda puede explicarse en ausencia de sus características intrínsecas. En contraste con esta premisa inicial, los hallazgos obtenidos en la competición de Kaggle evidencian que las predicciones que contemplan variables intrínsecas de la vivienda, junto con parámetros geográficos y variables derivadas del procesamiento de texto, presentan un poder predictivo significativamente superior. A pesar de esta distinción, las estimaciones de los modelos que emplean el conjunto completo de variables están disponibles para consulta en el repositorio de GitHub.

La comparación entre ambos enfoques muestra un contraste importante: mientras el entrenamiento tradicional permite ajustes rápidos y eficientes con métricas competitivas, el entrenamiento con validación cruzada espacial aporta mayor robustez y realismo en contextos geográficos, permitiendo al modelo generalizar mejor en zonas no observadas. Esto se traduce en una mejora sustancial del rendimiento fuera de muestra, especialmente en conjuntos de datos de evaluación como los de Kaggle, donde los patrones espaciales no observados previamente afectan significativamente la capacidad predictiva del modelo.

Cuadro 2: Comparación de los 10 mejores modelos predictivos según MAE en el conjunto de prueba

#	Modelo	MAE test
1	Random Forest 2	279.650.891,61
2	Regresión lineal 1	297.860.686,66
3	Red neuronal 4	315.492.284,76
4	CART 2	317.663.115,63
5	CART 1	317.663.115,635
6	Red Neuronal 3	331.175.086,17
7	Random Forest 1	343.102.214,1
8	Boosting 2	343.892.200,75
9	Elastic Net	346.112.282,23
10	SUPER LEARNER 1	346.344.845,12

Como se describio anteriormente y se envidencia en el Cuadro 2. el modelo Random Forest 2 se posicionó como el mejor entre las diez entregas más destacadas del equipo, logrando un MAE test de 279.650.891,61, significativamente inferior al de los demás modelos evaluados.

Esta ventaja en rendimiento puede atribuirse a dos factores clave: una cuidadosa estrategia de validación cruzada espacial y una sintonización precisa de los hiperparámetros del modelo. A diferencia de otros modelos que pudieron estar sobreajustados a ciertas particiones del conjunto de datos, el enfoque de Random Forest 2 garantizó una mayor capacidad de generalización en zonas geográficas no observadas, lo cual es esencial en contextos urbanos con alta heterogeneidad espacial como Chapinero.

En contraste, otros modelos como Regresión Lineal 1 y Red Neuronal 4, aunque lograron métricas competitivas (MAE de 297 y 315 millones respectivamente), mostraron menor robustez fuera de muestra, posiblemente por depender de supuestos lineales estrictos o requerir mayor ajuste de hiperparámetros y arquitectura. Asimismo, algoritmos como CART, Boosting o Super Learner se vieron limitados por un diseño subóptimo en la selección de variables o especificación de hiperparámetros. En el caso de Random Forest 2, se aplicó una receta de preprocesamiento con normalización, creación de variables ficticias y eliminación de predictores redundantes, todo dentro de un flujo de trabajo reproducible y afinado con un grid de parámetros optimizado para minimizar el MAE. Esto se tradujo en un equilibrio efectivo entre sesgo y varianza, explicando su superioridad frente a modelos más complejos pero menos especializados en la estructura espacial de los datos. En ese sentido, se presenta a continuación algunos de los resultados graficamente:

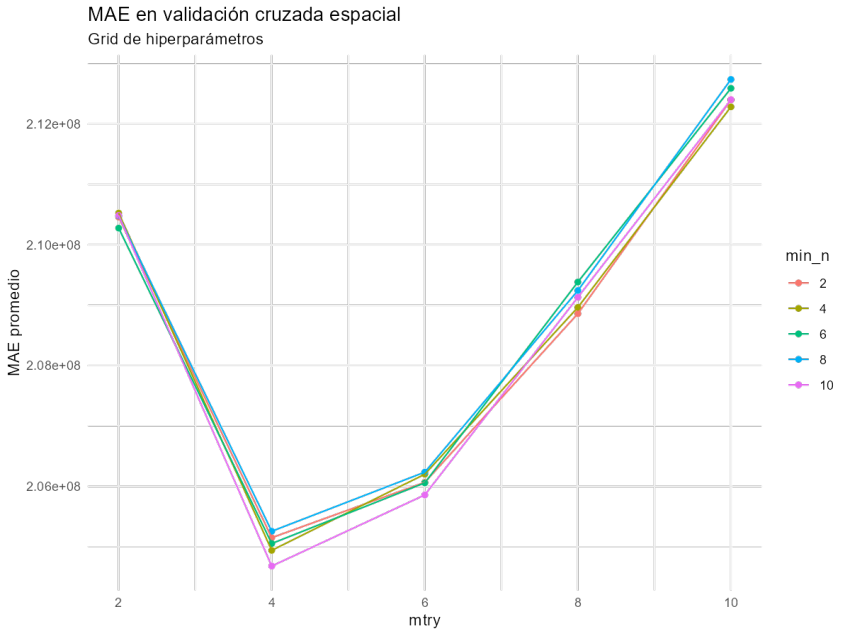

Figura 4: MAE - Validación cruzada

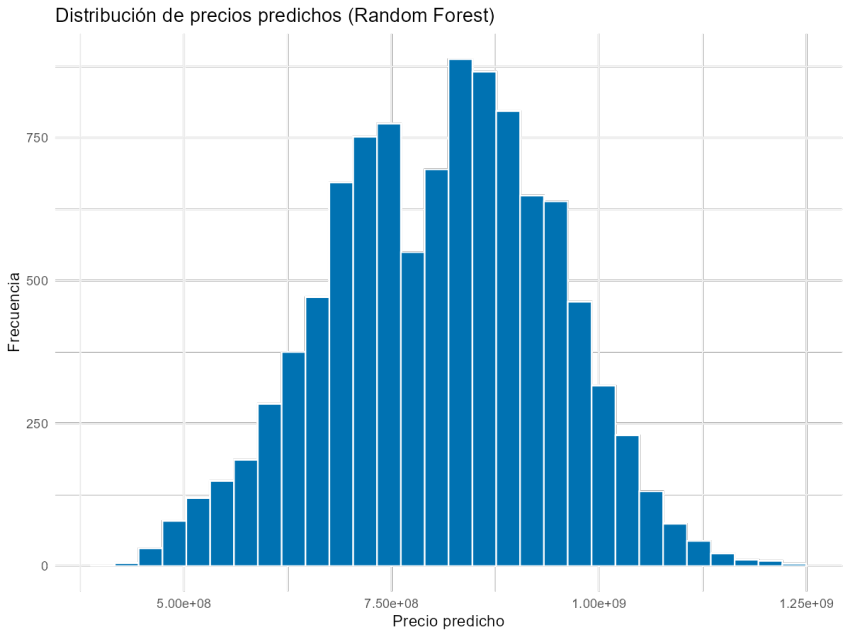

Figura 5: Predicciones del precio - Histograma

4. Conclusiones

Este estudio demuestra que la combinación de variables estructurales, espaciales y derivadas del texto permite capturar con precisión las dinámicas del mercado inmobiliario en Chapinero, Bogotá. A través de un análisis descriptivo riguroso y un modelo Random Forest optimizado con validación cruzada espacial, se logró predecir el precio de las propiedades. La importancia de variables como la cercanía a avenidas o centros comerciales, junto con la presencia de amenidades como parqueadero o ascensor, evidencia que el contexto urbano inmediato tiene un peso significativo en la valoración de los inmuebles, alineado con enfoques hedonísticos del precio.

Sin embargo, la exclusión de variables intrínsecas (área, habitaciones o baño), si bien es útil para evaluar el poder predictivo del entorno, también reveló limitaciones frente a modelos que incorporan esta información. La comparación con resultados de datasets más completos mostró que incluir características físicas mejora considerablemente el desempeño predictivo confirmando la literatura. En consecuencia, este trabajo reafirma la utilidad de los modelos de aprendizaje automático para estimar precios inmobiliarios, pero también destaca la necesidad de integrar múltiples dimensiones de información para mejorar la robustez y aplicabilidad de los modelos en contextos reales.

5. Referencias

1. Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. The Journal of Political Economy, 34-55.

2. Glaeser, E., Gyourko, J., Morales, E., & Nathanson, C. (2014). Housing dynamics: An urban approach. Journal of Urban Economics, 45-56.

3. Brueckner, J. (2011). Lectures on Urban Economics. Massachusetts: The MIT Press.

4. Morales, M., Laverde, M., & Castaño, J. (Junio de 2013). Índice de Precios de la Vivienda

Nueva para Bogotá: Metodología de Precios Hedónicos. Repote de Estabilidad Financiera(78).

5. Revollo, A. (2009). Calidad de vivienda a partir de la metodologia de precios hedonicos para la cuidad de Bogotá - Colombia. Revista digital Universitaria, 10(7).

6. Peñaranda Mogollón, A. (2013). Efecto del costo de acceso a centros de empleo sobre el precio de la vivienda urbana - un modelo hedónico para el caso de Bogotá. Disponible en: `http://hdl.handle.net/1992/12311`

7. Bailey, M., Richard, M., & Hugh, N. (1963). A Regression Method For Real Estate Price Index Construction. Journal of the American Statistical Association, 58(304), 933-942.

8. Linneman, P. (1982). Hedonic Prices and Residential Location. The economics of urban amenities, 69-88.

9. Grether, D., & Mieszkowski, P. (1974). Determinants of Real Estate Values. Journal of Urban Economics, 127-146.

10. Mojica, A., & Viáfara, J. (2019). Determinantes de los precios de la propiedad en Bogota D. C. Tesis de grado Maestria en Finanzas Corporativas, Colegio de Estudios Superiores de Administración – CESA, Bogotá.

6. Anexos

6.1 Elastic Net

La función objetivo del Elastic Net se define como:

$$L(\boldsymbol{\beta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda_1||\boldsymbol{\beta}||_1 + \lambda_2||\boldsymbol{\beta}||^2 * * \tag{1}$$

Donde:

- λ_1 controla la penalización L1 (Lasso) que promueve la selección de variables
- λ_2 controla la penalización L2 (Ridge) que maneja la multicolinealidad
- $\alpha = \frac{\lambda_1}{\lambda_1+\lambda_2}$ es el parámetro de mezcla que balancea ambas penalizaciones

El modelo se parametrizó conforme a la literatura de precios hedónicos, utilizando modelos lineales multivariados para predecir el precio de la propiedad. La definición del método de regularización se centró en la especificación de los parámetros α (mixture) y λ (penalty) mediante una cuadrícula de búsqueda definida en escala logarítmica base 10. A continuación la configuración de los hiperparámetros:

- **Penalty (λ)**: Rango de 10^{-4} a 10^{-1} (0.0001 a 0.1) con 15 niveles
- **Mixture (α)**: Rango de 0 a 1 con 8 niveles
- **Total de combinaciones**: 120 configuraciones evaluadas

Esta configuración permitió una exploración exhaustiva del espacio de hiperparámetros, optimizando tanto la capacidad predictiva como la parsimonia del modelo.

El modelo incluyó variables estructurales (superficie total y cubierta, número de habitaciones, baños), variables temporales (año), variables de localización (tipo de propiedad), variables de distancia espacial (a CAI, centros comerciales, paradas de bus, avenidas principales) y variables de amenidades extraídas mediante procesamiento de texto (parqueadero, terraza, gimnasio, lavandería, ascensor).

Validación Cruzada Espacial: Se implementó mediante bloques espaciales (`spatial_block_cv`) con $v = 5$ folds, lo que permitió mantener la independencia espacial entre los bloques de muestra y evitar estimaciones demasiado optimistas del rendimiento del modelo. Esta técnica es crucial en datos geoespaciales para prevenir el sesgo de autocorrelación espacial.

Validación Cruzada Convencional: Se aplicó k-fold cross-validation ($k = 5$) con estratificación por precio, dividiendo los datos aleatoriamente y evaluando el rendimiento del modelo de manera estándar, sin considerar la estructura espacial de los datos. Manteniendo todos los parámetros iguales y variando únicamente el método de validación cruzada, se observó que:

Modelo con Validación Convencional:

- MAE: 193,213,499 pesos
- Mejores hiperparámetros: penalty $= 1 \times 10^{-4}$, mixture $= 0.857$
- CV de predicciones: 7.57 %

Modelo con Validación Espacial:

- MAE: 199,745,227 pesos
- Mejores hiperparámetros: penalty $= 1 \times 10^{-4}$, mixture $= 0.714$
- CV de predicciones: 7.47 %

El método convencional arrojó mejores indicadores de MAE mientras que la validación espacial mostró ligeramente menor variabilidad en las predicciones. Para datos inmobiliarios, se recomienda el modelo con validación convencional por su superior capacidad predictiva, especialmente cuando las variables espaciales ya están capturadas mediante las distancias a sitios específicos, reduciendo la importancia de la estructura espacial en la validación cruzada.

6.2 CART

El modelo de regresión CART (Árboles de Clasificación y Regresión) fue entrenado inicialmente de forma tradicional utilizando la función `rpart()`, con la fórmula `price ~.` aplicada sobre la base `train_factors`. Se especificó el argumento `method = anova` para realizar regresión sobre una variable continua. Este enfoque permite generar árboles de decisión simples, interpretables y computacionalmente eficientes. La principal ventaja de este entrenamiento tradicional es su rapidez y facilidad de implementación. No obstante, no contempla la posible dependencia espacial entre observaciones, lo cual puede llevar a una sobreestimación de la capacidad predictiva del modelo en contextos geográficos distintos.

Se visualizó el árbol resultante con `rpart.plot()`, y se evaluó el error absoluto medio (MAE) tanto en la muestra de entrenamiento como en la de prueba. Este primer modelo sirve como línea base para comparar con versiones más robustas que consideren la dimensión espacial.

Con el objetivo de mejorar la capacidad del modelo para generalizar en el espacio geográfico y evitar el sesgo inducido por la autocorrelación espacial, se implementó un segundo entrenamiento utilizando validación cruzada espacial. Para esto se trabajó dentro del `framework tidymodels`, aprovechando el paquete `spatialsample` para dividir el conjunto de entrenamiento en bloques geográficos mutuamente excluyentes.

En esta configuración, se entrenó un modelo CART con tres hiperparámetros principales a sintonizar: `cost_complexity`, que controla la penalización por complejidad del árbol (valores más altos favorecen árboles más pequeños, reduciendo el sobreajuste); `tree_depth`, que fija la profundidad máxima del árbol (limitarla evita sobreajustes y mejora la interpretabilidad); y `min_n`, que determina el número mínimo de observaciones requeridas para dividir un nodo (valores más altos promueven estabilidad y generalización).

La sintonización se llevó a cabo mediante una búsqueda en una grilla regular de 4 niveles para cada hiperparámetro, evaluando el desempeño en 2 bloques espaciales con la métrica MAE. Finalmente, se seleccionaron los mejores hiperparámetros con `select_best()` y se ajustó el modelo final usando `fit()` sobre toda la base de entrenamiento. Este modelo final fue utilizado para predecir precios en el conjunto de prueba.

En términos metodológicos, el entrenamiento tradicional permite construir rápidamente modelos y obtener interpretaciones inmediatas sobre las variables predictoras. Sin embargo, en problemas espaciales, este enfoque puede resultar optimista al evaluar el desempeño sobre datos cercanos geográficamente a los de entrenamiento.

En cambio, la validación cruzada espacial ofrece una evaluación más realista del error de predicción fuera de muestra, obligando al modelo a aprender patrones que puedan generalizar a nuevas zonas geográficas. Además, la búsqueda sistemática de hiperparámetros permite controlar el balance entre flexibilidad y sobreajuste, produciendo árboles más robustos y adaptados a la estructura espacial de los datos.

6.3 Boosting

En esta sección se implementa un modelo de boosting usando el algoritmo XGBoost, con el objetivo de predecir precios de apartamentos a partir de características estructurales y de ubicación. Para comenzar, se entrena un modelo con validación cruzada tradicional `(kfold)` utilizando el paquete `caret`. Se define una grilla de hiperparámetros (como el número de rondas, la profundidad del árbol y la tasa de aprendizaje), y se selecciona el MAE (Error Absoluto Medio) como métrica de evaluación. Este modelo se ajusta sobre la muestra de entrenamiento, y posteriormente se realizan predicciones sobre la muestra de prueba, asegurando la coherencia en los niveles de las variables categóricas entre ambas bases.

Posteriormente, se entrena un segundo modelo XGBoost, pero esta vez utilizando validación cruzada espacial, con el objetivo de controlar por la dependencia geográfica de los datos y evitar que la proximidad espacial entre observaciones de entrenamiento y prueba genere un sesgo optimista en la evaluación. Para ello, se construyen bloques espaciales usando la función `spatial_block_cv()` del paquete `spatialsample`, que divide la muestra en cinco subconjuntos basados en su ubicación geográfica.

Esto simula un escenario más exigente donde el modelo debe predecir en áreas no incluidas en el entrenamiento. Los identificadores de las observaciones de entrenamiento de cada bloque se convierten en índices y se incorporan en la validación mediante el argumento validRows, permitiendo a caret ajustar los hiperparámetros respetando la estructura espacial. Luego, se generan los índices de entrenamiento y prueba personalizados, que son incorporados en `trainControl()` para que caret entrene el modelo respetando estas divisiones.

Ambos modelos comparten la misma grilla de hiperparámetros y la misma fórmula predictiva, lo que permite una comparación directa entre sus desempeños. La diferencia clave radica en cómo se construyen las particiones de validación: en el modelo tradicional, las observaciones se reparten aleatoriamente, mientras que en el modelo espacial se garantiza que los subconjuntos estén separados geográficamente. Esta diferencia es especialmente relevante en problemas espaciales como el presente, donde las características del entorno pueden inducir una fuerte autocorrelación espacial. Finalmente, se realizan las predicciones con el modelo entrenado con validación espacial sobre la misma muestra de prueba utilizada antes, asegurando nuevamente la coherencia en los niveles de las variables categóricas.

6.4 Redes Neuronales

La estimación de los modelos de redes neuronales se realizó utilizando la librería `keras` en R, en combinación con el ecosistema de `tidymodels` para el preprocesamiento y validación. Los emplean una arquitectura `feedforward` con una capa densa inicial de 64 neuronas con activación ReLU, seguida de una capa de dropout con una tasa de 0.3 (para evitar sobreajuste), una segunda capa densa con 32 unidades y una capa de salida de una neurona, adecuada para problemas de regresión. El optimizador utilizado fue `RMSprop`, con una función de pérdida de error cuadrático medio (MSE), y como métrica de evaluación se usó el error absoluto medio (MAE). El conjunto de datos fue transformado mediante una receta que incluyó codificación de variables categóricas, normalización de predictoras numéricas y manejo de categorías poco frecuentes.

Los modelos con validación cruzada tradicional emplearon cinco particiones aleatorias del conjunto de entrenamiento. En cada iteración, el modelo se ajustó durante 100 `epoch` con `batch size` de 32, utilizando un 20 % del conjunto como validación interna. Los resultados de los cinco folds se promediaron para estimar la capacidad predictiva fuera de muestra del modelo. Este enfoque es útil cuando se asume que las observaciones están distribuidas de forma aleatoria e independiente, aunque puede subestimar el error si existen dependencias espaciales en los datos, como suele ocurrir con precios de inmuebles.

Para abordar esa posible dependencia espacial, se aplicó una validación cruzada espacial agrupando los datos según zonas geográficas obtenidas a partir de un algoritmo de `k-means` con cinco clústeres. Cada fold dejó fuera una zona geográfica distinta, replicando mejor el escenario de aplicar el modelo en una región no observada durante el entrenamiento. Esto proporciona una evaluación más realista de la capacidad de generalización espacial del modelo. Aunque ambos modelos usaron la misma arquitectura y parámetros, el segundo ofreció una validación más conservadora, reflejando la importancia del contexto geográfico en la predicción de precios inmobiliarios.

6.5 Regresión lineal

La estimación de los modelos de regresión lineal desarrollados en este proyecto se fundamentó en un enfoque clásico de regresión ordinaria por mínimos cuadrados (OLS), con el objetivo de explicar y predecir el precio de venta de viviendas a partir de una combinación de variables estructurales, contextuales y de amenidades. Para ello, se definieron diferentes especificaciones del modelo (incluyendo logarítmicas) en función de la selección de variables explicativas, buscando capturar la mayor cantidad posible de variación del precio sin incurrir en sobreajuste o multicolinealidad.

En términos metodológicos, las variables predictoras se seleccionaron con base en su relevancia teórica dentro del enfoque hedónico del valor de los bienes inmuebles, así como por su disponibilidad y calidad en los datos recopilados. Se incluyeron características como el año de construcción,

número de habitaciones y baños, área construida, tipo de propiedad (categórica), y variables espaciales como la distancia a centros comerciales, estaciones de transporte público y avenidas principales. Además, se incorporaron variables binarias que capturan la presencia de amenidades como parqueadero, gimnasio y ascensor, extraídas mediante técnicas de procesamiento de texto de las descripciones de los inmuebles.

Cada modelo fue ajustado utilizando la función `lm()`, y se evaluaron distintas combinaciones de predictores. Uno de los modelos se ajustó con un conjunto amplio de variables, mientras que otros modelos exploraron especificaciones más parsimoniosas. Posteriormente, se utilizó la función `vip()` para visualizar la importancia relativa de cada predictor en la explicación del precio, facilitando así la interpretación y comparación entre modelos. Esta aproximación permitió no solo obtener estimaciones robustas, sino también comprender la contribución individual de cada variable al desempeño predictivo general. A pesar que fue el segundo mejor modelo en terminos de predicción de los precios de viviendas en Bogotá, el $R2$ era muy bajo para considerarse un modelo aceptable.

6.7 Super Learners

En esta sección se implementa el algoritmo Super Learner, una técnica de ensamble que combina de manera óptima múltiples modelos base con el objetivo de minimizar el error de predicción. Para este ejercicio, se construyó una librería de aprendizaje que incluye un modelo de regresión lineal y un modelo de boosting con xgboost, combinados mediante regresión no negativa de mínimos cuadrados (method.NNLS).

El primer entrenamiento del Super Learner se realiza bajo un esquema de validación cruzada tradicional con 5 particiones (5fold CV). En este caso, los datos se dividen aleatoriamente en cinco subconjuntos, rotando el conjunto de validación en cada iteración. Este proceso permite estimar el desempeño del modelo en datos no observados sin necesidad de un conjunto de validación externo. La regresión no negativa determina el peso óptimo que debe asignarse a cada modelo base según su desempeño promedio en las particiones, maximizando la capacidad predictiva del ensamble.

Una vez entrenado el modelo Super Learner tradicional, se generan las predicciones para el conjunto de prueba, usando únicamente la combinación final de modelos `(onlySL = TRUE)`. Estas predicciones se agregan al conjunto de prueba y se ajustan al formato requerido para la entrega de resultados. Esta especificación sirve como línea base para comparar el impacto de considerar o no la dimensión espacial en la validación del modelo, dado que la validación tradicional asume que las observaciones son independientes y no presentan autocorrelación espacial.

Para capturar de forma más realista la estructura de los datos georreferenciados, se entrena una segunda versión del Super Learner utilizando validación cruzada espacial. Este enfoque emplea el método de bloques espaciales `(spatial_block_cv)`, que divide el espacio geográfico en cinco zonas distintas. A diferencia de la validación tradicional, aquí cada fold representa una región diferente, lo cual permite simular mejor la tarea de predecir en zonas nunca vistas durante el entrenamiento. Una vez definidos los folds espaciales, se extraen los identificadores correspondientes a los datos de entrenamiento de cada bloque y se transforman en índices que el paquete `SuperLearner` puede utilizar mediante el argumento `validRows`.

Este segundo modelo se entrena exactamente con la misma librería de modelos base y el mismo método de combinación, pero empleando los folds espaciales. Finalmente, se generan predicciones para el conjunto de prueba, tal como en el modelo anterior. Esta configuración permite evaluar si el modelo mantiene un buen desempeño al extrapolar en el espacio, es decir, al predecir precios en zonas diferentes a aquellas usadas para entrenar.