Problem Set 3: Making Money with ML? "It's all about location location location!!!

Andrés Rueda López - a.ruedal@uniandes.edu.co

November 28, 2024

Todos los documentos y archivos utilizados para la solución del problema se encuentran en el siguiente repositorio: Repositorio en GitHub. Adicionalmente, se ha creado la sección de Apéndice 5 para mayor detalle sin incurrir en exceso de paginas.

1 Introducción

Las viviendas urbanas pueden entenderse como bienes que incorporan múltiples atributos que influyen en su precio dentro del mercado inmobiliario. El modelo hedónico permite analizar cómo los compradores valoran las viviendas en función de características propias de la propiedad (como el número de habitaciones, baños y el espacio) y de atributos específicos de la ubicación (como la calidad del aire, la seguridad y la accesibilidad) [1].

Formalmente, el precio de una vivienda puede modelarse como una función hedónica de sus atributos, expresada de la siguiente forma:

$$P = f(a_1, a_2, \ldots, a_m) \tag{1}$$

Sin embargo, identificar y medir adecuadamente estos atributos es un desafío complejo. Además, la forma funcional de f no está claramente definida en el modelo, lo cual dificulta la comprensión de la relación exacta entre cada atributo $(a_1, ..., a_m)$ y el precio P de la propiedad.

A pesar de esta complejidad, diversos estudios han logrado identificar componentes clave que influyen en el valor asignado a las viviendas. Por ejemplo, los modelos monocéntricos muestran que el precio de una vivienda tiende a disminuir conforme aumenta su distancia al centro de la ciudad [2]. Asimismo, se ha investigado cómo el índice de criminalidad en áreas cercanas puede impactar negativamente en el valor de las propiedades [3]. De manera más específica, este tipo de estudios han mostrado como la distancia a colegios [4], sistemas de transporte público [5] e incluso la cercanía a instalaciones industriales [6] influyen de manera diversa y signifcativa en el valor de las propiedades.

Este estudio, más allá de analizar los efectos causales de diversos atributos en el precio de las viviendas, busca explorar dichos atributos y emplearlos para predecir el valor de ciertas propiedades en la localidad de Chapinero, en Bogotá. Para ello, se utilizan datos abiertos sobre precios de vivienda en la ciudad y se aplican métodos estadísticos y de machine learning que buscan estimar el precio de las propiedades de la forma más precisa posible. Particularmente, gran cantidad de la información es obtenida a partir del texto con el cual los propietarios describen sus viviendas, lo cual involucrando un reto adicional.

En primer lugar, se detalla el proceso de adquisición y manejo de los datos, cuyo objetivo fue la construcción de atributos y predictores clave para modelar el precio de las viviendas. A continuación, se analiza en profundidad el desempeño del mejor modelo predictivo identificado, destacando sus resultados y comparándolos con otros modelos que mostraron un rendimiento inferior. En particular, se resalta el uso de un modelo basado en boosting, el cual logró un error absoluto medio de 250.158 millones de pesos. Por último, se presentan conclusiones concisas basadas en los resultados obtenidos, identificando posibles áreas de mejora para futuros trabajos.

2 Datos

En esta sección se presenta una descripción detallada de los datos proporcionados y las transformaciones aplicadas para la creación y evaluación de los modelos de predicción propuestos. Por un lado, la competencia ha entregado un conjunto de datos con información de 38,644 hogares de diversas localidades de Bogotá, que se utilizarán en el proceso de entrenamiento. Por otro lado, se dispone de datos específicos sobre 10,286 propiedades en la localidad de Chapinero, donde se espera realizar predicciones del valor de venta basadas en sus características.

Primero, se analizó la distribución de los precios de las viviendas en el conjunto de entrenamiento, ya que esta es la variable objetivo para la predicción. La Figura 1 muestra la distribución de los precios de las propiedades en estudio, acompañada de un mapa que distingue las zonas de entrenamiento de la zona de evaluación.

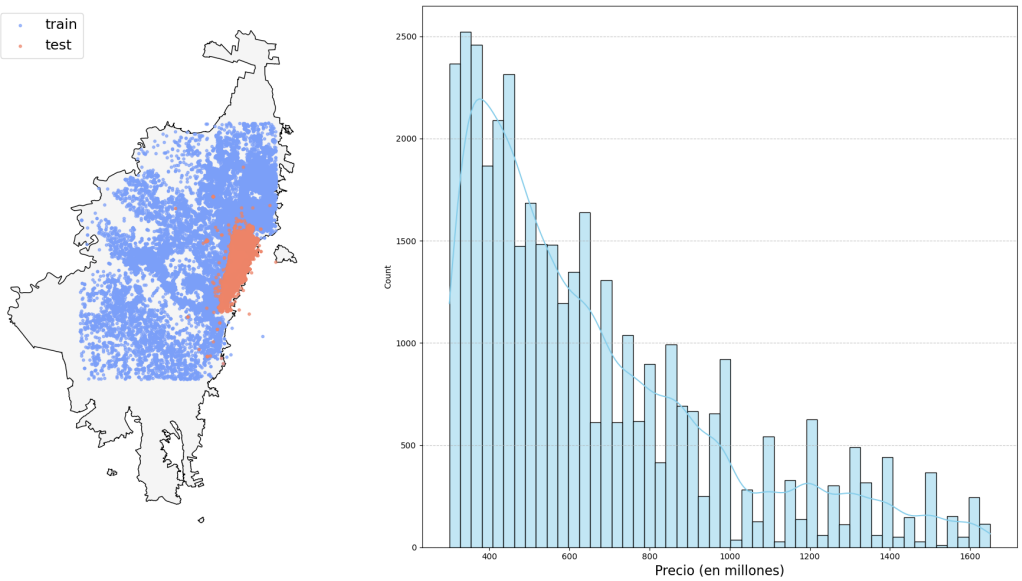

Figure 1: Distribución de los precios (en millones de pesos) de las propiedades en el conjunto de entrenamiento y mapa de Bogotá con la ubicación de los puntos de entrenamiento y evaluación.

De manera adicional, se realizó un análisis descriptivo básico del conjunto de entrenamiento y evaluación, con el fin de determinar potenciales predictores a utilizar en los modelos predictivos. La Tabla 2 contiene la información descriptiva de los datos de entrenamiento

	Precio (millones)	Mes	Año	Superficie Total	Superficie Cubierta	Habitaciones	Alcobas	Baños	Latitud	Longitud
Count	38,644	38,644	38,644	7,854	8,565	20,384	38,644	28,573	38,644	38,644
Mean	654.53	5.67	2020.29	153.95	131.93	3.01	3.14	2.88	4.691	-74.063
Std	311.42	3.29	0.76	274.37	76.62	1.37	1.53	1.09	0.037	0.032
Min	300.00	1	2019	16	2	1	0	1	4.577	-74.170
25%	415.00	3	2020	84	81	2	2	2	4.679	-74.076
50%	559.99	5	2020	119	108	3	3	3	4.700	-74.050
75%	810.00	8	2021	185	160	3	3	3	4.718	-74.039
Max	1,650.00	12	2021	17,137	1,336	11	11	13	4.765	-74.026

Table 1: Resumen estadístico de las variables principales

La Tabla 2 presenta información clave para la selección de los predictores. Por ejemplo, se observa que la distribución del número de habitaciones es bastante amplia, abarcando valores entre 1 y 11, lo que sugiere la presencia de estructuras residenciales muy diversas en el conjunto de datos. Asimismo, el valor medio del precio y la distribución de los cuartiles, respaldados por la representación gráfica de la Figura 1, evidencian una notable falta de uniformidad en los datos. En términos generales, la Tabla refleja un conjunto de datos altamente heterogéneo, con un comportamiento variado en cada una de las variables analizadas.

Dado el alto número de valores faltantes en atributos clave de las propiedades, se recurrió a la información contenida en las descripciones textuales de las mismas. Este enfoque permitió recuperar datos faltantes para cada propiedad, posibilitando así su inclusión en el modelo.

Para extraer y construir información a partir del texto presente en las descripciones de las viviendas, se llevó a cabo un proceso de limpieza y normalización utilizando la librería `re` de Python. Posteriormente, mediante el uso de expresiones regulares diseñadas específicamente para cada atributo, fue posible identificar y extraer el valor del atributo de interés en varios casos. Este enfoque fue utilizado para encontrar los valores faltantes y completar la información con respecto al **número de habitaciones**, **número de baños** y **superficie cubierta** (en m^2).

Por otro lado, se procedió a crear y evaluar nuevos predictores que pudiese ofrecer mayor información sobre las propiedades, para lo cual fue necesario utilizar la ubicación espacial de estas. En primer lugar, utilizando la información del grado de estratificación por cuadrante en la ciudad proporcionado por el DANE [7], se procedió a asignar el estrato correspondiente a cada vivienda. A continuación se muestra la distribución de estratos a lo largo de la ciudad de Bogotá junto con la distribución del precio de las propiedades a partir de su estrato socioeconómico.

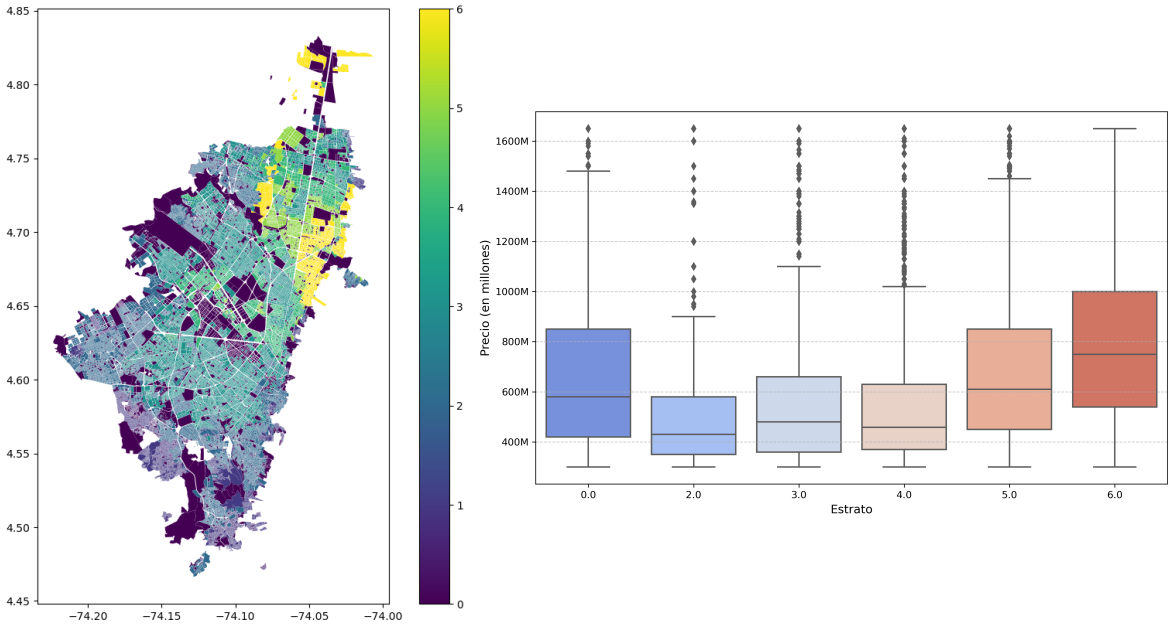

Figure 2: Distribución de los estratos socioeconómicos y precios (millones COP) de vivienda por estrato en la ciudad de Bogotá.

La Figura 2 muestra la relación entre el precio de las propiedades y el estrato socioeconómico, evidenciando un incremento en los precios a medida que el estrato aumenta. Es importante destacar que el estrato cero presenta un comportamiento particular, ya que puede corresponder a zonas históricas preservadas o áreas con algún tipo de jurisdicción especial, o bien asociarse a bodegas o espacios de gran tamaño. Por otro lado, el mapa de la Figura 2 muestra que la zona de estudio mayormente corresponde a los estratos 4, 5 y 6 por lo que se decidió descartar aquellas zonas cuyo estrato promedio fuese inferior al estrato 3.5, dejando así únicamente con las localidades: Barrios Unidos, Teusaquillo, Santa Fe, Usquén, Suba y Fontibon.

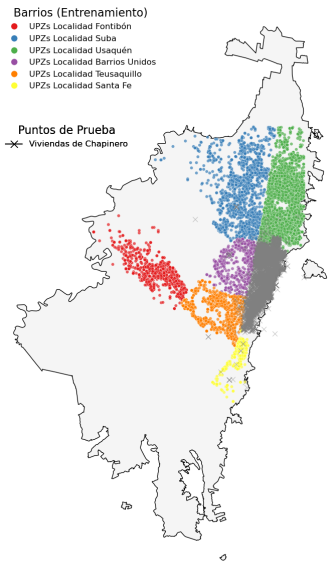

Figure 3: Zonas de Bogotá filtradas y seleccionadas a partir del estrato socioeconómico.

La Figura 3 muestra una distribución de las viviendas que se utilizarán para entrenamiento. Es importante notar que mayoritariamente son zonas conjuntas lo cual busca tener en cuenta las posibles relaciones espaciales compartidas que estas zonas comparten con el fin de poder estimar otros predictores. Teniendo en cuenta la literatura revisada, se procedió a estimar para cada hogar de manera adicional los siguientes atributos utilizando *OpenStreatsMap* [8]:

- Distancia al centro de la ciudad: Teniendo en cuenta que el centro de la ciudad es un gran aglomerado de universidades y zonas de trabajo, se calculo la distancia al centro internacional de la ciudad ubicado en la localidad de Santa Fe.
- Distancia a colegios: Se construyo un listado de los colegios más importantes de la zona nororiental de la ciudad y se calculo la distancia al colegió más cercano.
- Distancia a centros empresariales: Se calcula la distancia más cercana a alguno de los centros empresariales más importantes de la ciudad como la zona financiera de la calle 72, parque de la 93 y centro empresarial Santa Barbara.
- Distancia a zonas gastronómicas: Se calcula la distancia más cercana a alguna de las zonas de restaurantes y bares más importantes y grandes de la zona centro-oriente y norte-oriente.
- Distancia a transporte público: Se calcula la distancia más cercana a alguna estación de transmilenio (por ser el sistema de transporte más importante de la ciudad) de la zona centro-oriente y norte-oriente.
- Distancia a parques: Se calcula la distancia más cercana a algún parque o espacio abierto de la zona centro-oriente y norte-oriente.

Finalmente, con el fin de enriquecer la base de datos con posibles comportamientos y relaciones complejas entre los predictores espaciales y propios de las viviendas, se realizaron múltiples transformaciones y combinaciones no lineales entre las variables. A continuación se muestra un ejemplo:

$$X_{(\text{superficie y habitaciones})} = X_{\text{superficie}} * X_{\text{habitaciones}} \tag{2}$$

En caso de buscar mayor detalle sobre la matriz X de predictores que serán utilizados en el proceso de entrenamiento y evaluación del modelo, se invita al lector a visitar el apéndice 5.

3 Modelos y Resultados

El objetivo principal de este estudio es predecir el precio de propiedades en Bogotá a partir de ciertos atributos específicos. Para ello, se entrenaron y evaluaron diferentes modelos de Machine Learning con el propósito de comparar sus resultados y determinar cuál ofrece el mejor desempeño.

En esta tarea, se configuraron cinco modelos distintos: Regresión Lineal, Elastic Net, Random Forest, XGBoost y Redes Neuronales Profundas. Todos los modelos fueron entrenados utilizando el mismo conjunto de variables predictoras, asegurando así una comparación justa y consistente de su rendimiento. Los predictores empleados fueron previamente descritos en la sección anterior y se presentan de forma técnica en la sección 5 para mayor claridad sobre su implementación.

Es importante mencionar que previo al entrenamiento, los datos de entrenamiento y evaluación fueron sometidos a un proceso de normalización. Lo anterior, buscando evitar impresiciones que pudiesen causar las diferentes escalas en las que se presetan los predictores.

$$X^{\text{s}}_{i,j} = \frac{X_{i,j} - \bar{X}_i}{\sigma_i} \tag{3}$$

Por otro lado, teniendo en cuenta la gran escala numérica en la que se encuentra el precio de las propiedades, el precio de las propiedades fue escalado de manera logarítmica.

3.1 Mejor Modelo: Xg-Boost

3.1.1 Funcionamiento

El modelo de *Xg-Boost* pertenece a la familia de modelos basados en arboles, siendo este una versión optimizada del algoritmo de *Gradient Boosting*. En este caso, se usa como base el principio de *boosting*, el cual utiliza una combinación de múltiples estimadores débiles (arboles en este caso) para la construcción de un estimador fuerte. Lo anterior, permite que cada árbol busque corregir los errores del árbol anterior, logrando así mejorar la función objetivo de manera iterativa.

Teniendo en cuenta que los modelos basados en arboles tienden a sobreajustar los datos de entrenamiento, Xg-Boost incorpora dentro de su función objetivo un termino de regularización, el cual permite reducir la complejidad y por ende reducir la varianza del modelo. De manera general, sea $\mathcal{L}$ la función objetivo del problema, este algoritmo soluciona el siguiente problema:

$$\min \mathcal{L} = \min \left[\sum_{i=1}^{N} L(y_i, \hat{y}_i) + \sum_{k=1}^{m} \Omega(f_k) \right] \tag{4}$$

Donde:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||\phi|| \tag{5}$$

Este problema carece de solución por métodos tradicionales de optimización, por lo cual se invocan modelos basados en boosting por gradiente. Sin embargo, estos métodos son altamente paralelizables, permitiendo así que este modelo sea muy efectivo en términos computacionales.

3.1.2 Implementación

La implementación del algoritmo se llevó a cabo utilizando el paquete `XGBRegressor` de la librería `xgboost` en Python (versión 3.11.5). Este paquete se integra con la librería `sklearn`, que proporciona las métricas y funciones necesarias para el entrenamiento y evaluación de los modelos.

Primero, los datos de la base `train` se dividieron en dos conjuntos: un conjunto de entrenamiento (80% de los datos) y un conjunto de validación (20%). Esta partición permitió realizar un proceso de validación cruzada más riguroso, evaluando el rendimiento del modelo con un subconjunto de datos no utilizado durante el entrenamiento.

Posteriormente se implemento como métrica de evaluación del modelo al *error cuadrático absoluto medio* el cual corresponde a la métrica con la cual son evaluados los modelos en la competencia de Kaggle. Recuerde que el error cuadrático absoluto medio se define como:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{6}$$

Una vez implementada la métrica, se procedió a construir una grilla de hiper-parámetros con el fin de explorar la mejor combinación de estos en el modelo. A continuación se mencionan los parámetros explorados y los rangos en los que fueron evaluados en la búsqueda.

- **Learning rate**: Este parámetro controla la velocidad con el el modelo se ajusta a los datos de entrenamiento. Su función es la de regular el impacto de cada arbol en la construcción del árbol final. **Espacio explorado:** [0.005, 0.01, 0.05, 0.1, 0.2, 0.3]
- **Número de estimadores**: Es el número total de árboles que el modelo entrenará. Más árboles suelen mejorar la capacidad del modelo para capturar patrones complejos, pero un número excesivo puede llevar a un sobreajuste y aumentar el tiempo de entrenamiento. **Espacio explorado:** [200, 500, 800, 1000].
- **Profundidad máxima**: Define la profundidad máxima de cada árbol en el modelo. Una mayor profundidad permite capturar interacciones más complejas entre las variables, pero también incrementa el riesgo de sobreajuste. Una profundidad menor puede resultar en un modelo más generalizable. **Espacio explorado:** [3, 5, 7, 10].
- **subsample**: Representa la fracción de las muestras de entrenamiento que se utilizarán para entrenar cada árbol. Un valor menor que 1 introduce aleatoriedad en el modelo, lo que puede reducir el sobreajuste y mejorar la generalización. **Espacio explorado:** [0.6, 0.8, 1.0].
- **colsample bytree**: Indica la proporción de características (columnas) seleccionadas de forma aleatoria para construir cada árbol. Ayuda a evitar el sobreajuste al reducir la dependencia del modelo en ciertas características específicas. **Espacio explorado:** [0.6, 0.8, 1.0]
- **gamma**: Es un parámetro de regularización que establece un umbral mínimo para dividir un nodo en un árbol. Si el beneficio de dividir el nodo no supera el valor de gamma, no se realiza la división. Un gamma mayor fomenta árboles más simples y reduce el sobreajuste. **Espacio explorado:** [0, 1, 5].
- **regularization** α: Representa el término de regularización L1. Este parámetro penaliza los pesos altos de las características, promoviendo la eliminación de aquellas menos importantes y produciendo modelos más simples y esparsos. **Espacio explorado:** [0, 0.1, 1].
- **regularization** λ: Es el término de regularización L2, que penaliza los pesos altos de las características, pero de manera más suave que α. Esto ayuda a reducir la sensibilidad del modelo a pequeñas fluctuaciones en los datos y mejora la generalización. **Espacio explorado:** [1,5,10].

La selección de estos hiper-parámetros fue realizada mediante una búsqueda aleatoria en la grilla previamente mencionada. Para ello se definio un total de 50 combinaciones aleatorias de parámetros que permite tener una gran exploración en el espacio de solución sin necesidad de incurrir en el ejercicio exhaustivo. De esta forma, los mejores hiper parámetros encontrados fueron los siguientes:

Hiperparámetro	**Valor Óptimo**
Learning rate	0.1
Número de estimadores	1000
Profundidad máxima	10
subsample	1.0
colsample bytree	0.4
gamma	0
regularization α	0
regularization λ	1.0

Table 2: Valores óptimos de hiperparámetros explorados para XG-Boost mediante búsqueda aleatoria.

Por otro lado, se realizo un proceso de validación cruzada utilizando el 20% de los datos de entrenamiento. Este proceso estuvo basado en el principio de *Early Stop*, donde se realiza el entrenamiento del modelo evaluando constantemente en el conjunto de validación y encontrando el punto en el cual el

modelo no tiene un mejor desempeño y deteniendo el proceso de forma que no se incurra en sobreajuste. Aunque estos problemas suelen trabajarse con procesos de validación cruzada espacial, este proceso no mostró una mejora considerable en los datos y si tenia un costo computacional considerable. Esto puede deberse al filtrado inicial de los datos en el que la exclusión de estratos bajos redujo el conjunto de entrenamiento y validación a una serie de barrios aledaños.

3.2 Otros modelos

3.2.1 Regresión Lineal:

Se estimó un modelo de regresión lineal utilizando `sklearn` en Python. Dado que esta librería no admite valores faltantes, los datos ausentes fueron reemplazados por la mediana correspondiente a cada estimador. Este procedimiento introdujo un sesgo significativo en el modelo, lo que impactó negativamente los resultados. Además, cabe destacar que un modelo de regresión lineal no puede capturar de manera autónoma relaciones complejas y no lineales entre las variables, lo que contribuyó a errores más grandes tanto en la etapa de entrenamiento como en la de evaluación.

Si bien la regresión lineal es una herramienta poderosa en muchas aplicaciones, en este caso, no se conocía de antemano la naturaleza de las relaciones entre los atributos de las propiedades. Esto dificultó la incorporación manual de no linealidades, lo que resultó en un desempeño inferior frente a los otros cuatro modelos probados, posicionando a la regresión lineal como la de menor rendimiento en el conjunto de evaluaciones realizadas.

3.2.2 Elastic Net:

Se entreno un modelo de Elastic Net utilizando `sklearn` en Python. Dado que esta librería no admite valores faltantes, los datos ausentes fueron reemplazados por la mediana correspondiente a cada estimador. Este procedimiento introdujo un sesgo significativo en el modelo, lo que impactó negativamente los resultados. Aunque este modelo busca utilizar estrategias de regularización con el fin de reducir la información redundante y el grado de sobreajuste a los datos de entrenamiento, su desempeño no fue superior al mostrado por las estrategias basadas en arboles. Esto puede deberse a que el número de estimadores no era lo suficientemente grande como para que las estrategias de regularización y selección de variables avanzadas muestren superioridad sobre otras estrategias.

3.2.3 Redes Neuronales Profundas:

Se entreno un modelo basado en Redes Neuronales Profundas utilizando `keras` en Python con diferentes arquitecturas. Aunque las Redes Neuronales Profundas actualmente muestran un gran desempeño en diferentes tareas predictivas, es importante recordar que su potencial se alcanza cuando existe una gran cantidad de datos disponibles. Sin embargo, teniendo en cuenta que fue necesario hacer el tratamiento de datos nulos al igual que en regresión lineal y Elastic Net, se introdujo un gran sesgo dentro del modelo y el número de datos no parece lo suficientemente alto para poder explotar el rendimiento de las redes. Por esta razón, a pesar de haber explorado ampliamente el espacio de los hiper-parámetros, el rendimiento de las redes fue incluso inferior al alcanzado por métodos como Random Forest.

3.2.4 Random Forest:

Se entreno un modelo basado en arboles (en este caso Random Forest) explorando una amplia grilla de hierparámetros que permitiese encontrar la combinación de mayor rendimiento. Este modelo, obtuvo un excelente desempeño en términos relativos con los otros modelos, debido a que las características de las variables se adecúan bastante bien al funcionamiento que tienen los arboles de decisión. Sin embargo, es habitual que este tipo de modelos tiendan a tener problemas de sobre-ajuste y en algunas ocasiones la exploración de los parámetros puede llegar a ser computacionalmente muy costosa.

3.3 Resultados

Los resultados presentados se basan en los submissions realizados en Kaggle, los cuales son calculados usando el 20% de los datos de manera preliminar. La Tabla 3 muestra que el modelo basado Xg-Boost

obtuvo el mejor desempeño (menor valor de perdida), seguido por los modelos (en orden): Random Forest, Redes Neuronales, Elastic Net y Regresión Logística.

Model	**Score (M.A.E)**
XGBoost	250,158,377.90
Random Forest	251,668,956.20
Neural Network	256,784,383.15
Elastic Net	276,815,342.53
Linear Regression	307,729,413.94

Table 3: Resultados de los modelos utilizados en evaluación. El score corresponde al error absoluto medio en evaluación.

Es interesante analizar la interpretabilidad de dichos resultados para este caso particular. Recordemos que en la expresión del error absoluto medio 6 indica la perdida absoluta promedio por parte del modelo, lo cual corresponde (en este estudio) a la diferencia absoluta entre el precio predicho y el precio real de las propiedades. De esta manera, basado en el mejor modelo obtenido, en promedio se espera que exista una diferencia de 250.158 millones de Pesos Colombianos entre el precio predicho por el modelo construido y el valor real del predio.

4 Conclusiones

Es posible concluir que modelar el precio de las viviendas como una función de algunos de sus atributos, plantea el reto de determinar dichos atributos junto con las relaciones complejas entre estos y el precio del predio. Aunque la teoría hedónica no da respuesta explicita a dicho interrogante, el acceso a la gran cantidad de datos sobre viviendas y el avance vertiginoso de los modelos estadísticos y computacionales puede resultar en una gran aproximación a una solución. Este trabajo se centro en la tarea de predecir el precio de 10,286 viviendas en la localidad de Chapinero usando información de 38,644 viviendas de la ciudad de Bogotá. Tras un análisis de la literatura y una evaluación empírica de los datos, se construyo un set de predictores basados en la ubicación espacial de las viviendas y algunos atributos propios de las mismas como: el número de habitaciones, baños y superficie cubierta.

Una vez construido el set de datos, se procedió a entrenar 5 modelos con diferentes características en busca del mejor desempeño en evaluación de los mismos. Lo anterior, implico una exploración detallada de hiperparámetros de cada modelo con el fin de alcanzar el mejor rendimiento en cada caso. De esta forma, se encontró que un modelo basado en arboles y bosting obtuvo el mejor rendimiento debido a las características de los datos y su manejo de datos faltantes. En múltiples casos el manejo básico de datos faltantes mediane la imputación de la mediana constituyo una introducción de sesgo importante que pudo afectar de manera negativa a los otros modelos evaluados, sin embargo, este tipo de estrategias son necesarias en algunos casos cuando el número de datos faltantes es alto o se carece de otros predictores que pudiesen brindar esta información.

Aunque el modelo basado en XgBoost mostró un desempeño aceptable, con un valor medio de perdida absoluta de 250.158 millones de pesos, existe un gran campo de mejora mediante la imputación de nuevos predictores tanto espaciales como propios de la vivienda que permitan tener más información sobre el predio. Por ejemplo, usando información presente en las descripciones de las viviendas, trabajos futuros pueden contemplar utilizar modelos de lenguajae avanzados que puedan obtener predictores como número de parqueaderos, piso de la vivienda, entre otros. Adicionalmente, modelos más complejos basados en ensamble pueden usarse como *super-learners*, los cuales combinen enfoques diferentes grados de complejidad. De esta forma, el proceso de validación cruzada se haría más robusto al combinar estrategias con un mayor grado de sesgo junto con estrategías de mayor variabilidad, obteniendo así mejores resultados.

5 Apendice

5.1 Matriz de predictores

La matriz de predictores es de dimensión 22110 x 64 para el set de entrenamiento y de 10286 x 64 para el conjunto de evaluación.

Matriz de Predictores

$$\begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,64} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,64} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,64} \end{bmatrix} \quad \text{de dimensiones: } \begin{cases} 22110 \times 64 & \text{(Entrenamiento)} \\ 10286 \times 64 & \text{(Evaluación)} \end{cases}$$

La totalidad de columnas con su nombre textual del script se muestran a continuación:

Lista de Variables:

- surface_covered
- bedrooms
- bathrooms
- ESTRATO
- dist_centro
- distancia_col
- distancia_Calle_72_(Avenida_Chile)
- distancia_a_Parque_de_la_93
- distancia_Santa_Bárbara
- dist_chapinero
- distancia_a_Zona_G
- distancia_a_Zona_T
- distancia_a_Usaquén
- distancia_a_La_Macarena
- distancia_a_Chapinero_Alto
- distancia_a_La_Candelaria
- distancia_a_Calle_Bonita_(Calle_30)
- distancia_estacion
- distancia_parque
- surface_bedrooms
- surface_bathrooms
- bedrooms_bathrooms
- surface_dist_centro
- surface_estrato
- bedrooms_dist_centro
- surface_dist_col
- bedrooms_dist_col
- bathrooms_dist_col
- surface_dist_parque
- bedrooms_dist_parque
- bathrooms_dist_parque
- surface_dist_estacion
- bedrooms_dist_estacion
- bathrooms_dist_estacion
- surface_Calle_72_(Avenida_Chile)
- bedroomsCalle_72_(Avenida_Chile)
- bathrooms_72_(Avenida_Chile)
- surface_Parque_de_la_93
- bedrooms_Parque_de_la_93
- bathrooms_Parque_de_la_93
- surface_Centro_Empresarial_Santa_Bárbara
- bedrooms_Centro_Empresarial_Santa_Bárbara
- bathrooms_Centro_Empresarial_Santa_Bárbara
- surface_Zona_G
- bedrooms_Zona_G
- bathrooms_Zona_G
- surface_Zona_T
- bedrooms_Zona_T
- bathrooms_Zona_T
- surface_Usaquén
- bedrooms_Usaquén

- bathrooms_Usaquén
- surface_La_Macarena
- bedrooms_La_Macarena
- bathrooms_La_Macarena
- surface_Chapinero_Alto
- bedrooms_Chapinero_Alto
- bathrooms_Chapinero_Alto
- surface_La_Candelaria
- bedrooms_La_Candelaria
- bathrooms_La_Candelaria
- surface_Calle_Bonita_(Calle_30)
- bedrooms_Calle_Bonita_(Calle_30)
- bathrooms_Calle_Bonita_(Calle_30)

References

[1] Kelly C. Bishop, Nicolai V. Kuminoff, H. Spencer Banzhaf, Kevin J. Boyle, Kay von Gravenitz, Jaren C. Pope, and Christopher D. Timmins. Best practices for using hedonic property value models to measure willingness to pay for environmental quality. *Review of Environmental Economics and Policy*, 2020.

[2] S. Peeta, Thomas Gebhardt, and M. Stumpf González. Análisis de precios hedónicos de viviendas. *Revista Ingeniería de Construcción*, 34(2):215–220, 2019.

[3] David Albouy, Peter Christensen, and Ignacio Sarmiento-Barbieri. Unlocking amenities: Estimating public good complementarity. *Journal of Public Economics*, 182:104110, 2020.

[4] Sandra E. Black. Do better schools matter? parental valuation of elementary education. *The Quarterly Journal of Economics*, 114(2):577–599, 1999.

[5] Nick Tsivanidis. Evaluating the impact of urban transit infrastructure: Evidence from bogotá's transmilenio. *American Economic Review*, Forthcoming.

[6] Peter C. Boxall, Wing H. Chan, and Melville L. McMillan. The impact of oil and natural gas facilities on rural residential property values: A spatial hedonic analysis. *Resource and Energy Economics*, 27(3):248–269, 2005.

[7] Departamento Administrativo Nacional de Estadística (DANE). Estratificación socioeconómica para servicios públicos domiciliarios, 2024. Accedido: 14 de noviembre de 2024.

[8] OpenStreetMap contributors. Openstreetmap: Collaborative mapping platform. https://www.openstreetmap.org, 2024. Accessed: 2024-11-22.