

# Taller 3

Paula Amado y David Acero

## 1 Introducción

Una nueva start-up en el sector inmobiliario busca establecer una estrategia sólida para la compra y venta de propiedades en Chapinero, Bogotá. Su principal objetivo es adquirir la mayor cantidad de inmuebles en esta zona estratégica mientras optimiza los costos y minimiza los riesgos. Para ello, han solicitado el desarrollo de un modelo predictivo que permita tomar decisiones fundamentadas y maximizar el rendimiento de sus inversiones.

Uno de los desafíos más significativos es la falta de información detallada sobre las propiedades en Chapinero, lo que dificulta la estimación precisa de precios y la definición de estrategias efectivas. Además, la empresa está alerta ante los riesgos asociados al uso de modelos predictivos en el sector, recordando casos como el de Zillow, donde la sobreestimación de precios derivó en pérdidas millonarias y un impacto negativo en su operación. Este proyecto busca superar tales desafíos mediante el desarrollo de un modelo robusto, adaptado a las limitaciones de datos, que garantice predicciones confiables y facilite una toma de decisiones eficiente y sostenible.

La literatura existente subraya la relevancia de aplicar enfoques avanzados de machine learning para mejorar la precisión en la predicción de precios de vivienda. Por ejemplo, Truong et al. (2020) destacaron cómo técnicas como la regresión híbrida y la generalización apilada combinan estrategias como Random Forest, XGBoost y LightGBM para manejar datos complejos, reducir errores y equilibrar el sesgo y la varianza en los modelos. Estas metodologías resultan clave para prevenir errores costosos y asegurar resultados sostenibles.

Por su parte, Adetunji et al. (2022) demostraron que el algoritmo Random Forest es particularmente eficaz para predecir precios de viviendas al considerar factores como ubicación, tamaño y condiciones físicas. Asimismo, investigaciones como las

de Varma et al. (2018) han explorado las ventajas de combinar diferentes métodos de predicción para obtener estimaciones más precisas. Inspirados en estos avances, este proyecto aprovechará los conocimientos existentes para construir un modelo predictivo que se ajuste de manera óptima al contexto de Chapinero y sus particularidades.

## 2 Datos

## 3 Modelos y resultados

Con base en las variables identificadas como relevantes para la predicción de precios en esta localidad, se desarrollaron varios modelos con el objetivo de ajustar de manera óptima las estimaciones a las características del problema. En este ejercicio se implementaron modelos de regresión lineal, red elástica, árboles aleatorios, técnicas de boosting y redes neuronales. Entre estos enfoques, el modelo que mostró el mejor desempeño fue XGBoost, por lo que se profundizará en los resultados obtenidos con esta metodología.

### 3.1 XgBoost

Este es un método de aprendizaje supervisado que se basa en árboles de decisión potenciados por gradiente. Una de las principales ventajas de este modelo es la escalabilidad a grandes volúmenes de datos.

### 3.2 Estimación

Matemáticamente, podemos expresar nuestro modelo de la siguiente manera:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}$$

Donde:

- $K$  es el número de árboles.
- $f_k$  representa el espacio funcional de  $\mathcal{F}$ .
- $\mathcal{F}$  es el conjunto de posibles CARTs.

- $\hat{y}_i$  Representa el precio de la vivienda a predecir.

La función objetivo asociada a este modelo se define como:

$$\text{obj}(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

Donde:

- $l(y_i, \hat{y}_i)$  es la función de pérdida que mide la discrepancia entre los valores reales  $y_i$  y las predicciones  $\hat{y}_i$ .
- $\Omega(f_k)$  es un término de regularización que controla la complejidad de los árboles, ayudando a prevenir el sobreajuste.

En este contexto, XGBoost utiliza esta estructura matemática para optimizar las predicciones mediante el ajuste iterativo de los árboles de decisión. Cada árbol corrige los errores cometidos por los árboles previos, lo que permite al modelo alcanzar un equilibrio entre sesgo y varianza, y producir estimaciones más precisas. Esta característica lo convierte en una herramienta altamente eficaz para problemas de predicción complejos.

### 3.2.1 Parámetros de XGboost

Los principales parámetros de xgboost que permiten optimizar el modelo son los siguientes:

- Tasa de aprendizaje: Este parámetro es útil para controlar la rapidez del algoritmo gracias al descenso del gradiente. De esta forma, este valor entre más pequeño que sea garantiza una mayor precisión, dado que permite encontrar el punto de optimización con mayor precisión. No obstante, tiene asociado un costo computacional.
- Profundidad máxima de los árboles: Controla la complejidad de los árboles para ajustar o evitar el sobreajuste.
- Gamma: Umbral para reducir la pérdida y evitar divisiones innecesarias.
- Submuestra: Este parámetro es útil para controlar el porcentaje de observaciones usados para cada iteración.
- Parámetros de regularización:

- Lambda: Penalización para reducir sobreajuste.
- Alpha: Penalización para manejar características irrelevantes.

Para la obtención de los parámetros óptimos, se realiza una red en la que se permita la iteración entre los parámetros que la literatura considera pertinentes. Para evaluar la pertinencia de cada uno de estos parámetros, se realiza cross validation a través de 5 divisiones. De esta forma, al entrenar el modelo los hiperparámetros obtenidos fueron los siguientes:

	Parámetros sugeridos	Parámetros obtenidos
Número de rondas	100, 250	250
Profundidad máxima	2, 4	4
Tasa de aprendizaje	0.01, 0.05	0.05
Gamma	0,1	0
Submuestra	0.4, 0.8	0.8

Este modelo obtuvo un MAE (Error Absoluto Medio) en Kaggle de \$221,666,308.55. Esto significa que, en promedio, nuestras predicciones presentan una desviación de aproximadamente 221 millones de pesos frente al precio real de las viviendas según los datos de Properati.

### 3.3 Otros modelos

Tal como se mencionó previamente, se realizaron estimaciones adicionales buscando el mejor que tuvieran una mejor predicción.

#### 3.3.1 Regresión lineal

En su forma matricial, el modelo de regresión lineal se expresa como:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

Donde:

- $\mathbf{y}$  es un vector  $n \times 1$  que representa los valores observados del precio de la vivienda.

- $\mathbf{X}$  es una matriz  $n \times (p + 1)$ , que representan las variables independientes (Metros cuadrados, habitaciones, Parqueadero, Estrato, ascensor, valor de referencia y demás).
- $\beta$  es un vector  $(p + 1) \times 1$  que contiene los coeficientes del modelo.
- $\epsilon$  es un vector  $n \times 1$  que representa los términos de error (residuos).

Al evaluar este modelo en kaggle, se obtuvo un MAE de \$374,114,594.82. De esta forma, este modelo se consolida como el de peor desempeño.

### 3.3.2 Red elástica

El modelo de red elástica (Elastic Net) es un método de regresión lineal regularizado que combina las penalizaciones de Lasso (L1) y Ridge (L2). Este modelo es especialmente útil cuando las variables independientes están altamente correlacionadas o cuando el número de predictores es mayor al número de observaciones.

El modelo de Elastic Net encuentra los coeficientes  $\beta$  que minimizan la siguiente función objetivo:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \mathbf{y} - \mathbf{X}\beta^2 + \lambda_1\beta_1 + \lambda_2\beta_2^2 \right\}$$

Donde:

- $\mathbf{y} - \mathbf{X}\beta^2$ : Error cuadrático entre las observaciones y las predicciones.
- $\beta_1 = \sum_{j=1}^p \beta_j$ : Penalización L1 que favorece la selección de variables al forzar algunos coeficientes a ser exactamente cero.
- $\beta_2^2 = \sum_{j=1}^p \beta_j^2$ : Penalización L2 que reduce la magnitud de los coeficientes para evitar el sobreajuste.
- $\lambda_1$  y  $\lambda_2$ : Hiperparámetros que controlan la intensidad de las penalizaciones L1 y L2, respectivamente.

Elastic Net combina ambas penalizaciones de manera controlada a través del parámetro de mezcla  $\alpha$ :

$$\lambda_1 = \alpha\lambda, \quad \lambda_2 = (1 - \alpha)\lambda$$

Donde:

- $\lambda$ : Hiperparámetro global que ajusta la regularización total.
- $\alpha \in [0, 1]$ : Controla el balance entre L1 y L2.

Al entrenar este modelo con los datos de entrenamiento y usar las diferentes variables que se han explicado anteriormente, se obtuvo un puntaje de \$286092295.87.

### 3.3.3 Redes neuronales

Las redes neuronales son un modelo de aprendizaje profundo inspirado en el funcionamiento del cerebro humano. Están diseñadas para aprender relaciones no lineales complejas entre los datos de entrada y salida mediante la combinación de capas de neuronas conectadas.

En términos generales, una red neuronal consta de las siguientes partes:

- **Capa de entrada:** Recibe las variables independientes ( $\mathbf{X}$ ).
- **Capas ocultas:** Procesan las entradas mediante transformaciones no lineales.
- **Capa de salida:** Genera la predicción ( $\hat{\mathbf{y}}$ ).

El funcionamiento de una red neuronal se describe matemáticamente de la siguiente forma:

$$\mathbf{a}^{(l)} = f^{(l)}(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})$$

Donde:

- $l$ : Índice de la capa.
- $\mathbf{a}^{(l)}$ : Vector de activaciones de la capa  $l$ .
- $\mathbf{W}^{(l)}$ : Matriz de pesos  $m_l \times m_{l-1}$  de la capa  $l$ .
- $\mathbf{b}^{(l)}$ : Vector de sesgos de la capa  $l$ .
- $f^{(l)}$ : Función de activación de la capa  $l$ .

El proceso completo de la red se puede representar como:

$$\hat{\mathbf{y}} = f^{(L)}(\mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{b}^{(L)})$$

Donde  $L$  es la última capa (capa de salida).

Adicionalmente, este modelo se entrenó con validación cruzada espacial, en la que se dividieron el mapa de datos en cinco grillas y se usó esto para encontrar los parámetros óptimos para este modelo. Este modelo tuvo un desempeño de \$ 333,166,543.70 en kaggle.

## 4 Conclusiones y recomendaciones

## 5 Referencias

- Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2021). House price prediction using random forest machine learning technique. *Procedia Computer Science*, 199, 806–813.
- Varma, A., Sarma, A., Doshi, S., & Nair, R. (2018). House price prediction using machine learning and neural networks.
- Truong, Q., Nguyen, M., Dang, H., & Mei, B. (2020). Housing price prediction via improved machine learning techniques. *Procedia Computer Science*, 174, 433–442.