Informe Proyecto Final

Daniel Andrés Agudelo García, Paulina García Aristizábal, Emanuel Munera Pérez
Facultad de Ingeniería, Universidad de Antioquia
Medellín, Colombia
Materia: Modelos II
Grupo: 12
daniel.agudelo9@udea.edu.co, paulina.garcia1@udea.edu.co, emanuel.munera@udea.edu.co

I. Introducción

La estimación del valor de arriendo de una vivienda es un desafío relevante en contextos urbanos donde los precios fluctúan según factores estructurales, geográficos y socioeconómicos. Variables como la ubicación, el tamaño, el número de habitaciones o el nivel de amueblamiento influyen de manera conjunta en el precio final, haciendo que los métodos tradicionales de valoración resulten limitados.

En este escenario, las técnicas de *Machine Learning* ofrecen una alternativa eficiente para modelar relaciones y descubrir patrones en grandes volúmenes de datos. El presente proyecto aplica este enfoque al *House Rent Prediction Dataset* —disponible en Kaggle— que recopila información de propiedades en distintas ciudades de India con el fin de predecir el valor mensual de arriendo (`Rent`).

Mediante un enfoque de aprendizaje supervisado, el modelo busca aprender la relación entre las características del inmueble y su valor de renta, permitiendo estimaciones más precisas y objetivas. Este tipo de herramienta puede servir de apoyo a propietarios, arrendadores y plataformas inmobiliarias, contribuyendo a una mayor transparencia y eficiencia en el mercado del alquiler.

II. Descripción del problema

El acceso a la vivienda es uno de los factores más determinantes en la calidad de vida urbana. En ciudades con una alta densidad poblacional y dinámicas inmobiliarias cambiantes, predecir el valor de arriendo de una vivienda puede ser una tarea compleja, influenciada por variables como la ubicación, el tamaño, las comodidades del inmueble, la infraestructura disponible en la zona y las condiciones del mercado en cada momento.

La estimación precisa del precio de arriendo constituye un desafío tanto para arrendadores como para arrendatarios. Por un lado, los propietarios y agentes inmobiliarios buscan fijar precios competitivos que maximicen la ocupación sin sacrificar rentabilidad; por otro, los arrendatarios requieren información confiable que les permita tomar decisiones informadas y evitar sobrecostos injustificados. Sin embargo, en la práctica, los métodos tradicionales de valoración suelen basarse en la experiencia subjetiva o en comparaciones limitadas a propiedades cercanas, lo cual introduce sesgos y dificulta capturar la complejidad real del mercado.

Además, los precios de alquiler están determinados por múltiples factores. Características estructurales de la vivienda, como el número de habitaciones o el área construida, se combinan con factores contextuales como la ciudad, el vecindario o el nivel de amueblamiento. Esta combinación de variables hace que el problema sea ideal para ser abordado mediante técnicas de Machine Learning, las cuales son capaces de modelar relaciones complejas y capturar patrones ocultos en los datos.

En este contexto, una solución basada en aprendizaje automático permite automatizar el proceso de estimación del precio de arriendo, ofreciendo una herramienta útil para agencias inmobiliarias, arrendadores y potenciales inquilinos. Este tipo de sistema podría integrarse fácilmente en plataformas de búsqueda de vivienda, recomendando precios sugeridos en tiempo real o alertando sobre valores atípicos que podrían indicar sobrevaloración o subvaloración del inmueble.

II-A. Aproximación desde Machine Learning

Dado que el objetivo del proyecto es estimar un valor numérico continuo del cual se tienen datos etiquetados —es decir, para cada propiedad se conoce el valor real del arriendo (`Rent`)—, se adoptará un enfoque de aprendizaje supervisado. En este paradigma, el modelo aprende una función de mapeo entre un conjunto de variables de entrada (características del inmueble y su contexto) y una variable objetivo conocida, con el propósito de minimizar el error entre las predicciones y los valores reales.

El uso de aprendizaje supervisado resulta adecuado no solo porque se dispone de datos históricos confiables, sino también porque posibilita la comparación entre distintos algoritmos de regresión, tales como la Regresión Lineal, los Árboles de Decisión, el *Random Forest* o las Redes Neuronales Multicapa (MLP). Esto permitirá analizar el equilibrio entre interpretabilidad y capacidad predictiva, identificando cuál modelo logra capturar mejor las relaciones no lineales entre las variables y el precio de arriendo.

En síntesis, este enfoque proporciona una base sólida para desarrollar un sistema de predicción de rentas interpretable, eficiente y adaptable, que aproveche el potencial del aprendizaje automático para ofrecer estimaciones más precisas y fundamentadas.

II-B. Descripción de la base de datos

El conjunto de datos empleado, *House Rent Prediction Dataset*, contiene información de diferentes propiedades en India, recopiladas a partir de listados reales de arriendo.
Número de muestras: 4,746 registros.
Número de variables: 12 características, incluyendo la variable objetivo.
Variable objetivo: `Rent` — valor del arriendo mensual en rupias.

Las principales variables incluidas en el dataset son las siguientes:

Variable	Descripción
`Posted On (Cat)`	Fecha de Publicación.
`BHK (Num)`	Número de habitaciones.
`Rent(Num)`	Valor mensual del arriendo (variable objetivo).
`Size(Num)`	Área total de la propiedad (en pies cuadrados).
`Floor (Cat)`	Piso o nivel en el que se encuentra la vivienda.
`Area Type (Cat)`	Tipo de medición del área (Super built-up, Built-up, Carpet).
`Area Locality (Cat)`	Nombre del vecindario o zona
`City (Cat)`	Ciudad donde se ubica la propiedad.
`Furnishing Status (Cat)`	Nivel de amueblamiento (Unfurnished, Semi-Furnished, Furnished).
`Tenant Preferred (Cat)`	Tipo de arrendatario preferido (Family, Bachelors, Company).
`Bathroom (Num)`	Número de baños.
`Point of Contact (Cat)`	Medio de contacto del anunciante.

Durante la inspección inicial de los datos no se identificaron valores faltantes en las variables. Para la limpieza de datos fue necesario estandarizar los formatos de texto a minúsculas y convertir algunas variables categóricas en representaciones numéricas mediante codificación *One Hot encoding*. Dichas variables fueron `Point of Contact`, `Tenant Preferred`, `Furnishing Status`, `Area Type`, `City`. La columna `Floor` fue dividida en 2: `Current Floor` y `Total Floors`, ambas numéricas representando el piso actual del inmueble y el total de pisos del edificio en el que está ubicado. Para la variable `Posted On` también se hizo una división en 2 columnas, una columna para el día en el que fue publicada la oferta y otra para el mes. No se tuvo en cuenta el año ya que todas las fechas corresponden al 2022. Finalmente, la variable `Area Locality` presentaba una gran cantidad de valores únicos (2000+), por lo que con un mapeo de frecuencia se realizó un *One Hot Encoding* con las localidades que sobrepasaran un umbral de 20 veces apareciendo en las muestras. Las localidades que no sobrepasaran dicho umbral fueron agrupadas en una columna `Locality Rare`. En total, al final de la limpieza de datos el dataset resultó compuesto por 33 columnas o variables.

III. Estado del arte

Artículo I: Housing Price Prediction Using Machine Learning Algorithms in COVID-19 Times [1]

El estudio de Mora-García *et al.* (2022), publicado en *Land*, analiza la predicción del precio de vivienda en Alicante durante el periodo de pandemia, empleando técnicas de regresión supervisada sobre más de 47 000 registros.

- **Configuración del problema:** Tarea de regresión para estimar precio de venta de vivienda.
- **Modelos evaluados:** Regresión Lineal, Random Forest, Extra Trees, Gradient Boosting, XGBoost y LightGBM.
- **Validación:** División entrenamiento/validación/prueba bajo esquema *pooled cross-sectional*.
- **Métricas:** MAE, MSE, RMSE y R^2. La RMSE se define como:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

- **Resultados:** Métodos de boosting obtuvieron los menores errores; variables espaciales y socioeconómicas mejoraron el desempeño.

Referencia: Mora-García, R. T., et al. (2022). *Housing Price Prediction Using Machine Learning Algorithms in COVID-19 Times*. Land, 11(10), 2100.

Artículo II: Machine Learning Techniques for Predicting Home Rental Prices in India (Jayadharshini et al., 2023)

El estudio de Jayadharshini *et al.* (2023) aborda la predicción de precios de arriendo en ciudades indias, utilizando datos comparables a los de este proyecto.

- **Configuración del problema:** Regresión para estimar el precio de alquiler usando variables estructurales y contextuales.
- **Modelos evaluados:** Regresión Lineal, Random Forest, Gradient Boosting, XGBoost.
- **Validación:** Validación cruzada *k-fold* y división independiente de prueba.
- **Métricas:** MAE, RMSE, R^2 y MAPE:

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- **Resultados:** XGBoost y Gradient Boosting ofrecieron mejor desempeño; la correcta codificación de categóricas fue clave.

Referencia: Jayadharshini, P., et al. (2023). *Machine Learning Techniques for Predicting Home Rental Prices in India*. Applied and Computational Engineering.

Artículo III: Predicting Rental Price of Lane Houses in Shanghai with Machine Learning Methods and Large Language Models (Chen & Si, 2024)

El trabajo de Chen y Si (2024), disponible en arXiv, analiza la predicción del precio de arriendo en "lane houses" de Shanghái utilizando un enfoque mixto: métodos tradicionales de ML y modelos de lenguaje de gran escala (LLM) como ChatGPT.

- **Configuración del problema:** Regresión supervisada usando un dataset público de 2 609 registros con características estructurales, distritales y de mobiliario.
- **Modelos evaluados:**
  - Métodos tradicionales: MLR, Ridge, Lasso, Decision Tree y Random Forest.
  - LLM: ChatGPT en configuraciones 0-shot, 1-shot, 5-shot y 10-shot mediante el método *prompt-as-prefix*.
- **Metodología de validación:** División entrenamiento/prueba 80-20. Los modelos tradicionales se ajustaron con GridSearchCV; los LLM fueron evaluados con diferentes números de ejemplos incluidos en el *prompt*.
- **Métricas empleadas:** MSE, MAE y R^2, definidas como:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \qquad \text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|,$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}.$$

- **Resultados:**
  - Entre los modelos clásicos, **Random Forest** obtuvo el mejor desempeño (menor MSE y MAE).
  - ChatGPT mostró mejoras significativas a medida que aumentaron los *shots*, alcanzando **R^2=0.80** en el escenario de 10-shot, superando a los modelos tradicionales.
- **Conclusiones:** Los autores resaltan el potencial de los LLM para predicción numérica cuando se formulan adecuadamente los *prompts*, aunque los modelos clásicos continúan siendo más consistentes y eficientes en datos estructurados.

Referencia: Chen, T., & Si, S. (2024). *Predicting Rental Price of Lane Houses in Shanghai with Machine Learning Methods and Large Language Models*. arXiv:2405.17505.

Artículo IV: Spatial prediction of apartment rent using regression-based and machine learning-based approaches with a large dataset (Yoshida & Seya, 2021)

El trabajo de Yoshida y Seya (2021) investiga la predicción espacial del arriendo de apartamentos a gran escala, comparando enfoques basados en regresión con métodos de machine learning y considerando la dependencia espacial entre observaciones.

- **Configuración del problema:** Se aborda una tarea de regresión supervisada sobre un conjunto de datos muy grande (hasta 1 millón de observaciones) de alquiler de apartamentos en Japón.
- **Modelos utilizados:**
  - Regresión: proceso Gaussiano con vecinos más cercanos (NNGP) para incorporar correlación espacial (kriging).
  - Machine Learning: XGBoost, Random Forest (RF) y red neuronal profunda (DNN).
- **Metodología de validación:** Se realizan comparaciones "out-of-sample" para distintos tamaños de muestra (n = 10^4, 10^5, 10^6) para estudiar cómo crece el desempeño con más datos y cómo afecta la escala espacial.
- **Métricas empleadas:** Se evalúan el error en escala logarítmica y real (no solo en logaritmo), usando medidas típicas de regresión (por ejemplo, RMSE o MSE).
- **Resultados:**
  - XGBoost logra la mayor precisión en predicción para todos los tamaños de muestra y para diferentes bandas de precio.
  - Random Forest también supera al modelo de regresión espacial (NNGP) en muchas configuraciones.
  - En RF, agregar las coordenadas espaciales (latitud/longitud) como variables predictoras es suficiente para capturar gran parte de la dependencia espacial, lo que simplifica el modelo.
- **Conclusiones:** Los autores muestran que los métodos clásicos de machine learning (como XGBoost y RF) pueden superar enfoques geostatísticos sofisticados cuando se tienen muchos datos, incluso sin modelar explícitamente toda la estructura espacial compleja.

Referencia: Yoshida, T., & Seya, H. (2021). *Spatial prediction of apartment rent using regression-based and machine learning-based approaches with a large dataset*. arXiv:2107.12539.

IV. Entrenamiento y evaluación de los modelos

Para este proyecto, evaluamos 6 modelos de distintos tipos. 5 de ellos fueron los requeridos en la guía del proyecto y el 6to modelo fue el XGBoost, el cuál decidimos implementar dado que varios artículos revisados durante el estado del arte presentaron este tipo de modelo con muy buenos resultados.

Esta fue la malla de hiperparámetros que utilizamos:

Cuadro I
Hiperparámetros analizados para Regresión Lineal

Hiperparámetro	Valores evaluados
–	No aplica (sin túnel de hiperparámetros)

Cuadro II
Hiperparámetros analizados para Árbol de Decisión

Hiperparámetro	Malla de valores
max_depth	{None, 5, 10, 20, 30, 50}
min_samples_split	{2, 5, 10, 20}
min_samples_leaf	{1, 2, 4, 10}
max_features	{None, sqrt, log2}
criterion	{squared_error, absolute_error}

Cuadro III
HIPERPARÁMETROS ANALIZADOS PARA KNN

Hiperparámetro	Malla de valores
n_neighbors	1 a 49
weights	{uniform, distance}
metric	{euclidean, manhattan}

Cuadro IV
HIPERPARÁMETROS ANALIZADOS PARA SVM (SVR)

Hiperparámetro	Malla de valores
C	{1, 10, 50, 100}
gamma	{scale, auto, 0.01, 0.001}
epsilon	{0.1, 0.2, 1, 5}

Cuadro V
HIPERPARÁMETROS ANALIZADOS PARA XGBOOST

Hiperparámetro	Malla de valores
n_estimators	{100, 300, 500}
max_depth	{3, 5, 7}
learning_rate	{0.01, 0.05, 0.1}
subsample	{0.7, 0.9, 1.0}
colsample_bytree	{0.7, 0.9, 1.0}
gamma	{0, 1}

Cuadro VI
HIPERPARÁMETROS ANALIZADOS PARA RED NEURONAL (MLPREGRESSOR)

Hiperparámetro	Malla de valores
hidden_layer_sizes	{(50,), (100,), (50,50), (100,50)}
activation	{relu, tanh}
solver	{adam}
learning_rate_init	{0.001, 0.01}
alpha	{0.0001, 0.001, 0.01}

IV-A. Métricas de desempeño y justificación

Para la evaluación cuantitativa de los modelos de regresión se utilizaron las métricas descritas a continuación. La selección de estas medidas se fundamenta tanto en su interpretabilidad como en su capacidad para capturar distintos aspectos del error cometido por los modelos.

IV-A0a. Mean Absolute Error (MAE): El MAE es la métrica principal empleada en este proyecto. Se define como el promedio del valor absoluto de los errores de predicción. Su principal ventaja es que mantiene las mismas unidades de la variable objetivo (*Rent*), lo que permite interpretar directamente la magnitud típica del error cometido por el modelo. Además, es una medida robusta frente a valores atípicos, pues penaliza linealmente los desvíos. Por estas razones, fue utilizada como criterio de optimización en la búsqueda de hiperparámetros mediante `GridSearchCV`.

IV-A0b. Root Mean Squared Error (RMSE): El RMSE complementa al MAE al penalizar de manera más severa los errores grandes, incorporando un término cuadrático. Esto permite detectar modelos que, aunque mantengan un error medio aceptable, presentan desviaciones considerables en casos particulares. Dado que estos errores suelen ser relevantes en problemas de predicción de precios, el RMSE constituye una medida indispensable para comparar modelos.

IV-A0c. Intervalos de confianza al 95 %: Para cada una de las métricas principales (MAE y RMSE) se estimaron intervalos de confianza del 95 %, con el fin de cuantificar la incertidumbre asociada a cada estimación. Estos intervalos permiten evaluar si las diferencias entre modelos son estadísticamente significativas, lo cual agrega rigor al proceso de selección del modelo final.

IV-B. Resultados del entrenamiento

En esta sección se presentan los resultados obtenidos durante el proceso de validación cruzada para cada uno de los modelos evaluados. Para cada caso se reporta el error absoluto medio (MAE) junto con su intervalo de confianza al 95 % (IC95 %), lo cual permite cuantificar la variabilidad del modelo ante diferentes particiones del conjunto de entrenamiento. Adicionalmente, se incluye el RMSE para el modelo lineal, dado que su comportamiento exhibió una dispersión considerable de errores.

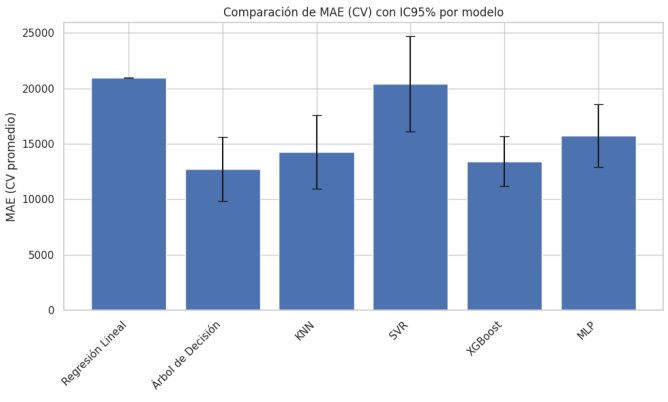

Figura 1. Comparación del MAE promedio obtenido mediante validación cruzada para todos los modelos entrenados. Las barras de error representan los intervalos de confianza al 95 %.

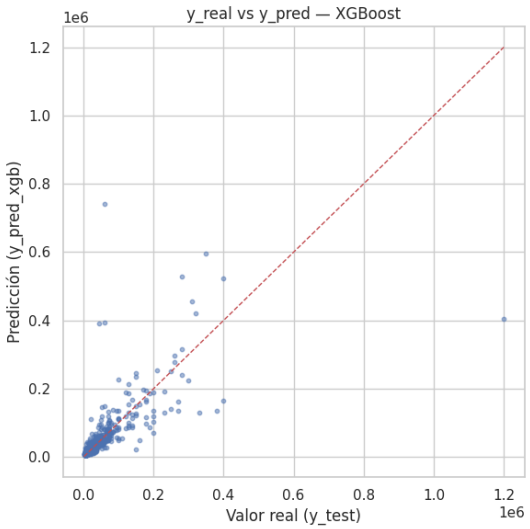

Figura 2. Relación entre valores reales y predichos para el modelo XGBoost en el conjunto de prueba. La línea diagonal representa la predicción perfecta.

Los resultados evidencian diferencias claras en la capacidad predictiva de los modelos. El modelo de **Regresión Lineal** obtuvo el peor desempeño, con un MAE promedio de MAE $=$ 22002,72 y un IC95 % estrecho [20738,87, 23266,57], lo cual indica que, aunque su error es alto, su variabilidad es relativamente baja. El RMSE asociado fue considerablemente elevado, lo que sugiere la presencia de errores de gran magnitud producidos por valores atípicos o relaciones no lineales que el modelo no es capaz de capturar.

El modelo basado en **Máquinas de Vectores de Soporte (SVR)** presentó también un rendimiento limitado, con un MAE $=$ 20406,79 e intervalos amplios [16110,50, 24703,08], reflejando alta sensibilidad a la partición de datos y menor estabilidad en comparación con otros métodos no lineales. De forma similar, el modelo **MLP** mostró un error promedio de MAE $=$ 15731,55 y un IC95 % [12901,50, 18561,59], confirmando que, aunque logra capturar cierta no linealidad, su desempeño no supera al de los métodos basados en árboles.

Por otro lado, el modelo **KNN** logró un desempeño competitivo con MAE $=$ 14249,99 y un intervalo de confianza relativamente amplio [10912,90, 17587,09], lo cual indica que su efectividad depende de la distribución local de los datos y, por lo tanto, muestra mayor variabilidad en los diferentes folds.

Los dos mejores resultados se obtuvieron con modelos basados en árboles. En particular, el **Árbol de Decisión** presentó el menor error promedio de todos los modelos evaluados, alcanzando un MAE $=$ 12699,96 con IC95 % [9815,96, 15583,97]. Su desempeño superior puede explicarse por la capacidad del modelo para capturar relaciones no lineales y por la relativa simplicidad del conjunto de hiperparámetros óptimos encontrados.

Finalmente, el modelo de **XGBoost** obtuvo el segundo mejor resultado general, con un MAE $=$ 13421,20 y un IC95 % [11179,22, 15663,17]. Si bien su error promedio es ligeramente mayor al del árbol de decisión, su intervalo de confianza es más estrecho, lo cual indica una mayor estabilidad y robustez durante el proceso de validación cruzada. Esto se ve reflejado también en la gráfica de error promedio donde XGBoost muestra una combinación favorable entre bajo error y baja variabilidad.

En conjunto, estos resultados permiten concluir que los modelos basados en árboles ofrecen un mejor ajuste para el problema abordado. El **Árbol de Decisión** se posiciona como la mejor alternativa en términos de error absoluto medio, mientras que **XGBoost** constituye la opción más robusta debido a su menor variabilidad entre folds.

V. Reducción de Dimensión

En esta etapa se evaluó si era posible disminuir el número de variables sin deteriorar el desempeño del modelo final. El análisis se realizó únicamente sobre los dos modelos con mejor desempeño: Árbol de Decisión y XGBoost.

Modelo base: Árbol de Decisión

Análisis individual de variables

El análisis comenzó evaluando la capacidad discriminativa de cada variable mediante dos criterios: la correlación individual con la variable objetivo, y la importancia asignada por el Árbol de Decisión. Las variables con correlación absoluta menor a 0.05 o con importancia inferior a 0.01 fueron consideradas como posibles candidatas a eliminación. Aunque se identificaron algunas variables con bajo aporte, la reducción manual de variables no se aplicó en esta fase, ya que el enfoque principal era evaluar métodos de reducción automática (PCA y UMAP). Sin embargo, los resultados confirmaron que varias características poseen poco impacto.

Extracción lineal de características con PCA

Para evaluar la reducción lineal, se aplicó PCA sobre las variables estandarizadas. La curva de varianza explicada mostró que la primera componente captura la mayor proporción de la variabilidad del conjunto, pero el uso de una única componente producía una pérdida excesiva de información. Por esta razón, se seleccionaron dos componentes principales, logrando una reducción de dimensión significativa con un nivel razonable de preservación de información.

Posteriormente, se entrenó nuevamente el Árbol de Decisión usando estas dos componentes como entrada. Los resultados mostraron que el rendimiento del modelo disminuyó con respecto al modelo sin reducción: el MAE aumentó y el RMSE también. Esto indica que, para este dataset en particular, las transformaciones lineales de PCA no capturan suficientemente bien las interacciones relevantes que el modelo necesita para predecir.

Cuadro VII
Comparación antes y después de aplicar PCA

Modelo	MAE	RMSE
Árbol sin PCA	11901.49	33457.65
Árbol con PCA (2 componentes)	14625.57	46378.24

Extracción no lineal de características con UMAP

Para evaluar la reducción no lineal, se aplicó UMAP como técnica de reducción. Al igual que en PCA, se decidió usar dos componentes para mantener coherencia y permitir una comparación similar entre ambos enfoques.

Se entrenó el Árbol de Decisión utilizando las dos componentes generadas por UMAP. Los resultados mostraron nuevamente un deterioro importante en el rendimiento del modelo frente al árbol original. Tanto el MAE como el RMSE aumentaron de forma más notable que en PCA. Esto sugiere que, aunque UMAP es útil para capturar estructuras complejas, las representaciones obtenidas no preservaron adecuadamente la información necesaria para el problema de regresión de este proyecto.

Cuadro VIII
COMPARACIÓN ANTES Y DESPUÉS DE APLICAR UMAP

Modelo	MAE	RMSE
Árbol sin UMAP	11901.49	33457.65
Árbol con UMAP (2 componentes)	18037.31	51723.22

V-A. Conclusiones de la reducción de dimensión para Árbol de Decisión

- Las variables individuales mostraron correlaciones débiles, pero colectivamente aportan información útil, por lo que no se recomienda eliminarlas.
- Tanto PCA como UMAP reducen la dimensionalidad pero degradan el rendimiento del Árbol de Decisión.
- La reducción a 2 componentes no logró preservar suficiente información predictiva.
- Se concluye que para este conjunto de datos y para el Árbol de Decisión, mantener todas las características originales produce el mejor desempeño.

En esta etapa del proyecto, el objetivo fue analizar si es posible reducir la complejidad del conjunto de datos (que consta de 32 variables procesadas) a un espacio latente de solo **5 componentes**, minimizando la pérdida de capacidad predictiva.

Para este análisis, se seleccionó el modelo **XGBoost**, dado que fue el algoritmo que presentó el mejor desempeño y mayor estabilidad durante la fase de entrenamiento (Sección 4). A continuación, se detalla el modelo base y se comparan dos técnicas de reducción: una lineal (PCA) y una no lineal (UMAP).

V-B. Modelo Base: XGBoost

XGBoost (*Extreme Gradient Boosting*) es un algoritmo de aprendizaje supervisado basado en ensambles de árboles de decisión. A diferencia de un árbol tradicional, XGBoost utiliza una técnica de *boosting*, donde los modelos se construyen de manera secuencial: cada nuevo árbol intenta corregir los errores residuales cometidos por los anteriores.

Técnicamente, este modelo fue seleccionado por sus características avanzadas:

- **Regularización (L1 y L2):** Evita el sobreajuste (*overfitting*), penalizando modelos demasiado complejos.
- **Optimización del Gradiente:** Utiliza el descenso de gradiente para minimizar la función de pérdida de manera rápida y eficiente.
- **Manejo de valores faltantes:** Aprende automáticamente la mejor dirección para imputar datos ausentes durante el entrenamiento.

El MAE base de este modelo con las 32 variables originales fue de **13,116.62**.

V-C. Técnica 1: PCA (Reducción Lineal)

El Análisis de Componentes Principales (PCA) es una técnica estadística que transforma las variables originales, posiblemente correlacionadas, en un nuevo conjunto de variables no correlacionadas llamadas componentes principales.

Metodología: Se aplicó PCA para proyectar las 32 dimensiones originales en 5 componentes ortogonales que capturan la mayor varianza posible de los datos. Posteriormente, se re-entrenó el modelo XGBoost utilizando únicamente estos 5 componentes como entrada.

Resultado: El modelo entrenado con PCA obtuvo un MAE de **14,386.98**. Esto indica que las 5 componentes principales lograron retener la mayor parte de la información estructural necesaria para predecir el precio del alquiler, con una pérdida de precisión controlada.

V-D. Técnica 2: UMAP (Reducción No Lineal)

UMAP (*Uniform Manifold Approximation and Projection*) es una técnica de aprendizaje de variedades que busca preservar tanto la estructura local como la global de los datos en un espacio dimensional inferior. A diferencia de PCA, UMAP no asume linealidad en los datos.

Metodología: Se configuró el algoritmo UMAP con 5 componentes, utilizando 15 vecinos cercanos (*n_neighbors*) y una distancia mínima (*min_dist*) de 0.1, con el fin de capturar relaciones complejas y no lineales entre las características de las viviendas. Se entrenó nuevamente el modelo XGBoost con esta nueva representación.

Resultado: El modelo entrenado con UMAP obtuvo un MAE de **17,695.31**. Aunque la reducción fue exitosa en términos computacionales, el aumento en el error fue considerablemente mayor en comparación con el modelo original.

V-E. Comparativa: PCA vs UMAP

Tras evaluar ambas técnicas sobre el modelo XGBoost, se presenta la siguiente comparación de desempeño:

Cuadro IX
COMPARACIÓN DE TÉCNICAS DE REDUCCIÓN EN XGBOOST

Técnica	Tipo	MAE (Error)	Impacto vs Original
Ninguna (Base)	Original	13,116.62	–
PCA	**Lineal**	**14,386.98**	**+9.6 % (Bajo)**
UMAP	No Lineal	17,695.31	+34.9 % (Alto)

La técnica **PCA (Lineal)** resultó ser superior a UMAP para este problema específico.

1. **Estructura de los datos:** El hecho de que PCA funcione mejor sugiere que las relaciones entre las características

de las viviendas (tamaño, número de baños, ubicación) y el precio tienen un comportamiento predominantemente lineal y global, el cual PCA captura eficientemente.

2. **Pérdida de Información:** UMAP, al intentar preservar la topología local, parece haber descartado información global valiosa para la regresión de precios, resultando en un error un 34.9 % más alto.
3. **Eficiencia:** Logramos un modelo XGBoost con solo 5 variables que mantiene una precisión muy cercana al original (diferencia de 1,200 rupias), demostrando que PCA es la estrategia óptima para simplificar este modelo.

VI. Evaluación

Link al video de sustentación