Escola Tècnica Superior d'Enginyeria ETSE-UV

Grado en Ciencia de Datos

Universitat de València

Trabajo Fin de Grado

MODELOS DE PREDICCIÓN DE PRECIOS INMOBILIARIOS EN LA CIUDAD DE VALENCIA.

Autor:
Miguel Caurín Perpiñá

Tutor:
Valero Laparra Pérez-Muelas

VNIVERSITAT DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria ETSE-UV

Trabajo Fin de Grado

MODELOS DE PREDICCIÓN DE PRECIOS INMOBILIARIOS EN LA CIUDAD DE VALENCIA.

Autor: Miguel Caurín Perpiñá

Tutor: Valero Laparra Pérez-Muelas

Declaración de autoría:

Yo, Miguel Caurín Perpiñá, declaro la autoría del Trabajo Fin de Grado titulado "MODELOS DE PREDICCIÓN DE PRECIOS INMOBILIARIOS EN LA CIUDAD DE VALENCIA. " y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.

Valencia, 26 de junio de 2025

Fdo: Miguel Caurín Perpiñá

Resumen:

Este trabajo de fin de grado tiene como objetivo desarrollar un modelo de predicción de precios inmobiliarios en la ciudad de Valencia mediante técnicas de aprendizaje automático. Dado que el mercado inmobiliario está en constante cambio y la valoración de los inmuebles depende de múltiples factores, se ha diseñado un sistema capaz de estimar el precio de venta de una vivienda a partir de sus características.

Para ello, se ha llevado a cabo la recopilación de datos del portal inmobiliario Idealista a través de su API, obteniendo información relevante y actualizada. Posteriormente, los datos han sido procesados y limpiados para garantizar su calidad antes de ser utilizados en la construcción del modelo predictivo.

Se han empleado diversos algoritmos de aprendizaje automático, evaluando su rendimiento mediante métricas estándar como el error absoluto medio (MAE) y la raíz del error cuadrático medio (RMSE) entre otras. Además, se ha realizado un ajuste de hiperparámetros para optimizar la precisión de algunos modelos y mejorar su capacidad de generalización.

Los resultados obtenidos indican que el modelo desarrollado de XGBoost tiene un alto potencial para predecir los precios de los inmuebles con una precisión aceptable. Finalmente, se presentan las conclusiones del estudio y se proponen posibles aplicaciones del modelo en el ámbito inmobiliario, como la valoración automática de propiedades y el análisis de características.

El código desarrollado se puede encontrar en https://github.com/Miguelcp23/Trabajo-TFG

Abstract:

The aim of this thesis is to develop a model for predicting real estate prices in the city of Valencia using machine learning techniques. Given that the real estate market is constantly changing and the valuation of real estate depends on multiple factors, a system capable of estimating the selling price of a property based on its characteristics has been designed.

To do this, data has been collected from the Idealista real estate portal through its API, obtaining relevant and updated information. Subsequently, the data has been processed and cleaned to ensure its quality before being used in the construction of the predictive model.

Several machine learning algorithms have been employed, evaluating their performance using standard metrics such as mean absolute error (MAE) and Root Mean Square Error (RMSE), among others. In addition, hyperparameter tuning has been performed to optimise the accuracy of some models and improve their generalisability.

The results obtained indicate that the developed XGBoost model has a high potential to predict property prices with acceptable accuracy. Finally, the study's conclusions are presented, along with potential applications of the model in the real estate sector, such as automated property valuation and feature analysis.

Resum:

Aquest treball de fi de grau té com a objectiu desenvolupar un model de predicció de preus immobiliaris a la ciutat de València mitjançant tècniques dáprenentatge automàtic. Atés que el mercat immobiliari està en constant canvi i la valoració dels immobles depén de múltiples factors, s'ha dissenyat un sistema capaç d'estimar el preu de venda d'un habitatge a partir de les seues característiques.
Per a això, s'ha dut a terme la recopilació de dades del portal immobiliari Idealista a través de la seua API, obtenint informació rellevant i actualitzada. Posteriorment, les dades han sigut processades i netejades per tal de garantir-ne la qualitat abans de ser utilitzades en la construcció del model predictiu.
S'han emprat diversos algorismes d'aprenentatge automàtic, i s'ha avaluat el seu rendiment mitjançant mètriques estàndard com l'error absolut mitjà (MAE) i l'arrel de l'error quadràtic mitjà (RMSE), entre altres. A més, s'ha realitzat un ajust d'hiperparàmetres per optimitzar la precisió d'alguns models i millorar-ne la capacitat de generalització.
Els resultats obtinguts indiquen que el model desenvolupat amb XGBoost té un alt potencial per predir els preus dels immobles amb una precisió acceptable. Finalment, es presenten les conclusions de l'estudi i es proposen possibles aplicacions del model en l'àmbit immobiliari, com ara la valoració automàtica de propietats i l'anàlisi de característiques.

Agradecimientos:

En primer lugar quiero agradecer a mi familia por el apoyo incondicional que me dan y por animarme siempre a sacar mi mejor versión, enseñadome que siempre hay que luchar y esforzarse al máximo.
Gracias a Juan, Jaume, Christian y Luis. Las amistades más auténticas que he tenido en mi vida. No hay palabras que puedan describir el sentimiento de familia que hay, desde pequeños, pudiendo compartir momentos y viéndonos crecer.
Gracias a Pablo y Pere, amistades que forjas en la universidad, pero sabes que se quedan para toda la vida. En especial, gracias a Pablo, por ser mi pilar fundamental y no dejar ni un instante en la etapa universitaria sin risas.
Gracias a Amparo, por ser una de las amistades más genuinas y de bienestar mutuo que he tenido en mi vida.
Por último, gracias a mi tutor Valero por guiarme en este proyecto y hacerlo posible.

Índice general

Capítulo 1

Introducción y Objetivos

1.1. Introducción

El mercado inmobiliario es un sector dinámico y complejo en el que la determinación precisa del valor de una propiedad juega un papel fundamental en las decisiones de compra, venta e inversión. En los últimos años, este mercado ha experimentado importantes fluctuaciones debido a diversos factores económicos, como el crecimiento de la demanda, la escasez de oferta en determinadas áreas y las políticas monetarias que afectan el acceso a la financiación. En este contexto, disponer de herramientas avanzadas para la estimación de precios sería de gran utilidad tanto para compradores y vendedores como para agentes inmobiliarios e inversores.

Tradicionalmente, la tasación de inmuebles ha estado basada en la experiencia de agentes del sector y en comparaciones con propiedades similares en la misma ubicación. Sin embargo, este enfoque presenta limitaciones debido a la subjetividad, la variabilidad de los criterios utilizados y a la gran cantidad de variables que deben considerarse , lo que puede derivar en discrepancias en las valoraciones y en una falta de transparencia en el mercado. La incorporación de técnicas avanzadas de análisis de datos y aprendizaje automático permite superar estas limitaciones al ofrecer modelos basados en datos objetivos y en el reconocimiento de patrones ocultos en grandes volúmenes de información.

En los últimos años, el mercado inmobiliario en Valencia ha experimentado un notable incremento en los precios de la vivienda. Según datos de Idealista, en febrero de 2025, el precio medio por metro cuadrado se situó en 1.598 euros, lo que representa un aumento del 12,7 % respecto al mismo mes del año anterior. Esta tendencia alcista refleja la creciente demanda y la limitada oferta de propiedades en la región. [1]

El mercado inmobiliario de Valencia enfrenta una notable escasez de viviendas de obra nueva, evidenciada por datos recientes que reflejan una disminución significativa en la construcción y disponibilidad de estas propiedades. En el primer semestre de 2024, el Ayuntamiento de València concedió licencias para la construcción de 1.182 viviendas de nueva planta, de las cuales solo el 11 % correspondieron a viviendas de protección pública .[2]

Esta cifra contrasta marcadamente con los niveles de construcción previos a la crisis financiera de 2008. Por ejemplo, en 2006, se construyeron aproximadamente 665.000 viviendas en España, mientras que en 2012, este número se redujo drásticamente a 34.000, evidenciando una caída del 94 % . Aunque estos datos son a nivel nacional, reflejan una

tendencia generalizada que también afecta a Valencia.[3]

La escasez de obra nueva es aún más evidente al analizar la distribución geográfica en la ciudad. En 60 de los 88 barrios de Valencia, es decir, en dos de cada tres, no existen promociones de vivienda nueva a la venta . Esta falta de oferta ha contribuido a un aumento del 80 % en el precio medio de la vivienda de obra nueva en la ciudad durante los últimos cinco años .[4][5]

En este contexto de fluctuaciones y desafíos en el mercado inmobiliario valenciano, la implementación de herramientas basadas en análisis de datos y modelos predictivos se vuelve esencial. Para los compradores, contar con estimaciones precisas del valor de mercado de una propiedad facilita decisiones informadas y evita pagar precios inflados. Los vendedores pueden establecer precios adecuados y resaltar las características distintivas de sus inmuebles. Los inversores, por su parte, se benefician al identificar áreas de alta rentabilidad y tomar decisiones estratégicas fundamentadas en datos confiables. La aplicación de estas herramientas no solo mejora la transparencia y eficiencia del mercado, sino que también contribuye a equilibrar la oferta y la demanda, promoviendo un entorno inmobiliario más justo y sostenible.

1.2. Objetivos

El presente Trabajo de Fin de Grado tiene como finalidad el desarrollo de un modelo predictivo basado en técnicas de aprendizaje automático que permita estimar, con un alto grado de precisión, el precio de venta de inmuebles en la ciudad de Valencia. Asimismo, se propone evaluar el impacto de posibles reformas en el valor final de la vivienda, con el objetivo de proporcionar una herramienta analítica que facilite la toma de decisiones fundamentadas tanto para propietarios como para potenciales inversores en el mercado inmobiliario.

Para alcanzar este propósito general, se establecen los siguientes objetivos específicos:

1. **Definir las características o atributos de los inmuebles necesarios para desarrollar un algoritmo de aprendizaje automático**. Se analizarán variables como ubicación, superficie, número de habitaciones, antigüedad, estado de conservación, eficiencia energética, accesibilidad a transporte y servicios, entre otros factores determinantes del precio de una vivienda.

2. **Obtener un conjunto de datos que cumpla con los requisitos de calidad y representatividad**. Para ello haremos uso de la API del portal inmobiliario Idealista de esta forma facilitandonos el acceso a información actualizada sobre inmuebles en Valencia.

3. **Realizar la limpieza, transformación y tratamiento de los datos** para asegurar su coherencia y calidad antes de su uso en la modelización. Se eliminarán valores nulos, duplicados y posibles errores, y se aplicarán técnicas de normalización y codificación de variables para optimizar el desempeño del algoritmo.

4. **Seleccionar el algoritmo de aprendizaje automático más adecuado** para la predicción de precios de vivienda, evaluando opciones como regresión lineal simple, regresión lineal múltiple, Random Forest Regressor, Extreme Gradient Boosting y redes neuronales.

5. **Entrenar y evaluar el rendimiento del modelo** utilizando métricas como error absoluto medio (MAE), error cuadrático medio (MSE), raíz del error cuadrático medio (RMSE), error relativo medio (MRE) y coeficiente de determinación (R^2). Se aplicará validación cruzada y optimización de hiperparámetros para mejorar la precisión del modelo.

6. **Implementar una interfaz de usuario que permita la interacción con el modelo**. Se desarrollará una herramienta en la que el usuario pueda introducir las características de un inmueble y obtener una estimación del precio de venta, así como evaluar escenarios con diferentes niveles de reforma.

7. **Ofrecer dos casos de uso y analizar la viabilidad del modelo en un contexto real**. Se aplicará la solución sobre dos casos reales comprobando así su aplicabilidad en el mercado inmobiliario de Valencia y evaluar su potencial uso en la toma de decisiones del sector.

Capítulo 2

Marco Teórico

2.1. Mercado Inmobiliario

El mercado inmobiliario es un ámbito económico que comprende la compra, venta, alquiler e inversión en propiedades inmuebles como viviendas, oficinas, locales comerciales, terrenos y edificios industriales. Este mercado presenta una gran complejidad y dinamismo debido a la interacción de múltiples factores económicos, sociales, políticos y demográficos. Entre los elementos determinantes que condicionan su comportamiento destacan la oferta y la demanda, los tipos de interés, las regulaciones gubernamentales, las tendencias demográficas y el desarrollo económico regional.

En el contexto español, el mercado inmobiliario ha experimentado tres grandes ciclos económicos en las últimas décadas. El primer ciclo, entre mediados de los años 90 y 2007, se caracterizó por un auge significativo impulsado por políticas crediticias expansivas, bajas tasas de interés y un rápido crecimiento económico, resultando en un incremento considerable de los precios inmobiliarios y un alto volumen de transacciones. Esta etapa culminó con la creación de una burbuja inmobiliaria cuya ruptura coincidió con la crisis financiera global iniciada en 2008, lo que llevó a una profunda recesión del mercado.[6]

La segunda etapa, comprendida entre 2008 y 2014, fue de crisis y ajuste severo, marcada por la contracción del crédito, el aumento del desempleo, el desplome en la demanda y una significativa caída de precios inmobiliarios. La recesión tuvo un impacto negativo considerable tanto en el sector de la construcción como en la economía en general, afectando a bancos, empresas y particulares.[7]

Desde 2014 hasta la actualidad, el mercado inmobiliario español ha mostrado signos de recuperación gradual. Se observó un aumento sostenido en el número de operaciones de compra-venta y una estabilización progresiva de precios. Factores como la mejora de las condiciones económicas generales, la recuperación del empleo y la creciente inversión extranjera, especialmente en regiones como la Comunidad Valenciana, Cataluña y Madrid, han impulsado esta fase de recuperación. Sin embargo, la pandemia del COVID-19 en 2020 provocó un breve pero significativo retroceso en esta tendencia positiva, afectando temporalmente la demanda y provocando incertidumbre en el mercado.[8]

Las perspectivas recientes, según diversos estudios económicos, prevén un crecimiento moderado y estable del mercado inmobiliario en los próximos años. Se anticipa una leve apreciación en los precios, una actividad constructora controlada y una demanda sostenida, aunque condicionada por factores macroeconómicos como la política monetaria del

Banco Central Europeo y la evolución económica general.[9]

2.1.1. Mercado Inmobiliario Valencia

La ciudad de Valencia representa un mercado inmobiliario singular debido a sus características específicas, entre las cuales destacan su atractivo turístico, su creciente relevancia económica y su posición estratégica dentro del panorama nacional e internacional. En las últimas décadas, Valencia ha atravesado diferentes fases claramente identificables: desde la rápida expansión inmobiliaria previa a la crisis de 2008, hasta el fuerte ajuste posterior y la recuperación gradual observada en los últimos años.[6, 10]

Tras la crisis económica global, el mercado inmobiliario valenciano sufrió una notable caída en la actividad, con una reducción significativa de los precios y un exceso de oferta generado por construcciones previas. Sin embargo, a partir de 2014 se inició una recuperación sostenida, impulsada principalmente por el incremento de la demanda, especialmente en zonas céntricas y turísticas, así como por un contexto económico más favorable.[11, 12]

En años recientes, Valencia ha experimentado un incremento significativo en la demanda de propiedades, impulsado por factores como el crecimiento poblacional, la afluencia de inversión extranjera, y un notable interés por el desarrollo urbanístico sostenible. Este crecimiento en la demanda se ha visto acompañado de un alza en los precios, particularmente en áreas céntricas y costeras, generando una presión creciente sobre la disponibilidad de vivienda, especialmente en el segmento de obra nueva.

La realidad inmobiliaria valenciana también refleja una notable disparidad geográfica en cuanto a disponibilidad y precios, observándose diferencias sustanciales entre barrios céntricos, periféricos y aquellos en proceso de regeneración urbana. Además, las políticas urbanísticas locales y regionales están desempeñando un papel clave en la configuración actual y futura del mercado, incentivando proyectos de rehabilitación y nuevas promociones residenciales, aunque todavía son insuficientes para cubrir plenamente la demanda existente.

En definitiva, analizar y comprender las particularidades y tendencias históricas del mercado inmobiliario valenciano resulta esencial para el desarrollo de estrategias efectivas, tanto desde la perspectiva pública como privada, destinadas a garantizar un crecimiento sostenible, equitativo y accesible del sector inmobiliario en la ciudad.[13, 14]

2.1.2. API

Una API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés Application Programming Interface) es un conjunto de definiciones, protocolos y herramientas que permiten la comunicación e interacción entre distintos sistemas de software. Las APIs actúan como intermediarias, facilitando que distintas aplicaciones y servicios intercambien información de manera eficiente y segura, sin necesidad de compartir sus detalles internos de implementación.

El uso de APIs permite a los desarrolladores acceder a funcionalidades específicas de otras aplicaciones o servicios sin tener que desarrollar esos recursos desde cero. De este modo, se logra una mayor eficiencia, reusabilidad del código y facilidad de integración. Entre los tipos más comunes de APIs se encuentran las APIs web, que utilizan protocolos como HTTP para comunicarse entre cliente y servidor.

2.1.3. API de Idealista

La API de Idealista es un servicio ofrecido por el portal inmobiliario Idealista, diseñado específicamente para facilitar la extracción y consulta automatizada de datos sobre propiedades inmobiliarias. Esta API permite acceder a información detallada sobre inmuebles en venta o alquiler, incluyendo atributos tales como precio, ubicación geográfica exacta, tamaño del inmueble, número de habitaciones y baños, características específicas del inmueble, estado de conservación, certificación energética y otros detalles relevantes para la evaluación inmobiliaria.

La API de Idealista es especialmente útil para desarrolladores, investigadores, analistas del sector inmobiliario y empresas tecnológicas que buscan realizar estudios de mercado, desarrollar aplicaciones o implementar sistemas de valoración automatizados. Al utilizar la API, los usuarios pueden realizar búsquedas específicas mediante parámetros y filtros personalizados, lo que les permite obtener conjuntos de datos actualizados y precisos para diferentes fines, tales como análisis de mercado, modelos predictivos basados en Machine Learning o sistemas avanzados de recomendación inmobiliaria.

La implementación y el uso de la API de Idealista suelen realizarse a través de solicitudes HTTP utilizando métodos estándar (GET, POST), y los datos obtenidos generalmente se entregan en formatos estructurados como JSON o XML. Esta facilidad de acceso y estructura clara facilita enormemente su integración con otras aplicaciones y sistemas de análisis.

En conclusión, el uso de la API de Idealista representa una ventaja significativa en el mercado inmobiliario, proporcionando acceso rápido y estructurado a datos fiables y actualizados, lo cual permite mejorar considerablemente la eficiencia, precisión y alcance de los análisis y herramientas desarrolladas para este sector.

2.1.4. Aplicación en el mercado Inmobiliario

La aplicación del Machine Learning (ML) en el mercado inmobiliario se ha consolidado en los últimos años como una herramienta clave para mejorar la precisión y objetividad en la tasación de propiedades. Este desarrollo responde a la complejidad inherente al mercado inmobiliario, en el que intervienen numerosos factores económicos, sociales, geográficos y físicos, que tradicionalmente se valoraban mediante métodos subjetivos, basados principalmente en la experiencia personal.

Los modelos de ML ofrecen una alternativa objetiva, precisa y escalable frente a estos métodos tradicionales. Para su desarrollo, el primer paso consiste en la recopilación de grandes volúmenes de datos sobre propiedades inmobiliarias, incluyendo características como ubicación geográfica, superficie, tipo de construcción, número de habitaciones y baños, estado de conservación, entre otros. Estos conjuntos de datos suelen extraerse de fuentes como portales inmobiliarios, registros públicos y APIs especializadas, garantizando así la calidad y actualidad de la información.

Tras la recolección de datos, se lleva a cabo un exhaustivo análisis exploratorio con el fin de identificar patrones y relaciones significativas entre las variables involucradas. Este proceso es fundamental, pues permite comprender en profundidad cómo afectan cada una de estas variables al precio final del inmueble.

En la fase de desarrollo de modelos predictivos, se emplean diversos algoritmos avanzados de machine learning, tales como la regresión lineal múltiple, bosques aleatorios

(Random Forest), Extreme Gradient Boosting (XGBoost) y redes neuronales artificiales (MLP). Estos modelos suelen ser implementados con bibliotecas y herramientas especializadas como Scikit-Learn, TensorFlow y Keras. Además, es habitual realizar un ajuste fino de hiperparámetros mediante técnicas como la validación cruzada, optimizando así el rendimiento del modelo y minimizando errores de predicción.

La validación de estos modelos predictivos se realiza mediante métricas de rendimiento ampliamente aceptadas en el ámbito científico y profesional, como el Error Cuadrático Medio (MSE), la Raíz del Error Cuadrático Medio (RMSE), y el Error Medio Absoluto (MAE). Estas métricas permiten evaluar objetivamente la precisión y fiabilidad de las predicciones realizadas por los modelos.

La implementación del Machine Learning en el mercado inmobiliario ofrece numerosas aplicaciones prácticas, tales como la identificación automatizada de oportunidades de inversión, la realización de tasaciones inmobiliarias más precisas y objetivas, y la mejora en la eficiencia de procesos de compraventa. Estos avances permiten reducir significativamente los costos y tiempos asociados tradicionalmente al mercado inmobiliario, facilitando la toma de decisiones informadas tanto para agentes inmobiliarios como para inversores y compradores particulares. Además, el desarrollo de una interfaz que interactúe con el modelo predictivo permite simular precios de inmuebles ficticios o analizar cómo varía el precio estimado al modificar ciertas características de una vivienda existente. Esta funcionalidad proporciona un recurso valioso para estudiar el impacto de atributos individuales (como añadir ascensor, reformar la cocina o disponer de terraza), facilitando así estrategias de mejora o reforma que maximicen el valor de mercado del inmueble.

En definitiva, la integración de técnicas de Machine Learning está transformando positivamente la dinámica del mercado inmobiliario, proporcionando herramientas innovadoras que mejoran la transparencia, objetividad y eficacia en la valoración y comercialización de inmuebles.

2.2. Machine Learning

El Machine Learning (aprendizaje automático) es una rama de la inteligencia artificial centrada en la creación de algoritmos capaces de aprender patrones de datos para realizar predicciones o tomar decisiones sin haber sido explícitamente programados para hacerlo. Este proceso de aprendizaje se basa en la identificación de patrones a partir de grandes volúmenes de datos y permite a las máquinas mejorar su rendimiento de manera progresiva y autónoma con cada interacción. [15]

El aprendizaje automático puede clasificarse principalmente en tres enfoques según la naturaleza de los datos y del problema que se busca resolver:

1. **Aprendizaje Supervisado:** En este enfoque, los algoritmos aprenden a partir de conjuntos de datos etiquetados, es decir, cada dato de entrada está asociado a una salida conocida o etiqueta. El objetivo del modelo es predecir o clasificar con precisión nuevas instancias no vistas previamente. Los algoritmos más comunes en esta categoría incluyen la regresión lineal y múltiple, árboles de decisión, bosques aleatorios (Random Forest), máquinas de soporte vectorial (SVM) y redes neuronales artificiales.[16]

2. **Aprendizaje No Supervisado:** Este enfoque trabaja con conjuntos de datos no

etiquetados y busca identificar patrones ocultos o estructuras inherentes en los datos. Los métodos comunes incluyen algoritmos de agrupamiento (clustering), como K-Means, así como métodos de reducción de dimensionalidad, como análisis de componentes principales (PCA).[17]

3. **Aprendizaje por Refuerzo:** Se centra en cómo un agente toma decisiones secuenciales en un entorno para maximizar una recompensa acumulada a largo plazo. Es ampliamente utilizado en escenarios donde la toma de decisiones implica un proceso continuo, como la robótica o los juegos.

El aprendizaje automático ha demostrado ser especialmente eficaz en el ámbito de la predicción de precios inmobiliarios, un campo donde la precisión y objetividad en la tasación son cruciales para la toma de decisiones de inversión y compraventa. Modelos predictivos avanzados como el Extreme Gradient Boosting (XGBoost), Random Forest son utilizados con frecuencia debido a su capacidad para gestionar conjuntos de datos grandes y complejos, así como para proporcionar predicciones altamente precisas.[18]

2.2.1. Preprocesado

El *preprocesado de datos* constituye una fase esencial en el desarrollo de cualquier proyecto de análisis predictivo, pues de ello depende la calidad y la fiabilidad del modelo final. Durante esta etapa se llevan a cabo transformaciones que preparan el conjunto de datos crudo para su posterior empleo en algoritmos de aprendizaje automático. Entre los objetivos principales del preprocesado destacan:

- **Limpieza**: detección y corrección de valores faltantes o inconsistentes.
- **Selección**: identificación y descarte de variables irrelevantes o redundantes.
- **Transformación**: ajuste de la escala de las variables y codificación de datos categóricos.
- **Tratamiento de valores atípicos**: mitigación del impacto de outliers mediante métodos como el rango intercuartílico (IQR).

Algunas de las herramientas que hemos empleado son las siguientes.

2.2.2. One-Hot Encoding

El *One-Hot Encoding* es una técnica de ingeniería de variables categóricas muy utilizada en aprendizaje automático para transformar atributos cualitativos en un formato numérico adecuado para la mayoría de los algoritmos. Cuando disponemos de una variable categórica **C** que puede tomar **k** valores distintos $\{c_1, c_2, \ldots, c_k\}$(por ejemplo, distrito de Valencia: "Ciutat Vella", "L'Eixample", "Camins al Grau", ...), no es apropiado asignarles valores ordinales (1, 2, 3, ...), pues ello implicaría un orden implícito y distancias arbitrarias entre categorías.

El One-Hot Encoding crea k nuevas variables binarias $\{z_1, z_2, \ldots, z_k\}$, de modo que para cada observación:

$$z_j = \begin{cases} 1, & \text{si } C = c_j, \\ 0, & \text{en caso contrario.} \end{cases}$$

Por ejemplo, para la variable "distrito" con las categorías (Ciutat Vella,L'Eixample,Algirós), una vivienda en "L'Eixample" se representaría como (0, 1, 0)

Este procedimiento asegura:

1. **Ausencia de ordenación artificial**: no se asume que una categoría "vale más" que otra.
2. **Separación completa**: cada categoría queda representada por una dimensión propia, lo que facilita al modelo diferenciar entre todas ellas.

2.2.3. Filtro por rango intercuartílico (IQR)

El método del *Filtro por Rango Intercuartílico* (IQR, *Interquartile Range*) es una técnica estadística robusta para la detección y eliminación de valores atípicos (*outliers*) en variables continuas. Está basado en medidas de posición (los cuartiles) que no se ven afectadas por valores extremos, lo que lo convierte en un procedimiento preferido en preprocesado de datos inmobiliarios, donde precios o superficies muy alejados de la distribución típica pueden distorsionar el ajuste de los modelos predictivos.

Dada una muestra ordenada de tamaño n, los cuartiles se definen como:

- **Primer cuartil** Q1: valor tal que el 25 % de los datos son menores o iguales.
- **Tercer cuartil** Q3: valor tal que el 75 % de los datos son menores o iguales.

El **Rango Intercuartílico** se calcula como:

$$\text{IQR} = Q_3 - Q_1$$

Este IQR mide la dispersión central de la variable, y al estar basado en percentiles, es insensible a valores extremos.

Criterios para identificar outliers

Los límites para considerar una observación x como atípica se definen convencionalmente como:

$$\text{Límite inferior} = Q_1 - 1,5 \times \text{IQR}$$
$$\text{Límite superior} = Q_3 + 1,5 \times \text{IQR}$$

Cualquier $x < Q_1 - 1,5\,\text{IQR}$ o $x > Q_3 + 1,5\,\text{IQR}$ se clasifica como outlier.

2.2.4. Algoritmos implementados en el proyecto

2.2.5. Modelo de Regresión Lineal

El **modelo de regresión lineal** es una técnica estadística y de aprendizaje automático destinada a describir o predecir una variable de interés continua Y en función de una o

varias variables explicativas X_j. Se basa en la hipótesis de que la relación entre la(s) variable(s) independiente(s) y la dependiente es de tipo lineal. [19, 20].

Cuando existe un único predictor X, hablamos de **regresión lineal simple**. Su formulación matemática es:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

donde:

- β_0 es el *término independiente* o intercepto, que representa el valor esperado de Y cuando X=0.
- β_1 es la *pendiente* del modelo, que indica el cambio promedio en Y por unidad de variación en X.
- ε_i es el término de error o residuo para la i-ésima observación, asumido con $E[\varepsilon_i] = 0, \quad \mathrm{Var}(\varepsilon_i) = \sigma^2$

Para estimar β_0 y β_1 se emplea el **método de mínimos cuadrados ordinarios (MCO)**, que minimiza la suma de los cuadrados de los residuos:

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} \big(y_i - (\beta_0 + \beta_1 x_i)\big)^2$$

Las soluciones analíticas son:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \, \bar{x}$$

Cuando disponemos de p variables predictoras ($X_1, X_2, \ldots, X_p$), el modelo se extiende a la **regresión lineal múltiple**:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \varepsilon_i.$$

cuya estimación matricial es:

$$\hat{\boldsymbol{\beta}} \;=\; \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \, \mathbf{Y}.$$

Para que las estimaciones sean válidas e interpretables, **la regresión lineal requiere cumplir varias condiciones sobre el término de error** ε:

1. **Linealidad**: la relación entre cada X_j y Y es lineal.
2. **Independencia**: los errores ε_i son mutuamente independientes.

3. **Homoscedasticidad**: $\text{Var}(\varepsilon_i)$ es constante $\forall i$.
4. **Normalidad**: $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$
5. **Ausencia de multicolinealidad**: no existen correlaciones lineales exactas entre las variables X_j

Ventajas

- Fácil interpretación de los coeficientes: cada β_j mide el efecto marginal de X_j sobre Y.
- Algoritmo computacionalmente eficiente incluso con grandes volúmenes de datos.
- Base para métodos más complejos (p. ej. regresión regularizada, modelos generalizados).

Limitaciones

- Solo capta relaciones lineales; no modela interacciones o curvaturas sin extensiones previas.
- Sensible a valores atípicos y a la presencia de variables irrelevantes.
- Requiere verificar los supuestos estadísticos; violaciones pueden sesgar o ineficiencias en las estimaciones.

2.2.6. Random Forest Regressor

Random Forest (RF) es un método de ensamble propuesto por Leo Breiman (2001) que combina múltiples árboles de decisión entrenados sobre subconjuntos aleatorios de datos para tareas de clasificación y regresión . [21]

Se basa en el principio de **bagging**, donde se generan múltiples muestras bootstrap del conjunto de datos y se entrena un árbol de decisión en cada una. A esto se suma la selección aleatoria de un subconjunto de variables en cada división del árbol, lo que favorece la diversidad entre árboles y reduce la varianza del modelo. [22]

La predicción final del modelo se obtiene promediando (en regresión) como es nuestro caso o votando (en clasificación) los resultados de los árboles individuales:

$$\hat{f}_{\text{RF}}(x) \;=\; \frac{1}{B}\sum_{b=1}^{B} f_b(x).$$

Una característica destacada del Random Forest es la estimación del **error Out-Of-Bag (OOB)**, que permite validar internamente el modelo sin requerir un conjunto de prueba adicional.

También ofrece medidas de **importancia de variables**, útiles en análisis exploratorio y selección de características, basadas en el incremento del error OOB al alterar cada predictor.

Ventajas

- **Robustez ante el sobreajuste**: la agregación de muchos árboles decorrelacionados mitiga el exceso de adaptarse al ruido del conjunto de entrenamiento .
- **Manejo de datos mixtos**: puede procesar simultáneamente variables numéricas y categóricas sin transformaciones complejas.
- **Estimación interna de error**: el error OOB evita la necesidad de particiones adicionales de validación.
- **Interpretabilidad parcial**: ofrece rangos de importancia de variables.

Limitaciones

- **Complejidad computacional**: el entrenamiento y la predicción pueden resultar costosos en memoria y tiempo cuando B y p son grandes.
- **Predicciones suaves**: al promediar, tiende a no producir valores extremos fuera del rango observado en el entrenamiento.
- **Poca transparencia individual**: el bosque carece de la simplicidad interpretativa de un único árbol de decisión.

2.2.7. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting es una implementación optimizada del algoritmo de **Gradient Boosting**, desarrollada por Chen y Guestrin (2016), que destaca por su velocidad, precisión y capacidad de regularización [23][24]. Pertenece a los métodos de **boosting**, que construyen modelos secuenciales donde cada nuevo árbol mejora los errores del conjunto anterior.

El modelo final se construye como una suma ponderada de árboles:

$$\hat{y}(x) \;=\; \sum_{m=1}^{M} \eta\, f_m(x).$$

donde η es la tasa de aprendizaje, que controla el impacto de cada árbol y reduce el riesgo de sobreajuste.

XGBoost optimiza una función de pérdida mediante **descenso de gradiente**, ajustando cada nuevo árbol en la dirección opuesta al gradiente del error. Incorpora **regularización L2 y L1** para controlar la complejidad del modelo y fomentar la selección de variables relevantes:

$$\mathcal{L} \;=\; \sum_{i=1}^{n} L\big(y_i, \hat{y}_i\big) \;+\; \lambda \sum_{j=1}^{T} w_j^2 \;+\; \gamma\, T.$$

donde λ penaliza los pesos de las hojas y γ penaliza el número de nodos terminales T, promoviendo árboles más simples.

Optimización y eficiencia

- **Submuestreo**: XGBoost puede muestrear aleatoriamente tanto filas como columnas en cada iteración, lo que reduce la correlación entre árboles y el riesgo de sobreajuste.

- **Paralelización**: aprovecha múltiples núcleos para construir diferentes ramas del árbol simultáneamente.
- **Manejo de valores faltantes**: asigna automáticamente las observaciones con datos ausentes a la rama que maximiza el beneficio.

Ventajas

- Regularización integrada que mitiga el sobreajuste.
- Alta eficiencia computacional y capacidad para manejar datos mixtos y faltantes.
- Control fino de la complejidad mediante hiperparámetros (η, λ, γ, profundidad máxima, etc.).

Limitaciones

- Requiere sintonía cuidadosa de múltiples hiperparámetros.
- Menor interpretabilidad que modelos lineales o un solo árbol de decisión.
- Consumo de recursos elevado si se usan muchos árboles o gran profundidad.

2.2.8. Multi Layer Perceptron (MLP)

El **Multi Layer Perceptron** es una red neuronal de avance directo *(feedforward)* que extiende el perceptrón simple mediante una o varias capas ocultas, permitiendo modelar relaciones no lineales entre variables. Su estructura en capas y el uso de funciones de activación lo hacen útil para problemas de regresión y clasificación en contextos complejos .[25]

Componentes principales:

- **Neuronas**: combinan entradas ponderadas y un sesgo, aplicando una función de activación (ReLU, sigmoide, etc.).
- **Capas**: organizadas en entrada, ocultas y salida, estas últimas encargadas de producir la predicción final.
- **Parámetros**: pesos y sesgos ajustados durante el entrenamiento para minimizar el error.

Entrenamiento:

1. **Propagación hacia adelante**: las entradas atraviesan la red capa a capa.
2. **Cálculo de pérdida**: se evalúa el error (p. ej. MSE o entropía cruzada).
3. **Retropropagación**: el error se propaga hacia atrás y se ajustan los parámetros mediante gradiente descendente.
4. **Optimización**: se emplean algoritmos como SGD, Adam o RMSprop.

Regularización:

Para evitar el sobreajuste, se emplean técnicas como:

- **L2**: penaliza pesos grandes.
- **Dropout**: desconecta neuronas aleatoriamente en el entrenamiento.
- **Early stopping**: detiene el proceso cuando la validación deja de mejorar.
- **Batch normalization**: normaliza las activaciones intermedias, estabilizando el entrenamiento .[26]

Ventajas:

- Alta versatilidad en distintos tipos de datos.
- Capacidad universal de aproximación de funciones continuas.

Limitaciones:

- Requiere cuidado en la elección de arquitectura e hiperparámetros.
- Poca interpretabilidad comparado con modelos más simples.

2.2.9. GridSearchCV

GridSearchCV es una herramienta de Scikit-learn que automatiza la búsqueda de la mejor combinación de hiperparámetros para un modelo, en nuestro caso aplicada tanto a Random Forest como a XGBoost. Su funcionamiento básico consiste en:

1. Recibir un diccionario de hiperparámetros con las posibles opciones para cada parámetro.
2. Dividir el conjunto de entrenamiento en k particiones (k-fold) y, para cada combinación, entrenar k veces el modelo, usando $k-1$ pliegues para ajustar y el pliegue restante para validar .[27]
3. Calcular la métrica media (p. ej. MAE o R^2) sobre las k validaciones y seleccionar la configuración que optimiza dicho indicador.
4. Reentrenar el modelo completo con esos valores óptimos.

Este procedimiento garantiza una calibración sistemática y robusta de los parámetros, reduciendo el riesgo de sobreajuste y evitando depender de una única partición de datos .[28]

2.2.10. RandomizedSearchCV

Para la optimización de los hiperparámetros del **Multilayer Perceptron**, caracterizado por disponer de un espacio de búsqueda muy amplio (arquitectura de capas, función de activación, algoritmo de optimización, regularización, tasa de aprendizaje, etc.), se empleó **RandomizedSearchCV** de Scikit-learn en lugar de una búsqueda exhaustiva.

RandomizedSearchCV realiza una **búsqueda aleatoria** sobre las posibles combinaciones de parámetros: en lugar de probar todas las configuraciones (como hace GridSearchCV), selecciona al azar un número fijado de muestras del espacio definido. Para cada muestra, ejecuta validación cruzada k-fold y calcula la métrica media (por ejemplo, MAE o R^2) sobre los k pliegues, devolviendo al final la configuración que maximiza el desempeño. Este enfoque reduce drásticamente el tiempo de cómputo, especialmente cuando el número de parámetros y valores posibles es elevado, sin sacrificar de forma significativa la calidad del ajuste . [29]

Utilizar RandomizedSearchCV con un número limitado de iteraciones y validación cruzada, consigue:

- **Acotar el tiempo de entrenamiento** de la red neuronal, evitando ejecuciones prohibitivas en la malla completa de parámetros.
- **Mantener robustez** en la selección de hiperparámetros al explorar de forma representativa el espacio de búsqueda.
- **Mejorar la escalabilidad** del proceso de ajuste, haciendo viable la experimentación iterativa con distintas configuraciones de red.

2.2.11. Métricas

A continuación se definen las métricas más empleadas para cuantificar la precisión y capacidad explicativa de un modelo de regresión en precios inmobiliarios.

2.2.12. MEAN SQUARED ERROR (MSE)

Esta métrica calcula el promedio de los cuadrados de las diferencias entre las predicciones $\hat{y}_i$ y los valores reales y_i. Al elevar al cuadrado cada desviación, los errores de gran magnitud reciben un peso mucho mayor que los pequeños, con lo que se penalizan fuertemente las predicciones muy alejadas de la realidad.

$$\text{MSE} \;=\; \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

Interpretación: se mide en unidades al cuadrado (por ejemplo, euros^2), lo que dificulta su interpretación directa. Ventaja: penaliza más fuertemente los errores grandes, útil cuando queremos evitar predicciones muy alejadas de la realidad. Desventaja: sensible a outliers, puede estar dominado por pocos errores muy elevados.

2.2.13. ROOT MEAN SQUARED ERROR (RMSE)

Es la raíz cuadrada del MSE, de modo que vuelve a expresarse en las mismas unidades de la variable objetivo. Gracias a ello, facilita la interpretación directa del error medio "ajustado" por la penalización de las inexactitudes más graves.

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

Interpretación: mide el error promedio en las mismas unidades de y, pero con impacto aumentado de los desvíos grandes.

2.2.14. MEAN ABSOLUTE ERROR (MAE)

Calcula la media de los valores absolutos de las diferencias $|y_i - \hat{y}_i|$. Dado que no se elevan al cuadrado, todos los errores contribuyen de forma lineal y por igual, lo que hace que esta métrica sea menos susceptible a magnificar los outliers en comparación con el MSE.

$$\text{MAE} \;=\; \frac{1}{n}\sum_{i=1}^{n}\big|y_i - \hat{y}_i\big|$$

Interpretación: expresa el error medio en las mismas unidades de la variable de interés (por ejemplo, euros). Ventaja: es robusto frente a valores atípicos moderados (no penaliza excesivamente los grandes desvíos). Desventaja: no distingue bien la dispersión de errores grandes frente a pequeños.

2.2.15. ERROR RELATIVO MEDIO (MRE)

El **Error Relativo Medio (MRE)** es una métrica utilizada para evaluar modelos de regresión. Mide el sesgo promedio de las predicciones en relación con los valores reales, expresado de forma adimensional, lo que permite comparar modelos en contextos con diferentes escalas [30].

Dado un conjunto de n observaciones $(y_i,\ \hat{y}_i)$, donde y_i es el valor real y $\hat{y}_i$ la predicción del modelo. Se define el MRE como:

$$\text{MRE} \;=\; \frac{1}{n}\sum_{i=1}^{n} d_i \;=\; \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}_i}{y_i}.$$

- **Signo del MRE**
  - $MRE > 0$: el modelo tiende a **subestimar** los valores reales en promedio.
  - $MRE < 0$: existe un sesgo de **sobreestimación**.
- **Magnitud del MRE**
  - Un MRE de 0,10 indica que, en promedio, las predicciones son un 10 % menores que los valores reales.
  - Al ser adimensional, permite comparar el desempeño entre distintos segmentos del mercado (por ejemplo, viviendas de precios altos vs. bajos) .

Ventaja: es invariante a la escala, lo que permite evaluar modelos en propiedades con rangos de precio muy distintos. Aporta sesgo direccional, ya que el signo revela si el modelo tiende a sobreestimar o subestimar de forma global. Desventaja: la cancelación de errores puede ocultar desviaciones significativas si las sobreestimaciones compensan las subestimaciones. Puede verse dominado por valores extremos, pues observaciones con precios muy bajos influyen de manera desproporcionada en el cómputo.

2.2.16. R-SQUARED (R^2)

El coeficiente de determinación mide la fracción de la variabilidad total de la variable dependiente que explica el modelo. Un R^2 cercano a 1 indica que el modelo captura casi toda la dispersión de los datos; valores próximos a 0 (o negativos) señalan un ajuste pobre o peor que predecir siempre la media.

$$R^2 \;=\; 1 \;-\; \frac{\displaystyle\sum_{i=1}^{n}\bigl(y_i - \hat{y}_i\bigr)^2}{\displaystyle\sum_{i=1}^{n}\bigl(y_i - \bar{y}\bigr)^2} \;=\; \frac{\mathrm{SS}_{\mathrm{tot}} - \mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}$$

Interpretación:

- $R^2 = 1$ indica ajuste perfecto.
- $R^2 = 0$ equivale a predecir siempre el valor medio $\bar{y}$.
- $R^2 < 0$ sugiere que el modelo es peor que la constante $\bar{y}$.

2.3. Lenguajes de programación

Para la implementación de los modelos predictivos y el resto del pipeline de análisis de datos, se ha empleado **Python**, un lenguaje de programación de alto nivel, interpretado y multiparadigma, que destaca por su sintaxis clara y legibilidad. Diseñado originalmente por Guido van Rossum en 1991, Python integra características de programación imperativa, orientada a objetos y, en menor medida, funcional, lo que facilita la construcción de prototipos rápidos y el desarrollo de soluciones complejas con un menor coste de mantenimiento[31]. Gracias a su extensa colección de librerías —como NumPy para cálculo numérico, pandas para manipulación de datos, Scikit-learn para aprendizaje automático y XGBoost para boosting de gradiente—, Python se ha consolidado como uno de los entornos favoritos en la comunidad de ciencia de datos y aprendizaje automático.

El entorno de programación utilizado ha sido **Google Colab**, una plataforma basada en cuadernos de Jupyter que Google ofrece de forma gratuita. Google Colab proporciona un entorno interactivo alojado en la nube, donde el usuario puede escribir y ejecutar código Python, visualizar gráficos y compartir fácilmente resultados con otros colaboradores. Entre sus principales ventajas se encuentran el acceso gratuito a GPU y TPU, la integración directa con Google Drive para almacenamiento de archivos, y la posibilidad de instalar paquetes adicionales mediante simples comandos de terminal [32]. Esto facilita la reproducibilidad de los experimentos y reduce las barreras de configuración del entorno local, permitiendo centrar el esfuerzo en el desarrollo y la validación de los modelos.

2.3.1. Librerías

En este proyecto se ha aprovechado la versatilidad de Python para cubrir todas las fases, desde el procesamiento de datos hasta la puesta en marcha de la aplicación. Para la manipulación y limpieza de la información se ha contado con **NumPy** y **pandas**, que permiten gestionar arrays y DataFrames de forma eficiente, mientras que **Matplotlib** y **Seaborn** se han empleado para generar las visualizaciones exploratorias que facilitan la comprensión de patrones y distribuciones. En paralelo, librerías como **Requests**, **json** y **base64** han servido para manejar comunicaciones y codificar datos multimedia, y módulos como **re**, **unidecode** y **locale** han ayudado a normalizar textos y formatear salidas numéricas según convenciones locales.

En la fase de modelado, **Scikit-learn** ha actuado como columna vertebral, ofreciendo regresión lineal, bosques aleatorios, redes MLP y utilidades para particionar datos, escalar variables y calcular métricas (MAE, MSE, RMSE, R^2), así como para ajustar hiperparámetros con **GridSearchCV** y **RandomizedSearchCV** y evaluar la relevancia de cada característica mediante . A este ecosistema se sumó **XGBoost** para aprovechar su potente algoritmo de boosting de gradiente y **SciPy** para definir espacios de búsqueda en la optimización aleatoria. Finalmente, **Joblib** se encargó de la serialización de los modelos y **Dash**, junto con **Dash Bootstrap Components**, permitió construir la interfaz web interactiva que el usuario emplea para introducir datos y visualizar predicciones al instante.

Capítulo 3

Materiales y métodos

En este capítulo se describen con detalle los recursos utilizados y el proceso seguido para la construcción del modelo de predicción de precios inmobiliarios. En primer lugar, se aborda la obtención del conjunto de datos a través de la API de Idealista, incluyendo la configuración de los filtros de búsqueda, la autenticación y el almacenamiento eficiente de los resultados. A continuación, se detalla el proceso de preprocesamiento de los datos, incluyendo limpieza, transformación y codificación de variables, así como la generación de nuevas características mediante técnicas de procesamiento de lenguaje natural. Posteriormente, se presenta un análisis exploratorio de los datos con visualizaciones clave, seguido de estudios preliminares con relaciones entre variables. Finalmente, se introducen los modelos predictivos empleados, la metodología de validación y evaluación para la selección del modelo ganador, el desarrollo de una interfaz gráfica para la interacción con el usuario, y un análisis de la importancia de las variables en la predicción. Todo este proceso constituye la base técnica sobre la que se sustenta el estudio.

3.1. Recopilación de datos

Para la recopilación de los datos utilizados en este estudio, se ha optado por la extracción de información directamente desde **Idealista**, una de las principales plataformas inmobiliarias en España, mediante el uso de su **API oficial**. Este procedimiento ha permitido obtener un conjunto de datos representativo del mercado inmobiliario en la ciudad de Valencia, asegurando la fiabilidad y actualidad de la información. **El proceso de obtención de datos ha seguido las siguientes fases:**

3.1.1. Solicitud de acceso a la API de Idealista

Para utilizar la API de Idealista, es necesario solicitar acceso a través de su plataforma para desarrolladores. En esta solicitud se ha especificado el propósito del estudio, describiendo su carácter académico y su enfoque en la predicción del precio de venta de viviendas en Valencia. Una vez concedido el acceso, se han obtenido las credenciales necesarias (API Key y Secret) para la autenticación en el sistema.

3.1.2. Autenticación y generación de token de acceso

Dado que la API de Idealista implementa un sistema de autenticación basado en OAuth 2.0, el primer paso en la extracción de datos ha sido la obtención de un token de acceso temporal. Para ello, se ha utilizado un script en Python, que envía una solicitud POST a los servidores de Idealista incluyendo las credenciales (API Key y Secret). En respuesta, se obtiene un token de acceso con una validez limitada, el cual es necesario para realizar consultas a la API.

3.1.3. Definición de la URL de búsqueda y establecimiento de filtros

Con el token de acceso activo, se ha definido la estructura de la consulta a la API, estableciendo los criterios de búsqueda para acotar los resultados a las características de interés. Entre los filtros aplicados se incluyen:

country	es
language	es
max items	50
operation	sale
property type	homes
center	39.46993,-0.37359
distance	4000
sort	desc

Partimos de una URL base general para cualquier búsqueda en la API de Idealista *"https://api.idealista.com/3.5/"* a esta le tenemos que añadir nuestros filtros de interés especificados anteriormente. Entre los parámetros elegidos para las solicitudes incluyen *'center', 'distance'*y *'maxitems'* los cuales vamos a desarrollar para una mayor claridad.

- Para especificar la búsqueda en el centro de Valencia lo hacemos mediante los parámetros *center* y *distance*. Seleccionamos los valores especificados ya que queremos un punto céntrico y un area alrededor que no tomase viviendas de pueblos cercanos como Quart, Burjasot Tabernes...

- En cuanto a *MaxItems* es el límite de 50 elementos por solicitud establecido por Idealista, se realizaron múltiples llamadas variando el número de página para obtener todos los resultados

3.1.4. Extracción de los datos en formato JSON

Una vez configurada la URL de búsqueda con los filtros adecuados, se han realizado las primeras consultas a la API, obteniendo los resultados en **formato JSON**. Estos datos contienen información detallada de los inmuebles disponibles en la plataforma, incluyendo precios, descripciones, ubicación geográfica, características estructurales y otros atributos de interés.

3.1.5. Almacenamiento y estructuración de los datos

Para facilitar el tratamiento y análisis posterior, los datos extraídos de cada consulta han sido convertidos en un DataFrame de Pandas, permitiendo una manipulación eficiente en Python. Dado que la API de Idealista devuelve los resultados en páginas limitadas (paginación), se ha implementado un proceso iterativo que recorre todas las páginas disponibles, almacenando progresivamente la información en el DataFrame hasta completar la extracción de la totalidad de registros relevantes. Por último, almacenamos este DataFrame en CSV debido a que cada página que estemos estudiando es una llamada a la API y tenemos llamadas limitadas al mes; de esta forma, sólo tendremos que hacer las llamadas iniciales.

Este proceso ha permitido la obtención de un conjunto de datos actualizado y representativo del mercado inmobiliario en el centro de Valencia, que servirá como base para el desarrollo del modelo de predicción de precios. Además, se han aplicado técnicas de limpieza y depuración de datos para garantizar su calidad antes de proceder con la fase de análisis y modelización.

3.2. Preprocesado de datos

Una vez recopilado el conjunto de datos a través de la API de Idealista, se llevó a cabo un exhaustivo trabajo de preprocesamiento con el objetivo de garantizar la calidad, consistencia y adecuación de la información para su uso en modelos de aprendizaje automático.

3.2.1. Eliminación de duplicidades

La primera acción consistió en identificar y eliminar entradas duplicadas dentro del conjunto de datos, dado que algunas propiedades pueden aparecer en varias ocasiones por estar promocionadas por diferentes inmobiliarias o estar listadas múltiples veces debido al proceso de carga. Esta limpieza fue esencial para evitar que los modelos aprendieran patrones redundantes o sesgados. La identificación de duplicados se basó en combinaciones de atributos únicos como longitud, latitud, la superficie, el distrito... Para los duplicados por error de carga, fue suficiente con eliminar los inmuebles que tuvieran repetida la variable *'propertyCode'* que sería como una especie de ID para los inmuebles.

3.2.2. Selección de variables explicativas

A partir del conjunto completo de variables disponibles, se realizó una selección de aquellas que, desde una perspectiva teórica y empírica, se consideran más influyentes en la determinación del precio de una vivienda. Entre estas variables se encuentran:

Variables más importantes del dataset con su tipo y descripción

Variable	Tipo de variable	Descripción
propertyCode	Numérica	Indicador único de la propiedad
numPhotos	Numérica	Número total de fotografías que contiene el anuncio
floor	Categórica	Planta en la que se encuentra la vivienda
price	Numérica	Precio de venta
propertyType	Categórica	Tipo de vivienda (Piso, penthouse, duplex, studio)
size	Numérica	Tamaño de la vivienda en m2
exterior	Boolean	Vivienda exterior
rooms	Numérica	Número de habitaciones
bathrooms	Numérica	Número de baños
address	Categórica	Dirección
district	Categórica	Distrito
neighborhood	Categórica	Vecindario
latitude	Numérica	Coordenadas de latitud
longitude	Numérica	Coordenadas de longitud
description	Categórica	Descripción de la vivienda
hasVideo	Boolean	Disponibilidad de vídeo del inmueble
status	Categórica	Estado de la vivienda
newDevelopment	Boolean	Indicador de si se trata de obra nueva
hasLift	Categórica	Disponibilidad de ascensor
parkingSpace	Boolean	Disponibilidad de parking
priceByArea	Numérica	Precio por m2
hasPlan	Boolean	Disponibilidad de plano del inmueble
has3DTour	Boolean	Disponibilidad de recorrido virtual en 3D
has360	Boolean	Disponibilidad de vista 360º
hasStaging	Boolean	Disponibilidad de decoración simulado
newDevelopmentFinished	Boolean	Indicador de obra nueva finalizada

3.2.3. Tratamiento de valores ausentes

Decidimos que no vamos a eliminar los registros que contengan valores ausentes ya que, si no, nos quedaríamos con una muestra muy pequeña. Lo que hacemos es que las variables que contenían valores nulos o ausentes fueron analizadas individualmente y, en los casos en los que la ausencia podía interpretarse como falta de información proporcionada por el anunciante (por ejemplo, *'hasLift'* o *'exterior'*), se optó por asignar un valor **"Desconocido (DK)"** y, para variables que posteriormente íbamos a dummificar (por ejemplo, *'district'*o *'status'*), les poníamos otro valor **"Desconocido (DP)"** para facilitar el proceso. Este enfoque permite al modelo considerar estos casos como una categoría distinta en lugar de eliminar registros o imputar de forma arbitraria.

3.2.4. Unificación y transformación de variables relacionadas con el aparcamiento

Una de las variables explicativas seleccionadas es *'ParkingSpace'*, esta tiene un formato original de diccionario en el que encontramos *'hasParkingSpace'*y *'isParkingSpaceIncludedInPrice'*así que se realizó una transformación para consolidar la información dispersa en varias variables. Si la variable *'hasParkingSpace'* no estaba incluida en el precio (*'isParkingSpaceIncludedInPrice'* era *False*) obteníamos una nueva variable *'parkingSpacePrice'*. Así nos quedamos con 3 variables relacionadas con Parking:

- **HasParking**: indica si el inmueble dispone de plaza de aparcamiento.
- **ParkingIncluded**: señala si el aparcamiento está incluido en el precio de venta.
- **parkingSpacePrice**: refleja el coste adicional del aparcamiento si no está incluido.

3.2.5. Extracción de nuevas variables a partir de descripciones textuales mediante lenguaje natural

En muchas bases de datos inmobiliarias, las descripciones textuales proporcionadas por los agentes o los propietarios contienen información cualitativa relevante que no se encuentra en los campos estructurados. Con el fin de enriquecer el conjunto de variables disponibles, se propuso aplicar técnicas de procesamiento de lenguaje natural (PLN) sobre el campo de descripción de los inmuebles. A través de modelos de lenguaje (Language Models), se pretende identificar la presencia de términos clave que pueden reflejar características adicionales no estructuradas, como 'amueblado', 'ocupado', 'piscina'o 'trastero'entre otras. Estas variables extraídas se integrarán posteriormente en el modelo como nuevas variables binarias. Por tanto, el objetivo de este apartado es describir el proceso seguido para procesar y transformar las descripciones textuales en variables categóricas útiles para el análisis.

Limpieza y preprocesamiento del texto

1. **Normalización de mayúsculas/minúsculas**: Convertimos todo el texto a minúsculas para evitar duplicidades inducidas por el uso de mayúsculas.
2. **Eliminación de signos de puntuación y caracteres especiales**: Se eliminaron comas, puntos, paréntesis y otros caracteres no alfanuméricos, manteniendo únicamente letras y dígitos.

De esta manera, se garantiza que posteriores búsquedas de patrones sean consistentes y no dependan de diferencias ortográficas o de formato.

Extracción de variables a partir de palabras clave

Para capturar la presencia o ausencia de ciertos atributos en la descripción, definimos listas de palabras clave agrupadas por temática:

- **Contrato de alquiler**: Para saber si un inmueble tiene contrato de alquiler emplearemos palabras clave como: {contrato en vigor, alquilado, renta, arrendado, inquilino con contrato...}
- **Ocupación**: Para saber si un inmueble está ocupado emplearemos: {ocupado, inquilino sin contrato, inquilino moroso, renta activa...}
- **Piso turístico**: Para conocer si una construcción cuenta con licencia turística: {licencia turística, vivienda turística, alquiler vacacional, uso turístico, airbnb...}
- **Zonas comunes**: Para verificar si una propiedad cuenta con zonas comunes: {gimnasio, salón comunitario, club social, zona infantil, áreas recreativas, zonas verdes,,,}
- **Accesibilidad**: Para comprobar si una vivienda tiene buena accesibilidad: {ascensor adaptado, sin barreras arquitectónicas, adaptado para discapacitados, accesible para personas con movilidad reducida,,,}
- **Amueblado**: Para determinar si una propiedad está amueblada: {amueblado, equipado con mobiliario, muebles incluidos,,,}

Para cada grupo de palabras clave se creó una variable binaria (dummy) que toma valor 1 si al menos una de las palabras clave aparece en la descripción y 0 en caso contrario.

Compilación de patrones

Para mejorar el rendimiento del proceso de extracción, todas las expresiones regulares definidas se compilan una sola vez antes de iterar sobre las descripciones. Estos patrones precompilados incluyen marcadores que delimitan palabras completas y funcionan en modo insensible a mayúsculas, de modo que las búsquedas sean tanto eficaces como exhaustivas.

Extracción y expansión de variables

Se implementa una función que, para cada descripción, invoca el pre-procesado y luego evalúa cada patrón compilado contra el texto limpio. El resultado es un diccionario con valores 0 o 1 según la ausencia o presencia de cada característica. Finalmente, este diccionario se transforma en un DataFrame con columnas adicionales que se concatenan al conjunto de datos original, enriqueciendo el dataset con indicadores binarios extraídos del texto.

3.2.6. Codificación de variables booleanas

Las variables booleanas, que originalmente toman valores de verdadero o falso (como ascensor, piscina o trastero), fueron transformadas a formato numérico (1 para verdadero, 0 para falso). En los casos en los que no se disponía de información clara, se asignó el valor 3 (código para "DK") con el objetivo de que el modelo pueda tratar de forma diferenciada los casos con información incompleta, permitiéndole identificar posibles patrones ocultos en esa incertidumbre.

3.2.7. Codificación de variables categóricas

En los modelos de predicción, las variables categóricas deben transformarse en un formato numérico que el modelo pueda interpretar. La codificación one-hot (dummificación) es la técnica estándar para convertir cada categoría de una variable en una nueva variable binaria.

Selección de variables categóricas

Se identificaron dentro del conjunto de datos las variables con tipo *'object'* o categorías discretas relevantes, tales como:

- **Tipo de vivienda**: (p. ej., 'Piso', 'Duplex', 'Chalet')
- **Distrito**: nombre del distrito de la vivienda
- **Estado de conservación**: ('Bueno', 'Reformado', etc.)

Entre estas variables con tipo *'object'* nos encontramos con la variable *'Floor'*, esta no debería ser categórica, entonces tendremos que hacer un estudio de por qué no es considerada como tipo *'int'* como nos cabría imaginar.

Tratamiento de la variable Floor

1.- Identificación de valores no numéricos

Al inspeccionar los valores únicos de *floor*, se encontraron registros con etiquetas especiales: **'bj'**, **'ss'**, **'en'** y **'DP'**, que representan:

- **bj**: Planta baja (se asigna el valor **0**)
- **en**: Entresuelo (se asigna **0.5**)
- **ss**: Subsuelo (se asigna **-0.5**)
- **DP**: Dato pendiente

2.- Tratamiento de los datos faltantes (DP)

Los registros marcados como `DP` no ofrecían información sobre la planta en la que se encuentra el inmueble. Para no perder estos datos y mantener la consistencia del conjunto, se sustituyó el valor `DP` por la media de todas las observaciones numéricas válidas de la variable *floor*. De este modo, se minimiza el impacto de la ausencia de datos sin introducir sesgos extremos.

Adicionalmente, se creó una nueva variable binaria auxiliar llamada *isfloorDP*, que toma valor `1` si el valor original era DP y fue sustituido por la media, o `0` si se trata de un valor original del conjunto de datos. Esta variable permite al modelo identificar y tratar de forma diferenciada los registros con imputación, aportando información adicional sobre la fiabilidad de cada observación.

3.- Corrección de errores tipográficos

Se detectó un valor **'-2'** que, tras investigar el registro, correspondía en realidad a un segundo piso y no a un subsuelo. Se corrigió reemplazando '-2'por '2'.

4.- Conversión a formato numérico

Una vez realizadas las sustituciones, la columna se convirtió a tipo numérico *'float'*, permitiendo operaciones matemáticas y su tratamiento en el pipeline.

Procedimiento de codificación

Para cada variable categórica seleccionada se aplicó el siguiente flujo:

1. Verificación de valores únicos: en primer lugar, se identificaron las distintas categorías presentes en la columna (por ejemplo, ejecutando una inspección de los valores únicos) para conocer los estados o niveles posibles

2. Crea un DataFrame de variables dummy: Se crea un nuevo dataframe en el qe para cada categoría, se generó una nueva columna que asocia el prefijo de la variable original (por ejemplo, *'status'*) con el nombre de la categoría. Cada una de estas columnas toma valor *True* si la observación pertenece a esa categoría y *False* en caso contrario.

3. Se concatenan estas nuevas columnas al DataFrame original: Una vez generadas todas las columnas binarias, se añadieron al conjunto de datos original mediante concatenación de columnas, de manera que cada registro contiene explícitamente información sobre a qué categoría pertenece y a cuáles no.

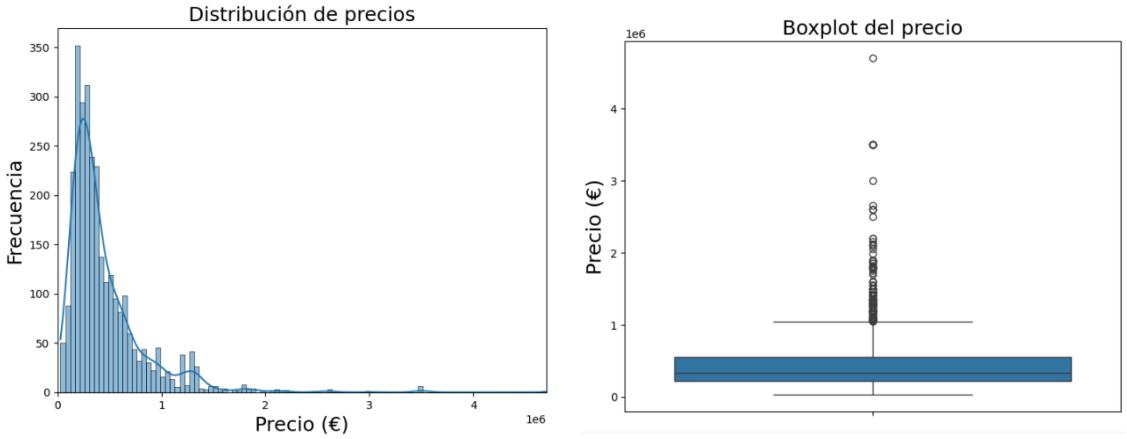
Este flujo de trabajo se replicó para todas las variables categóricas seleccionadas, asegurando que cada posible categoría quedara representada por su correspondiente indicador binario.

3.3. Visualización básica de datos

El análisis exploratorio se completó con una batería de visualizaciones diseñadas para entender la forma de la distribución de precios, el comportamiento de las variables explicativas y sus interacciones.

3.3.1. Análisis de la variable objetivo

Para comprender la naturaleza de la variable objetivo (precio en euros), se presentan a continuación dos representaciones gráficas: un **histograma** acompañado de su función de densidad estimada y un **diagrama de caja** (*boxplot*), junto con un resumen de las principales estadísticas descriptivas.



El histograma muestra que la distribución de precios presenta una **asimetría positiva** muy marcada. La mayoría de las observaciones se concentran en el rango inferior (menores de 600000€), mientras que existe una **larga cola** (*long tail*) que se extiende hasta valores cercanos a los 4,7 millones de euros. Este comportamiento es característico de datos económicos y de mercado inmobiliario, donde unos pocos inmuebles de alta gama elevan considerablemente el extremo derecho de la distribución.

- La **mediana** (330000 €) se encuentra por debajo de la **media** (448930€), evidenciando que la media está sesgada al alza debido a estos valores extremos.

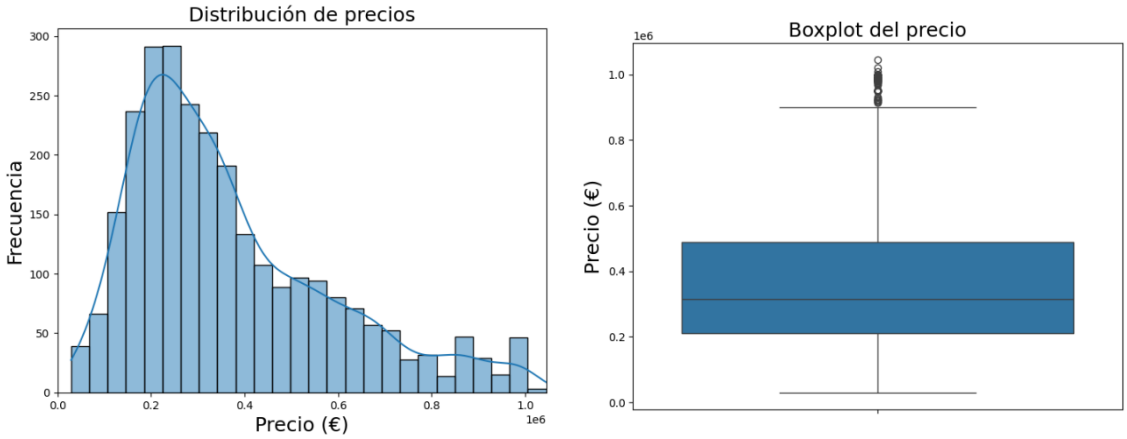
- La función de densidad suavizada corrobora la concentración de frecuencias en valores bajos y un descenso gradual que se prolonga a lo largo de sucesivos rangos de precio.

La presencia de esta cola larga sugiere que el conjunto de datos no sigue una distribución gaussiana.

El boxplot afirma la existencia de múltiples valores atípicos; todos los inmuebles con precio superior a 1,05 millones de euros quedan representados como puntos aislados. Se observa un número significativo de estos, con algunos casos extremos por encima de 3millones de euros, confirmando la "long tail" observada en el histograma.

Tratamiento de outliers

Para garantizar que el análisis de la variable precio no estuviese distorsionado por unos pocos inmuebles de precio excepcionalmente alto o bajo, se aplicó la regla del rango intercuartílico (IQR). Este método resulta apropiado en contextos de tasación inmobiliaria, donde conviene centrar los modelos en el "mercado habitual" y no en extremos puntuales. Tras aplicar este filtro, las gráficas y estadísticas de la variable *price* cambian.

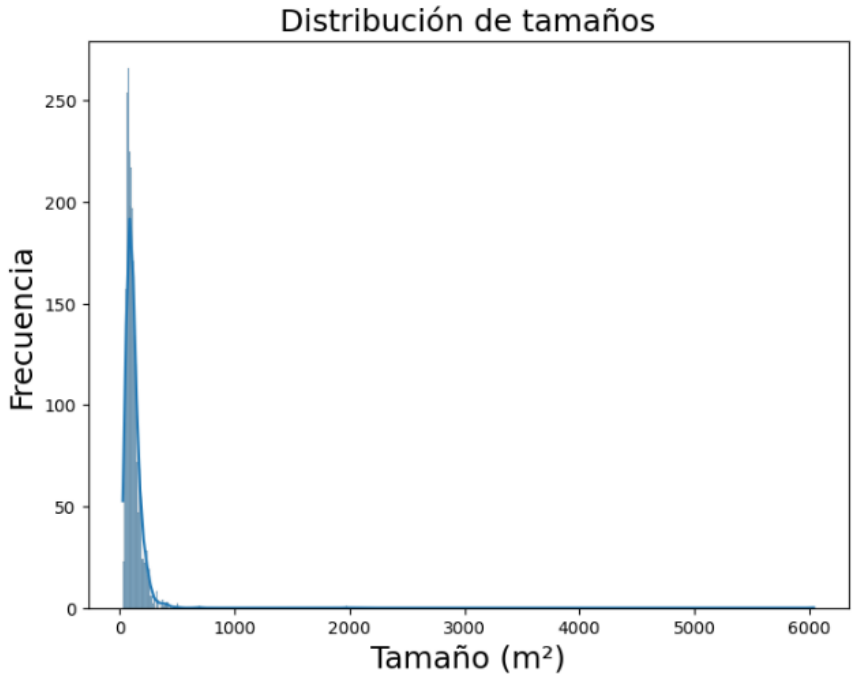


El histograma muestra ahora una cola derecha notablemente más corta, concentrando aproximadamente el 95 % de las observaciones en un rango inferior a 1200000€. La asimetría positiva persiste, pero con menor grado, lo que facilita el ajuste mediante métodos basados en supuestos de normalidad.

La caja se extiende desde Q1 = 210000€ hasta Q3 = 490000€, reduciendo el IQR a 280000€. El bigote superior alcanza ahora aproximadamente 880000€, y la cantidad de valores atípicos ha disminuido drásticamente, quedando relegados a precios superiores a 1045000€.

3.3.2. Análisis de las variables explicativas

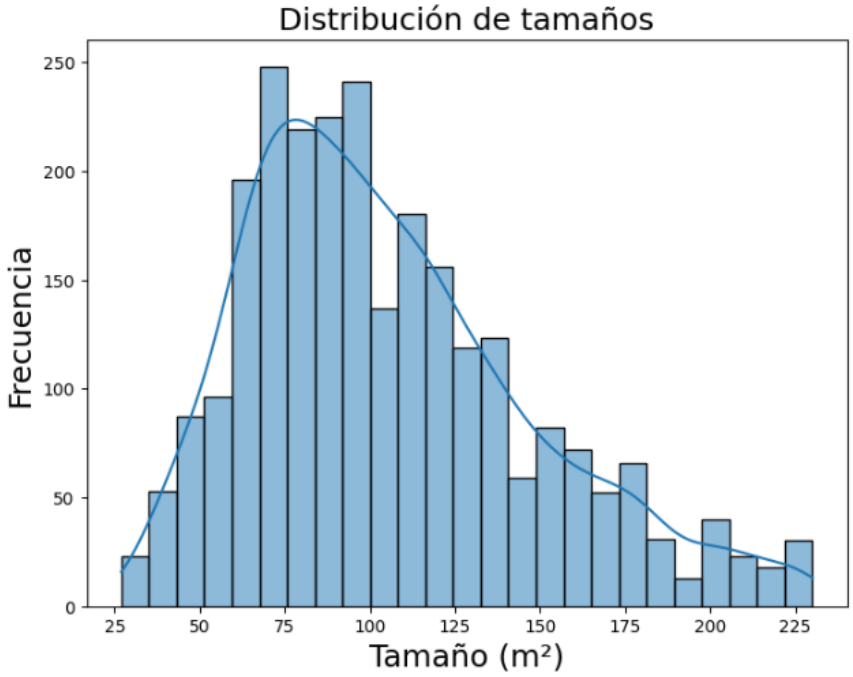
Analizamos ahora la variable explicativa **tamaño** (superficie en m^2) tras filtrar previamente los outliers en precio mediante IQR. A continuación, se muestra el histograma con estimación de densidad de probabilidad:



El histograma revela una **asimetría positiva** pronunciada en la distribución de las superficies de las viviendas.

La gran mayoría de las observaciones se sitúa por debajo de los 200 m^2, concentrándose especialmente en rangos entre 40 y 120 m^2, lo cual coincide con las tipologías de piso estándar en el mercado urbano de Valencia. Existe una **cola larga** que se extiende hasta valores cercanos a los 6 000 m^2, correspondiendo a bienes inmuebles atípicos (chalés unifamiliares, fincas rústicas u otros inmuebles de gran extensión). La densidad suavizada confirma un pico claro en torno a 80 m^2 y un decrecimiento gradual hacia tamaños superiores.

Este patrón de distribución es habitual en datos de vivienda, donde la mayoría de las unidades son pisos de tamaño medio, y un reducido número de propiedades con metrados muy altos genera una cola que podría distorsionar métricas de tendencia central si no se transforma o regula adecuadamente el conjunto..

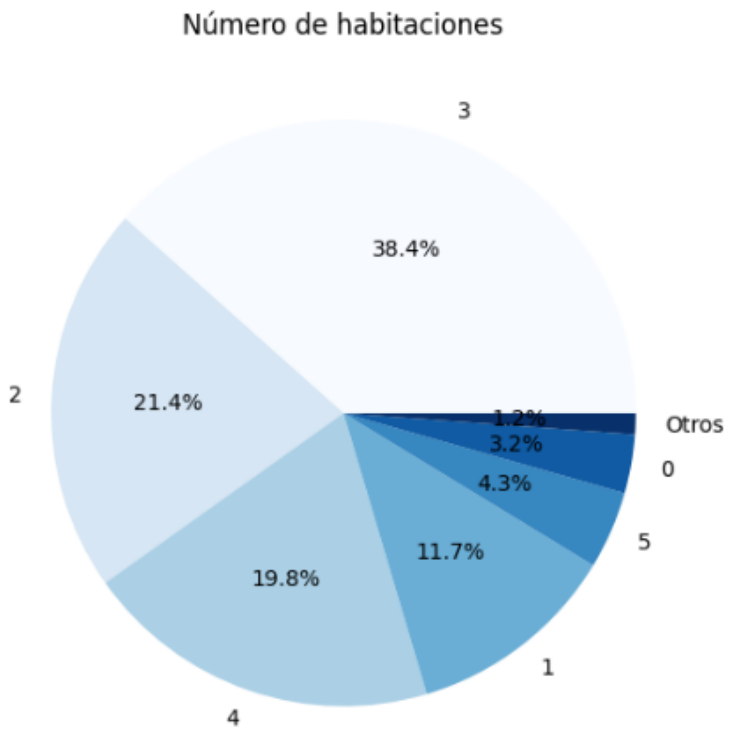


Al aplicar el método de rango intercuartílico (IQR) a la variable tamaño, eliminando observaciones por debajo de Q1 − 1,5·IQR y por encima de Q3 + 1,5·IQR, se obtiene el siguiente comportamiento en la distribución:

La cola derecha se reduce significativamente, concentrando la mayor parte de las superficies entre 40 y 230 m^2. La asimetría persiste aún de manera moderada, con un desplazamiento hacia la izquierda que sugiere un predominio de viviendas medianas. La función de densidad suavizada refleja un pico más definido en torno a los 80–90 m^2 y un tramo más uniforme en rangos superiores.

Este filtrado mejora la representatividad de la muestra para el modelo de predicción, ya que reduce el sesgo inducido por inmuebles atípicos de gran tamaño.

A continuación, se presenta la **distribución porcentual** de las viviendas según el número de habitaciones, ilustrada mediante un diagrama de sectores (*pie chart*):



La gráfica revela la siguiente distribución:

- **3 habitaciones**: con un **38,4 %**, constituyen el grupo mayoritario. Lo cual se explica por la prevalencia de familias de tamaño medio que requieren dos dormitorios para los hijos y uno para los padres, así como por el diseño urbanístico predominante en promociones recientes, donde tres habitaciones se han convertido en la tipología estándar.
- **2 habitaciones**: representan el **21,4 %** de la muestra, reflejando la presencia significativa de pisos más compactos, habituales en parejas o personas individuales que buscan economizar espacio.
- **4 habitaciones**: un **19,8 %**, indicando una proporción relevante de inmuebles de mayor tamaño, potencialmente dirigidos a familias numerosas o ocupaciones compartidas.
- **1 habitación**: **11,7 %**, englobando estudios y pisos de una sola estancia, muy comunes en el mercado de alquiler para estudiantes o trabajadores temporales.
- **5 habitaciones**: **4,3 %**, y **0 habitaciones** (lo que incluye estudios indefinidos o lofts muy pequeños): **3,2 %**.
- **Otros (6 o más)**: **1,2 %**, corresponden a propiedades atípicas o agrupar varios espacios habilitados como dormitorios.

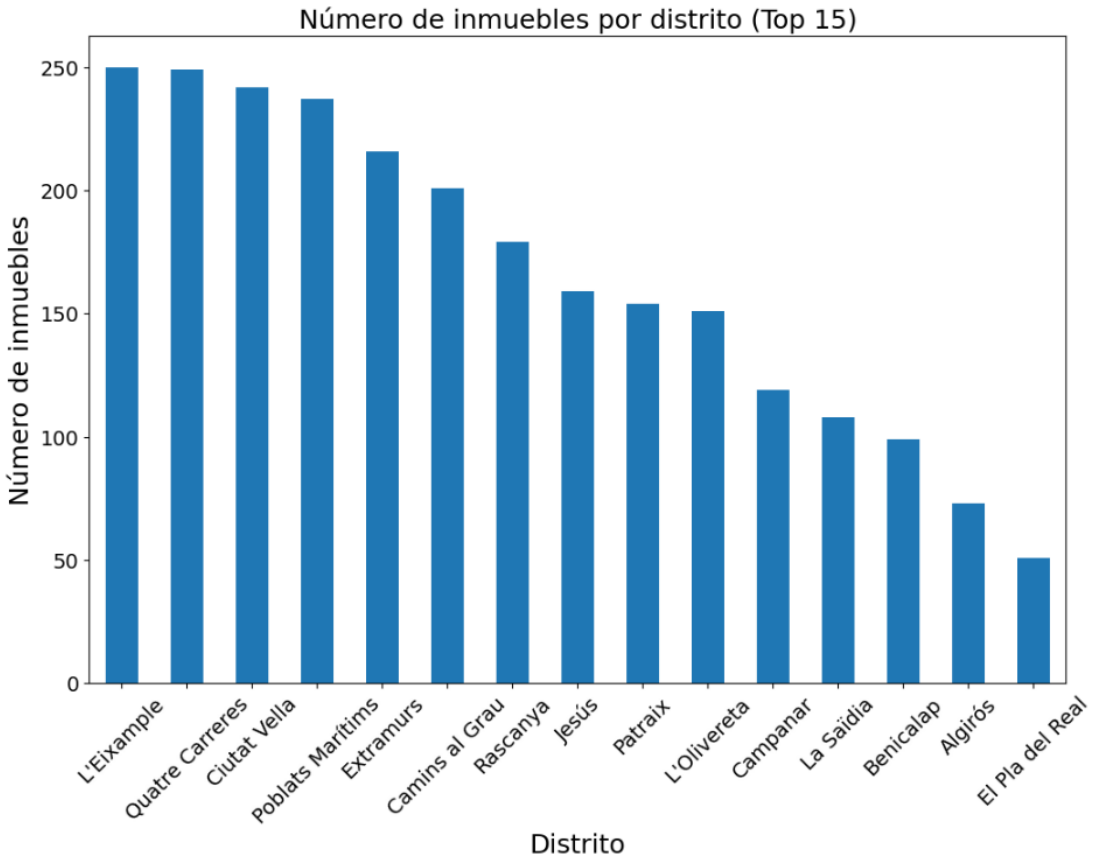
En el siguiente apartado integramos cuatro visualizaciones para narrar cómo interactúan la densidad de la oferta, la composición tipológica y el valor de la vivienda en los 15 barrios con mayor presencia de anuncios. Se incluyen cuatro gráficas principales:

1. **Número de viviendas por barrio (Top 15)**: muestra la frecuencia absoluta de inmuebles ofertados en los diez distritos con mayor presencia de datos.

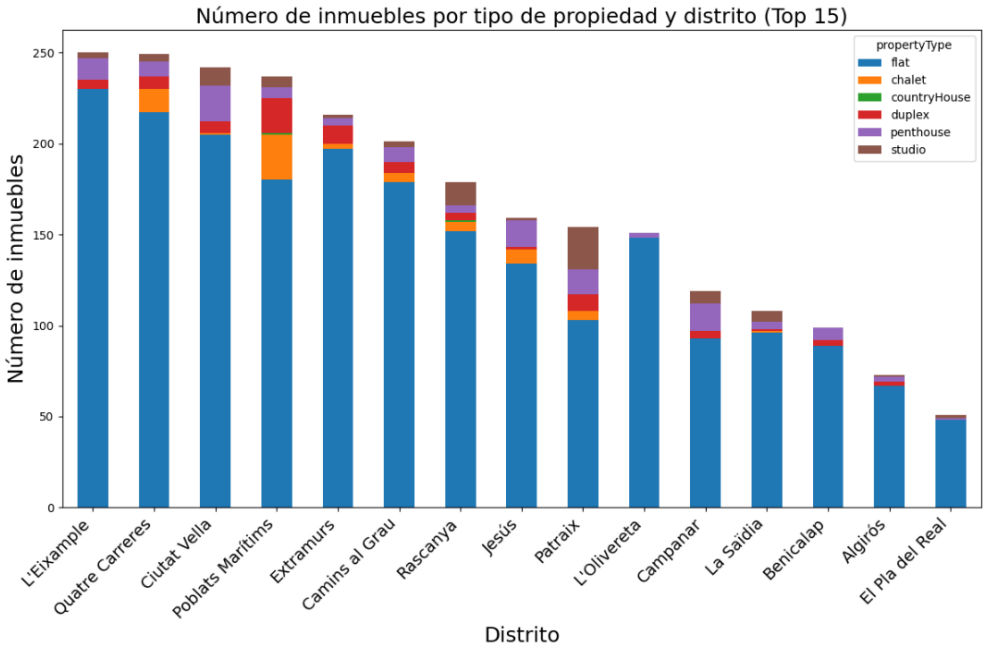
2. **Número de inmuebles por tipo de propiedad y distrito (Top 15)**: descompone la oferta de cada uno de estos distritos según la categoría de la vivienda (piso, chalet, dúplex, etc.).

3. **Precio según distrito (Top 15)**: diagramas de caja que ilustran la dispersión y rango de precios en cada distrito, permitiendo comparar medianas, cuartiles y outliers.

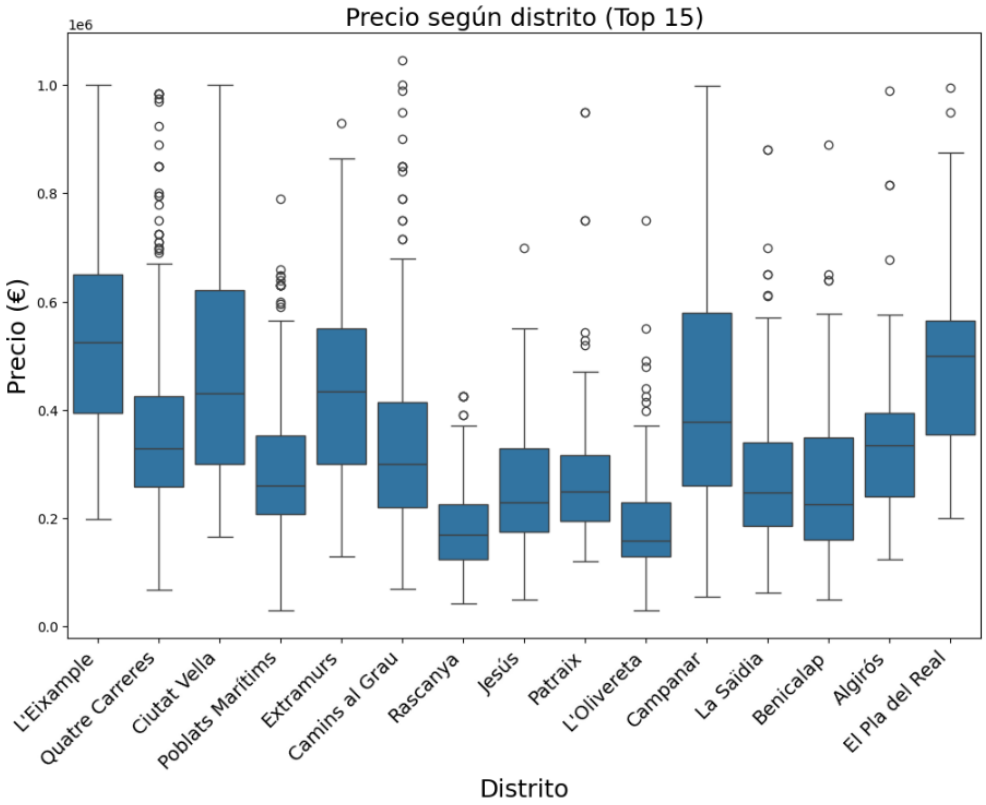
4. **Precio según vecindario (Top 15)**: similar al anterior, pero con mayor granularidad, mostrando la variabilidad de precios en unidades territoriales más pequeñas (barrios específicos dentro de los distritos).



Concentración de la oferta: La primera gráfica muestra que unos pocos distritos —como L'Eixample, Quatre Carreres y Ciutat Vella— concentran la mayor parte de los anuncios, mientras que barrios al margen del núcleo urbano (p. ej. El Pla del Real) cuentan con un número mucho más reducido de inmuebles. Esta distribución desigual de la muestra marca el escenario sobre el que se superponen los restantes factores: donde hay más oferta, la dinámica de precios tenderá a reflejar una mayor competencia y diversidad de productos.

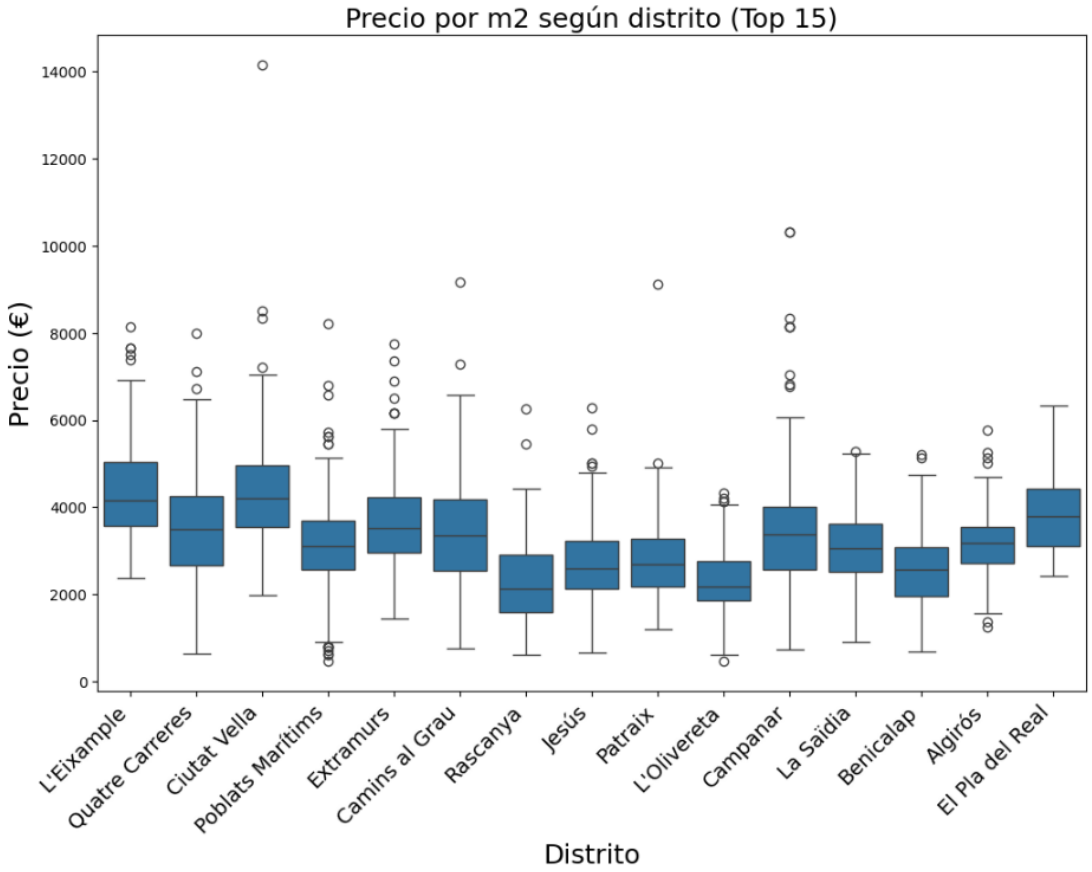


Heterogeneidad tipológica: La descomposición por tipo de propiedad muestra que los **pisos (flat)** dominan con más del 80 % del total en todos los distritos, pero surgen matices significativos en la penetración de otras tipologías. En **Poblats Marítims** y **Camins al Grau**, la proporción de **chalets** y **casas de campo (countryHouse)** asciende por encima del 10 %, reflejando la proximidad al litoral y la existencia de promociones unifamiliares. En contraste, **Ciutat Vella** y **L'Eixample** destacan por su porcentaje de **dúplex** y **penthouses**, vinculado a la rehabilitación de edificaciones históricas y a la búsqueda de producto premium con terrazas y vistas privilegiadas. Esta mayor heterogeneidad tipológica introduce un "ruido" adicional en la comparación de precios: donde coexisten múltiples formatos de vivienda, la dispersión de precios crece.



Impacto sobre el precio de venta: Los diagramas de caja por distrito revelan un contraste significativo entre la estabilidad de precios y la variabilidad asociada a tipologías exclusivas. En los barrios centrales como **L'Eixample** y **Ciutat Vella**, a pesar de contar con un elevado volumen de oferta, las medianas de precio se sitúan en rangos altos (aprox. 550000 € y 600000 €, respectivamente). Esta homogeneidad tipológica—mayoritariamente pisos de tamaño medio—tiende a comprimir la dispersión intercuartílica, favoreciendo precios relativamente estables alrededor de la mediana. No obstante, la rica oferta de productos premium (dúplex, penthouses) introduce algunos valores extremos, pero su incidencia es minoritaria frente al grueso de la muestra.

En cambio, distritos como **Rascanya** y **L'Olivereta**, donde coexisten pisos de protección oficial con chalets y unifamiliares de alto valor, exhiben medianas de precio más modestas (200 000–260 000 €) y una dispersión interna moderada. Sin embargo, los outliers en el extremo superior reflejan la presencia puntual de viviendas unifamiliares que, por su exclusividad, elevan la cola de la distribución. Este patrón ilustra cómo una oferta tipológicamente mixta amplía la variabilidad, al introducir formatos de valor muy distintos dentro del mismo territorio.



Ajuste por superficie: €/m²: Al normalizar el precio por metro cuadrado, la jerarquía espacial sufre ligeras modificaciones que ponen de manifiesto la densidad de valor frente al tamaño absoluto. Barrios de alta demanda y dimensiones compactas, como **Ciutat Vella Centro** y **L'Eixample Norte**, igualan o incluso superan a distritos de mayor metraje en términos de €/m², debido a que el precio concentrado en pisos más pequeños multiplica el coste unitario.

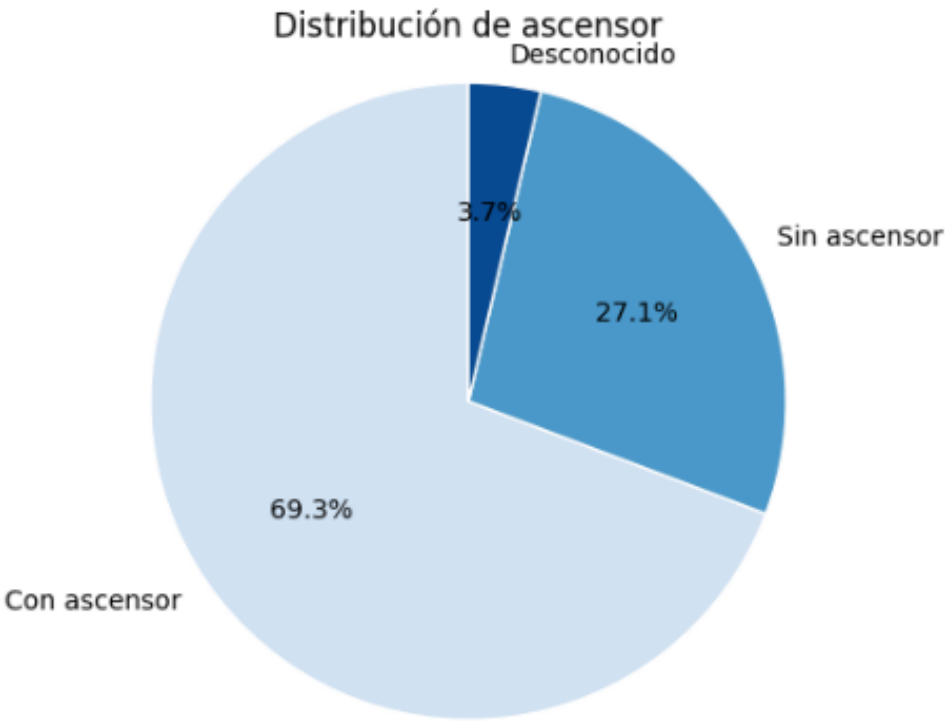
Por el contrario, zonas con predominio de chalets y dúplex, como **Patraix** y **Jesús**, descienden en el ranking de precio por metro cuadrado. A pesar de contar con precios

absolutos razonablemente altos, la gran superficie media diluye el coste unitario, reflejando un perfil de vivienda más espaciosa pero menos densa en valor.

Este análisis combinado muestra que el precio por m^2 no solo confirma la clasificación tradicional de barrios "premium" y "económicos", sino que atenúa el efecto de la superficie en distritos periféricos y refuerza la importancia de la densidad y la tipología de producto en el cálculo del valor inmobiliario.

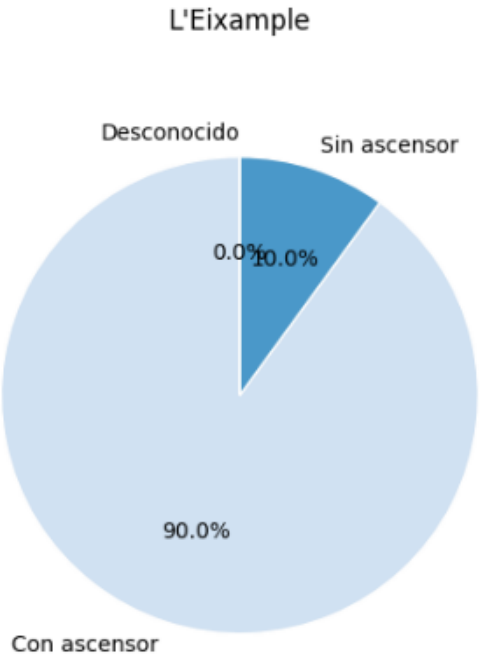
Distribución global

En el conjunto de 2724 registros depurados, el **69,3 %** de las viviendas cuenta con ascensor, el **27,1 %** carece de él y el **3,7 %** corresponde a casos sin información fiable. Este predominio de edificios dotados de ascensor refleja la fuerte implantación de la normativa de accesibilidad en las promociones recientes, así como la demanda de confort por parte de los compradores.



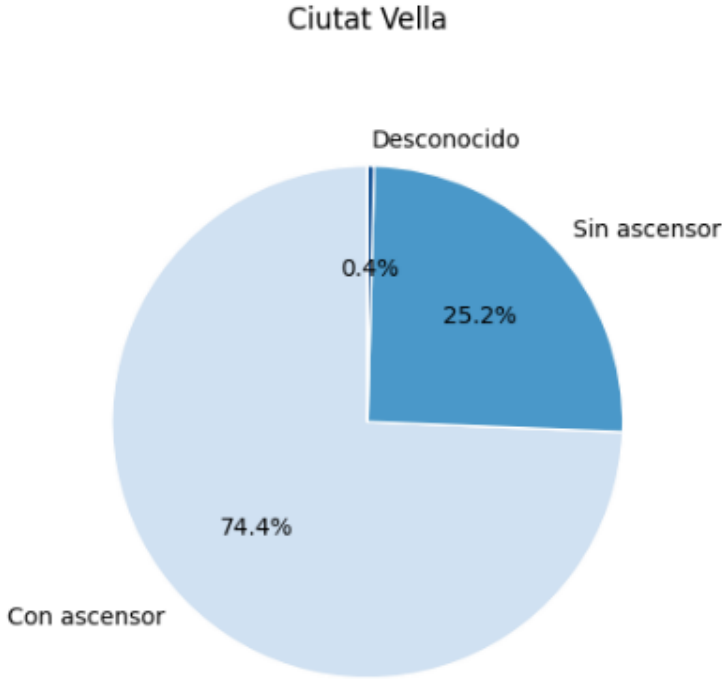
Análisis por distrito

A continuación se destacan tres distritos cuyo perfil de ascensor ilustra la interacción entre antigüedad de la edificación, intensidad de oferta y nivel de precios:



L'Eixample

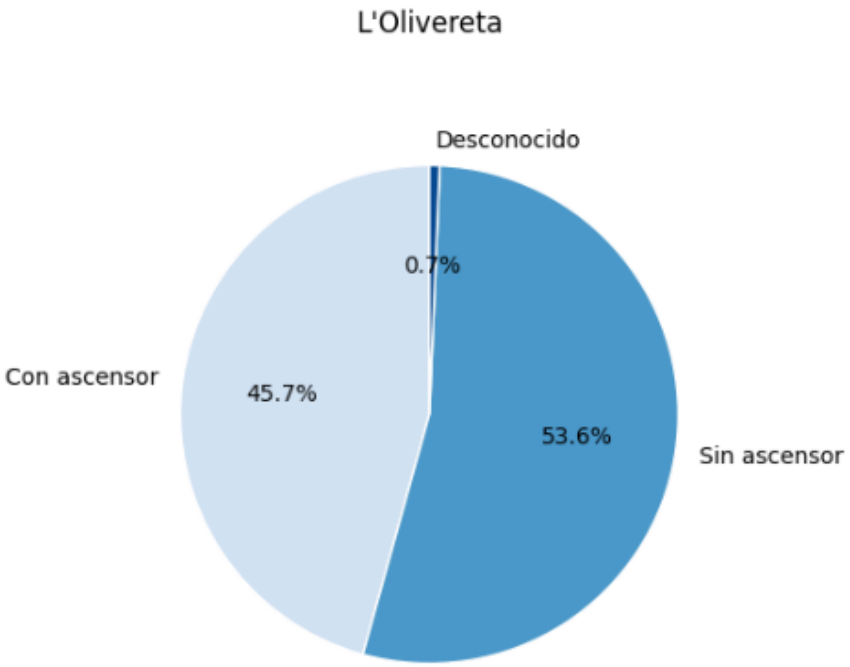
En este distrito, el **90,0 %** de los inmuebles dispone de ascensor y solo un **10,0 %** carece de él. L'Eixample concentra una de las medianas de precio más altas (550000€) y un elevado número de anuncios (>240). La casi universal presencia de ascensor pone de manifiesto que sus promociones, en gran parte del siglo XX y XXI, se han diseñado cumpliendo los estándares de accesibilidad más exigentes.



Ciutat Vella

A pesar de ser el corazón histórico de Valencia, un 74,4 % de las viviendas incorpora ascensor, mientras que el 25,2 % permanece en edificios antiguos sin esta instalación. La mediana de precio en Ciutat Vella supera los 600000€, gracias a la rehabilitación de

cascos tradicionales y la proliferación de dúplex y penthouses. La menor proporción de ascensores indica que persisten barreras de accesibilidad en fincas centenarias, aunque su renovación parcial ha elevado considerablemente el nº de instalaciones nuevas.



L'Olivereta

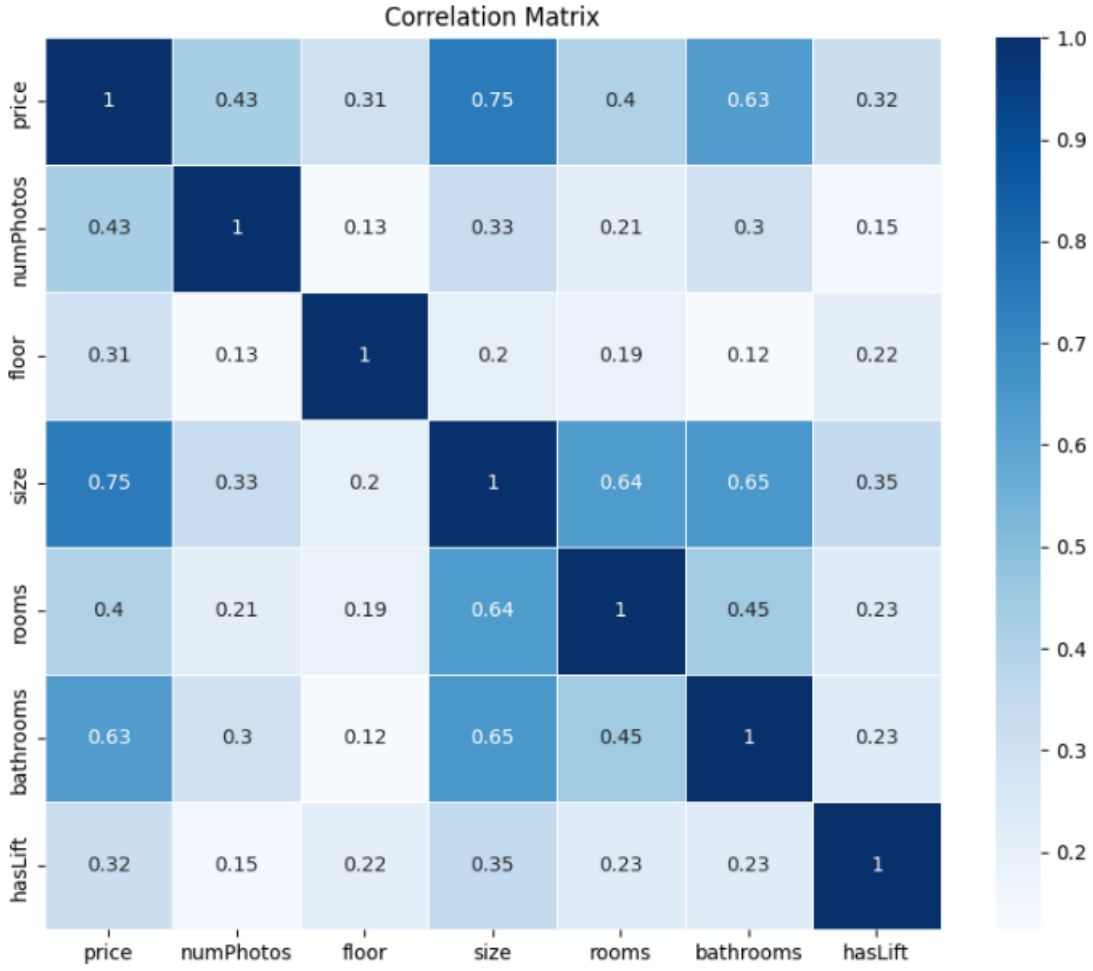
Este distrito presenta el $45,7\,\%$ de pisos con ascensor y un $53,6\,\%$ sin él, siendo el único de los analizados donde la ausencia de ascensor supera la presencia. Con una mediana de precio alrededor de 210000€, L'Olivereta refleja un parque de viviendas más antiguo o de tipología residencial unifamiliar baja, donde la normativa de accesibilidad ha llegado con menor intensidad y el coste de instalación en edificaciones de pocas plantas no siempre está justificado por el mercado.

En general el análisis de la distribución de *hasLift* pone de manifiesto que la presencia de ascensor está estrechamente ligada a la edad, la densidad de promoción y el valor inmobiliario de cada distrito. Los resultados clave son:

- **Alta penetración en mercados premium**: distritos con medianas de precio elevadas como L'Eixample ($90\,\%$) y Ciutat Vella ($74,4\,\%$) presentan una implantación casi universal del ascensor, reflejando la renovación urbana y la exigencia de accesibilidad en entornos de alto standing.

- **Ausencia destacada en zonas de menor valor**: distritos como L'Olivereta ($45,7\,\%$) y Poblats Marítims ($50,6\,\%$) conservan un parque considerable de edificios anteriores a la normativa de accesibilidad, lo que delimita un perfil de vivienda de menor coste y menor número de plantas.

3.3.3. Correlaciones entre variables



El patrón de correlaciones observado refleja un entramado de **variables físicas**, **económicas** y **estratégicas** que determinan el precio de mercado:

1. **Superficie** (size) **como factor estructural** (r = 0,75)
   - Cada metro cuadrado adicional añade un valor significativo, especialmente en distritos centrales con escasez de suelo.
   - La superficie es proxy de **economías de escala**: viviendas más grandes atraen a compradores con poder adquisitivo elevado, como familias y profesionales que teletrabajan.
   - En barrios consolidados (Ciutat Vella, L'Eixample) la demanda de espacios amplios empuja el precio al alza.
2. **Baños** (bathrooms) (r = 0,63)
   - Un mayor número de baños se asocia a **reformas recientes** y vivienda de "gama media-alta".
   - Las familias valoran la **privacidad y confort** que aportan baños adicionales, incrementando la disposición a pagar.
3. **Habitaciones** (rooms) (r = 0,40)

- Refleja la **densidad interna** del inmueble; su correlación parcial con `size` indica que la mera adición de habitaciones no equivale a mayor superficie.
- La subdivisión en más dormitorios aporta **flexibilidad** de uso (despacho, trastero), un factor diferencial en mercados de alquiler compartido.

4. **Fotografías (**numPhotos**)** (r = 0,43)

   - Indica **inversión en marketing**: inmuebles de precio elevado suelen presentar más fotografías profesionales, mejorando la percepción de transparencia y calidad.
   - Actúa como proxy de la **percepción de valor**: la abundancia de imágenes genera confianza y demanda más alta.

5. **Planta (**floor**)** (r = 0,31)

   - Plantas intermedias proporcionan **luminosidad** y **vistas** sin el coste extra de edificaciones muy altas.
   - Afecta a la accesibilidad: plantas bajas son cómodas pero ruidosas, plantas altas requieren ascensor.

6. **Ascensor (**hasLift**)** (r = 0,32)

   - Marca la **antigüedad y normativa**: edificios posteriores a los años 60 suelen incorporar ascensor, un símbolo de modernidad.
   - Su presencia añade valor en pisos altos y mejora la **accesibilidad** para personas mayores o con movilidad reducida.

3.4. Primeros estudios con dos variables

3.4.1. Relación variables HasLift y Price

La agrupación de los datos por la variable **hasLift** (código 0 = sin ascensor, 1 = con ascensor, 3 = valor desconocido) arroja las siguientes medias de precio:

hasLift	Significado	Precio medio (€)
0	Viviendas sin ascensor	235234.223€
1	Viviendas con ascensor	514858.279€
3	Se desconoce si hay ascensor	564942.109€

Las propiedades dotadas de ascensor muestran un precio medio **más del doble** (aprox. 515000 €) que aquellas que carecen de esta característica (235000 €). Este incremento obedece probablemente a varios factores:

1. **Accesibilidad y confort**: el ascensor es percibido como un servicio esencial en edificios de varias plantas, especialmente para familias con niños, personas mayores o residentes con movilidad reducida.

2. **Edad y tipología del edificio**: las promociones más modernas incorporan ascensor como estándar y suelen situarse en zonas de mayor valor (nuevos desarrollos en distritos premium).

3. **Tamaño y número de plantas**: los inmuebles con ascensor tienden a encontrarse en edificios de más altura y con mayor número de viviendas, lo cual a su vez se asocia a calidades constructivas superiores y, por ende, a precios elevados.

El valor medio más alto registrado en la categoría "desconocido" indica que este grupo agrupa tanto inmuebles de alta gama sin información correcta como posibles inconsistencias.

3.4.2. Relación Variables HasParking, ParkingIncluded y Price

Para analizar el efecto de la plaza de garaje, se ha considerado la variable **HasParking** (1 = dispone de plaza, 3 = desconoce) conjuntamente con **ParkingIncluded** (0 = plaza no incluida en el precio, 1 = plaza incluida en el precio, 3 = desconocido). El precio medio resultante es:

HasParking	ParkingIncluded	Significado	Precio medio (€)
1	0	Vivienda con Parking pero no está incluido en precio	607237.666€
1	1	Vivienda con Parking pero no está incluido en precio	630697.968€
3	3	No se sabe si tiene Parking ni si está incluido	383737.145€

Las viviendas que disponen de plaza de garaje alcanzan medias altas en ambos casos, con un ligero sobreprecio (aprox. 23000 €) cuando la plaza viene **incluida en el precio de venta**. Esto sugiere que los compradores valoran la inclusión directa de este servicio, prefiriendo evitar costes adicionales y gestiones de aparcamiento.

Nuevamente, la categoría "desconocido" presenta un precio medio intermedio-bajo, lo que dificulta su interpretación.

Estos análisis bivariados muestran cómo elementos de **confort** (ascensor) y **servicios asociados** (plaza de garaje) tienen un impacto sustancial en el precio medio de las viviendas, con incrementos de entre un 100 % (ascensor) y un 4 % (plaza incluida) respecto a sus homólogos sin dicha característica.

3.5. Modelos predictivos

Antes de proceder con la aplicación de los modelos predictivos, se realizó una partición del conjunto de datos con el fin de evaluar de forma objetiva su capacidad de generalización. Para ello, los datos se dividieron aleatoriamente en dos grupos: uno destinado al entrenamiento del modelo (80 % del total) y otro reservado para su evaluación final (20 %). Esta división aleatoria permite asegurar que ambos subconjuntos sean representativos y que los resultados obtenidos no estén condicionados por una distribución particular.

En los modelos que requieren ajustar ciertos parámetros internos para mejorar su rendimiento, se utilizó una técnica conocida como validación cruzada, implementada mediante la herramienta *GridSearchCV* o *RandomizedSearchCV*

Por otro lado, para los modelos de mayor complejidad como Random Forest, XGBoost y MLP se consideró imprescindible analizar el impacto de los *outliers* en el desempeño predictivo. Dado que la presencia de valores atípicos puede distorsionar significativamente las métricas de evaluación, se definieron dos esquemas paralelos de partición de los datos:

1. **Tr_1 + Ts_1**: conjunto de entrenamiento y prueba tras eliminar los *outliers*.
2. **Tr_2 + Ts_2**: conjunto de entrenamiento y prueba conservando los *outliers*.

Cada modelo fue entrenado por separado sobre Tr_1 y Tr_2, y posteriormente evaluado tanto en Ts_1 como en Ts_2. Este enfoque permitió comparar el comportamiento del mismo algoritmo ante distintos contextos de entrada, proporcionando una perspectiva más completa sobre su robustez.

3.5.1. Modelo de Regresión Lineal Simple entre las variables price y size

Para establecer una relación directa entre la superficie de la vivienda y su precio, implementamos un modelo de regresión lineal simple. En primer lugar, seleccionamos como variable independiente el tamaño (m^2) y como variable dependiente el valor de venta.

3.5.2. Modelo de Regresión Lineal Múltiple

Como el objetivo es encontrar una relación lineal entre un conjunto de variables independientes y una variable dependiente, empleamos como variables explicativas todas aquellas que tenemos en formato numérico y como variable predictiva el precio de la vivienda. El uso de datos numéricos facilita esta aproximación y evita la necesidad de transformaciones adicionales que podrían introducir ruido o complejidad innecesaria en este tipo de modelo.

3.5.3. Random Forest Regressor

El modelo de bosques aleatorios se caracteriza por construir múltiples árboles de decisión y combinar sus predicciones para obtener un resultado más robusto y generalizable. Para mejorar su rendimiento, se ajustaron diversos hiperparámetros mediante búsqueda en malla que incorporó validación cruzada de cinco particiones, de modo que en cada iteración el modelo se entrena con cuatro quintas partes de los datos y se valida con la quinta restante, rotando este rol hasta completar las cinco combinaciones. Como criterio de selección de la mejor configuración se empleó el error absoluto medio (MAE), traducido al formato de GridSearchCV como "neg_mean_absolute_error" para que el algoritmo busque minimizar precisamente el MAE. Asimismo, se habilitó la ejecución en paralelo aprovechando todos los núcleos disponibles, lo que permitió reducir drásticamente el tiempo de cómputo sin sacrificar exhaustividad en la exploración de la malla. A continuación, se describen los más relevantes:

- **n estimators**: Este parámetro define la cantidad de árboles que conforman el bosque. Un número elevado suele traducirse en predicciones más estables, ya que reduce la varianza del modelo al promediar múltiples estimaciones independientes.
- **max depth**: Controla la profundidad máxima permitida para cada árbol. Limitar esta profundidad puede evitar que los árboles se vuelvan excesivamente complejos y se ajusten demasiado a los datos de entrenamiento, lo que favorece una mejor generalización.
- **max features**: Indica la cantidad de características que pueden considerarse al buscar la mejor división en cada nodo. Al restringir este número, se introduce diversidad entre los árboles del conjunto, lo cual es clave para el buen desempeño del modelo en datos no vistos.
- **min samples split**: Establece el número mínimo de muestras requerido para dividir un nodo. Este parámetro ayuda a controlar el crecimiento de los árboles, evitando que se generen divisiones con poca información estadística.
- **min samples leaf**: Define la cantidad mínima de observaciones que debe haber en una hoja terminal. Ajustar este valor permite regular la granularidad de las predicciones, ya que evita que se creen hojas que representen un número muy reducido de ejemplos.

Siendo los mejores hiperparámetros:

- Entrenado con outliers: 'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 300
- Entrenado sin outilers: 'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 200

3.5.4. Extreme Gradient Boosting

El algoritmo XGBoost, basado en la técnica de *gradient boosting*, se caracteriza por construir modelos de forma secuencial, donde cada nuevo árbol trata de corregir los errores cometidos por el conjunto anterior. Para controlar su comportamiento y mejorar su capacidad de generalización, se ajustaron de la misma manera que el modelo anterior (validación cruzada de cinco particiones y criterio de selección de la mejor configuración se empleó el error absoluto medio)varios hiperparámetros clave:

- **learning rate**: Este parámetro determina la magnitud del ajuste que realiza cada nuevo árbol sobre las predicciones anteriores. Un valor bajo reduce el impacto individual de cada árbol, lo que puede hacer que el modelo aprenda de forma más lenta pero con mayor precisión, mientras que valores más altos aceleran el proceso pero incrementan el riesgo de sobreajuste.
- **n estimators**: Indica el número total de árboles que se entrenarán de forma secuencial. Un mayor número de estimadores permite capturar patrones más complejos, aunque también puede incrementar el riesgo de sobreentrenamiento si no se regula adecuadamente.

- **max depth**: Controla la profundidad máxima permitida para cada árbol. Una profundidad reducida genera árboles más simples y menos propensos al sobreajuste, mientras que mayores profundidades permiten capturar relaciones más complejas en los datos.
- **subsample**: Define la fracción del conjunto de datos de entrenamiento que se utiliza para construir cada árbol. Introducir esta aleatoriedad contribuye a reducir la varianza del modelo y mejora su capacidad de generalización.
- **colsample bytree**: Especifica la proporción de características que se consideran al construir cada árbol. Limitar el número de variables disponibles en cada iteración introduce diversidad entre los árboles y puede ayudar a prevenir el sobreajuste.

Siendo los mejores hiperparámetros:

- Entrenado con outliers: 'colsample_bytree': 0.7, 'learning_rate': 0.01, 'max_depth': 9, 'n_estimators': 500, 'subsample': 0.7
- Entrenado sin outilers: 'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 500, 'subsample': 1.0

3.5.5. Multilyer Perceptron

El perceptrón multicapa es una red neuronal artificial que aprende a partir de datos mediante un proceso iterativo de ajuste de pesos. Al tratarse de un modelo sensible a la configuración de sus hiperparámetros, se exploraron diversas opciones. Para afinar los hiperparámetros del perceptrón multicapa se recurrió a una búsqueda aleatoria con validación cruzada, de manera que el proceso resultara viable en tiempo sin renunciar a la robustez de la selección. Concretamente, se utilizó **RandomizedSearchCV** para muestrear cincuenta configuraciones distintas de la red neuronal, evaluando cada una mediante una validación de tres pliegues. Este planteamiento reduce significativamente el coste computacional frente a una exploración exhaustiva, al tiempo que garantiza que la elección de parámetros se basa en un desempeño consistente sobre distintas particiones de los datos. Como criterio de optimización se empleó el error absoluto medio (MAE), configurado en el sistema como "neg_mean_absolute_error", de modo que el algoritmo busca minimizar la desviación absoluta promedio. Además, se habilitó el entrenamiento paralelo en todos los núcleos disponibles . A continuación se detallan los parámetros más relevantes:

- **hidden layer sizes**: Define la arquitectura de la red, es decir, el número de capas ocultas y la cantidad de neuronas en cada una. Este parámetro es fundamental para determinar la capacidad del modelo para aprender representaciones complejas de los datos.
- **alpha**: Corresponde al parámetro de regularización L2, que actúa como penalización para los pesos grandes. Su ajuste ayuda a controlar el sobreajuste y mejora la generalización del modelo.
- **learning rate init**: Representa el valor inicial de la tasa de aprendizaje. Un valor adecuado permite que el modelo comience a aprender con pasos razonables, ni demasiado pequeños ni excesivos.

Siendo los mejores hiperparámetros:

- Entrenado con outliers: ‘alpha’: np.float64(0.008731034258755935), ’hidden_layer_sizes’: (100, 50), ’learning_rate_init’: np.float64(0.004594506741382034)

- Entrenado sin outilers: ‘alpha’: np.float64(0.008731034258755935), ’hidden_layer_sizes’: (100, 50), ’learning_rate_init’: np.float64(0.004594506741382034)

Capítulo 4

Resultados y Discusión

4.1. Resultados

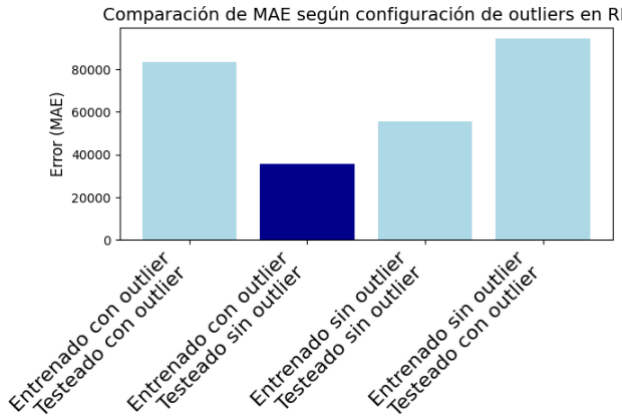
4.1.1. Validación y Evaluación

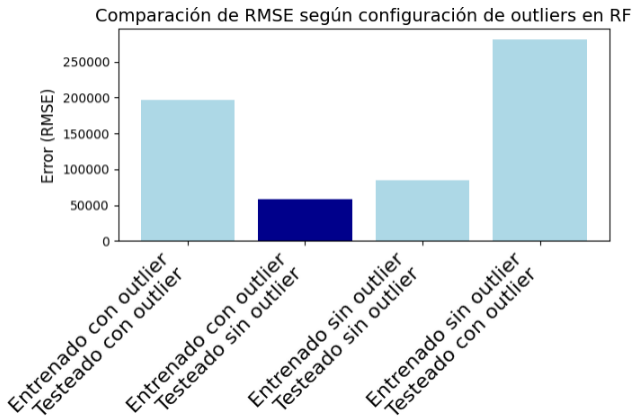
Antes de entrar en el análisis detallado de los resultados, cabe señalar que, aunque en la fase de experimentación se calcularon cinco métricas de error y ajuste (MAE, MSE, RMSE, R^2 y MRE), para la comparación de los modelos de predicción de precio inmobiliario nos centraremos únicamente en **MAE, RMSE y R^2**. La decisión de limitar el informe a estas tres métricas responde a varias razones:

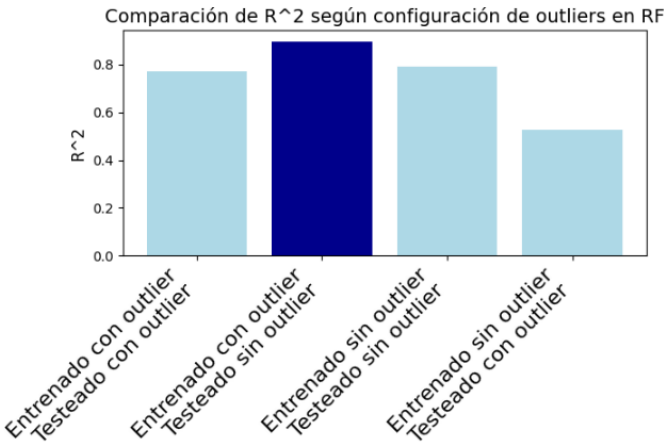
- **MAE y RMSE en euros**: el MAE ofrece el error medio directo; el RMSE penaliza más los desvíos grandes.
- **R^2 adimensional**: mide la proporción de varianza explicada, facilitando comparaciones entre modelos.
- **Evita redundancias**: la MSE usa unidades al cuadrado y la MRE resulta menos estable e intuitiva con valores extremos.

Comenzaremos comparando las cuatro variantes de cada modelo que hemos indicado al inicio del apartado anterior.

Random Forest Regressor







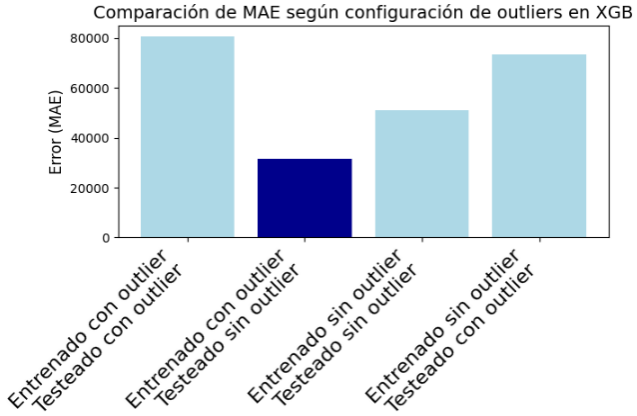
Variante ganadora: *Entrenado con outliers / Testeado sin outliers*

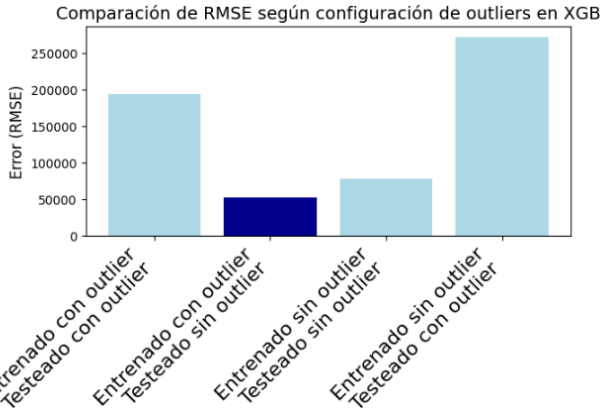
- **Mayor diversidad en el entrenamiento.** Mantener los valores extremos durante el ajuste permite que los árboles aprendan umbrales de decisión más amplios y capturen relaciones que de otro modo quedarían fuera de la muestra típica.

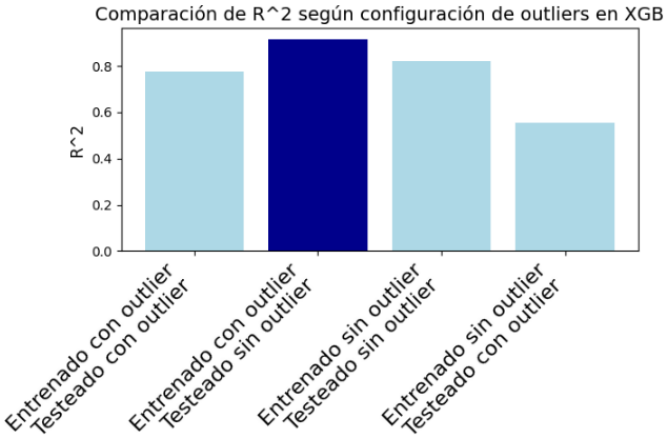
- **Evaluación en datos limpios.** Al probar sobre un conjunto libre de outliers, las predicciones no sufren penalizaciones por errores en valores atípicos, lo que reduce drásticamente MAE y RMSE y eleva el R^2.

- **Robustez ante la varianza.** Los bosques aleatorios, al promediar muchos árboles, toleran bien la presencia de ejemplos extremos en el entrenamiento, pero se benefician de una evaluación en escenarios menos ruidosos.

XGBoost







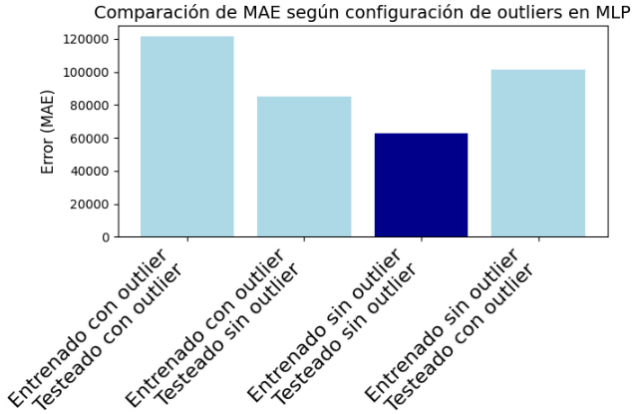
Variante ganadora: *Entrenado con outliers / Testeado sin outliers*

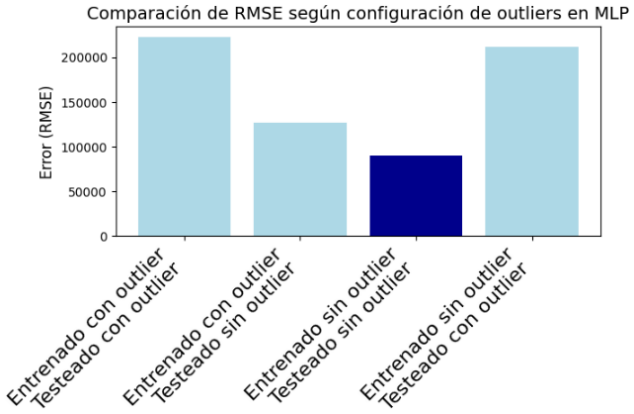
- **Aprovechamiento de gradientes fuertes.** XGBoost ajusta nuevos árboles para corregir los errores del conjunto previo; disponer de outliers en entrenamiento refuerza su capacidad de atacar grandes desvíos.

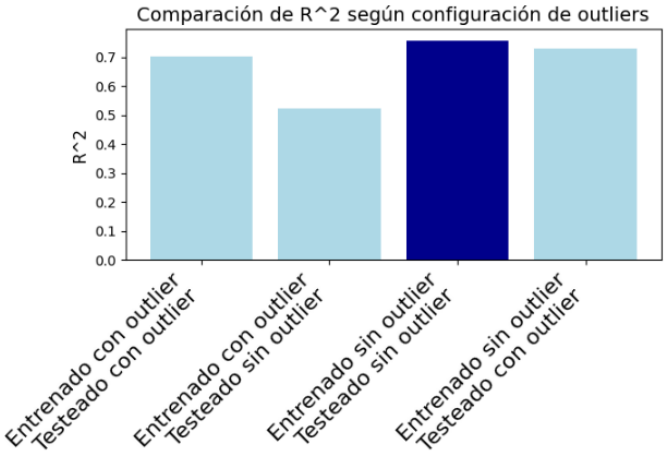
- **Reducción de penalización en test.** Al evaluar sobre datos sin valores extremos, se minimiza la contribución de errores grandes al RMSE, mientras el MAE se mantiene bajo como en el anterior caso.

- **Regularización implícita.** Aunque XGBoost incorpora técnicas para evitar el sobreajuste, su poder de corrección en presencia de outliers se traduce en umbrales más precisos que, al aplicarse en un test limpio, elevan la métrica de R^2.

Multilayer Perceptron (MLP)







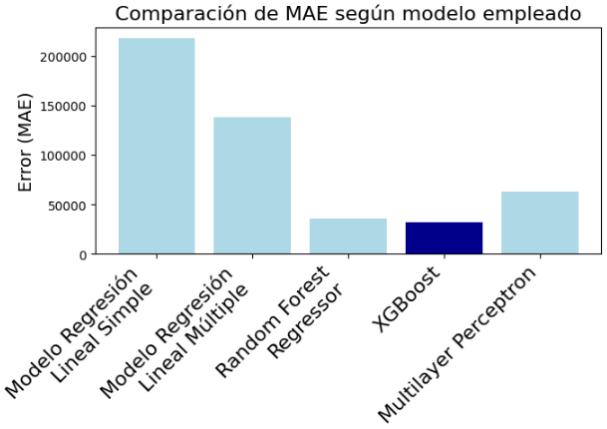
Variante ganadora: *Entrenado sin outliers / Testeado sin outliers*

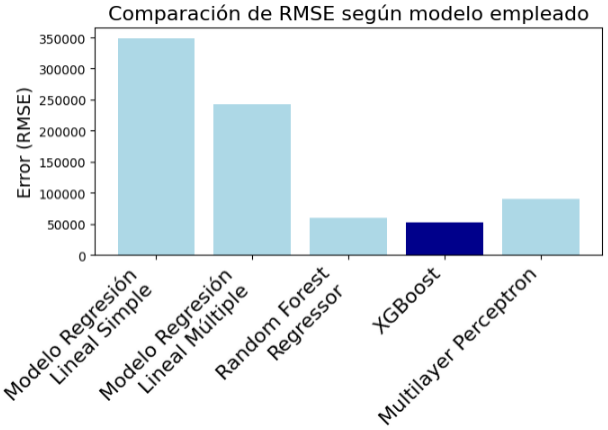
- **Sensibilidad a extremos.** Las redes neuronales basadas en gradiente son muy influenciables por valores atípicos, que pueden generar gradientes excesivos y dificultar la convergencia de los pesos.

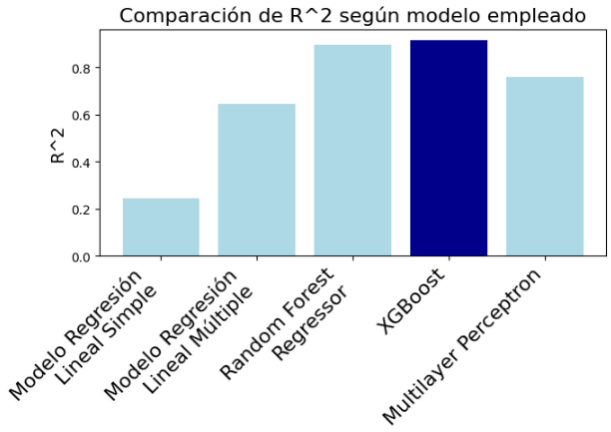
- **Convergencia más estable.** Al entrenar con datos depurados, el MLP ajusta sus funciones de activación y tasas de aprendizaje de forma más homogénea, reduciendo oscilaciones y evitando puntos muertos en el entrenamiento.

- **Evaluación coherente.** Probar sobre el mismo tipo de datos (sin outliers) asegura que las métricas reflejen la capacidad real de predicción en la "zona normal" del mercado inmobiliario, sin distorsiones por extremos.

Para decidir **qué algoritmo emplear finalmente** en la predicción del precio inmobiliario, examinamos de forma conjunta las tres métricas seleccionadas en la configuración óptima para cada modelo. A continuación se presentan los resultados de manera gráfica, lo que facilita la comparación visual de los diferentes modelos:







Además de las representaciones gráficas, a continuación se presentan los mismos resultados de manera numérica en una tabla para permitir una comparación más precisa de los valores obtenidos por cada modelo:

	MAE	RMSE	R^2
Regrseión Lineal Simple	218291,61	348873,25	0,243
Regresión Lineal Múltiple	138213,74	243272,52	0,646
Random Forest Regressor	35567,58	58889,48	0,897
XGBoost	**31661,19**	**52976,49**	**0,917**
Multilayer Perceptron	51094,29	77971,71	0,820

- **Modelos lineales** (simple y múltiple) quedan muy rezagados: muestran errores medios y cuadráticos muy elevados y una capacidad explicativa (R^2) insuficiente para usos prácticos.
- **Random Forest** ya reduce drásticamente el error y explica alrededor del 90 % de la varianza, pero XGBoost mejora aún más todas las métricas.
- El **MLP**, aunque supera ampliamente a las regresiones lineales, no alcanza la precisión ni la estabilidad de los métodos basados en árboles.

Elección del modelo definitivo

De acuerdo con estos resultados, **XGBoost Regressor** es el candidato más adecuado para la predicción de precios inmobiliarios, porque:

- **Menor error absoluto y cuadrático**: sus MAE y RMSE son los más bajos, lo que implica predicciones más cercanas al valor real y menos penalización por errores extremos.
- **Mayor capacidad explicativa**: alcanza el R^2 más alto $(0,92)$, indicando que captura mejor las relaciones entre las características y el precio.
- **Robustez y regularización incorporada**: combina el poder de ensamble con mecanismos de control de complejidad, evitando el sobreajuste incluso en presencia de outliers en entrenamiento.

Al ser el modelo definitivo que vamos a emplear en la interfaz de predicción, considero que también es importante tener en cuenta la métrica **MRE (Mean Relative Error)**, que ha obtenido un valor de -0.061. El MRE mide el error relativo promedio de las predicciones respecto al valor real. Un valor de MRE cercano a cero indica que las predicciones del modelo son muy cercanas a los valores reales, lo cual es esencial para aplicaciones prácticas donde la precisión es clave. En este caso, un MRE negativo de **-0.061** sugiere que, en promedio, las predicciones tienden a subestimar ligeramente el precio real de los inmuebles. Aunque un MRE negativo no es necesariamente un problema, es importante monitorear esta tendencia, ya que implica que el modelo podría estar prediciendo precios ligeramente por debajo de los valores reales, lo que podría ser relevante dependiendo del uso final de la herramienta, especialmente en aplicaciones que buscan ofrecer una estimación conservadora o ajustada de los precios del mercado.

Dicho esto, XGBoost será el modelo empleado en la fase final de predicción mediante interfaz, garantizando el mejor equilibrio entre precisión y generalización.

4.1.2. Interfaz Gráfica

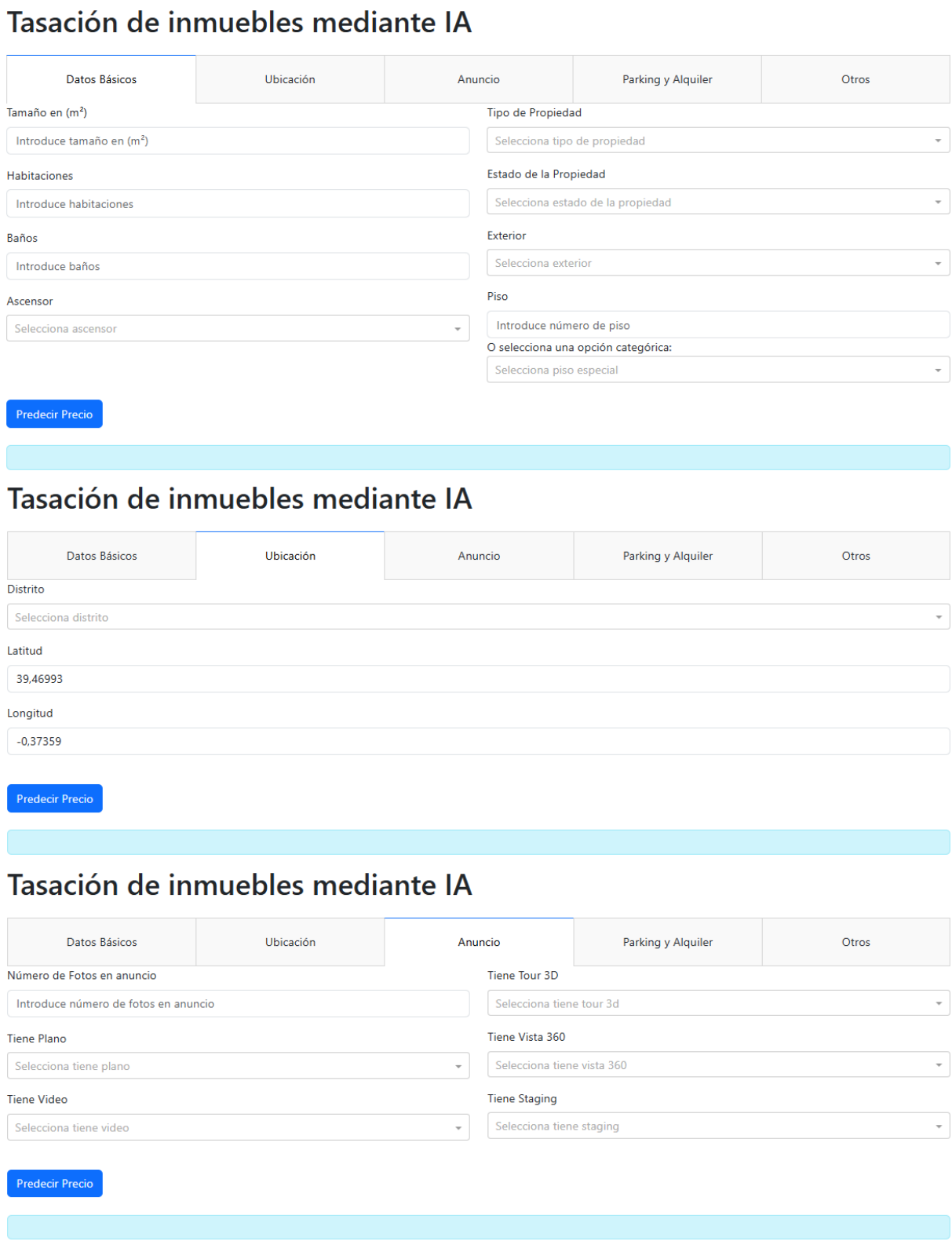
Para facilitar la explotación práctica del modelo XGBoost seleccionado, se desarrolló una interfaz web que permite a cualquier usuario—desde un profesional inmobiliario hasta un particular sin conocimientos técnicos—introducir las características de una vivienda y obtener al instante una estimación de su precio de venta. En la parte superior de la pantalla se muestra un formulario sencillo con diferentes pestañas, en las que el usuario puede indicar tanto los atributos básicos del inmueble (superficie en metros cuadrados, número de habitaciones y baños) como parámetros asociados a la ubicación anuncio... Cada campo cuenta con controles diseñados para reducir errores de entrada: se emplean menús desplegables para elecciones discretas y validaciones en tiempo real que alertan al usuario si se introduce un valor fuera de los límites razonables.

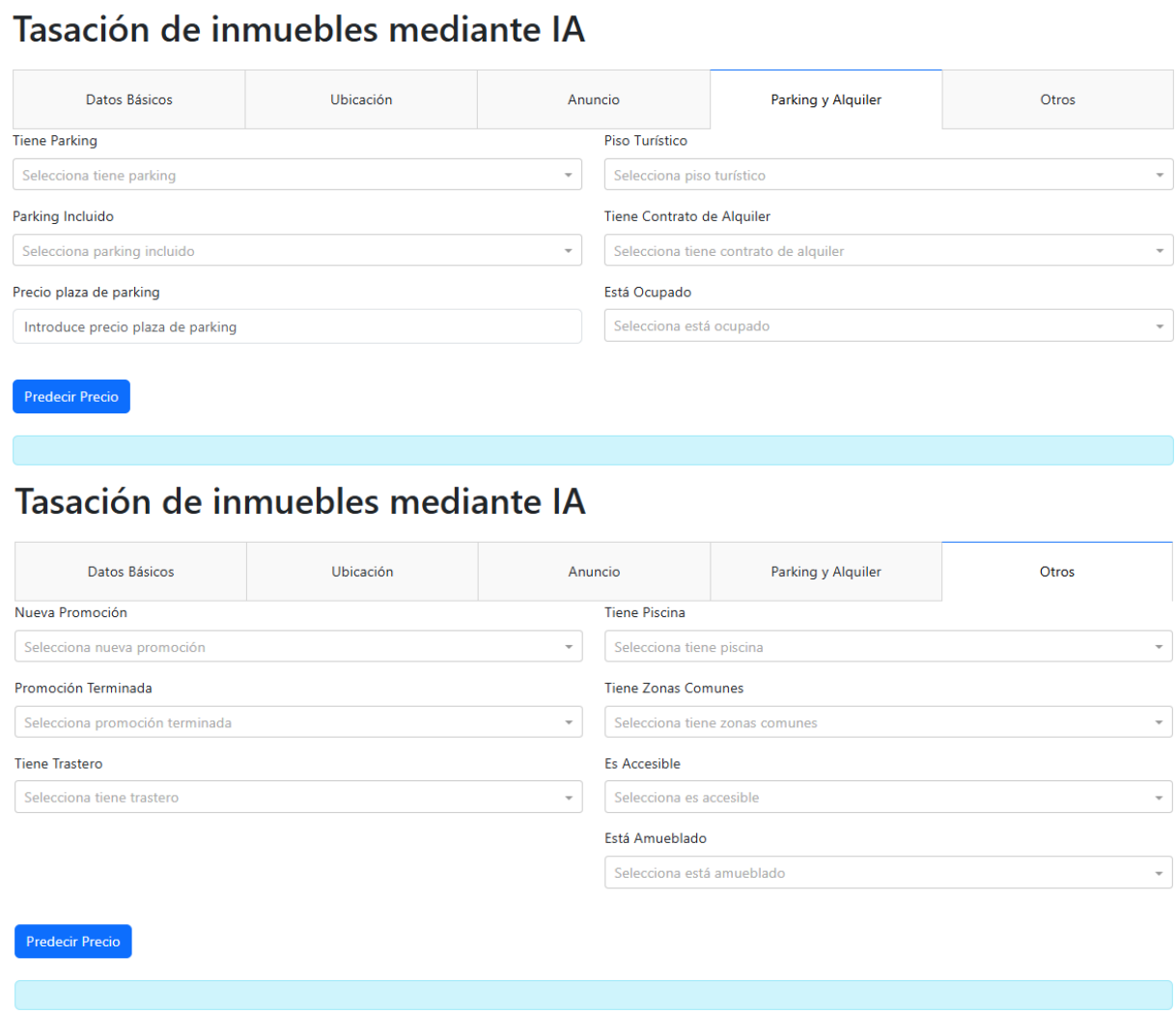
Una vez completados todos los apartados, el botón azul “Predecir Precio” envía los datos al servidor donde se ejecuta XGBoost entrenado, devolviendo en tiempo real la estimación puntual del valor de mercado. Junto a estos resultados, la interfaz muestra automáticamente la media de precio de los inmuebles del distrito en el que se ubica la vivienda introducida y calcula la diferencia absoluta con la predicción del modelo. Esta información comparativa permite al usuario contextualizar la tasación frente al comportamiento medio de su zona y realizar sus propios cálculos de margen o ajuste.

Un aspecto clave de esta herramienta es la posibilidad de comparar escenarios de reforma o viviendas completamente hipotéticas. El usuario puede clonar los datos introducidos y modificar selectivamente parámetros—por ejemplo, añadir un baño, añadir una habitación o mejorar el acabado—para observar al instante cómo varía el precio estimado. Del mismo modo, es posible generar “viviendas ficticias” ajustando todas las variables libremente, incluso fuera de los rangos estándar, con el fin de explorar la respuesta del modelo ante combinaciones de características no presentes en el conjunto original.

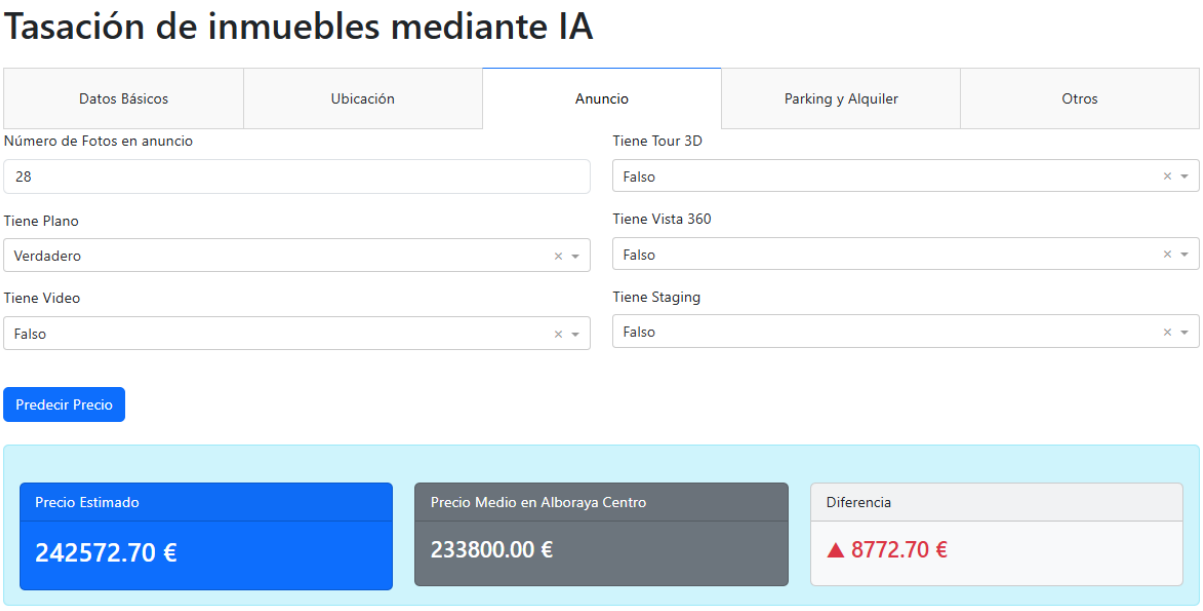
La elección de esta interfaz responde a la necesidad de unir un backend predictivo avanzado con una capa de interacción amigable y transparente. Finalmente, su flexibilidad—capaz de manejar comparaciones simultáneas y simulaciones de distintas configuraciones—la convierte en una herramienta práctica para la toma de decisiones en procesos de compra, venta o reforma de viviendas, cuantificando de forma inmediata el impacto de cada variable sobre el valor final.

Para ilustrar de forma práctica el funcionamiento de la herramienta, a continuación se incluyen unas imágenes del estado inicial en el que se encuentra la interfaz.





Y una vez introducidos los datos reales o ficticios correspondientes al inmueble que se desea predecir, así es como se vería el resultado:



4.1.3. Análisis de características

El análisis de características constituye un pilar esencial de la herramienta, pues es en este apartado donde el usuario puede profundizar en el comportamiento del modelo y extraer información útil para la toma de decisiones.

En primer lugar, la interfaz presenta de forma numérica la **predicción puntual** del precio junto a la **media del distrito** en el que se ubica el inmueble y la **diferencia** numérica entre ambas cifras. Esta comparación inmediata permite al usuario contextualizar su propiedad: saber si su tasación está por encima o por debajo del valor promedio de la zona es el punto de partida para cualquier estrategia de venta o reforma. Sobre esa base, la sección de análisis de características ofrece controles que permiten **modificar dinámicamente** atributos.

Cada ajuste actualiza en tiempo real la estimación de precio y la diferencia con la media distrital, de modo que el usuario puede responder a preguntas tales como "¿Cuánto aumentaría el valor de mi piso si añado un baño extra?" o "¿Qué impacto tiene añadir un ascensor?". Esta capacidad de simulación facilita el cálculo de rentabilidades potenciales, ya que el usuario conoce de antemano el coste aproximado de la reforma y puede compararlo con el incremento de tasación que generaría.

En el apartado de análisis de características dirigido a agencias inmobiliarias, la herramienta va más allá de los atributos físicos del inmueble y se adentra en los **elementos que componen el anuncio**, demostrando cómo cada uno de ellos repercute en la diferencia entre la predicción del modelo y la media distrital. De este modo, al duplicar la configuración del anuncio en la interfaz, el agente puede modificar en tiempo real la **cantidad de fotografías** para comprobar cómo un mayor número de imágenes —que transmite transparencia y confianza— reduce la brecha negativa con respecto al precio medio. De igual forma, la **inclusión de un vídeo de presentación**, que permite una visión más inmersiva del espacio, se refleja inmediatamente en una mejora de la estimación y en un acortamiento de dicha diferencia.

A modo de cierre de este apartado, se incluyen dos casos prácticos en los que se ejemplifica cómo:

1. **Un propietario evalúa una posible reforma.**

   Analizamos un inmueble situado en el distrito de Ciutat Vella cuyo estado "de serie" presenta estas características básicas: se trata de un edificio exterior de 100 m^2, en tercera planta, con cuatro habitaciones y un único baño. Al introducir estos datos en la interfaz, obtenemos los siguientes resultados.

   Seguidamente, para valorar el efecto de añadir un segundo baño a costa de reducir en una unidad el número de dormitorios (pasando a tres habitaciones y dos baños), duplicamos la configuración inicial y modificamos únicamente esos dos parámetros. Lo apreciamos en la segunda estimación.

Pudiendo observar así cómo intervenir sobre una característica específica del inmueble (en este caso, la instalación de un segundo baño) impacta directamente en la valoración, permitiendo al usuario cuantificar de forma inmediata el retorno potencial de una reforma.

2. **Una agencia evalúa un anuncio para posibles mejoras y justificar la subida de precio en el anuncio.**

   En el segundo caso práctico analizamos un inmueble situado en el distrito de L'Eixample con estas características básicas: se trata de un edificio exterior de 68 m^2, en cuarta planta, con dos habitaciones, dos baños y en buen estado general. Además de estos atributos físicos, hemos incorporado al anuncio inicialmente solo **19 fotografías**, sin vídeo de presentación, plano de planta, recorrido 3D, vista 360 ni staging virtual. Al ejecutar la predicción en este escenario, obtenemos:

   A continuación, duplicamos la configuración y mejoramos los recursos multimedia del anuncio: elevamos el número de imágenes a 30, añadimos plano de planta, tour virtual en 3D y vista 360 . Tras estos cambios, la estimación cambia a:

   Este ejemplo podría demostrar cómo, sin tocar las características físicas del inmueble, la simple mejora de los elementos del anuncio —número de fotografías, planos y experiencias virtuales— se traduce en un aumento cuantificable de la valoración y en una reducción significativa de la diferencia frente al precio medio de la zona. De cara a una agencia inmobiliaria, estas cifras sirven para justificar la inversión en contenidos multimedia como parte de una estrategia de marketing orientada a maximizar el valor percibido.

Estos ejemplos permitirán ilustrar de manera concreta las posibilidades del análisis de características y su aplicación directa en escenarios reales de mercado.

No obstante, es importante no confundir esta correlación con una relación causal directa: que los inmuebles caros dispongan de más imágenes no implica que añadir fotos a

un anuncio eleve por sí mismo el valor de mercado de la vivienda. Más bien, las agencias suelen invertir en contenidos multimedia en aquellos anuncios que ya corresponden a propiedades de mayor valor, por lo que la abundancia de imágenes actúa aquí como un proxy de la calidad y del perfil de precio alto del inmueble, y no como un factor que modifique por sí mismo el precio. Por tanto, cualquier estrategia de optimización del anuncio deberá considerar este sesgo: la mejora de los recursos visuales contribuye a reflejar mejor el valor percibido, pero no reemplaza la influencia de las características estructurales y de ubicación en la determinación del precio real.

Capítulo 5

Conclusiones y proyección futura.

5.1. Conclusiones

A lo largo de este trabajo, se ha logrado desarrollar un sistema de predicción de precios inmobiliarios en la ciudad de Valencia, basado en técnicas avanzadas de aprendizaje automático. Utilizando datos extraídos a través de la API de Idealista, se ha conseguido construir un modelo capaz de estimar el precio de venta de propiedades con un alto nivel de precisión, lo que representa una herramienta valiosa para agentes inmobiliarios, compradores, vendedores e inversores.

Uno de los principales retos enfrentados fue la recopilación de datos a partir de una fuente no completamente estructurada, como la API de Idealista. Aunque esta fuente presenta ciertos inconvenientes, como limitaciones en la cobertura geográfica y el tipo de información disponible, se diseñó un proceso automatizado eficiente para obtener los datos necesarios. La limpieza de los mismos fue otro desafío importante, ya que se trataron diversas variables, tanto numéricas como categóricas, que requerían un preprocesamiento adecuado para asegurar la calidad de la información utilizada en la construcción del modelo. El manejo de valores nulos y outliers fue crucial para evitar sesgos en las predicciones y mejorar la capacidad de generalización de los modelos.

El análisis exploratorio de los datos permitió identificar las variables más influyentes en la determinación del precio de los inmuebles, como el tamaño, el número de habitaciones y la existencia de ascensor.

En cuanto a los modelos de aprendizaje automático implementados, se evaluaron diferentes algoritmos, entre los que destacan la regresión lineal, Random Forest, Extreme Gradient Boosting (XGBoost) y redes neuronales (MLP). Tras la comparación de los resultados, el modelo XGBoost fue seleccionado como el más adecuado debido a su excelente rendimiento en todas las métricas evaluadas (MAE, RMSE y R^2), así como su capacidad para manejar datos complejos y no lineales.

El trabajo realizado ha permitido no solo demostrar la viabilidad de aplicar aprendizaje automático en el ámbito inmobiliario, sino también sentar las bases para futuras investigaciones y mejoras.

Por último, se considera que la herramienta desarrollada tiene un gran potencial para ser aplicada en el mercado inmobiliario de Valencia, proporcionando estimaciones precisas del precio de las viviendas que facilitarían la toma de decisiones en diversos contextos. Este trabajo no solo contribuye al avance del aprendizaje automático en este sector, sino que

también representa una aportación valiosa a la optimización de los procesos de compra, venta e inversión inmobiliaria.

5.2. Trabajo futuro

Este proyecto, enfocado en la predicción de precios inmobiliarios en el centro de Valencia, ofrece una base sólida sobre la cual se podrían implementar diversas mejoras y ampliaciones que potenciarían tanto su capacidad de predicción como su aplicabilidad a nivel más amplio. A continuación, se presentan algunos posibles enfoques para escalar y enriquecer el sistema en el futuro.

Expansión geográfica

Actualmente, el modelo se basa únicamente en inmuebles del centro de Valencia, lo que limita su aplicabilidad y generalización a otros contextos urbanos. Un paso importante en el futuro sería ampliar la cobertura geográfica del modelo, abarcando toda la Comunidad Valenciana o incluso toda España. Esta ampliación permitiría evaluar cómo varían los factores que influyen en el precio según la región, como el entorno económico, la demanda en áreas rurales frente a urbanas, o las políticas locales de urbanismo. Para ello, se necesitaría integrar datos de inmuebles de otras ciudades y ajustar el modelo para que tenga en cuenta las peculiaridades de cada zona.

Integración de nuevos portales inmobiliarios

La base de datos actual proviene de un único portal inmobiliario, Idealista, lo que puede limitar la diversidad y cantidad de los datos disponibles. Para enriquecer la calidad del modelo, se podrían incorporar datos de otros portales inmobiliarios como Fotocasa, Habitaclia o Milanuncios. Estos sitios contienen información similar pero con distintos enfoques y públicos, lo que podría aportar una visión más amplia del mercado inmobiliario. La integración de múltiples fuentes también permitiría contrarrestar posibles sesgos de un único portal, aumentando la robustez y generalización del modelo.

Actualización continua de datos

Una de las limitaciones actuales del proyecto es que se realizan llamadas a la API de Idealista para obtener una cantidad fija de inmuebles, lo que significa que la base de datos se queda estática tras la recopilación inicial. En el futuro, sería útil establecer un sistema de **actualización diaria** para realizar llamadas periódicas a la API, de manera que se pueda incorporar en tiempo real la nueva oferta inmobiliaria disponible. Este enfoque permitiría mantener el modelo actualizado con las últimas tendencias del mercado, reflejando con mayor precisión las fluctuaciones de precios y las características de los inmuebles conforme cambian con el tiempo.

Extracción de nuevas variables con LLMs

Actualmente, el modelo ya emplea **Procesamiento de Lenguaje Natural (PLN)** basado en patrones para extraer variables de las descripciones textuales de los inmuebles. Sin embargo, este enfoque tiene limitaciones en cuanto a precisión y capacidad para capturar el significado contextual y semántico más profundo de los textos. Con el avance de los **Modelos de Lenguaje de Gran Escala (LLMs)**, como GPT-4 o BERT, sería posible implementar un sistema mucho más preciso e innovador para el análisis de las descripciones. Los LLMs tienen la capacidad de comprender mejor el contexto y las sutilezas del lenguaje, lo que permitiría extraer características adicionales de las descripciones de los inmuebles, como çerca de transporte público.º "vistas al mar", de manera más eficiente y con una mayor precisión. Esta mejora en el análisis textual enriquecería aún más las variables del modelo, aumentando su capacidad predictiva y permitiendo una comprensión más completa de los factores que afectan al precio de los inmuebles.

Corrección automática de incongruencias en los datos

Uno de los desafíos actuales es la posible **incongruencia de los datos** en variables como "floor", donde pueden aparecer valores poco realistas como -2, o incluso valores faltantes. Un enfoque interesante sería la implementación de **LLMs** para realizar un análisis semántico de las descripciones textuales de los inmuebles y corregir automáticamente estas incongruencias. Por ejemplo, si un inmueble se describe como ubicado en la "segunda planta", el modelo podría automáticamente inferir que el valor de "floor"debería ser 2. Esto mejoraría la calidad de los datos sin necesidad de intervención manual.

Análisis de causalidad de variables

El modelo actual identifica correlaciones significativas entre características del anuncio y el precio, como el número de fotografías, e ilustra en el segundo caso práctico cómo aumentar los recursos multimedia incrementa la valoración estimada. No obstante, conviene diferenciar entre correlación y causalidad: son precisamente las propiedades de mayor precio las que suelen beneficiarse de un mayor número de imágenes, por lo que este factor actúa como proxy del valor y no como una causa que lo eleve. Para superar este sesgo, en trabajos futuros será necesario diseñar un análisis de inferencia causal que incluya la definición de diagramas de relaciones causales. Este enfoque permitirá cuantificar de forma rigurosa el impacto real de los recursos visuales en la valoración de las viviendas, evitando interpretaciones erróneas basadas en correlaciones falsas.

Desarrollo de una interfaz pública y capacidad de carga en masa

En cuanto a la interfaz, la actual versión se presenta como una herramienta de predicción individual para usuarios que introducen características de un inmueble. En el futuro, sería beneficioso transformar esta herramienta en una **página web pública**, lo que permitiría que cualquier usuario, sin necesidad de conocimientos técnicos, pueda acceder a las predicciones. Además, se podría incorporar una funcionalidad que permita cargar archivos CSV con las características de **varios inmuebles a la vez**, de forma que un agente inmobiliario o un desarrollador pueda obtener rápidamente las tasaciones de

múltiples propiedades de una sola vez. Esta funcionalidad masiva mejoraría la eficiencia de la herramienta, permitiendo su uso en aplicaciones más profesionales y a gran escala, como la gestión de grandes carteras de propiedades.

Bibliografía

[1] Idealista. 'evolución del precio de la vivienda en venta en valencia', 2025.

[2] Cadena SER. 'el ayuntamiento de València concede licencias de obras de nueva planta para 1.182 viviendas en 2024', 2024.

[3] La Sexta. 'el número de viviendas construidas cayó entre 2006 y 2012', 2014.

[4] El Economista. 'la vivienda nueva, en peligro de extinción en València: no hay en dos de cada tres barrios', 2024.

[5] Ismael Cirujeda. 'la obra nueva se desploma en València: cae un 30 % la superficie visada para construir viviendas', 2024.

[6] Wikipedia. Burbuja inmobiliaria en españa, 2025.

[7] Carlos Ocaña P. de Tudela y Raymond Torres. 'impacto de la pandemia sobre el sector inmobiliario', 2020.

[8] Banco de España. 'el impacto de la crisis sanitaria del covid-19 sobre el mercado de la vivienda en españa', 2021.

[9] La Vanguardia. 'la previsión del precio de la vivienda para 2025', 2025.

[10] Dialnet. 'la transformación del mercado inmobiliario español', 2025.

[11] Levante-EMV. 'València: El precio de los pisos sube hasta un 24 % en los distritos ", 2025.

[12] Cadena SER. 'los extranjeros acaparan la mayoría de las compras de vivienda en ciutat vella, en València', 2025.

[13] Generalitat Valenciana. 'ayudas a las actuaciones de rehabilitación a nivel de edificio', 2025.

[14] Valencia Plaza. 'los desarrollos urbanísticos residenciales que están por llegar a València', 2025.

[15] Google Cloud. ¿qué es el aprendizaje automático?, 2025. Consultado el 28 de mayo de 2025.

[16] IBM. ¿qué es el aprendizaje automático (ml)?, 2025. Consultado el 28 de mayo de 2025.

[17] Alteryx. Aprendizaje supervisado vs. no supervisado; ¿cuál es mejor?, 2025. Consultado el 28 de mayo de 2025.

[18] Amazon Web Services. ¿qué es el aprendizaje mediante refuerzo?, 2025. Consultado el 28 de mayo de 2025.

[19] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. *Introduction to Linear Regression Analysis*. John Wiley & Sons, 5th edition, 2012.

[20] A. Pértega Díaz and M. Pita Fernández. Modelos de regresión lineal. *Revista de Estadística Aplicada*, 12(3):45–63, 2001.

[21] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[23] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[24] Chris Parr and Greg Howard. Gradient boosting explained. https://explained.ai/gradient-boosting/, s.f. Consultado en mayo de 2025.

[25] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[27] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 1137–1145, San Francisco, CA, 1995. Morgan Kaufmann Publishers Inc.

[28] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Max Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[30] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[31] Guido van Rossum and Fred L. Drake. *The Python Language Reference Manual*. Network Theory Limited, 2009.

[32] Emmanuel Bisong. Google colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 59–64. Springer, 2019.