Universidad Pontificia de Comillas
Facultad de Ciencias Económicas y Empresariales – ICADE
(E2+Business Analytics)

DETECCIÓN DE OPORTUNIDADES EN EL MERCADO INMOBILIARIO: UN ENFOQUE ANALÍTICO PARA LA IDENTIFICACIÓN Y VALORACIÓN DE BIENES INMUEBLES

Alumno: Rafael Loureiro Álvarez
Director: José Portela González

MADRID | Junio 2023

RESUMEN

El mercado inmobiliario es de gran importancia en la vida de las personas, ya que está estrechamente ligado a la adquisición de viviendas y propiedades, y juega un papel crucial en la economía de un país. Sin embargo, es un mercado caracterizado por su constante variabilidad y la influencia de diversos factores que afectan a los precios de los inmuebles.

En los últimos años, se ha producido un gran avance en las herramientas analíticas y las técnicas de modelado de datos, que permiten un mejor entendimiento y predicción de los precios en este mercado. Este avance ha generado la posibilidad de desarrollar modelos capaces de identificar viviendas sobrevaloradas, aquellas cuyo precio excede su valor real y que representan una mala inversión, así como viviendas infravaloradas, que se presentan como grandes oportunidades de inversión.

En este Trabajo de Fin de Grado, se busca abordar esta problemática y desarrollar un modelo predictivo que pueda discernir entre viviendas sobrevaloradas e infravaloradas.

Para lograrlo, se empleará un enfoque basado en técnicas de aprendizaje automático, análisis de datos y detección de outliers, utilizando un conjunto de variables relevantes para la valoración de las propiedades.

Con la implementación de este modelo, se busca contribuir a la mejora del proceso de toma de decisiones en el mercado inmobiliario, fomentando inversiones más seguras y rentables, facilitando así la vida de las personas.

Palabras clave: Inmuebles, inversión, outliers, modelo predictivo, Random Forest, Isolation Forest.

ABSTRACT

The real estate market is of great importance in people's lives, as it is closely linked to the acquisition of homes and properties and plays a crucial role in a country's economy. However, it is a market characterized by constant variability and the influence of various factors that affect property prices.

In recent years, there has been significant progress in analytical tools and data modeling techniques, allowing for a better understanding and prediction of prices in this market. This advancement has opened up the possibility of developing models capable of identifying overvalued homes, those whose price exceeds their actual value and represent a poor investment, as well as undervalued homes, which present great investment opportunities.

This thesis aims to address this issue and develop a predictive model that can distinguish between overvalued and undervalued homes. To achieve this, we will employ an approach based on machine learning techniques, data analysis, and outlier detection, using a set of relevant variables for property valuation.

By implementing this model, the aim is to contribute to the improvement of the decision-making process in the real estate market, promoting safer and more profitable investments, and ultimately enhancing people's lives.

Keywords: Real estate, investment, outliers, predictive model, Random Forest, Isolation Forest.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

El mercado inmobiliario es un sector altamente influenciado por diversos factores económicos, políticos y sociales, lo que genera una amplia variabilidad en los precios de las viviendas a lo largo del tiempo y de los diferentes países. Esta situación puede generar dificultades a la hora de buscar una vivienda adecuada a un precio razonable, ya sea para satisfacer necesidades básicas como para fines comerciales relacionados con la intermediación inmobiliaria.

La capacidad de identificar de manera rápida y eficiente las viviendas que puedan representar oportunidades o descartes inmediatos podría ser de gran ayuda en un momento como el actual, donde existe una gran incertidumbre y descontrol en los precios de los inmuebles. Esto permitiría ahorrar tiempo y dinero a las personas involucradas en el proceso de búsqueda y adquisición de propiedades. Sin embargo, esta identificación no siempre es sencilla.

En los últimos años, se ha producido un gran avance en el desarrollo de diversas herramientas tecnológicas, muchas de las cuales están estrechamente vinculadas con el campo del Machine Learning y el análisis de datos. Estas herramientas han revolucionado la forma en que se aborda la identificación y predicción de variables en diversos ámbitos.

En el presente trabajo se busca lograr desarrollar una herramienta que trate de resolver el problema planteado. Al ser capaz de detectar propiedades sobrevaloradas en comparación con su valor real de acuerdo con el mercado, así como aquellas que presentan un valor inferior al que deberían tener y que representan oportunidades de inversión, se podría obtener un gran método de inversión y solucionar el problema de muchas personas. El mercado inmobiliario es muy amplio y afecta de manera muy directa a la sociedad en general.

1.1 SITUACIÓN DEL MERCADO INMOBILIARIO

Para entender la relevancia que puede llegar a tener un trabajo como este, es importante poner en contexto y describir la situación actual del mercado inmobiliario en España.

En los últimos años, tras la crisis de 2008 y el desplome de precios de la vivienda, se ha podido observar que estos precios, en España, han ido aumentando hasta el día de hoy. Esto se debe, entre otros muchos factores, a un crecimiento positivo del mercado de trabajo y a una disminución en los costes de financiación para operaciones de crédito destinadas a la compra de inmuebles (Alves & Urtasun, 2019).

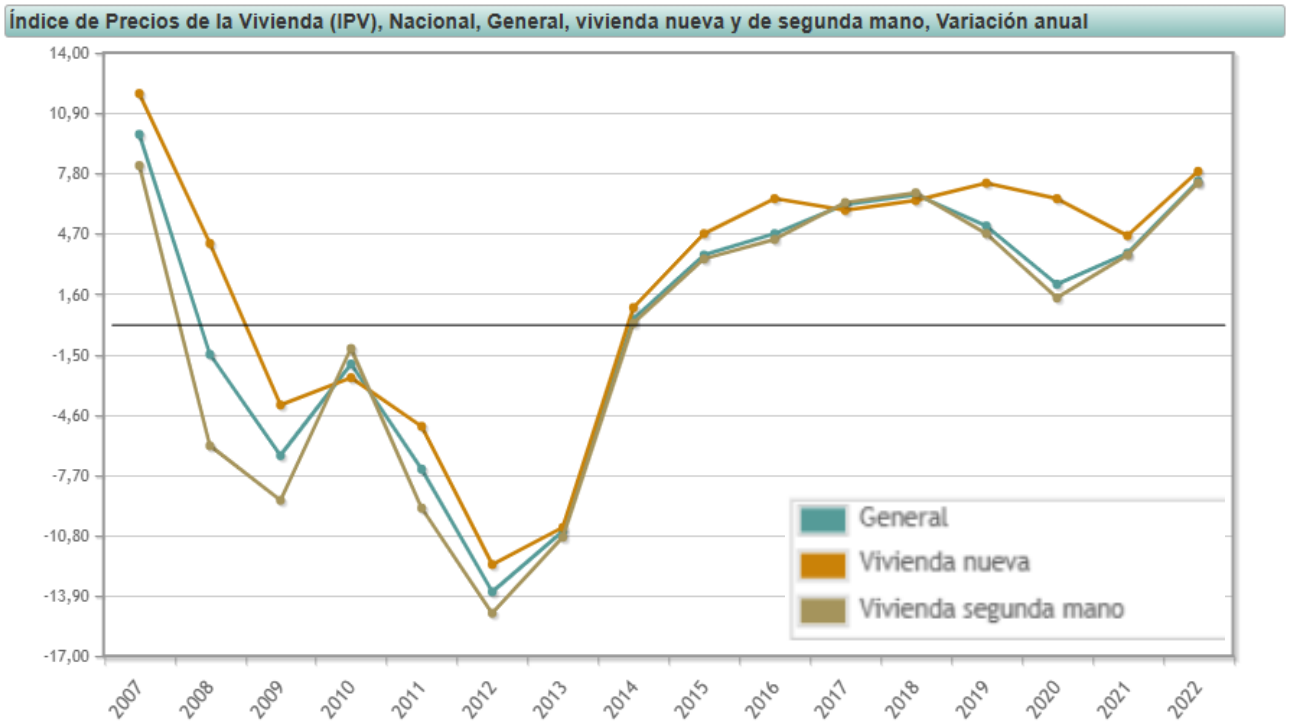

Figura 1. Evolución Índice de precios de la vivienda. *INE.*

Este crecimiento relativamente rápido y constante que se venía produciendo tras la recuperación de la crisis económica del 2008 recién mencionada, se detuvo en 2019 a causa de otro fenómeno que impactó de manera trágica en el mundo entero, la pandemia producida por el Covid-19. Debido al confinamiento y el detenimiento de la economía, tanto la compra, como la subida del precio de las viviendas se detuvieron de golpe.

Sin embargo, todos esos meses sirvieron para que las familias ahorrasen dinero y acumulasen lo necesario para entrar en una hipoteca cuando las cosas volvieron a la normalidad. Junto a esto, es relevante mencionar que, durante este periodo, los intereses estaban muy bajos. En diciembre de 2019, el Euribor 12 meses estaba a -0,27% y en 2020 a -0,48% (Euribor rates, s.f.), lo que hizo que, en cuanto fue posible, la gente comenzase a comprar casas de nuevo y el crecimiento del precio comenzase a subir nuevamente. Esta evolución de la cantidad de compraventas que se acaba de describir desde la crisis del 2008 hasta ahora, pasando por la pandemia, se puede observar claramente en el gráfico de la figura 2, donde se pueden ver agrupadas las cifras de ventas tanto de viviendas usadas como nuevas, además de los totales correspondientes. Se ve perfectamente la caída desde 2008 hasta 2013, donde comienza otra vez la recuperación y el alza de compraventas, el desplome en 2019 y el comienzo de la nueva recuperación a finales del 2020, principios del 2021.

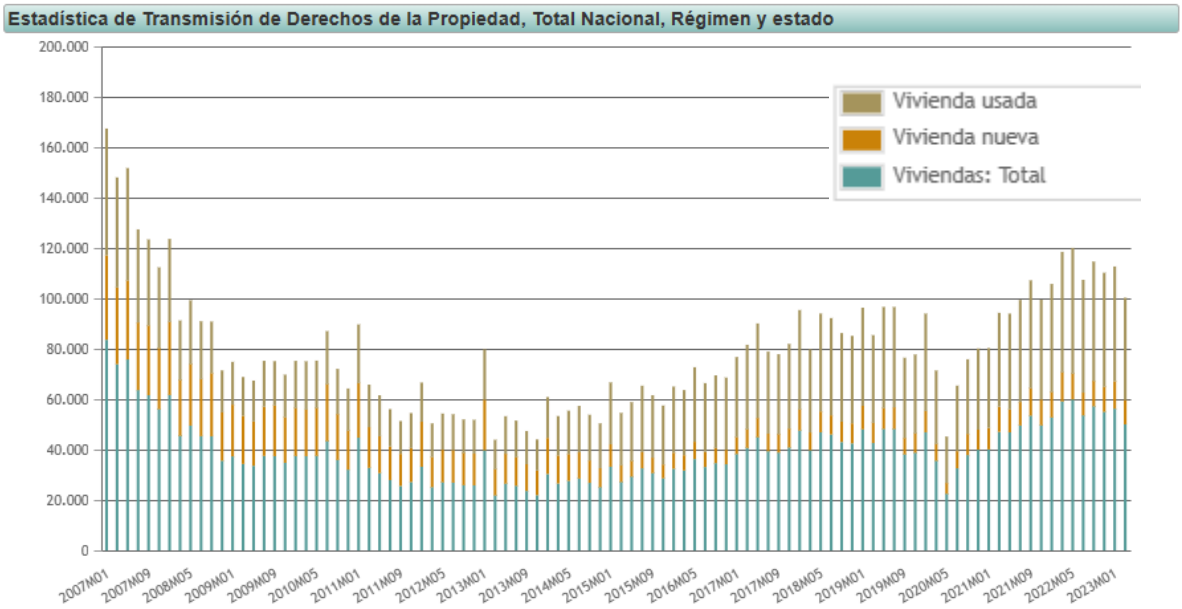

Figura 2.Evolución compraventa de viviendas en España. *INE.*

Sin embargo, poco tiempo después de esta última recuperación tras la pandemia, todo vuelve a cambiar. Con la invasión rusa a Ucrania, se desata una crisis energética a nivel mundial que impacta de manera significativa en la inflación, haciendo que todos los precios comiencen a subir. Como respuesta a esta inflación y tratando de solventar

alguna otra situación adicional, los bancos centrales comienzan a subir los tipos de interés, llegando a los niveles actuales de 3,651 % en abril de 2023 (Euribor rates, s.f.).

Todas estas circunstancias descritas repercuten al mercado actual de la vivienda. La gran inflación y las medidas correctivas de subidas de tipo de interés hacen que se encarezcan las hipotecas. Según el último dato oficial del Banco de España correspondiente al mes de marzo, el coste medio de las hipotecas ha subido del 3,5 al 3,54%, tasa que no se registraba desde marzo del 2012 (Utrera, 2023). Esto hace que la subida de precios que veíamos hasta ahora, en el plano teórico se detenga; pues influirá a que menos familias puedan acceder a la compra de viviendas y bajará la demanda de las mismas. Es por ello, que varios analistas piensan que este 2023, el precio de la vivienda debe bajar, como ejemplo de ello, nos encontramos casos como el del departamento de análisis de Bankinter, que afirma que estiman caídas sobre el precio del 3% en 2023 y del 2% en el 2024 (Departamento Análisis Bankinter, 2023). A su vez, el banco ING, en su reciente informe "*Perspectivas del mercado inmobiliario español 2023"* prevé una caída de precios del 1% (Aparicio, 2023), lo que supone en términos reales una caída del 3,4% (Eseiza, 2023). Por último, Atlas Real Estate Analytics estima una caída del 0,9% en el precio de la vivienda en 2023 (Eseiza, 2023).

En la realidad, si se analizan las últimas cifras de compraventas, según el INE, se puede observar que, en enero, cayeron un 10,16% con respecto al año anterior y en febrero un 6,6% (Instituto Nacional de Estadística; INE, 2022). Para este mes en concreto, en Madrid, cayeron un 15%, siendo esta una de las 4 ciudades del país donde la disminución fue mayor (López, 2023).

A la vez que cae el número de compraventas de viviendas, se podría asumir que, de acuerdo con la ley de la oferta y la demanda, también el precio de la vivienda caería, pues a menor demanda, mayor oferta y bajada de precios hasta llegar al punto de equilibrio. Sin embargo, según Idealista, el precio del metro cuadrado aumentó un 7,3% interanual en el mes de marzo de 2023 (Eseiza, 2023). Este es un dato que puede sorprender y parecer contradictorio, sin embargo, esto se debe a que plataformas como esta, muestran el precio al que se publica la oferta de la vivienda, pero no siempre se corresponde con el precio final al que realmente se compra o vende el inmueble. De

hecho, según informes de notarios de marzo de 2023, los cuales indican el precio al que finalmente se produce la venta, muestran una disminución del precio en un 2,6% (Eseiza, 2023).

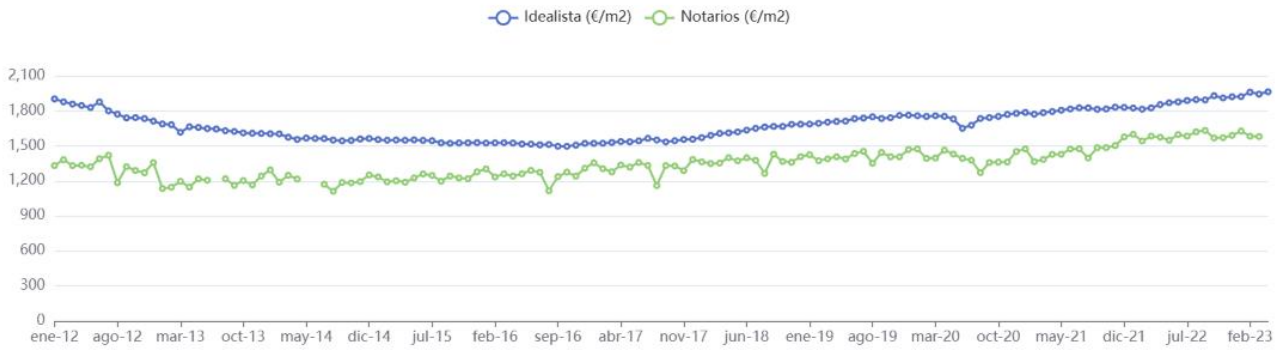

Figura 3. Evolución precio de la vivienda en España. *Eseiza 2023.*

En el gráfico anterior se puede ver la diferencia previamente mencionada entre el precio de oferta en el mercado y el precio de los notarios, que tienen el verdadero precio de venta de los inmuebles.

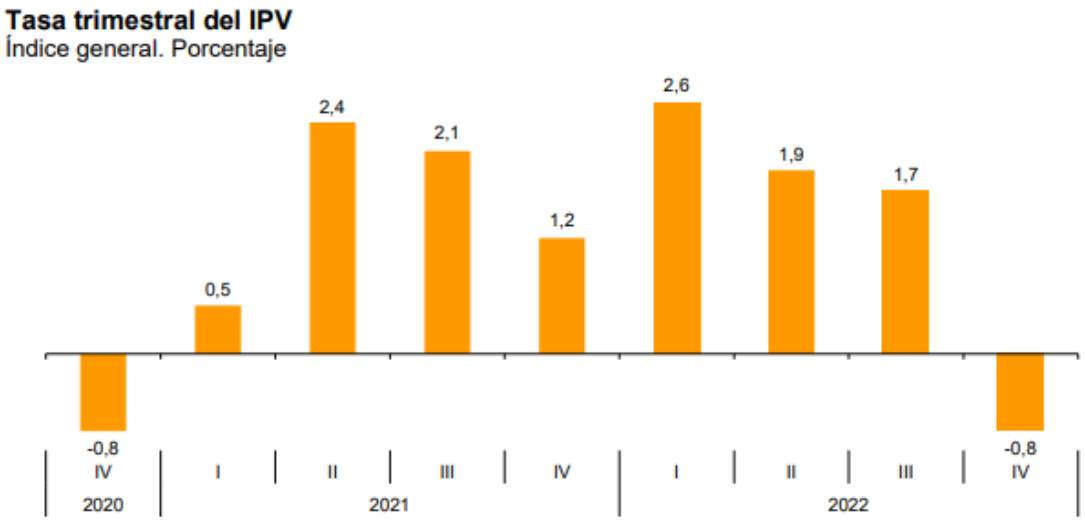

Figura 4. Tasa trimestral del IPV. INE.

En este otro gráfico, se puede observar que la variación trimestral del índice general de precios de la vivienda (IPV) en el cuarto trimestre de 2022 es del –0,8%. Esta es la primera tasa trimestral negativa desde el cuarto trimestre de 2020 (INE, 2022).

Como se puede observar tras este rápido análisis, el mercado inmobiliario es un mercado que está fluctuando continuamente, pues está afectado por un gran número de factores (oferta-demanda, inflación, tipos de interés, ahorro, etc.). Sin embargo, es una actividad fundamental y esencial en cualquier sociedad. De hecho, en los últimos tres años se ha producido una media de 545.517 compraventas al año en España, entre vivienda nueva y usada (INE, 2023). Es por ello por lo que se puede afirmar que, en España, tener la propiedad de una vivienda sigue siendo un valor muy arraigado en la mente colectiva, con un alto porcentaje de personas que poseen un inmueble.

Según Fotocasa, portal inmobiliario especializado en compraventa y alquiler de viviendas en España y Andorra, en 2022 el 77% de los españoles vivía en una vivienda en propiedad, mientras que un 23% vivía en una vivienda de alquiler (López, 2022). A su vez, según el INE, en el 2020, el 69,4 de los españoles vivía en piso, cifra solo superada por Letonia (66,2%) (INE, 2020).

Aunque gran parte de la sociedad española viva en propiedad, son los jóvenes los que más encuentran dificultades para acceder a una vivienda, lo que resulta especialmente frustrante dada su precaria situación laboral y la falta de estabilidad en el empleo, dificultando cada vez más la obtención de una hipoteca. En cifras, según Fotocasa, tan solo el 13% de entre los más jóvenes (18-24 años) son propietarios. Cifra que va incrementando hasta alcanzar el 82% en el grupo de mayor edad (55-75 años) (López, 2022).

Como se puede ver, este es un mercado que afecta de lleno a cualquier sociedad, y que tiene gran relevancia en la vida de las personas.

Dada la gran complejidad y sensibilidad que tiene este sector, no siempre es sencillo poder tener en cuenta toda esta situación del mercado que se acaba de describir, sin embargo, cualquier individuo le gustaría poder comprar una vivienda al mejor precio posible.

1.2 MOTIVACIÓN

Dada esta complejidad del mercado inmobiliario, se podría decir que este TFG surge con el propósito de explorar y aplicar herramientas analíticas y de aprendizaje automático, que como se ha mencionado, se han desarrollado a gran escala en los últimos años, tratando de ofrecer una visión más precisa y eficiente en el proceso de identificación y valoración de propiedades en este mercado. Dado el gran número de personas involucradas en este mercado y el interés generalizado en las sociedades, es fundamental mejorar los procesos relacionados con la compra y venta de viviendas. Esto ayudaría a muchas personas y les facilitaría la vida en muchos momentos clave.

Además, este trabajo busaca aplicar estas herramientas, no solo para satisfacer a las personas en un plano individual a la hora de comprar una vivienda, sino que también en el plano empresarial, muchos negocios que están fundamentados en el sector inmobiliario podrían beneficiarse si contasen con algunas herramientas como las que se describen. Aunque muchas empresas ya trabajan con herramientas y modelos similares al que se quiere plantear, si se logra un modelo muy preciso que detectase esas viviendas infravaloradas, podrían hacer que sus beneficios fuesen aún mayores. Permitiendo alcanzar una mejora en el bienestar de las sociedades y un mayor crecimiento de la economía.

1.3 OBJETIVOS

El presente trabajo parte con unos objetivos clave:

- El objetivo principal del presente TFG es el de crear un modelo que permita identificar qué viviendas pueden estar sobrevaloradas y cuáles infravaloradas de entre un conjunto de observaciones en un periodo de tiempo determinado y ser capaz de identificar alguna “gran oportunidad” o “ganga” del mercado, así como poder descartar de manera rápida una posible inversión.
- El siguiente objetivo es el de identificar entre un conjunto de observaciones, cuáles de estas se podrían considerar como atípicos y ver si alguna de las causas por las cuales han sido clasificadas como tal es el hecho de que su precio no sea el más acorde con el mercado.

- El tercer objetivo es el de encontrar un modelo innovador a la hora de hacer la detección de outliers previamente mencionada.
- Por último, se busca identificar las características de una vivienda sobrevalorada y las de una vivienda infravalorada dentro de un conjunto de datos, logrando así tener unos criterios de referencia a la hora de comparar inmuebles.

1.4 METODOLOGÍA

La metodología de trabajo establecida para lograr estos objetivos consiste, en primer lugar, en hacer una revisión rápida de la literatura, tratando de identificar cuáles son las herramientas más utilizadas a la hora de crear modelos similares al que se pretende construir para lograr el objetivo principal. A continuación, se buscará obtener una base de datos razonablemente buena y completa, con información sobre las viviendas de alguna ciudad del país para poder trabajar sobre ella. Tal como se mencionó anteriormente, Madrid es una de las ciudades donde más ha caído la compraventa de inmuebles, pero que, sin embargo, sigue teniendo los precios más altos. Es por ello por lo que se tratará de encontrar una base de datos con información de esta ciudad. Una vez obtenida, se hará un análisis de los datos, una limpieza y una depuración de toda la información hasta lograr una base de datos óptima para realizar el estudio.

A continuación, se seleccionarán las mejores variables disponibles para el modelo y mediante un mecanismo de detección de atípicos se separará la base de datos en dos, una primera que serán los datos clasificados como atípicos o outliers y una segunda, que serán los datos “limpios”, sin estos otros atípicos previamente mencionados.

Con estos segundos datos, se probarán varios modelos de regresión para intentar encontrar el que mejor prediga el precio de la vivienda. Una vez encontrado cuál de ellos predice mejor el precio de compra de la vivienda, permitirá ver cuáles de esas casas están siendo vendidas a un precio mayor de lo que el modelo predice y cuáles a un precio menor. Sin embargo, el trabajo no acaba aquí, sino que, a continuación, lo que se busca es aplicar el modelo de predicción entrenado que mejor predice, en la base de datos de los outliers. De esta manera se podrá ver, cuáles de estas viviendas que fueron identificadas como “atípicas”, están sobrevaloradas en precio y cuáles están

infravaloradas ya que el motivo de esta identificación se puede deber a una inconsistencia en el precio de valoración. Será aquí donde quizás se encuentren grandes oportunidades de inversión. Y no solo eso, sino que gracias a que el modelo también detectará cuáles considera que están sobrevaloradas, evitará en cierto modo poder cometer el error de comprar una vivienda que está a un precio superior de lo que el mercado considera.

2. ESTADO DEL ARTE

2.1. PREDICCIÓN DE PRECIOS EN EL MERCADO INMOBILIARIO

En los últimos años, con el desarrollo y evolución de las herramientas de Machine Learning, así como por la relevancia e impacto social del tema, se han desarrollado varios trabajos en los que se busca predecir diferentes variables relacionadas con la vivienda y el sector inmobiliario. En todos ellos se pueden encontrar diferentes enfoques, algunos de ellos se centran en buscar la localización optima de una vivienda, otros cual es el precio de alquiler más apropiado según la zona, otros analizan la evolución temporal de los precios, etc.

Para lograr esos enfoques se han aplicado diferentes modelos y diferentes herramientas tratando de alcanzar aquel que mejor encaja con los datos y la problemática que se quiere resolver en cada situación.

Sawant et al. (2018) en su trabajo llamado "*Comprehensive Analysis of Housing Price Prediction in Pune using Multi-Featured Random Forest Approach*" analizan el mercado de la vivienda en India utilizando árboles de decisión y Random Forest. En ese mismo estudio también mencionan cómo, trabajos similares como el de Yu y Jiafu Wu, que tratan de predecir los precios de las viviendas a partir de variables explicativas, utilizan varias técnicas de regresión como SVM, regresión logística, Lasso o SVR con Kernel gaussiano. También mencionan a Nissan Pow, Emil Janulewicz y Lui que con el mismo objetivo logran una muy buena predicción en su estudio usando un ensamble de KNN y Random Forest. Por último, también se hace referencia al trabajo de Jiao Yang Wu, que para predecir los precios de las viviendas usa SVR haciendo una reducción de dimensionalidad con análisis de componentes principales (PCA) (Sawant et al., 2022).

También se han identificado otros trabajos como el de Baldominos et al. (2018) en los que tratan de identificar oportunidades en el mercado inmobiliario utilizando herramientas de Machine Learning. En este, las herramientas principales que se usan son Support Vector Regression (SVR); KNN, donde afirma que "Este método es interesante ya que considerará activos similares al que queremos predecir" (Baldominos et al., 2018); ensambles de árboles de regresión y Multi-layer Perceptron (MLP) que es un tipo de red neuronal que se basa en el concepto del perceptrón.

Martínez (2019), también trata de hacer un acercamiento al precio de la vivienda mediante el uso de redes neuronales. Una técnica que implica una mayor complejidad pero que en ciertas situaciones logra muy buenos resultados.

Por último, también se ha identificado el trabajo "*Using Machine Learning Algorithms for Predicting Real Estate Values in Tourism Centers*" realizado por Alkan et al. (2022) donde hacen una revisión de los principales modelos utilizados en este tipo de predicciones y a continuación utiliza el Support Vector Regression, el K Nearest Neighbors y el Random Forest Regression para hacer su propio estudio y obtener las predicciones de los precios de las viviendas de su base de datos. En la revisión de los principales algoritmos utilizados, se pueden ver que los profesionales de este campo utilizan principalmente Artificial Neural Networks, Support Vector Machines, K-Nearest Neighbors y Random Forest. También se menciona el trabajo de Ravikumar (2017), que además de los algoritmos mencionados, incluye el Gradient Boosted como una herramienta relevante y que puede ayudar a obtener grandes resultados en la predicción.

Según afirman varios profesores de universidad de la India, en su trabajo "Prediction of House Price Using XGBoost Regression Algorithm", "XGBoost es uno de los mejores algoritmos de aprendizaje supervisado…" (Avanija et al., 2021) y es utilizado por ellos para entrenar su modelo de predicción que busca ayudar a compradores y vendedores a establecer un precio justo por la vivienda.

Todos estos trabajos mencionados, tratan de aplicar un modelo de regresión a una base de datos tratando de predecir el valor de una determinada variable o clasificar una serie de observaciones. Se ha podido ver, que son varias las aproximaciones que se hacen y

los algoritmos utilizados. Sin embargo, ninguno hasta la fecha ha tratado de realizar un enfoque como el que se propone en este trabajo, en el que se aplique un modelo de regresión a unos datos previamente clasificados como outliers intentando identificar oportunidades entre estos datos. Hasta ahora, la detección de outliers únicamente había sido utilizada para limpiar los datos y trabajar sin estos atípicos. Es por ello, por lo que se puede decir que este trabajo trata de aportar un enfoque diferente y ver si se obtienen resultados interesantes prediciendo entre valores atípicos en lugar de solamente hacerlo entre los datos limpios de una base de datos.

2.2. MODELOS DE MACHINE LEARNING

Tal y como se acaba de analizar, existe un gran número de algoritmos de Machine Learning para predecir variables, cada uno de ellos se ajusta mejor a una serie de datos y según lo que se esté buscando en cada momento, pueden ser más o menos apropiados de utilizar.

En este trabajo, tras haber considerado varios de los modelos de predicción, se han decidido aplicar cuatro de ellos: Regresión lineal múltiple, Random Forest, K-nearest neighbors (KNN) y XGBoost. Algunos algoritmos conllevan mayor complejidad que otros, pero estos cuatro han sido seleccionados debido a que son generalmente preferidos por su rendimiento superior en comparación con otros algoritmos de aprendizaje automático para fines de regresión, especialmente el KNN y el Random Forest (Alkan et al., 2022).

A continuación, se describe de manera un poco más detallada en qué consiste cada uno de estos cuatro modelos seleccionados.

2.2.1. REGRESIÓN LINEAL

El primer modelo entrenado es el más básico de todos, se trata de una técnica utilizada para examinar la relación entre una variable dependiente y múltiples variables independientes. Este enfoque permite predecir y comprender cómo las variables

independientes afectan la variable dependiente. Este modelo se basa en la ecuación siguiente (Amat, 2016):

$$Y_i = (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_n X_{ni}) + e_i$$

- β_0 representa el valor de la variable dependiente cuando el resto son cero.
- β_n son los coeficientes parciales de regresión que indican el efecto que tiene el aumento de una unidad de la variable independiente Xi cuando las otras se mantienen constantes.
- e_i es la perturbación aleatoria, error de producción ocasionado por una serie de variables que no son controladas.

En un gran número de ocasiones este algoritmo se utiliza para hacer también un modelo explicativo e intentar ver cómo cada una de las variables influyen sobre la variable independiente.

2.2.2. RANDOM FOREST

Random Forest es una técnica de aprendizaje supervisado que se basa en la creación de múltiples árboles de decisión a partir de unos datos de entrenamiento. El modelo busca combinar los resultados obtenidos de cada árbol, para lograr un modelo único más robusto. Se podría decir que Random Forest es un caso particular de "ENSEMBLE" de modelos mediante "BAGGING" (Martínez, 2021). Las ideas principales de este modelo son: En primer lugar, se extraen muchas muestras aleatorias con reemplazamiento, es decir, mediante "Bootstrap" a partir de los datos de entrenamiento para crear múltiples árboles de decisión. Cada uno de estos árboles está formado por un conjunto de variables predictoras "m" de forma que esta "m" sea menor que el número total de variables predictoras existentes (Espinosa-Zúñiga, 2020). Haciendo esto, se consigue mitigar el predominio de alguna variable dominante o influyente en los primeros "splits" de los árboles, logrando que los árboles no salgan muy similares y se descorrelacionen (Martínez, 2021). A continuación, se deja crecer cada árbol hasta alcanzar su máxima extensión. Por último, como se ajustan varios árboles obteniendo un bosque de "árboles", se combinan los resultados de los árboles individuales para hacer la

predicción. En el caso de una regresión, mediante el promedio de las predicciones de los árboles, obteniendo así mejores resultados. Las observaciones que no se hayan utilizado en los árboles, son las que se van a utilizar para validar el modelo.

La idea principal detrás de un modelo de Random Forest es que, al combinar múltiples árboles de decisión, se reducen los sesgos y las varianzas inherentes a cada árbol individual. Esto conduce a un modelo más robusto y generalizable, capaz de manejar características no lineales y de alta dimensionalidad.

Además, los modelos de Random Forest también proporcionan información sobre la importancia de las características utilizadas en la predicción, lo que permite evaluar la relevancia de cada variable en el modelo (Breiman, 2021). Como comenta Espinosa-Zúñiga (2020), algunas ventajas de este modelo son: puede utilizarse para clasificación y regresión, es simple de entrenar, es eficiente y usa una de las técnicas más certeras en grandes bases de datos. Como desventajas tiene que sus resultados tienen difícil visualización gráfica, en presencia de ruido puede sobre ajustar algunos datos y se tiene un control muy limitado sobre las acciones del modelo (Espinosa-Zúñiga, 2020). Este modelo es considerado como uno de los favoritos y genera altas expectativas en términos de lograr los mejores resultados al aplicarlo a los datos de la base de datos.

2.2.3. XGBOOST

El modelo de XGBoost proviene de la abreviación de "*Extreme Gradient Boosting*". Es un algoritmo de aprendizaje que se basa en una implementación optimizada de la técnica de boosting en los árboles de decisión. Se podría decir que es similar a un Random Forest, sin embargo, en el XGBoost, se define la extensión de los árboles mientras que en el Random Forest crecen hasta su máxima extensión (Espinosa-Zúñiga, 2020). Como afirman Chen y Guestrin (2016), el XGBoost consiste en un ensamblado secuencial de árboles de decisión. Estos árboles se agregan de manera secuencial aprendiendo de los resultados obtenidos por los árboles anteriores y corrigiendo el error que producen los mismos, hasta que ya no se pueda corregir más el error (Chen & Guestrin, 2016). Este proceso descrito es una simplificación del, como bien indica su nombre, gradient boosting. De una forma más teórica, este consiste en que partiendo de una función de

pérdida como el error cuadrático medio (MSE) y un aprendiz débil, el algoritmo trata de encontrar un modelo aditivo que minimice la función de pérdida. El algoritmo, en un primer momento se inicia con la mejor estimación, luego calcula el "gradiente" o residuo, y, por último, se ajusta un modelo a los residuos para minimizar la función de pérdida. Este modelo se agrega al modelo anterior y el proceso continúa durante un número predefinido de iteraciones (Kuhn & Johnson, 2013). En definitiva, el boosting busca mejorar iterativamente un modelo combinando varios modelos débiles de manera aditiva, donde cada nuevo modelo se enfoca en los errores cometidos por los modelos anteriores.

Las principales ventajas son que puede manejar grandes bases de datos compuestas por varias variables gracias a sus varios métodos de optimización, maneja valores perdidos, proporciona resultados muy precisos y la carga computacional no es elevada por lo que la velocidad es alta (Espinosa-Zúñiga, 2020). Además, incluye técnicas de penalización para evitar el sobreajuste.

2.2.4. KNN

K-nearest neighbors (KNN) es un algoritmo de aprendizaje supervisado sencillo pero productivo. Se usa para regresión y clasificación. Aunque sí que es cierto que en el mundo del Machine Learning, es más usado en modelos de clasificación. Como modelo de regresión, busca predecir un valor numérico en función de los datos de entrenamiento etiquetados, donde los puntos de datos se caracterizan en diferentes clases, de forma que permita predecir la clase y el valor de los datos no etiquetados (Taunk, De, Verma & Swetapadam, 2019).

En primer lugar, al algoritmo se le facilitan un conjunto de datos de entrenamiento etiquetados, a continuación, para predecir el valor de una determinada nueva observación no etiquetada, KNN calcula las distancias entre el nuevo punto y cada uno de los puntos de entrenamiento quedándose con los K valores más cercanos. Para escoger cuáles son los más cercanos, este algoritmo utiliza varias medidas, entre ellas están la distancia euclídea, que es la más usada ya que es idónea cuando se están utilizando variables cuantitativas, pues, en el fondo, lo que está haciendo es calcular una especie de hipotenusa con el teorema de Pitágoras. Otras medidas utilizadas son la

distancia Manhattan o la distancia de Jaccard. Es muy importante a la hora de calcular las distancias que las variables estén estandarizadas, esto consiste en igualar el rango y la escala entre las variables para que no se distorsionen las medidas de distancia (Martínez, 2021).

Por último, para asignar el valor predicho del punto se calcula el valor medio de esos k puntos más cercanos seleccionados. La decisión del valor que se le va a asignar al hiperparámetro K es bastante relevante, ya que de ello dependerá el valor de la predicción (Taunk, De, Verma & Swetapadam, 2019). Si este hiperparámetro es muy pequeño, el algoritmo podría tener mucha varianza y poco sesgo, es decir, estaría haciendo overfitting. A su vez, si K es muy grande, el algoritmo tendrá mucho sesgo, poca varianza y hará underfitting. Normalmente, lo que se hace y lo que se ha hecho en el modelo del presente trabajo ha sido probar varios valores de K y quedarse con el que proporciona un resultado más efectivo.

El KNN es un algoritmo simple, comprensible y escalable, con gran facilidad de interpretación y un tiempo de cálculo bajo. Además, tiene un poder predictivo muy alto, lo que lo hace efectivo y eficiente, especialmente en grandes conjuntos de entrenamiento (Taunk, De, Verma & Swetapadam, 2019).

2.3. MODELOS DE DETECCIÓN DE OUTLIERS

Uno de los pasos en la metodología del trabajo es la detección de aquellas observaciones que se podrían considerar como atípicos. Un outlier o atípico es una observación que se desvía mucho de las otras observaciones y despierta sospechas de que se generó por un mecanismo diferente (Hawkins, 1980). En otras palabras, es una observación que no es consistente con las demás. La detección de outliers es un tema importante en el análisis de datos y estadísticas, ya que los outliers pueden afectar significativamente a los resultados de los análisis posteriores.

En términos más técnicos, la detección de outliers se enfoca en encontrar valores que se encuentran fuera de la distribución normal de los datos, lo que puede ser identificado mediante diferentes métodos y modelos. Estos, pueden ser univariantes o multivariantes (Muñoz, 2013). Los univariantes, toman los “valores extremos” como atípicos. Sin embargo, los multivariantes consideran los atípicos como tales no por el

valor que toman para una determinada variable, sino el que toman varias variables simultáneamente. Es por ello por lo que son más difíciles de detectar (Muñoz, 2013).

De acuerdo con el informe elaborado por Fei Tony Liu, Kai Ming Ting y Zhi-Hua Zhou (2008), gran parte de los enfoques existentes que se basan en modelos de detección de outliers siguen una estrategia común. Primero, elaboran un perfil de instancias normales para después identificar las instancias que no se ajustan al perfil normal previamente elaborado como outliers. En otras palabras, el detector de anomalías es optimizado para perfilar instancias normales, no optimizado para detectar anomalías. La mayor parte de estos algoritmos se basan en medidas de distancia, similitud o densidad del conjunto de datos, que suele tener un coste computacional muy alto, algunos ejemplos de estos pueden ser el Z-Score, análisis del rango intercuartílico, método de desviación típica, distancia de mahalanobis, etc. Sin embargo, para este trabajo, se ha optado por introducir otro método de detección de outliers algo diferente, que, a diferencia de estos otros, busca aislar específicamente las anomalías, sin depender de suposiciones sobre la distribución de los datos. Este es el modelo de Isolation Forest.

Isolation Forest es un método no supervisado desarrollado por Fei Tony Liu, Kai Ming Ting y Zhi-Hua Zhouen en 2008 para identificar anomalías (outliers) cuando los datos no están etiquetados, es decir, cuando no se conoce la clasificación real (anomalía - no anomalía) de las observaciones (Amat, 2020).

El proceso de construcción de un modelo de Isolation Forest está basado en el algoritmo de regresión y clasificación Random Forest ya que se forma combinando distintos árboles binarios llamados *isolation trees*. El algoritmo de Isolation Forest comienza dividiendo aleatoriamente los datos en dos partes: una muestra de entrenamiento y una muestra de prueba. A continuación, se crea un conjunto de árboles de decisión, cada uno de los cuales se construye de manera recursiva dividiendo aleatoriamente los datos en dos subconjuntos. El árbol se detiene cuando se llega a un punto en el que todas las observaciones quedan aisladas en un nodo terminal (Amat, 2020). La idea principal detrás de la técnica de aislamiento es que los puntos de datos anómalos son menos frecuentes y, por lo tanto, requieren menos particiones para ser aislados. Por lo tanto, los outliers que se separan más rápidamente a lo largo del proceso de construcción del árbol, tendrán una profundidad menor en los árboles de decisión. Más en concreto, a la

hora de construir los árboles, tal y como bien explica Amat, los pasos que sigue son los siguientes (Amat, 2020):

1. Se crea un nodo raíz que contiene las N observaciones de entrenamiento con sus respectivas variables.
2. Se selecciona aleatoriamente una variable "i" y un valor aleatorio "a" que va a estar entre el mínimo y el máximo de la variable seleccionada.
3. Crea dos nuevos nodos según si Xi≤a o Xi>a.
4. A continuación, se irán repitiendo los pasos 2 y 3, pero cada vez de forma más acotada, siendo el nuevo máximo o mínimo el punto de corte de la rama creada. Todo ello, hasta lograr tener todas las observaciones de forma individualmente aislada en nodos terminales.

En la práctica, si no se entra en términos matemáticos, se puede decir que este modelo funciona de tal manera que, una vez que se han construido los árboles, se utiliza la muestra de prueba para detectar los puntos de datos anómalos. Para un punto de datos dado, se calcula el número de árboles que han sido necesarios para aislarlo (es decir, el número de divisiones que se han necesitado en los árboles para separar el punto de datos de los demás). Si este número es menor que un umbral predefinido, se considera que el punto de datos es anómalo.

Si por el contrario se profundiza un poco más en la formulación matemática, se podría decir que Isolation Forest, tras la creación de los árboles, se basa en la función:

$$s(x,n) = 2^{\frac{-E\left(h(x_i)\right)}{c(n)}}$$

donde S es el score o puntuación que se le da a cada una de las $x_i$observaciones. $E\left(h(x_i)\right)$ es el promedio de longitud de las ramas y $c(n)$ es la longitud promedio de una búsqueda no exitosa en un árbol binario (Liu, Ting & Zhou, 2008). Por lo tanto, al final de todo, se tendrá para cada observación una puntuación que cuanto más cercana a 1, indicará que existen más probabilidades de que sea un atípico, pues se ha quedado mucho más cerca de la raíz del árbol.

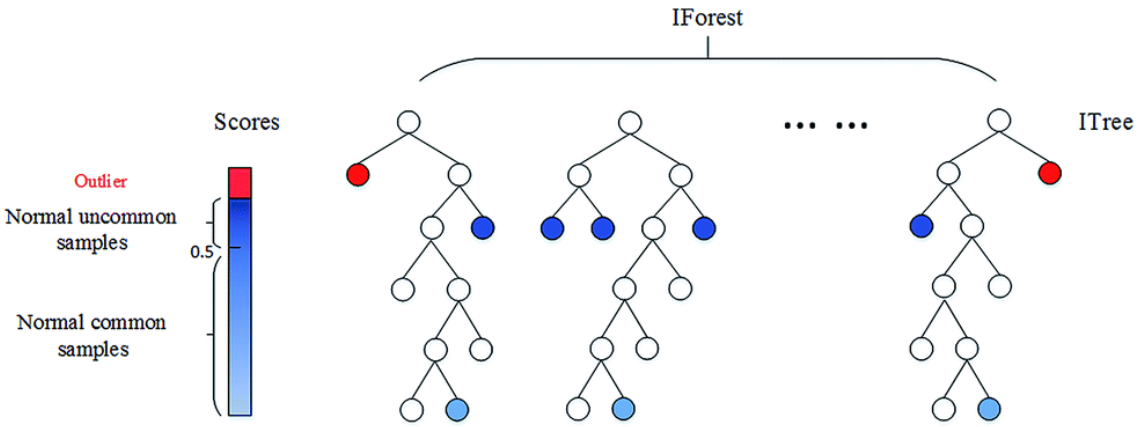

Figura 5. Gráfico resumen Isolation Forest. *Analytical Methods.*

En la figura anterior, se puede observar una representación de cómo funcionaría el modelo de Isolation Forest donde se ve que crea diferentes árboles (ITree) formando así el IForest. Las observaciones que están más arriba en los árboles obtienen un mayor score y, por lo tanto, tienen mayor probabilidad de ser outliers.

Este método de aislamiento funciona mejor cuando el tamaño de las muestras para crear cada uno de los árboles se mantiene pequeño. Muestras grandes, reducen la habilidad del Isolation Forest de separar las anomalías ya que observaciones normales pueden interferir en el proceso de aislamiento.

En conclusión, se puede decir que el Isolation Forest es muy eficiente detectando anomalías cuando existen grandes volúmenes de datos. Aprovechándose de que por naturaleza las anomalías son menos y más diferentes, los *isolation trees* aíslan las anomalías que están más cerca de la raíz del árbol. Esta característica es muy relevante para este modelo, ya que permite construir modelos parciales y utilizar solo una porción de todos los datos de entrenamiento utilizados para construir un modelo efectivo. Incluso, funciona muy bien cuando no hay anomalías en el training set (Liu, Ting & Zhou, 2009). La gran ventaja de este algoritmo en relación con otros métodos se basa en que este, evita el uso de medidas de distancia, similitud o densidad del conjunto de datos, lo cual suele tener un coste computacional muy alto. Isolation Forest tiene una complejidad que crece linealmente gracias a las bondades del sub-muestreo, computa árboles por subpartes del conjunto de datos. Así, tiene la capacidad de escalar en datasets grandes y con muchas variables irrelevantes (Forteza, 2019).

3. CASO DE ESTUDIO

3.1. DESCRIPCIÓN DE LA MUESTRA SELECCIONADA

El dataset con el que he realizado el estudio ha sido "houses_Madrid.csv", extraído del repositorio de Kaggle *"Madrid real estate market. Real estate listings in Madrid crawled from popular internet portals"* (Kaggle, s.f.). Para obtener los datos de estudio, se barajó la opción de hacer webscraping a páginas como Idealista o Fotocasa ya que son los principales portales inmobiliarios del país. Esto permitiría que los datos a analizar fueran de fechas mucho más recientes. Sin embargo, en los últimos años, Idealista ha introducido varios elementos que dificultan el hacer esta tarea, como un sistema de análisis de logs que detecta scraping y banea las IPs. Al descubrir esto fue cuando se decidió descartar este método y tratar de encontrar otra alternativa.

Realmente, se optó por centrarse en trabajar sobre los datos y el modelo en lugar de dedicar demasiado tiempo a la obtención y organización de los datos, que en general suele consumir gran parte del tiempo. Kaggle es una plataforma web que reúne la comunidad de Data Science más grande del mundo, cuenta con más de 536 mil miembros activos en 190 países y recibe más de 150 mil publicaciones cada mes (DataScientest, 2021), cuenta con una gran cantidad de datos publicados y de calidad, por lo que sacar la base de datos de aquí pareció una buena opción. El dataset escogido es bastante interesante ya que es de 2020 y está compuesto inicialmente por 21.709 observaciones y 47 variables, entre estos datos también hay que mencionar que se encuentran, valores faltantes, valores que no tienen sentido, etc. Por lo tanto, uno de los primeros pasos necesarios fue la limpieza de los datos y la selección de aquellas variables consideradas verdaderamente relevantes para el proyecto. Lo que cambió el número de observaciones finales que quedan para trabar y entrenar los modelos.

Durante la limpieza, lo primero que se hizo fue identificar el número de valores faltantes que había para cada una de las variables del dataset. Una vez calculados, se optó por retener únicamente aquellas variables que no presentaban muchos valores faltantes y eliminar estos últimos de forma que se obtuviese una base de datos con la mayor cantidad de observaciones limpias y completas posible. Tras la realización de este análisis, se decidió conservar solamente 12 variables totalmente limpias, correspondientes a un total de 18.688 observaciones.

3.2. ANÁLISIS EXPLORATORIO

A continuación, se puede ver una tabla resumen de todas las variables analizadas.

3.2.1. TABLA DESCRIPCIÓN DE VARIABLES

Variable	Explicación	Medición	Tipo
"sq_mt_built"	Número de metros cuadrados construidos	Metros cuadrados	Numérica
"n_rooms"	Número de habitaciones	Unidades	Numérica
"n_bathrooms"	Número de baños	Unidades	Numérica
"buy_price"	Precio de compra del inmueble	Euros	Numérica
"is_renewal_needed"	Si es necesaria una renovación en el inmueble	0: No es necesaria 1: Sí es necesaria	Numérica (dicotómica)
"energy_certificate"	Tipo de certificado energético	10 posibilidades: "A"; "B"; "C"; "D"; "E"; "F"; "G"; "En trámite"; "Inmueble exento"; "No indicado"	Categórica
"has_parking"	Si el inmueble cuenta con aparcamiento propio	0: No cuenta con parking 1: Sí cuenta con parking	Numérica (dicotómica)
"house_type_id"	Tipo de inmueble	4 opciones: "HouseType1: Pisos"; "HouseType2: Casa o Chalet"; "HouseType4: Dúplex"; "HouseType5: Áticos"	Categórica
"rent_price"	Precio de alquiler del inmueble	Euros	Numérica
"has_pool"	Si tiene piscina	0: No tiene piscina 1: Sí tiene piscina	Numérica (dicotómica)
"has_terrace"	Si tiene terraza	0: No tiene terraza 1: Sí tiene terraza	Numérica (dicotómica)
"has_garden"	Si tiene jardín	0: No tiene jardín 1: Sí tiene jardín	Numérica (dicotómica)

Tabla 1. Tabla variables empleadas. *Elaboración propia*.

Como se irá describiendo, a muchas de estas variables, ha sido necesario aplicar diferentes transformaciones a la hora de prepararlas para su utilización en los diferentes modelos de regresión diseñados para predecir el precio de compra de la vivienda en Madrid.

Por ejemplo, las variables numéricas se han dejado tal cual venían en la base de datos, sin embargo, de cara a un correcto funcionamiento de los modelos de regresión, fue necesario transformar las variables categóricas en variables factor, como una alternativa cómoda a la realización del "One-hot encoding", también muy comúnmente utilizado en estas situaciones.

3.2.2. ANÁLISIS DESCRIPTIVO INDIVIDUAL DE LAS VARIABLES

1. **Precio de compra (*"buy_price"*)**

   El precio de compra de los inmuebles va a ser la variable dependiente que se utilizará en los modelos de regresión. Como se ha mencionado, se busca ser capaces de predecir el precio de esta variable de la mejor forma posible para luego lograr el objetivo de analizar qué observaciones están sobrevaloradas y cuáles están infravaloradas.

Mínimo	36.000€
Máximo	1.380.000€
Mediana	325.000€
Moda	550.000€
Media	422.991€
Desviación típica	298.477,3€

Tabla 2. Resumen variable "*buy_price*". *Elaboración propia.*

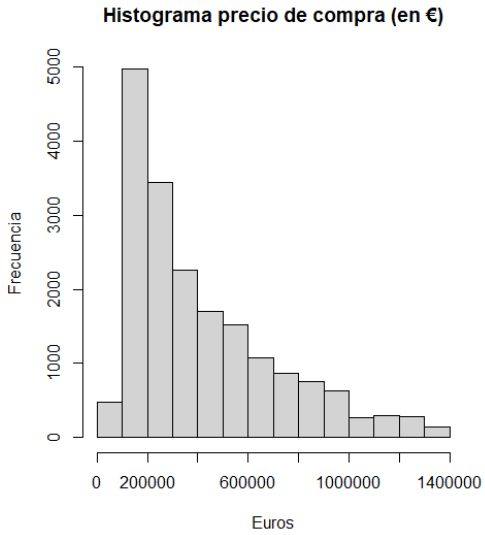

Figura 6. Histograma variable "*buy_price*". *Elaboración propia.*

Como se puede comprobar, la vivienda más cara que se identifica tiene un valor de compra de 1.380.000€ y la más barata de 36.000€. Si bien es cierto que, en la base de datos original, el valor máximo era de 8.800.000€ pero, tras el ajuste y limpieza de los datos, antes de quitar outliers, queda como se muestra en la tabla. La mediana de precios está en los 325.000€ y la media en 422.991€.

Si analizamos el histograma se podría decir que existe cierta asimetría a la derecha. Esta información es importante y se debe tener en cuenta para valorar si se debe introducir una transformación logarítmica en los modelos de regresión

que se hagan posteriormente, comprimiendo así los valores más altos y ampliando los más bajos. De esta manera se podría ayudar a reducir la influencia de los valores extremadamente altos a la vez que hacer que la relación entre las variables sea más lineal.

2. Metros cuadrados construidos (*"sq_mt_built"*)

La variable metros cuadrados construidos, indica la superficie del inmueble y es una de las variables más relevantes y que más impacto tiene a la hora de establecer el precio de compra de la vivienda. Entre los inmuebles que tenemos en la base de datos, se observa que existe una gran dispersión en cuanto a tamaños. Las viviendas más grandes pueden llegar a los 847 metros cuadrados y las más pequeñas a los 16 metros cuadrados. Sin embargo, si se analiza el diagrama de caja o los datos de la tabla, se puede ver que la mediana está en torno a los 92 metros cuadrados y el 75% de los inmuebles tiene menos de 134 metros cuadrados. Las viviendas que más se repiten son las de 70 metros cuadrados. Con esto se puede concluir que muchos de ellos son datos bastante extraordinarios.

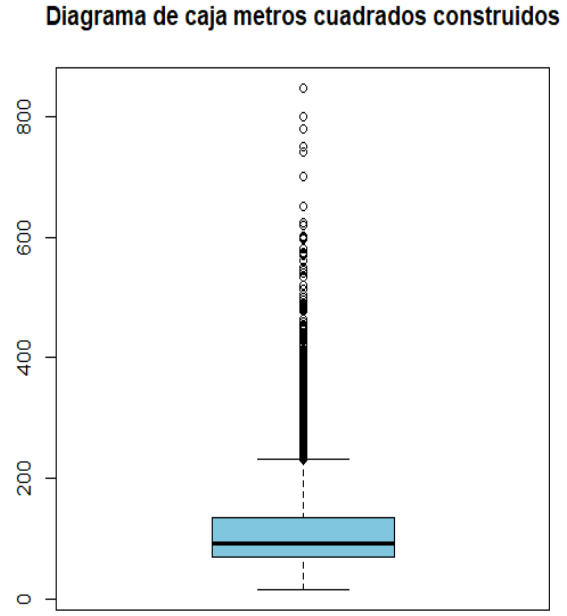

Figura 7. Distribución variable m2. *Elaboración propia.*

Mínimo	16 m2
Máximo	847 m2
Mediana	92 m2
Moda	70 m2
Media	113,1 m2
Desviación típica	71,30 m2

Tabla 3. Resumen variable m2. *Elaboración propia.*

También se puede decir que, a mayor número de metros cuadrados, mayor es el precio de compra, tal y como se muestra en el gráfico de la figura 8. En otras palabras, existe una correlación positiva entre los metros cuadrados y el precio de compra de los inmuebles.

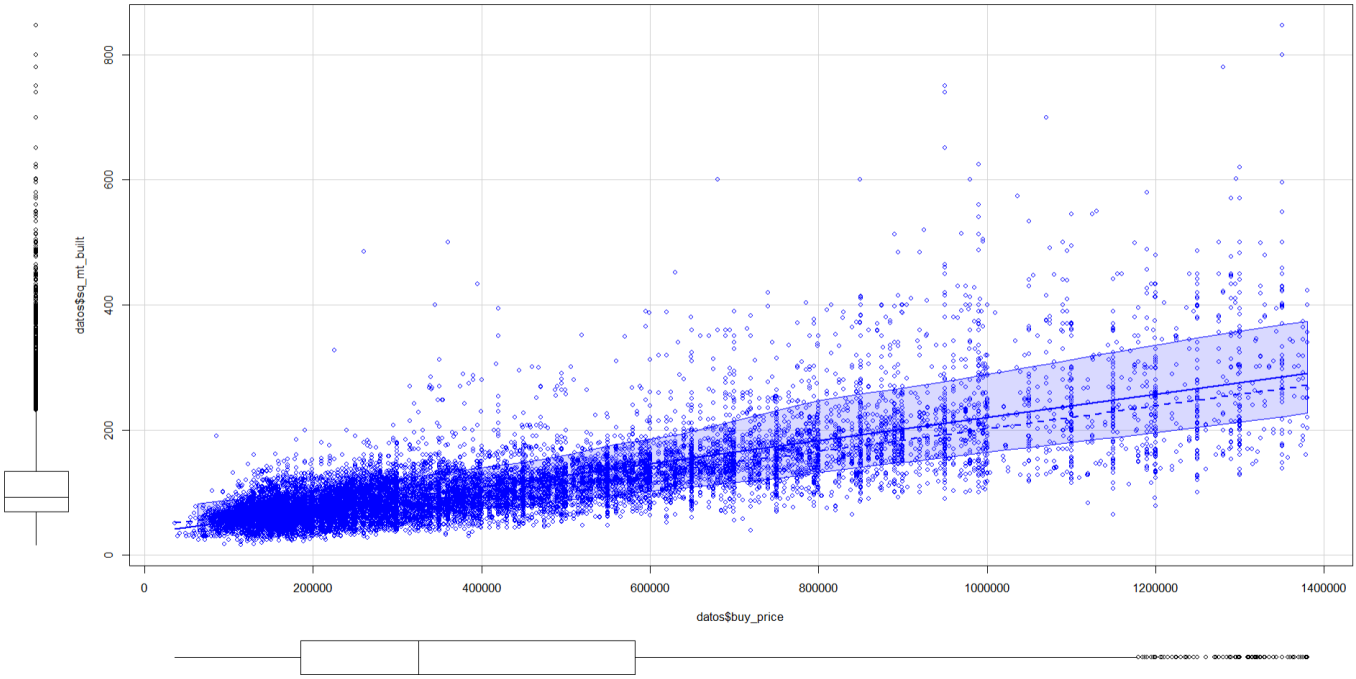

Figura 8. Gráfico dispersión *"buy_price*" vs. "*sq_mt_biult*". *Elaboración propia.*

3. Número de habitaciones y número de baños (*"n_rooms" / "n_bathrooms"*)

Estas son dos variables bastante simples de analizar ya que muestran el número de habitaciones y el número de baños de los inmuebles. El número máximo de habitaciones que tiene alguna vivienda de la base de datos es de 10 y el número máximo de baños es 14, este último dato, llama la atención y hace sospechar de la posible existencia de atípicos que tengan este número tan elevado de baños. Si se va más al detalle se puede ver que solo hay una observación con 14 baños. Esta observación probablemente haya sido debido a un error cometido al publicar las

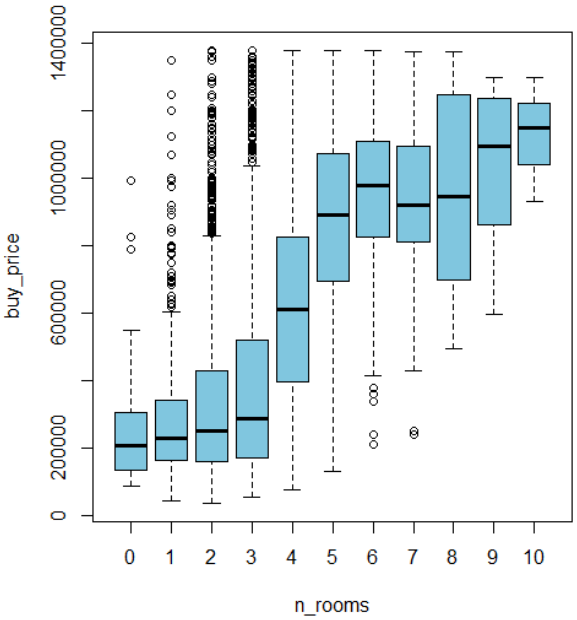

Figura 9. Distribución "*buy_price*" por cada "*n_rooms*". *Elaboración propia*.

ofertas o a la hora de crear la base de datos por lo que se podría considerar la eliminación de esta. Su eliminación no afectaría al estudio, ya que solo se trata de una observación.

La mediana está en viviendas que cuentan con 3 habitaciones y 2 baños. Las viviendas que no tienen ninguna habitación probablemente sean estudios que tienen una distribución abierta en la que no existen separaciones entre zonas.

Como se puede observar en la figura 9, existe una relación lineal bastante positiva entre el número de habitaciones y el precio de compra del inmueble, dato que tiene lógica y sentido si se analiza desde un punto de vista práctico.

4. **Certificado energético (*"energy_certificate"*)**

El certificado energético es un documento oficial, imprescindible para vender o alquilar un inmueble, redactado por un técnico competente que incluye información objetiva sobre las características energéticas de la vivienda. Califica energéticamente un inmueble calculando el consumo anual de energía necesario para satisfacer la demanda energética de un edificio en condiciones normales de ocupación y funcionamiento. Incluyendo la producción de agua caliente, calefacción, iluminación, refrigeración y ventilación (Qué es el Certificado de Eficiencia Energética?, 2023).

En la base de datos de estudio, esta es una variable categórica compuesta por 10 valores diferentes. En el código ha sido necesaria codificar como factor para poder trabajar con ella, pues algunos modelos de regresión no funcionan correctamente si se introducen variables categóricas.

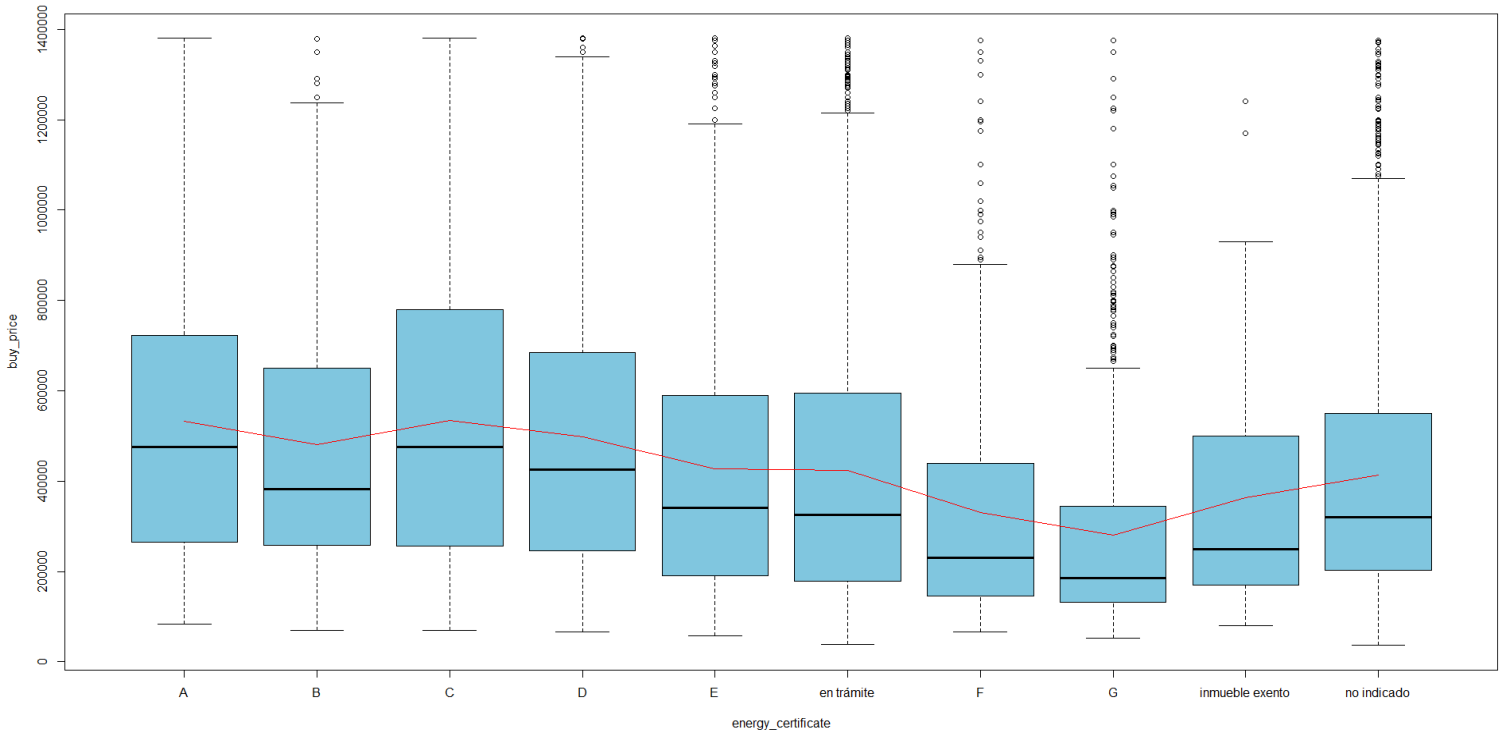

Figura 10. Distribución "*buy_price*" según tipo de certificado eléctrico. *Elaboración propia.*

En la figura 10, se puede observar que el precio medio de compra (línea roja) no varía en exceso en función del certificado energético, aunque sí que es cierto que las viviendas con certificados “F” y “G” tienen una mediana y media en precio de compra menor al resto. Esto se debe a que estos inmuebles están peor acondicionados y su consumo energético es superior al de la media nacional. Esto se puede deber a que sean viviendas más antiguas, que tienen una mala construcción y están peor aisladas, con sistemas de calefacción menos eficientes, etc. En la base de datos de estudio, la mayor parte de observaciones entran en la categoría de “en trámite”, que cuenta con 9.385 observaciones.

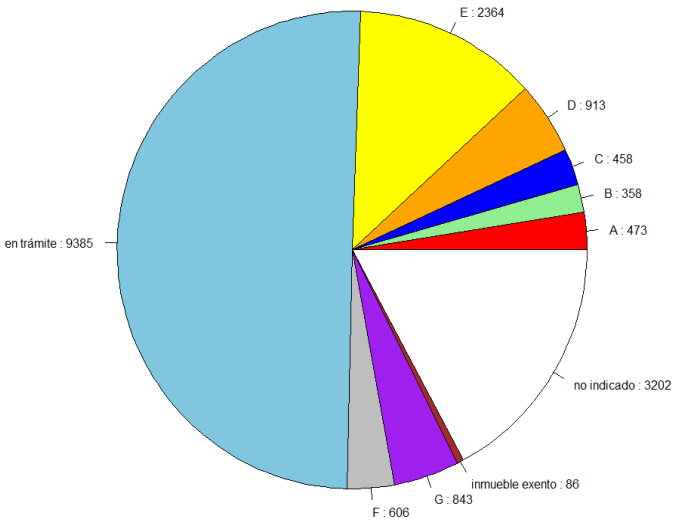

Figura 11. Número de inmuebles por cada tipo de certificado energético. *Elaboración propia.*

De acuerdo con el Real Decreto 235/2013, del 5 de abril, para la compraventa de un inmueble es obligatorio disponer de un certificado energético en vigor, registrado ante el organismo competente de cada comunidad autónoma (Ortega, 2021). Sin embargo, en ocasiones nos encontramos anuncios de compraventa en los que no aparece este atributo o está “en trámite”. Esta última situación indica que se ha iniciado el trámite del registro del certificado ante la administración, pero sí que es cierto que hay ocasiones en las que esto no es cierto del todo.

5. Tipo de inmueble (*"house_type_id"*)

En la base de datos nos encontramos con cuatro tipos de inmuebles diferentes: pisos, casas/chalets, dúplex y áticos. En la figura 12 se puede observar que las casas tienen en general un mayor precio de compra que los otros tres tipos de inmuebles. Se puede ver también como hay una serie de pisos cuyo valor es muy alto y se pueden llegar a considerar atípicos.

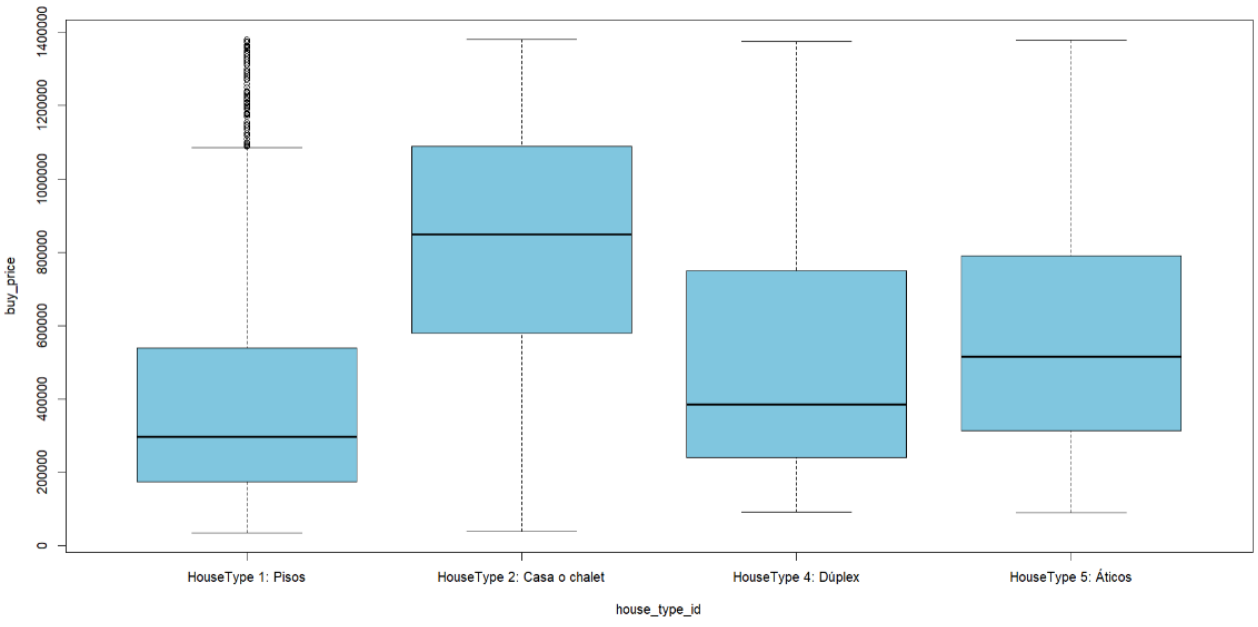

Figura 12. Distribución "*buy_price*" según el tipo de inmueble. *Elaboración propia.*

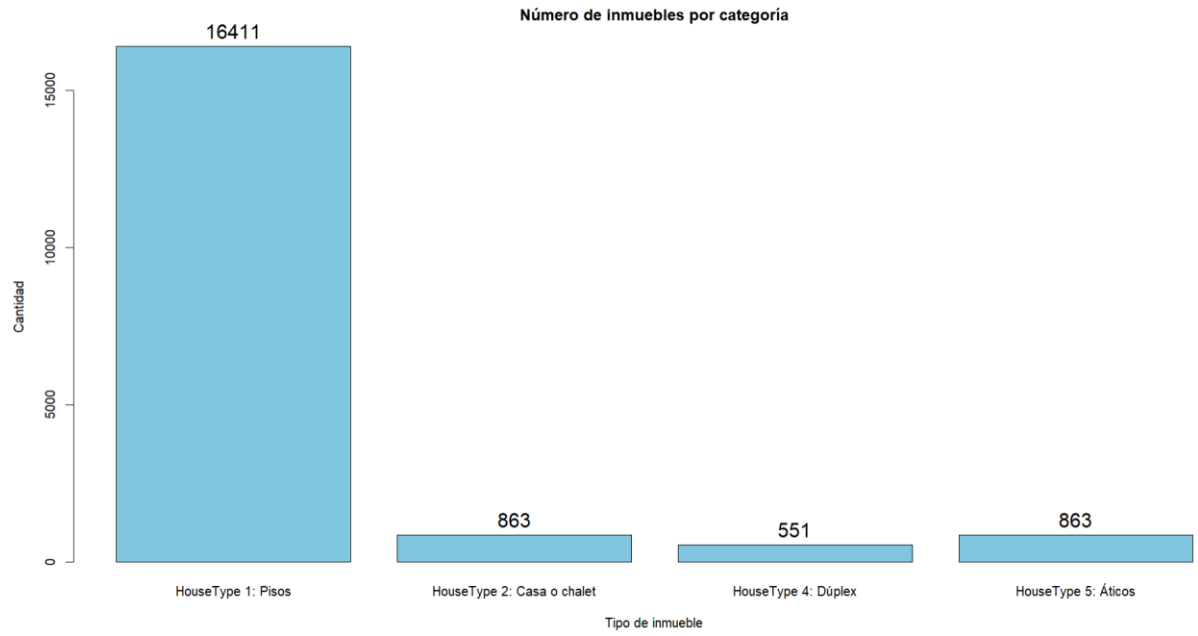

Figura 13. Número de inmuebles por categoría. *Elaboración propia.*

En la base de datos utilizada para el estudio, se puede observar que la mayor parte de las observaciones pertenecen a la categoría de "Pisos" representando un 88% de la muestra.

Para poder trabajar de manera más precisa y cómoda con los modelos de regresión, ha sido necesario transformar esta esta variable categórica en una variable factor, al igual que la variable categórica certificado energético.

6. Precio de alquiler ("rent_price")

Esta variable indica el precio por el que se alquilarían cada uno de los inmuebles. Esta variable tiene una distribución algo menos asimétrica que otras y se puede destacar que el precio medio de alquiler en Madrid en el año 2020 era de 1.354€.

En los datos originales aparecían observaciones con precio de alquiler negativo, por lo que se decidió eliminar esas observaciones ya que probablemente se deba a errores de la plataforma; no tienen ningún sentido un precio de alquiler negativo.

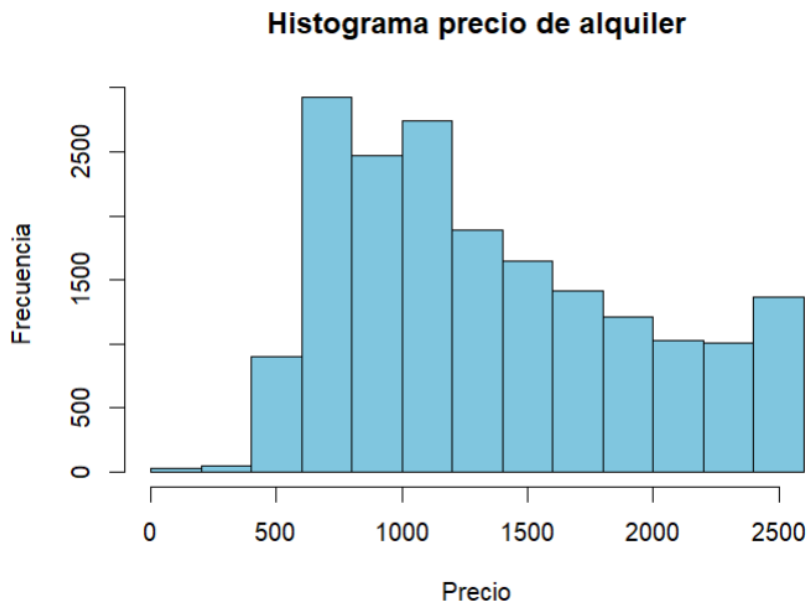

Figura 14. Histograma precio de alquiler. *Elaboración propia.*

Realmente es una variable que está muy correlacionada con la variable dependiente "*buy_price*". Tienen una correlación de 0,812 antes de quitar las observaciones que son consideradas outliers y de 0,99 después de eliminarlos quedándonos solo con la base de datos limpia. Como se mencionará en más detalle más adelante, en un primer momento, se incluyó esta variable entre las variables principales para crear y entrenar los modelos. Sin embargo, tras el análisis de los resultados, se observó que, debido a la alta correlación, se convertía en prácticamente la única variable necesaria para entrenar los modelos y esto hacía que los resultados tuviesen comportamientos extraños en ciertas observaciones, por lo que finalmente se decidió eliminarla de las variables principales y usar otras que no fuesen tan únicas y con menos correlación con la variable dependiente.

7. **Tiene jardín (“*has_garden*”)**

La variable jardín indica si la vivienda cuenta con un espacio verde de jardín o no. Esta variable se debe comentar de forma individualizada ya que, de las viviendas analizadas, solamente 614 tienen jardín, mientras que las 18.074 restantes no lo tienen. La implicación que tiene esto en los modelos con los que se ha trabajado es muy relevante, ya que como se ha mencionado en la descripción metodológica, en este trabajo, el paso previo a la creación de los modelos predictivos es la aplicación de un modelo algorítmico de detección de outliers. Este, debido a que tan solo 614 observaciones tienen jardín, hace que todas ellas sean directamente clasificadas como outliers.

Figura 15. Número de inmuebles con jardín. *Elaboración propia*.

Esto, a la hora de montar algunos modelos con la base de datos ya limpia, hace que las predicciones no estén bien, por lo que se decidió eliminarla de las variables principales del modelo y trabajar sin ella.

8. **Renovación necesaria; tiene piscina, tiene terraza y tiene parking (*"is_renewal_needed" / "has_pool" / "has_terrace” / "has_parking"*)**

Estas cuatro variables, al igual que “*has_garden*”, son variables numéricas dicotómicas. Si se analizan en detalle, se puede obtener una idea más clara de cómo son el tipo de inmuebles que se analizan en la base de datos.

A rasgos generales se podría decir que la gran mayoría de ellos (81,9%) no necesitan una renovación, ya que se encuentran en un buen estado. Esta característica suele hacer que el precio de la vivienda suba. Aquellas que necesitan una renovación, como es lógico, tienden a tener precios más bajos para intentar captar la atención de la gente.

También se puede decir que la mayoría no cuentan con piscina, esto tiene sentido puesto que, como se ha mencionado antes, la mayoría de los inmuebles

pertenecen a la categoría de pisos que están situados en el área geográfica de Madrid, donde las zonas céntricas disponen de espacio limitado como para construir una piscina. Normalmente los inmuebles con piscina tienden a tener precios más elevados.

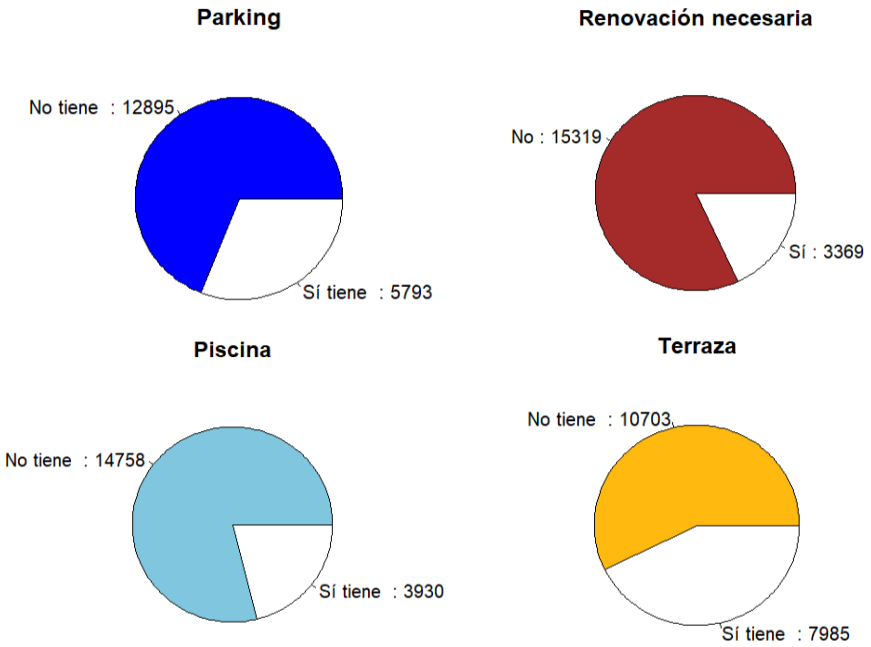

Figura 16. Número de inmuebles según variable parking, renovación, piscina y terraza. *Elaboración propia.*

En cuanto al parking, el 69% de ellos no cuentan con plaza de parking. Característica que hace que el precio de las viviendas tienda a bajar. Como se puede observar en los diagramas de cajas de la figura 17, tanto en parking como en piscina, existe una diferencia entre las viviendas que tienen piscina o parking (1) y los que no tienen piscina o parking (0). Los pertenecientes al grupo (1), tienen precios de compra más altos; en el gráfico se ve la mediana por encima de los que no tienen piscina o parking. Por eso se concluye que las características piscina y parking tienen una correlación positiva con el precio de compra.

El poder aparcar es algo muy valorado en las ciudades de hoy en día, la mayoría de las familias cuentan con al menos un vehículo y debido a que las ciudades tienden a poner cada vez más dificultades para aparcar debido a la búsqueda de la reducción del trasporte individual y el fomento del transporte público, hace que el disponer de una plaza de garaje revalorice el valor de la vivienda.

Por último, en cuanto a la característica de si tiene terraza o no, está más igualado entre las observaciones que sí tienen (42,7%) y las que no tienen terraza (57,3%). Esta característica, no está tan correlacionada con el precio. Se puede ver en la figura 17 que ambas cajas son similares para los inmuebles con terraza y sin terraza. Sin embargo, tras la pandemia, esta característica, junto con tener jardín han ido cobrando cada vez más fuerza. Según el estudio realizado por el *Observatorio Aedas Homes Julio 2021*, tener una terraza o jardín tiene una importancia de 8 sobre 10 a la hora de elegir una vivienda (Moreno, 2021), lo que al fin y al cabo hace que también impacte en el precio de compra de la vivienda.

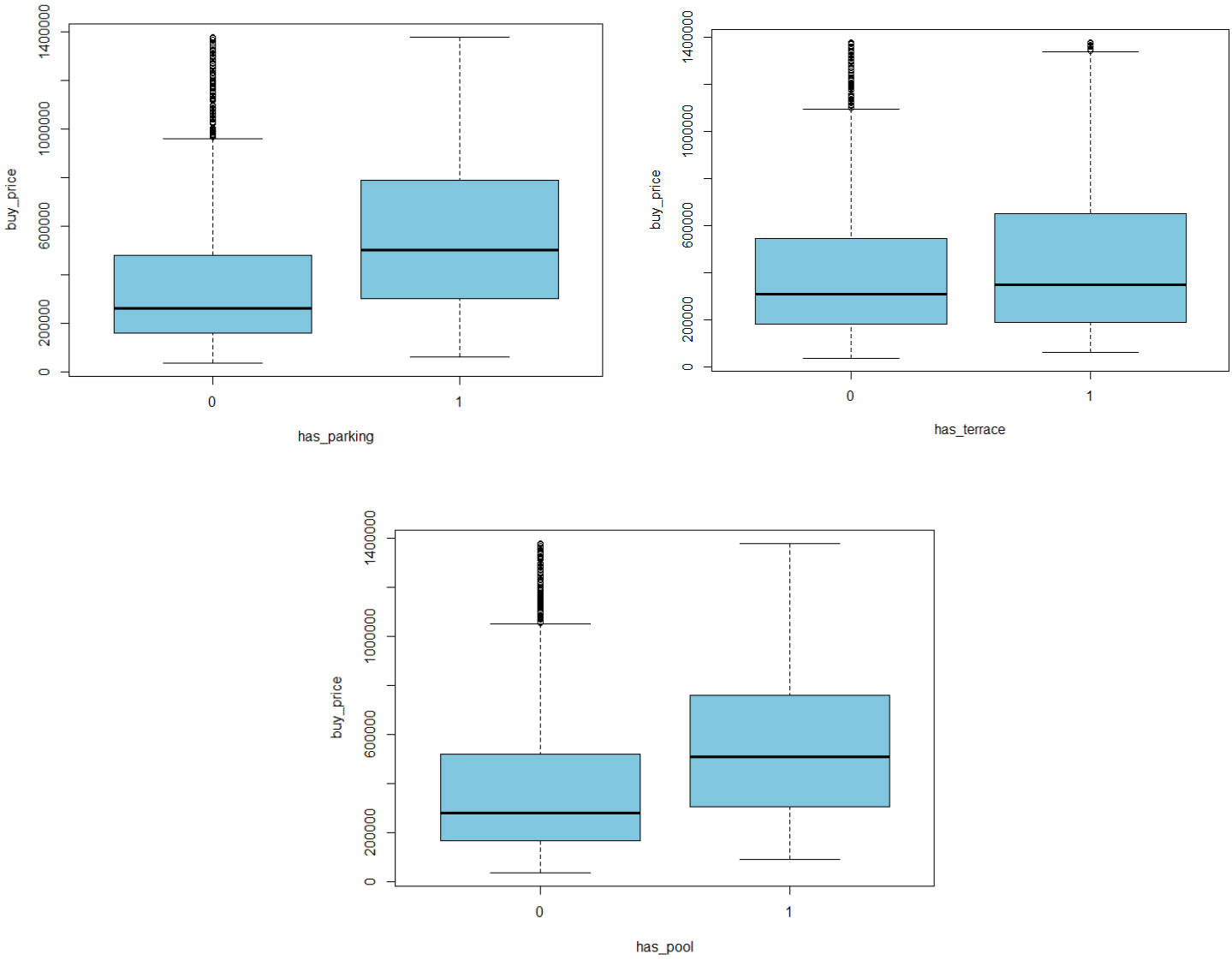

Figura 17. Distribución "*buy_price*" según inmuebles que tienen o no parking, terraza y piscina. *Elaboración propia.*

Tras haber realizado el análisis descriptivo de las variables y haber hecho una primera limpieza y transformación de los datos, se puede observar que la muestra original del dataset estaba formada por 21.709 observaciones y tras este primer acercamiento mencionado, se obtuvo un conjunto de datos formado por 18.687 observaciones y 10 variables, ya que como se verá a continuación con más profundidad, *"rent_price"* tampoco ha sido finalmente seleccionada.

Ahora, como siguiente paso importante, se encuentra el de analizar las correlaciones existentes entre todas estas variables definitivas seleccionadas. Esta medida solamente se puede obtener de las variables numéricas, es por ello por lo que, de la base de datos, solo se han seleccionado las ocho variables numéricas a las que luego se les ha calculado la correlación. En la figura 18, se pueden observar las variables que están más correlacionadas.

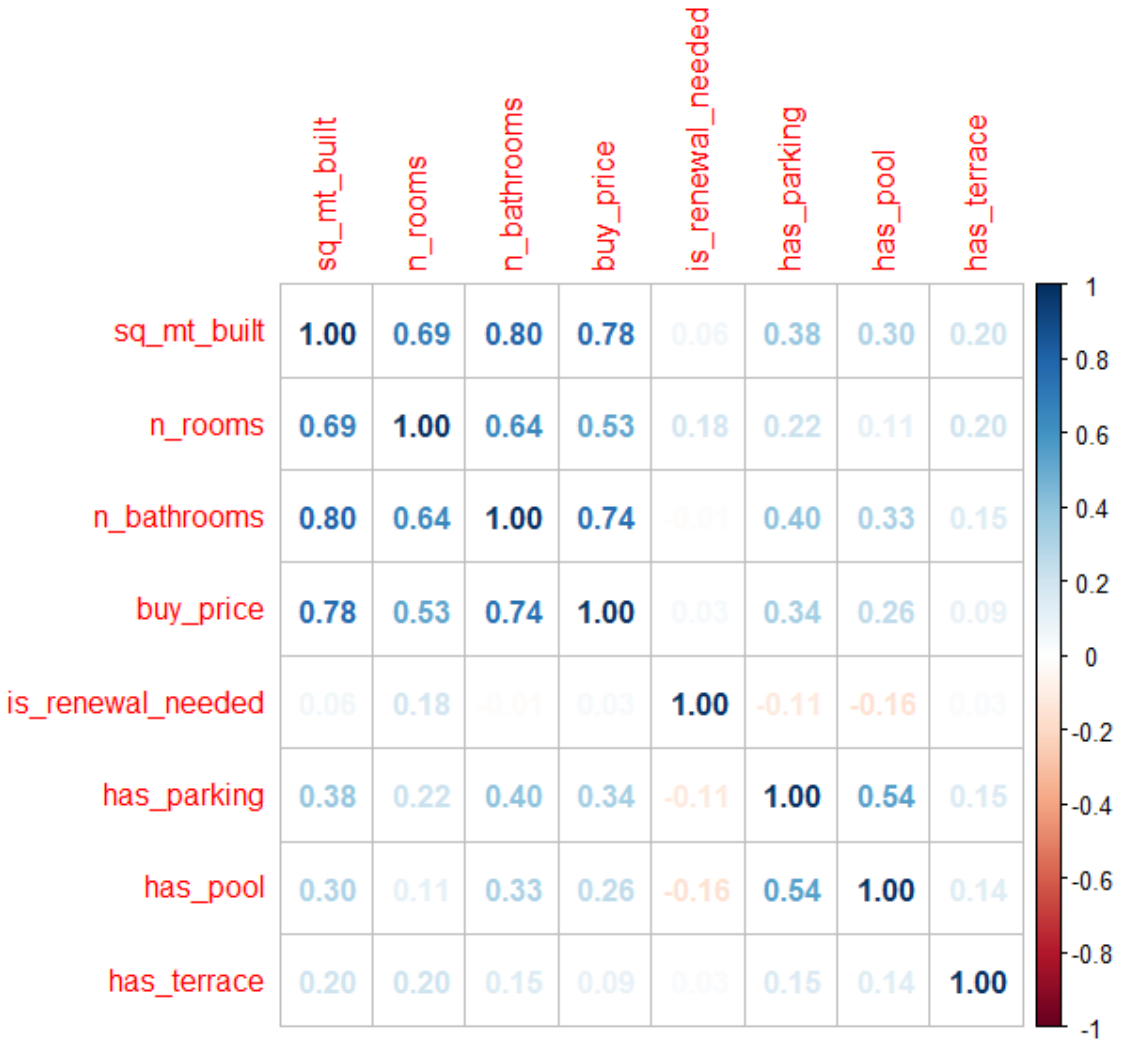

Figura 18. Correlación entre variables numéricas de la base de datos. *Elaboración propia.*

Las variables número de baños y metros cuadrados construidos, son las que mayor correlación tienen. Esto tiene sentido desde el punto de vista de que, si la vivienda tiene más baños, necesariamente debe tener mayor cantidad de metros cuadrados. Debido a que estas dos variables formarán parte de las variables independientes del modelo, quizás haya que considerar la eliminación de alguna de ellas para evitar un suceso de multicolinealidad.

La multicolinealidad sucede cuando las variables independientes de un modelo de regresión están muy correlacionadas. Si esto sucede, se puede ver afectada la estabilidad de los coeficientes de regresión del modelo y la interpretación de la importancia relativa de las variables independientes. En este caso, puesto que la correlación no es de más de 0,9 y lo que más interesa es generar un modelo predictivo y no entender el papel de cada una de las variables independientes (modelo explicativo), no se procede a eliminar estas variables, evitando también así que se produzca endogeneidad, que sucede cuando se omite alguna variable independiente del modelo que es importante para el mismo y está correlacionada con las otras variables (Martínez, 2021). Cuando esto no sucede, se dice que hay exogeneidad y es lo que se busca en el modelo.

A continuación, también se podría destacar que las variables "*buy_price*" y "*sq_mt_built*", tal y como también se identificó en el análisis descriptivo, están bastante correlacionadas, lo que indica que esta variable independiente puede ser un predictor importante para predecir la variable dependiente en el modelo.

El resto de las variables tienen una correlación por debajo de 0,7 por lo que no será necesario prestar mayor atención de la necesaria a ninguna de estas variables restantes.

3.3. METODOLOGÍA DE MODELADO

Una vez realizado el análisis descriptivo y exploratorio de las variables, se procede a describir de forma más detallada cuál va a ser la metodología empleada de cara al procesamiento de los datos y obtención de los resultados.

En primer lugar, una vez que se han descrito las variables, ajustado los datos y se tiene plena conciencia de los datos con los que se está trabajando, se procederá a aplicar el modelo de detección de outliers Isolation Forest, con el objetivo de separar de la base de datos de 18.688 observaciones todas aquellas que son consideradas como outliers. De esta manera se crearán dos conjuntos de datos separados, los que no contienen outliers y los outliers.

El siguiente paso consistirá en entrenar cuatro modelos de regresión: regresión lineal, KNN, Random Forest y XGBoost, cuyo funcionamiento ya ha sido previamente descrito en el apartado 2.2 del trabajo. Estos modelos han sido seleccionados por ser los más utilizados y por ofrecer los mejores resultados debido a sus características.

Una vez entrenados los cuatro modelos, se analizarán sus resultados y se escogerá aquel que logre un mayor rendimiento, que en este caso será el que obtenga un menor RMSE con los datos de validación. Con este modelo, será con el que se obtenga una predicción del precio de compra de las viviendas más preciso.

A continuación, lo que se hará es, el modelo seleccionado que ha sido previamente entrenado se aplicará al conjunto de datos formado por los outliers y se analizarán los resultados, intentando de esta forma identificar cuáles de estas observaciones podrían estar sobrevaloradas y cuáles infravaloradas. Como se ha mencionado, esto se realiza de esta manera innovadora ya que el hecho de que sean clasificados como outliers, se puede deber a una valoración poco precisa.

4. RESULTADOS

4.1. DETECCIÓN DE OUTLIERS

El primer paso en la metodología del modelado, como se acaba de mencionar, es la detección de aquellas observaciones que se podrían considerar como atípicos. Para ello se va a utilizar el modelo de Isolation Forest.

A la hora de implementar este modelo en la base de datos, como primer paso, se requiere la instalación de la librería "isotree" en RStudio, que es la que lleva incorporada el paquete de Isolation Forest.

A continuación, se aplica el comando del modelo de Isolation Forest a los datos compuestos por las 10 variables y las 18.687 observaciones. En el diseño del modelo también se indica que genere 1.000 árboles de decisión, mediante el comando "ntrees=1000". A la hora de generar el código para este algoritmo, R permite introducir una gran cantidad de argumentos para ajustarlo de la mejor manera que uno considere. Por ejemplo, otro campo importante es el del "sample_size" que indica el tamaño muestral de las submuestras de datos con las que se construirá cada árbol binario. En este campo, se ha probado a introducir diferentes valores según las recomendaciones de Rstudio, sin embargo, cuando mejor selecciona outliers, dejando una base de datos limpios con la que funcionan bien en los modelos, es cuando no se indica nada en este campo de "sample_size" y coge el que tiene por defecto, que para este paquete es de 256. Esta cantidad, en la teoría, es bastante apropiada, ya que, aunque perezca pequeña, en la propia descripción de Cortes (2020) donde habla de cómo funciona el algoritmo, afirma que, si la cantidad de datos es grande, se sugiere establecer un tamaño de muestra menor para cada árbol (Cortes, 2022).

Una vez se tiene el modelo construido y ejecutado a la base de datos, se obtiene como resultado las puntuaciones entre 0 y 1 antes mencionadas en la explicación del funcionamiento del algoritmo. A continuación, en el gráfico de la figura 19 se muestra el histograma de estos resultados.

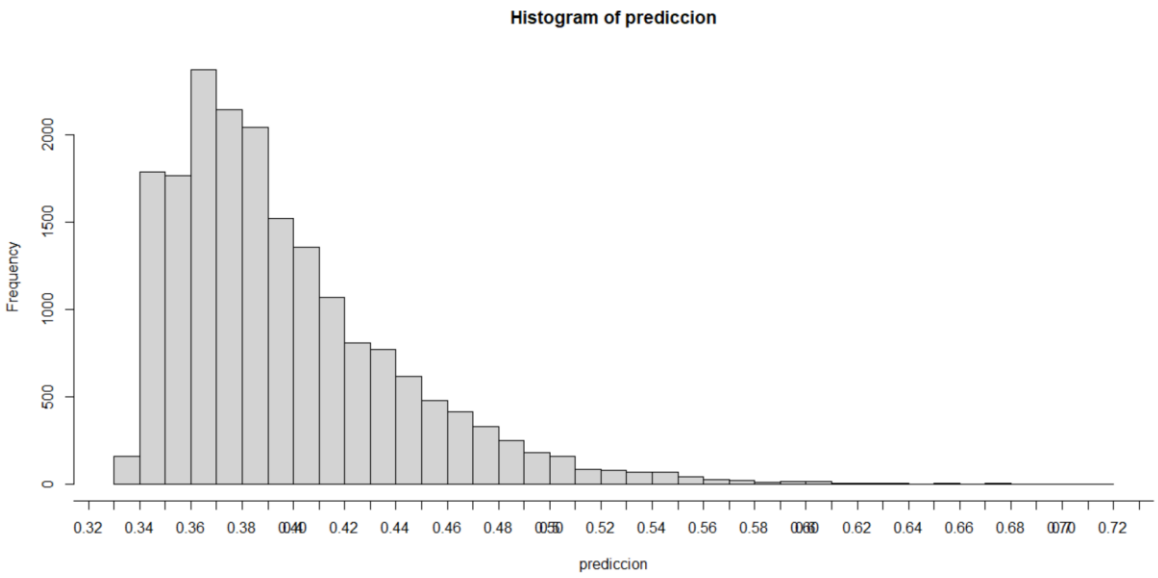

Figura 19. Histograma resultados predicción Isolation Forest. *Elaboración propia.*

Estos resultados no proporcionan una clasificación directa de qué observaciones son valores atípicos y cuáles no. Como se mencionaba antes, cada observación tiene una puntuación que cuanto más cercana a 1, indicará que existen más probabilidades de que sea un atípico. Sin embargo, no existe un valor exacto a partir del cual podamos afirmar a ciencia cierta cuáles son atípicos. Como bien señala Amat en su explicación, "al ser un método no supervisado, no hay forma de conocer el valor óptimo a partir del cual se debe de considerar que se trata de una anomalía. La puntuación asignada a cada observación es una medida relativa respecto al resto de observaciones" (Amat, 2020). A la hora de poner esto en práctica, lo que se hace normalmente es considerar como posibles atípicos aquellas observaciones que tengan una distancia predicha por debajo de un determinado cuantil.

Para los datos de estudio en concreto, por la distribución del histograma, se va a establecer el cuantil de 85%, lo cual da un umbral por encima del cual se va a clasificar las observaciones como outliers de 0,4409.

Una vez identificados los atípicos, se observa que 2.803 observaciones se han clasificado como outliers, quedando entonces en la base de datos, como entradas "limpias" 15.884 observaciones.

4.2. ENTRENAMIENTO Y MODEADO SIN OUTLIERS

Una vez se tienen los datos separados entre limpios y atípicos, tal y como se indica en la metodología, se procede a entrenar distintos modelos intentando encontrar el que mejor realice una predicción con los datos proporcionados.

El código utilizado para este entrenamiento, al igual que el utilizado para la limpieza y estudio de los datos, se puede ver de manera detallada en el Anexo I.

Es relevante mencionar que todos los modelos antes de ser entrenados se les ha aplicado un tipo especifico de generador de números aleatorios y una semilla para dicho generador. Esto permite obtener los mismos resultados aleatorios cada vez que se ejecutan los modelos de regresión y así facilitar la comparación de los resultados de los diferentes modelos.

En este caso se ha usado “RNGkind("Super", "Inversion", "Rounding")” y “set.seed(123)”. Aquí el “RNGkind()” se utiliza para llamar al generador de números aleatorios, mientras que el “set.seed()” se utiliza para establecer una semilla para el anterior generador de números aleatorios, este toma un número entero y garantiza que los números aleatorios necesarios que se generen sean reproducibles (Gilbert, 2022).

Como para la realización de este trabajo ha sido necesario realizar múltiples modificaciones con el fin de mejorar los modelos y comparar los resultados obtenidos entre ellos, esta es la mejor forma de poder comprar los resultados y evitar que el componente aleatorio de estos modifique los resultados en cada ejecución.

4.2.1. MODELO REGRESIÓN LINEAL MÚLTIPLE

El primero de los modelos entrenados es el más simple de todos, el de regresión lineal. En este trabajo, dado que se quiere predecir el precio de compra de las viviendas, la variable Y sería “*buy_price*” y las otras 9 variables, las diferentes Xi de la ecuación en la que está basada este algoritmo.

A la hora elaborar el modelo, debemos considerar introducir una transformación logarítmica a esta variable dependiene, ya que, si se analiza su forma en un histograma, se puede observar cierta asimetría de los datos hacia la derecha. Esta posible trasformación logarítmica podría ayudar a reducir la asimetría y hacer que los datos se ajustasen mejor a los supuestos del modelo. Sin embargo, esto no siempre es necesario o apropiado, por lo que se decidió probar a crear el modelo de ambas formas y analizar sus resultados, tratando de ver cuál predice mejor y comete un menor error.

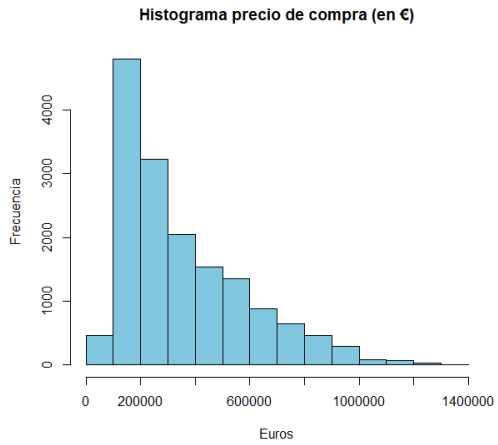

Figura 20. Histograma *"buy_price". Elaboración propia.*

Para sorpresa, pese a que la forma aconsejaba hacer esta transformación, los resultados obtenidos son mucho mejores sin aplicar la transformación logarítmica a la variable dependiente.

Para la selección de las variables que finalmente se iban a introducir en el modelo, se procedió a realizar una especie de selección de variables backward stepwise, que consiste en primero introducir todas las variables en la ecuación y a continuación ir retirando una tras otra y viendo si va mejorando o empeorando el modelo (Ferrero, 2017).

La ecuación utilizada en el código para entrenar el modelo ha sido:

"modeloreg <- lm(log(buy_price) ~ sq_mt_built + n_rooms + n_bathrooms + energy_certificate + house_type_id + has_terrace, data = train)"

Esta, como se puede observar, incluye todas las variables que tenemos en la base de datos limpia, mientras que, en la logarítmica, se eliminaron algunas como *"has_pool" o "is_renewal_needed"* ya que de esta forma se mejoraba su precisión.

En la tabla 4 se muestran los resultados de cada uno de los modelos, los cuales fueron necesarios elaborar para poder ver de qué manera proporcionaba mejores resultados.

	Sin transformación logarítmica	Con transformación logarítmica
R2 ajustado	0,6873	0,6386
RMSE (test)	131.660,2	181.265,1
AIC	293.748,8	10.671,66
BIC	293.902,5	10.803,36

Tabla 4. Resultados modelo de regresión lineal sin y con transformación logarítmica. *Elaboración propia.*

Dado que la raíz cuadrada del error cuadrático medio (RMSE), es menor en el modelo sin la transformación logarítmica y tanto el AIC como el BIC son mayores en este, es el que es seleccionado como modelo de regresión lineal múltiple.

Como se sabe, la regresión lineal es un modelo muy simple en comparación con muchos otros existentes dentro del campo del Machine Learning, por lo que se espera que los otros modelos que se entrenen predigan mejor que este.

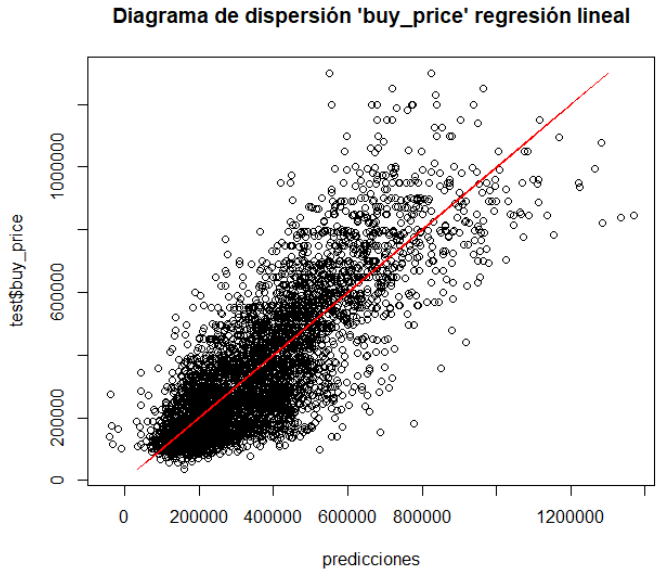

Figura 21. Resultados predicción modelo regresión lineal. *Elaboración propia.*

En el gráfico de la izquierda se ve la dispersión de las observaciones. La línea roja muestra cuáles serían los resultados si la predicción y las observaciones reales fuesen iguales. Como se mencionó antes, existe un error de 131.660€. Por lo que se deben probar otros algoritmos para intentar encontrar alguno que cometa menor error.

4.2.2. MODELO RANDOM FOREST

El segundo modelo a entrenar es el de Random Forest, el que, como se ha mencionado antes, tiene las expectativas más altas. En comparación con los otros modelos probados, este es el que más carga computacional tiene y por lo tanto el que más tiempo tarda en ejecutarse.

A la hora de crear este modelo con las variables de la base de datos limpia, se ha establecido que utilice 500 árboles, número que también utiliza por defecto en el caso de que no se le indique otra cifra. Por otro lado, en cuanto al número "m" de variables predictoras, también se ha utilizado el que coge de manera predeterminada, que en el caso de la función randomForest de la librería "*randomForest*" de Rstudio, es la raíz cuadrada del número total de variables predictoras que tiene el modelo. En definitiva, "m" ha tomado el valor de la raíz cuadrada de 10. Es con estos parámetros, con los que se obtienen los mejores resultados, que aplicando una partición de los datos en 70% para entrenamiento y 30% para test, son de un RMSE en test de 122.644,9€ y un r cuadrado de 0,7116. El r cuadrado indica la proporción de la varianza de la variable dependiente que queda explicada por el modelo, en otras palabras, da una medida de lo bien o mal que se ajusta el modelo elaborado a los datos que tenemos.

Con este modelo, como se explicaba antes, se tiene la ventaja de que se puede ver la importancia que se les da a las distintas variables. En la figura 22, se puede ver como la variable metros cuadrados construidos (sq_mt_built) es la variable más importante, seguido del número de baños, el número de habitaciones y el certificado energético que tienen.

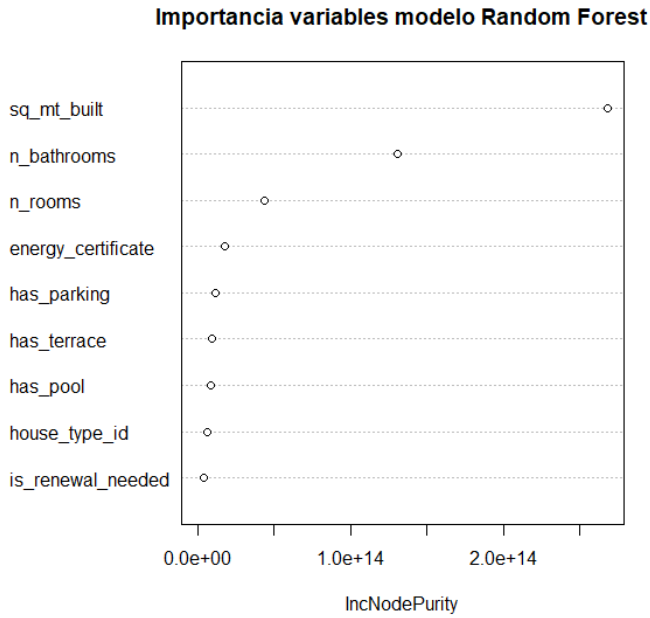

Figura 22. Importancia variables Random Forest. *Elaboración propia.*

A la hora de graficar las predicciones que realiza el modelo, comparándolas con el precio real de los inmuebles, se obtienen las diferencias que se muestran en el diagrama de dispersión de la figura 23. La línea roja muestra aquellos valores cuyo valor predicho y valor real es el mismo. Con esto se podría decir que, si el modelo fuese muy fiable y predijese con mucha precisión, las observaciones que están muy por encima de la línea roja están sobrevaloradas y las que están por debajo infravaloradas, pues su valor real está muy por encima o muy por debajo respectivamente, del valor predicho.

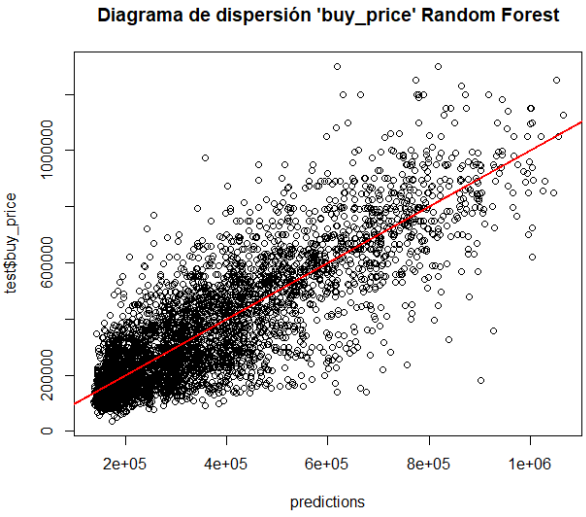

Figura 23. Resultados predicción Random Forest. *Elaboración propia.*

Al evaluar los resultados de los modelos desarrollados hasta la fecha, es posible observar que ambos presentan errores significativos de más de 100.000€ en las predicciones. Estos resultados plantean la posibilidad de que existan deficiencias en la implementación de los modelos o, por ejemplo, sugieren que la heterogeneidad de las unidades en las variables utilizadas podría requerir un proceso de estandarización. Sin embargo, no es algo que se vaya a hacer, puesto que los modelos de Random Forest que se acaban de describir y los XGBoost que serán desarrollados a continuación, no son sensibles a la distancia de las variables, por lo que este aspecto no supone una preocupación. En el caso del KNN, se verá que, para el cálculo de las distancias, la diferencia de unidades sí que puede llegar a suponer un problema, pero dentro de los parámetros del modelo se ajusta para que sus valores sean estandarizados automáticamente.

4.2.3. MODELO XGBOOST

El tercero de ellos es el XGBoost. Una de las principales desventajas de este modelo es que solamente admite variables numéricas. En el caso de la base de datos seleccionada, fue necesario realizar varias transformaciones para conseguir adaptar los datos de

manera correcta. La primera de ellas fue hacer “one hot encoding” en las variables certificado energético “energy_certificate” y tipo de vivienda “house_type_id” ya que son variables categóricas y de esta manera se transforman en numéricas.

A continuación, se crearon las matrices de entrenamiento y test, eliminando de ellas la variable dependiente. Luego se establecieron los parámetros del modelo de la manera más efectiva posible, algo muy importante para un modelo de XGBoost, ya que estos pueden hacer que varíe en gran medida el resultado del modelo.

En el caso de la base de datos de la vivienda, se decidió establecer una profundidad máxima de 3, un número de rondas de 200 y una *eta* de 0,3. La profundidad máxima “*max_depth*” en un modelo de XGBoosting, controla la profundidad y complejidad máxima de los árboles de decisión en el conjunto de modelos ensamblados. Este valor es importante ya que, si el valor es muy alto, el modelo podría generar overfitting y dificultar la generalización del modelo a la hora de probarlo con nuevas observaciones. Por el contrario, si este valor es muy bajo, limitaría la complejidad de los árboles, evitando sobreajuste, pero podría hacer que el modelo no capturase bien los patrones y relaciones de los datos, proporcionando así un resultado ineficiente. Por defecto, la función xgb.train(), utilizada para elaborar el modelo e incluida dentro de la librería “xgboost”, escoge una max_depth de 6. En un primer momento, se elaboró el modelo con esta profundidad, sin embargo, probando diferentes alternativas se identificó que estableciendo este parámetro a 3, se obtenían mejores resultados. En cuanto al parámetro “eta”, este también afecta al rendimiento del modelo, por eso hay que tener cuidado con el valor que se le asigna. “eta” es un factor de escala entre 0 y 1 que controla la velocidad de aprendizaje del modelo y equilibra el ajuste y generalización del mismo. Un valor alto, permite un ajuste más rápido pero que puede llevar a sobreajuste, un valor menor, implica más iteraciones, pero puede ayudar a evitar sobreajuste y mejorar la generalización del modelo (Chen & Guestrin, 2016). En este caso, una vez probados varios valores, se ha decidido dejar el 0.3, que es el que utiliza el algoritmo también por defecto, ya que es con el que se han conseguido mejores resultados en términos de menor RMSE y mayor r cuadrado. Por último, el “nrounds” es el número máximo de iteraciones de “boosting” o de refuerzo que se van a realizar. Con cada una de estas iteraciones, el modelo tiene que ajustar los datos de entrenamiento, calcular los

gradientes, actualizar los pesos de los árboles y realizar la predicción (Chen & Guestrin, 2016). Este parámetro, una vez más va a influir en la precisión y rendimiento del modelo, así como aumentar el riesgo de overfitting. Por defecto, si no se especifica nada, el algoritmo asigna un valor de 100 a este parámetro, sin embargo, en la construcción del modelo para el presente trabajo, se le ha asignado el valor de 200, obteniendo así los mejores resultados posibles.

Con la base de datos introducida, el modelo obtiene un RMSE para el test set de 124.499,9 y un r cuadrado de 0.7037. A la hora de visualizar los resultados, se identifica la figura 24. Al igual que en los resultados de otros modelos, las observaciones que están por encima de la línea roja, podrían ser observaciones sobrevaloradas y las que están por debajo, más si están muy por debajo, observaciones infravaloradas. En este gráfico se pueden ver muy bien los candidatos a ser grandes oportunidades de inversión, aunque habría que analizar detalladamente sus características.

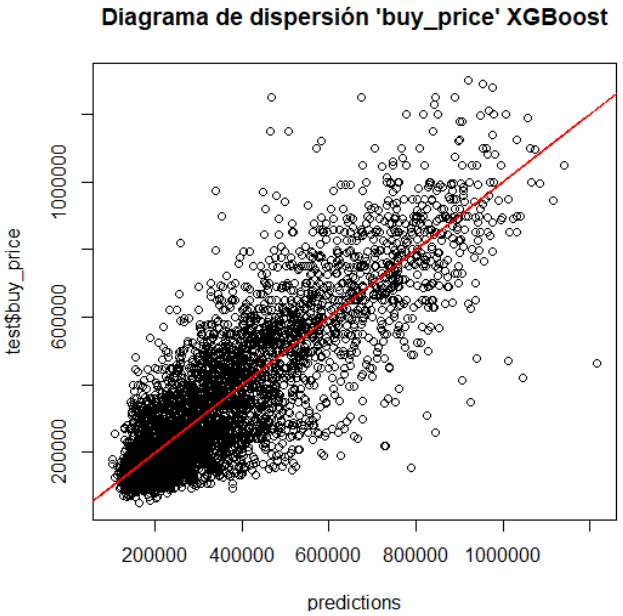

Figura 24. Resultados predicción XGBoost. *Elaboración propia.*

4.2.4. MODELO KNN

Por último, el cuarto modelo entrenado es el KNN. A la hora de elaborar el modelo para la base de datos de las viviendas, se han establecido una serie de controles para definir cómo se quiere realizar el proceso de entrenamiento del modelo para intentar así lograr los mejores resultados posibles. Estos pueden hacer que la ejecución del modelo implique una mayor carga computacional, pero normalmente consigue lograr un modelo más preciso.

El primero de estos controles ha sido que se repita el procedimiento de remuestreo 3 veces “repeats=3”, en segundo lugar, que el número de folds en la cross-validation sea

de 10. La cross-validation es un método para obtener estimaciones fiables del rendimiento de un modelo utilizando únicamente los datos de entrenamiento.

Para poder después predecir el rendimiento de un modelo en un nuevo conjunto de datos, realmente es necesario evaluar su rendimiento en un conjunto de datos que no participe en la formación del modelo, es decir en el test set. Al comparar el rendimiento en el conjunto de prueba y el rendimiento en el conjunto de entrenamiento, se puede analizar si existe sobreajuste y tratar de evitarlo si es necesario (Atanasovski et al., 2020). En tercer lugar, en los parámetros de construcción del modelo, se establece que se pruebe 15 valores distintos para el hiperparámetro K, "tunel=15", de esta manera probará 15 parámetros de K y escogerá aquel que reduzca más la RMSE. Por último, también se indica que se realice un preprocesamiento de los datos de centrado y escalado tipificando así los valores de las variables, necesario, como se mencionaba antes para que no se distorsionen las medidas de distancias.

Una vez construido el modelo y ejecutado el código de Rstudio que se puede ver en el Anexo I, el modelo utiliza K=9 como valor del hiperparámetro ya que es con este con el que logra obtener el menor RMSE. Por lo tanto, para predecir un punto del modelo, va a coger los 9 valores más cercanos a este y calculará la media de sus valores, asignando ese resultado como valor predicho.

En el gráfico de la figura 25 se pueden ver los diferentes valores obtenidos según el valor del hiperparámetro y como con K=9 se obtiene una RMSE de 133.228 con los datos de entrenamiento.

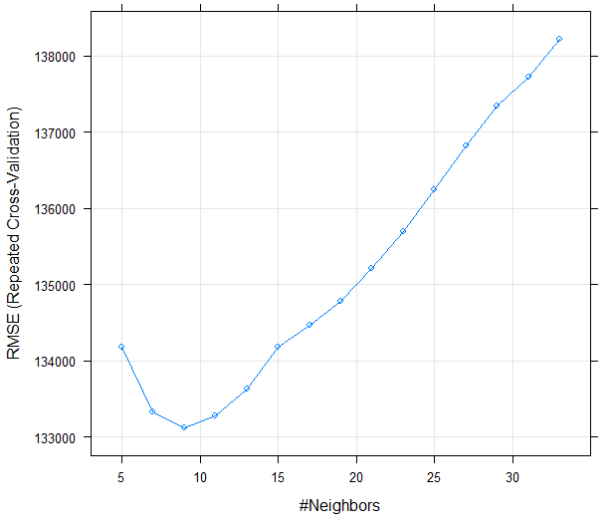

Figura 25.RMSE según el valor del hiperparámetro K. *Elaboración propia.*

Con el modelo también se obtienen unos resultados de un r cuadrado de 0,6809 y un RMSE en los datos de test de 132.228,6; bastante similar a los obtenidos en los otros modelos entrenados.

En el gráfico de dispersión de la figura 26, se puede observar el resultado obtenido con ese modelo.

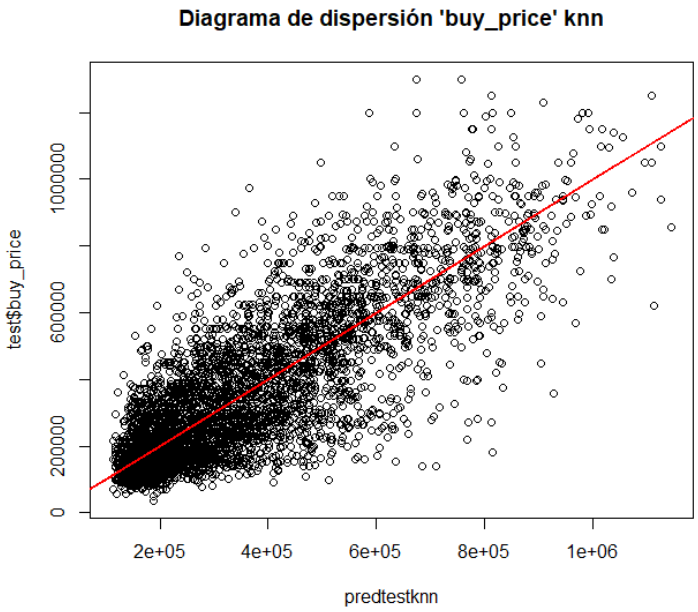

Figura 26. Resultados predicción KNN. *Elaboración propia.*

4.2.5. ANÁLISIS Y COMPARACIÓN DE RESULTADOS

Una vez entrenados los cuatro modelos de regresión escogidos y aplicados a la base de datos, de acuerdo con la metodología, el siguiente paso es comparar los resultados de cada uno y escoger cuál de ellos es el que mejor realiza las predicciones. A continuación, será el escogido el que se utilice para predecir los valores de la base de datos de outliers previamente separada de los datos limpios y así poder analizar con más detalle cuáles de esos pueden estar sobrevalorados y cuales infravalorados.

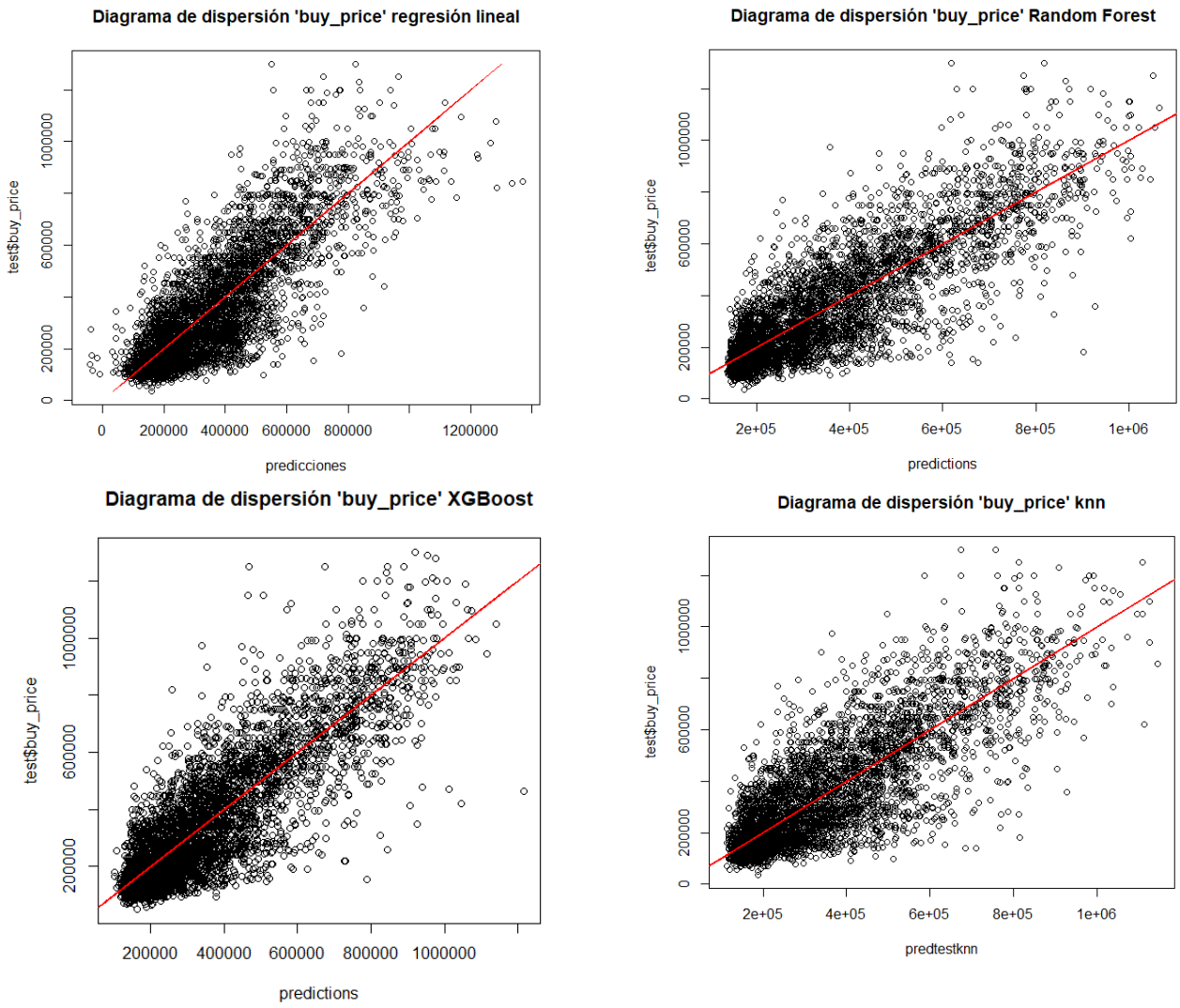

Figura 27. Comparación diagramas de dispersión con los resultados de predicción de todos los modelos entrenados. *Elaboración propia.*

TABLA RESUMEN MODLEOS

	Regresión lineal	KNN	Random Forest	XGBoost
R2 ajustado	0,6873	0,6809	0,7116	0,7037
RMSE (test)	131.660,2	132.228,6	122.644,9	124.499,9
AIC	293.748,8			
BIC	293.902,5			

Tabla 5. Comparación resultados numéricos modelos predictivos. *Elaboración propia.*

Tanto de los gráficos con los resultados de la predicción de los distintos modelos como de la tabla 5 de resultados, se puede observar que los que mejores resultados proporcionan son el Random Forest y el XGBoost. El Random Forest es el que comete un menor error de predicción y en el que mejor se explica la variabilidad de los datos para la variable "*buy_price*", es decir, el que mejor se ajusta a los datos con los que se trabaja.

A pesar de que no se haya mencionado antes, es importante destacar que finalmente los cuatro modelos fueron entrenados para predecir el valor de "*buy_price*" utilizando 9 variables dependientes que son: "*sq_mt_built*" *"n_rooms" "n_bathrooms", "is_renewal_needed", "has_parking", "energy_certificate", "house_type_id", "has_pool"* y *"has_terrace*".

Como ya se mencionó durante el análisis exploratorio, la variable *"rent_price"*, aunque en un primer momento parecía prometedora y significativa, no ha sido utilizada para entrenar los modelos debido a que estaba demasiado correlacionada con la variable independiente, lo que generaba que los modelos ignorasen en cierto modo al resto de variables, obteniendo prácticamente los mismos resultados que si se introdujese solamente esta variable. Después de preparar y entrenar los distintos modelos, se obtenían resultados con muy poco error, pero cuando se aplicaba a nuevos datos, la predicción dependía únicamente a la variable "*rent_price*". Cuando el modelo se aplicaba a los outliers, se observaban resultados excesivamente extraños que se debían a que si la variable "*rent_price*" era muy baja por algún motivo, predecía un valor muy bajo, aunque el resto de las características o incluso el "*buy_price*" fuera muy alto. Un ejemplo de esto se muestra en los siguientes gráficos:

Random Forest:

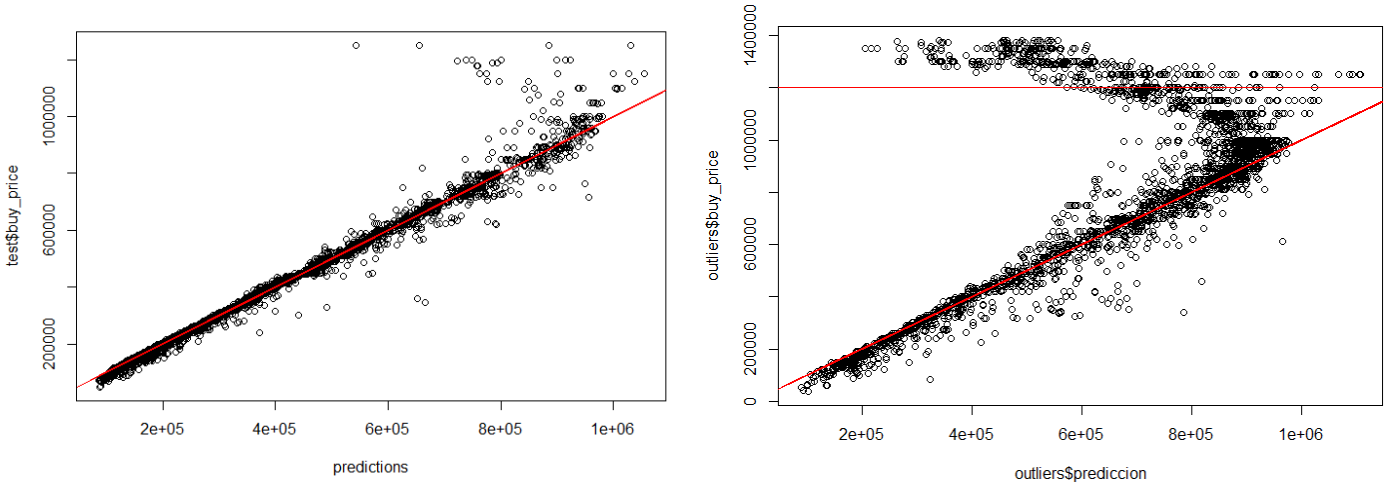

Figura 28.Predicción modelo de Random Forest en base de datos limpios y outliers. *Elaboración propia.*

XGBoost

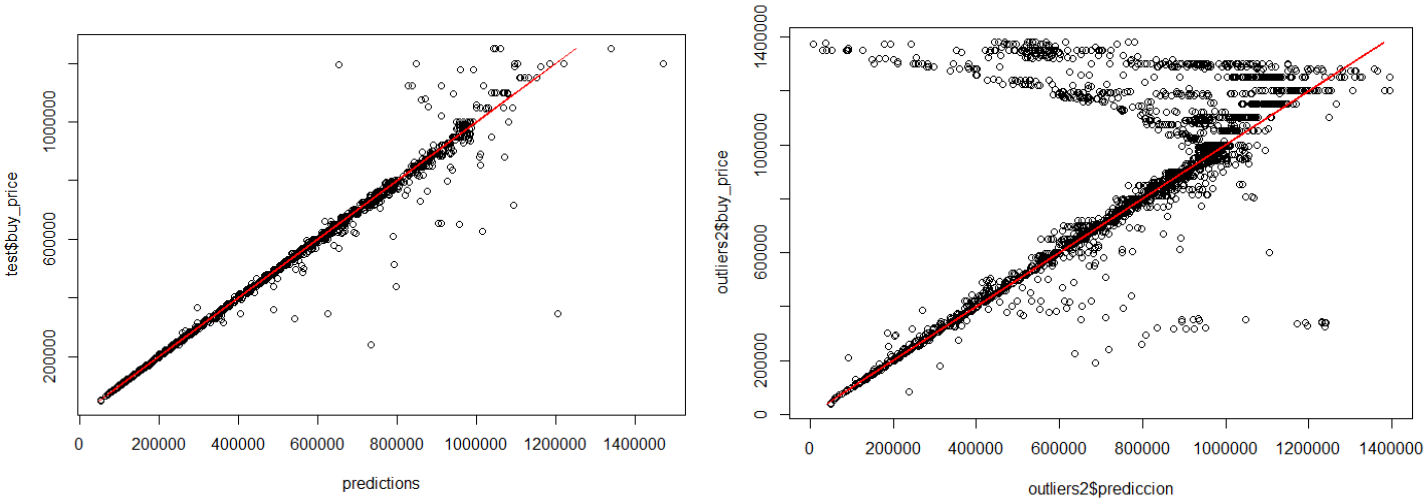

Figura 29.Predicción modelo de XGBoost en base de datos limpios y outliers. *Elaboración propia.*

En ellos se pueden ver los resultados obtenidos por los modelos que mejores resultados generan, el Random Forest y el XGBoost. En el diagrama de dispersión de la izquierda de cada uno de los modelos se pueden ver los resultados de la predicción realizada a los datos del test set, en ellos se compara, el valor original de la base de datos "test$buy_price" y el valor predicho por el modelo para cada una de las observaciones "predictions". Como se puede observar, los puntos que representan cada una de las observaciones están muy cerca de la línea roja, lo que significa que los modelos cometen

muy pocos errores, pues esta representa los valores en los que la predicción es igual al valor original.

El diagrama de dispersión de la derecha es la representación de la predicción realizada por los modelos sobre la base de datos de los outliers. Es aquí donde se puede ver el fenómeno antes descrito por el que se decidió eliminar la variable *"rent_price*". En estas se puede ver como observaciones que originalmente tienen un precio de compra muy alto, el modelo predice unos valores muy muy bajos y es por eso por lo que en el gráfico aparece esa forma de "c invertida". A la hora de analizar la base de datos, se ve que esas observaciones, por algún motivo tienen un "*rent_price*" muy bajo que es lo que hace que el modelo prediga de esta forma. En la figura 30, que muestra los valores de algunas observaciones extraídas de la base de datos de outliers, se ve que para observaciones que tienen un precio de compra muy alto (mayor de 1.000.000€) el precio de alquiler es sorprendentemente bajo, valores de entre 150 y 450€, cuando la media del mercado es de 1.354€. Quizás es por este motivo por el que el Isolation Forest las detecta como outliers. Sin embargo, esto también muestra que el modelo está totalmente influenciado por esta variable y su predicción solo dependía de ella.

	buy_price	rent_price
33	1370000	169
34	1370000	169
35	1365000	242
36	1365000	242
37	1365000	242
38	1363000	270
39	1360000	313
40	1360000	313
41	1360000	313
42	1360000	313
43	1357000	355

44	1350000	452
45	1350000	452
46	1350000	452
47	1350000	452
48	1350000	452
49	1350000	452
50	1350000	452
51	1350000	452
52	1350000	452
53	1350000	452
54	1350000	452
55	1350000	452

Figura 30.Muestra de datos extraída del conjunto de observaciones clasificados como outliers. *Elaboración Propia.*

Por todo ello, se decidió prescindir de esta variable, y entrenar los modelos sin ella. Aunque esto conlleva a que los errores cometidos sean más altos, los resultados obtenidos sin la variable son más coherentes y con más sentido que los que se generaban al incluirla.

4.3. PREDICCIÓN OUTLIERS

A continuación, siguiendo con la metodología descrita, es el momento de aplicar el modelo escogido a los atípicos y tratar de identificar cuáles son posibles observaciones sobrevaloradas y cuales posibles infravaloradas.

Si bien es cierto, que, si se asume que el modelo entrenado/ajustado predice correctamente, esta identificación de observaciones sobrevaloradas e infravaloradas también se puede realizar dentro de la base de datos sin outliers, pero tal y como se especificó ya al principio del trabajo, en este, se buscaba hacer una identificación de outliers y ver si dentro de estos se pueden identificar grandes oportunidades de inversión y que esta sea una de las posibles causas por las cuales estén siendo identificados como outliers.

Para hacer esto, lo primero que se ha hecho es aplicar el modelo de Random Forest previamente entrenado a las 2.803 observaciones que se habían identificado como outliers. A continuación, si se supone que el modelo predice correctamente, se observaban las diferencias entre las predicciones y el valor real, obteniendo así los errores. Por último, se calcula el porcentaje que representa ese error para poder analizar los resultados.

En el diagrama de dispersión de la figura 31 se pueden observar los resultados de la aplicación de este modelo a la base de datos de outliers generada. A simple vista y de manera general se podría decir que hay más observaciones sobrevaloras, que son las que están por encima de la línea roja, que observaciones infravaloradas, por debajo de dicha línea.

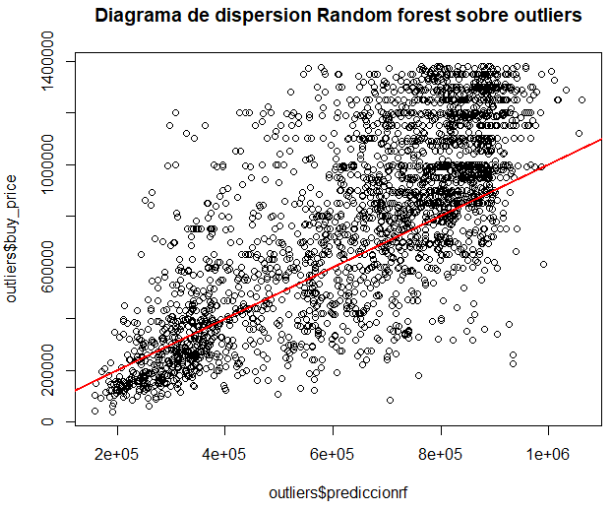

Figura 31.Resultado predicción Random Forest sobre outliers. *Elaboración Propia.*

Como se puede ver, aquí la diferencia entre predicción y observación es mucho más grande que con los datos de la base de datos "limpia". La raíz cuadrada de la media de estas diferencias al cuadrado, en este caso es de 279.747€. Esto en realidad era de esperar ya que el modelo está siendo

aplicado sobre unas observaciones que han sido consideradas como outliers o atípicos debido a alguna característica. Por si acaso, sí que es cierto que se probó a utilizar el modelo de XGBoost, que es el segundo que mejor predice de entre los entrenados. Sin embargo, como era de esperar, los resultados fueron peores. Aquí el cálculo de la diferencia fue de 287.759€.

Si se asumiera que el modelo predice perfectamente y se le llama error a esta diferencia entre el precio que tienen las observaciones en el mercado y el precio que predice el modelo, se podría decir que todos aquellos que tienen un error por encima de cero van a ser clasificados como inmuebles sobrevalorados, mientras que todos aquellos que tengan un error por debajo de cero, van a ser observaciones infravaloradas. En un primer momento, para hacer un primer filtrado, esto fue lo que se hizo, seleccionar como "Sobrevalorado" todas aquellas observaciones cuyo error era mayor que 0 y como "Infravalorado" a todos aquellos con un error menor que 0. Sin embargo, para ser más realistas, una vez separadas las observaciones entre sobrevaloradas e infravaloradas en términos generales, fue necesario establecer algún criterio para considerar cuáles realmente dentro de cada grupo deben considerarse como tal y que deberían hacer saltar la alarma de análisis, ya que, en verdad, el modelo no predice con una perfección del 100%, por lo tanto, cierto error sí que se debe permitir.

Dentro de las observaciones "sobrevaloradas", aparecen 1.908 inmuebles. El "*buy_price*" medio de estos es de 956.762€ y la mayor parte son pisos (el 44%) seguido de casas (el 27%). La media de diferencias entre el precio real y el precio predicho (lo que se llamará "el error") es de 261.091,7€, llegando a aparecer una diferencia máxima de 892.217€. Debido a esto, se van a considerar observaciones realmente sobrevaloradas, y que por lo tanto deberían ser un descarte inmediato de compra, aquellas que cuenten con un error mayor de 300.000€, aunque las restantes habría que analizarlas de manera detallada.

En cuanto a las "infravaloradas", se observan 895 inmuebles cuyo precio medio es de 416.851€. En cuanto al valor de los errores, la media está en -132.419,9€, este valor es negativo ya que representa que se está vendiendo por un valor menor del que el modelo considera que valen. La diferencia más grande de todas llega a los -708.310,8€ pero la mediana se mantiene en los -102.421,9€. Por lo tanto, se ha decido considerar

verdaderos inmuebles infravalorados aquellos que tienen un error que supere los 200.000€ en valor negativo.

Una vez se filtran y se cogen los que se consideran verdaderamente sobrevalorados e infravalorados se obtienen 759 y 182 observaciones respectivamente, guardados en los conjuntos de datos *"sobrevalorados_real"* e *"infravalorados_real".* Ahí quedarían almacenados para poder consultarlos y analizarlos en más detalle si se quiere. En el Anexo II, se pueden ver ejemplos de algunas observaciones pertenecientes a cada uno de estos conjuntos.

Dentro de los verdaderos sobrevalorados se identifican viviendas que pertenecen la mayoría a la categoría pisos o casas, que cuentan con 4 habitaciones y 3 baños de media; 254 metros cuadrados; que no suelen necesitar renovación; la mayoría con parking y terraza; casi la mitad con piscina y están sobrevalorados de media un 38,7% por encima del precio que deberían tener, alcanzando un precio de compra (buy_price) medio de 1.182.291€.

Por otro lado, los pisos infravalorados y que más podrían interesar, son aquellos que tienen de media 200 metros cuadrados, la mayoría pisos o casas, con 3-4 habitaciones y 2-3 baños de media, que no suelen necesitar renovación y tienen terraza. Estos tienen un precio medio de compra de 358.110€ lo que supone que de media estén infravalorados en un -107,51% por debajo del precio que se predice que deberían tener.

Una vez descritas las características de estos, se puede ver que estas clasificaciones tienen sentido. Viviendas que al fin y al cabo tienen unas características bastante similares, luego tienen unos precios de venta muy dispares y que se alejan mucho de lo que realmente establece el mercado.

Este modelo de predicción se podría usar ahora para detectar más viviendas de las que se obtuvieron inicialmente en la base de datos. Sin embargo, es importante mencionar que el precio de los inmuebles está también influenciado por muchos otros factores que no se han recogido en el presente trabajo, algunos de ellos podrían ser la oferta y demanda del momento, la zona geográfica donde estén ubicados, la orientación, lo lejos o cerca que estén de lugares de interés básico como puede ser un supermercado o una farmacia o las expectativas de crecimiento de la zona. También la inflación u otros

factores económicos como los que se mencionaban al principio pueden marcar grandes diferencias en los precios. Es por todo ello, que es importante que las viviendas que se vayan a comprar con este modelo sean recogidas bajo un mismo marco temporal. Y sería bueno que, cada vez que se busque hacer una predicción, se entrene el modelo con datos de ese mismo año para tratar de reducir el ruido y diferencias que se generan al pasar los años debido a elementos como los recién mencionados.

5. **CONCLUSIONES**

A modo de conclusión, se podría decir que en ese trabajo se ha logrado detectar a partir de una base de datos con las características de viviendas del área geográfica de Madrid cuáles de esos inmuebles estaban siendo vendidos por un precio mayor y un precio menor al que distintos modelos de predicción estimaban. La base de datos partía con 21.709 observaciones y 47 variables, que tras un análisis detallado y una limpieza profunda se acabó trabajando con 18.688 observaciones y 10 variables. Es por ello por lo que este proceso de limpieza y análisis de los datos iniciales, a pesar de que pueda suponer gran parte del tiempo de trabajo, cobra gran relevancia de cara a obtener unos resultados lo más precisos y reales posibles.

Desde un primer momento, se partió con el objetivo de ser capaces de aislar las observaciones que, debido a algunas de sus características, el Isolation Forest (método de detección de anomalías muy interesante) identificaba que se desviaban de las demás y las consideraba como valores atípicos. Dentro de estas, se buscaba identificar las que estuviesen sobrevaloradas, suponiendo un descarte inmediato hacia una posible inversión y cuáles infravaloradas, generando interés y un potencial activo de inversión. Esto se realizó sobre los datos atípicos y no sobre los datos limpios, lo cual también se puede hacer, con el fin de que el modelo se aplicase sobre unos datos en los cuales una de las causas por las que hayan sido clasificados como tal haya podido ser que su precio de compra estuviese alejado de la realidad.

Existen varios modelos de regresión que se pueden utilizar para alcanzar el objetivo establecido, pero tras haber entrenado varios de ellos, se ha escogido el de Random Forest por su precisión y menor error. A la hora de analizar sus resultados comparando

la predicción con los datos de validación (test set) se obtuvo un RMSE de 122.644,9€, el más bajo de los cuatro modelos entrenados. También hay que mencionar que con este modelo y con las 9 variables, metros cuadrados, número de baños, número de habitaciones, renovación necesaria, certificado energético, si tiene piscina, si tiene parking, si tiene terraza y tipo de vivienda, se explica el 71,16% de la variabilidad del precio de compra de una vivienda en Madrid para el año 2020.

Tras el entrenamiento del modelo y predicción sobre los outliers, sumado a una serie de criterios de selección, se seleccionaron 759 observaciones como viviendas sobrevaloradas y 182 como infravaloradas. Es normal que existan muchas más viviendas sobrevaloradas que infravaloradas ya que, en circunstancias normales, las personas tienden a intentar sacar el máximo beneficio a cualquier activo del que dispongan.

Las viviendas infravaloradas encontradas tienen un precio promedio de 358.110€, 200 metros cuadrados, 2-3 baños, 3-4 habitaciones y mayormente con terraza mientras que las sobrevaloradas su precio medio es de 1.182.291€, tienen 254 metros cuadrados, 4 habitaciones, 3 baños, no necesitan renovación y la mitad de ellas cuentan con piscina.

Los datos analizados fueron extraídos de una base de datos basada en ofertas del año 2020, por lo cual es muy importante que, si se quieren predecir nuevos precios de compra para inmuebles para el día de hoy, no se utilice este modelo tal cual. Es necesario volver a entrenarlo con observaciones que reflejen el mercado actual del momento. A lo largo del trabajo, se ha mencionado en varias ocasiones que los precios de compra de las viviendas están influenciados por diversos factores, algunos de los cuales no están incluidos en el modelo como variables. Por esta razón, será más preciso si se entrena el modelo con la mayor cantidad de datos disponibles en el mismo periodo de tiempo. De esta manera, se podrán capturar, en cierta medida, algunos de los efectos externos del momento y generar un modelo más preciso.

Con herramientas como estas, la vida de las personas se puede ver facilitada, especialmente en momentos cruciales como en la compra de una vivienda o a la hora de hacer una inversión inmobiliaria. De esta forma, es posible ahorrar mucho tiempo y dinero, dos recursos muy valorados en una sociedad como la que vivimos.

BIBLIOGRAFÍA

Alkan, T., Dokuz, Y., Ecemis, A., Bozdag, A., & Durduran, S. (2022). *Using Machine Learning Algorithms for Predicting Real Estate Values in Tourism Centers*. Research Square. https://doi.org/10.21203/rs.3.rs-1757533/v1

Alves, P., & Urtasun, A. (2019). Evolución reciente del mercado de la vivienda en España. *Banco de España*. https://www.bde.es/wbe/es/publicaciones/publicaciones-discontinuadas/articulos-analiticos/evolucion-reciente-mercado-vivienda-espana.html

Amat, J. (2016). *Introducción a la Regresión Lineal Múltiple*. Cienciadedatos. https://www.cienciadedatos.net/documentos/25_regresion_lineal_multiple

Amat, J. (2020). *Detección de anomalías: Isolation Forest*. Cienciadedatos. https://www.cienciadedatos.net/documentos/66_deteccion_anomalias_isolationforest.html#:~:text=Isolation%20Forest%20es%20una%20m%C3%A9todo,clasificaci%C3%B3n%20y%20regresi%C3%B3n%20Random%20Forest.

Aparicio, L. (2023, 1 de abril). Cambio de ciclo en vivienda: el vendedor ya negocia rebajas de precio. *Cinco Días*. https://cincodias.elpais.com/mercados-financieros/2023-04-01/cambio-de-ciclo-en-vivienda-el-vendedor-ya-negocia-rebajas-de-precio.html

Atanasovski, M.,Kostov, M., Arapinoski, B. & Spirovski, M. (2020). *K-Nearest Neighbor Regression for Forecasting Electricity Demand.* IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9232768

Avanija, J., Sunitha, G., Madhavic, K. R., Korad, P. & Hitesh, R. (2021*). Prediction of House Price Using XGBoost Regression Algorithm*. Turkish Journal of Computer and Mathematics Education, Vol.12, No.2, pp. 2151 – 2155. https://turcomat.org/index.php/turkbilmat/article/download/1870/1615/3517

Baldominos, A., Blanco. I, Moreno, A. J., Iturrarte, R., Bernárdez, O. & Afonso, C. (2018). Identifying Real Estate Opportunities Using Machine Learning. *Applied Sciences,* 8(11), 2321. https://doi.org/10.3390/app8112321

Breiman, L. (2021). Random Forest. *Statistics Department University of California Berkeley*. pp. 1-12. https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

Chen, T. & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785-794. DOI: 10.1145/2939672.2939785

Cortes, D. (2022). isotree: Isolation-Based Outlier Detection. R package version 0.5.17. https://CRAN.R-project.org/package=isotree

DataScientest (2021). *Kaggle: todo lo que hay que saber sobre esta plataforma. datascientest.com. https://datascientest.com/es/kaggle-todo-lo-que-hay-que-saber-sobre-esta-plataforma#:~:text=Kaggle%20es%20una%20plataforma%20web,al%20m%C3%A1ximo%20en%20data%20science*.

Departamento Análisis Bankinter (2023, 17 de marzo). *Los precios de vivienda deberían entrar en fase de ajuste en 2023*. Bankinter. https://www.bankinter.com/blog/finanzas-personales/prevision-precio-vivienda

Eseiza, P. (2023, 5 de mayo). Mercado inmobiliario en España ¿qué nos depara?. *HelpMyCash*. https://www.helpmycash.com/cat/inmobiliarias/analisis-mercado-inmobiliario/

Espinosa-Zúñiga, J. J. (2020). *Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito*. Ingeniería, investigación y tecnología 21(3). https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432020000300002

Euribor rates, (s.f.). Tipos de interés Euribor por año. Euribor-rates.eu. https://www.euribor-rates.eu/es/tipos-de-interes-euribor-por-ano/2019/

Ferrero, R. (2017). Selección paso a paso e importancia de los predictores. *Máxima Formación.* https://www.maximaformacion.es/blog-dat/seleccion-paso-a-paso-e-importancia-de-los-predictores/

Forteza, N. (2019). Isolation Forest: el algoritmo estrella para detección de anomalías. *Medium*. https://medium.com/@keepler.io/isolation-forest-el-algoritmo-estrella-para-detecci%C3%B3n-de-anomal%C3%ADas-416bb5892f10

Gilbert, P. (2022). *Set (Normal) Random Number Generator and Seed*. CRAN. https://cran.r-project.org/web/packages/setRNG/setRNG.pdf

Hawkins D. (1980) *Identification of Outliers*. Chapman and Hall.

INE (2020*). Los hogares y la vivienda en la Unión Europea.* Productos y Servicios / Publicaciones / Colección Cifras INE. Instituto Nacional de Estadística. Productos y Servicios / Publicaciones / Colección Cifras INE

INE (2022). *Estadística de Transmisiones de Derechos de la Propiedad. Compraventa de viviendas según régimen y estado*. Instituto Nacional de Estadística. https://www.ine.es/jaxiT3/Tabla.htm?t=6150&L=0

INE (2022). *Índice de Precios de Vivienda (IPV). Base 2015 Cuarto trimestre de 2022.* Nota de prensa. Instituto Nacional de Estadística. https://www.ine.es/daco/daco42/ipv/ipv0422.pdf

Kaggle (s.f.). *Madrid real estate market. Real estate listings in Madrid crawled from popular internet portals*. Kaggle. https://www.kaggle.com/datasets/mirbektoktogaraev/madrid-real-estate-market

Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York, NY. https://doi.org/10.1007/978-1-4614-6849-3

Liu, Fei Tony, Kai Ming Ting y Zhi-Hua Zhou (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining. IEEE*, pp. 413-422. doi: 10.1109/ ICDM.2008.17

López, A. (2022, 8 julio). Aumentan los propietarios en España: 8 de cada 10 españoles tiene una casa en propiedad. *Fotocasa Life*. https://www.fotocasa.es/fotocasa-life/comprar-piso/aumentan-los-propietarios-en-espana-8-de-cada-10-espanoles-tiene-una-casa-en-propiedad/

López, S. (2023). La vivienda se desinfla: las compras caen un 13% y las hipotecas bajan un 24% en febrero. *El País*. https://elpais.com/economia/negocios/2023-04-13/la-vivienda-se-desinfla-las-compras-caen-un-13-y-las-hipotecas-bajan-un-24-en-febrero.html

Martínez Barbero, X. (2019). Predicción del precio de la vivienda mediante redes neuronales artificiales en la ciudad de Madrid. *Universidad Politécnica de Valencia*. http://hdl.handle.net/10251/125320

Martínez, C. (2021). *Machine Learning I. KNN*. PowerPoint. Universidad Pontificia Comillas.

Martínez, C. (2021). *Machine Learning I. Random Forest*. PowerPoint. Universidad Pontificia Comillas.

Martínez, C. (2021). *Machine Learning I. Regresión I*. PowerPoint. Universidad Pontificia Comillas.

Moreno, M. G. (2021, 7 de agosto). Así ha cambiado el mercado de la vivienda en un año de pandemia. *elEconomista.es.* https://www.eleconomista.es/vivienda/noticias/11352826/08/21/Asi-ha-cambiado-el-mercado-de-la-vivienda-en-un-ano-de-pandemia.html

Muñoz, J. A. (2013). Técnicas para detección de outliers multivariantes. Revista en Telecomunicaciones e Información. Vol. 3, No. 5, p. 11-25, ISSN2215-8200. https://repository.upb.edu.co/bitstream/handle/20.500.11912/6582/T%C3%A9cnicas%20para%20detecci%C3%B3n%20de%20outliers%20multivariantes.pdf?sequence=1&isAllowed=y#:~:text=La%20detecci%C3%B3n%20de%20outliers%20s%C3%B3lo,medidas%20en%20un%20espacio%20bidimensional.

Ortega, J.M. (2021). *Certificado energético «en trámite» ¿qué significa?* Certicalia. https://www.certicalia.com/blog/certificado-energetico-en-tramite-que-significa

Qué es el Certificado de Eficiencia Energética? (2023). Certificado de Eficiencia Energética. https://certificadodeeficienciaenergetica.com/que-es-certificado-eficiencia-energetica-definicion

Sawant, R., Jangid, Y., Tiwari, T., Jain, S. & Gupta, A. (2018). *Comprehensive Analysis of Housing Price Prediction in Pune using Multi-Featured Random Forest Approach.* Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8697402

Taunk, K., De, S., Verma, S. & Swetapadam, A. (2019*). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification.* IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9065747

Utrera, E. (2023, 9 de mayo). El precio medio de las hipotecas salta a barrera del 3.5%. *Expansión*. https://www.expansaion.com/ahorro/2023/05/09/6459996be5fdea3b748b4653.html

6. **ANEXOS**

ANEXO I. Resumen parte del código de Rstudio utilizado.

```
#Codigo utilizado para la realización del tfg. Rafael Loureiro Alvarez
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
getwd()#establacer directorio de trabajo
#Carga de librerias necesarias
library(dplyr)
library(readxl)
library(corrplot)
library(caret)
library(rpart)
library(car)
library(ROCR)
library(NeuralNetTools)
library(nnet)
library(e1071)
library(ranger)
library(randomForest)
library(caretEnsemble)
library(ggplot2)
library(factoextra)
library(reshape2)
library(cluster)
library(isotree)
library(xgboost)

#Carga fichero xlsx con la base de datos de las viviendas de Madrid extraida de Kaggle.
datos <- read_excel("Base_datos_vivienda_Madrid_boolean.xlsx")
View(datos)
summary(datos)
```

```
sapply(datos,(anyNA))

apply(X = is.na(datos), MARGIN = 2, FUN = sum) #para identificar que variables tienen pocos valores faltantes

str(datos)


#eliminar las filas con pocos valores faltantes

datos <- datos[!is.na(datos$n_bathrooms),] #borra 16 observaciones

datos <- datos[!is.na(datos$sq_mt_built),] #borra 115 observaciones

datos <- datos[!is.na(datos$house_type_id),] #borra 389 observaciones


#Elimina aquellas entradas que tienen un precio de alquiler negativo, ya que no tienen sentido.

datos <- datos[datos$rent_price>0,]

#eliminar aquellas con un numero de baños sin sentido

datos<- datos[datos$n_bathrooms<14,]



#TRANSFORMACION PARA CAMBIAR EL TRUE POR 1

datos$has_garden <- ifelse(datos$has_garden == 'True',1,0)

datos$has_garden <-ifelse(is.na(datos$has_garden), 0, 1)

table(datos$has_garden)


#SELECCION DE VARIABLES CON LAS QUE VAMOS A TRABAJAR

datos <- mutate_at(datos, c("has_pool","has_terrace"), ~replace(., is.na(.), 0))

#SIN RENT_PRICE

datos3<- datos[,c("sq_mt_built","n_rooms","n_bathrooms", "buy_price", "is_renewal_needed", "energy_certificate", "has_parking", "house_type_id", "has_pool","has_terrace")]


#convertir en variables FACTOR para el one hot encoding

datos3$energy_certificate<-as.factor(datos$energy_certificate)

summary(datos3$energy_certificate)
```

```
datos3$house_type_id<-factor(datos$house_type_id)

summary(datos3$house_type_id)


#################################################################

####ESTUDIO DE LAS VARIABLES#####################################

#Analizar correlaciones de las variables.

datosnum <- datos3[,sapply(datos3, is.numeric)]

correlation<-cor(datosnum)

corrplot(cor(datosnum), method = "number", main = "Correlación variables
numéricas")


#Para comprobar si podemos hacer log de variables con asimetria hacia la derecha:

any(datos3$sq_mt_built==0)

hist(datos$sq_mt_built, main = "Histograma metros cuadrados construidos",
xlab="Metros cuadrados", ylab="Frecuencia",col = "#80C6DF")

any(datos3$n_rooms==0)

hist(datos$n_rooms)


any(datos3$buy_price==0)

hist(datos_limpios$buy_price, main = "Histograma precio de compra (en €)",
xlab="Euros", ylab="Frecuencia", col = "#80C6DF") #vemos que buy price si que tiene
asimetria a la derecha.

boxplot(datos$buy_price)

hist(datos$n_bathrooms)

hist(datos$rent_price)


boxplot(buy_price~n_rooms, data=datos3)


#energy

tabla_energy<-table(datos3$energy_certificate)

media_por_categoria <- aggregate(buy_price ~ energy_certificate, data = datos, FUN =
mean)

boxplot(buy_price ~ energy_certificate, data = datos, col = "#80C6DF")
```

```
lines(x = 1:10, y = media_por_categoria$buy_price, col = "red")

pie(tabla_energy, labels = paste(names(tabla_energy), ":", tabla_energy), col = c("red",
"lightgreen", "blue","orange","yellow","#80C6DF","grey","purple","brown","white"))


#house type

boxplot(buy_price~house_type_id, data=datos, col = "#80C6DF") #OJO A LOS PISOS
!!!!#

tabla_house<-table(datos3$house_type_id)

pos <- barplot(tabla_house,

        xlab = "Tipo de inmueble",

        ylab = "Cantidad",

        main = "Número de inmuebles por categoría",

        names.arg = names(tabla_house), col = "#80C6DF")

text(x = pos,    y = tabla_house,  labels = tabla_house,   pos = 3,cex = 1.5,xpd = TRUE)


#sq_mt_built

boxplot(datos$sq_mt_built, col = "#80C6DF", main = "Diagrama de caja metros
cuadrados construidos")

scatterplot(datos$buy_price, datos$sq_mt_built)


#rent_price

hist(datos3$rent_price, main = "Histograma precio de alquiler", xlab="Precio",
ylab="Frecuencia",col = "#80C6DF")

summary(datos3$rent_price)

boxplot(datos3$rent_price)

cor(datos3$rent_price, datos3$buy_price) #vemos que son variables correlacionadas


#renewal

t<-datos3$is_renewal_needed

mi_variable_factor <- factor(t, labels = c("No", "Sí"))

tablarenew<-table(mi_variable_factor)

pie(tablarenew,labels = paste(names(tablarenew), ":", tablarenew), col =
c("brown","white"), main = "Renovación necesaria")
```

```
#has_pool

t<-datos3$has_pool

mi_variable_factor <- factor(t, labels = c("No tiene ", "Sí tiene "))

tablarenew<-table(mi_variable_factor)

pie(tablarenew,labels = paste(names(tablarenew), ":", tablarenew), col =
c("#80C6DF","white"), main = "Piscina")


#has_terrace

t<-datos3$has_terrace

mi_variable_factor <- factor(t, labels = c("No tiene ","Sí tiene "))

tablarenew<-table(mi_variable_factor)

pie(tablarenew,labels = paste(names(tablarenew), ":", tablarenew), col =
c("grey","#FFB90F"), main = "Terraza")


#has_parking

t<-datos3$has_parking

mi_variable_factor <- factor(t, labels = c("No tiene ","Sí tiene "))

tablarenew<-table(mi_variable_factor)

pie(tablarenew,labels = paste(names(tablarenew), ":", tablarenew), col =
c("grey","blue"), main = "Parking")

boxplot(buy_price ~has_parking, data=datos3, col = "#80C6DF")


#has_garden

t<-datos$has_garden

mi_variable_factor <- factor(t, labels = c("No", "Sí"))

tablarenew<-table(mi_variable_factor)

pie(tablarenew,labels = paste(names(tablarenew), ":", tablarenew), col =
c("#80C6DF","green"), main = "Tiene jardín ")

#####################################################################


#LIMPIEZA DE OUTLIERS:
```

```
######Opcion definitiva ISOLATION FOREST-----
iso<- isolation.forest(datos3, ntrees = 1000)
prediccion<- predict(iso, datos3, type = "score")

hist(prediccion, breaks = 30)
axis(1, at = seq(0, 1, 0.01))
boxplot(prediccion)

umbral <- quantile(prediccion, 0.85) #considerar poner quantil 0.75
umbral

# Identificar las observaciones que son outliers
outliers <- datos3[prediccion > umbral, ]
nrow(outliers)

hist(outliers$buy_price)
boxplot(outliers$buy_price)

cuantiles <- quantile(x = prediccion, probs = seq(0, 1, 0.05))
cuantiles

datos_limpios <- anti_join(datos3, outliers)
View(datos_limpios)
hist(datos_limpios$buy_price)
str(datos_limpios)

#########################################################################

#PARTICION DE LOS DATOS EN TRAIN Y TEST PARA MODELO 6 QUE CONSIDERA LA
ASIMETRÍA A LA DERECHA DEL BUY_PRICE
RNGkind("Super", "Inversion", "Rounding")
```

```
set.seed(123)

index<-createDataPartition(datos_limpios$buy_price, p = 0.7, list=FALSE)

train<- datos_limpios[index,]

test<-datos_limpios[-index,]


#Modelo regresión lineal con trasnformación logarítmica en variable dependiente

modeloreg<-
lm(log(buy_price)~sq_mt_built+n_rooms+n_bathrooms+energy_certificate+house_type_id+has_terrace, data = train)

summary(modeloreg) #r2=0.6285

#plot(modeloreg)


#Obtencion de las PREDICCIONES

predicciones<-predict(modeloreg, newdata = test)

prediccionesreales<- exp(predicciones)

#Obtencion de los erroes

errores<-test$buy_price-prediccionesreales

boxplot(errores)

hist(errores, breaks=20)

h<-nrow(test)


plot(test$buy_price)

points(prediccionesreales, col="green")

#Obtencion de las medidas de comparacion

MAE<-sum(abs(errores))/h

RMSE<-sqrt(sum(errores^2)/h) #181,265.1

RMSE

AIC(modeloreg) #10674.74

BIC(modeloreg) #10821.07
```

```
plot(prediccionesreales, test$buy_price)
lines(test$buy_price, test$buy_price, col="red")

plot(test$buy_price, test$buy_price-prediccionesreales)
lines(test$buy_price, 0*test$buy_price, col="red")

####ANALISIS DE OUTLIERS####
outliers$prediccion <- predict(modeloreg, newdata = outliers)
outliers$prediccion<- exp(outliers$prediccion)
outliers$error<- (outliers$buy_price - outliers$prediccion)
outliers$error_porcent<- ((outliers$buy_price -
outliers$prediccion)/outliers$buy_price)*100
out<- outliers[,c("buy_price", "prediccion", "error","error_porcent")]
View(out)

plot(outliers$prediccion, outliers$buy_price, xlim = c(400000, 10000000))
lines(outliers$buy_price, outliers$buy_price, col="red")

plot(outliers$buy_price, outliers$buy_price-outliers$prediccion)
lines(outliers$buy_price, 0*outliers$buy_price, col="red")
abline(v = 1000000, col = "red")

#PARTICION DE LOS DATOS EN TRAIN Y TEST PARA MODELO 7 EN EL QUE NO SE
CONSIDERA LA ASIMETRIA A LA DERECHA DE BUY PRICE
RNGkind("Super", "Inversion", "Rounding")
set.seed(123)
index<-createDataPartition(datos_limpios$buy_price, p = 0.7, list=FALSE)
train<- datos_limpios[index,]
```

```
test<-datos_limpios[-index,]

#COSNTRUCCIÓN MODELO

modeloreg<-
lm(buy_price~sq_mt_built+n_rooms+n_bathrooms+is_renewal_needed+has_parking+
energy_certificate+house_type_id+has_pool+has_terrace , data = train)

summary(modeloreg) #r2=0.6873


#Obtencion de las predicciones

predicciones<-predict(modeloreg, newdata = test)

summary(predicciones)

sum(predicciones<0)

#Obtencion de los erroes

errores<-test$buy_price-predicciones

boxplot(errores)

hist(errores, breaks=20)

h<-nrow(test)


plot(test$buy_price)

points(predicciones, col="green")

#Obtencion de las medidas de comparacion

ME<-sum(errores)/h

RMSE<-sqrt(sum(errores^2)/h) #131,660.2

RMSE

AIC(modeloreg)#293748.8

BIC(modeloreg)#293902.5



plot(predicciones, test$buy_price, main = "Diagrama de dispersión 'buy_price'
regresión lineal" )

lines(test$buy_price, test$buy_price, col="red")


plot(test$buy_price, test$buy_price-predicciones)
```

```
lines(test$buy_price, 0*test$buy_price, col="red")


####ANALISIS DE OUTLIERS####

outliers$prediccion <- predict(modeloreg, newdata = outliers)

outliers$error<- (outliers$buy_price - outliers$prediccion)

outliers$error_porcent<- ((outliers$buy_price -
outliers$prediccion)/outliers$buy_price)*100

out<- outliers[,c("buy_price", "prediccion", "error","error_porcent")]

View(out)

hist(out$error_porcent)

sum(outliers$prediccion<0)


plot(outliers$prediccion, outliers$buy_price)

lines(outliers$buy_price, outliers$buy_price, col="red")


plot(outliers$buy_price, outliers$buy_price-outliers$prediccion)

lines(outliers$buy_price, 0*outliers$buy_price, col="red")


##------------------------------------------------------------------------------------------------------------

##----------------------[ RANDOM FOREST (modelo con mejor performance) ]--------------

##------------------------------------------------------------------------------------------------------------

#Partición de los datos

RNGkind("Super", "Inversion", "Rounding")

set.seed(123)

index<-createDataPartition(datos_limpios$buy_price, p = 0.7, list=FALSE)

train<- datos_limpios[index,]

test<-datos_limpios[-index,]

# Construcción del modelo y predicción

model_rdmfs <-
randomForest(buy_price~sq_mt_built+n_rooms+n_bathrooms+is_renewal_needed+h
as_parking+energy_certificate+house_type_id+has_pool+has_terrace, data = train,
ntree=500, trainControl=control)
```

```
predictions <- predict(model_rdmfs, newdata = test)

RMSEtestknn<-RMSE(predictions, test$buy_price) #122,644.9
RMSEtestknn
r_squaredrf <- cor(predictions, test$buy_price)^2 #0.7116512


# Obtener la importancia de las variables
var_importance <- importance(model_rdmfs)
# Graficar la importancia de las variables
varImpPlot(model_rdmfs, main="Importancia variables modelo Random Forest")

# Obtener las predicciones y los residuos del modelo
pred <- predict(model_rdmfs, datos_limpios)
residuals <- pred - datos_limpios$buy_price

# Graficar los residuos versus los valores ajustados
plot(pred, residuals, xlab = "Predicted values", ylab = "Residuals")

plot(predictions, test$buy_price, main="Diagrama de dispersión 'buy_price' Random
Forest")
lines(test$buy_price, test$buy_price, col="red")

plot(test$buy_price, test$buy_price-predictions)
lines(test$buy_price, 0*test$buy_price, col="red")


####ANALISIS DE OUTLIERS####
outliers$prediccionrf <- predict(model_rdmfs, newdata = outliers)
outliers$errorrf<- (outliers$buy_price - outliers$prediccionrf)
```

```
outliers$error_porcentrf<- ((outliers$buy_price -
outliers$prediccionrf)/outliers$buy_price)*100

out<- outliers[,c("buy_price", "prediccionrf", "errorrf","error_porcentrf")]

View(out)


hist(outliers$errorrf)

axis(1, at = seq(-1000000, 1000000, 10000))

hist(outliers$error_porcentrf, xlim=c(250,-250))

sum(outliers$prediccionrf<0)

sum(abs(outliers$error_porcentrf)>50)

sum((outliers$errorrf)>300000)

sum((outliers$errorrf)< -300000)



rmserf<-sqrt(mean((outliers$prediccionrf - outliers$buy_price)^2))

rmserf #279747.6

plot(outliers$prediccionrf, outliers$buy_price, main="Diagrama de dispersion Random
forest sobre outliers")

lines(outliers$buy_price, outliers$buy_price, col="red")

abline(h = 1200000, col = "red")


plot(outliers$buy_price, outliers$buy_price-outliers$prediccionrf)

lines(outliers$buy_price, 0*outliers$buy_price, col="red")



#Clasificacion general entre sobrevalorados e infravalorados

outliers$valoracionrf <- ifelse(outliers$errorrf > 0, "Sobrevalorado", "Infravalorado")

pisos_sobrevalorados_rdmfs <- subset(outliers, valoracionrf == "Sobrevalorado")

pisos_infravalorados_rdmfs <- subset(outliers, valoracionrf == "Infravalorado")


#sobrevalorados
```

```
summary(pisos_sobrevalorados_rdmfs) #comentar el precio medio de estos y el tipo que más hay

boxplot(pisos_sobrevalorados_rdmfs$errorrf)

hist(pisos_sobrevalorados_rdmfs$error_porcentrf)

hist(pisos_sobrevalorados_rdmfs$errorrf)


#infravalorados

summary(pisos_infravalorados_rdmfs) #comentar el precio medio de estos y el tipo que más hay

boxplot(pisos_infravalorados_rdmfs$errorrf)

hist(pisos_infravalorados_rdmfs$error_porcentrf)

hist(pisos_infravalorados_rdmfs$errorrf)


#Clasificación real de las observaciones

sobrevalorados_real<-
pisos_sobrevalorados_rdmfs[pisos_sobrevalorados_rdmfs$errorrf>300000,]

infravalorados_real<-pisos_infravalorados_rdmfs[pisos_infravalorados_rdmfs$errorrf<
-200000,]

summary(sobrevalorados_real)

summary(infravalorados_real)

barplot(infravalorados_real$n_rooms)

hist(infravalorados_real$sq_mt_built)


##########################################################

#------------------------KNN-----------------------------------------------

#Particion de los datos

RNGkind("Super", "Inversion", "Rounding")

set.seed(123)

index<-createDataPartition(datos_limpios$buy_price, p = 0.7, list=FALSE)

train<- datos_limpios[index,]

test<-datos_limpios[-index,]
```

```
#establacimiento de train controls:

repeats = 3   #veces que se repite el procedimiento de remuestro

numbers = 10   #numero de folds en la cross-validation

tunel = 15    #numero de valores del hiperparametro que se van a probar

x = trainControl(method = "repeatedcv",                 # metodo de remuestreo, en este caso cv repetida; en este caso se va a hacer cross validacion de 10 folds y se va a hacer 3 veces

            number = numbers,                                   # numero de folds

            repeats = repeats,                                  # numero de repeticiones de la cross-validation

            classProbs = TRUE   )                  # se obtienen las probabilidades de clasificacion


#Estimacion del modelo.

modelocompleto <-
train(buy_price~sq_mt_built+n_rooms+n_bathrooms+is_renewal_needed+has_parking+energy_certificate+house_type_id+has_pool+has_terrace , data = train,

                method = "knn",                                                 #algoritmo a usar

                preProcess = c("center","scale"),                   #transformaciones de los datos, aquí se tipifican

                trControl = x,                                                  #como se realiza el proceso de training

                tuneLength = tunel)                                     #numero de valores de K se prueban

# Resumen del modelo

modelocompleto    #K=9   #RMSE: 133,112.1   #r2=0.6809


plot(modelocompleto)    # salen los diferentes valores del hyperparametro k probados y su AUC


# Predicciones en el train set


predtrainknn<-predict(modelocompleto, newdata=train)

RMSEtrknn<-RMSE(predtrainknn, train$buy_price)
```

```
RMSEtrknn  #119,200.5


# Predicciones en test set
predtestknn<-predict(modelocompleto, newdata=test)
RMSEtestknn<-RMSE(predtestknn, test$buy_price)
RMSEtestknn #132,228.6


plot(predtestknn, test$buy_price, main = "Diagrama de dispersión 'buy_price' knn")
lines(test$buy_price, test$buy_price, col="red")



####ANALISIS DE OUTLIERS####
outliers$prediccion <- predict(modelocompleto, newdata = outliers)
outliers$error<- (outliers$buy_price - outliers$prediccion)
outliers$error_porcent<- ((outliers$buy_price -
outliers$prediccion)/outliers$buy_price)*100
out<- outliers[,c("buy_price", "prediccion", "error","error_porcent")]
View(out)
hist(out$error_porcent)
sum(outliers$prediccion<0)



#-------------------------------------------------------------------------------------------------------------
#----------------------------[ XG BOOST (también buena performance) ]----------------------------


#convertir en variables NUMERICAS para el one hot encoding
cert_energetico_dummy <- model.matrix(~energy_certificate-1, data=datos_limpios)
house_type_id_dummy<- model.matrix(~house_type_id-1, data=datos_limpios)
datos_limpios <- cbind(datos_limpios[,c("sq_mt_built","n_rooms","n_bathrooms",
"buy_price", "is_renewal_needed","has_parking",
"has_pool","has_terrace")],house_type_id_dummy, cert_energetico_dummy)
```

```
str(datos_limpios)


cert_energetico_dummy <- model.matrix(~energy_certificate-1, data=outliers)

house_type_id_dummy<- model.matrix(~house_type_id-1, data=outliers)

outliers2 <- cbind(outliers[,c("sq_mt_built","n_rooms","n_bathrooms", "buy_price",
"is_renewal_needed","has_parking",
"has_pool","has_terrace")],house_type_id_dummy, cert_energetico_dummy)

str(outliers2)

#View(outliers2)



#Partición de los datos

RNGkind("Super", "Inversion", "Rounding")

set.seed(123)

index<-createDataPartition(datos_limpios$buy_price, p = 0.7, list=FALSE)

train<- datos_limpios[index,]

test<-datos_limpios[-index,]


#creación de matriz eliminando la variable dependiente buy_price

train_matrix <- xgb.DMatrix(data = as.matrix(train[, -4]), label = train$buy_price)

test_matrix <- xgb.DMatrix(data = as.matrix(test[, -4]))



# Definir los parámetros del modelo

params <- list(

  objective = "reg:linear",

  max_depth = 3,

  eta = 0.3

)

# Entrenar el modelo

xgb_model <- xgb.train(params = params, data = train_matrix, nrounds = 200)
```

```
# Realizar la predicción
predictions <- predict(xgb_model, test_matrix)

msexg <- mean((predictions - test$buy_price)^2)
msexg
rmsexg<-sqrt(mean((predictions - test$buy_price)^2)) #124,499.9
rmsexg
r_squaredxg <- cor(predictions, test$buy_price)^2 #0.7037118
r_squaredxg


plot(predictions, test$buy_price, main="Diagrama de dispersión 'buy_price' XGBoost")
lines(test$buy_price, test$buy_price, col="red")

plot(test$buy_price, test$buy_price-predictions)
lines(test$buy_price, 0*test$buy_price, col="red")


##ANALISIS DE OUTLIERS####
outliers_matrix<- xgb.DMatrix(data = as.matrix(outliers2[, -4]), label =
outliers2$buy_price)
outliers2$prediccionxgb <- predict(xgb_model, newdata = outliers_matrix)
outliers2$errorxgb<- (outliers2$buy_price - outliers2$prediccionxgb)
outliers2$error_porcentxgb<- ((outliers2$buy_price -
outliers2$prediccionxgb)/outliers2$buy_price)*100
out<- outliers2[,c("buy_price", "prediccionxgb", "errorxgb","error_porcentxgb")]
#View(out)
sum(outliers2$prediccionxgb<0)
hist(outliers2$error_porcentxgb, xlim = c(-400,400), breaks = 50)
axis(1, at = seq(-200, 200, 30))
```

```
rmsexg<-sqrt(mean((outliers2$prediccionxgb - outliers2$buy_price)^2)) #287,759
rmsexg


plot(outliers2$prediccionxgb, outliers2$buy_price)
lines(outliers2$buy_price, outliers2$buy_price, col="red")


plot(outliers2$buy_price, outliers2$buy_price-outliers2$prediccionxgb, ylim = c(-
200000, 200000))
lines(outliers2$buy_price, 0*outliers2$buy_price, col="red")
```

ANEXO II. Extracción observaciones bases de datos viviendas infravaloradas y sobrevaloradas.

Observaciones pertenecientes al conjunto de viviendas *"infravalorados_real".*

sq_mt_built	n_rooms	n_bathrooms	buy_price	is_renewal_needed	energy_certificate	has_parking	house_type_id	has_pool	has_terrace
120	5	2	190000	0	F	1	HouseType 1: Pisos	1	1
125	3	3	275000	0	D	0	HouseType 4: Dúplex	1	1
150	2	1	159000	1	F	1	HouseType 2: Casa o chalet	0	1
140	3	3	272500	0	D	0	HouseType 4: Dúplex	1	1
133	3	3	275000	0	D	0	HouseType 1: Pisos	1	1
125	3	3	275000	0	en trámite	0	HouseType 5: Áticos	1	1
125	3	3	275000	0	D	0	HouseType 4: Dúplex	1	1
140	3	3	340000	1	A	1	HouseType 4: Dúplex	1	1
115	3	3	245000	0	en trámite	1	HouseType 4: Dúplex	0	1
242	4	3	380000	0	no indicado	1	HouseType 2: Casa o chalet	0	1
166	3	1	300000	1	no indicado	0	HouseType 2: Casa o chalet	0	0
160	2	1	260000	0	no indicado	1	HouseType 5: Áticos	1	1
125	3	3	275000	0	en trámite	0	HouseType 4: Dúplex	1	1
125	3	3	275000	0	D	0	HouseType 1: Pisos	1	1

Observaciones pertenecientes al conjunto de viviendas *"sobrevalorados_real".*

sq_mt_built	n_rooms	n_bathrooms	buy_price	is_renewal_needed	energy_certificate	has_parking	house_type_id	has_pool	has_terrace
238	5	4	1300000	0	E	1	HouseType 1: Pisos	0	0
176	3	2	1160000	0	en trámite	1	HouseType 5: Áticos	0	0
198	4	5	1290000	0	B	1	HouseType 1: Pisos	0	0
301	5	4	1290000	1	E	1	HouseType 1: Pisos	0	1
218	2	1	885000	1	E	1	HouseType 4: Dúplex	0	0
128	3	3	1119200	0	C	0	HouseType 1: Pisos	0	1
135	2	2	855000	0	F	1	HouseType 5: Áticos	0	1
137	4	2	1150000	0	en trámite	1	HouseType 2: Casa o chalet	0	0
110	2	2	990000	0	en trámite	0	HouseType 4: Dúplex	1	1
101	3	1	629000	0	inmueble exento	0	HouseType 5: Áticos	0	1
138	3	2	1060000	1	A	1	HouseType 1: Pisos	0	0
140	2	1	800000	1	en trámite	0	HouseType 2: Casa o chalet	0	0
148	2	2	1350000	0	no indicado	0	HouseType 1: Pisos	0	0
118	3	2	1050000	0	C	1	HouseType 5: Áticos	0	1