Predicción del mercado inmobiliario en la Comunidad de Madrid

Juan María Guerrero Carrasco

Máster en Ciencia de Datos

Área 5 – Modelos predictivos

Tutor/a de TFM
Jorge Segura Gisbert

Profesor/a responsable de la asignatura
Albert Solé Ribalta

Enero - 2024

Universitat Oberta de Catalunya

Dedicado a mi familia
y en especial a mi hermano Jaime,
futuro compañero de profesión.

Ficha del Trabajo Final

Título del trabajo:	Predicción del mercado inmobiliario en la Comunidad de Madrid
Nombre del autor/a:	Juan María Guerrero Carrasco
Nombre del Tutor/a de TFM:	Jorge Segura Gisbert
Nombre del/de la PRA:	Albert Solé Ribalta
Fecha de entrega:	01/2024
Titulación o programa:	Máster en Ciencia de Datos
Área del Trabajo Final:	Área 5: Modelos predictivos
Idioma del trabajo:	Castellano
Palabras clave	Mercado inmobiliario, predicción de precios, vivienda

Resumen del Trabajo

En los últimos años el mercado inmobiliario ha resultado uno de los más influyentes en la economía y la sociedad, debido a que el acceso a una vivienda es una necesidad vital y un derecho universal, que no siempre resulta de fácil acceso. Llegando incluso a provocar una de las mayores recesiones a nivel mundial de la historia, y que afectó especialmente a España.

El objetivo de este trabajo responde en parte a las necesidades de este mercado, pues se pretende utilizar un conjunto de datos basado en viviendas en venta entre los años 2021 y 2022 para, mediante un proceso de analítica avanzada basado en técnicas de Aprendizaje Automático (ML) tales como estadística descriptiva, minería de datos (DM) o aprendizaje profundo (DL), llegar a la elaboración de modelos que nos puedan proporcionar predicciones fiables sobre el valor de la vivienda.

Dentro del desarrollo del trabajo se incluirán múltiples tareas relacionadas con el ámbito de la Ciencia de Datos, como pueden ser un análisis detallado del potencial de los datos con los que contamos inicialmente; la complementación de estos datos con otros relacionados con la situación económica, que puedan estar relacionados con el precio en sí de la vivienda, así como la limpieza de

estos datos. Una vez estructurados, podremos aplicar nuestros procesos de Machine Learning y Deep Learning para obtener las predicciones deseadas. Finalmente, se llevarán a cabo distintas visualizaciones de los resultados obtenidos, así como un análisis global de estos que pueda determinar en qué medida se predice el valor de la vivienda.

Abstract

In recent years, the real estate market has been one of the most influential in the economy and society, due to the fact that access to housing is a vital necessity and a universal right, which is not always easy to access. It has even provoked one of the biggest recessions worldwide in history, which particularly affected Spain.

The objective of this work responds in part to the needs of this market, as it aims to use a dataset based on homes for sale between 2021 and 2022 to, through a process of advanced analytics based on Machine Learning (ML) techniques such as descriptive statistics, data mining (DM) or deep learning (DL), arrive at the development of models that can provide reliable predictions about the evolution of the market in the future.

The development of the work will include multiple tasks related to the field of Data Science, such as a detailed analysis of the potential of the data we initially have; the complementation of this data with other data related to the economic situation, which may be related to the price of housing itself, as well as the cleaning of this data. Once structured, we will be able to apply our Machine Learning and Deep Learning processes to obtain the desired accuracy. Finally, different visualizations of the results obtained will be carried out, as well as a global analysis of these results that can determine to what extent can we predict housing prices.

Índice

1. Introducción

1.1. Contexto y justificación del Trabajo

A lo largo de este trabajo se pretende realizar un análisis predictivo que nos permita dar pasos en favor de cubrir una necesidad vital, hasta el punto de estar recogida a nivel constitucional en el mundo entero como un derecho universal, y es el derecho a la vivienda.

Mediante el mismo se pretende obtener modelos predictivos que nos permitan entender la situación del mercado inmobiliario a largo plazo, lo cuál puede ser muy útil para futuros inversores, sobre todo teniendo en cuenta que el mercado inmobiliario se ha convertido en uno de los más atractivos en los últimos años, y en especial si consideramos la relación entre su alta rentabilidad y su bajo riesgo.

Pero no sólo es útil desde el punto de vista de aquellos que quieren encontrar la mejor oportunidad para invertir, sino que este creciente interés por parte de grandes inversores también ha ocasionado un grave problema a nivel social, puesto que han subido los precios y cada vez resulta más difícil el poder acceder a la compra de una vivienda para uso propio.

Este problema se ha acentuado de forma especial en la Comunidad de Madrid, área sobre la que se desarrollará el trabajo, y es que cada vez se concentra más población en las grandes ciudades, lo que hace que la demanda de vivienda crezca muy rápido disparando los precios. Esto también supone una gran motivación personal, puesto que en el momento en que se desarrolla este trabajo, soy uno de los afectados por las complicaciones que supone acceder a la compra de una primera vivienda; y el tener una estimación precisa del precio adecuado puede suponer una gran ayuda para dar el paso, así como para todas esas personas que, como yo, se encuentran en la búsqueda de este derecho universal.

Por contextualizar con información del periodo en que se encuentran los datos utilizados (2021 - 2022), según informes de Fotocasa [1], durante este periodo se recuperó el valor de la vivienda hasta valores normales anteriores a la crisis del COVID-19. De hecho, el valor de la vivienda aumentó en 2022 un 7.5%, recordando este aumento a los valores vistos durante la burbuja inmobiliaria en 2006. Siendo en Madrid esta subida aún mayor y llegando al 8.3%. El precio medio del metro cuadrado a final de año en Madrid se situó en 3382€, siendo con diferencia el más elevado de España. Otro dato relevante es que dentro de los distritos de Madrid los que tuvieron una mayor subida fueron Puente de Vallecas (15.8%) y Villaverde (11.6%), siendo también los distritos más humildes y baratos de Madrid junto con Usera. Esto sugiere que el precio de la vivienda en la Comunidad de Madrid está tendiendo a igualarse a la alza como sugiere la siguiente figura:

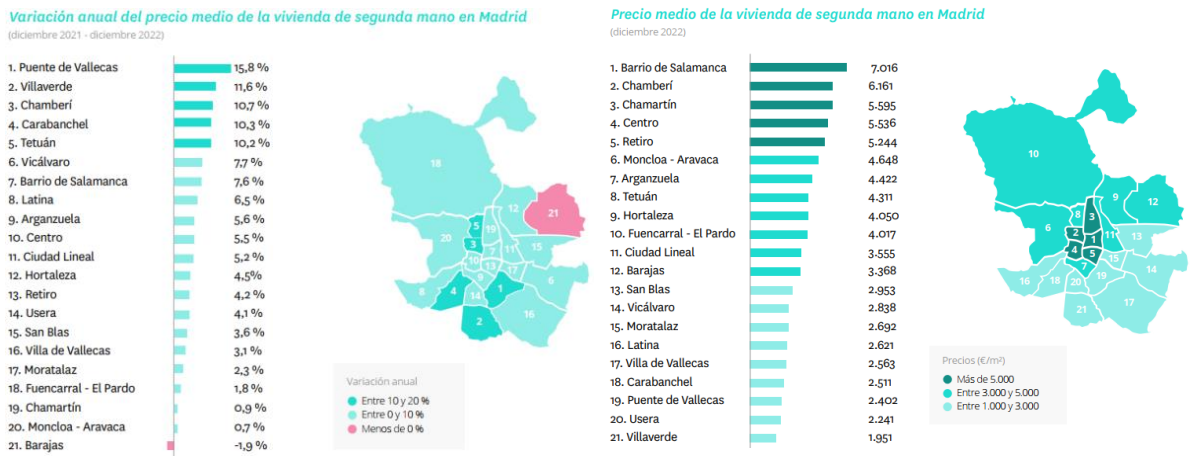


Figura 1: Evolución en 2022 por distritos de Madrid según Fotocasa.

1.2. Objetivos del Trabajo

A lo largo del camino hacia el objetivo principal del trabajo nos encontraremos con una serie de objetivos secundarios como pueden ser los siguientes:

- Ser capaces de realizar un procesamiento adecuado de los datos de partida. Haciendo una limpieza de éstos, así como enriqueciéndolos con otras fuentes que nos aporten variables que puedan ser relevantes en nuestro estudio.

- Generar visualizaciones que nos permitan entender en profundidad el dato del que disponemos, así como encontrar diferentes correlaciones entre las variables y establecer patrones que determinen el precio de la vivienda.

- Utilizar diferentes técnicas de Machine Learning basadas en minería de datos para comprender en profundidad los datos, así como obtener las primeras predicciones a partir de distintos algoritmos para poder así comparar los resultados obtenidos.

- Usar técnicas de Deep Learning para obtener modelos predictivos, compararlos con los obtenidos anteriormente y estudiar la precisión en nuestras predicciones.

Todos estos objetivos nos conducirán al propósito principal de ser capaces de predecir el precio de una vivienda en base a sus características más relevantes en cada una de las diferentes áreas de la Comunidad de Madrid, identificadas por el código de distrito, de manera que esto pueda resultar de utilidad a la hora de plantearse la compra de una vivienda, o de establecer un precio en la venta de la misma.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

Continuando con lo citado en el punto anterior, este trabajo se alinea con el ideal que se busca desde la competencia de compromiso ético y global (CCEG): "*Actuar de manera honesta, ética, sostenible, socialmente responsable y respetuosa con los derechos humanos y la diversidad, tanto en la práctica académica como en la profesional, y diseñar soluciones para mejorar estas prácticas*". Representando la sostenibilidad desde el ODS 12 - Responsible consumption and production; el Comportamiento ético y responsabilidad social (RS) desde el ODS 8 - Decent work and economic growth, al comprobar si los indicadores económicos del país están relacionados fuertemente con el precio de la vivienda o no; y finalmente la desigualdad y los derechos humanos desde el ODS 10 - Reduced inequalities, ya que recordemos que el objetivo primordial de este trabajo es proporcionar las herramientas necesarias para que aquellas personas que no tengan claro en qué momento podrán acceder a una vivienda, encuentren la vivienda que más se adecúe a su presupuesto, así como puedan tener una primera aproximación con cuánto podría costar una vivienda en base a las características buscadas. Esto se traduce en el ámbito práctico del mundo real, en que puede resultar en una ventaja decisiva para aquellas personas que estén en las siguientes situaciones:

- Compradores que buscan una vivienda y quieren conocer si el precio de venta de un inmueble es adecuado.

- Inversores que quieren comprar una vivienda. Una herramienta de este tipo puede permitir identificar inmuebles con un precio por debajo del mercado.

- Vendedores que quieren conocer cuál es el precio adecuado de venta al público de un inmueble de su propiedad (tasación).

- Analistas que buscan entender el momento actual del mercado inmobiliario.

Las herramientas comunes para este cometido suelen ser solicitar una tasación oficial, las cuales pueden ser muy costosas; contar con asesoramiento de una inmobiliaria, que a su vez suelen cobrar una comisión por sus servicios; o utilizar herramientas de tasación online, que no siempre son precisas y, de nuevo, acostumbran a tener un coste asociado.

1.4. Enfoque y método seguido

Tras analizar las diferentes opciones, el enfoque escogido en este trabajo será el de la metodología **CRISP-DM**. Este facilita la comprensión integral del problema desde la perspectiva del negocio y asegura que el análisis de datos esté alineado con los objetivos

del proyecto. Su naturaleza iterativa permite ajustar el análisis a medida que se adquiere mayor comprensión de los datos y los requisitos del problema. Además, al ser un estándar de la industria, proporciona un marco reconocido y probado que puede guiar eficientemente a estudiantes y profesionales a través del proceso de minería de datos. Esta metodología se basa en las siguientes fases:

- **Definición de las necesidades del cliente – Comprensión del negocio**: Consiste en identificar y entender los objetivos y requisitos del proyecto desde una perspectiva de negocio.
- **Estudio y comprensión de los datos**: Recopilar los datos necesarios y familiarizarse con ellos para comprender su estructura, calidad y potencial.
- **Análisis de los datos y selección de las características**: Explorar los datos para encontrar patrones y seleccionar las características más relevantes para el modelado.
- **Modelado**: Aplicar técnicas de minería de datos y modelado estadístico para construir modelos que puedan responder a las necesidades del negocio o, en este caso concreto, para cumplir con los objetivos fijados del trabajo.
- **Evaluación**: Evaluar los modelos en términos de eficacia y precisión, asegurándose de que satisfacen los objetivos a cumplir.
- **Despliegue**: Implementar el modelo en un entorno de producción y monitorear su rendimiento. Esta fase no se llevará a cabo dentro del marco del trabajo, pero dado que los resultados y modelos obtenidos serán públicos, sí podría darse en un futuro.

1.5. Planificación del trabajo

- En primer lugar, se realizarán las tareas de introducción al trabajo. Estas tareas consistirán en el desarrollo de la introducción y definición de este mismo, así como simultáneamente nos familiarizaremos con el set de datos con el que vamos a trabajar. Una vez nos hayamos familiarizado con nuestros datos y planteado un enfoque de trabajo, deberemos identificar cuáles pueden ser las variables con las que podría ser interesante complementar el dataset y buscar los conjuntos de datos adecuados para poder agregarlas. Estos datos se obtendrán mayoritariamente del INE como principal fuente de información.

- Una vez hecho esto procederemos a identificar el estado del arte del trabajo que intentaremos mejorar más adelante en la medida de lo posible, así como aplicarlo a nuestro conjunto de datos. Mientras en paralelo cruzaremos nuestro dataset con los nuevos datos obtenidos para agregar nuevas variables que puedan aportar valor.

- Una vez iniciado el desarrollo principal del trabajo, el flujo de acción será el siguiente:

- En primer lugar, se realizará una limpieza profunda de los datos. Descartaremos valores nulos y aseguraremos que el formato y la calidad del dato sean las requeridas para los procesos posteriores.

- En segundo lugar, realizaremos un análisis completo de los datos con los que trabajaremos finalmente; incluyendo estadísticas descriptivas, correlaciones entre variables, histogramas de valores... Y en general cualquier forma de análisis o visualización que pueda aportarnos valor.

- Acto seguido, procederemos a obtener nuestros primeros modelos predictivos basándonos en técnicas de minería de datos. Para esto probaremos diferentes enfoques y algoritmos basados en árboles de decisión como pueden ser Random Forest o Gradient Boosting entre otros posibles, con la consiguiente búsqueda de sus hiper parámetros óptimos.

- A continuación, haremos lo mismo con los modelos basados en Deep Learning, probando con diferentes tipos de redes y realizando el fine tunning* adecuado hasta obtener los mejores resultados posibles.

- Finalmente, compararemos todos los resultados obtenidos para tratar de sacar el máximo valor posible a toda la información y recursos disponibles.

- Por último, se realizará la memoria final del trabajo donde se recogerá todo lo obtenido en el mismo, se llevarán a cabo las correcciones pertinentes y se elaborará una presentación a modo de resumen visual del trabajo realizado y los resultados obtenidos.

Se adjunta un diagrama de Grantt como resumen de la planificación previamente citada:

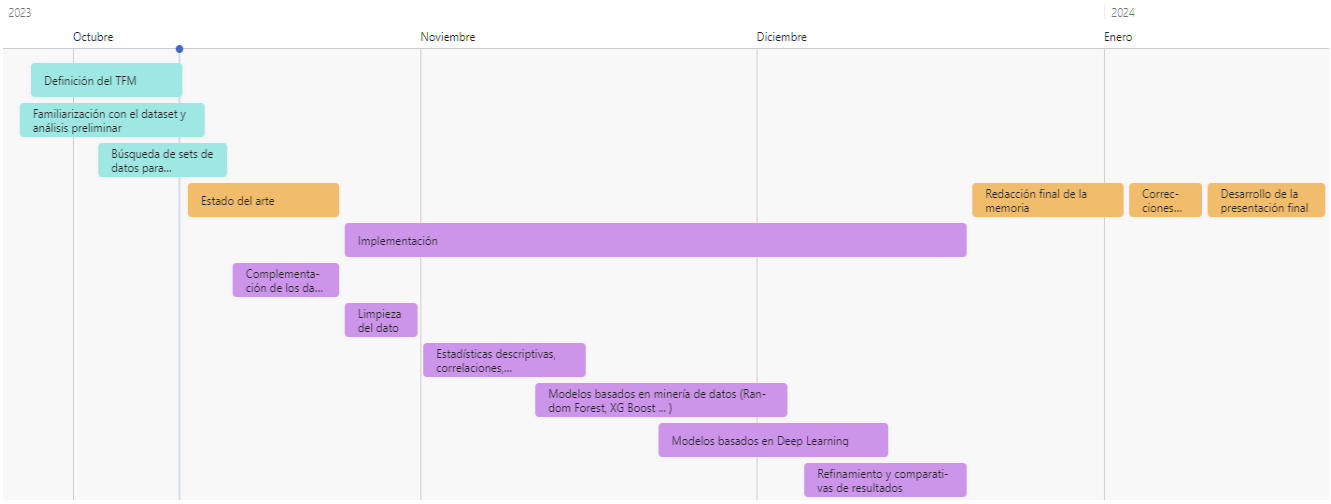


Figura 2: Planificación del trabajo completo.

1.6. Breve sumario de productos obtenidos

A lo largo del desarrollo del trabajo se han obtenido múltiples modelos en el ámbito del Machine Learning que nos permiten cumplir de forma solvente con el propósito marcado, siendo este el de predecir el valor de una vivienda en base a sus características.

Como se relataba anteriormente, estos estarán principalmente divididos en dos enfoques, siendo el primero los modelos basados en árboles decisión como serán Random Forest, Gradient Boosting, XG Boost y LGBM; y los modelos basados en Deep Learning como serán las Redes Densas (DNN), Convolucionales (CNN), Regresivas (LSTM) y Transformers.

Más adelante veremos cada una en detalle, así como una comparativa entre todas las soluciones obtenidas y determinaremos cuál es la que mejor se ajusta a nuestro problema en el contexto de nuestros datos.

1.7. Breve descripción de otros capítulos de la memoria

Para cada uno de los métodos de analítica avanzada empleados en el desarrollo del trabajo, se destinará un espacio a lo largo de esta memoria para detallar la teoría en la que se basan y explicar de forma concisa cómo se comportan y cuál es el uso práctico que les estamos dando.

En el apartado del *Glosario* podremos encontrar una breve descripción de cada uno de los conceptos que encontremos a lo largo de la memoria seguidos del símbolo * .

Dentro del apartado *Bibliografía* podremos encontrar un subapartado denominado como *Recursos*. Dentro de este encontraremos los enlaces de los que hemos obtenido datos con los que enriquecer nuestro dataset, así como un enlace de github en el que se encontrará todo el código del trabajo y los recursos necesarios para replicar los resultados.

2. Estado del arte

Se ha basado la investigación del estado del arte en dos líneas principales para determinar el estado de lo desarrollado hasta ahora. Por un lado, se ha hecho un análisis, utilizando como fuente principal Google Scholar, de cuáles son los principales papers que tratan la predicción del mercado inmobiliario en general, para determinar hasta dónde llega la línea de investigación en este ámbito hasta la fecha. Por otro lado, se han buscado diferentes TFMs desarrollados previamente que aborden esta cuestión, y se han seleccionado los más innovadores o completos, para tomar posibles referencias o ideas que puedan ayudar en el desarrollo de este trabajo.

A lo largo de los trabajos de investigación que componen el estado del arte se repite una premisa común, y es que está comprobado que el precio de la vivienda depende de muchos factores con los que no guarda una relación lineal, y que podrían ser difícilmente identificables mediante procesos de analítica convencionales. Esto se ve reflejado especialmente en el paper desarrollado en 2020 por diversos investigadores chinos [2] donde se analiza y predice el mercado de Taiwan, utilizando redes neuronales BPNN y CNN (que en un principio se usan más en procesamiento imágenes) para la predicción de series temporales, obteniendo buenos resultados con las CNN tras ajustar debidamente las variables de estudio. Otro muy buen estudio, y muy reciente, es el publicado en el Civil Engineering Journal, en marzo de 2023 [3], donde se predice el precio en 28 ciudades de Egipto basándose en un modelo de regresión con aprendizaje supervisado, y tiene la ventaja de que se explica de forma muy detalla los métodos de construcción y ajuste del modelo hasta obtener los mejores resultados posibles. Finalmente quiero destacar otra publicación académica que, en mi opinión, es la más completa en este ámbito a pesar de ser la más antigua; y es la publicada por MDPI y llevada a cabo principalmente por investigadores de la Universidad Carlos III de Madrid en 2018 [4]. En esta no se limitan simplemente a generar un modelo predictivo, sino que se lleva a cabo un análisis completo de las variables referentes a la vivienda en el Barrio de Salamanca (Madrid) buscando correlaciones y posibles relaciones ocultas, para explotar todas las opciones posibles en torno a la predicción del precio de la vivienda, no solo en el ámbito del Deep Learning, también con una fuerte carga de minería de datos. Obteniendo con la combinación de estos tres trabajos y los papers previos en los que se basan, la referencia del estado del arte en investigación.

Como se mencionaba anteriormente, no solo se ha tenido en cuenta el ámbito de la investigación, si no que también utilizaremos TFMs anteriores que puedan proporcionar diferentes enfoques o referencias. En este ámbito, de nuevo destacan tres entre los seleccionados. El primero de ellos será el desarrollado en la facultad de negocios y tecnología digital, de la Universidad de Southampton [5] para predecir el precio de la vivienda en California. En este se le da mucho peso a la parte de posibles aplicaciones del trabajo y a lo que implica el análisis del mercado inmobiliario como ente económico, de lo cual se pueden tomar algunas referencias, aunque en el tema del procesamiento no se llega a

aplicar Deep Learning. Nuestra segunda referencia será el desarrollado en la facultad de estadística de la Universidad Complutense de Madrid en 2018 [6], donde se predice el mercado en la Comunidad de Madrid, teniendo en cuenta una gran variedad de variables y aplicando métodos muy variados en todos los ámbitos, llegándose a usar redes neuronales autorregresivas (NNAR) y redes ELM con regresión dinámica (de las que obtiene los mejores resultados). Aparte de este último que probablemente se tome como referencia principal a mejorar, también considero muy interesante el desarrollado en la facultad de analítica de negocio de la Universidad de Tilburg en 2021 [7], donde se evalúa el precio del metro cuadrado en viviendas de nueva construcción y se propone el uso de técnicas de Transfer Learning, basado en modelos altamente probados en la predicción de precios en series temporales.

3. Materiales y métodos

El desarrollo de esta parte de la memoria estará principalmente dividido en tres partes:

En la primera de ellas nos centraremos en describir los métodos utilizados para el análisis preliminar y preparado del dato.

En la segunda construiremos métodos predictivos basados en árboles de decisión.

Finalmente en la tercera construiremos métodos basados en redes neuronales de aprendizaje profundo o DL.

3.1 Análisis preliminar

3.1.1 Descripción y enriquecimiento

En primer lugar vamos a describir de forma detallada los datos con los que contamos:

Nuestro dataset inicial cuenta con datos de características de inmuebles anunciados en un portal inmobiliario con presencia online, referentes a viviendas de todo el territorio nacional de España, anunciadas entre el **7 de Septiembre de 2021** y el **28 de Enero de 2022.** Este cuenta con 954157 filas**,** sin embargo, para acotar el análisis vamos a seleccionar solo aquellas que hagan referencia a inmuebles en la Comunidad de Madrid, quedándonos así con **66390 filas** y **27 columnas** o variables a estudiar entre las que se encuentran las siguientes:

- **Unnamed**: 0 : Índice de la fila en formato entero.
- **Fecha** : Fecha de publicación del anuncio en la página en formato texto.
- **ID** : Identificador único de la oferta en formato texto.
- **URL** : URL de la oferta en cuestión en la página de origen en formato texto.
- **ID_Cliente** : Identificador único del anunciante en formato entero.
- **URL_Cliente** : URL al perfil del anunciante en formato texto.
- **Inmueble** : Tipo de inmueble entre los que, en nuestro conjunto de datos, solo tendremos viviendas y vendrá indicado en formato texto.
- **Caracteristicas** : Tipo de vivienda (apartamento, chalet, duplex, etc) en formato texto.
- **Habitaciones** : Número de habitaciones en formato entero.
- **Aseos** : Número de aseos en la vivienda en formato entero.
- **Terraza** : Variable booleana que indica si tiene terraza o no.
- **Piscina** : Variable booleana que indica si tiene piscina o no.
- **Garaje** : Variable booleana que indica si tiene garaje o no.
- **Precio** : Valor en euros por el que se vende la vivienda en formato numérico.

- **Metros** : Metros cuadrados de los que dispone la vivienda en formato numérico.
- **Relacion** : Cociente entre las variables Precio y Metros en formato texto.
- **CodigoPostal** : Código postal de la vivienda en formato texto.
- **Latitud** : Valor de la latitud geográfica expresado como texto.
- **Longitud** : Valor de la longitud geográfica expresado como texto.
- **Precision** : Variable booleana que indica si la ubicación es precisa o no.
- **CMUN** : Código del municipio en formato texto.
- **CPRO** : Código de provincia en formato texto.
- **CCA** : Código de la comunidad autónoma en formato texto.
- **CUDIS** : Código del distrito en formato texto.
- **NPRO** : Nombre de la provincia en formato texto.
- **NCA** : Nombre de la comunidad autónoma en formato texto.
- **NMUN** : Nombre del municipio en formato texto.

Antes de comenzar con el análisis en profundidad, se ha planteado la necesidad de complementar estos datos con nuevas variables que nos den un contexto del momento económico en el que se encuentran nuestros datos, de igual forma que se hace en [6].

Para llevar a cabo este enriquecimiento se ha realizado un proceso de búsqueda en el que se han seleccionado las variables que podrían estar más relacionadas con la variación del mercado inmobiliario y se han obtenido los datos principalmente del Instituto Nacional de Estadística (INE). Las variables añadidas han sido las siguientes:

- **euribor*** [8]: Valor del euribor medio mensual en formato numérico.
- **numero_hipotecas** [9]: Número de hipotecas concedidas en ese mes en formato texto.
- **ipc*** [10]: Valor del Índice de Precios al Consumidor mensual en formato numérico.
- **tipo_interes*** [9]: Valor del tipo de interés general medio mensual en formato texto.
- **paro** [11]: Valor del porcentaje total de paro trimestral en formato texto.

Una vez enriquecido el dataset con la información pertinente, es el momento de proceder con la limpieza del dato.

3.1.2 Limpieza del dato

Correcciones de formato

En primer lugar se ha corregido el formato de las variables Relacion, CodigoPostal, paro, tipo_interes, ipc y numero_hipoteca; que originalmente contaban con un formato de texto por lo que las se convirtieron a un formato numérico. De igual forma, se han convertido las variables Caracteristicas, Habitaciones, Aseos, Terraza, Piscina, Garaje y CUDIS a variables enteras discretas, para poder ser utilizadas posteriormente en los modelos.
Se ha cambiado la variable Fecha por tres variables Año, Mes y Dia que no tengan formato de fecha, si no entero, para que puedan ser utilizadas en los posteriores procesos de entrenamiento de modelos sin incurrir en errores.

Se ha corregido también los valores de la variable Relacion para que se ajusten a lo que representa la variable, ya que no siempre estaban acertados. Y de igual forma corregimos los valores de las variables Latitud y Longitud para que encajen en los valores máximos y mínimos que llegan a alcanzar en la Comunidad de Madrid.

Eliminación de columnas que toman un único valor

Entre las columnas de nuestro dataset, tenemos que las variables Inmueble, CCA, CPRO, NPRO y NCA tomarán un único valor, por lo que podremos prescindir de ellas.

Eliminación de columnas que no aportan información

Entre las variables restantes, tenemos que las variables Unnamed:0, ID, ID_Cliente, URL y URL_Cliente no nos aportarán información al no ser características de la vivienda como tal, por lo que podemos eliminarlas.

De forma adicional, las variables CodigoPostal, CMUN y NMUN serán redundantes con la variable CUDIS, ya que en el propio código del distrito se incluyen el código postal y el de municipio. También la variable Fecha, ya que esta no tendría un formato adecuado y es redundante con Año, Mes y Dia. Lo mismo pasa con la variable Relacion, que está directamente relacionada con la variable objetivo Precio, por lo que mantenerla podría ocasionar problemas de multicolinealidad.

Seleccionamos finalmente aquellas variables que nos ofrecen información útil y no redundante para tenerlas en cuenta a la hora de realizar nuestro estudio y desarrollar nuestros modelos, independientemente de lo útiles que parezcan a simple vista, ya que podrían ocultar relaciones ocultas que los modelos de aprendizaje profundo sean capaces de detectar. Estas variables serán: **Año, Mes, Dia, Caracteristicas, Habitaciones, Aseos, Terraza, Piscina, Garaje, CUDIS, Latitud, Longitud, Metros, euribor, numero_hipotecas, ipc, tipo_interes, paro, Precio**.

Tratamiento de valores nulos y outliers

Una vez corregidos todos los problemas de formato y seleccionadas las variables, es el momento de analizar la existencia de valores nulos y atípicos o outliers.

- Con respecto a los valores nulos, su número total es inferior a los 3000 registros, siendo estos todos los que se ven en la captura a continuación, en un dataset de 66390 filas. Es por esto que, al no suponer ni el 5% del total de los datos y contar con dato suficiente, la política ante estos ha sido de eliminar directamente todas las filas que tuviesen datos nulos en alguna de las columnas utilizando el comando *df.dropna().*

Caracteristicas	6
Habitaciones	2069
Aseos	1183
Terraza	0
Piscina	0
Garaje	0
Precio	395
Metros	257
Relacion	598

- Con respecto a los valores atípicos, se ha definido un limite superior e inferior para las variables numéricas continuas, y en función de este se ha diseñado una función para eliminar dichos valores. Esta función se ha definido como:

```
def eliminar_atipicos(df, columna, a=0.01):
    Q1 = df[columna].quantile(a)
    Q3 = df[columna].quantile(1-a)
    IQR = Q3 - Q1
    limite_inferior = Q1
    limite_superior = Q3 + 0.5 * IQR # Multiplicador estándar = 1.5

    return (df[columna] >= limite_inferior) & (df[columna] <= limite_superior)


def eliminar_nulos_atipicos(df):
    # Eliminar filas con valores nulos
    df = df.dropna()

    # Eliminar valores atípicos
    df = df[eliminar_atipicos(df, 'Habitaciones') & eliminar_atipicos(df, 'Aseos')
            & eliminar_atipicos(df, 'Metros') & eliminar_atipicos(df, 'Precio')]

    return df
```

Aquí Q1 y Q3 son el valor del primer y el tercer cuantil para cada una de las variables en nuestros datos, mientras que el rango intercuartílico (IQR) se define como la diferencia entre ambos valores.

El multiplicador de 1.5 es la convención que asume una distribución más o menos simétrica. En distribuciones sesgadas, como veremos que son la mayoría de estas variables, este enfoque puede no ser el más adecuado, ya que el IQR será demasiado grande en comparación con el rango de valores que se encuentra fuera de este.
Es por esto que el enfoque estándar del multiplicador de IQR no sería el óptimo para distribuciones de datos no normales similares a este caso, por ello se ha escogido un multiplicador más adecuado como es 0.5 para el límite superior y 0 para el límite inferior. La elección de este límite inferior se basa en que IQR será muy grande para los valores pequeños y nos conviene descartar algunos de estos debido a la aparente existencia de valores que podrían corresponder a alquileres, como se verá más adelante.

Se han construido una serie de visualizaciones para controlar los valores atípicos descartados:

```
Valores Atípicos en 'Habitaciones':
 1728     153.0
4228       20.0
15051      32.0
18480      60.0
24281      18.0
24735      30.0
25097      30.0
29834     117.0
34471      32.0
39378      19.0
40472      52.0
40648     154.0
41569     113.0
42162      25.0
42323      25.0
45047      19.0
60210      21.0
```

```
Valores Atípicos en 'Aseos':
 3145     1992.0
4228       20.0
8402     1992.0
11413      14.0
18154      20.0
24735      20.0
25097      20.0
40047    2007.0
40314      14.0
40648     138.0
42162      17.0
42323      17.0
43798      16.0
44002      14.0
52319    1977.0
54501      15.0
55639      14.0
```

```
Valores Atípicos en 'Metros
 969        2700.0
3149        3630.0
3981       33836.0
3982        4748.0
4228     1990000.0
              ...
55519       7506.0
55520       5433.0
59169       4500.0
64738      34000.0
65731       3500.0
Name: Metros, Length: 161,
```

```
Valores Atípicos en 'Precios
 2295     195000000.0
2613      15500000.0
2838      15500000.0
4228       7000000.0
4346      11950000.0
8777       8000000.0
8885       8000000.0
27416      6800000.0
27476      7000000.0
28030      8700000.0
29285      6800000.0
31426     11111111.0
33092     13000000.0
33152     10900000.0
```

Visualizamos ahora los valores outliers mediante la representación boxplot, donde se aprecia que la mayoría de valores que podemos considerar atípicos son cotas superiores, así como la asimetría en la distribución del dato.

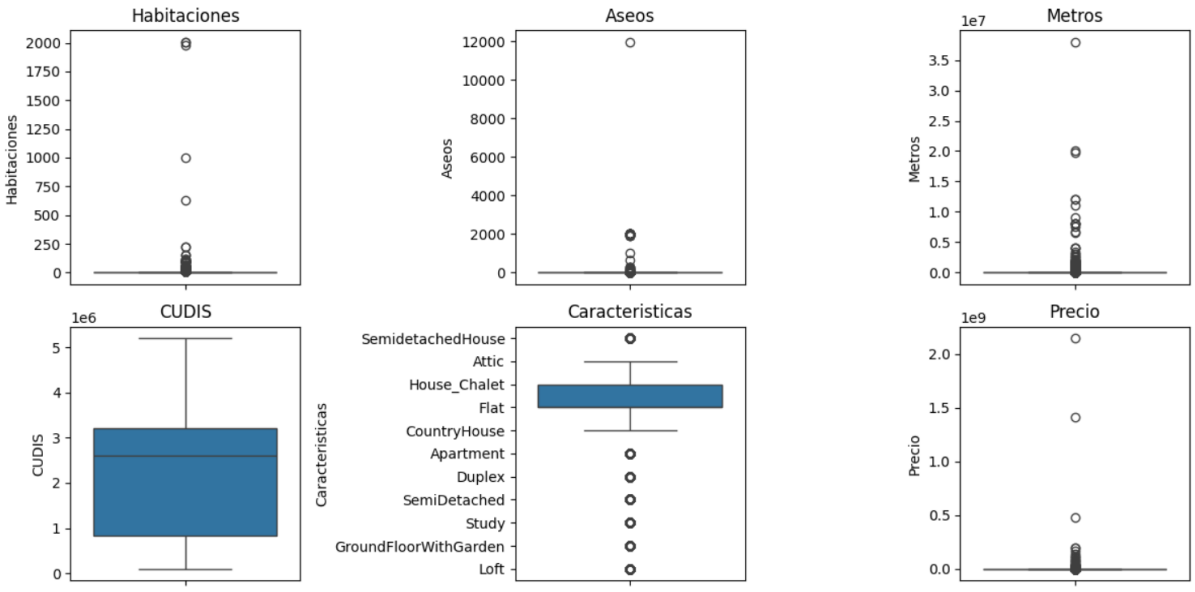


Figura 3: Representación de valores atípicos con cajas y bigotes.

Antes de continuar con la visualización de datos, cabe destacar que se ha visto que en la distribución de precios, así como sobre todo la de la variable Relacion, que hacía referencia al cociente entre el Precio y los Metros, existen valores muy inusuales incluso fuera de los propios outliers. Es por esto que se ha realizado un análisis más a fondo de estas variables, donde podemos ver que existen picos inusuales en los valores mas pequeños:

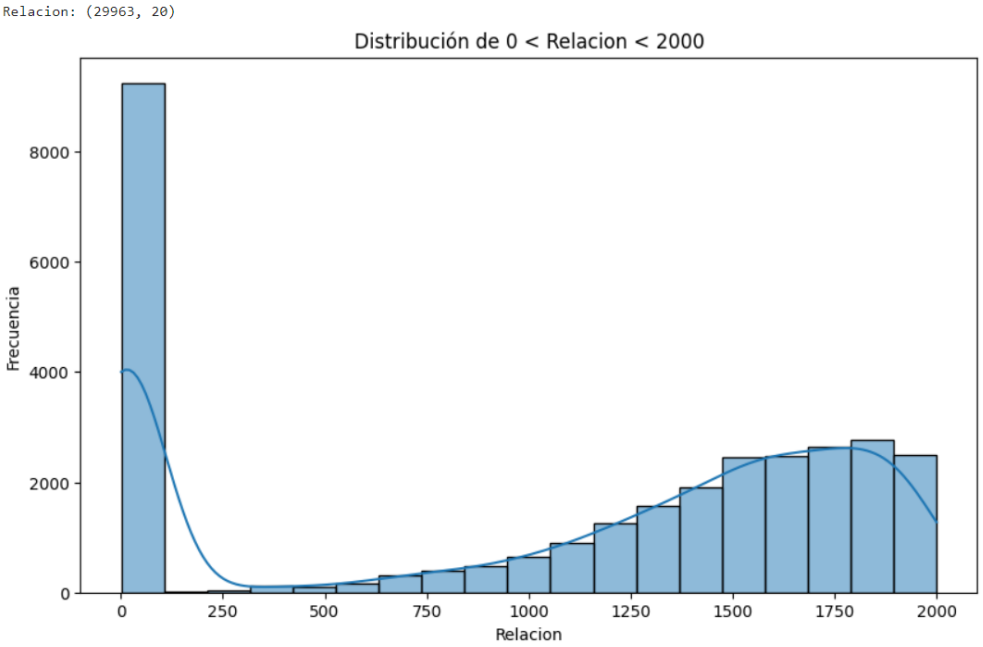


Figura 4: Distribución de la Relacion en valores inferiores a 2000.

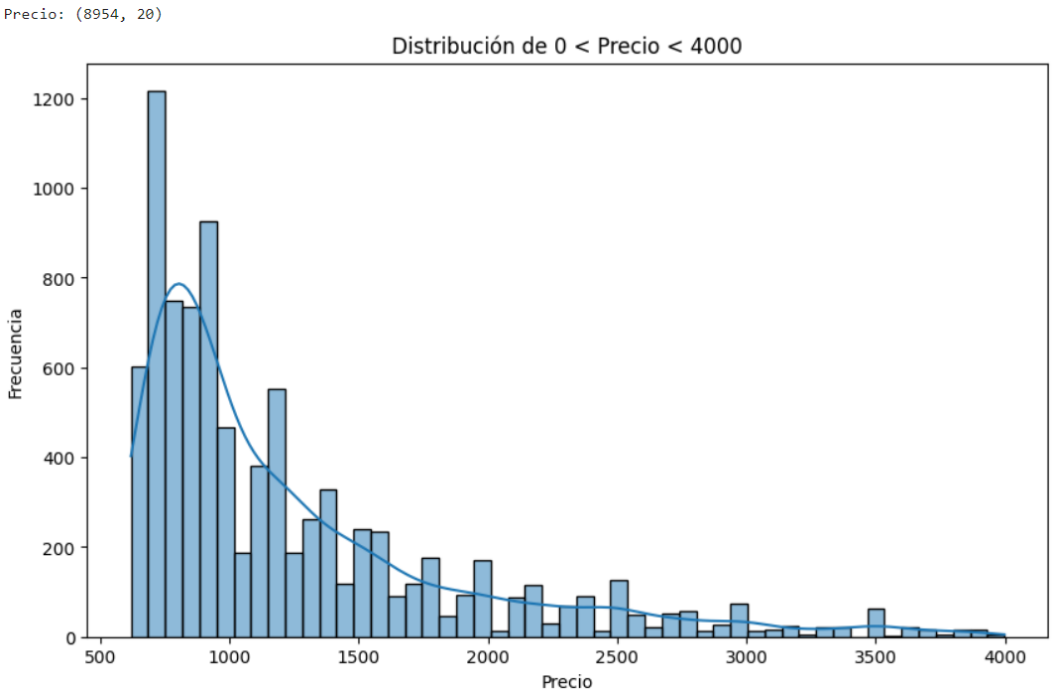


Figura 5: Distribución del Precio en valores inferiores a 4000.

Estos valores suponen un porcentaje importante del dato en nuestro conjunto y todo apunta a que son datos de viviendas en alquiler. Como nuestro análisis se centra exclusivamente en la predicción de precios de compra y venta, se ha establecido un criterio para poder descartar estos registros de alquiler.

La idea es establecer un umbral que, aún no pudiendo tener una eficacia del 100% debido a que en la Comunidad de Madrid puede haber alquileres superiores a lo que pueda llegar a costar una vivienda, encaje con los datos visualizados anteriormente de forma que nuestro dato quede lo más coherente posible, eliminando el menor número de ofertas de venta posible, y a su vez manteniendo el menor número de ofertas de alquiler posible.

Para determinar si eliminamos o no aquellos datos con una relación de valor del metro cuadrado muy baja, se ha seguido un criterio experto, tomando como referencia el precio mínimo del metro cuadrado en la Comunidad de Madrid según el portal de vivienda **Idealista** [13]. En este se sitúa dicho mínimo en el municipio de Fuentidueña de Tajo, con una relación de 854€/m2. Sobre este daremos un margen de error del 10% por la posible variación del precio en los años que distan nuestros datos de la referencia, por lo que estableceremos nuestro umbral en **770€/m2** y descartaremos todos los registros por debajo de este valor.

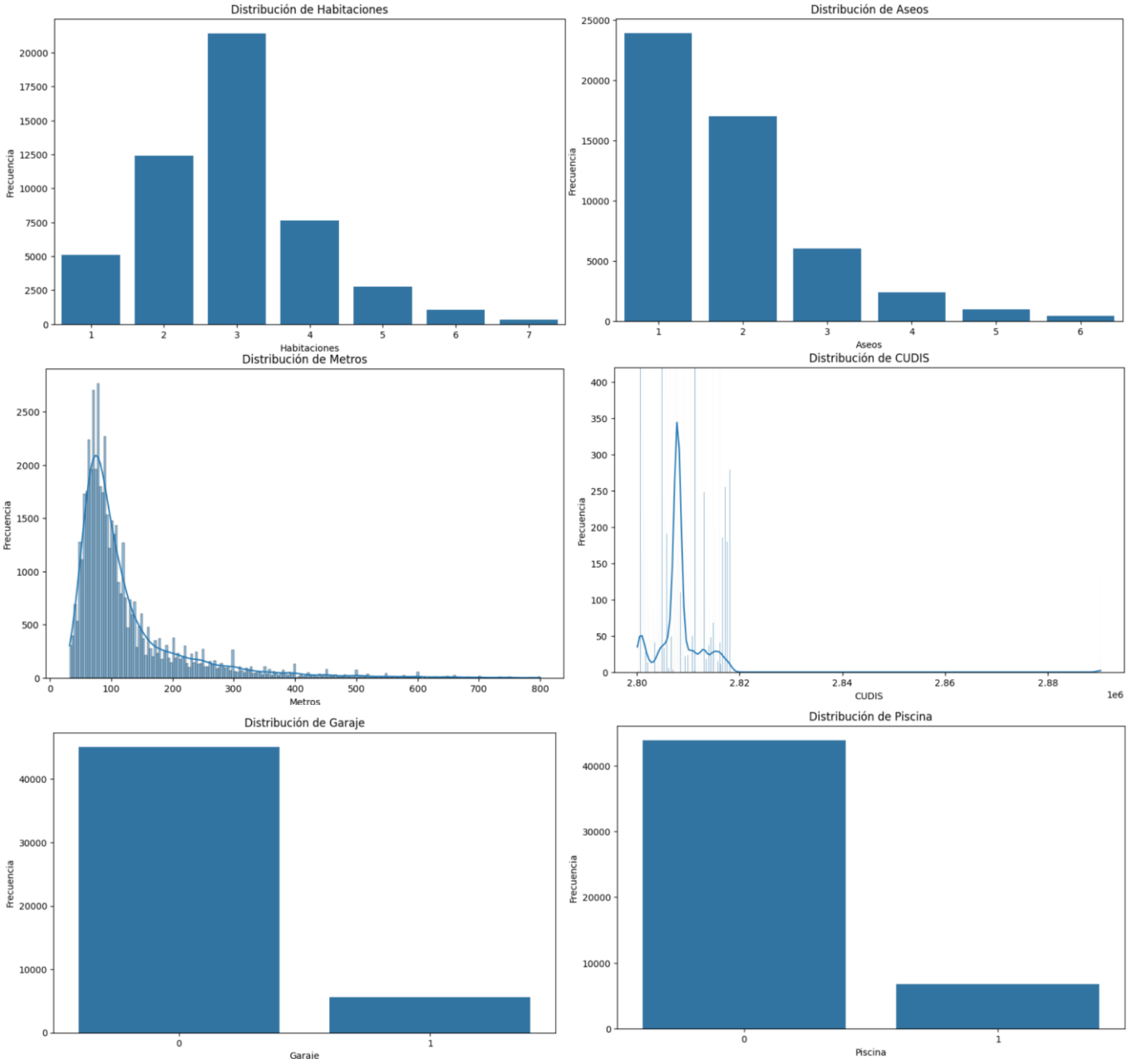
3.1.3 Visualización de datos

En primer lugar, se ha construido una tabla donde poder apreciar las estadísticas principales de cada una de las variables, posteriormente, se visualizará la distribución de los datos en las variables más representativas y se han analizado los resultados:

	Variable	Maximo	Minimo	Media	Moda	Varianza	Desviacion Tipica	Quantil 1	Quantil 2	Quantil 3	Kurtosis
0	Año	2022	2021	2021.11	2021	1.000000e-01	0.31	2021.00	2021.00	2021.00	4.10
1	Mes	12	1	8.61	9	8.170000e+00	2.86	9.00	9.00	9.00	2.82
2	Dia	31	1	11.99	8	5.468000e+01	7.39	8.00	8.00	15.00	0.15
3	Caracteristicas	10	1	2.08	1	5.260000e+00	2.29	1.00	1.00	2.00	2.84
4	Habitaciones	10	1	2.97	3	1.560000e+00	1.25	2.00	3.00	3.00	2.78
5	Aseos	8	1	1.89	1	1.300000e+00	1.14	1.00	2.00	2.00	3.83
6	Terraza	1	0	0.43	0	2.500000e-01	0.50	0.00	0.00	1.00	-1.93
7	Piscina	1	0	0.14	0	1.200000e-01	0.35	0.00	0.00	0.00	2.33
8	Garaje	1	0	0.11	0	1.000000e-01	0.31	0.00	0.00	0.00	4.27
9	CUDIS	2890301	2800101	2808374.79	2807901	3.504784e+07	5920.12	2807901.00	2807910.00	2808301.00	111.94
10	Latitud	41.13	39.99	40.41	40.44	1.000000e-02	0.12	40.36	40.42	40.46	2.47
11	Longitud	-3.08	-4.51	-3.71	-3.61	3.000000e-02	0.16	-3.77	-3.70	-3.64	2.09
12	Metros	1184.0	32.0	132.43	80.0	1.323661e+04	115.05	70.00	93.00	143.00	15.00
13	euribor	-0.48	-0.5	-0.49	-0.49	0.000000e+00	0.01	-0.49	-0.49	-0.49	0.31
14	numero_hipotecas	10.48	7.46	9.55	10.48	1.590000e+00	1.26	8.06	10.48	10.48	-1.51
15	ipc	6.2	3.6	4.29	3.6	9.300000e-01	0.96	3.60	3.60	5.30	-1.10
16	tipo_interes	1.88	1.84	1.85	1.84	0.000000e+00	0.01	1.84	1.84	1.85	-0.34
17	paro	10.68	9.01	9.82	9.98	2.600000e-01	0.51	9.98	9.98	9.98	-0.49
18	Precio	3970000.0	27000.0	383200.85	185000.0	1.947105e+11	441260.16	148000.00	229900.00	425000.00	16.08

Figura 6: Tabla de estadísticas por variables.

Histogramas:



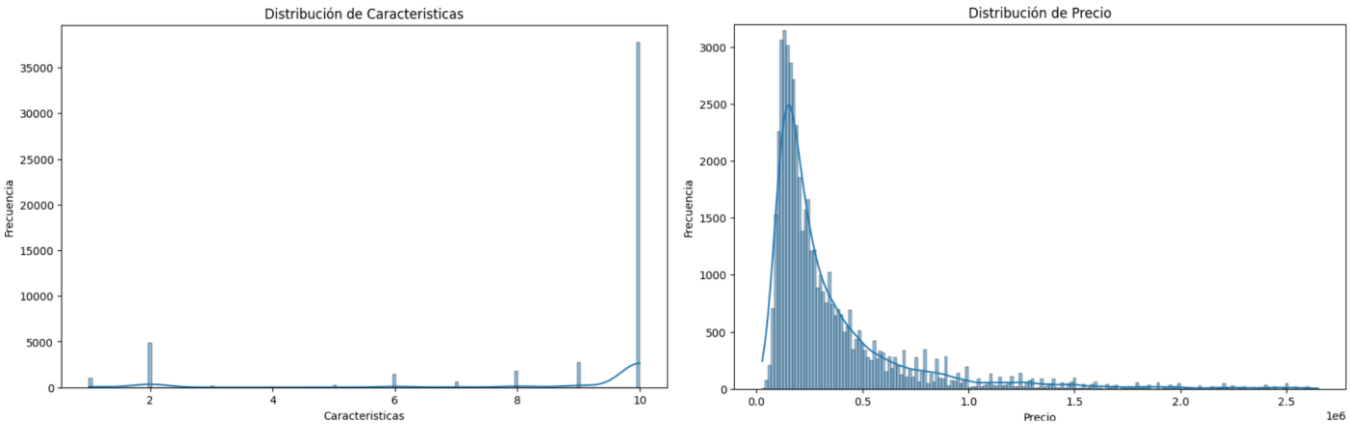


Figura 7: Histograma de las principales variables.

Análisis:

- **Habitaciones**: Podemos observar que el valor más repetido son 3 habitaciones, seguido de 2 y 4. Finalmente también podemos encontrar bastantes viviendas con 1 y 5 habitaciones, mientras que las de 6, y especialmente las de 7, son prácticamente residuales. La media de habitaciones estará en 2.97 mientras que la moda se situará en 3.

- **Aseos**: Siendo lo más común encontrar viviendas con un aseo, seguido relativamente de cerca por las de dos, a medida que aumentamos el número de aseos cada vez resulta en situaciones más difíciles de encontrar. La media de aseos es 1.89 mientras que la moda es 1, vislumbrando esto que es mucho menos común comprar una casa que cuente con varios aseos que comprar una casa con múltiples habitaciones.

- **Metros**: Podemos ver como la mayoría de valores se sitúa entre 70 y 143 aproximadamente, siendo la media 132.43 y la moda 80. A partir de 140 la distribución de los metros va disminuyendo hasta alcanzar los valores más altos que apenas son residuales.

- Distritos (**CUDIS**): Podemos ver que prácticamente todos se distribuyen entre los que empiezan por 280 y los que empiezan por 281, esto tiene una explicación sencilla y encaja con los valores esperados puesto que estos códigos de distrito son los que pertenecen a la capital y los grandes pueblos cercanos de alrededor.

- **Garaje**: La proporción de las casas que cuenta con garaje apenas llega al 5%, haciendo de esta condición algo realmente extraño.

- **Piscina**: Tener piscina también será una condición similarmente extraña para una vivienda en venta, aunque puede sorprender que es más común que el hecho de tener garaje. La media de la variable piscina se encuentra en 0.14 frente al 0.11 de la variable garaje, por lo que definitivamente se venden más viviendas con piscina que con garaje.

- **Características**: En esta gráfica podemos ver como el valor más repetido con muchísima diferencia es el que referencia a los pisos, lo cual tiene mucho sentido en un lugar como la Comunidad de Madrid que concentra la mayoría de su población en la capital, donde lo más común es vivir en un piso. A este valor le siguen, aunque muy alejados, el valor que referencia a los chalets y el que referencia a los duplex.

- **Precio**: Como podemos observar, nuestra variable objetivo concentra la mayoría de sus datos en los primeros valores del gráfico. Teniendo una media de 383200.85 y siendo los 185000 euros la moda, podemos decir que la mayoría de las viviendas se concentra en valores entre 100000 y 400000, siendo los valores superiores a estos cada vez más escasos hasta acabar siendo prácticamente residuales. Es reseñable destacar que la distribución de la gráfica de la variable objetivo guarda muchas similitudes con las gráficas obtenidas para Aseos, Habitaciones y Metros, por lo que es de esperar que este muy relacionada con estas.

De forma adicional, vamos a visualizar la distribución geográfica de los datos en función de su Latitud y Longitud sobre el mapa, para poder así analizar dónde se encuentra la mayoría de la oferta y si coincide con lo que se podría esperar:

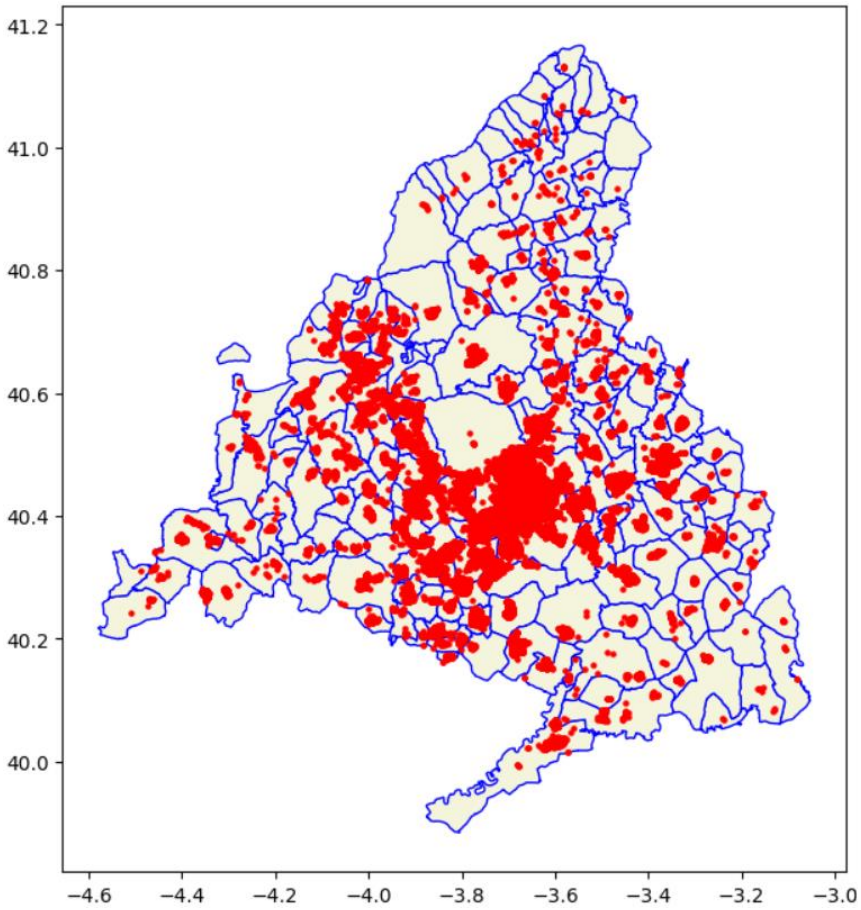


Figura 8: Distribución geográfica de la vivienda.

Se vislumbra claramente que la distribución de las viviendas en venta responde a lo esperado. La mayoría de estas se concentra en la capital y sus alrededores, coincidiendo estos puntos con aquellos que más población concentran y, por lo tanto, que tendrán mayor densidad de viviendas y con esto más probabilidades de tener también una mayor oferta en cuanto a la venta de estas se refiere.

Aparte de la gran concentración de la capital podemos ver como los municipios más grandes de la comunidad acaparan una mayor densidad de las viviendas en venta, mientras que los pequeños pueblos, tanto en la sierra del norte como los que colindan con Castilla la Mancha al sur, tendrán una densidad de ofertas notablemente menor.

Finalmente, estudiaremos la correlación entre las diferentes variables para determinar aquellas que, a priori, tendrán más peso en nuestro estudio mediante la representación de una matriz de correlaciones:

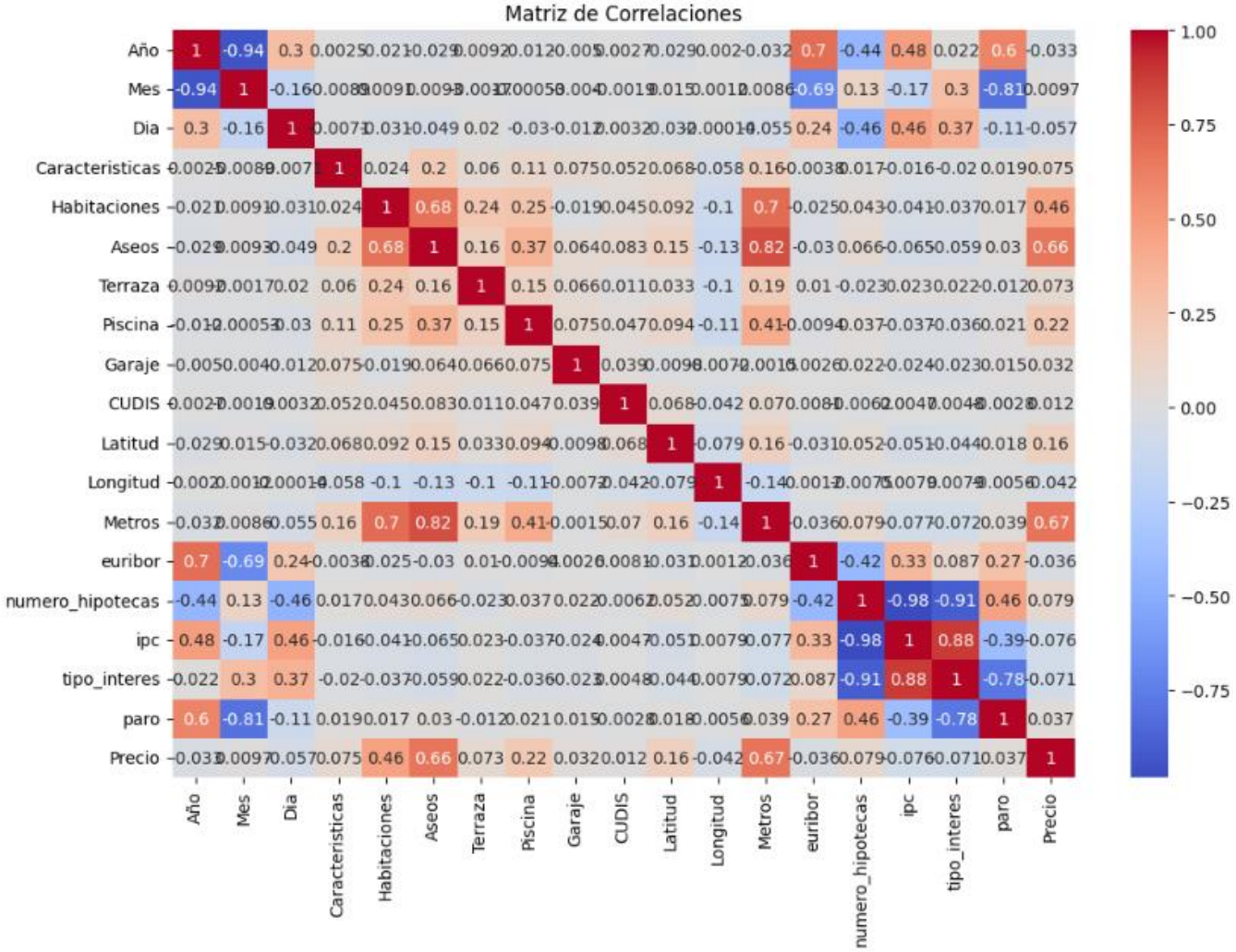


Figura 9: Matriz de correlaciones.

Como podemos ver, la variable a priori más relacionada con el precio será la cantidad de metros cuadrados, lo cual entraría dentro de lo esperado.
La segunda variable más relacionada con el precio, prácticamente al mismo nivel que la primera, será el número de aseos. Esto puede tener sentido desde el enfoque de que para construir una casa con muchos aseos, suele ser indispensable que cuente ya con un número considerable de habitaciones y metros.
La tercera variable más relacionada con el precio será el número de habitaciones y, por último, la cuarta más relacionada aunque en muchísima menor medida que las anteriores será el hecho de tener piscina o no.
Un hecho remarcable es que la quinta sería la Latitud, y es que es popularmente conocido que en Madrid los distritos más caros tienden a estar al norte.

Finalmente, cabe destacar que las variables añadidas que hacen referencia a la situación económica estarán muy relacionadas entre sí, tanto de forma directa como inversamente. Sin embargo, en esta primera aproximación que podemos ver en la matriz de correlaciones, no parece que estén aparentemente relacionadas con la variable objetivo.

No obstante, vamos a calcular el Factor de Inflación de la Varianza (VIF), el cual es una medida que se utiliza para detectar la multicolinealidad en un conjunto de variables predictoras en modelos de regresión. La idea detrás del VIF es cuantificar cuánto se infla la varianza de un coeficiente de regresión debido a la multicolinealidad. La multicolinealidad ocurre en un análisis de regresión cuando dos o más variables independientes están altamente correlacionadas entre sí. Esto significa que una variable independiente puede predecirse desde las otras con un alto grado de precisión. Una alta multicolinealidad puede producir aumentos en los tiempos de entrenamiento, así como resultados erróneos, sobre todo en los modelos basados en árboles.

Al calcular el VIF se aprecia que existe una multicolinealidad perfecta entre las variables económicas, por lo que vamos a combinar estas en dos variables PCA. Para esto, en primer lugar, escalamos los datos y calculamos el porcentaje de varianza que se explica con nuestras PCA, siendo este un 92,9%, lo cual es un muy buen resultado. Dentro de este un 65,85% vendrá explicado por la primera componente y un 27,05% por la segunda, por lo que decidimos añadir las dos. Una vez añadidas, el antes y después del VIF queda como se muestra a continuación:

	feature	VIF
0	Año	inf
1	Mes	inf
2	Dia	1.289323
3	Caracteristicas	1.066121
4	Habitaciones	2.251426
5	Aseos	3.708820
6	Terraza	1.088883
7	Piscina	1.243464
8	Garaje	1.031044
9	CUDIS	1.017186
10	Latitud	1.043765
11	Longitud	1.037766
12	Metros	4.123837
13	euribor	inf
14	numero_hipotecas	inf
15	ipc	inf
16	tipo_interes	inf
17	paro	inf
18	Precio	2.039562

	feature	VIF
0	Año	322491.114999
1	Mes	94.201712
2	Dia	1.289259
3	Caracteristicas	1.066117
4	Habitaciones	2.251408
5	Aseos	3.708478
6	Terraza	1.088787
7	Piscina	1.243448
8	Garaje	1.030892
9	CUDIS	1.017126
10	Latitud	1.043765
11	Longitud	1.037600
12	Metros	4.123804
13	PCA1	1.382645
14	PCA2	9.116646
15	Precio	2.039479

Figura 10: Factor de Inflación de la Varianza.

Como podemos ver, dado que para que exista una multicolinealidad alta debemos tener un VIF de al menos 5, podremos concluir que habremos resuelto el problema que teníamos en todas nuestras variables con la única excepción de la fecha, que a pesar de estar muy relacionada con las nuevas variables PCA, no debería dar problemas. Aparte de esto, nunca tuvimos problemas de multicolinealidad con la variable objetivo Precio, por lo que podemos dar nuestro conjunto con **51585 filas** y las **16 columnas** que se ven en la figura 10, por preparado para el entrenamiento de los modelos.

3.2 Árboles de decisión

Una vez tenemos nuestro dataset procesado, es el momento de llevar a cabo el entrenamiento de los modelos para lograr el objetivo del trabajo, el cuál se puede traducir en un problema de regresión como es predecir el valor del Precio en función de las características de la vivienda. Para esto empezaremos entrenando modelos basados en árboles de decisión, no sin antes explicar qué son y cómo funcionan.

Un árbol de decisión es un modelo predictivo que divide un conjunto de datos en ramas más pequeñas y más homogéneas basándose en decisiones sucesivas. Cada nodo del árbol representa una característica del conjunto de datos, y cada rama representa una decisión que lleva a diferentes resultados.

Características Principales:

- Interpretabilidad: Los árboles de decisión son fáciles de entender e interpretar, ya que simulan el proceso de toma de decisiones humano.

- No linealidad: Pueden capturar relaciones no lineales entre las características y la variable objetivo.

- Manejo de diversos tipos de datos: Pueden manejar datos numéricos y categóricos.

- Poda: Para evitar el sobreajuste, los árboles pueden ser "podados", lo que limita su profundidad.

Por otro lado, la regresión es una técnica estadística y de aprendizaje automático para modelar y analizar la relación entre variables. Busca establecer una relación entre una variable dependiente y una o más variables independientes. En su forma más simple (regresión lineal simple), se ajusta una línea a los datos para minimizar la suma de las diferencias al cuadrado entre los valores observados y los valores predichos.

Los árboles de decisión se pueden aplicar a problemas de regresión (conocidos como árboles de regresión). En lugar de predecir una clase, predicen un valor continuo, y cuentan con diversas ventajas:

- A diferencia de la regresión lineal, pueden manejar relaciones complejas y no lineales entre las características.
- Son generalmente más robustos a los valores atípicos o dispersos que los modelos de regresión lineal.
- Pueden modelar interacciones complejas entre variables.

Vamos ahora a ver los diferentes enfoques que hemos utilizado para construir nuestros modelos:

3.2.1 Random Forest

Random Forest es un método de ensamble que crea múltiples árboles de decisión durante el entrenamiento y produce la media de las predicciones de los árboles individuales para la regresión, o el voto mayoritario para la clasificación. Cada árbol se entrena con una muestra aleatoria de los datos y un subconjunto de las características, lo que aumenta la diversidad y reduce el sobreajuste.

Características principales:

- Reducción de sobreajuste: Al promediar varios árboles, reduce el riesgo de sobreajuste presente en árboles de decisión individuales.

- Robustez: Es menos sensible a valores atípicos y ruido.

- Manejo de diversos tipos de datos: Puede manejar datos numéricos y categóricos sin necesidad de preprocesamiento extenso.

3.2.2 Gradient Boosting

Gradient Boosting construye secuencialmente árboles de decisión, donde cada árbol nuevo intenta corregir los errores del árbol anterior. Se basa en el método de gradiente descendente para minimizar la pérdida.

Características principales:

- Enfoque secuencial: A diferencia de Random Forest, que construye árboles de forma independiente, Gradient Boosting los construye de manera secuencial.

- Optimización de pérdida: Minimiza una función de pérdida, lo que lo hace flexible para adaptarse a diferentes problemas y métricas.

3.2.3 XG Boost

XGBoost es una optimización del Gradient Boosting. Introduce mejoras en la eficiencia computacional, uso de recursos y precisión.

Características principales:

- Optimización y rendimiento: Incluye optimizaciones en la velocidad y el uso de memoria.
- Manejo de datos faltantes: Puede manejar internamente datos faltantes.
- Regularización: Incorpora términos de regularización para controlar el sobreajuste, lo que no está presente en el Gradient Boosting tradicional.

3.2.4 Ligth GBM

LightGBM es una implementación de Gradient Boosting que utiliza un algoritmo basado en histogramas, lo que le permite ser más eficiente con conjuntos de datos grandes y de alta dimensionalidad.

Características principales:

- Alto rendimiento con grandes datos: Más eficiente en términos de memoria y velocidad, especialmente con grandes volúmenes de datos.
- Soporte para categorías: Maneja características categóricas automáticamente.
- Reducción del uso de memoria: Su técnica basada en histogramas reduce el uso de memoria.

Principales diferencias:

- **Random Forest vs métodos de Boosting**: Random Forest construye árboles de manera independiente, mientras que los métodos de boosting construyen árboles de manera secuencial. Random Forest es generalmente más robusto y menos propenso al sobreajuste, pero los métodos de boosting a menudo pueden ser más precisos.

- **Gradient Boosting vs XGBoost**: XGBoost mejora el Gradient Boosting tradicional en eficiencia y rendimiento, con optimizaciones adicionales y manejo de regularización.

- **XGBoost vs LightGBM**: Ambos son variantes de Gradient Boosting, pero LightGBM es más eficiente en el manejo de grandes conjuntos de datos, gracias a su algoritmo basado en histogramas y su enfoque en la eficiencia computacional.

3.2.5 Optimización de Hiperparámetros

Aparte de estos métodos se han utilizado diversas técnicas para optimizar el entrenamiento y la búsqueda de hiperparámetros, como son la validación cruzada con k-fold y la optimización bayesiana en la búsqueda de los mejores hiperparámetros:

Validación Cruzada

La validación cruzada es una técnica utilizada para evaluar la capacidad predictiva de modelos estadísticos y para proteger contra el sobreajuste. Es especialmente útil en situaciones donde no se dispone de una gran cantidad de datos. A continuación, se explica cómo funciona y las ventajas de aplicarla en modelos de árboles de regresión:

Funcionamiento:

- División de datos: El conjunto de datos se divide en 'k' subconjuntos (o "pliegues"). Comúnmente se utiliza un valor de 'k' como 5 o 10, pero puede variar dependiendo del tamaño del conjunto de datos. En nuestro caso utilizaremos 5.

- Entrenamiento y evaluación iterativos: El modelo se entrena y evalúa 'k' veces. En cada iteración, uno de los 'k' subconjuntos se utiliza como conjunto de prueba y el resto se utiliza para entrenar el modelo.

- Promedio de resultados: Finalmente, se promedian los resultados de las 'k' iteraciones para obtener una única estimación del rendimiento del modelo.

Ventajas de la validación cruzada en Árboles de Regresión:

- Estimación más precisa del rendimiento del modelo: Al usar todo el conjunto de datos tanto para entrenar como para validar el modelo, la validación cruzada proporciona una medida más precisa de cómo se desempeñará el modelo en datos no vistos.

- Protección contra el sobreajuste: Los árboles de regresión son propensos al sobreajuste, especialmente si se les permite crecer demasiado profundos o complejos. La validación cruzada ayuda a identificar si un modelo está sobreajustado, ya que un modelo sobreajustado tendrá un buen rendimiento en los datos de entrenamiento pero un rendimiento pobre en los datos de validación.

- Selección de hiperparámetros: La validación cruzada es útil para seleccionar los hiperparámetros óptimos para el modelo de árbol de regresión, como la profundidad máxima del árbol, el número mínimo de muestras para dividir un nodo, etc.

- Uso eficiente de los datos: En lugar de dividir los datos en conjuntos de entrenamiento y prueba fijos, la validación cruzada asegura que cada observación se use tanto para entrenamiento como para validación. Esto es particularmente útil cuando se dispone de un conjunto de datos limitado.

Optimización Bayesiana

La optimización bayesiana es un método avanzado para encontrar los mejores hiperparámetros de un modelo, basado en el principio bayesiano de estadísticas. Se utiliza para optimizar funciones costosas y es particularmente útil cuando la evaluación de cada conjunto de hiperparámetros es computacionalmente intensiva, como en los modelos de árboles de regresión.

Funcionamiento:

- Modelo de probabilidad: La optimización bayesiana comienza con la construcción de un modelo probabilístico de la función objetivo (en este caso, la función que evalúa los hiperparámetros del modelo). Este modelo, usualmente un Proceso Gaussiano (GP), es una representación de la creencia sobre el espacio de búsqueda.

- Selección de puntos basada en criterios: Utiliza criterios como la "Expected Improvement" (Mejora Esperada) para decidir dónde evaluar la función objetivo a continuación. En lugar de buscar aleatoriamente, elige puntos que probablemente ofrezcan una mejora sobre el mejor resultado encontrado hasta ahora.

- Actualización del modelo con nuevos datos: Después de cada evaluación, el modelo probabilístico se actualiza con los nuevos resultados. Esto refina continuamente la comprensión del modelo sobre el espacio de búsqueda.

- Iteración hasta convergencia: El proceso se repite hasta que se cumple un criterio de convergencia, como un número máximo de iteraciones o una mejora mínima entre iteraciones. En nuestro caso se ha establecido dicho límite en 50 iteraciones.

Ventajas de la optimización Bayesiana en la búsqueda de hiperparámetros para Árboles de Regresión:

- Eficiencia en la búsqueda: La optimización bayesiana es más eficiente que los métodos de búsqueda aleatoria o de búsqueda en rejilla (grid search), especialmente en espacios

de alta dimensión. Esto se debe a su enfoque en evaluar los hiperparámetros que tienen más probabilidades de mejorar el rendimiento.

- Mejor uso de los recursos computacionales: Al ser más eficiente, reduce la cantidad de recursos computacionales necesarios para encontrar los mejores hiperparámetros.

- Evita el sobreajuste: En los árboles de regresión, un ajuste fino de los hiperparámetros puede prevenir el sobreajuste. La optimización bayesiana, al ser más precisa, puede encontrar un equilibrio óptimo entre el sesgo y la varianza.

- Manejo de espacios de búsqueda complejos: Puede manejar espacios de búsqueda no lineales y con múltiples mínimos locales, algo común en la configuración de hiperparámetros para árboles de regresión.

3.3 Aprendizaje profundo

El aprendizaje profundo o deep learning es una rama del aprendizaje automático que utiliza redes neuronales profundas para modelar y resolver diversos problemas, incluyendo la predicción de variables continuas (regresión) basada en sus características.

1. Estructura de las Redes Neuronales Profundas:

- Neuronas y capas: Una red neuronal está compuesta de unidades básicas llamadas neuronas, organizadas en capas. Hay una capa de entrada que recibe las características, varias capas ocultas que procesan la información, y una capa de salida que emite la predicción.
- Conexiones y pesos: Cada neurona en una capa está conectada a las neuronas de la siguiente capa. Estas conexiones tienen pesos que se ajustan durante el entrenamiento.

2. Propagación hacia adelante (Forward Propagation):

- Entrada de datos: Los datos de entrada (características) se introducen en la capa de entrada.
- Cálculos en capas ocultas: Cada neurona calcula una suma ponderada de sus entradas y aplica una función de activación (como ReLU, sigmoide, etc.) para introducir no linealidades en el modelo.

3. Predicción de salida:

- Capa de salida: En problemas de regresión, la capa de salida suele tener una sola neurona que emite un valor continuo, que es la predicción del modelo.

4. Función de pérdida y retropropagación (Backpropagation):

- Cálculo de error: Se utiliza una función de pérdida (como el error cuadrático medio) para calcular la diferencia entre la predicción y el valor real.
- Ajuste de pesos: Mediante backpropagation, el error se propaga de vuelta a través de la red, y se utilizan algoritmos de optimización (como SGD, Adam, etc.) para ajustar los pesos de las conexiones para minimizar el error. En nuestro caso se utilizará **Adam**.

Cabe destacar que en nuestro caso se ha utilizado la función **MAE** como función de pérdida, ya que nuestros datos cuentan con una distribución que se aleja mucho de la normal, teniendo la variable objetivo una **kurtosis muy elevada** de **16.08**. En este tipo de distribuciones, la gran disparidad de valores puede hacer que el error cuadrático medio sea muy elevado, no proporcionando una visión fidedigna de los resultados conseguidos y no generando así un entrenamiento óptimo de la red. Es por esto que en estos casos suele ser mejor usar como función de pérdida el error medio absoluto.

5. Entrenamiento Iterativo:

- Épocas y Mini-Batch: El entrenamiento se realiza en múltiples iteraciones o épocas, y los datos a menudo se dividen en mini-lotes para mejorar la eficiencia y la estabilidad del entrenamiento.

6. Prevención de sobreajuste:

- Técnicas como Dropout o regularización: Se aplican técnicas para prevenir el sobreajuste, como el dropout (que desactiva aleatoriamente algunas neuronas durante el entrenamiento) o la regularización (que añade un término de penalización a la función de pérdida). Aparte de esto, se utilizará también validación cruzada como en los casos anteriores, para encontrar el mejor modelo evitando el sobreajuste.

7. Afinación de hiperparámetros:

- Elección de arquitectura y parámetros: La selección de la arquitectura de la red (número de capas, número de neuronas por capa), tasa de aprendizaje, y otros hiperparámetros es crucial para el buen desempeño del modelo. En nuestro caso utilizaremos la **Optimización Bayesiana**, al igual que hicimos con los modelos basados en árboles, para buscar los mejores hiperparámetros.

Ventajas en la predicción de variables continuas:

- Modelado de relaciones complejas: Las DNN pueden modelar relaciones complejas y no lineales entre las características y la variable objetivo.

- Capacidad de aprendizaje automático: Pueden aprender automáticamente características relevantes de los datos, lo que es especialmente útil en casos donde las relaciones entre las características y la variable objetivo son difíciles de capturar con modelos lineales o tradicionales.

3.3.1 Redes Neuronales Densas (DNN)

Las redes neuronales densas, también conocidas como redes neuronales completamente conectadas, son un tipo fundamental de redes neuronales utilizadas en el aprendizaje profundo cuyas características básicas son:

- **Estructura de capas completamente conectadas**:
  - Neuronas en capas: Las redes densas se componen de capas de neuronas, donde cada neurona en una capa está conectada a todas las neuronas de la capa anterior y de la capa siguiente.
  - Capa de entrada, capas ocultas y capa de salida: Tienen una capa de entrada que recibe los datos, una o más capas ocultas que procesan los datos, y una capa de salida que produce la predicción o clasificación.

- **Funcionamiento de las neuronas**:
  - Suma ponderada: Cada neurona realiza una suma ponderada de sus entradas, utilizando pesos que se ajustan durante el entrenamiento.
  - Función de activación: Después de la suma ponderada, se aplica una función de activación (como ReLU, sigmoide o tanh) para introducir no linealidades en el modelo.

- **Propagación hacia adelante (Forward Propagation):**
  - Flujo de datos: Los datos se introducen en la capa de entrada y fluyen a través de las capas ocultas hasta llegar a la capa de salida, donde se emite la predicción.

- **Retropropagación y aprendizaje**:
  - Ajuste de pesos: Mediante el algoritmo de retropropagación y técnicas de optimización como el Descenso del Gradiente, los pesos de las conexiones se ajustan para minimizar la función de pérdida (error entre la predicción y el valor real).

- **Regularización y prevención de sobreajuste**:
  - Técnicas como Dropout y regularización L1/L2: Se utilizan para evitar que la red se ajuste demasiado a los datos de entrenamiento, lo que mejoraría su capacidad de generalizar a datos no vistos. Las utilizaremos sobre todo

- **Flexibilidad en el diseño de la arquitectura**:
    - Número y tamaño de capas: Se puede experimentar con el número de capas y el número de neuronas en cada capa para encontrar la arquitectura óptima para un problema específico.

- **Aplicabilidad universal**:
    - Uso en diversos problemas: Adecuadas para una amplia gama de tareas de aprendizaje automático, incluyendo clasificación, regresión y más.

Las redes neuronales densas son conocidas por su versatilidad y capacidad para aprender representaciones complejas de los datos, pero también requieren una cuidadosa configuración y pueden ser propensas al sobreajuste, especialmente en conjuntos de datos más pequeños o menos complejos.

3.3.2 Redes Neuronales Convolucionales (CNN)

Aunque las CNN están originalmente diseñadas para el tratamiento de imágenes, lo cierto es que también pueden ser utilizadas en otros ámbitos como la predicción de series temporales o la estimación de atributos objetivo en base a las características, como es nuestro caso con el precio de la vivienda. Para poder llevar a cabo esta técnica y poder utilizar una capa Conv1D, la clave será tratar cada una de las características como un canal con formato unidimensional.
Para entender cómo funcionan las capas Conv1D y cómo se pueden aplicar a datos que no son imágenes, como series temporales o datos tabulares, es útil revisar primero qué hacen las capas convolucionales y luego cómo se adaptan a estos contextos:

Capas Convolucionales en el contexto de imágenes:

En el procesamiento de imágenes, una capa convolucional (como Conv2D en el caso de las CNN) aplica una serie de filtros a la imagen para extraer características. Estos filtros se deslizan a través de la imagen (2D) como si esta estuviese dividida en una cuadrícula y fuésemos recorriendo cada pequeño cuadrito y extrayendo información cuantitativa de las características de este, procesando la información espacial. En las imágenes, cada "canal" suele corresponder a un color (por ejemplo, rojo, verde y azul en imágenes RGB).

Capas Convolucionales para datos unidimensionales (Conv1D):

En el caso de los datos unidimensionales, como las series temporales o los datos tabulares, utilizamos capas Conv1D. Aquí, los filtros se deslizan a lo largo de una sola dimensión. En lugar de tratar la información espacial como en las imágenes, se tratan secuencias de datos o características individuales.
Cuando aplicamos Conv1D a datos tabulares (como en nuestro caso de estimación de precios de viviendas), tratamos cada característica (o columna en nuestros datos tabulares)

como si fuera un "canal" en el sentido de procesamiento de imágenes. El filtro de la capa Conv1D se deslizará a través de cada una de estas características, buscando patrones útiles para la tarea de predicción.

3.3.3 Redes Neuronales Recurrentes (RNN - LSTM)

Las Redes Neuronales Recurrentes (RNN) son una clase de redes neuronales diseñadas para manejar secuencias de datos, como series temporales o texto. Son únicas en su capacidad de procesar secuencias de longitud variable, recordando información pasada a través de conexiones recurrentes.

Funcionamiento básico de las RNN:

- **Estructura y flujo de datos**: En una RNN, cada nodo de la red procesa un elemento de la secuencia de entrada, uno tras otro. Esta unidad procesa la entrada actual (por ejemplo, una palabra en una oración) junto con la información transmitida del paso anterior.
- **Conexiones recurrentes**: A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones recurrentes que les permiten mantener un 'estado' o memoria a corto plazo. Este estado almacena información sobre los elementos procesados anteriormente en la secuencia.
- **Problema de las pependencias a largo plazo**: Las RNN simples tienen dificultades para aprender dependencias a largo plazo en secuencias debido al problema del desvanecimiento y la explosión del gradiente. Esto ocurre porque los gradientes, que se utilizan en el algoritmo de retropropagación, tienden a desvanecerse o crecer exponencialmente en secuencias largas.

Redes LSTM (Long Short-Term Memory):

Las redes LSTM son una variante de las RNN diseñadas para superar el problema de las dependencias a largo plazo.

Estructura de una celda LSTM: Una celda LSTM es más compleja que un nodo típico de una RNN. Contiene varias 'puertas' que controlan el flujo de información:
- **Puerta de olvido**: Decide qué información se descarta del estado de la celda.
- **Puerta de entrada**: Actualiza el estado de la celda con nuevas entradas.
- **Puerta de salida**: Determina qué parte del estado de la celda se transmite al siguiente paso de tiempo.

Estado de la celda: El corazón de una LSTM es su estado de celda, una especie de 'memoria' que viaja a lo largo de la cadena de la secuencia con mínimas alteraciones. Las puertas mencionadas anteriormente modifican este estado de celda de manera controlada.

Funcionamiento de las puertas: Cada puerta es una estructura tipo sigmoide que toma la salida anterior y la entrada actual, y decide cuánto dejar pasar de cada una. Por ejemplo, la puerta de olvido puede decidir eliminar información irrelevante del estado de la celda, mientras que la puerta de entrada añade nueva información relevante.

Capacidad para aprender dependencias a largo plazo: Gracias a este mecanismo de puertas y al estado de la celda, las LSTM pueden aprender cuándo recordar y cuándo olvidar, lo que les permite mantener dependencias a largo plazo en los datos.

Estas redes están principalmente orientadas al tratamiento de series temporales. Es por esto que para utilizarlas se ha ordenado y transformado nuestros datos para que puedan ser tratados como tal. Aunque estas redes podrían encajar mucho mejor en otro tipo de análisis, como colapsar los datos obtenidos a nivel de día e intentar predecir el precio medio del mercado a largo plazo, se comprobará su comportamiento con nuestro conjunto de datos y objetivo.

3.3.4 Transformers

Los Transformers son una arquitectura de modelo de aprendizaje profundo que ha revolucionado el campo del procesamiento del lenguaje natural (NLP) entre otros. Introducidos en 2017, los Transformers superan algunas de las limitaciones de las RNN y LSTM, especialmente en términos de manejar dependencias a largo plazo y eficiencia computacional.

Componentes clave de los Transformers:

- **Mecanismo de atención**: En el corazón de los Transformers está el mecanismo de atención, específicamente la "atención auto-dirigida". Este mecanismo permite al modelo ponderar la importancia relativa de diferentes partes de la entrada. En el contexto de NLP, por ejemplo, permite al modelo enfocarse en partes relevantes de una oración o un texto para realizar una tarea específica.
- **Estructura sin recurrencia**: A diferencia de las RNN y LSTM, los Transformers no procesan los datos de manera secuencial. En cambio, procesan toda la entrada a la vez. Esta característica permite un paralelismo masivo durante el entrenamiento y supera el problema de dependencia secuencial de las RNN.
- **Capas codificadoras y decodificadoras**: La arquitectura Transformer original se compone de bloques de codificación y decodificación. Los codificadores procesan la entrada, y los decodificadores generan la salida. Cada bloque consta de una capa de atención seguida de redes neuronales de avance (feed-forward).
- **Conexiones residuales y normalización de capas**: Cada subcapa en los bloques del Transformer, incluidas las capas de atención y las redes neuronales de avance, tiene

una conexión residual alrededor de ella seguida de una normalización de capa. Esto ayuda a evitar el problema del desvanecimiento del gradiente en redes profundas.

Funcionamiento de los Transformers:

- **Codificación de entrada**: Las entradas al Transformer son codificadas con embeddings que incluyen información posicional, proporcionando un contexto sobre la posición de cada palabra, o dato en general, en la secuencia.
- **Atención auto-dirigida**: En la capa de atención, el modelo calcula conjuntos de puntuaciones de atención que determinan cuánto se enfoca en otras partes de la entrada para cada palabra. Esto permite capturar contextos y dependencias, independientemente de la distancia entre las palabras en la secuencia.
- **Flujo de información**: La información fluye a través de las capas del Transformer, siendo refinada sucesivamente en cada paso. En el caso de tareas de generación de texto, por ejemplo, cada bloque decodificador atiende no solo a la salida del bloque codificador correspondiente, sino también a las salidas de los decodificadores anteriores.
- **Salida final**: La salida del último bloque decodificador se pasa a través de una capa lineal y una capa softmax para generar predicciones, como el siguiente dato en una secuencia.

Aunque originalmente están diseñados para optimizar el aprendizaje en modelos relacionados con procesamiento de textos, análisis de sentimientos y NLP en general, pueden llegar a ser utilizados también en este tipo de problemas de estimación de valores en base a las características. Por ello se va a probar cómo se comportan y ver si pueden mejorar los resultados obtenidos hasta ahora.

En nuestro modelo, cada muestra de los datos tabulares (características de una vivienda) se transforma primero a través de una capa densa, y luego estas representaciones son procesadas por la atención multi-cabeza. Esto permite que el modelo aprenda cómo diferentes características de una vivienda (como tamaño, ubicación, número de habitaciones) interactúan y afectan el precio de la vivienda.
Este modelo puede aportar diversas ventajas, aunque no es común usar Transformers en datos tabulares puede ser útil para capturar relaciones complejas y no lineales entre características. De igual forma, los Transformers son modelos muy potentes y flexibles, lo que les permite adaptarse a una variedad de tareas más allá del procesamiento de lenguaje.

4. Resultados

Antes de proceder a comentar los resultados obtenidos por los modelos, es importante dedicar un espacio a comprender las diferentes métricas que se han utilizado para llevar a cabo la evaluación. Para poder comparar los árboles los modelos basados en árboles de decisión con los modelos basados en aprendizaje profundo se han utilizado las siguientes métricas:

- Mean Squared Error (MSE):

El **Mean Squared Error (MSE)** es una métrica que mide el promedio de los cuadrados de los errores, es decir, la diferencia cuadrática promedio entre los valores observados y los valores predichos. Se define como:

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

donde y_i son los valores reales y $\hat{y}_i$ son los valores predichos.

Un MSE alto indica que el modelo tiene un error considerable en sus predicciones. Dado que MSE penaliza más los errores grandes (debido al cuadrado), un MSE alto podría ser un indicador de que hay algunas predicciones del modelo que están significativamente alejadas del valor real. Nuesrtos datos serán especialmente sensibles a este error debido a su elevada kurtosis.

- Root Mean Squared Error (RMSE):

El **Root Mean Squared Error (RMSE)** es simplemente la raíz cuadrada del MSE. Proporciona una medida del error en las mismas unidades que la variable dependiente y se define como:

$$\text{RMSE} = \sqrt{MSE}$$

El RMSE proporciona una medida más interpretable del error promedio del modelo, ya que está en las mismas unidades que los precios de las viviendas. Un RMSE más bajo significa que el modelo es más preciso en sus predicciones.

- Mean Absolute Error (MAE):

El **Mean Absolute Error (MAE)** mide el promedio de los errores absolutos, es decir, la diferencia promedio entre los valores observados y los valores predichos, sin considerar la dirección, y se define como:

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

El MAE es una medida robusta que no penaliza tanto los errores grandes como el MSE. Proporciona una idea del error típico que se puede esperar en las predicciones del modelo y se espera más bajo que los anteriores en el contexto del entrenamiento de nuestros modelos, teniendo en cuenta que es menos sensible a los errores grandes provocados por la existencia de outliers.

- Median Absolute Error (MedAE):

El **Median Absolute Error (MedAE)** mide la mediana de los errores absolutos, es decir, si ordenásemos todos los valores de los errores absolutos de menor a mayor en una lista, el MedAE sería el valor de error que se encontrase en la posición central de dicha lista.
Esta es una métrica especialmente útil cuando los datos toman valores muy dispares, como podría ser nuestro caso, ya que al existir mucha disparidad, los valores grandes pueden influir de forma mucho más notable en la media de error que en la mediana, dando esta métrica una estimación más precisa en estos casos.

- Mean Absolute Percentage Error (MAPE):

El **Mean Absolute Percentage Error (MAE)** mide el promedio del porcentaje de error absoluto cometido en las estimaciones, siendo el porcentaje de error el error absoluto dividido por el valor esperado. Esta métrica se define como:

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

De nuevo será una métrica muy útil en conjuntos con valores muy dispares, ya que al estimar el error en porcentaje tendrá menos influencia cuán grande sea el valor que estemos estimando.

- R^2 Score (Coeficiente de Determinación):

El **R^2 Score** es una medida estadística que indica la proporción de la varianza en la variable dependiente que es predecible a partir de las variables independientes. Varía entre 0 y 1 y de define como:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y})^2}$$

donde $\hat{y}$ es el promedio de los valores observados.

Un R^2 más alto significa que una mayor proporción de la variabilidad en los precios de las viviendas es explicada por el modelo. Un R^2 cercano a 1 indica que el modelo explica una gran parte de la variación en los precios de las viviendas, mientras que un valor cercano a 0

indica lo contrario. Otra forma de visualizarlo en casos donde es muy cercano a 1, y como lo haremos en el siguiente caso, es utilizando 1 - R^2, ya que esto nos permite comparar mejor los resultados.

4.1 Árboles de regresión

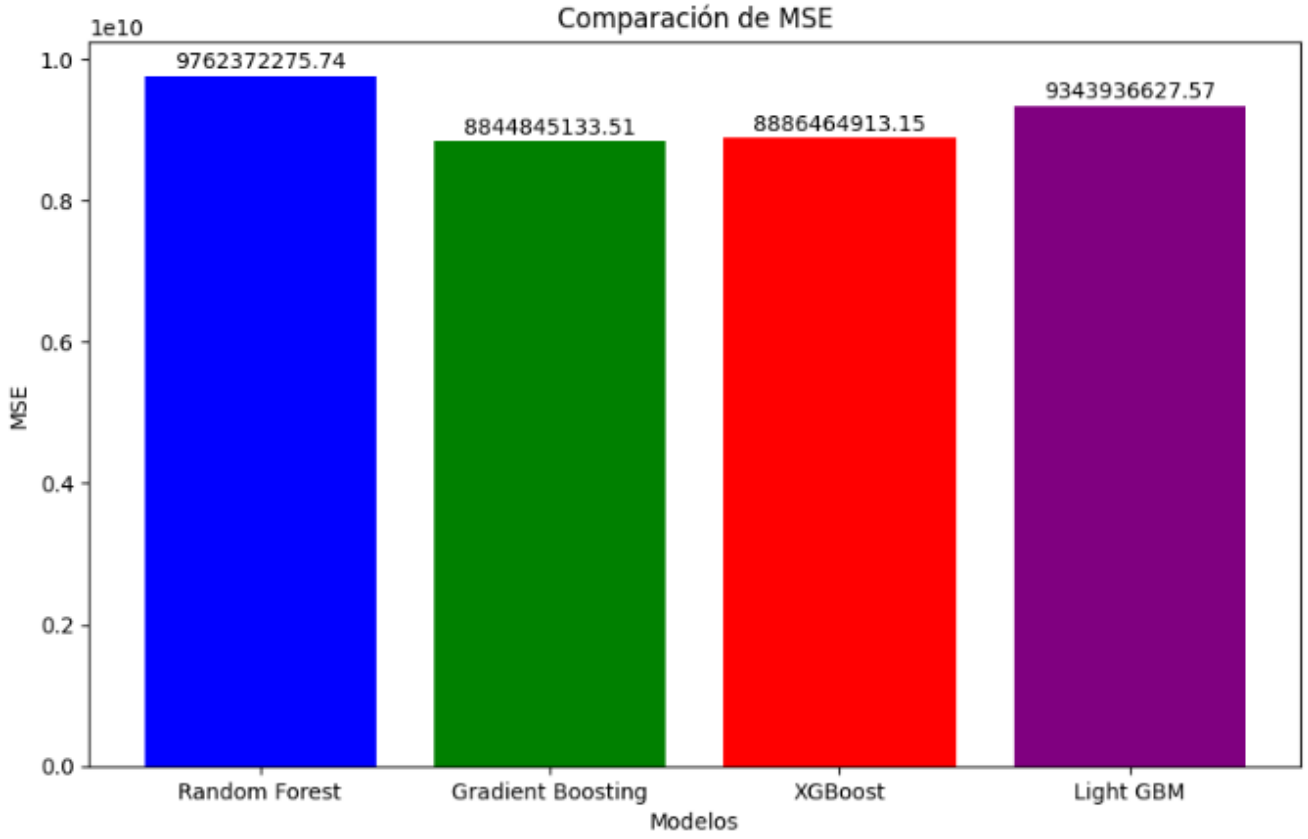
En primer lugar, se ha aplicado un proceso de validación cruzada donde se divide el dataset en 5 subconjuntos y todos se usan como conjunto de test para determinar cuál es la arquitectura de árbol que mejor se ajusta a nuestros datos. Se muestran a continuación las métricas obtenidas tras la realización de este proceso:

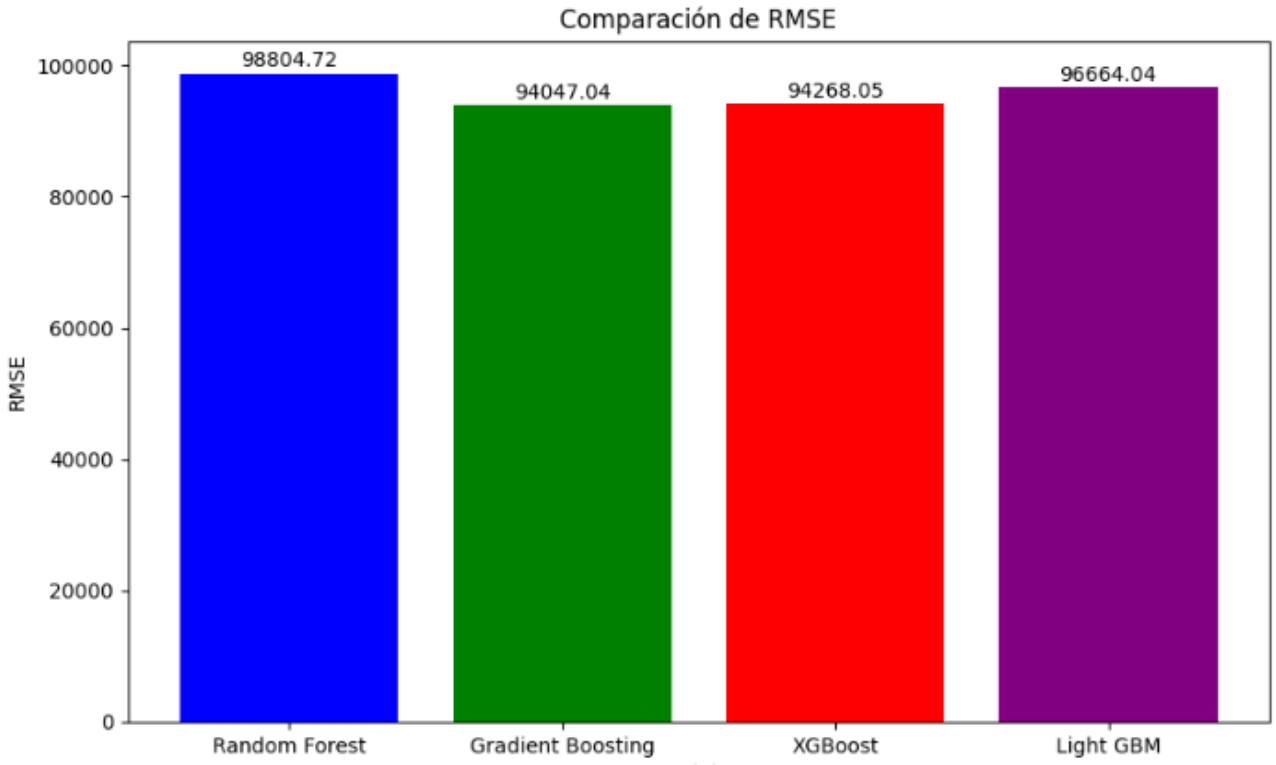
	MSE	RMSE	MAE	R2	Tiempo de entrenamiento
Random Forest	1.560606e+10	124924.215140	54241.310005	0.079756	296.724270
Gradient Boosting	1.913398e+10	138325.632832	67871.448534	0.097786	179.912091
XGBoost	1.698295e+10	130318.660932	58684.733286	0.086793	53.527200
Light GBM	1.595087e+10	126296.755118	56535.319443	0.081519	43.731656

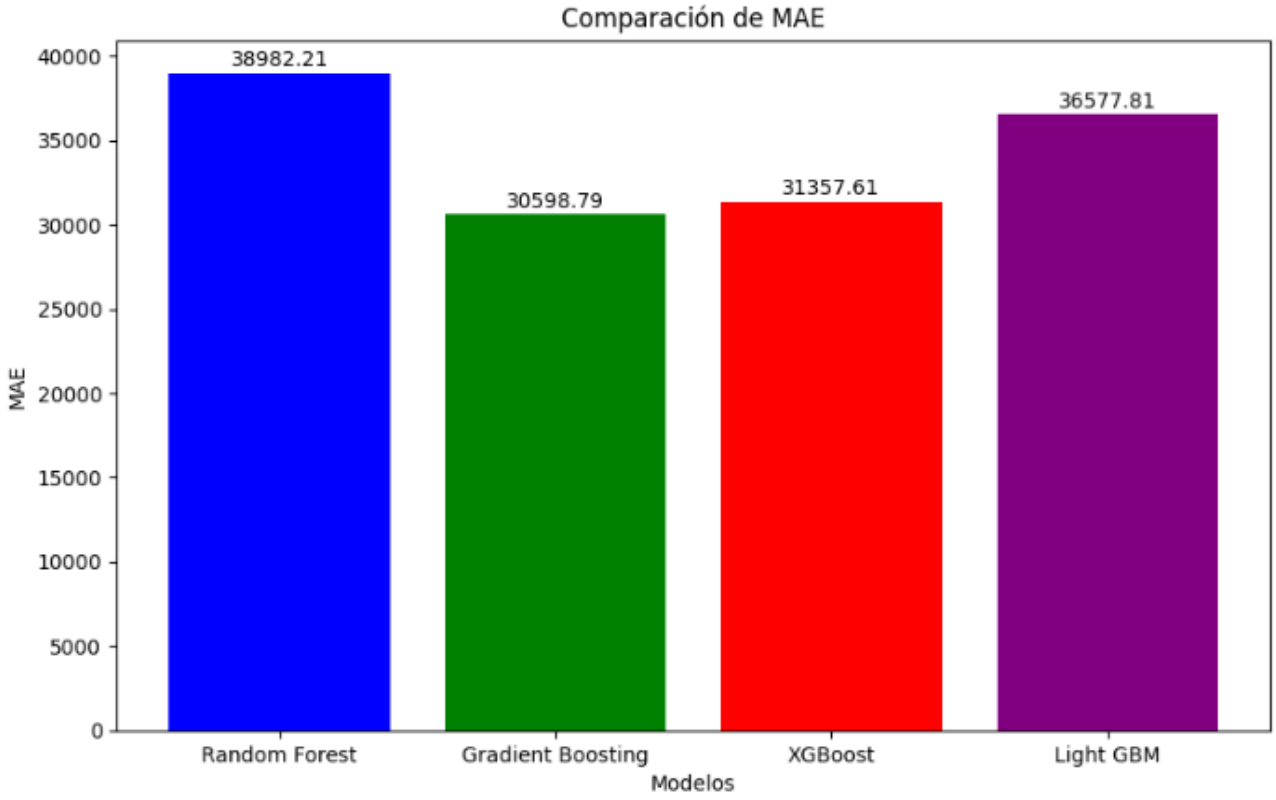
Figura 11: Métricas validación cruzada árboles de regresión.

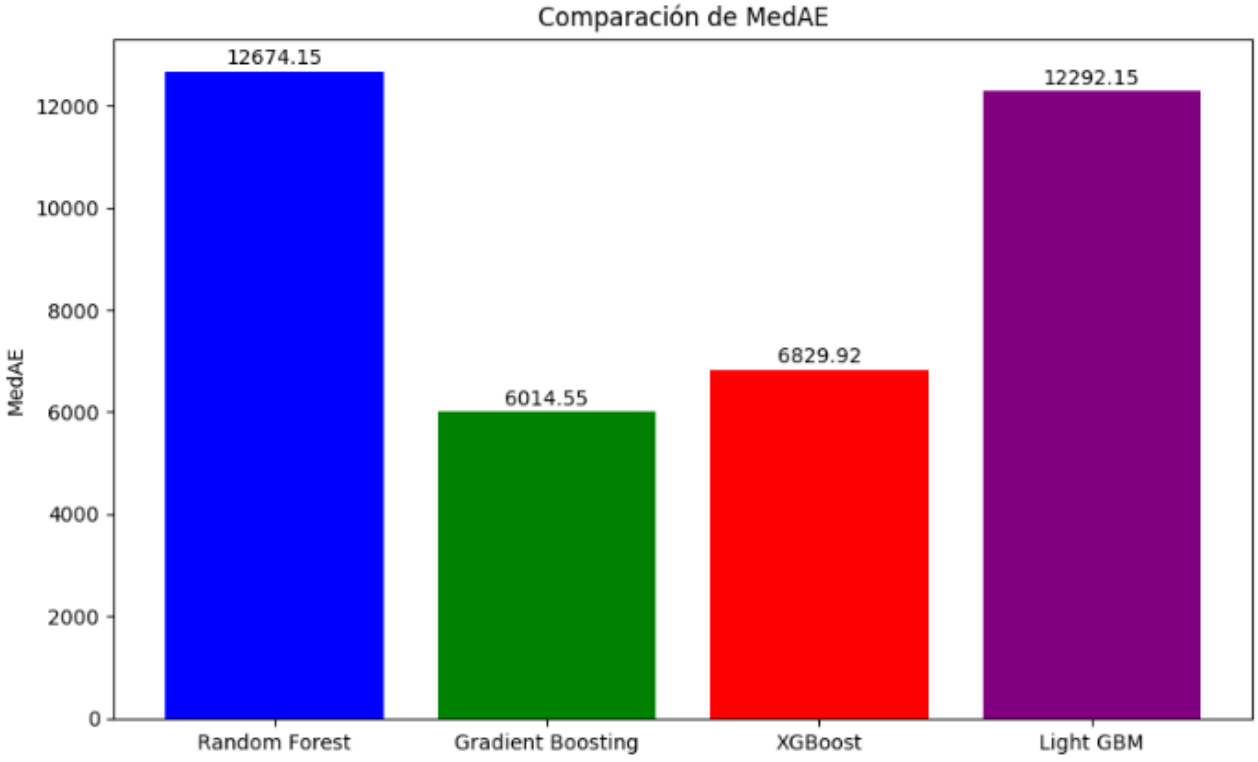
Como podemos observar en esta primera aproximación creada utilizando cross – validation para determinar cuál es la arquitectura de árbol de regresión que mejor se ajusta a nuestros datos. Si tuviéramos que escoger uno solo para realizar la búsqueda de hiperparámetros con optimización bayesiana, el mejor sería Random Forest, aunque seguido muy de cerca por Light GBM, el cuál tarda muchísimo menos en entrenar.

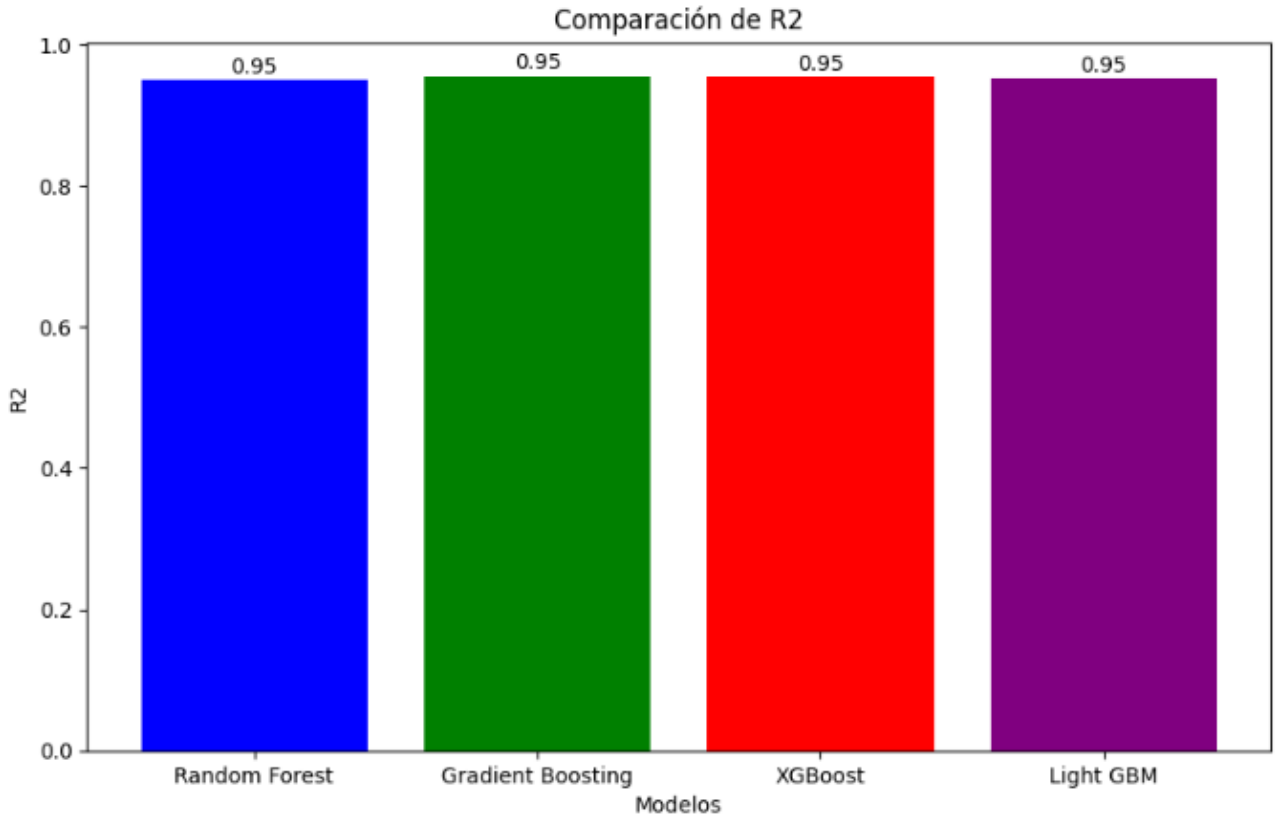
No obstante, al estar los resultados de todos los modelos muy parejos, vamos a aplicar optimización bayesiana sobre todos ellos, combinándolos de nuevo con validación cruzada para mitigar así el sobreajuste.
Tras realizar todo el proceso de optimización, se ha dividido nuestro conjunto en datos de entrenamiento y test en una proporción de 0.8 y 0.2 respectivamente utilizando el método **model_selection.train_test_split** de sklearn y mezclando los datos de manera aleatoria. Los resultados obtenidos tras el entrenamiento los modelos son los siguientes:

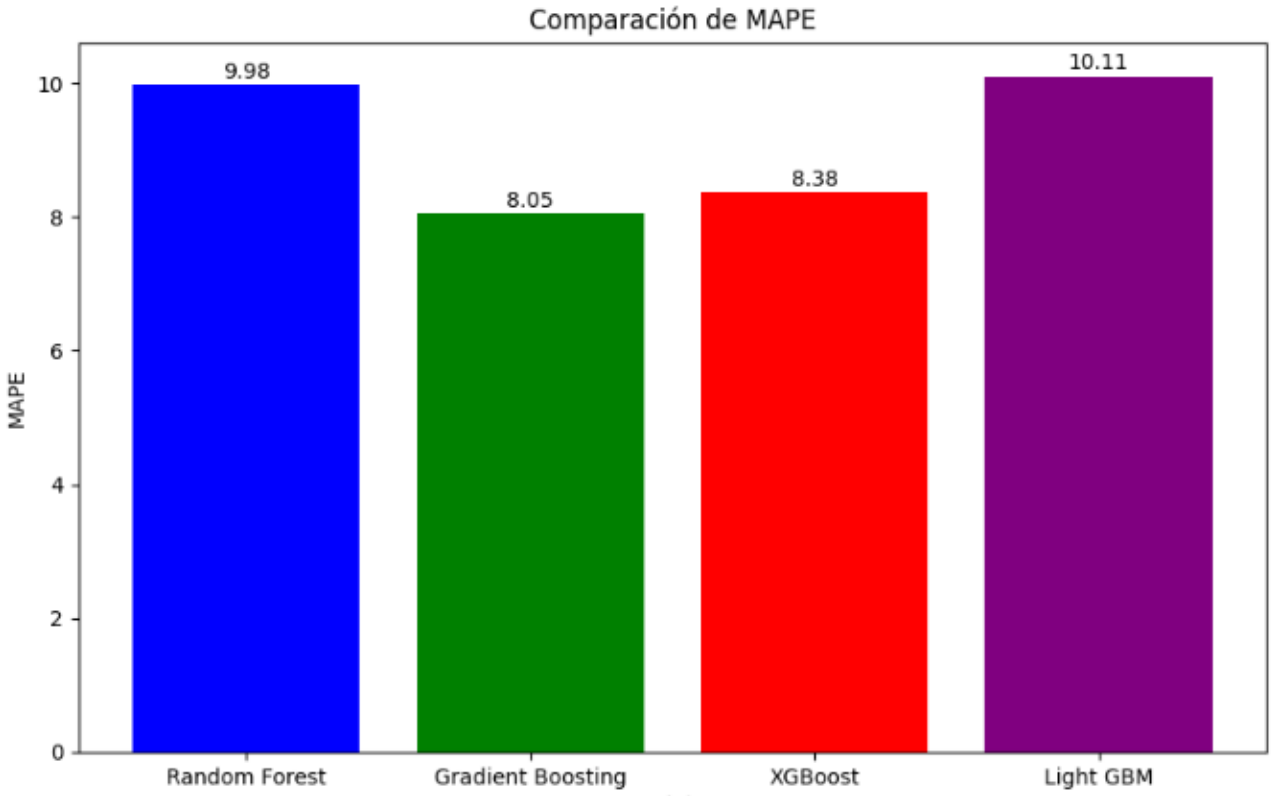

Comparación de MSE
1e10
1.0
0.8
0.6
0.4
0.2
0.0
MSE
9762372275.74
8844845133.51
8886464913.15
9343936627.57
Random Forest
Gradient Boosting
XGBoost
Light GBM
Modelos

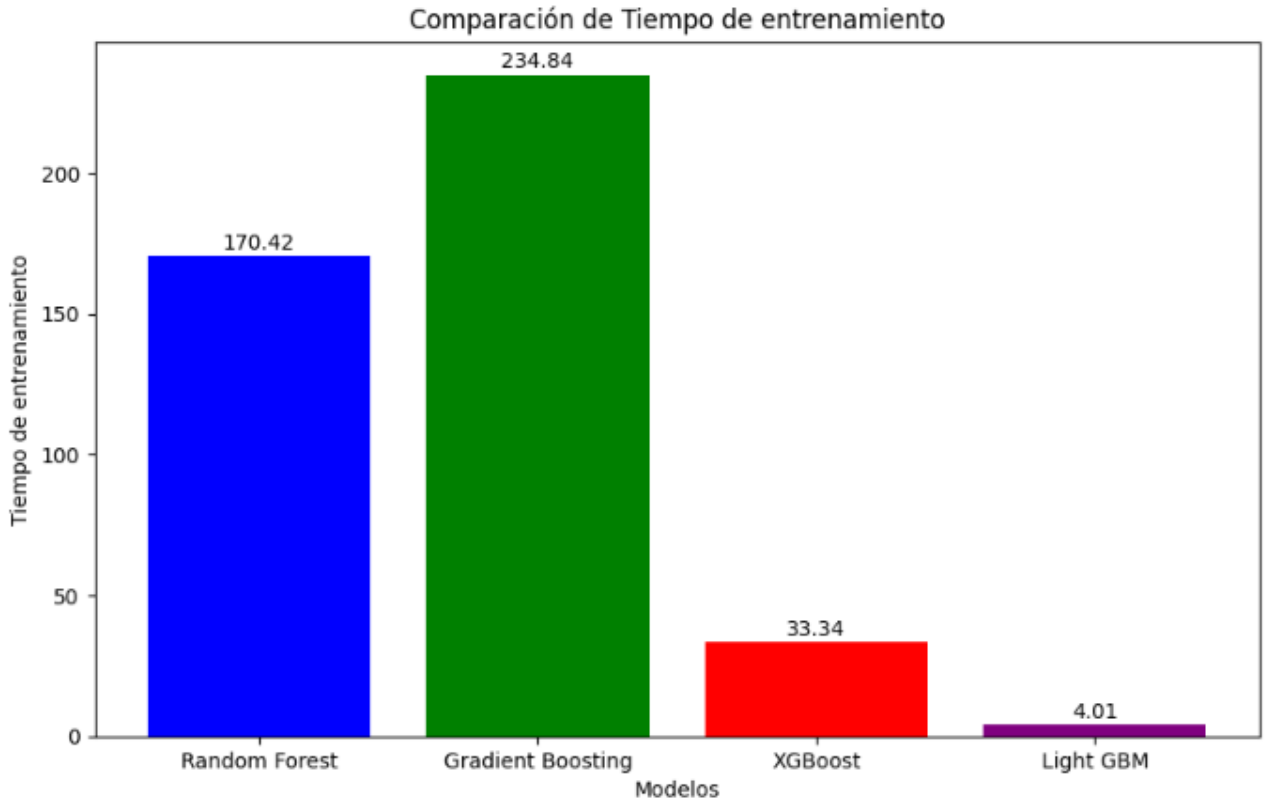

Comparación de RMSE
100000
80000
60000
40000
20000
0
RMSE
98804.72
94047.04
94268.05
96664.04
Random Forest
Gradient Boosting
XGBoost
Light GBM


Comparación de MAE
MAE
Modelos
38982.21
30598.79
31357.61
36577.81
Random Forest
Gradient Boosting
XGBoost
Light GBM


Comparación de MedAE
MedAE
12674.15
6014.55
6829.92
12292.15
Random Forest
Gradient Boosting
XGBoost
Light GBM


Comparación de R2
1.0
0.8
0.6
0.4
0.2
0.0
R2
0.95
0.95
0.95
0.95
Random Forest
Gradient Boosting
XGBoost
Light GBM
Modelos


Comparación de MAPE
10
8
6
4
2
0
MAPE
9.98
8.05
8.38
10.11
Random Forest
Gradient Boosting
XGBoost
Light GBM



	MSE	RMSE	MAE	MedAE	R2	MAPE	Tiempo de entrenamiento
Random Forest	9.762372e+09	98804.717882	38982.208094	12674.150000	0.950108	9.979983	170.424947
Gradient Boosting	8.844845e+09	94047.036814	30598.789779	6014.546247	0.954798	8.047522	234.838128
XGBoost	8.886465e+09	94268.048209	31357.605420	6829.921875	0.954585	8.379833	33.343387
Light GBM	9.343937e+09	96664.039992	36577.809422	12292.148186	0.952247	10.105932	4.006396

Figura 12: Métricas árboles de regresión tras optimización bayesiana.

Los resultados de los errores pueden parecer muy grandes a simple vista para una variable donde la media de valores es 383200.85€, ya que tenemos un RMSE superior a 94000€ , así como un MAE superior a 30000€ incluso en el mejor de los casos. Sin embargo estos resultados se pueden explicar si tenemos en cuenta la alta varianza de la vatiable Precio, que observábamos en la Figura 6, lo cual indica una gran dispersión de valores en nuestros datos. Esto se traduce en que tendremos muchos pisos que valen varios millones de euros, cuyos errores son más grandes en términos absolutos y podrían distorsionar la media global. Estos errores se verán aún más incrementados al usar un error cuadrático como MSE.

Si nos fijamos en la mediana de los errores absolutos podemos ver que esta se sitúa entre 6000€ y 13000€ dependiendo del modelo. Este dato nos da un margen de error mucho más comedido y nos indica que, cuando nos alejamos de los valores más altos, el error absoluto que obtenemos es mucho más aceptable.

Otra métrica muy importante debido a la naturaleza de nuestros datos es el MAPE. Como podemos ver aquí, el error absoluto variará entre el 8% y el 10% de error, lo cual es un margen de error muy bueno en este tipo de problemas, sobre todo si tenemos en cuenta que la mayoría de la gente que establece un precio para su vivienda no siempre suele recurrir a herramientas analíticas o tasaciones oficiales, por lo que el error suele ser mucho mayor que este. E incluso en tasaciones oficiales, se contemplan errores de hasta el 20% del valor en función de la situación del mercado.

Aparte de esto, podemos ver en la gráfica de R^2 que para todos los modelos, el porcentaje de varianza de las variables dependientes que explican está siempre por encima del 95%, lo cual indica que se ajustan muy bien a nuestros datos.

Con todos estos datos, podemos concluir que se ha llegado a unos más que satisfactorios resultados, siendo **los dos mejores modelos obtenidos los basados en Gradient Boosting y en XGBoost**. El primero tendrá un ligero mejor desempeño, aunque si se quisieran realizar futuros ajustes en el modelo escogido, XGBoost sería mucho mejor opción debido a su mayor eficiencia en costo computacional y temporal.

4.2 Modelos de aprendizaje profundo

Se ha llevado a cabo un proceso análogo al anterior aunque con una pequeña diferencia. A diferencia de los modelos basados en árboles de regresión, que pueden manejar de manera eficiente las diferentes escalas en los datos, las redes neuronales son muy sensibles a los conocidos como **errores de escala**. Es por esto que se ha aplicado el método **StandardScaler()** de skalearn.processing para **normalizar todas las variables dependientes en valores entre 0 y 1**. La variable independiente Precio se ha mantenido intacta ya que no se ha observado que fuese necesario normalizarla.

Una vez normalizados los datos, se ha aplicado de nuevo validación cruzada con Kfold, donde k es igual a cinco, para las cuatro arquitecturas de red explicadas en el apartado 3.3 y un modelo híbrido adicional construido a partir de capas LSTM, CNN y densas.

Los resultados han sido los siguientes:

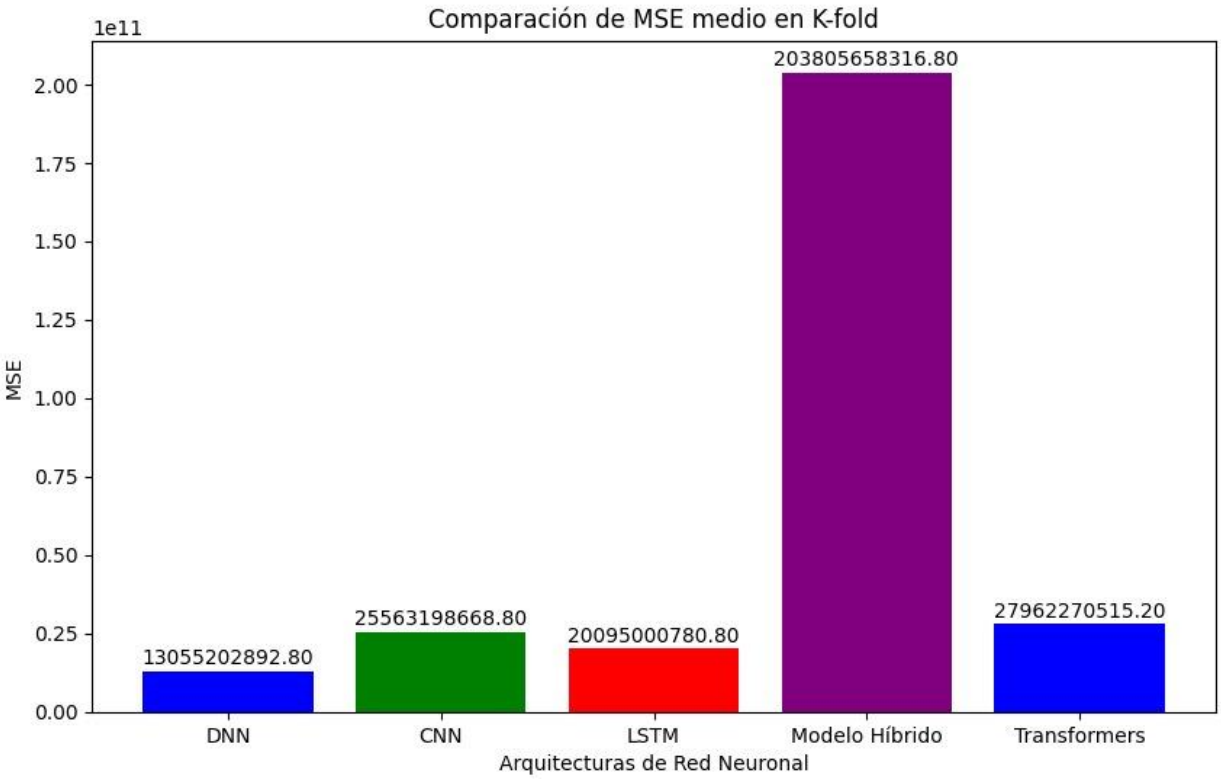


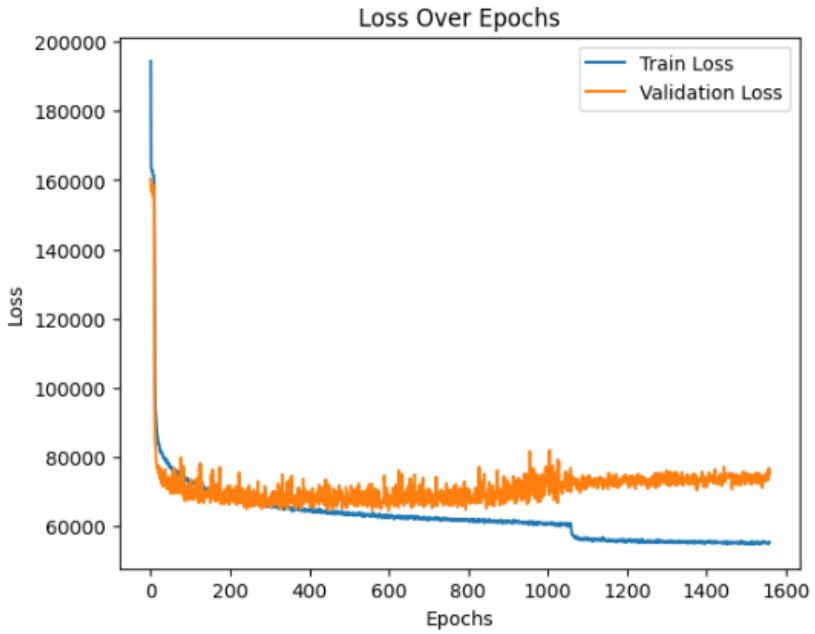
Figura 13: Métricas validación cruzada en redes con aprendizaje profundo.

Como se puede apreciar, parece que todos los modelos proporcionan unos resultados similares, a excepción del construido de forma híbrida, el cuál ha tenido un enorme sobreajuste, y esto ha resultado en que no sea capaz de predecir sobre nuestro conjunto de test de manera eficiente.

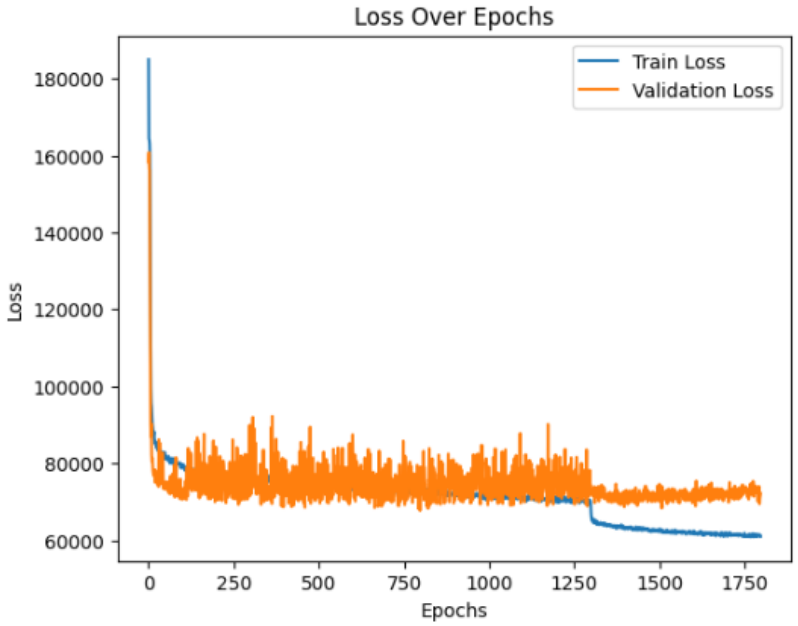
Entre el resto el que aparentemente cuenta con más potencial es el modelo DNN, seguido en la distancia por la red LSTM. Sin embargo, todos han mostrado un desempeño aceptable más allá del modelo hibrído, así que realizaremos la optimización bayesiana combinándola con validación cruzada para evitar el sobreajuste sobre todos ellos y con esto determinaremos cuál es realmente el mejor.

Una vez realizada la optimización bayesiana combinada con la validación cruzada, se han obtenido los mejores hiperparámetros posibles.
A partir de aquí se ha vuelto a usar **train_test_split** para dividir nuestros datos en conjuntos de test, validación y entrenamiento, en una proporción 0.8, 0.1 y 0.1 respectivamente, y se han entrenado los modelos definitivos con los mejores hiperparámetros, siendo el objetivo minimizar la función de pérdida para el conjunto de validación.

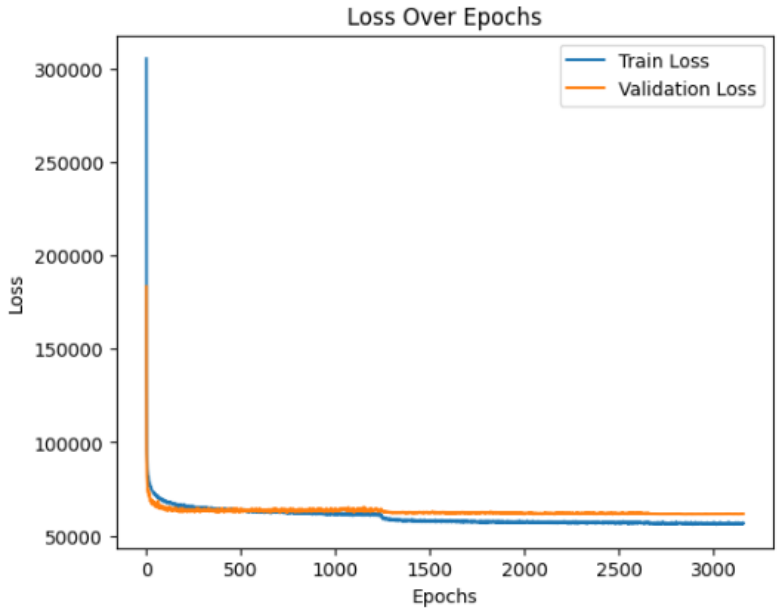
En base a dichos entrenamientos, se ha visualizado el histograma de la mencionada función de pérdida para los conjuntos de entrenamiento y validación, con el objetivo de detectar posibles problemas de sobreajuste. Los resultados han sido los siguientes:



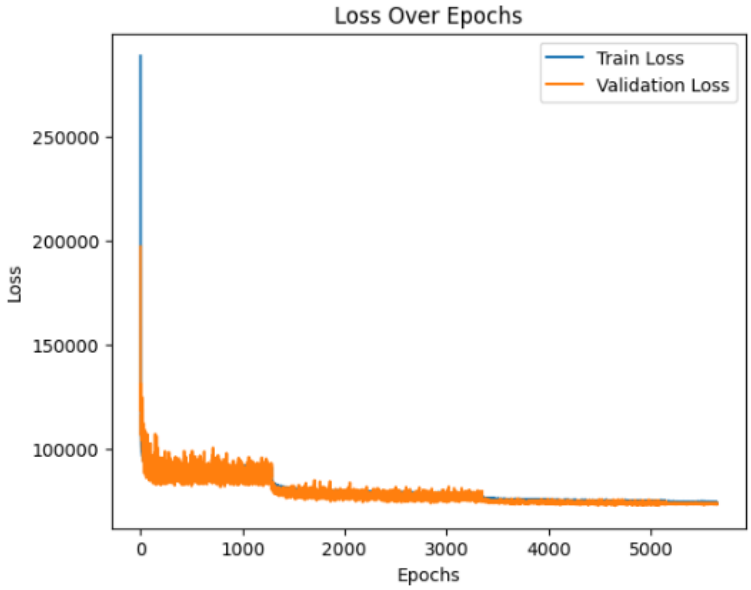
Esta es la gráfica para el modelo DNN. La pérdida disminuye de forma continua tanto para el entrenamiento como para la validación al principio, sin embargo, cerca de las 200 epochs se estabiliza para la varianza. A partir de aquí podemos ver sobre entrenamiento, ya que la línea de la pérdida del conjunto de validación queda cada vez más por encima que la del entrenamiento. La fluctuación en la pérdida de la validación impidió que el EarlyStopping actuase antes, empeorando así los resultados.



Esta es la gráfica para el modelo CNN. Es un caso análogo a lo visto en la gráfica anterior, aunque esta vez se puede ver que el sobreajuste será menor puesto que ambas líneas descienden de forma más pareja hasta que la de validación empieza a fluctuar.



Esta es la gráfica del modelo LSTM. Hasta el momento es la gráfica más prometedora ya que la línea de validación desciende de forma limpia y sin fluctuar. El escalón que podemos ver en las 1244 epochs en el entrenamiento, al igual que pasaba en las otras gráficas, se debe al callback ReduceOnPlateau, que reduce el learning rate cuando la validación deja de mejorar como un último intento de que esto ocurra antes del EarlyStopping, aunque podemos ver que no ha tenido mucho éxito.



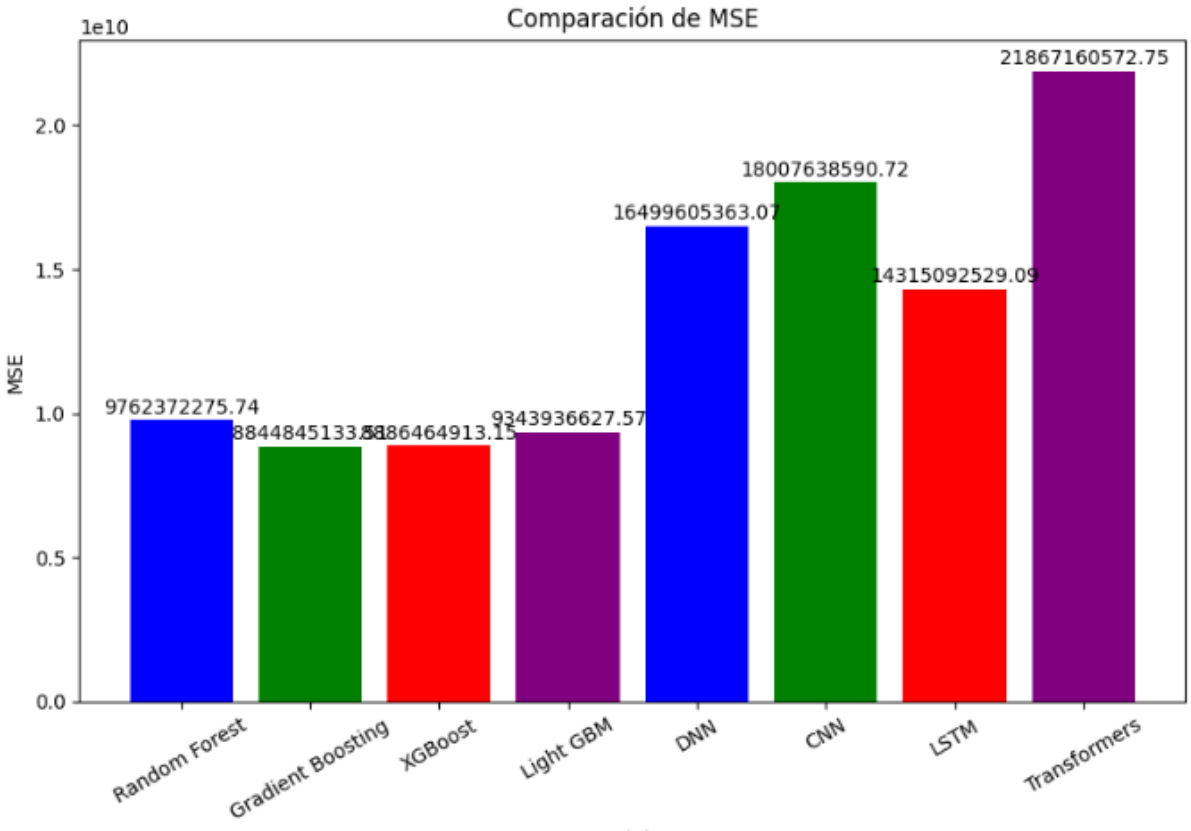
Esta es la gráfica para el modelo Transformer. Aunque el entrenamiento no parece haber sido tan eficiente como en el caso anterior ya que la línea de validación fluctua bastante, en este caso podemos ver como el callback ReduceOnPlateau se ha aplicado hasta dos veces con éxito, a las 1287 y las 3349 epochs, mejorando el aprendizaje como se observa.

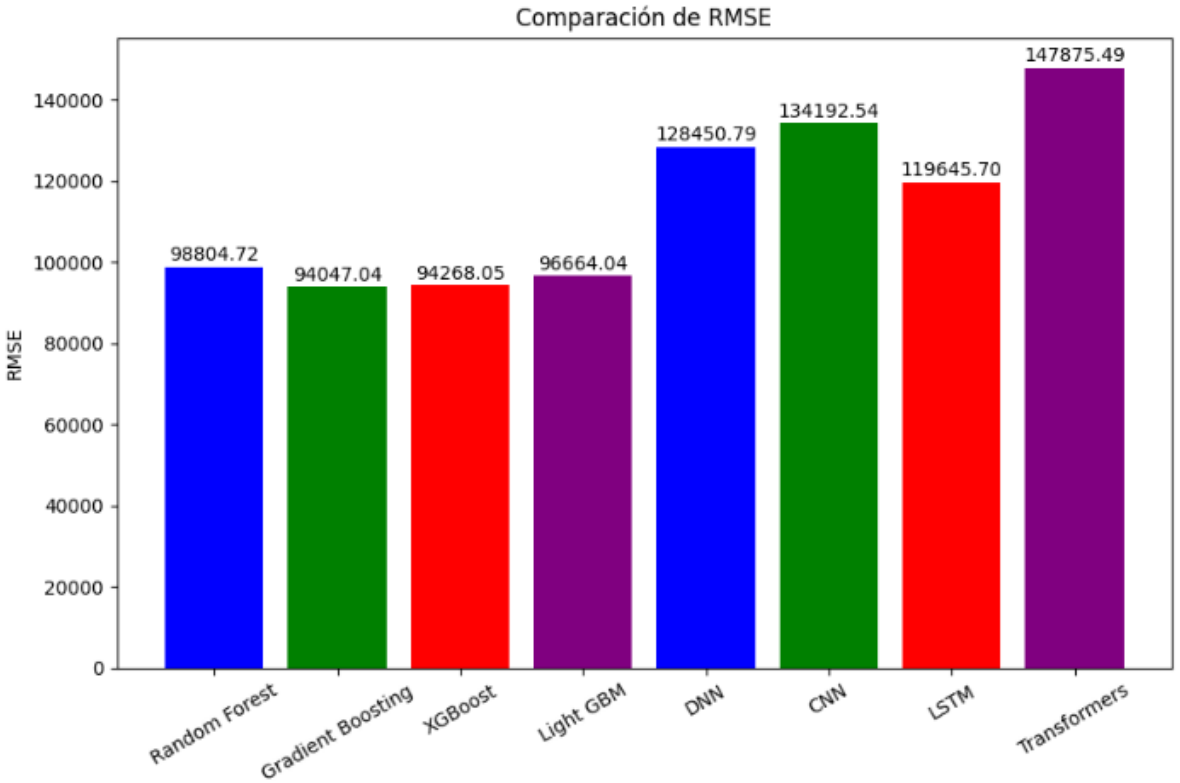
Los resultados en estos últimos modelos son coherentes, ya que son arquitecturas muy potentes diseñadas para conjuntos de datos mucho más masivos que el nuestro. Y al entrenar con datos no masivos, tienen una gran tendencia al sobreajuste.

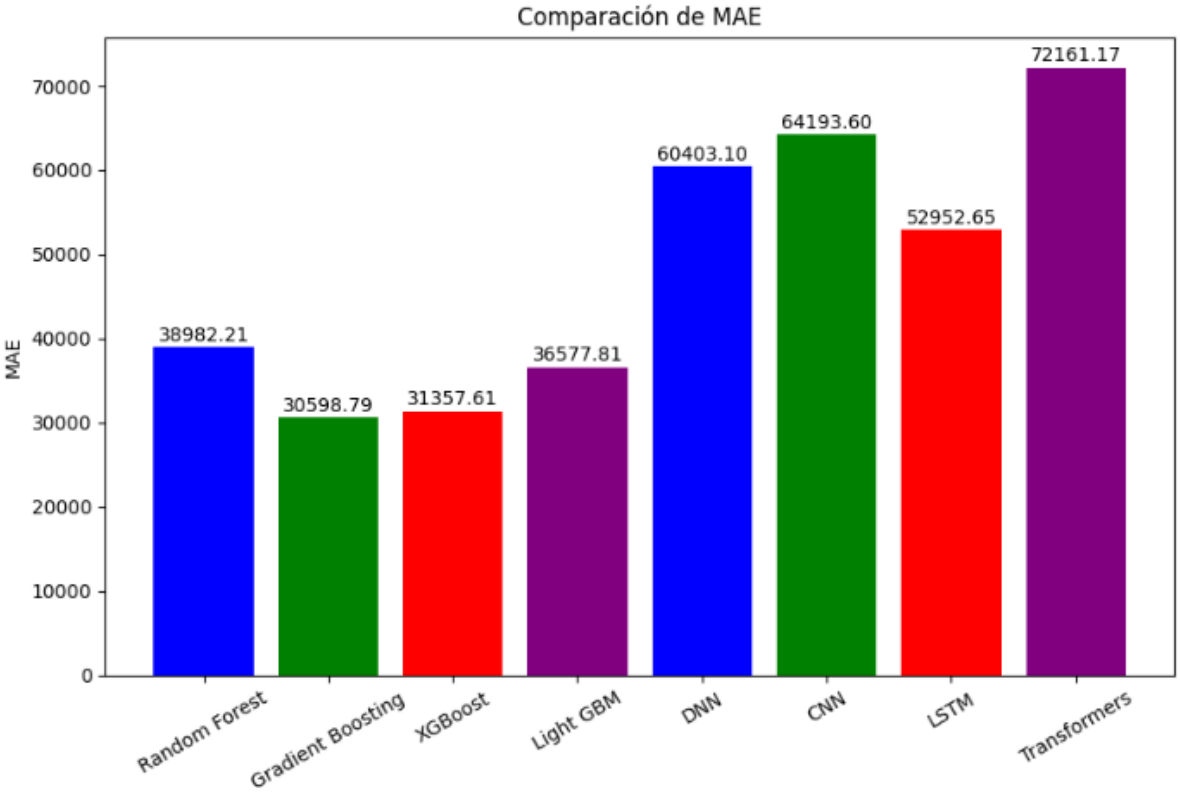
Resultados finales:

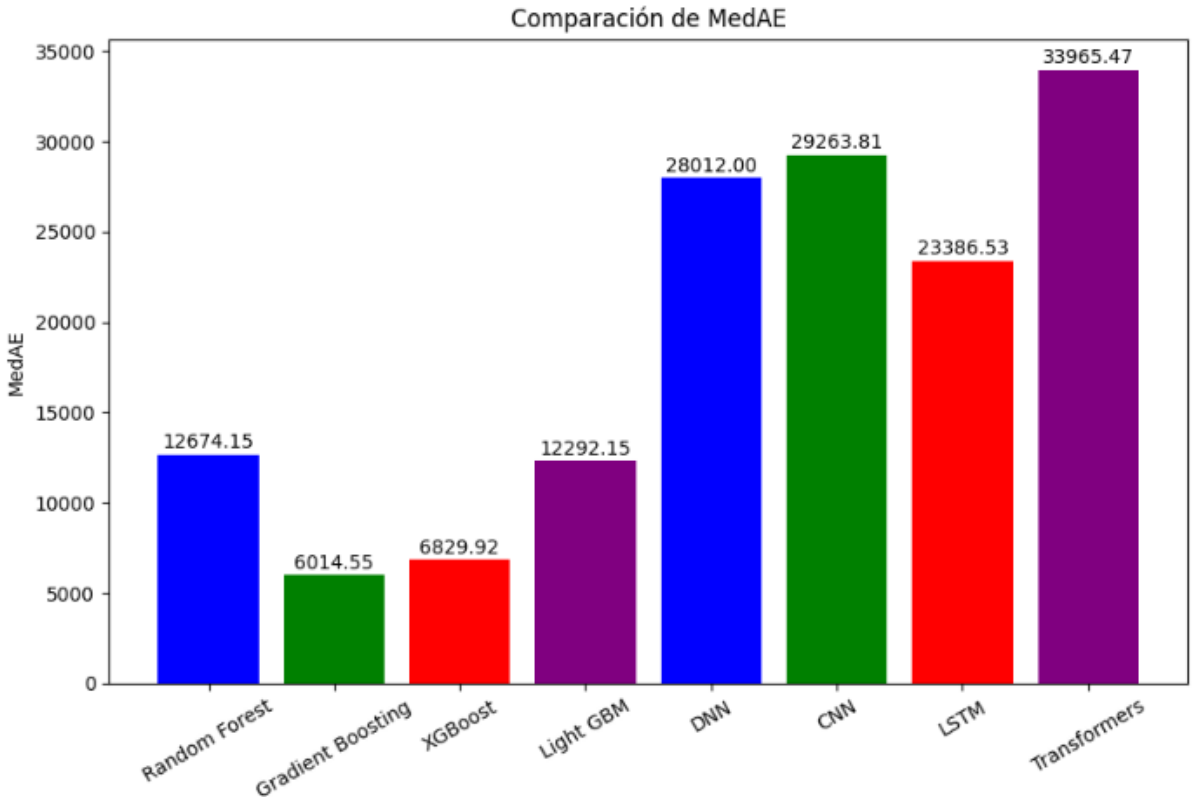
Más allá de lo visto en las gráficas anteriores, se ha aplicado en el entrenamiento el callback **ModelCheckpoint**, que guarda el mejor modelo obtenido durante el entrenamiento para el conjunto de validación. Por lo que independientemente de que el entrenamiento haya sido eficiente o no, es necesario evaluar los resultados sobre el conjunto de test a partir de las métricas definidas anteriormente.

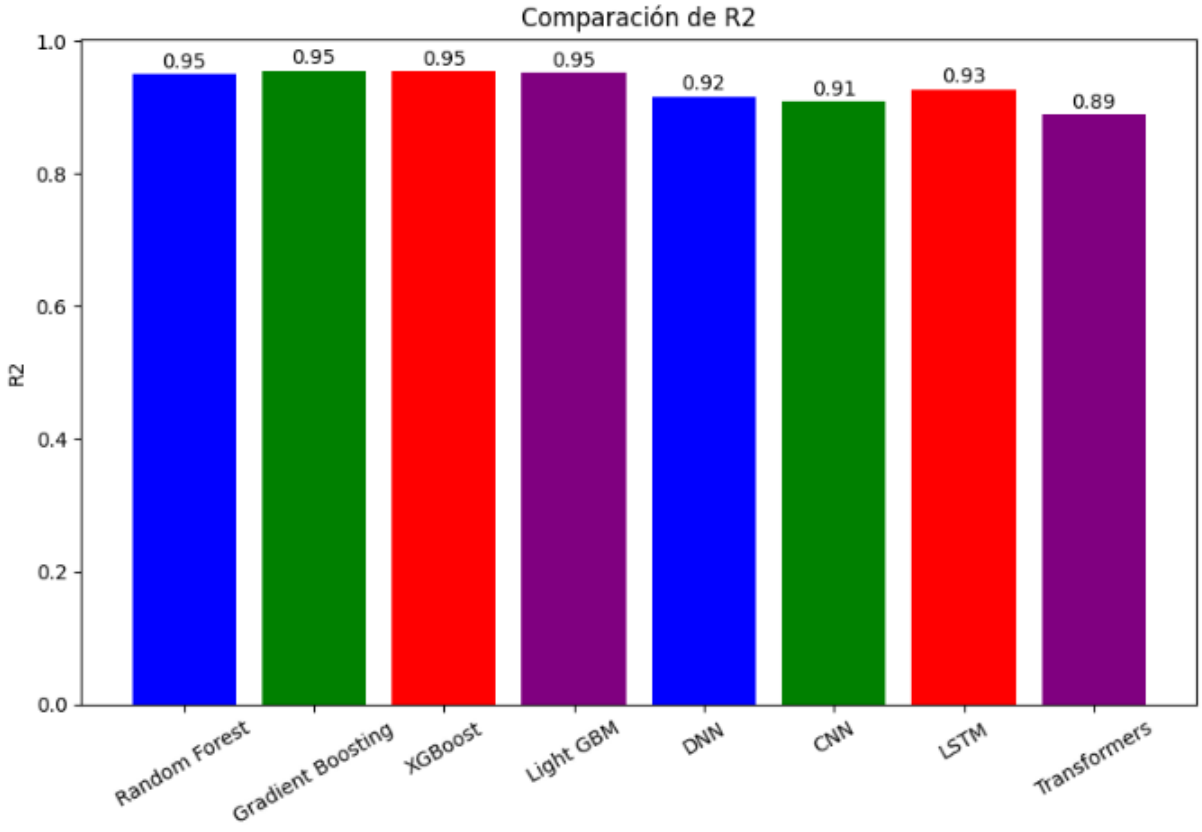
En esta presentación de los resultados también se visualizarán los obtenidos por los árboles de regresión, para tener una visión global de cuáles han resultado ser nuestros mejores modelos en el conjunto.

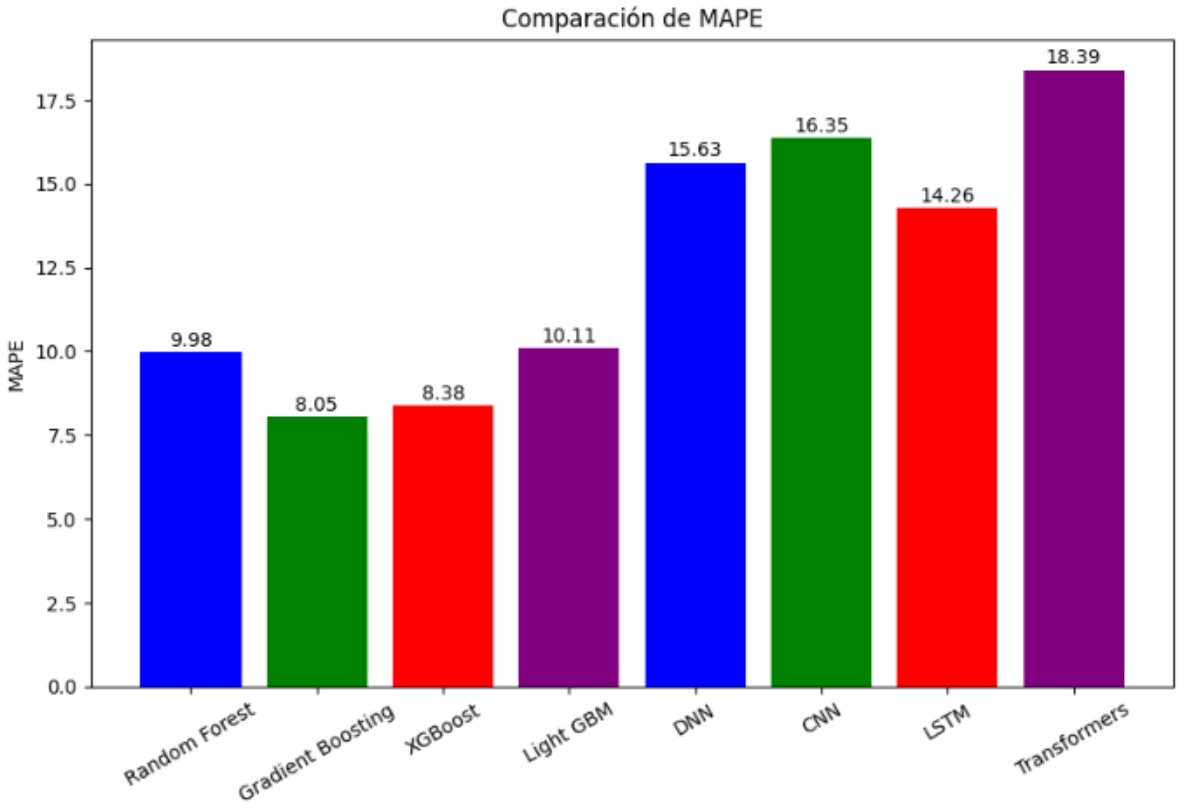


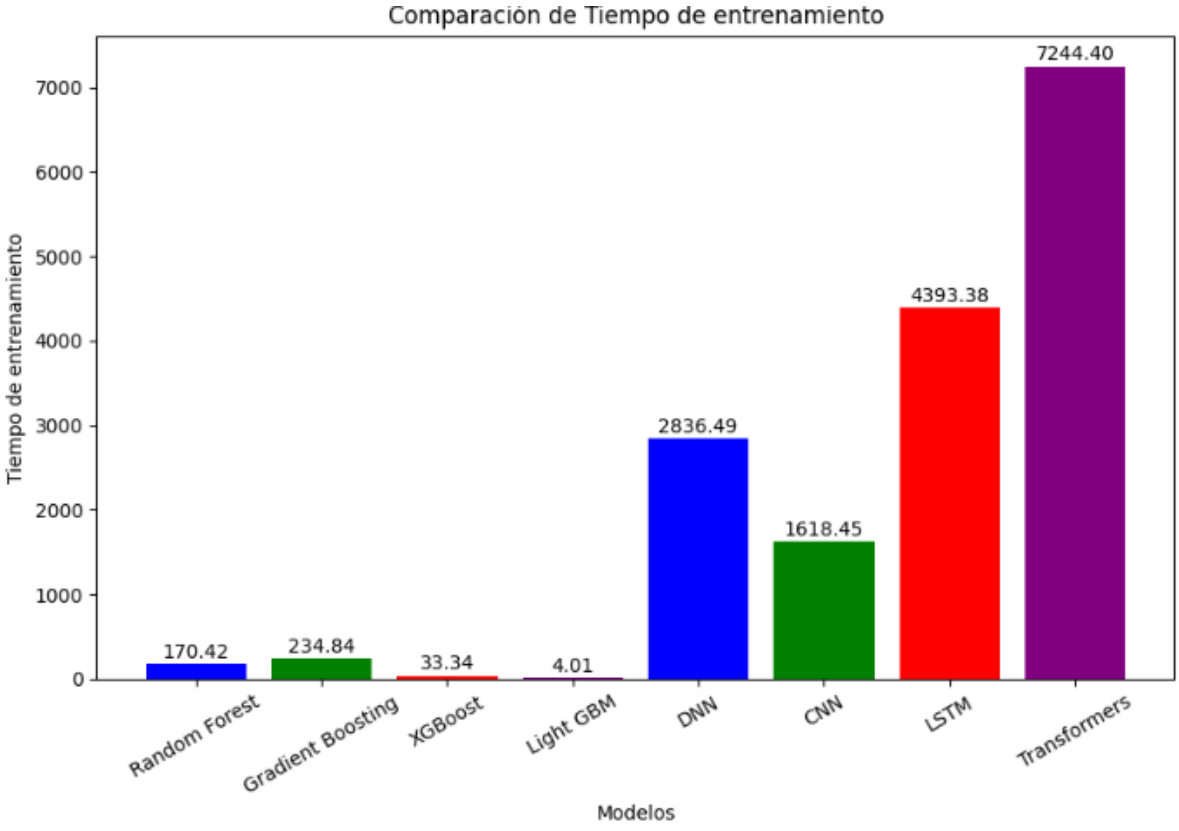

Comparación de RMSE
RMSE
0
20000
40000
60000
80000
100000
120000
140000
98804.72
94047.04
94268.05
96664.04
128450.79
134192.54
119645.70
147875.49
Random Forest
Gradient Boosting
XGBoost
Light GBM
DNN
CNN
LSTM
Transformers


Comparación de MAE
MAE
0
10000
20000
30000
40000
50000
60000
70000
38982.21
30598.79
31357.61
36577.81
60403.10
64193.60
52952.65
72161.17
Random Forest
Gradient Boosting
XGBoost
Light GBM
DNN
CNN
LSTM
Transformers


Comparación de MedAE
MedAE
35000
30000
25000
20000
15000
10000
5000
0
12674.15
6014.55
6829.92
12292.15
28012.00
29263.81
23386.53
33965.47
Random Forest
Gradient Boosting
XGBoost
Light GBM
DNN
CNN
LSTM
Transformers


Comparación de R2
R2
1.0
0.8
0.6
0.4
0.2
0.0
0.95
0.95
0.95
0.95
0.92
0.91
0.93
0.89
Random Forest
Gradient Boosting
XGBoost
Light GBM
DNN
CNN
LSTM
Transformers


Comparación de MAPE
MAPE
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
9.98
8.05
8.38
10.11
15.63
16.35
14.26
18.39
Random Forest
Gradient Boosting
XGBoost
Light GBM
DNN
CNN
LSTM
Transformers


Comparación de Tiempo de entrenamiento
Tiempo de entrenamiento
0
1000
2000
3000
4000
5000
6000
7000
170.42
234.84
33.34
4.01
2836.49
1618.45
4393.38
7244.40
Random Forest
Gradient Boosting
XGBoost
Light GBM
DNN
CNN
LSTM
Transformers
Modelos

	MSE	RMSE	MAE	MedAE	R2	MAPE	Tiempo de entrenamiento
Random Forest	9.762372e+09	98804.717882	38982.208094	12674.150000	0.950108	9.979983	170.424947
Gradient Boosting	8.844845e+09	94047.036814	30598.789779	6014.546247	0.954798	8.047522	234.838128
XGBoost	8.886465e+09	94268.048209	31357.605420	6829.921875	0.954585	8.379833	33.343387
Light GBM	9.343937e+09	96664.039992	36577.809422	12292.148186	0.952247	10.105932	4.006396
DNN	1.649961e+10	128450.789655	60403.098841	28012.000000	0.915677	15.626751	2836.493649
CNN	1.800764e+10	134192.542977	64193.602099	29263.812500	0.907970	16.354499	1618.452129
LSTM	1.431509e+10	119645.695824	52952.651148	23386.531250	0.926841	14.258871	4393.382445
Transformers	2.186716e+10	147875.490101	72161.170072	33965.468750	0.888246	18.386427	7244.399194

Figura 14: Métricas modelos definitivos tras optimización bayesiana.

Como se aprecia a simple vista, el rendimiento de los modelos basados en árboles de regresión ha sido notablemente mejor que el obtenido mediante redes neuronales. Esto se puede explicar teniendo en cuenta lo comentado anteriormente, y es que este tipo de modelos está diseñado para conjuntos de datos masivos, lo cual a pesar de tener un tamaño considerable no es el caso de nuestro dataset.

En general todos los modelos han tendido ligeramente al sobreajuste ya que se han entrenado con una gran cantidad de epochs, sin embargo esto era necesario para obtener los mejores resultados posibles. En los modelos basados en árboles se ha conseguido controlar mejor el sobreajuste mediante técnicas de poda, mientras que la regulación L1-L2 y el Dropout no siempre han funcionado de manera óptima.

En particular, podemos destacar el demepeño de la red **LSTM como el mejor modelo construido con Deep Learning**. Aunque inicialmente el modelo con más potencial parecía ser el DNN, este ha tenido demasiado overfiting al realizar un entrenamiento más exhaustivo, lo que le ha relegado al segundo lugar. Para el modelo LSTM la mediana de los errores se sitúa en 23386.53€ y el porcentaje medio de error en un 18.39%, los cuales son datos relativamente buenos que nos indican que este modelo es perfectamente utilizable para obtener predicciones relativamente precisas, aunque probablemente peores que las que obtendremos con los basados en árboles.

Por último, otro claro indicador de que nuestros datos se ajustan mucho mejor a los árboles de regresión, es el indicador R^2. Que para todos los árboles de regresión estará por encima del 95% mientras que para las redes neuronales variará entre el 88% y el 92%.

Con esto podemos concluir que **los dos mejores modelos obtenidos son el basado en Gradient Boosting tradicional y el XGBoost**.

5. Conclusiones y trabajos futuros

Análisis de la metodología seguida:

Con respecto a la metodología planteada inicialmente, se ha conseguido seguir con éxito, siendo capaces en primer lugar de plantear correctamente un objetivo claro en base a nuestros datos y llevar a cabo los pasos necesarios para alcanzarlo.

Sin embargo, a lo largo del desarrollo práctico del trabajo han ido surgiendo múltiples inconvenientes que han hecho difícil que se cumpliesen los plazos temporales establecidos. En primer lugar, el planteamiento inicial era contar con datos que hicieran referencia a la situación diaria del mercado. La idea era contar con un volumen mucho más masivo de datos, colapsarlos a nivel de día quedándonos con los valores medios de las variables más importantes, así como de la variable objetivo Precio para cada día, y con esta serie temporal desarrollar una serie de modelos de forecasting que nos permitiesen predecir la evolución del mercado inmobiliario.
Esto no fue posible debido a que las webs que ofertan viviendas en venta han actualizado sus robots.txt, prohibiendo todo tipo de Web Scraping y haciendo imposible obtener los datos requeridos de manera legítima.

Una vez ajustado el objetivo a predecir el valor de la vivienda en base a sus características, han ido apareciendo diversos problemas técnicos. En primer lugar, todos los modelos han sido entrenados de manera local, por lo que han requerido de la construcción de entornos optimizados donde poder llevar a cabo ejecuciones utilizando la GPU*. Aparte de esto, el entrenamiento de los modelos ha sido en general bastante insatisfactorio en primeras aproximaciones, lo que nos ha forzado a aplicar diversas técnicas como la modificación de funciones objetivo o el uso de validación cruzada y optimización bayesiana en la búsqueda de hiperparámetros. También, al tener un conjunto grande pero no masivo, el sobreajuste ha resultado un gran problema con el que lidiar, sobre todo en los modelos de aprendizaje profundo.

Finalmente, errores no identificados en la limpieza del dato nos llevaron a reejecutar el entrenamiento de los modelos al completo varias veces, lo que de nuevo se tradujo en varios días de ejecución que hicieron imposible que nos ajustásemos a los tiempos establecidos en la metodología.

Reflexión sobre resultados e impacto:

Sobre los rusultados obtenidos, podemos concluir que en general han sido buenos. El modelo seleccionado como el mejor ha sido el basado en **Gradient Boosting**. Podemos concluir que el modelo ha estimado de forma correcta el precio de una vivienda cualquiera en la Comunidad de Madrid con un margen de error asumible, al menos para una primera estimación.

Con respecto a los modelos en sí mismos, han sido guardados publicados en la plataforma *github* de forma pública, de manera que cualquiera puede acceder a estos y usarlos de manera fácil y gratuita para llevar a cabo las estimaciones que necesite. Lo único que necesitará será tener un entorno de python configurado como se describe en el propio enlace y sus datos en el formato de columnas que hemos utilizado, lo cual es sencillo teniendo en cuenta que también podemos encontrar aquí el notebook donde se realiza el tratamiento del dato.

Mediante la utilización de estos podemos dar por cumplidos también los objetivos de impacto a nivel social, habiendo creado una alternativa sencilla, precisa y gratuita a las herramientas existentes de tasación.

Futuros pasos:

Podría desarrollarse el front de una página web donde se utilizase el modelo obtenido para estimar el precio en base a unas características que el propio usuario proporcionase en la web, haciendo así mucho más fácil y accesible su uso.

Aparte de esto, como se comentaba al principio de este apartado, otro enfoque muy interesante del análisis del mercado inmobiliario podría haber sido el llevar a cabo un análisis predictivo de la situación global de este, lo cual podría ser el complemento perfecto a los modelos obtenidos en este trabajo. Contar con un estimador de precios que nos permita saber si el mercado se ajusta en general a sus características o está hinchado, así como tener estos segundos modelos que nos permitan prever la evolución que tendrá en el futuro, podría ser la combinación perfecta para lograr la información requerida al completo a la hora de intentar comprar o vender una vivienda.

6. Glosario

DL: Deep Learning o aprendizaje profundo, hace referencia a las técnicas de machine learning basadas en redes neuronales, las cuales están diseñadas para trabajar con volúmenes masivos de datos.

Fine Tunning: Búsqueda de los mejores hiperparámetros obtenibles para un modelo de aprendizaje automático.

Euribor: El Euribor (Euro Interbank Offered Rate) es una tasa de interés de referencia diaria promediada a partir de las tasas a las que los bancos en la zona euro se prestan dinero entre sí. Es un indicador importante del costo del crédito en la economía y se usa comúnmente para fijar las tasas de interés de préstamos, hipotecas y otros productos financieros.

Tipo de Interés: El tipo de interés se refiere a la tasa que los bancos centrales, como el Banco Central Europeo o la Reserva Federal en EE.UU., fijan para prestar dinero a los bancos comerciales. Esta tasa es fundamental para controlar la política monetaria y afecta las tasas de interés para préstamos, ahorros y otros productos financieros. Puede influir en la economía alentando o desalentando el gasto y la inversión.

IPC: El IPC (Índice de Precios al Consumidor) es una medida que examina el promedio ponderado de los precios de un conjunto de bienes y servicios de consumo, como transporte, alimentos y atención médica. Se utiliza como indicador de la inflación, reflejando cómo cambian los precios de los bienes y servicios de consumo con el tiempo y afectando así el poder adquisitivo de los consumidores.

GPU: La Unidad de Procesamiento Gráfico (Graphic Processing Units) en un ordenador nos permite ejecutar procesos con un alto nivel de paralelización. Funciona como como un conjunto de microprocesadores con una RAM independiente y compartida que pueden ejecutar procesos de forma concurrente.

7. Bibliografía

[1] Anaïs López, Ramón Torné, Fanny Merino, Paula Iglesias. Nota de prensa 2022: Precio vivienda en venta, Fotocasa, España. 2022
Online: https://s36360.pcdn.co/wp-content/uploads/2023/01/NdP-Espana-VENTA-diciembre-2022.pdf

[2] Choujun Zhan, Yonglin Liu, Wangling Chen, Zeqiong Wu, Zefeng Xie. Housing prices prediction with deep learning: an application for the real estate market in Taiwan, Nanfang College of Sun Yat-sen University Guangdong 510970, China. 2020
Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9442244

[3] Hossam H. Mohamed, Ahmed H. Ibrahim, Omar A. Hagras. Forecasting the Real Estate Housing Prices Using a Novel Deep Learning Machine Model, Civil Engineering Journal (E-ISSN: 2476-3055; ISSN: 2676-6957) Vol. 9, Special Issue. 2023
Online: https://www.civilejournal.org/index.php/cej/article/view/3947/pdf

[4] Alejandro Baldominos, Iván Blanco, Antonio José Moreno, Rubén Iturrarte, Óscar Bernárdez and Carlos Afonso. Identifying Real Estate Opportunities Using Machine Learning, Universidad Carlos III de Madrid. 2018
Online: https://www.mdpi.com/2076-3417/8/11/2321

[5] [TFM] Isaak Ake, Shadi Eltanani. Combining Machine Learning models to predict House Prices. Southampton Solent University FACULTY OF Business, Law, and Digital Technologies. 2022
Online: https://www.solent.ac.uk/documents/degree-shows/isaac-ake-project-scaids.pdf

[6] [TFM] Teresa Álvarez Martín, Juana María Alonso Revenga. ANÁLISIS Y PREDICCIÓN DEL MERCADO INMOBILIARIO EN LA COMUNIDAD DE MADRID. Facultad de estudios estadísticos, Universidad Complutense de Madrid. 2018
Online: https://docta.ucm.es/rest/api/core/bitstreams/a2ccacbf-580f-4340-ab09-fd57bd66e868/content

[7] [TFM] John van den Hurk, H.A.M.Daniels. Transfer Learning For Price Prediction In Real Estate. Tilburg School of Economics and Management Tilburg University. 2021
Online: https://arno.uvt.nl/show.cgi?fid=157041

Recursos:

[8] [ECB] European Central Bank Data Portal. Monthly Euribor data in 2021-2022.
Online:
https://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=143.FM.M.U2.EUR.RT.MM.EURIBOR1YD_.HSTA

[9] [INE] Instituto Nacional de Estadística. Monthly mortgage and interest rate data. Online: https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736170236&menu=ultiDatos&idp=1254735576757

[10] [INE] Instituto Nacional de Estadística. Monthly CPI data in 2021 - 2022. Online: https://www.ine.es/jaxiT3/Datos.htm?t=25333

[11] [INE] Instituto Nacional de Estadística. Monthly unemployment data in 2021 - 2022. Online: https://www.ine.es/jaxiT3/Tabla.htm?t=4247&L=0

[12] Personal Github repository with all code and resources used in development. Online: https://github.com/jmguerreroca/Prediccion_del_mercado_inmobiliario