TÉCNICAS DE MACHINE LEARNING E INTERPRETABILIDAD APLICADAS AL MERCADO INMOBILIARIO

MACHINE LEARNING AND INTERPRETABILITY TECHNIQUES APPLIED TO THE REAL ESTATE MARKET

TRABAJO FIN DE GRADO
CURSO 2022-2023

AUTOR
PABLO LÓPEZ MARTÍN

DIRECTOR
GABRIEL MARÍN DÍAZ

GRADO EN INGENIERÍA DEL SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

TÉCNICAS DE MACHINE LEARNING E INTERPRETABILIDAD APLICADAS AL MERCADO INMOBILIARIO

MACHINE LEARNING AND INTERPRETABILITY TECHNIQUES APPLIED TO THE REAL ESTATE MARKET

TRABAJO DE FIN DE GRADO EN INGENIERÍA DEL SOFTWARE
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

AUTOR
PABLO LÓPEZ MARTÍN

DIRECTOR
GABRIEL MARÍN DÍAZ

CONVOCATORIA: FEBRERO - 2023

GRADO EN INGENIERÍA DEL SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

10 DE ENERO DE 2022

DEDICATORIA

A mis padres, por enseñarme que vale
más la constancia que la virtud.

AUTORIZACIÓN DE DIFUSIÓN Y UTILIZACIÓN

El abajo firmante, matriculado en el Grado en Ingeniería del Software de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "Técnicas de Machine Learning e interpretabilidad aplicadas al mercado inmobiliario", realizado durante el curso académico 22/23 bajo la dirección de Gabriel Marín Díaz en el Departamento de Sistemas Informáticos y Computación, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Pablo López Martín

10 de enero de 2022

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a Gabriel Marín Díaz, director de este TFG, todo el apoyo ofrecido durante el desarrollo de este proyecto, así como todos los consejos brindados.

En segundo lugar, me gustaría agradecer a la facultad de informática todo el conocimiento que me ha aportado durante estos 4 años, así como también le estaré eternamente agradecido por despertar en mí unas ganas de aprender casi infinitas.

Por último, me gustaría agradecer a mi familia el apoyo ofrecido durante mi desarrollo académico.

RESUMEN

El mercado inmobiliario en España tiene una gran relevancia en la población. Los hogares españoles tienen depositados gran de parte de sus ahorros en activos inmobiliarios, siendo bastante relevante el porcentaje de estos dentro del patrimonio total de las familias.

Este trabajo es un acercamiento al mercado inmobiliario desde el punto de vista de la inteligencia artificial y el *Machine Learning*. El objetivo de este trabajo fue entender la evolución de los precios de la vivienda respecto a factores externos como pueden ser las variables demográficas y de esta manera, poder predecirlos.

Inicialmente se planteó obtener la información mediante *web scraping* de los de diversos portales inmobiliarios líderes en España, pero debido a las posibles implicaciones legales del uso de esta técnica, finalmente se optó por la información disponible en la web del Instituto Nacional de Estadística.

Tras la aplicación de modelos de *Machine Learning* a los datos disponibles, se consiguió demostrar que, con una elección de variables relevantes, es factible predecir la evolución de los precios de la vivienda en el tiempo en una comunidad autónoma.

Palabras clave

Machine Learning, Análisis exploratorio de datos, Inteligencia artificial explicable (XAI), Interpretabilidad, Aprendizaje Supervisado, *Machine Learning interpretable*.

ABSTRACT

The real estate market in Spain has a great relevance in the population. Spanish households have deposited a large part of their savings in real estate assets, the percentage of these within the total wealth of families is quite significant.

This work is an approach to the real estate market from the point of view of artificial intelligence and Machine Learning. The objective was to understand the evolution of housing prices with respect to external factors such as demographic variables and in this way, to be able to predict them.

Initially, it was proposed to obtain the information through web scraping from various leading real estate portals in Spain, but due to the possible legal implications of using this technique, the information available on the website of the National Institute of Statistics was finally chosen.

After applying Machine Learning models to the available data, it was possible to demonstrate that, with a choice of relevant variables, it is feasible to predict the evolution of housing prices over time in an autonomous community.

Keywords

Machine Learning, Exploratory data analysis, Explainable artificial intelligence, Interpretability, Supervised learning, Interpretable Machine Learning.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1 - Introducción

1.1 Motivación

Los hogares españoles mantienen una inversión media en activos inmobiliarios equivalente a 236.100 euros, frente a la inversión media que mantienen en activos financieros, situada en 56.300 euros [1]. Además, en una sociedad donde la educación financiera se encuentra por debajo de la media de la unión europea [2], cabe destacar la relevancia de proporcionar a la población herramientas analíticas que puedan ser útiles para la toma de decisiones en lo que a inversión inmobiliaria se refiere.

1.2 Objetivos

Este proyecto busca proporcionar un método de trabajo basado en técnicas de *Machine Learning,* aplicable a cualquier *dataSet* futuro relacionado con el mercado inmobiliario. De esta manera, el inversor tendrá un procedimiento analítico con el que obtener información útil y valiosa, además de ser una herramienta de apoyo para la toma de sus futuras decisiones de inversión.

1.3 Plan de trabajo

Para poder asegurar la consecución del proyecto, se realizó una planificación a alto nivel al comienzo del proyecto. La figura 1-1 detalla todas las tareas realizadas, así como las fechas en las que se ejecutó cada una de ellas.

Tarea	Fecha inicio	Fecha final	Duración (días)
TFG	05/09/2022	05/01/2023	122
Capitulo 2. Estado del arte	05/09/2022	06/10/2022	31
Investigación literatura actual sobre el aprendizaje automático aplicado al mercado inmobiliario	05/09/2022	10/09/2022	5
Memoria	10/09/2022	15/09/2022	5
Investigación bases de datos del mercado inmobiliario español	15/09/2022	21/09/2022	6
Aprendizaje python y sus librerías	21/09/2022	01/10/2022	10
Estudio del aprendizaje automático	01/10/2022	06/10/2022	5
Capitulo 3. Recolección y preparación de los datos	06/10/2022	10/10/2022	4
Memoria	06/10/2022	10/10/2022	4
Capitulo 4. Análisis exploratorio de datos	10/10/2022	21/10/2022	11
Estudio y análisis de la base de datos final	10/10/2022	13/10/2022	3
Selección de características relevantes de la base de datos	13/10/2022	16/10/2022	3
Memoria	16/10/2022	21/10/2022	5
Capítulo 5. Modelado	21/10/2022	15/11/2022	25
Entrenamientos de modelos y obtención de métricas	21/10/2022	31/10/2022	10
Memoria	31/10/2022	05/11/2022	5
Análisis de resultados y conclusiones	05/11/2022	10/11/2022	5
Memoria	10/11/2022	15/11/2022	5
Capitulo 6. Interpretabilidad - XAI	15/11/2022	05/01/2023	51
Aprendizaje interpretabilidad	15/11/2022	25/11/2022	10
Aprendizaje librería SHAP python	25/11/2022	30/11/2022	5
Aplicación de la interpretabilidad al modelo	30/11/2022	16/12/2022	16
Memoria	16/12/2022	05/01/2023	20
Entrega borrador	09/01/2022		
Entrega final	10/01/2022		

Figura 1-1. Plan de trabajo

1.4 Repositorio

Todo el código y la documentación generada durante el desarrollo del proyecto se encuentra compartida en el repositorio público de GitHub que es accesible a través del siguiente enlace bajo la licencia Creative Commons Atribución-NoComercial 4.0 Internacional (CC BY-NC 4.0): https://github.com/Plopezq/TFG-entrega

Capítulo 2 - Estado del arte

En este capítulo analizaré los aspectos más importantes, tanto en el ámbito académico como en el tecnológico, relacionados con el tema del proyecto.

Comenzando con las tendencias y trabajos actuales, así como con el análisis de las fuentes de datos disponibles sobre el mercado inmobiliario español.

Continuando con la exposición de las razones de utilizar una fuente u otra y las diferencias que este trabajo aporta respecto a la literatura existente, así como con la explicación de alguna de las técnicas de *Machine Learning* utilizadas durante el desarrollo del proyecto.

Finalmente, argumentaré porqué Python ha sido el lenguaje elegido para el desarrollo del proyecto, además de una breve introducción a las librerías utilizadas y que han permitido desarrollar este proyecto.

2.1 Tendencias y trabajos actuales

El mercado inmobiliario representa gran parte del activo de las personas físicas, tanto en España como en el resto de los países del mundo [3]. Es por ello, que hay bastante literatura de calidad.

En la figura 2-1 se puede observar el gran crecimiento que ha tenido en los últimos años el número de publicaciones en torno a la temática de aplicar técnicas de *Machine Learning* a la predicción del precio de la vivienda. Así mismo, en la figura 2-2 se pueden observar las áreas de estudio que más han analizado este tema, siendo las ciencias de la computación las que más han trabajado en ello.

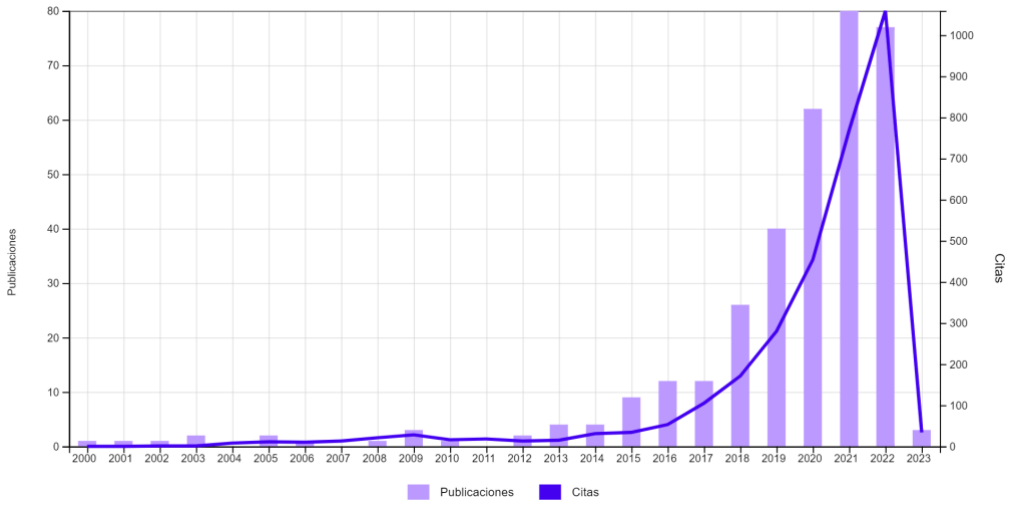

Figura 2-1. Número de publicaciones a lo largo del tiempo.

Fuente: www.webofscience.com

TS = ((MACHINE LEARNING) AND (PRICE HOUSES))

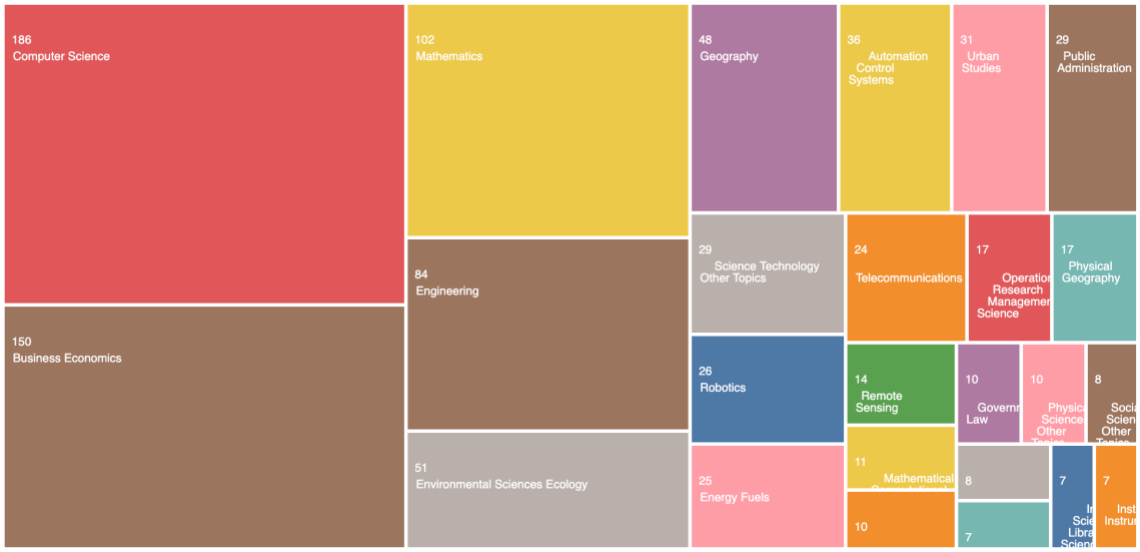

Figura 2-2. Áreas de estudio.

Fuente: www.webofscience.com

TS = ((MACHINE LEARNING) AND (PRICE HOUSES))

Ref.	Modelos utilizados
Hasan Selim 2009 [4]	Regresión hedónica Redes neuronales artificiales
Byeonghwa Park y otros 2014 [5]	C4.5 Naive Bayes Ada boost
Jieh-Haur Chen y otros 2017 [6]	SVM
Chun Haejung 2020 [7]	Series de datos temporales Redes neuronales artificiales

Tabla 2-1. Publicaciones TS = ((MACHINE LEARNING) AND (PRICE HOUSES))

Respecto a la literatura existente, cabe destacar varios artículos que han sido de gran ayuda para el desarrollo de este proyecto.

- En el primer artículo a destacar, escrito por Hasan Selim en 2009 [4], se aborda el estudio de varios modelos para la predicción del precio de las viviendas en Turquía. El primer modelo consistía en aplicar regresión hedónica y el otro en la utilización de redes neuronales artificiales (ANN). El autor llega a la conclusión de que con las redes neuronales artificiales se obtenía una mayor precisión que con la regresión hedónica.
- En el segundo artículo escrito por Byeonghwa Park y Jae Kwon Bae en 2014 [5], se utilizan algoritmos de *Machine Learning* para predecir el precio de las viviendas en el Condado de Fairfax, Virginia. Algunos de los algoritmos aplicados son C4.5, Naive Bayes y Ada Boost. Cabe destacar que la metodología utilizada en este artículo para la predicción del precio de la vivienda, así como para la evaluación de los distintos algoritmos, ha servido de base para el desarrollo de este proyecto.

- En el tercer artículo escrito por Jieh-Haur Chen, Chuan Fan Ong, Linzi Zheng y Shu-Chien Hsu en 2017 [6], se intenta predecir el precio de las viviendas en la ciudad de Taipei, Taiwan, utilizando únicamente algoritmos de máquinas de vectores de soporte (SVM).
- En el cuarto artículo, escrito por Chun Haejung en 2020 [7], el autor utiliza modelos de predicción basados en series de datos temporales y también modelos de redes neuronales artificiales (ANN), en especial entrena redes neuronales recurrentes y redes neuronales de corta memoria. Finalmente, el autor compara todos los modelos y llega a la conclusión de que la predicción del precio de las viviendas usando técnicas de *Machine Learning* es mucho más eficaz que usando modelos basados en series de datos temporales.

Como se puede observar, la literatura relacionada con el mercado inmobiliario y el *Machine Learning* ha sido de interés desde hace muchos años. Pese a eso, los modelos de estudio relacionados con el mercado español son escasos. Con este proyecto busco aportar un método de trabajo para el análisis del mercado inmobiliario español mediante algoritmos de *Machine Learning,* así como a la vez, proporcionar una transparencia a las decisiones tomadas por los algoritmos para dicha predicción mediante la aplicación de técnicas de interpretabilidad como son los valores de Shapley [8].

2.2 Bases de datos de viviendas en España

Son muchas las webs disponibles en España especializadas en la compraventa y alquiler de viviendas, en ellas se puede obtener información muy detallada del mercado inmobiliario actual, ya que incluyen información sobre la ubicación, el número de habitaciones, los metros cuadrados, el número de baños y el precio de venta. Las webs de compraventa de viviendas más utilizadas en España son Idealista y Fotocasa [9]. Por todo ello, inicialmente se analizó la posibilidad de obtener la información de alguna de estas webs.

En el caso de Idealista, existe un API disponible para los usuarios que permite acceder a todos los anuncios publicados en España y Portugal y descargarlos en formato JSON. El inconveniente encontrado es que este API solo permite 100 peticiones al mes, las cuales no eran suficientes para el desarrollo de este proyecto.

La segunda web considerada para la obtención de datos fue Fotocasa, que, aunque no disponía de API, se hubieran podido obtener los datos mediante *web scraping,* que es una técnica utilizada para obtener información de sitios web de manera automática [10]. Esta segunda opción fue también descartada debido a las posibles implicaciones legales que tiene la utilización de esta técnica.

Finalmente, tras la imposibilidad de obtener información de los portales lideres en España, opté por obtenerla del Instituto Nacional de Estadística, organismo autónomo español encargado de la coordinación general de los servicios estadísticos de la Administración General del Estado y la vigilancia, control y supervisión de los procedimientos técnicos de los mismos. Actualmente se encuentra adscrito al Ministerio de Asuntos Económicos y Transformación Digital [11]. El INE contenía datos bastante ordenados y aunque no contenía tantas variables, agregando distintas fuentes de datos, una con datos demográficos y otra con el número de compraventas de viviendas, también proporcionados por el INE, se consiguió obtener una buena base de datos.

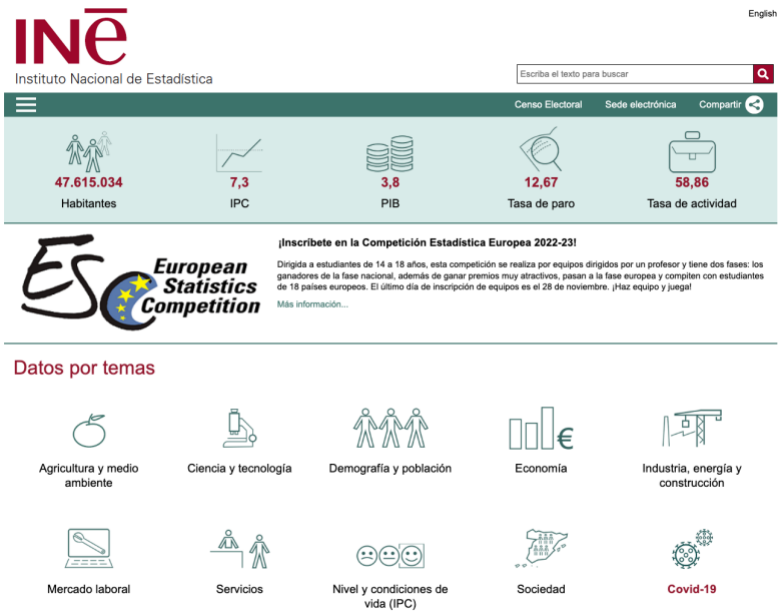

Figura 2-3. Página web del Instituto Nacional de Estadística.

Imagen capturada de https://www.ine.es/index.htm

Se han utilizado 3 bases de datos del INE:

- Índices de precios de vivienda en España [12].

- Población residente en España segmentada por fecha, sexo y edad [13].
- Estadística de Transmisiones de Derechos de la Propiedad en España [14].

2.3 Machine Learning

2.3.1 Introducción

El *Machine Learning* es una rama de la inteligencia artificial (IA) y la informática que se centra en el uso de datos y algoritmos para imitar la forma en la que aprenden los seres humanos [15].

El *Machine Learning* es un componente importante dentro del creciente campo de la ciencia de datos. Mediante la utilización de métodos estadísticos, los algoritmos se entrenan para hacer clasificaciones o predicciones, y descubrir información clave dentro de los proyectos de minería de datos.

Esta información facilita posteriormente la toma de decisiones dentro de los proyectos, lo que afecta directamente en la creación de valor y en la mejora de las métricas más relevantes.

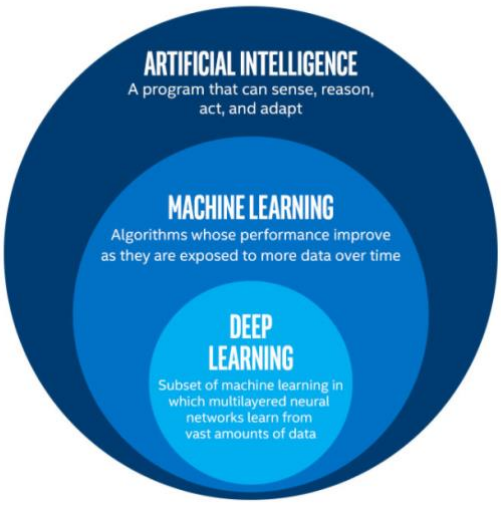

Figura 2-4. Diferencia entre Machine Learning e inteligencia artificial.

Imagen capturada de: https://ai.stackexchange.com/questions/15859/is-machine-learning-required-for-deep-learning

2.3.2 Metodología de trabajo

A la hora de aplicar técnicas de *Machine Learning* en un proyecto, hay una serie de pasos estándar que se deben seguir. En este caso, he elegido utilizar la metodología KDD *(Knowledge Discovery in Databases* [16], basada en descubrir el conocimiento en las bases de datos y CRISP-DM (*Cross Industry Standard Process for Data Mining)* [17] *que* proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos, de forma análoga a como se hace en la ingeniería del software con los modelos del ciclo de vida del desarrollo de software.

El ciclo de vida del proyecto de minería de datos consiste en seis fases, mostradas en la figura siguiente.

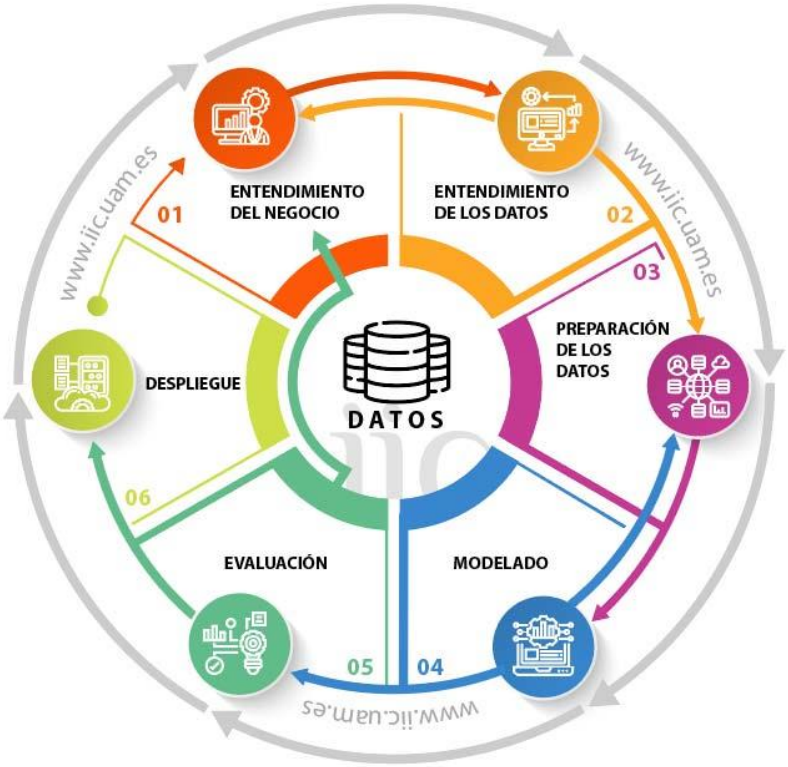

Figura 2-5. Metodología propuesta [18].

Imagen capturada de: www.ii.uam.es

Los proyectos de análisis de datos tienen una naturaleza cíclica. El proyecto no se da por acabado cuando se despliega la solución, sino que, con la información descubierta durante el proceso, se pueden producir nuevas iteraciones del modelo.

El modelo CRISP-DM tiene las siguientes fases:

1. Definición de necesidades del cliente (comprensión del negocio).
2. Estudio y comprensión de los datos.
3. Análisis de los datos y selección de características.
4. Modelado.
5. Evaluación (obtención de resultados).
6. Despliegue (puesta en producción).

La entidad que planteó CRISP-DM se disolvió hace unos años. Pese a ello, CRISP-DM es la metodología que se utiliza de facto, de una forma u otra, en los proyectos de análisis de datos que se pretenden abordar con seriedad y asegurando la calidad de los resultados.

2.3.3 Tipos de Machine Learning

- **Aprendizaje no supervisado.** En algunos problemas de reconocimiento de patrones, los datos de entrada para el entrenamiento contienen características, pero no contienen un valor objetivo ni un resultado esperado. Estos problemas son llamados problemas de aprendizaje no supervisado y algunos de los principales objetivos consisten en descubrir patrones, definir una métrica de similitud o distancia que sirva para comparar los datos entre sí o reducir la dimensionalidad con el objetivo de tener un conjunto de datos menor a la vez que se mantienen las características más relevantes [19].
- **Aprendizaje supervisado.** Los problemas donde los datos usados para el entrenamiento están etiquetados, es decir, se conoce de antemano el valor a predecir para algunas muestras, son conocidos como problemas de aprendizaje supervisado. Los algoritmos de aprendizaje supervisado utilizan los valores conocidos de salida para que después de la fase de entrenamiento, se pueda predecir la salida de nuevas muestras no presentes en el entrenamiento [20].

Dentro del aprendizaje supervisado, cabe hacer la distinción entre clasificación y regresión. En los problemas de clasificación, la salida puede tomar un número de valores discretos, es decir, cada muestra de entrada pertenece a una de las clases de salida disponibles. En cambio, si la salida esperada puede tomar uno o más valores continuos, estamos ante un problema de regresión.

- **Aprendizaje por refuerzo.** Estos algoritmos están muy presentes en los juegos donde hay una recompensa y a su vez hay diferentes tácticas u opciones que elegir en cada situación. El objetivo es maximizar la recompensa y para ello el algoritmo descubre las acciones óptimas a tomar en cada etapa mediante el proceso de prueba y error, en lugar de recibir un conjunto de datos de entrenamiento como si ocurría en el aprendizaje supervisado [21].

En este proyecto, utilizo algoritmos de aprendizaje supervisado con el objetivo de construir modelos matemáticos que describan o expliquen las relaciones que existen entre las variables de entrada [22] y el valor de salida. El motivo de utilizar este tipo de algoritmos es porque la variable a predecir es una variable continúa y además los datos de entrenamiento están categorizados con su valor esperado. Tras la utilización de varios modelos para la predicción, aplicaré técnicas de interpretabilidad con el objetivo de proporcionar transparencia a los resultados obtenidos.

2.3.4 Aprendizaje no supervisado

En esta sección se explican algunos de los algoritmos de aprendizaje no supervisado más utilizados.

2.3.4.1 K-means

K-means es un método de agrupamiento original del procesamiento de señales, cuyo objetivo es dividir n observaciones en k grupos, en el cual cada observación pertenece al grupo cuyo valor medio es el más cercano. La agrupación del conjunto de datos puede visualizarse como una partición del espacio de datos en celdas de Voronoi [23].

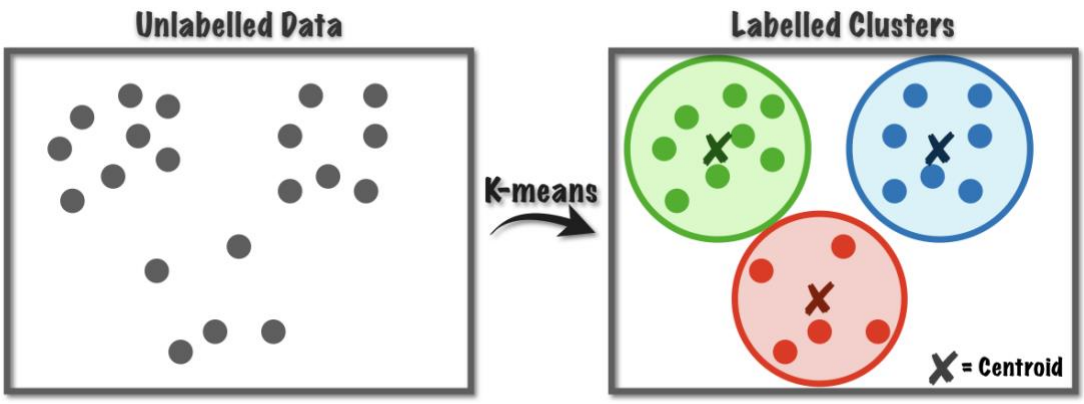

Figura 2-6. Categorización de datos en grupos mediante K-means

Imagen capturada de: https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c

Dado un conjunto de observaciones $(\chi_1, \chi_2, \ldots, \chi_n)$, donde cada observación es un vector real de *d* dimensiones, *k-means* distribuye las observaciones en *k* grupos $(k \leq n)$ a fin de minimizar la suma de los cuadrados dentro de cada grupo $(WCSS)$: $S = S_1, S_2, \ldots, S_k$.

$$\arg_S \min \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

Donde μ_i es la media de puntos en S_i [24].

2.3.4.2 Análisis de componentes principales

El análisis de componentes principales (PCA) es una técnica muy popular para analizar bases de datos grandes que contienen un gran número de características por cada observación, aumentando la interpretabilidad de los datos mientras se preserva la máxima información posible. Anteriormente, el análisis de componentes principales era una técnica estadística para reducir la dimensionalidad de la base de datos [25].

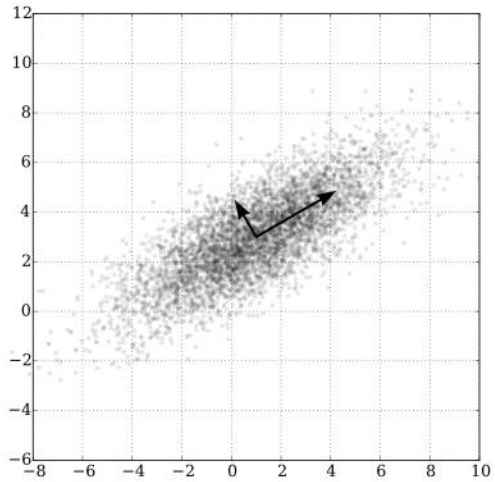

Figura 2-7. PCA de una distribución normal multivariada centrada en (1,3) y con una desviación estándar de 3 en la dirección (0.866, 0.5) y de 1 en la dirección ortogonal.

Imagen capturada de:
https://upload.wikimedia.org/wikipedia/commons/thumb/f/f5/GaussianScatterPCA.svg/450px-GaussianScatterPCA.svg.png

Hay dos formas de aplicar el análisis de componentes principales a una base de datos:

1. Método basado en la matriz de correlación. Se utiliza si los datos no son dimensionalmente homogéneos o cuando el orden de magnitud de las variables aleatorias medidas no es el mismo.

El método parte de la matriz de correlaciones, consideremos el valor de cada una de las m variables aleatorias F_j. Para cada uno de los n individuos tomemos el valor de estas variables y escribamos el conjunto de datos en forma de matriz: $(F_j^{\beta})_{j=1,\dots,m}^{\beta=1,\dots,n}$

Obsérvese que cada conjunto $M_j = \left\{ F_j^{\beta} \middle| \beta = 1, \dots, n \right\}$ puede considerarse una muestra aleatoria para la variable F_j. A partir de los $m \times n$ datos correspondientes a las m variables aleatorias, puede construirse la matriz de correlación muestral, que viene definida por $R = [r_{ij}] \in M_{m\times m}$, donde $r_{ij} = \frac{cov(F_i,F_j)}{\sqrt{var(F_i)var(F_j)}}$.

Puesto que la matriz de correlaciones es simétrica entonces resulta se puede diagonalizar y sus valores propios λ_i verifican: $\sum_{i=1}^{m}\lambda_i = \mathrm{m}$. Debido a la propiedad anterior estos m valores propios reciben el nombre de pesos de cada uno de los componentes principales. Los factores principales identificados matemáticamente se representan por la base de vectores propios de la matriz R. Está claro que cada una de las variables puede ser expresada como combinación lineal de los vectores propios o componentes principales [26].

2. Método basado en la matriz de covarianzas. Se utiliza si los datos son dimensionalmente homogéneos y presentan valores medios similares [26].

El objetivo es transformar un conjunto dado de datos X de dimensión $n \times m$ a otro conjunto de datos Y de menor dimensión $n \times l$ con la menor pérdida de información útil posible utilizando para ello la matriz de covarianza.

Se parte de un conjunto n de muestras cada una de las cuales tiene m variables que las describen y el objetivo es que, cada una de estas muestras, se describa con solo l variables, donde $l < m$. Además, el número de componentes principales l tiene que ser inferior a la menor de las dimensiones de X.

$$l \leq \min\{n, m\}$$

Los datos para el análisis tienen que estar centrados a media 0 (restándoles la media de cada columna) y/o auto escalados (centrados a media 0 y dividiendo cada columna por su desviación estándar).

$$X = \sum_{a=1}^{l} t_a p_a^t + E$$

Los vectores t_a son conocidos como *scores* y contienen la información de cómo las muestras están relacionadas unas con otras, además, tienen la propiedad de ser ortogonales. Los vectores p_a se llaman *loadings* e informan de la relación existente entre las variables y tienen la cualidad de ser ortonormales. Al coger menos componentes principales que variables y debido al error de ajuste del modelo con los datos, se produce un error que se acumula en la matriz E.

El PCA se basa en la descomposición en vectores propios de la matriz de covarianza. La cual se calcula con la siguiente ecuación:

$$cov(X) = \frac{X^T X}{n-1}$$

$$cov(X)p_a = \lambda_a\, p_a$$

$$\sum_{a=1}^{m} \lambda_a = 1$$

Donde λ_a es el valor propio asociado al vector propio p_a. Por último,

$$t_a = X\, p_a$$

Esta ecuación la podemos entender como que t_a son las proyecciones de X en p_a, donde los valores propios de λ_a miden la cantidad de varianza capturada, es decir, la información que representan cada uno de los componentes principales. La cantidad de información que captura cada componente principal va disminuyendo según su número, es decir, el componente principal número uno representa más información que el dos y así sucesivamente [26].

2.3.5 Aprendizaje supervisado

En esta sección se explican algunos de los algoritmos de aprendizaje supervisado más utilizados, algunos de ellos están presentes en este proyecto.

2.3.5.1 Regresión lineal

La regresión lineal es un modelo matemático que se utiliza cuando se quiere aproximar la relación de dependencia entre una variable dependiente Y, m variables independientes $\mathrm{X_i}$ con $m \in \mathbb{Z}^+$ y un término aleatorio ε. Este modelo se puede aplicar a muchas de las situaciones donde se estudia la relación entre 2 o más variables, así como cuando se busca predecir un resultado. En caso de no poder aplicar un modelo de regresión, se dice que no hay correlación entre las variables estudiadas. Este modelo puede ser expresado de la siguiente manera:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m + \varepsilon$$

Donde

- Y es la variable dependiente.
- $\mathrm{X_1}, \mathrm{X_2}$, …, $\mathrm{X_m}$ son las variables independientes.

- $\beta_0,\ \beta_1,\ \beta_2, \ldots, \beta_m$ son los parámetros del modelo, que explican la influencia que tienen las variables independientes sobre el regrediendo.

El término β_0 es la intersección, las $\beta_i (i \geq 1)$ son los parámetros respectivos a cada variable independiente, y m es el número de parámetros independientes que se deben tener en cuenta en la regresión [27].

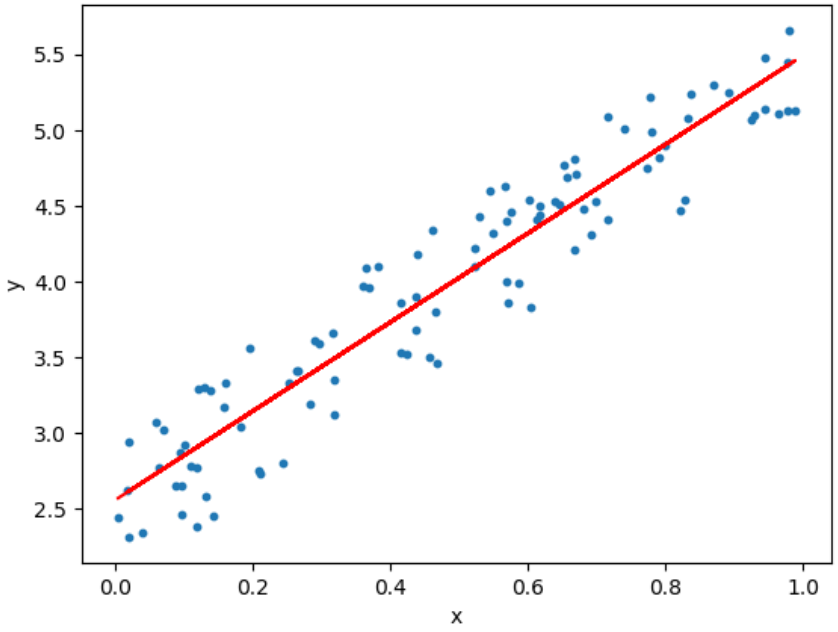

Figura 2-8. Ejemplo de una regresión lineal con una variable dependiente y una independiente.

Imagen capturada de: https://towardsdatascience.com/linear-regression-using-python-b136c91bf0a2

2.3.5.2 Random forest

Los *random forest* son un tipo de algoritmo de clasificación consistentes en varios árboles de decisión. Usan *bagging* y arbitrariedad de características al construir cada árbol individual para intentar crear un bosque de árboles no correlacionados cuya predicción en conjunto sea más precisa que la de un árbol individual [28].

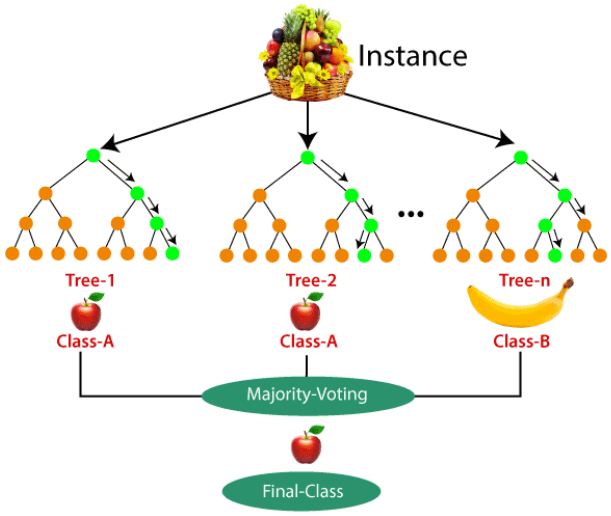

Figura 2-9. Ejemplo de predicción de un árbol de decisión.

Imagen capturada de: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

La idea esencial del *bagging* es promediar muchos modelos ruidosos, pero aproximadamente imparciales, y por tanto reducir la variación. Los árboles son los candidatos ideales para el *bagging*, dado que ellos pueden registrar estructuras de interacción compleja en los datos, y si crecen suficientemente profundo, tienen relativamente baja parcialidad.

Cada árbol es construido usando el siguiente algoritmo:

1. Sea *N* el número de casos de prueba, *M* es el número de variables en el clasificador.
2. Sea *m* el número de variables de entrada a ser usado para determinar la decisión en un nodo dado; *m* debe ser mucho menor que *M*.
3. Elegir un conjunto de entrenamiento para este árbol y usar el resto de los casos de prueba para estimar el error.
4. Para cada nodo del árbol, elegir aleatoriamente *m* variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las *m* variables.

Para la predicción un nuevo caso es empujado hacia abajo por el árbol. Luego se le asigna la etiqueta del nodo terminal donde termina. Este proceso es iterado por

todos los árboles en el ensamblado, y la etiqueta que obtenga la mayor cantidad de incidencias es reportada como la predicción [29].

El rendimiento del algoritmo de los random forest es muy similar a la del boosting, pero al ser más simple de entrenar y ajustar hace que sea muy popular y ampliamente utilizado. Además, este algoritmo es muy eficiente en base de datos grandes además de certero.

2.3.5.3 Support Vector Regression (SVR)

Support Vector Regression o regresión de Soporte Vectorial es una técnica de clasificación de *Machine Learning* que intenta buscar la línea o hiperplano que mejor separa las observaciones de los datos de entrenamiento en varias clases de salida, maximizando las distancias entre observaciones para establecer unas barreras llamadas vectores de soporte o *support vectors* [30]. Su ecuación es similar a la de la Regresión lineal que es $y = wx + b$, pero con la línea recta referida a un hiperplano [31].

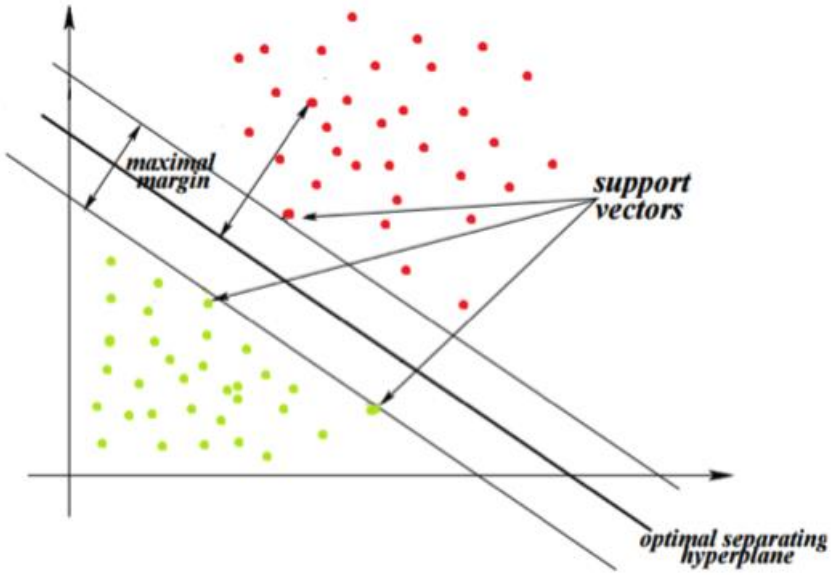

Figura 2-10. Regresión de soporte vectorial.

Imagen capturada de: www.coursera.com

2.3.5.4 XGBoost

XGBoost es una librería de código abierto que proporciona un marco estándar para la aplicación de la técnica de potenciación del gradiente o *Gradient Boosting*.

Esta es una técnica de *Machine Learning* utilizada para el análisis de la regresión y para problemas de clasificación estadística, la cual produce un modelo predictivo en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión. Construye el modelo de forma escalonada como lo hacen otros métodos de *boosting*, y los generaliza permitiendo la optimización arbitraria de una función de pérdida diferenciable [32].

XGBoost funciona como Newton-Raphson en el espacio de funciones, al contrario de *gradient booting* que funciona como una disminución del gradiente en el espacio de funciones. Se usa una aproximación de Taylor de segundo orden en la función de pérdida para hacer la conexión con el método de Newton Raphson.

Un algoritmo XGBoost genérico no regularizado sería [33]:

Entrada: conjunto de entrenamiento $\{(x_i, y_i)\}_{i=1}^{N}$, una función de pérdida diferenciable $L(y, F(x))$, un número de aprendices débiles M y un ratio de aprendizaje α.

Algoritmo:

1. Inicialice el modelo con un valor constante:

$$\hat{f}_o(x) = arg_o min \sum_{i=1}^{N} L(y_i, \theta)$$

2. Para $m = 1$ hasta M

    1. Calcule los '*gradients*' y las '*hessians*':

$$\hat{g}_m(x_i) = [\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f(x)=\hat{f}_{(m-1)(x)}}$$

$$\hat{h}_m(x_i) = [\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}]_{f(x)=\hat{f}_{(m-1)(x)}}$$

2. Ajuste un aprendiz base (o un aprendiz débil, como puede ser un árbol) usando el conjunto de entrenamiento $\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\}_{i=1}^{N}$ resolviendo el siguiente problema de optimización:

$$\widehat{\emptyset}_m = arg_{\emptyset \in \Phi} min \sum_{i=1}^{N} \frac{1}{2} \hat{h}_m(x_i)[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \emptyset(x_i)]^2$$

$$\hat{f}_m(x) = \alpha \widehat{\emptyset}_m(x)$$

3. Actualizar el modelo

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_m(x)$$

4. Salida $\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x)$

2.3.5.5 Artificial neural networks

Las artificial neural networks o redes neuronales artificiales son sistemas de computación inspirados en las redes neuronales biológicas que constituyen el cerebro animal.

Consisten en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida [34].

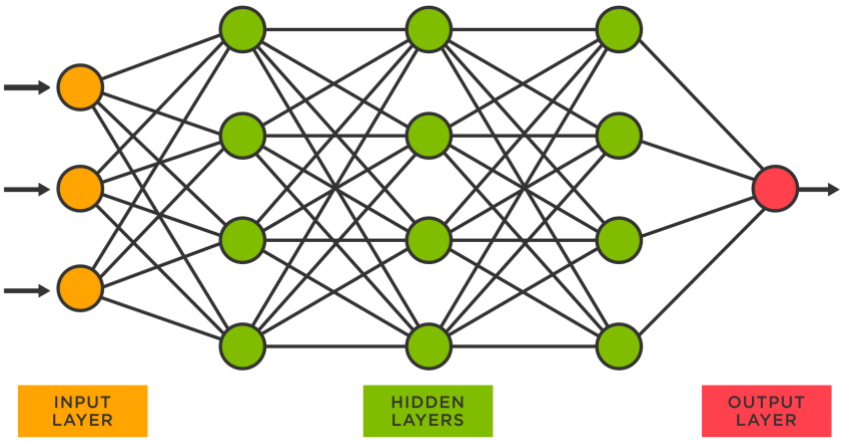

Figura 2-11. Red neuronal artificial.

Imagen capturada de: www.tibco.com

Estos modelos sobresalen en áreas donde la programación convencional no era capaz de encontrar soluciones, lo cual a su vez hace que sea más complejo explicar la toma de decisiones por lo que tienen una menor interpretabilidad que los modelos tradicionales. Estos modelos suelen llamarse de caja negra.

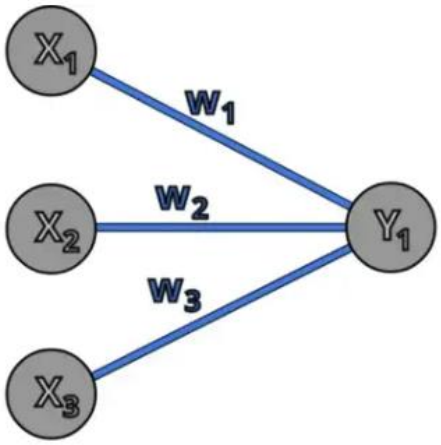

Figura 2-12. Red neuronal artificial.

Imagen capturada de: www. becominghuman.ai

La ecuación matemática que hace referencia a la red neuronal de la figura 2-12 es: $Y_1 = W_1X_1 + W_2X_2 + W_3X_3$ donde W_1, W_2 *y* W_3 hacen referencia al peso que tiene cada nodo en la decisión final [35].

2.3.6 Evaluación de rendimiento

Las siguientes métricas han sido calculadas y utilizadas para evaluar el rendimiento de los modelos de predicción empleados.

2.3.6.1 Error cuadrático medio

El error cuadrático medio (ECM) o *mean squared error (MSE)* de un modelo mide el promedio de los errores al cuadrado, es decir, la diferencia entre el modelo y lo que se estima. El ECM es una función de riesgo, correspondiente al valor esperado de la pérdida del error al cuadrado o pérdida cuadrática. La diferencia se produce debido a la aleatoriedad o porque el modelo no tiene en cuenta la información que podría producir una estimación más precisa [36].

Si $\hat{Y}$ es un vector de n predicciones y Y es el vector de los valores verdaderos, entonces el ECM del modelo es:

$$ECM = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2$$

2.3.6.2 Raíz del error cuadrático medio

La raíz del error cuadrático medio (RECM) o *root mean square error (RMSE)* es una medida usada muy frecuentemente para observar las diferencias entre los valores predichos por un modelo y los valores observados. Cuanto más pequeño es la RECM, más próximos están los valores predichos de los observados [37].

El RECM de un modelo $\hat{\theta}$ con respecto al parámetro estimado θ, se define como la raíz cuadrática del error cuadrático medio:

$$RECM(\hat{\theta}) = \sqrt{ECM(\hat{\theta})} = \sqrt{E\left((\hat{\theta} - \theta)^2\right)}$$

2.3.6.3 Error absoluto medio

El error absoluto medio (EAM) o *mean absolute error (MAE)* es una medida de la diferencia entre dos variables continuas. Considerando dos series de datos (unos calculados y otros observados) relativos a un mismo fenómeno, el error absoluto medio sirve para cuantificar la precisión de una técnica de predicción comparando por ejemplo los valores predichos frente a los observados, el tiempo real frente al tiempo previsto, o una técnica de medición frente a otra técnica alternativa de medición [38].

Dadas dos series de datos *(X e Y)* relativas a un mismo fenómeno, considérese un diagrama de dispersión de n puntos, donde el punto i tiene coordenadas (x_i, y_i). El error absoluto medio (EAM) es la distancia vertical promedio entre cada uno de los puntos y la recta identidad (y=x), o también la distancia horizontal promedio entre cada punto y la recta identidad.

Viene dado por:

$$EAM = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n}$$

2.3.6.4 Coeficiente de determinación

El coeficiente de determinación, denominado R^2 es usado en el contexto de un modelo estadístico cuyo principal propósito es predecir futuros resultados o probar una hipótesis. El coeficiente determina la calidad del modelo para replicar los resultados, y la proporción de variación de los resultados que puede explicarse por el modelo [39].

En regresión lineal es suficiente con hacer el cuadrado del coeficiente de correlación de Pearson.

$$R^2 = \frac{\sigma_{XY}^2}{\sigma_X^2 \sigma_Y^2}$$

Donde:

- σ_{XY} es la covarianza de (X, Y).
- σ_X^2 es la varianza de la variable X.
- σ_Y^2 es la varianza de la variable Y.

2.3.6.5 DataSet de entrenamiento y de prueba

Para comprobar la exactitud del modelo, se han utilizado datos distintos para su entrenamiento y para su validación. El *dataSet* completo tenía 1612 muestras, y este se ha dividido en 1209 muestras para entrenar y 403 muestras para validar.

```
Size of Full dataset is: (1612, 101)
```

```
from sklearn.model_selection import train_test_split
#Splitting the data
X_train, X_test, y_train, y_test = train_test_split(df,
                                                    target,
                                                    test_size=0.25,
                                                    random_state=7)
print("Number transactions X_train dataset: ", X_train.shape)
print("Number transactions y_train dataset: ", y_train.shape)
print("Number transactions X_test dataset: ", X_test.shape)
print("Number transactions y_test dataset: ", y_test.shape)
```

```
Number transactions X_train dataset:  (1209, 101)
Number transactions y_train dataset:  (1209,)
Number transactions X_test dataset:  (403, 101)
Number transactions y_test dataset:  (403,)
```

Figura 2-13. Extracto de código de la división del dataSet.

El motivo de utilizar datos distintos para el entrenamiento y para la validación es por evitar problemas de sobre aprendizaje o *over-fitting*, es decir, que el modelo esté demasiado ajustado a los datos proporcionados para el entrenamiento y no permita generalizar a nuevas muestras [40].

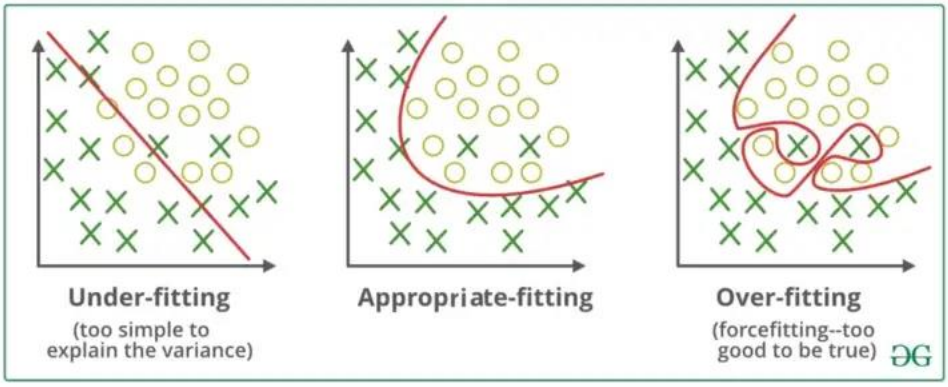

Figura 2-14. Imágenes de un modelo infra entrenado, entrenado correctamente y sobre entrenado.

Imagen capturada de: https://rubialesalberto.medium.com/qu%C3%A9-es-underfitting-y-overfitting-c73d51ffd3f9

2.4 Lenguaje y librerías utilizadas

2.4.1 Python

El lenguaje elegido para el desarrollo del proyecto ha sido Python. Esto ha sido así porque Python otorga muchísima flexibilidad y potencia gracias a sus librerías, además es el lenguaje líder actualmente en lo que a técnicas de *Machine Learning* se refiere.

Figura 2-15. Símbolo del lenguaje de programación Python.

Algunas de las características más importantes de Python son [41]:

- Lenguaje interpretado.
- Lenguaje tipeado dinámicamente.
- Lenguaje orientado a objetos.

2.4.2 NumPy

NumPy es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas [42].

Figura 2-16. Símbolo de la librería de Python numPy.

Algunas de las características más importantes son [43]:

- Está escrito en C por lo que proporciona una velocidad muy alta, elemento fundamental cuando se está trabajando con grandes conjuntos de datos.
- Incluye funciones para operaciones de diversos tipos: matemáticas, de ordenación, lógicas, estadísticas y para el tratamiento de ficheros.
- El objeto "ndarray" o "array" es el tipo de dato más importante en numPy y es clave en la gestión de matrices en Python.

2.4.3 Pandas

Pandas es una librería de Python especializada en la manipulación y el análisis de datos. Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales ***[44]***.

Figura 2-17. Símbolo de la librería de Python pandas.

Algunas de las características más importantes son:

- Tipo de datos *DataFrame* que permite manipular datos con indexación integrada.
- Herramientas para leer y exportar datos entre las estructuras cargadas en memoria y formatos de archivos diversos.
- Posibilidad de mezcla y unión de datos.
- Inserción y eliminación de columnas, así como renombramiento, en las estructuras de datos.

2.4.4 SeaBorn

SeaBorn es una librería de visualización de datos de Python basada en matplotlib. Proporciona una interfaz a alto nivel para dibujar atractivos e informativos gráficos estadísticos [45].

Figura 2-18. Símbolo de la librería de Python seaborn.

Algunas de las características más importantes son:

- Seaborn es capaz de entender nativamente un *DataFrame,* representando fácilmente distribuciones de datos, o agregaciones, sin desarrollar muchas líneas de código.
- Permite una amplia personalización de las visualizaciones.
- La galería de gráficos de Seaborn es de las más amplias y está centrada especialmente en la representación de análisis estadísticos de forma sencilla.

2.4.5 ScikitLearn

Scikit-learn es una biblioteca para *Machine Learning* de software libre para el lenguaje de programación Python. Incluye varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, *random forest*, Gradient boosting, K-means y DBSCAN. Está diseñada para interoperar con las bibliotecas numéricas y científicas NumPy y SciPy [46].

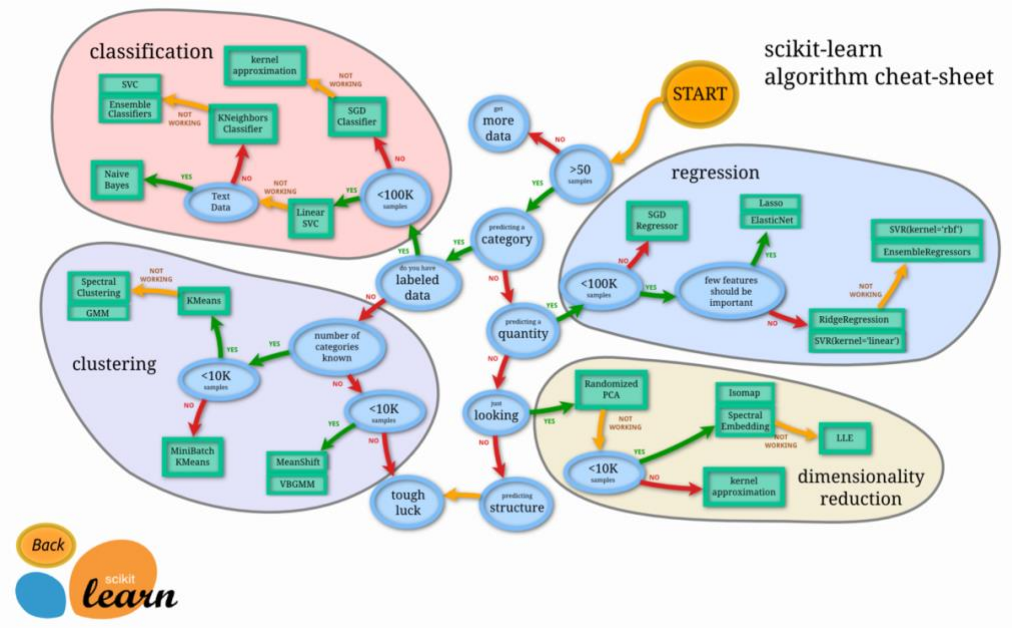

Figura 2-19. Esquema con los algoritmos más utilizados de scikitLearn.

Capítulo 3 - Recolección y preparación de los datos

Como se ha comentado previamente, se han utilizado 3 *dataSet* del Instituto Nacional de Estadística por lo que he tenido que realizar tareas de unificación para convertirlas en uno solo.

3.1 Fuentes de datos, descripción de características y homogeneización

El primer *dataSet* utilizado contenía información demográfica de la población española, así como su evolución en el tiempo, desglosada por comunidades.

	Sexo	Edad	Provincias	Periodo	Total
0	Ambos sexos	Total	Total Nacional	1 de enero de 2022	47.432.805
1	Ambos sexos	Total	Total Nacional	1 de julio de 2021	47.331.545
2	Ambos sexos	Total	Total Nacional	1 de enero de 2021	47.398.695
3	Ambos sexos	Total	Total Nacional	1 de julio de 2020	47.355.685
4	Ambos sexos	Total	Total Nacional	1 de enero de 2020	47.332.614
...	...	...	...	...	...
1686826	Mujeres	85 y más años	52 Melilla	1 de enero de 1973	192
1686827	Mujeres	85 y más años	52 Melilla	1 de julio de 1972	188
1686828	Mujeres	85 y más años	52 Melilla	1 de enero de 1972	183
1686829	Mujeres	85 y más años	52 Melilla	1 de julio de 1971	185
1686830	Mujeres	85 y más años	52 Melilla	1 de enero de 1971	189

1686831 rows × 5 columns

Figura 3-1. Extracto del primer dataSet utilizado.

Con el objetivo de homogeneizar los tres *dataSet*, se transformó la columna periodo. También se llevaron a cabo tareas de limpieza de datos nulos, así como la eliminación de información que no fuera necesaria para el modelo. El resultado final del *dataSet* se puede observar en la figura 3-2.

	Provincias	Periodo	TotalPoblación
0	Total Nacional	2022M01	47432805.0
1	Total Nacional	2021M07	47331545.0
2	Total Nacional	2021M01	47398695.0
3	Total Nacional	2020M07	47355685.0
4	Total Nacional	2020M01	47332614.0
...	...	...	...
5454	52 Melilla	1973M01	59549.0
5455	52 Melilla	1972M07	59882.0
5456	52 Melilla	1972M01	60260.0
5457	52 Melilla	1971M07	60514.0
5458	52 Melilla	1971M01	60815.0

5459 rows × 3 columns

Figura 3-2. Extracto del primer dataSet utilizado tras las tareas de limpieza y extracción de datos.

El segundo *dataSet* utilizado contiene información sobre la compraventa de las viviendas en España, es decir, sobre el número de transmisiones patrimoniales que se habían realizado en el tiempo, así como su desglose por comunidad autónoma y provincia.

	Total Nacional	Comunidades y Ciudades Autónomas	Provincias	Régimen y estado	Periodo	Total
0	Total Nacional	NaN	NaN	Viviendas: Total	2022M07	53.720
1	Total Nacional	NaN	NaN	Viviendas: Total	2022M06	58.010
2	Total Nacional	NaN	NaN	Viviendas: Total	2022M05	60.059
3	Total Nacional	NaN	NaN	Viviendas: Total	2022M04	47.349
4	Total Nacional	NaN	NaN	Viviendas: Total	2022M03	59.272
...	...	...	...	...	...	...
67315	Total Nacional	19 Melilla	52 Melilla	Vivienda protegida	2007M05	4
67316	Total Nacional	19 Melilla	52 Melilla	Vivienda protegida	2007M04	2
67317	Total Nacional	19 Melilla	52 Melilla	Vivienda protegida	2007M03	21
67318	Total Nacional	19 Melilla	52 Melilla	Vivienda protegida	2007M02	26
67319	Total Nacional	19 Melilla	52 Melilla	Vivienda protegida	2007M01	29

67320 rows × 6 columns

Figura 3-3. Extracto del segundo dataSet utilizado.

Al igual que con el primer *dataSet,* se realizaron tareas de limpieza y homogeneización de los datos, preparando así el *dataSet* para la unión con el primero. El resultado de estas tareas se puede observar en la figura 3-4.

	Comunidades y Ciudades Autónomas	Provincias	Periodo	NumCompraVentas
0	Total Nacional	Total Nacional	2022M07	53720.0
1	Total Nacional	Total Nacional	2022M06	58010.0
2	Total Nacional	Total Nacional	2022M05	60059.0
3	Total Nacional	Total Nacional	2022M04	47349.0
4	Total Nacional	Total Nacional	2022M03	59272.0
...	...	...	...	...
13459	19 Melilla	52 Melilla	2007M05	160.0
13460	19 Melilla	52 Melilla	2007M04	75.0
13461	19 Melilla	52 Melilla	2007M03	105.0
13462	19 Melilla	52 Melilla	2007M02	177.0
13463	19 Melilla	52 Melilla	2007M01	179.0

13464 rows × 4 columns

Figura 3-4. Extracto del segundo dataSet utilizado tras las tareas de limpieza y extracción de datos.

El siguiente paso fue, como se ha comentado, la unión de estos dos primeros *dataSet*, para ello se hizo uso de la función *merge* de la librería pandas, resultando el *dataSet* de la figura 3-5.

	Comunidades y Ciudades Autónomas	Provincias	Periodo	NumCompraVentas	TotalPoblación
0	Total Nacional	Total Nacional	2022M01	52684.0	47432805.0
1	Total Nacional	Total Nacional	2021M07	49732.0	47331545.0
2	Total Nacional	Total Nacional	2021M01	40213.0	47398695.0
3	Total Nacional	Total Nacional	2020M07	32751.0	47355685.0
4	Total Nacional	Total Nacional	2020M01	47017.0	47332614.0
...	...	...	...	...	...
1607	19 Melilla	52 Melilla	2009M01	33.0	73361.0
1608	19 Melilla	52 Melilla	2008M07	71.0	72213.0
1609	19 Melilla	52 Melilla	2008M01	79.0	71244.0
1610	19 Melilla	52 Melilla	2007M07	83.0	70080.0
1611	19 Melilla	52 Melilla	2007M01	179.0	68968.0

1643 rows × 5 columns

Figura 3-5. Extracto del dataSet resultante de la unión de los dos primeros.

Como se puede observar, el *dataSet* resultante es de menor tamaño que los originales debido a que la serie temporal disponible en el segundo *dataSet* es bastante menor que la del primer *dataSet.*

El tercer y último *dataSet* utilizado contiene información sobre el precio medio de la vivienda en cada comunidad y provincia española.

	Total Nacional	Comunidades y Ciudades Autónomas	General, vivienda nueva y de segunda mano	Índices y tasas	Periodo	Total
0	Nacional	NaN	General	Índice	2022T2	141.433
1	Nacional	NaN	General	Índice	2022T1	138.742
2	Nacional	NaN	General	Índice	2021T4	135.291
3	Nacional	NaN	General	Índice	2021T3	133.652
4	Nacional	NaN	General	Índice	2021T2	130.937
...	...	...	...	...	...	...
14875	Nacional	19 Melilla	Vivienda segunda mano	Variación en lo que va de año	2008T1	NaN
14876	Nacional	19 Melilla	Vivienda segunda mano	Variación en lo que va de año	2007T4	NaN
14877	Nacional	19 Melilla	Vivienda segunda mano	Variación en lo que va de año	2007T3	NaN
14878	Nacional	19 Melilla	Vivienda segunda mano	Variación en lo que va de año	2007T2	NaN
14879	Nacional	19 Melilla	Vivienda segunda mano	Variación en lo que va de año	2007T1	NaN

14880 rows × 6 columns

Figura 3-6. Extracto del tercer dataSet utilizado.

Al igual que con el primer y segundo *dataSets,* se realizaron tareas de limpieza y homogeneización de los datos. Se eliminaron los índices y tasas del *dataSet,* así como la segmentación entre viviendas nuevas y de segunda mano, quedándome solamente con el dato general. El resultado de estas tareas se puede observar en la figura 4-4.

	Comunidades y Ciudades Autónomas	Periodo	PrecioVivienda
0	Total Nacional	2022M04	141433.0
1	Total Nacional	2022M01	138742.0
2	Total Nacional	2021M10	135291.0
3	Total Nacional	2021M07	133652.0
4	Total Nacional	2021M04	130937.0
...	...	...	...
1235	19 Melilla	2008M01	135752.0
1236	19 Melilla	2007M10	134502.0
1237	19 Melilla	2007M07	132239.0
1238	19 Melilla	2007M04	128738.0
1239	19 Melilla	2007M01	124877.0

1240 rows × 3 columns

Figura 3-7. Extracto del tercer dataSet utilizado tras las tareas de limpieza y extracción de datos.

Por último, se procedió a unir este tercer *dataSet* con el previamente obtenido de la unión de los dos primeros. Como este último *dataSet* no tenía del detalle de las provincias, tras la unión, el precio medio de la vivienda en cada provincia es equivalente al precio medio de la vivienda en su comunidad.

	Comunidades y Ciudades Autónomas	Provincias	Periodo	NumCompraVentas	TotalPoblación	PrecioVivienda
0	Total Nacional	Total Nacional	2022M01	52684.0	47432805.0	138742.0
1	Total Nacional	Total Nacional	2021M07	49732.0	47331545.0	133652.0
2	Total Nacional	Total Nacional	2021M01	40213.0	47398695.0	127831.0
3	Total Nacional	Total Nacional	2020M07	32751.0	47355685.0	128255.0
4	Total Nacional	Total Nacional	2020M01	47017.0	47332614.0	126695.0
...	...	...	...	...	...	...
1638	19 Melilla	52 Melilla	2009M01	33.0	73361.0	135458.0
1639	19 Melilla	52 Melilla	2008M07	71.0	72213.0	133498.0
1640	19 Melilla	52 Melilla	2008M01	79.0	71244.0	135752.0
1641	19 Melilla	52 Melilla	2007M07	83.0	70080.0	132239.0
1642	19 Melilla	52 Melilla	2007M01	179.0	68968.0	124877.0

1643 rows × 6 columns

Figura 3-8. Extracto del dataSet final utilizado para entrenar al modelo.

Capítulo 4 - Análisis exploratorio de datos

En este capítulo se va a estudiar la distribución de los datos, así como la relación entre ellos para determinar las variables que más influyen en la predicción del precio medio de la vivienda en una comunidad.

4.1 Carga del *dataSet*

Para facilitar la importación y exportación de datos entre los distintos códigos fuente, he utilizado el formato. pkl, originario de Python.

```
df_train = pd.read_pickle('dataSet.pkl')
```

```
df_train.columns
```

```
Index(['Comunidades y Ciudades Autónomas', 'Provincias', 'Periodo',
       'NumCompraVentas', 'TotalPoblación', 'PrecioVivienda'],
      dtype='object')
```

```
df_train
```

	Comunidades y Ciudades Autónomas	Provincias	Periodo	NumCompraVentas	TotalPoblación	PrecioVivienda
0	Total Nacional	Total Nacional	2022M01	52684.0	47432805.0	138742.0
1	Total Nacional	Total Nacional	2021M07	49732.0	47331545.0	133652.0
2	Total Nacional	Total Nacional	2021M01	40213.0	47398695.0	127831.0
3	Total Nacional	Total Nacional	2020M07	32751.0	47355685.0	128255.0
4	Total Nacional	Total Nacional	2020M01	47017.0	47332614.0	126695.0
...	...	...	...	...	...	...
1638	19 Melilla	52 Melilla	2009M01	33.0	73361.0	135458.0
1639	19 Melilla	52 Melilla	2008M07	71.0	72213.0	133498.0
1640	19 Melilla	52 Melilla	2008M01	79.0	71244.0	135752.0
1641	19 Melilla	52 Melilla	2007M07	83.0	70080.0	132239.0
1642	19 Melilla	52 Melilla	2007M01	179.0	68968.0	124877.0

1643 rows × 6 columns

Figura 4-1. Extracto de la importación, las columnas y la información del dataSet a utilizar para el entrenamiento.

4.2 Inspección del *dataSet*

En la muestra de datos tenemos las siguientes columnas o características:

- Comunidades y Ciudades Autónomas: Esta columna contiene el nombre de la comunidad autónoma a la que hace referencia el precio de la columna "PrecioVivienda" o "Total Nacional" si se refiere al precio medio de la vivienda en toda España.

- Provincias: Esta columna contiene el nombre de la provincia a la que hace referencia el precio de la columna "PrecioVivienda" o "Total Nacional" si se refiere precio medio de la vivienda en toda España.
- Periodo: Esta columna indica la fecha en la que se tomaron los datos. Los valores van desde enero de 2007 hasta enero de 2022.
- NumCompraVentas: Esta columna indica el número de transmisiones patrimoniales que hubo en esa provincia, o en toda España, en un periodo determinado.
- TotalPoblación: Esta columna indica el número total de población que había en esa provincia, o en toda España, en el periodo indicado.
- PrecioVivienda: Esta columna indica el precio medio de la vivienda en esa provincia, o en toda España, en el periodo indicado. Quiero destacar que como comentado, el valor del precio medio de la comunidad se ha trasladado al valor del precio medio de cada provincia de dicha comunidad. Esta es la variable objetivo o variable que predecir.

Comencé analizando las variables numéricas, apoyándome en las distintas funciones de las que dispone la librería pandas de Python.

	NumCompraVentas	TotalPoblación	PrecioVivienda
count	1643.000000	1.643000e+03	1643.000000
mean	1572.932441	1.757390e+06	119806.125989
std	5900.354316	6.319045e+06	18960.845997
min	0.000000	6.896800e+04	91832.000000
25%	232.000000	3.241155e+05	102700.000000
50%	483.000000	6.349830e+05	115386.000000
75%	932.500000	1.067004e+06	134773.500000
max	83713.000000	4.743280e+07	186732.000000

Figura 4-2. Descripción de las variables numéricas del dataSet.

De la figura 4-2 se pueden obtener las siguientes conclusiones:

- El precio medio de vivienda más bajo en una provincia es de 91.832 euros.
- El precio medio de vivienda más alto es de 186.732 euros.

- La provincia con menos habitantes tiene 68.968.
- La población total de España es de 47,43 millones de habitantes.

El siguiente paso fue analizar la variable a predecir, así como la distribución de sus valores, para ello me basé en el histograma de la figura 4-3.

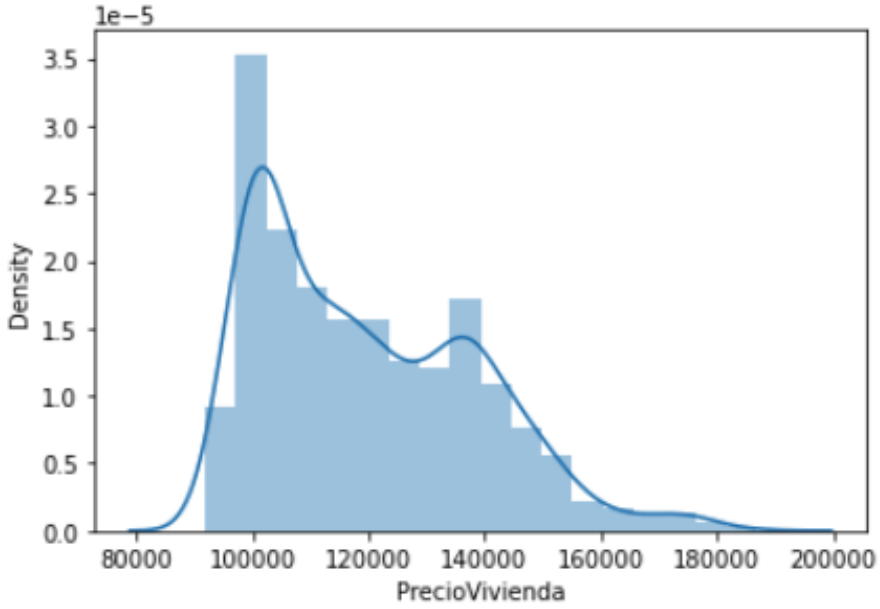

Figura 4-3. Histograma de la variable a predecir.

Así mismo, se analizaron las posibles relaciones lineales que pudiera haber en este *dataSet*, para ello observé la relación entre el número de compraventas y el precio medio de la vivienda, ya que debería ser lineal. Para los cálculos siguientes, eliminé del *dataSet* las filas correspondientes al total nacional para tener una mejor distribución de los datos.

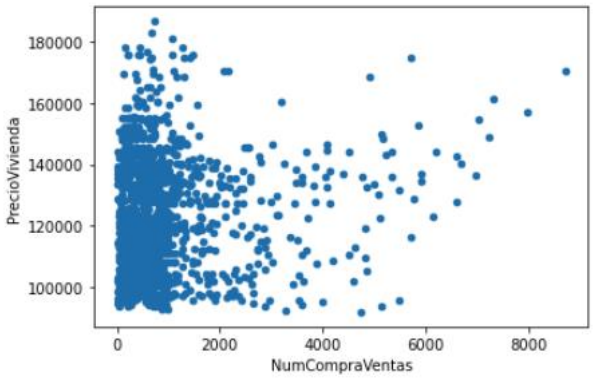

Figura 4-4. Relación entre el número de compraventas de vivienda y el precio medio.

Como se puede observar en las figuras 5-4, existe una relación lineal entre el número de compraventas que hay en una provincia y el precio medio de la vivienda en esta, especialmente notable para las provincias que tienen más de 2000 compraventas.

La siguiente variable que analicé fue la de la población. El objetivo era ver si existe una relación entre la población en una comunidad y el precio medio de la vivienda en esta. Para ello obtuve los gráficos de las figuras 4-5 y 4-6.

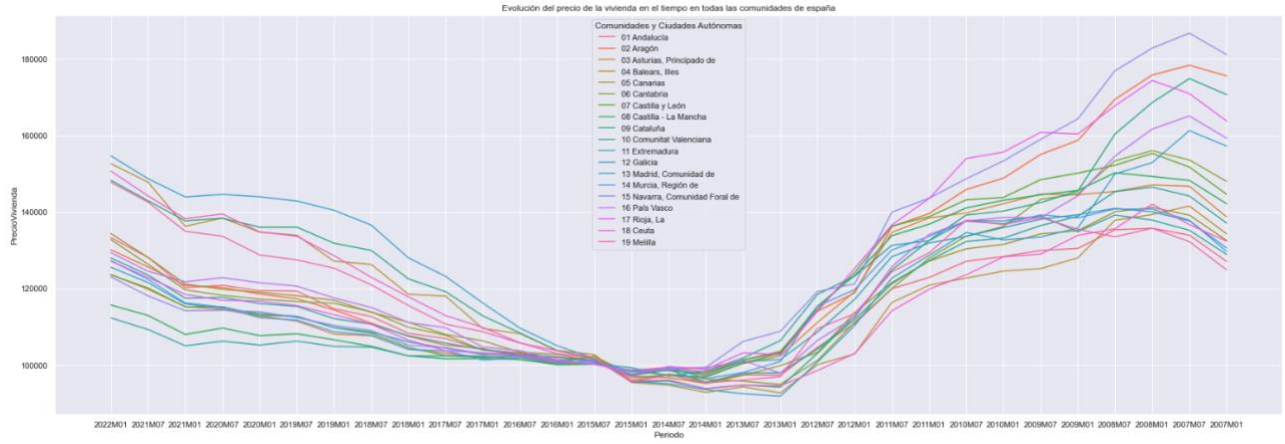

Figura 4-5. Evolución del precio de la vivienda a lo largo del tiempo.

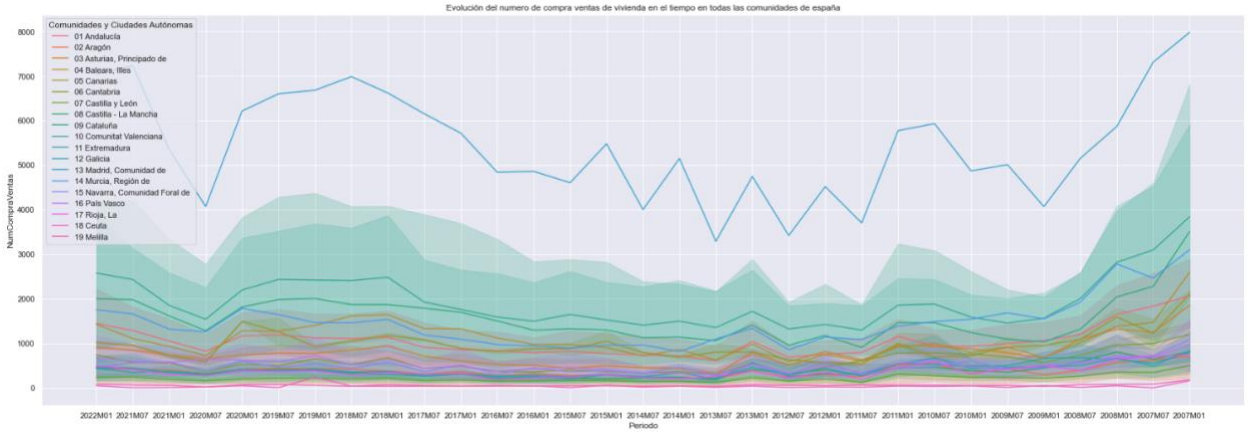

Figura 4-6. Evolución del número de habitantes a lo largo del tiempo.

Lo que se observa en las figuras 4-5 y 4-6 es que si existe una cierta relación directa entre el número de habitantes de una provincia y el precio medio de la vivienda en esa provincia. Cuando se realice el modelo, veremos si esto se cumple.

Así mismo, para confirmar las hipótesis, obtuve la matriz de correlación, presente en la figura 4-7. Como era de esperar, la variable que más influye en el precio medio de la vivienda es el número de compraventas que hay en esa provincia.

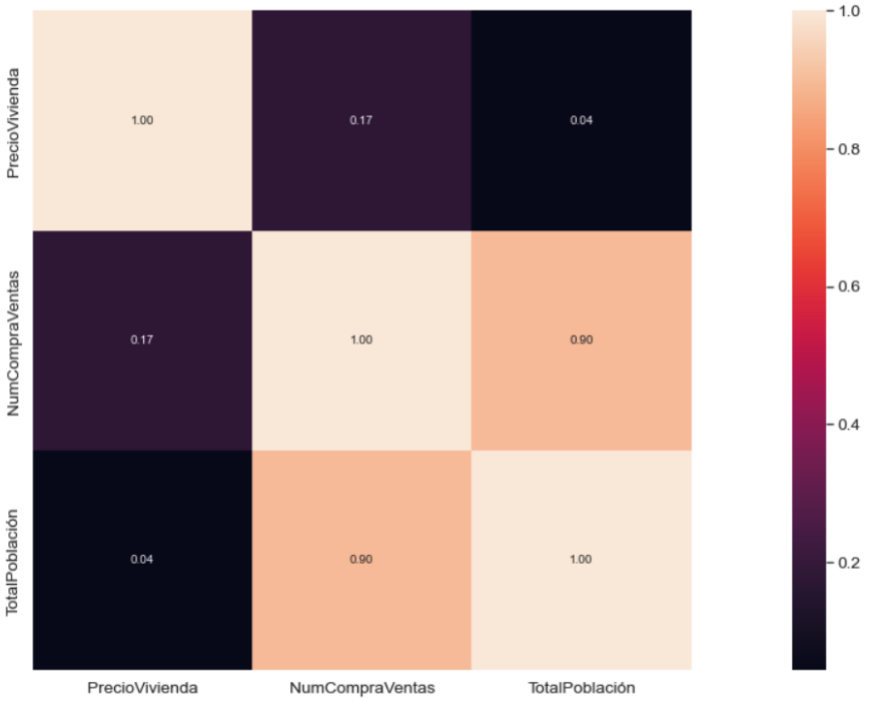

Figura 4-7. Matriz de correlación.

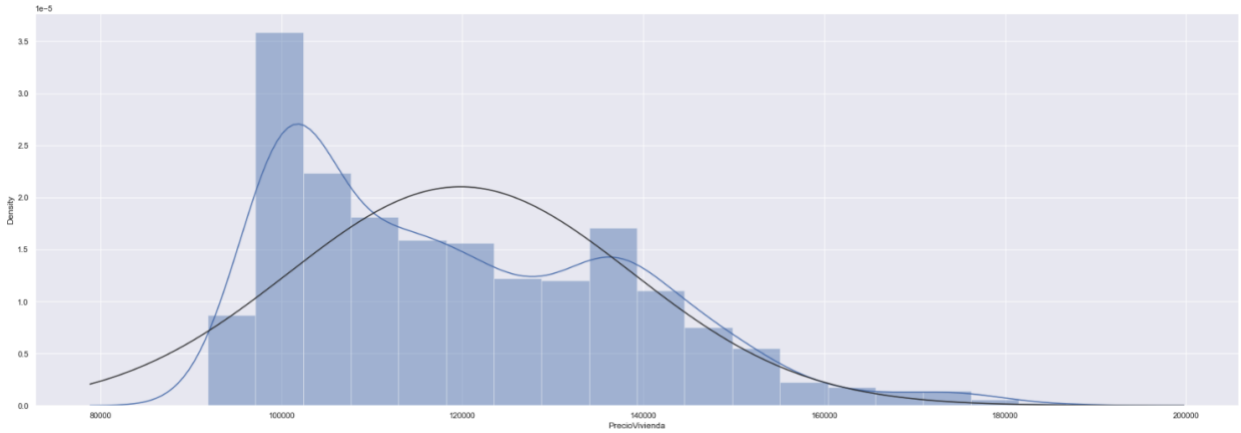

Figura 4-8. Histograma del precio de la vivienda

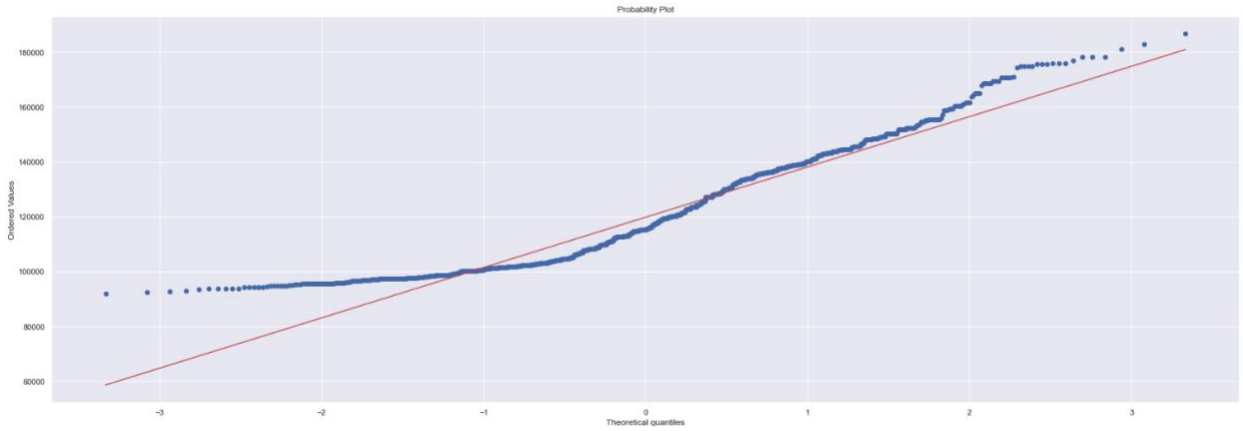

Figura 4-9. Gráfico de probabilidad normal del precio de la vivienda

4.3 Conclusiones

Tras haber entendido los datos y la relación entre las variables, el *dataSet* resultante es el expuesto en la figura 4-10. Este *dataSet* contiene información relevante y ordenada para intentar predecir el precio de la vivienda en una comunidad autónoma.

	Comunidades y Ciudades Autónomas	Provincias	Periodo	NumCompraVentas	TotalPoblación	PrecioVivienda
31	01 Andalucía	04 Almería	2022M01	1220.0	723899.0	11.800358
32	01 Andalucía	11 Cádiz	2022M01	1417.0	1259339.0	11.800358
33	01 Andalucía	14 Córdoba	2022M01	675.0	777414.0	11.800358
34	01 Andalucía	18 Granada	2022M01	1204.0	929968.0	11.800358
35	01 Andalucía	21 Huelva	2022M01	688.0	532865.0	11.800358
...	...	...	...	...	...	...
1638	19 Melilla	52 Melilla	2009M01	33.0	73361.0	11.816417
1639	19 Melilla	52 Melilla	2008M07	71.0	72213.0	11.801842
1640	19 Melilla	52 Melilla	2008M01	79.0	71244.0	11.818585
1641	19 Melilla	52 Melilla	2007M07	83.0	70080.0	11.792366
1642	19 Melilla	52 Melilla	2007M01	179.0	68968.0	11.735085

1612 rows × 6 columns

Figura 4-10. DataSet utilizado para el entrenamiento.

Capítulo 5 - Modelado

En este capítulo se entrenarán varios modelos de *Machine Learning* utilizando el *dataSet* resultante del tratamiento explicado en el capítulo 4 con el fin de predecir el precio de la vivienda. También se obtendrán métricas que permitan estudiar la exactitud de los modelos aplicados.

5.1 Pasos previos al modelaje

Lo primero que hice fue codificar las columnas que contenían texto en columnas con datos numéricos, para adecuar los datos al modelo. Así mismo realicé un escalado para que todos los valores se encontraran en el mismo rango. El resultado de estos dos pasos son las figuras 5-1 y 5-2.

	NumCompraVentas	TotalPoblación	PrecioVivienda	Comunidades y Ciudades Autónomas_02 Aragón	Comunidades y Ciudades Autónomas_03 Asturias, Principado de	Comunidades y Ciudades Autónomas_04 Balears, Illes	Comunidades y Ciudades Autónomas_05 Canarias	Comunidades y Ciudades Autónomas_06 Cantabria	Comunidades y Ciudades Autónomas_07 Castilla y León	Comunidades y Ciudades Autónomas_08 Castilla - La Mancha	...
31	1220.0	723899.0	11.800358	0	0	0	0	0	0	0	...
32	1417.0	1259339.0	11.800358	0	0	0	0	0	0	0	...
33	675.0	777414.0	11.800358	0	0	0	0	0	0	0	...
34	1204.0	929968.0	11.800358	0	0	0	0	0	0	0	...
35	688.0	532865.0	11.800358	0	0	0	0	0	0	0	...

5 rows × 102 columns

Figura 5-1. DataSet resultante de la categorización.

	NumCompraVentas	TotalPoblación	PrecioVivienda	Comunidades y Ciudades Autónomas_02 Aragón	Comunidades y Ciudades Autónomas_03 Asturias, Principado de	Comunidades y Ciudades Autónomas_04 Balears, Illes	Comunidades y Ciudades Autónomas_05 Canarias	Comunidades y Ciudades Autónomas_06 Cantabria	Comunidades y Ciudades Autónomas_07 Castilla y León	Comunidades y Ciudades Autónomas_08 Castilla - La Mancha	...
31	0.698900	0.488744	11.800358	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
32	0.811755	0.888317	11.800358	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
33	0.386687	0.528680	11.800358	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
34	0.689734	0.642523	11.800358	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
35	0.394134	0.346184	11.800358	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

5 rows × 102 columns

Figura 5-2. DataSet resultante del escalado.

```
NumCompraVentas                                          float64
TotalPoblación                                           float64
PrecioVivienda                                           float32
Comunidades y Ciudades Autónomas_02 Aragón               float64
Comunidades y Ciudades Autónomas_03 Asturias, Principado de    float64
                                                          ...
Periodo_2020M01                                          float64
Periodo_2020M07                                          float64
Periodo_2021M01                                          float64
Periodo_2021M07                                          float64
Periodo_2022M01                                          float64
Length: 102, dtype: object
Size of Full Encoded Dataset: (1612, 102)
```

Figura 5-3. Columnas del dataSet resultante de la categorización y el escalado.

5.2 Regresión lineal

La regresión lineal fue el primer estimador a aplicar, debido a su simplicidad, para predecir el precio de la vivienda. Los resultados obtenidos se pueden observar en la tabla 5-1.

MAE	MSE	RMSE	R cuadrado
0.03863639926200469	0.002543274226026331	0.05043088563595063	0.886492492052116

Tabla 5-1. Resultados regresión lineal.

5.3 Random forest

Random Forect fue el segundo estimador utilizado para predecir el precio de la vivienda y se utilizó con los siguientes parámetros:

- *Bootstrap=False*
- *Max_features=30*
- *N_estimator=300*
- *Random_state=42*

Cabe destacar que la elección de estos parámetros no ha sido aleatoria, sino que se utilizó el método *GridSearchCV* para tomar la decisión de que parámetros proporcionaban una mejor estimación [47].

Los resultados obtenidos con este estimador se pueden observar en la tabla 5-2.

MAE	MSE	RMSE	R cuadrado
0.02970551483684376	0.0024011377070017915	0.04900140515334016	0.8928361108006415

Tabla 5-2. Resultados random forest.

5.4 SVR

SVR fue el tercer estimador utilizado para predecir el precio de la vivienda y se usó con el siguiente parámetro:

- *Kernel='linear'*

Los resultados obtenidos con el estimador se pueden observar en la tabla 5-3.

MAE	MSE	RMSE	R cuadrado
0.049977196614524946	0.0037728902385113713	0.06142385724220982	0.8316141592786739

Tabla 5-3. Resultados SVR.

5.5 Regresor SGD

El regresor SGD fue el cuarto estimador utilizado para la predicción del precio de la vivienda. Los resultados obtenidos con este estimador se pueden observar en la tabla 5-4.

MAE	MSE	RMSE	R cuadrado
0.14773002112456973	0.08701437068255348	0.2949819836575676	-2.8834917095282244

Tabla 5-4. Resultados regresor SGD.

5.6 Regresor XGBoost

El regresor XGBoost fue el quinto y último estimador utilizado para predecir el precio de la vivienda. Los resultados obtenidos con este estimador se pueden observar en la tabla 6-5. Se puede observar que fue el estimador que mejor funcionó, llegando al predecir correctamente el 91,35% de los casos.

MAE	MSE	RMSE	R cuadrado
0.028959684	0.0019375305	0.04401739	0.9135271209014462

Tabla 5-5. Resultados regresor XGBoost.

5.7 Comparación de estimadores

En la figura 5-6, se resumen los resultados de los distintos estimadores utilizados siendo, como he comentado, XGBoost el más certero en la predicción del precio de la vivienda.

El hecho de que XGBoost haya sido el más certero, obliga a hacer un análisis de interpretabilidad para analizar los pesos de las variables en este modelo, ya que se trata de un estimador de caja negra. La interpretabilidad del algoritmo es fundamental en

este caso para saber que variables son las que más influyen en el precio de la vivienda y por tanto saber qué características se deben tener en cuenta a la hora de invertir en una vivienda.

Estimador	MAE	MSE	RMSE	R cuadrado o coeficiente de determinación
Regresión lineal	0.03863639926200469	0.002543274226026331	0.05043088563595063	0.886492492052116
Random forest	0.02970551483684376	0.0024011377070017915	0.04900140515334016	0.8928361108006415
SVR	0.049977196614524946	0.0037728902385113713	0.06142385724220982	0.8316141592786739
Regresor SGD	0.14773002112456973	0.08701437068255348	0.2949819836575676	-2.8834917095282244
Regresor XGBoost	0.028959684	0.0019375305	0.04401739	0.9135271209014462

Tabla 5-6. Resultado global de los estimadores.

Como se puede observar en la tabla, los estimadores que tienen mejores métricas y, por tanto, proporcionarán mejores predicciones, son el regresor XGBoost con un coeficiente de determinación de 0.9135 y el estimador Random Forest con un coeficiente de determinación de 0.8929.

Ambos estimadores tiene un coeficiente de determinación positivo y muy cercano al 1 por lo que las predicciones serán bastante precisas. El regresor SGD en cambio ha tenido un coeficiente de correlación muy negativo, por lo que se descarta su uso para la predicción de precios de viviendas.

Capítulo 6 - Interpretabilidad – XAI

En este capítulo se proporcionará una explicación a las decisiones tomadas por parte del algoritmo XGBoost, que utiliza *Gradient boosting* o potenciación del gradiente, técnica de *Machine Learning* utilizada para el análisis de la regresión y para problemas de clasificación estadística.

XGBoost produce un modelo predictivo en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión [48], por lo que es complicado saber qué características han sido tenidas en cuenta en mayor o menor medida para la obtención del resultado final sin la utilización de técnicas de interpretabilidad.

6.1 Interpretabilidad - XAI

La inteligencia artificial explicable (XAI) hace referencia a los métodos y técnicas aplicadas a la inteligencia artificial (IA) que permiten entender los resultados por parte de los humanos. Es el término opuesto al concepto de caja negra, donde sus diseñadores no pueden explicar porque la IA ha tomado una determinada decisión [49].

Los métodos usados para la interpretabilidad de los algoritmos de *Machine Learning* pueden ser clasificados de formas muy diversas. Caben destacar dos clasificaciones principales [50]:

- Modelos intrínsecamente interpretables: son los modelos que se consideran interpretables debido a la sencillez de su estructura, como puede ser un modelo de regresión lineal o un modelo basado en un árbol de decisión corto.
- Métodos de interpretabilidad *post hoc* (y modelo-agnósticos)*:* estos métodos separan las explicaciones del modelo de *Machine Learning*. Este procedimiento tiene algunas ventajas como que los métodos de interpretación son independientes del modelo por lo que tienen una gran flexibilidad, ya que son aplicables a cualquier modelo.

También existe otra clasificación tipos de métodos de interpretabilidad, que se basa en el alcance de estos [50]:

- Interpretabilidad local: Este tipo de interpretabilidad trata de comprender como toma decisiones el modelo, respecto a una instancia en concreto. Es

decir, analiza los valores del modelo que han provocado una predicción individual.

- Interpretabilidad global: Este otro tipo de interpretabilidad trata de comprender cómo toma decisiones el modelo, en función de una visión holística de sus características y de cada uno de los componentes aprendidos, como los pesos, parámetros y estructuras. La interpretación global del modelo ayuda a comprender la distribución de su resultado objetivo global en función de las características.

Para analizar la interpretabilidad del modelo XGBoost, utilizaré el método SHAP, que permite una interpretabilidad tanto local como global. Además, es un método agnóstico que puede ser aplicado a cualquier modelo.

6.2 Método SHAP

SHAP es un método basado en la teoría de juegos cooperativa y usado para incrementar la transparencia e interpretabilidad de los modelos de *Machine Learning* [51]. Además, existe una librería de Python [52], que permite la aplicación de este método.

Es importante destacar que, aunque SHAP muestra la contribución y la importancia de cada característica en la predicción del modelo, no evalúa la calidad de la propia predicción.

Cuando se trata de explicar la interpretabilidad del modelo utilizado, esta puede ser abordada desde dos puntos, una interpretabilidad local y otra global.

6.2.1 Interpretación local con SHAP

Para analizar una instancia local y ver los efectos de cada variable en la predicción de esa instancia, podemos usar el siguiente gráfico sobre una observación en particular.

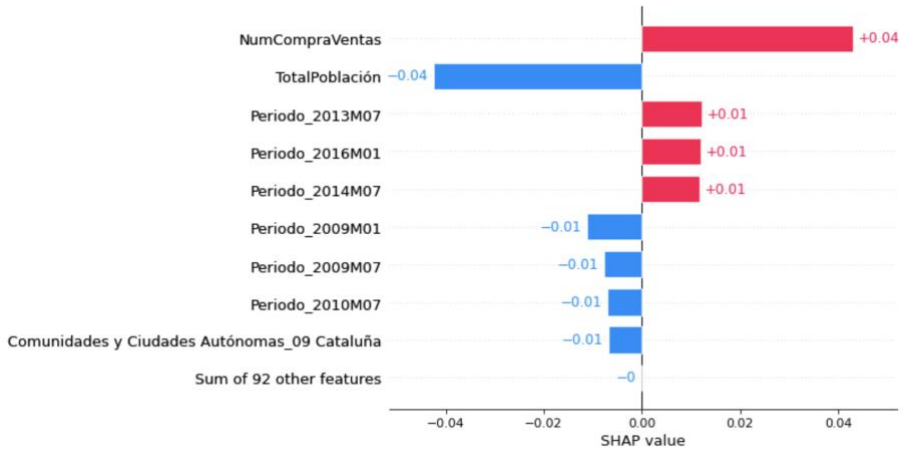

Figura 6-1. Valores de shapley para la observación siete.

En este caso, observamos en la figura 6-1 los valores de shapley para cada característica de una observación en concreto.

Se puede deducir que las variables que más inciden en la predicción son NumCompraVentas, que tiene una contribución positiva, y el TotalPoblación que tiene una contribución negativa en el valor a predecir.

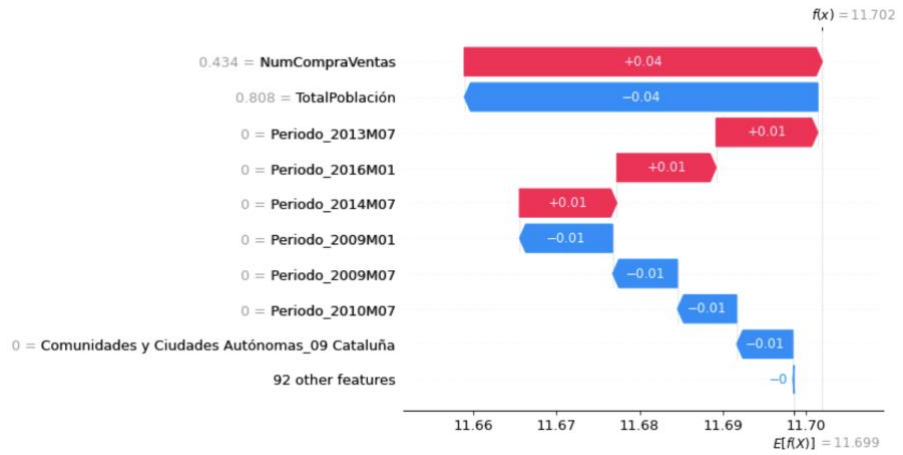

Figura 6-2. Valores de shapley para la observación siete junto con su valor de esperado.

Este gráfico contiene la misma información, representada de manera diferente. Aquí podemos observar como la suma de todos los valores de shapley equivalen a la diferencia entre la predicción f(x) que es 11.702 y el valor esperado E[f(x)] que es 11.699.

Por último, en la figura 6-3 se puede ver el efecto de cada característica en la predicción, para una observación dada, la siete en este caso.

En este gráfico, los valores de shapley positivos se encuentran en el lado izquierdo y los negativos en el lado derecho, como si estuvieran compitiendo entre ellos. El valor subrayado, es la predicción para esa observación.

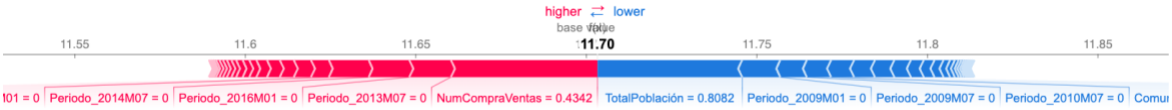

Figura 6-3. Valores de shapley para la observación siete junto con su valor de esperado en forma lineal.

6.2.2 Interpretación global con SHAP

En este apartado procedo a analizar el efecto global de cada característica en la predicción, es decir, la foto global de cómo se comporta el modelo para unos datos de entrada.

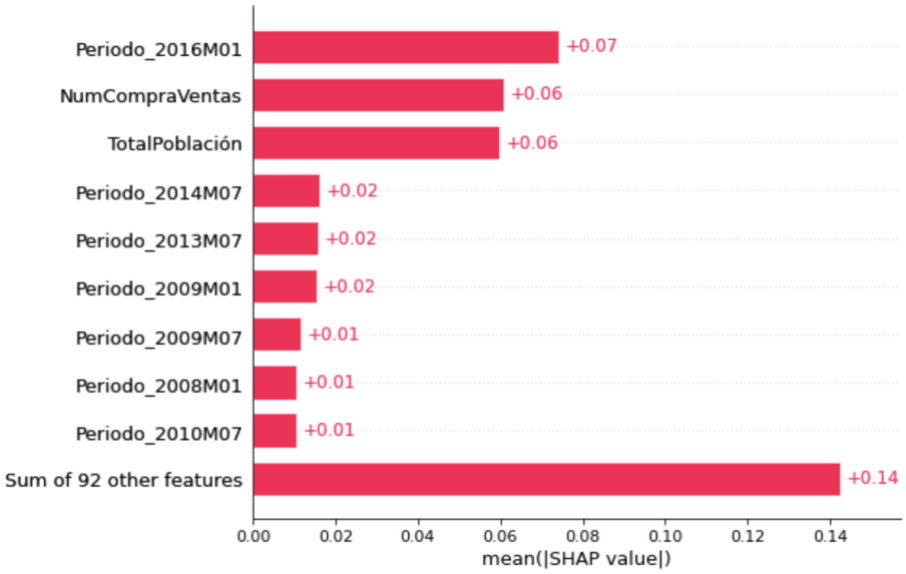

Figura 6-4. Valores de shapley en valor absoluto.

En la figura 6-4, las características están ordenadas desde las que tienen mayor efecto en la predicción hasta las de menor efecto. Se tiene en cuenta el valor absoluto de shapley, por lo que no importa si la característica afecta a la predicción en sentido positivo o negativo.

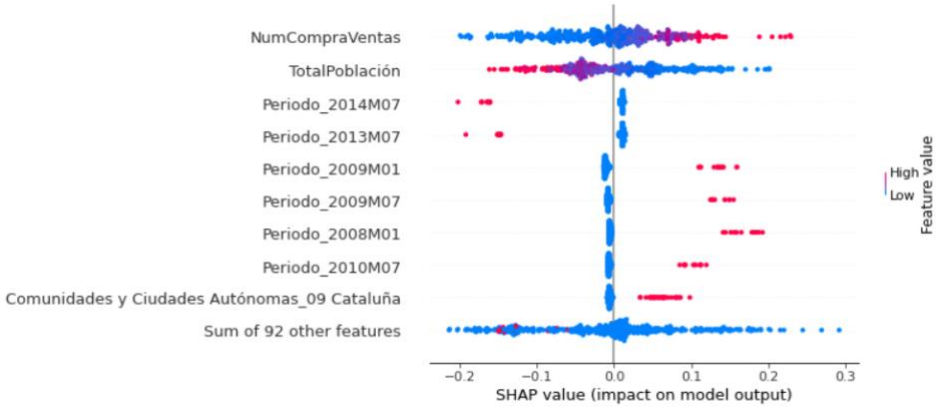

Figura 6-5. Valores de shapley para el modelo de regresión lineal.

En la figura 6-5, las características están también ordenadas en función del grado en el que afectan a la predicción, pero a mayores podemos ver cómo afectan en el resultado de la predicción los valores bajos o elevados.

Se observa que los valores bajos de la variable NumCompraVentas tienen una contribución negativa en la predicción mientras que los valores altos tienen una contribución positiva. Por otro lado, los valores bajos de la variable TotalPoblación tienen una contribución positiva en la predicción mientras que los valores altos tienen una contribución negativa.

Capítulo 7 - Conclusiones y trabajo futuro

7.1 Conclusiones

Al comienzo de este proyecto, se estableció un objetivo principal que a su vez se puede dividir en dos partes. La primera era proporcionar un método de trabajo basado en técnicas de *Machine Learning*, aplicable a cualquier dataSet futuro relacionado con el mercado inmobiliario. La segunda consistía en aplicar técnicas de interpretabilidad a los resultados obtenidos del modelo de *Machine Learning*.

Lo primera parte se ha conseguido mediante una introducción general de las técnicas de *Machine Learning* existentes, la metodología de trabajo que se debe seguir, las herramientas disponibles y las fuentes de datos que se pueden utilizar para analizar el mercado inmobiliario español. Así mismo se realizó el entrenamiento de varios modelos de *Machine Learning*, siguiendo la metodología propuesta, mediante datos obtenidos de la web del Instituto Nacional de Estadística.

Se ha de destacar el problema surgido con el API de idealista, cuya información era más completa que la obtenida del INE y por tanto se hubieran podido obtener unas mejores conclusiones.

La segunda parte se ha conseguido mediante la utilización de la librería de SHAP, que permite explicar las predicciones de cualquier modelo de *Machine Learning*. El resultado ha sido poder entender la importancia que le da el regresor XGBoost a cada una de las variables de entrada para predecir el resultado de salida.

7.2 Trabajo futuro

Como trabajo futuro, se plantean los siguientes puntos:

- Utilización de redes neuronales artificiales, como pueden ser las monocapa o perceptrón simple, las de perceptrón multicapa (MLP), las convolucionales (CNN), las redes neuronales recurrentes (RNN) y de retroalimentación o de base radial (RBF).
- Utilización de un dataSet con características propias de cada inmueble, como puede ser la superficie habitable, el número de habitaciones o el

año de construcción. Para ello se deberá obtener información mediante el API de Idealista o haciendo *web scraping* en la web de un portal inmobiliario español.

- Utilización de otros métodos de interpretabilidad.

Capítulo 8 - Conclusions and future work

8.1 Conclusions

At the beginning of this project, a main objective was established, which can be divided into two parts. The first was to provide a working method based on Machine Learning techniques, applicable to any future dataSet related to the real estate market. The second consisted of applying interpretability techniques to the results obtained from the Machine Learning model.

The first part has been achieved through a general introduction of the existing Machine Learning techniques, the work methodology that must be followed, the available tools and the data sources that can be used to analyze the Spanish real estate market. Likewise, the training of several Machine Learning models was carried out, following the proposed methodology, using data obtained from the website of the National Institute of Statistics.

The problem that arose with the idealista API should be highlighted, whose information was more complete than the obtained from the INE and therefore better conclusions could have been obtained.

The second part has been achieved by using the SHAP library, which allows explaining the predictions of any Machine Learning model. The result has been to be able to understand the importance that the XGBoost regressor gives to each of the input variables to predict the output result.

8.2 Future work

As future work, the following points are raised:

- Use of artificial neural networks, such as monolayer or simple perceptron, multilayer perceptron (MLP), convolutional (CNN), recurrent neural networks (RNN) and feedback or radial basis (RBF).
- Use of a dataSet with the characteristics of each property, such as the habitable area, the number of rooms or the year of construction. For this,

information must be obtained through the Idealista API or by doing web scraping on the website of a Spanish real estate portal.

- Use of other interpretability methods.

BIBLIOGRAFÍA

[1] C. García, «Solo el 22% de toda la riqueza de los españoles está en activos financieros,» 6 Junio 2021. [En línea]. Available: https://www.eleconomista.es/mercados-cotizaciones/noticias/11255132/06/21/Solo-el-22-de-toda-la-riqueza-de-los-espanoles-esta-en-activos-financieros.html.

[2] «El nivel de cultura financiera en España se sitúa por debajo de la media europea,» 5 Octubre 2020. [En línea]. Available: https://www.eleconomista.es/ecoaula/noticias/10808028/10/20/El-nivel-de-cultura-financiera-en-Espana-se-situa-por-debajo-de-la-media-europea.html.

[3] «Savills,» [En línea]. Available: https://www.savills.com/impacts/market-trends/the-total-value-of-global-real-estate.html.

[4] H. Selim, «Determinants of house prices in Turkey: Hedonic regression versus artificial neural network,» *Expert Systems with Applications,* vol. 36, nº 6, pp. 2843-2852.

[5] J. K. B. Byeonghwa Park, «Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data,» *Expert Systems with Applications,* vol. 42, nº 6, pp. 2928-2934, 2015.

[6] C. F. O. L. Z. y. S.-C. H. Jieh-Haur Chen, «Forecasting spatial dynamics of the housing market using Support Vector Machine,» *International journal of strategic property management,* vol. 21, pp. 273-283, 2017.

[7] C. Haejung, «Prediction of Housing Price Using Time Series Analysis and Machine Learning Methods,» *Journal of The Residential Environment Institute of Korea,* vol. 18, nº 1, pp. 49-65, 2020.

[8] J. Mendoza, «Estadisticamente,» 10 Noviembre 2021. [En línea]. Available: https://estadisticamente.com/shapley-values-o-valor-de-shapley-teoria-de-juegos/.

[9] «Help my Cash,» [En línea]. Available: https://www.helpmycash.com/blog/los-5-portales-inmobiliarios-mas-utilizados-en-espana-en-2020/.

[10] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Web_scraping.

[11] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Instituto_Nacional_de_Estad%C3%ADstica_(Espa%C3%B1a).

[12] INE, «Índice de Precios de Vivienda,» [En línea]. Available: https://www.ine.es/jaxiT3/Tabla.htm?t=25171.

[13] INE, «Población residente por fecha, sexo y edad,» [En línea]. Available: https://www.ine.es/jaxiT3/Tabla.htm?t=31304.

[14] INE, «Estadística de Transmisiones de Derechos de la Propiedad,» [En línea]. Available: https://www.ine.es/jaxiT3/Tabla.htm?t=6150.

[15] «IBM,» 15 Julio 2020. [En línea]. Available: https://www.ibm.com/es-es/cloud/learn/machine-learning.

[16] B. d. l. I. C. H. &. V. R.-S. J.C.W. Debuse, «Building the KDD Roadmap.,» de *Industrial Knowledge Management*, London, Springer, 201, p. 179–196.

[17] «Sngular,» [En línea]. Available: https://www.sngular.com/es/data-science-crisp-dm-metodologia/.

[18] [En línea]. Available: https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/.

[19] «javatpoint,» [En línea]. Available: https://www.javatpoint.com/types-of-machine-learning.

[20] T. G. J. L. &. R. A. J. Jiang, «Supervised Machine Learning: A Brief Primer.,» *Behavior therapy,*, pp. 675-687, 2020.

[21] H. Geng, H. Liu, B. Wang y F. Sun, «Reinforcement Extreme Learning Machine for Mobile Robot Navigation,» *Proceedings in Adaptation Learning and Optimization,* vol. 9, pp. 61-73, 2018.

[22] «IBM,» [En línea]. Available: https://www.ibm.com/topics/linear-regression.

[23] «Polígonos de Thiessen,» [En línea]. Available: https://es.wikipedia.org/wiki/Pol%C3%ADgonos_de_Thiessen.

[24] S. Rao, «Towards Data Science,» 24 Enero 2022. [En línea]. Available: https://towardsdatascience.com/k-means-clustering-explain-it-to-me-like-im-10-e0badf10734a.

[25] Z. Jaadi, «builtin,» 8 Agosto 2022. [En línea]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[26] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/An%C3%A1lisis_de_componentes_principales.

[27] A. Agarwal, «towardsdatascience,» 5 Octubre 2018. [En línea]. Available: https://towardsdatascience.com/linear-regression-using-python-b136c91bf0a2.

[28] T. Yiu, «towardsdatascience,» 19 Junio 2019. [En línea]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[29] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Random_forest.

[30] J. A. Camacho, «jacobsoft,» 27 Marzo 2027. [En línea]. Available: https://www.jacobsoft.com.mx/es_mx/support-vector-regression/#:~:text=Regresi%C3%B3n%20de%20Soporte%20Vectorial%20(Support%20Vector%20Regression%20%E2%80%93%20SVR).

[31] G. M. K, «Towards data science,» [En línea]. Available: https://towardsdatascience.com/machine-learning-basics-support-vector-regression-

660306ac5226#:~:text=Overview%20of%20SVR&text=Support%20Vector%20Regression%20is%20similar,is%20referred%20to%20as%20hyperplane..

[32] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Gradient_boosting.

[33] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/XGBoost.

[34] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Red_neuronal_artificial.

[35] A. Obuchowski, «www.becominghuman.ai,» [En línea]. Available: https://becominghuman.ai/understanding-neural-networks-2-the-math-of-neural-networks-in-3-equations-6085fd3f09df.

[36] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Error_cuadr%C3%A1tico_medio.

[37] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Ra%C3%ADz_del_error_cuadr%C3%A1tico_medio.

[38] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Error_absoluto_medio.

[39] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Coeficiente_de_determinaci%C3%B3n.

[40] «Medium,» 2 Julio 2020. [En línea]. Available: https://medium.com/@itbodhi/overfitting-and-underfitting-in-machine-learning-models-76cb60dbdaf6.

[41] «www.aws.amazon.com,» [En línea]. Available: https://aws.amazon.com/es/what-is/python/.

[42] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/NumPy.

[43] A. R. Mesa, «openwebinars.net,» [En línea]. Available: https://openwebinars.net/blog/por-que-usar-numpy/#:~:text=Caracter%C3%ADsticas%20principales%20de%20NumPy&text=Incluye%

20funciones%20para%20operaciones%20de,en%20conjunto%20son%20muy%20potentes..

[44] «WikiPedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Pandas_(software).

[45] [En línea]. Available: https://seaborn.pydata.org/.

[46] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Scikit-learn.

[47] «ScikitLearn,» [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[48] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Gradient_boosting.

[49] «WikiPedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Inteligencia_artificial_explicable.

[50] C. Molnar, Interpretability Machine Learning.

[51] «Towards Data Science,» [En línea]. Available: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137.

[52] «SHAP library,» [En línea]. Available: https://shap.readthedocs.io/en/latest/.