FID - Memoria del proyecto

Índice

Planteamiento

Contexto

En el contexto de la asignatura se plantea la implementación de una tarea de regresión. El **equipo 5** está integrado por:

- José Enrique Núñez García
- José Manuel Muñiz Peña
- Pedro Guembe Fernández
- Julia García Gallego

Este proyecto se ha considerado como una oportunidad de oro para plasmar los conocimientos aprendidos sobre Data Science. Se ha elegido la tarea de regresión por ser una de las más básicas (frente a otras de aprendizaje no supervisado), y para la que se presenta gran variedad de alternativas en cuanto a modelos y técnicas.

Dataset

El dataset empleado es ***House Prices - Advanced Regression Techniques*** (https://www.kaggle.com/c/house-prices-advanced-regression-techniques), un dataset de regresión para predicción de precios de viviendas.

Su elección viene determinada por su popularidad, la que se ha considerado sinónimo de calidad y de confianza. Además, supone un extra de motivación el hecho de que sea una competición abierta de Kaggle, y podamos subir los resultados obtenidos para compararlos con los del resto de la comunidad.

Dado que el dataset no es enorme (consta de algo más de 1400 filas), uno de los factores más importantes es el de decidir qué columnas son relevantes. El archivo original tiene más de 80 columnas, lo que supone un problema si no se realizara un minucioso preprocesamiento.

Organización

El proyecto se ha organizado mediante la herramienta Trello. Mediante tarjetas se han identificado pequeñas tareas que se iban realizando de manera individual. La gestión e integración del código se ha realizado mediante GitHub. El código y el dataset se encuentran disponibles en el siguiente enlace: https://github.com/josmunpen/house-price-regression.

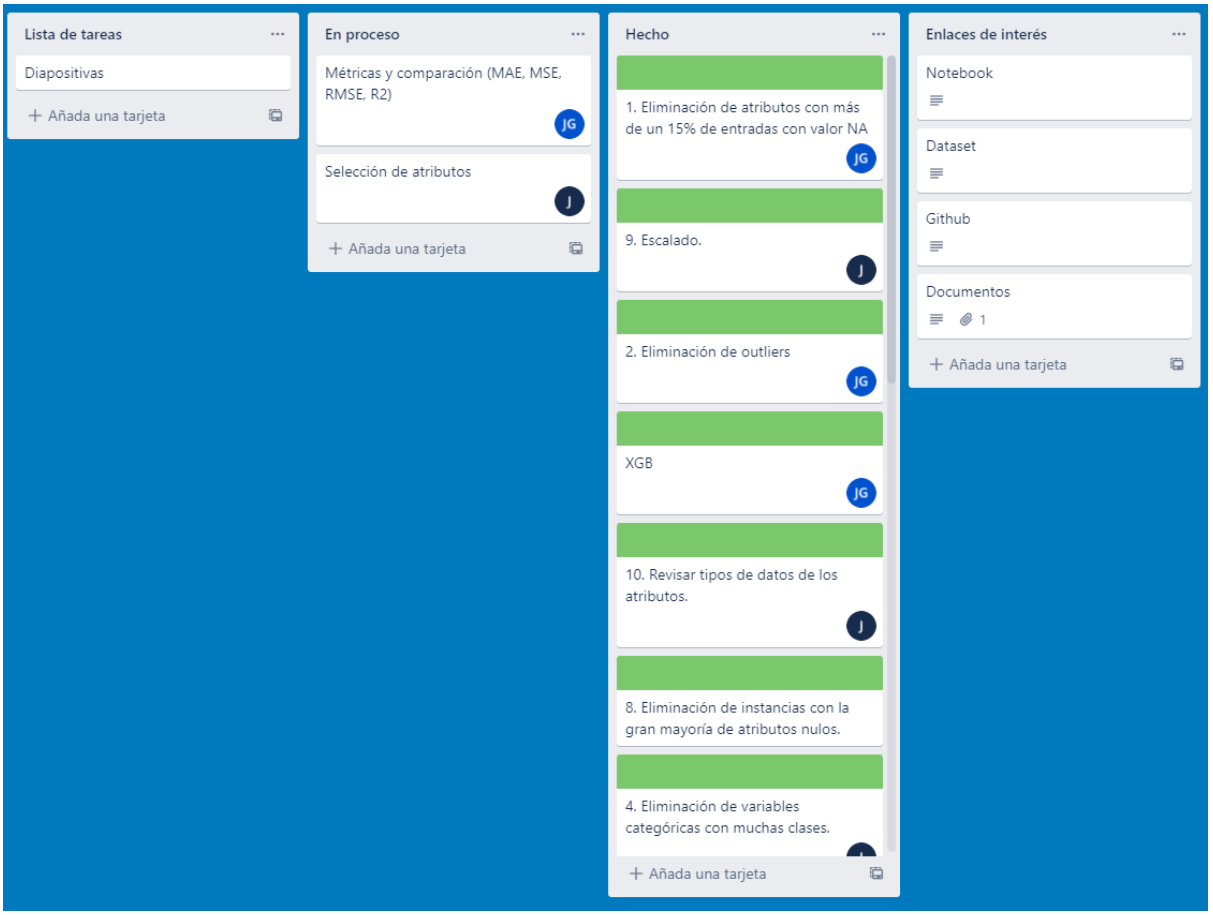

Figura 1. Gestión del proyecto en la herramienta Trello.

Procedimiento

Importación y carga de paquetes

Antes de comenzar, se realiza la carga de todas las librerías de R que se emplearán posteriormente. En este paso, también, se define una semilla, para facilitar la reproducibilidad de los resultados.

Visualización de los datos

Lo primero que se realiza tras cargar los datos es su visualización.Se van a observar las primeras entradas de nuestro conjunto de datos para conocer un poco más de información sobre el dataset.
Podemos ver que cuenta con un campo “Id” y una gran cantidad de atributos tanto categóricos como numéricos. Se puede observar también que existen instancias con atributos con valores nulos (“NA”, Not Avaliable).

Id <dbl>	MSSubClass <dbl>	MSZoning <chr>	LotFrontage <dbl>	LotArea <dbl>	Street <chr>	Alley <chr>	LotShape <chr>	LandContour <chr>	Utilities <chr>
1	60	RL	65	8450	Pave	*NA*	Reg	Lvl	AllPub
2	20	RL	80	9600	Pave	*NA*	Reg	Lvl	AllPub
3	60	RL	68	11250	Pave	*NA*	IR1	Lvl	AllPub
4	70	RL	60	9550	Pave	*NA*	IR1	Lvl	AllPub
5	60	RL	84	14260	Pave	*NA*	IR1	Lvl	AllPub
6	50	RL	85	14115	Pave	*NA*	IR1	Lvl	AllPub

LotConfig <chr>	LandSlope <chr>	Neighborhood <chr>	Condition1 <chr>	Condition2 <chr>	BldgType <chr>	HouseStyle <chr>	OverallQual <dbl>
Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7
FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	6
Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7
Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7
FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	8
Inside	Gtl	Mitchel	Norm	Norm	1Fam	1.5Fin	5

Figura 2. Muestra del dataset

Algunos valores interesantes del atributo objetivo “SalePrice” (atributo que se intentará predecir) son los siguientes:

Valor mínimo: 34900
Valor máximo: 755000
Valor medio: 190921
Primer cuartil: 129975
Tercer cuartil: 214000

El conjunto de entrenamiento cuenta con 1460 filas y 81 columnas y el conjunto de test cuenta con 1459 filas y 80 columnas.

Procedemos ahora a ver cómo se distribuyen los precios de las casas en nuestro dataset

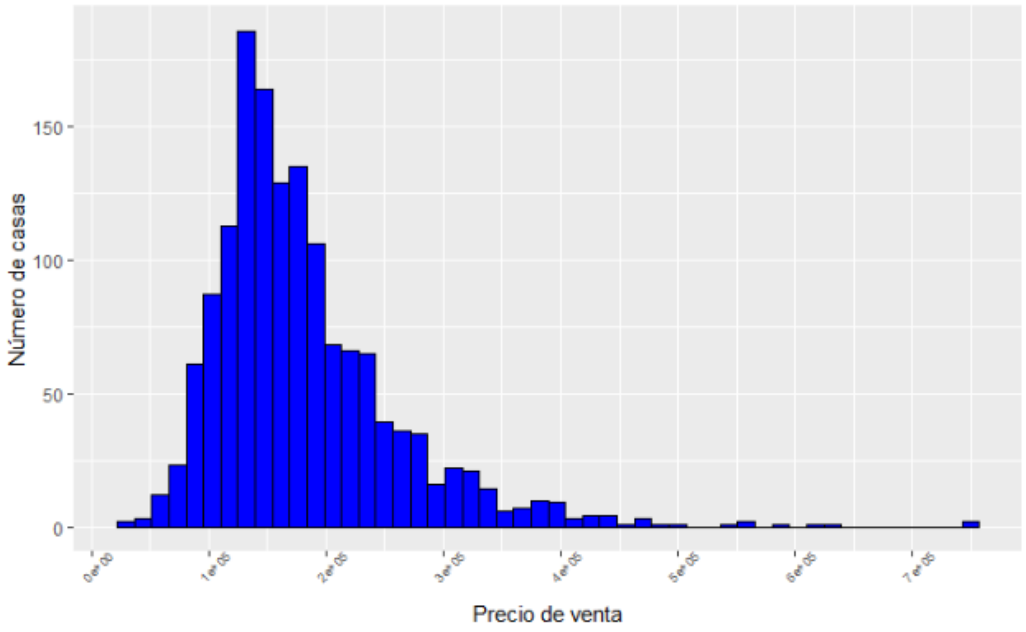

Figura 3. Distribución de variable precio de venta

Se observa que el rango de precios se mueve principalmente entre los valores 100.000 y 200.000.

Ahora se observan los atributos que contienen valores NA, los cuales se tratarán a continuación.

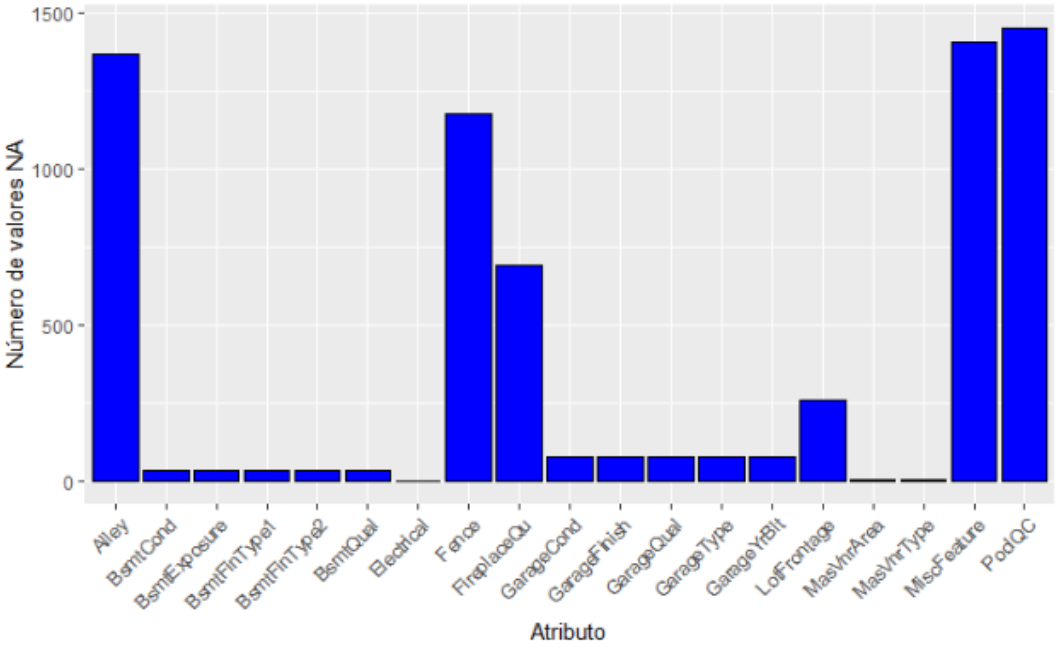

Figura 4. Valores "NA" por columna.

Se puede comprobar que existen 19 atributos que contienen valores "NA", 4 de ellos superando la cantidad de 1000 instancias.

Preprocesamiento

Valores “NA”

Vamos a observar más detenidamente la cantidad de valores “NA” que contiene cada atributo del conjunto de datos.

key <chr>	num.missing <int>
GarageYrBlt	81
BsmtExposure	38
BsmtFinType2	38
BsmtCond	37
BsmtFinType1	37
BsmtQual	37
MasVnrArea	8
MasVnrType	8
Electrical	1

key <chr>	num.missing <int>
PoolQC	1453
MiscFeature	1406
Alley	1369
Fence	1179
FireplaceQu	690
LotFrontage	259
GarageCond	81
GarageFinish	81
GarageQual	81
GarageType	81

Figura 5 Número de valores ausentes por columna.

El primer objetivo será eliminar aquellos atributos que contengan más de un 10% de sus entradas con valor “NA”. Estas columnas son: “LotFrontage”, “Alley”, “FireplaceQu”, “PoolQC”, “Fence” y “MiscFeature”.

Para el resto de columnas que tienen valores “NA”, pero que éstos no suponen más de un 10% del total, se tratarán de diferente manera. En los casos de variables categóricas, se sustituirán dichos valores por el valor más frecuente; mientras que en el de las variables numéricas, se sustituirán por la media.

Ahora el conjunto de datos estará completamente libre de valores “NA”.

Eliminación de variables no relevantes

Llegados a este punto, sigue habiendo muchas columnas para el número de filas con el que se está trabajando. Para lidiar con esto se plantea la eliminación de aquellas columnas numéricas que están poco correlacionadas con la variable objetivo.

Para ello se seleccionan las columnas numéricas y se obtiene la matriz de correlación enfrentando todas estas a la variable objetivo (precio de venta de la vivienda).

Llegados a este punto se debe elegir un límite a partir del cual descartar variables. En este caso se ha definido en 0.2, quedando las variables en rojo de la *Figura 5* descartadas.

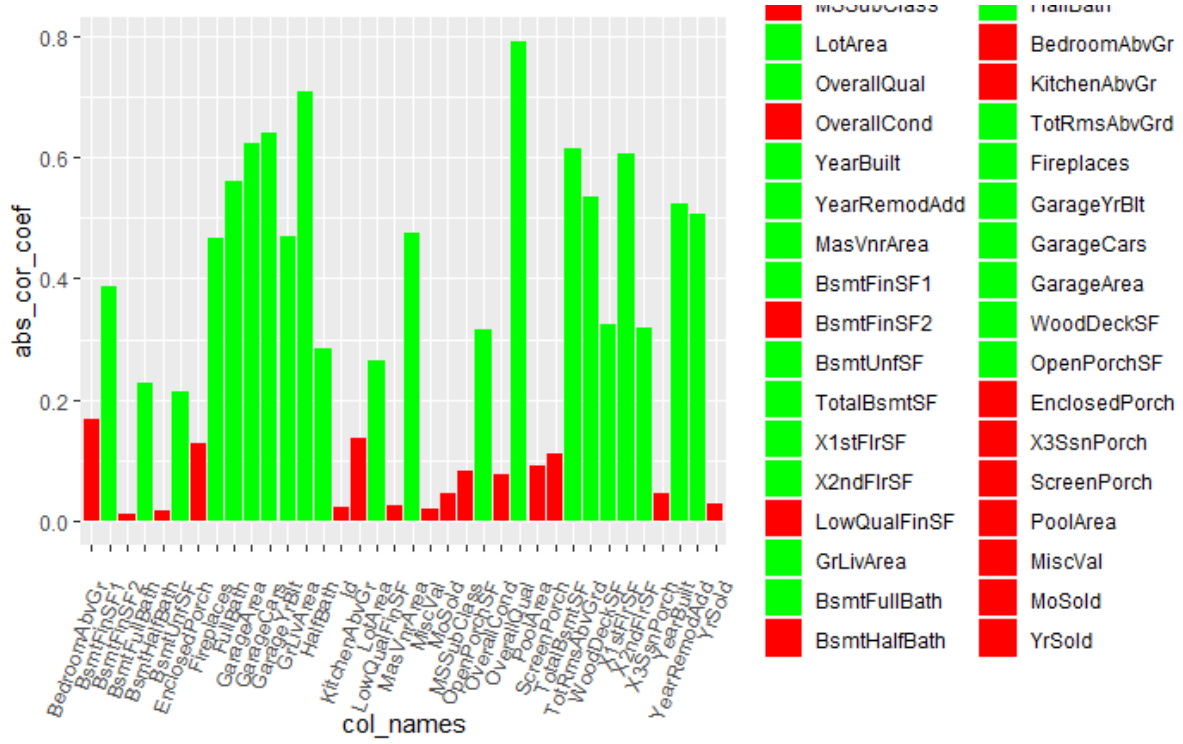

Figura 6. Selección de variables en función de su coeficiente de correlación. En rojo se muestran las variables descartadas y en verde las seleccionadas.

Eliminación de variables categóricas con demasiados valores

Con el fin de continuar reduciendo la dimensionalidad también se eliminarán aquellas variables categóricas que tengan demasiadas categorías o valores. Cuando se realice la codificación posterior de las variables categóricas, aparecerán tantas columnas como valores diferentes se encuentren, por lo que si no se controla ésto se podrían plantear demasiadas columnas.

En la *Figura 6* se muestran las variables eliminadas. En este caso se ha establecido el límite en 8 valores únicos, seleccionando sólo las columnas que tienen menos de dicho número.

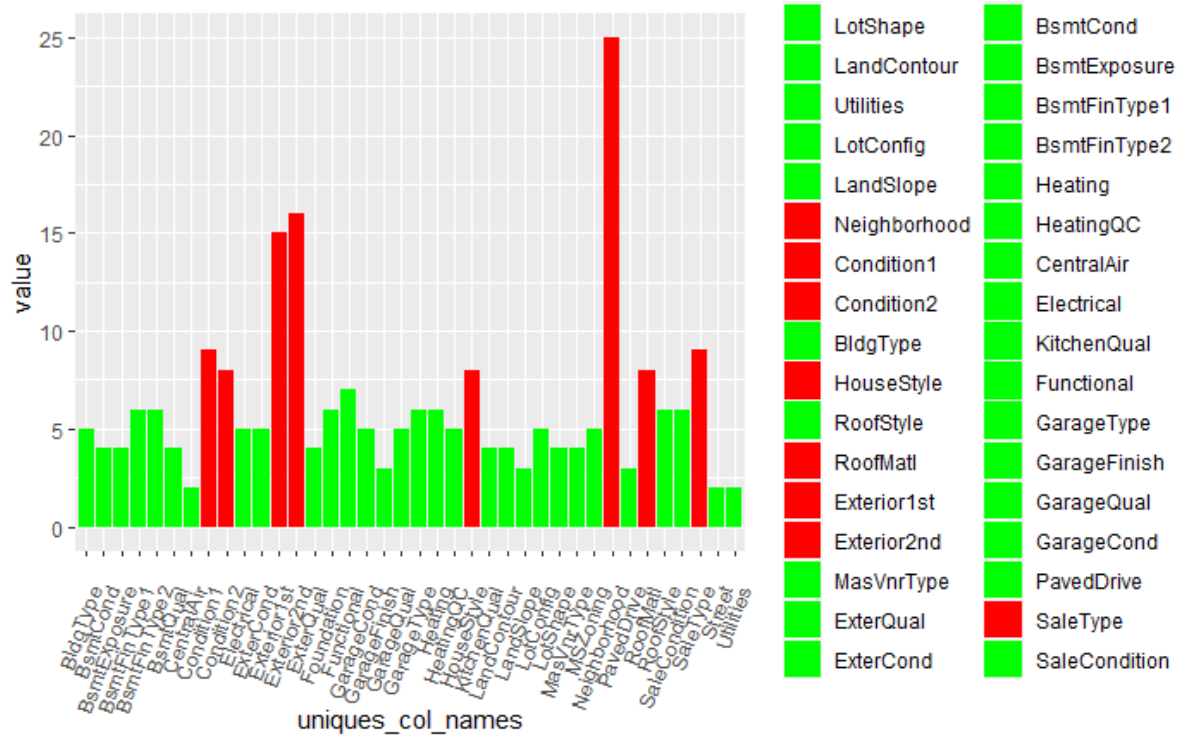

Figura 7. Selección de variables categóricas en función del número de valores únicos. En rojo se muestran las variables descartadas y en verde las seleccionadas.

Codificación

Dado que todas las entradas a los modelos deben ser números, se codifican las columnas categóricas. Se ha optado por One Hot Encoding, de manera que si una columna tiene *n* valores posibles, se generan *n* columnas. El valor de las columnas es 0, excepto para el de la columna asociada al valor de esa instancia.

Escalado

Algunos algoritmos, como las redes neuronales o las máquinas de vectores de soporte requieren escalado de las variables numéricas. Si no se realizara escalado, el algoritmo entendería que las variables cuyo valor suele ser alto tienen mucho más peso que otras. Por ejemplo, el número de cuartos de baño suele ser 1, 2 o 3. Sin embargo, el número de metros de la vivienda son valores mucho más altos, y no por ello se debería considerar esta variable como más relevante. Tras el escalado, los valores de las variables se encuentran alrededor de 0, principalmente en la horquilla [-2,2].

Reducción de dimensionalidad.

Como se han creado muchas columnas en la codificación One Hot Encoding, se procede a reducir el número de las mismas mediante el algoritmo RFE (Random Feature Elimination). Para ello se han realizado varias iteraciones, estudiando con diferentes números de columnas. En concreto: 80, 100, 120 y 140, como se puede observar en la *Figura 7*. Tras este paso, el dataset está listo para el entrenamiento y presenta [100] columnas.

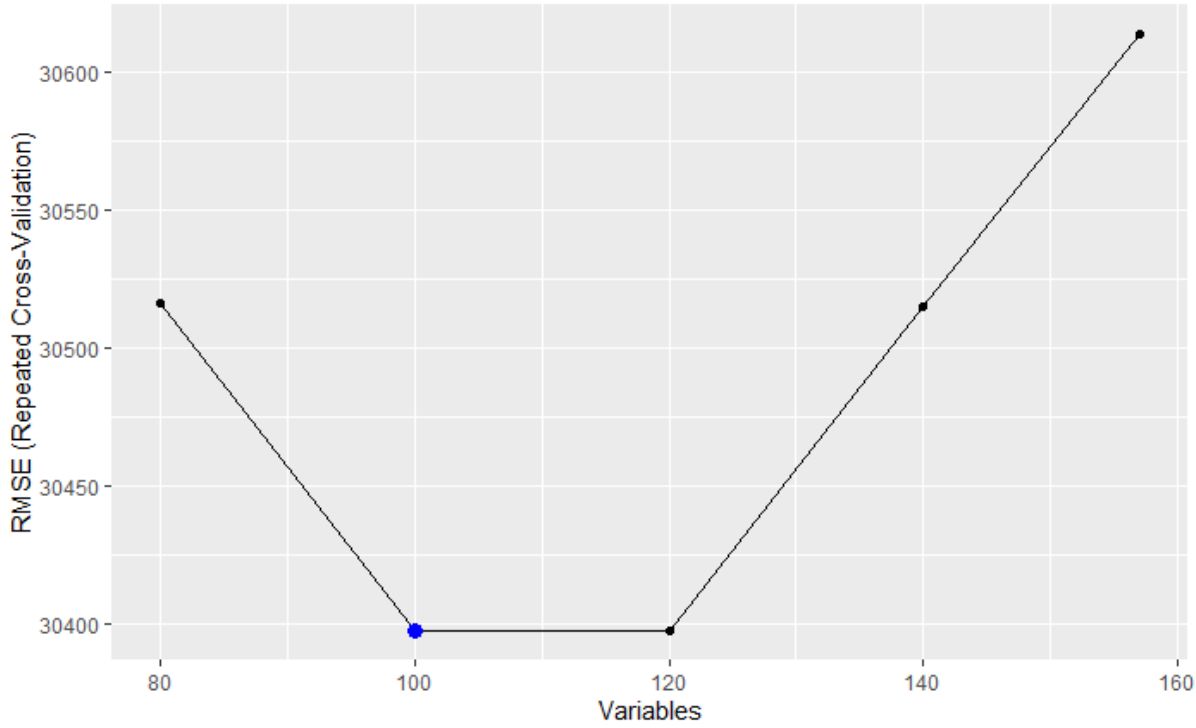

Figura 8. Valores obtenidos para la métrica RMSE probando con diferentes números de variables.

Entrenamiento

A la hora de implementar una tarea de regresión existe gran variedad de alternativas en cuanto a qué modelos emplear. Se han elegido los modelos de Extreme Gradient Boosting, Support Vector Machine y un regresor de Random Forest. Los factores para la elección de estos han sido, principalmente, su popularidad, pero también su eficiencia computacional.

Extreme Gradient Boosting

Para nuestra tarea de entrenamiento usaremos el algoritmo Extreme Gradient Boosting, se trata de unos de los algoritmos más utilizados en la plataforma Kaggle, debido a su increíble funcionamiento y velocidad de entrenamiento. Probaremos el algoritmo con 1000, 2000 y 3000 iteraciones y con unas tasas de aprendizaje de 0.1, 0.01 y 0.001. La profundidad del algoritmo será 5 (profundidad que alcanzarán los árboles de regresión)
Observaremos qué parámetros nos devuelven mejores resultados haciendo uso de las métricas RMSLE, R^2 y MAE.

Support Vector Machine (SVM)

Las máquinas de vectores de soporte (SVM) se han convertido en uno de los modelos más versátiles, tanto en regresión como en clasificación. Era, por lo tanto, una opción muy clara para esta tarea, pese a requerir alguna condición especial como el escalado. En concreto,

se ha realizado con las variantes lineal y radial. Para medir su desempeño se ha empleado validación cruzada con 3 repeticiones, partiendo los datos en 10 partes cada vez.

Random Forest

Los Random Forest son uno de los métodos más utilizados hoy en día en clasificación por su sencillez, ya que apenas necesita preprocesado de datos. Para compararlo con el resto de modelos le hemos proporcionado el mismo conjunto de datos preprocesado, aunque no habría hecho falta hacer tantos pasos y, de hecho, apenas es utilizado en problemas de regresión como este ya que tiene limitaciones como que no puede predecir más allá del rango de valores del conjunto de entrenamiento y que, por lo general, es poco interpretable. Por ello hemos decidido introducirlo a modo de curiosidad y experimentación y entrenarlo con una única validación cruzada de 10 particiones.

Evaluación

En la siguiente tabla se recogen los resultados obtenidos con los diferentes modelos.

	Random Forest	Extreme Gradient Boosting	Support Vector Machine
RMSE	27907.64	28343.97	29179.64
Rsquared	0.8789662	0.8691729	0.8661108
MAE	16719.93	17568.50	17889.18

Destacar que se ha escogido la mejor configuración para cada modelo después de haber probado estos con diferentes parámetros. Los resultados que se muestran en la tabla son los mejores que cada modelo ha entregado. Ejecutando el código se podrá observar los diferentes resultados que arroja cada modelo cambiando sus respectivos parámetros.

Como se puede observar, Random Forest ha conseguido los mejores resultados.

Conclusiones

Una vez finalizado el proyecto nos resulta de especial interés extraer conclusiones sobre lo aprendido.

Por un lado, estamos orgullosos de haber implementado una tarea común y obtener unos resultados que consideramos más que buenos. Trabajar en equipo ha sido muy positivo, puesto que la ayuda entre todos los miembros ha sido crucial en muchas ocasiones.

Por otro lado, hemos aprendido a enfrentarnos a un problema desde inicio hasta el fin. Uno de los aspectos que destacaríamos es que lo que más nos ha frenado ha sido lidiar con el lenguaje de programación R, con el que los miembros del equipo no tenían casi experiencia. Sin embargo, una vez resolvíamos dichas dificultades, no ha resultado excesivamente difícil llevar a la práctica los fundamentos teóricos que hemos aprendido.