



Universidad  
Europea  
LAUREATE INTERNATIONAL UNIVERSITIES

# Universidad Europea de Madrid

Escuela de Arquitectura, Ingeniería y Diseño

## MODELO DE PREDICCIÓN DEL PRECIO DE LAS CASAS EN IOWA.

Proyecto Open Data II



### House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting  
5,318 teams · Ongoing

**Profesor:** Rafael Muñoz Gil

**Integrantes:**

Pablo Neira Voces  
Alejandro Paul Schmid Hernández  
Marcos Carbonero García



## ÍNDICE

<b>1. OBJETIVO.....</b>	<b>2</b>
<b>2. ALGORITMO.....</b>	<b>2</b>
<b>3. HYPER-TUNNING DE PARÁMETROS .....</b>	<b>3</b>
<b>4. EXTRACCIÓN DE CARACTERÍSTICAS .....</b>	<b>4</b>
<b>5. TEST Y CARGA A KAGGLE .....</b>	<b>5</b>
<b>6. CONCLUSIONES.....</b>	<b>6</b>



## 1. OBJETIVO

Escogimos el DataSet *HousePrices* con el objetivo de poder llegar a realizar y diseñar por nosotros mismos un modelo de predicción que pudiera llegar a calcular el precio de las casas de la zona de Iowa en EE. UU., aceptando así el reto de Kaggle.

## 2. ALGORITMO

Teníamos muy claro que el tipo de algoritmo que debíamos usar era la regresión, ya que nuestra variable objetivo (SalePrice) es continua, por lo que no nos planteamos el hecho de usar un algoritmo de tipo clasificación. Una vez que sabíamos el tipo que debíamos usar, era amplio el abanico de algoritmos que podíamos utilizar:

- DecisionTreeRegressor.
- GBRegressor.
- LinearRegression.
- RandomForestRegressor.

Empezamos realizando una tabla de correlación de las variables con nuestra variable objetivo para ver las que tenían una mayor correlación con el precio de venta y una menor correlación entre ellas. De esta manera conseguimos las variables que íbamos a meter en nuestros primeros modelos a modo de prueba. Las variables escogidas fueron: "YearBuilt" (Año de construcción), "TotalBsmtSF" (metros cuadrados totales), "OverallQual" (calidad de los materiales), "GrLivArea" (superficie por encima del nivel del suelo habitable) y "GarageArea" (metros cuadrados del garaje).

Con estas variables probamos inicialmente con el algoritmo LinearRegression. A decir verdad, obtuvimos unas predicciones aceptables para tratarse de nuestro primer modelo, obteniendo un error cuadrático medio de 35619, algo que nos pareció bastante elevado, por lo que decidimos pasar a probar otro algoritmo.

Probamos con el algoritmo RandomForestRegressor. Obtuvimos una mejoría significativa de nuestros resultados, dejando el error cuadrático medio en 28265, seguía pareciéndonos bastante elevado, pero teníamos también que recordar que no habíamos realizado aún ningún tipo de extracción de características ni de optimización de parámetros. De todas formas, decidimos hacer una última prueba.

Probamos finalmente con el algoritmo GradientBoostedTreeRegressor. Las predicciones obtenidas con este algoritmo mejoraron significativamente,



llegando a acercarse al precio real en muchas de las casas (recordemos que empezamos entrenando nuestro modelo con un aprendizaje supervisado). El error cuadrático medio se estableció en 23666, seguía siendo elevado, pero tampoco podíamos esperar que ese fuera el modelo definitivo.

Así, decidimos apostar por el algoritmo GBTRRegressor, además observando la página de nuestro reto en Kaggle pudimos ver después que era uno de los más utilizados por el resto de los participantes.

### 3. HYPER-TUNNING DE PARÁMETROS

Así, una vez escogido nuestro algoritmo comenzamos con el proceso de hyper-tunning de los parámetros.

Inicialmente, para realizar la prueba simplemente le introducimos el parámetro “maxIter” con un valor de 10.

Para obtener los mejores valores de nuestros parámetros utilizamos el GridSearch, que recordamos que era un algoritmo que itera a través de la lista de valores de los parámetros, estima los modelos de manera independiente y selecciona la mejor opción.

Los parámetros que introducimos en el Grid ya que influyen en el GBTRRegressor y que quisimos obtener sus mejores valores “MaxDepth”, “MaxBins” y “MaxIter”. Previamente realizamos una pequeña lista de valores posibles y fuimos probando con un número reducido de ellos ya que probando con muchos valores podría desencadenar un tiempo de evaluación demasiado alto.

Definitivamente, tras un par de par de pruebas, los valores que obtuvimos a través del grid que pudimos saber que iban a ser los más óptimos fueron: “MaxDepth:”, “MaxBins: ” y “MaxIter: ”.

Una vez obtenidos los mejores valores para los parámetros decidimos pasar a la extracción de características para obtener las variables que más influyen en nuestra variable objetivo.



## 4. EXTRACCIÓN DE CARACTERÍSTICAS

Como afirmamos al inicio, introducimos unas variables según las correlaciones con nuestra variable objetivo, por lo que decidimos realizar la extracción de características para determinar realmente las variables más influyentes a la hora de predecir el precio de venta.

Nosotros contábamos con un dataset con 80 variables, por lo que decidimos comenzar con una previa selección de variables que introducirle al “vector assembler” que íbamos a usar en la extracción de características.

Comenzamos probando con nuestras seis variables iniciales a modo de prueba del algoritmo. Seguidamente, le introducimos al rededor de las 80 variables que teníamos y fuimos realizando pruebas variando el “numTopFeatures”, y introduciéndolas en nuestro modelo.

Comenzamos con un “numTopFeatures” bastante elevado y lo fuimos disminuyendo y probando las variables que obteníamos en nuestro modelo. Así continuamente hasta que la reducción de variables nos daba unas predicciones peores. De esta manera seleccionamos al rededor de treinta variables.

Ahora sí teníamos un modelo óptimo, con las variables con mayor influencia sobre nuestra variable objetivo y con los mejores parámetros para nuestro modelo. Nuestro error cuadrático medio lo acabábamos de reducir a 5000, una cifra ya bastante buena.

Estas fueron nuestras predicciones obtenidas con aprendizaje supervisado, como vemos, nuestra columna “prediction” se acerca bastante a la columna original



prediction	SalePrice	features
206265.86700477925	208500.0	[856.0,0.0,60.0,8...
180646.7631069654	181500.0	[1262.0,0.0,20.0,...
212033.33880225284	223500.0	[920.0,0.0,60.0,1...
143169.6361796585	140000.0	[756.0,0.0,70.0,9...
143963.95522178934	143000.0	[796.0,0.0,50.0,1...
299886.4911716766	307000.0	[1686.0,0.0,20.0,...
207153.03313951037	200000.0	[1107.0,0.0,60.0,...
134416.82757872323	129900.0	[952.0,0.0,50.0,6...
120358.9633280359	118000.0	[991.0,0.0,190.0,...
344732.28450423875	345000.0	[1175.0,0.0,60.0,...
139586.95799702543	144000.0	[912.0,0.0,20.0,1...
263737.5361947459	279500.0	(34,[0,2,3,4,5,6,...
154099.4593960667	157000.0	[1253.0,0.0,20.0,...
146161.84363034333	149000.0	[1004.0,0.0,20.0,...
94566.8029954398	90000.0	(34,[2,3,4,5,6,7,...
164343.38896668865	159000.0	[1114.0,0.0,20.0,...
134971.79935554415	139000.0	(34,[0,2,3,4,5,6,...
325424.6110815337	325300.0	[1158.0,0.0,60.0,...
138886.75539324587	139400.0	(34,[0,2,3,4,5,6,...
232253.1698544703	230000.0	(34,[0,2,3,4,5,6,...

only showing top 20 rows

## 5. TEST Y CARGA A KAGGLE

Una vez teníamos nuestro modelo realizado debíamos bajarnos el dataset de test de Kaggle, el cuál introducía unas nuevas 1460 casas con las mismas características. A este dataset le faltaba la columna "SalePrice" y ahí entraba nuestro trabajo de intentar predecir los valores de esta columna.

Importamos el dataset de test y le aplicamos nuestro modelo, obteniendo las siguientes predicciones:

Id	SalePrice
1461	122242.384649699
1462	162897.36956204264
1463	208399.19712976247
1464	191965.44692184945
1465	208018.16907134384
1466	180836.23746975974
1467	171808.25427882426
1468	171819.28103111722
1469	170633.7393687073
1470	105454.7528936456
1471	202055.5663265965
1472	102195.81270764845
1473	103212.09787314555
1474	157284.40720038384
1475	119631.15596478153
1476	422564.70282832143
1477	250571.67444021397
1478	331687.4611268366
1479	223985.44850914052
1480	374950.7641356278

only showing top 20 rows





Una vez habíamos conseguido predecir el precio de venta de las nuevas casas solo nos quedaba exportar la tabla en un archivo csv, subirlo a Kaggle y observar en que posición nos habían situado:

CasasUEM.csv

an hour ago by APMUEM

0.17461

Predicción del precio de las casas mediante el algoritmo GBRegressor. Estudiantes de la Universidad Europea de Madrid (Alex Schmid, Marcos Carbonero y Pablo Neira).

4044	APMUEM		0.17461	2	1m
<b>Your Best Entry</b> ↑					
You advanced 303 places on the leaderboard!					
Your submission scored 0.17461, which is an improvement of your previous score of 0.20784. Great job!					
					Tweet this!

Como podemos ver, Kaggle nos ha situado en la posición 4044 con una puntuación de 0,17461, el cuál creemos que es el fallo que hemos tenido con respecto a los valores reales. En este reto participaban al rededor de 7000 personas.

## 6. CONCLUSIONES

Finalmente terminamos nuestro modelo, lo conseguimos, y esta es la mejor conclusión que hemos sacado.

Escogimos este dataset al inicio del curso con el objetivo de poder llegar a predecir los precios de venta de las casas y lo hemos conseguido.

Hemos podido aprender y, especialmente, hemos podido darnos cuenta de nuestro progreso ya que como hemos dicho, escogimos este dataset con un objetivo que veíamos muy lejano y sin embargo, a pesar de que nuestra posición en Kaggle no ha sido la mejor aunque tampoco la peor, estamos muy contentos por el hecho de haber conseguido realizar un modelo de predicción.

Nos llevamos conclusiones satisfactorias de esta asignatura, la cuál pensamos además que nos va a ser de gran utilidad a la hora de nuestra futura profesión.