DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Predicción de precios de inmuebles

Métodos Numéricos

Integrante	LU	Correo electrónico
Zolezzi, Maria Victoria	222/19	zolezzivic@gmail.com
Miceli, Juan Pablo	424/19	micelijuanpablo@gmail.com
Lavalle Cobo, Ignacio	282/19	ignacio.lavalle.cobo@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Índice

1. Resumen

En este trabajo utilizaremos el método de cuadrados mínimos lineales (CML) para predecir los precios y la cantidad de metros cubiertos de los inmuebles de una base de avisos inmobiliarios de México. Buscaremos procesar estos datos de diferentes formas para minimizar el error de las predicciones.

2. Glosario

Keywords:
Cuadrados Mínimos Lineales,
Feature engineering,
Error cuadrático medio,
Housing prices

3. Introducción

Nuestro objetivo es crear un modelo capaz de predecir los precios y la cantidad de metros cubiertes de nuevos inmuebles basándonos en ciertas características como por ejemplo su ubicación, la cantidad de habitaciones que poseen o si tienen o no piscina, de la manera más fiable posible. Para esto, utilizaremos el método de cuadrados mínimos lineales para obtener un predictor apartir de la combinación de estas categorías.
La experimentación a llevar a cabo se basará en procesar los datos de entrada de distintas formas buscando siempre una mejora en el error obtenido con nuestro regresor. Para lograr este objetivo segmentaremos el dataset, realizaremos feature engineering y buscaremos una manera óptima de lidiar con los NaNs.

4. Dataset a analizar

En esta oportunidad utilizaremos el set de datos *Precios inmobiliarios*. El mismo está compuesto por 240000 avisos que cuentan con 22 categorías cada uno.
Es importante destacar que los avisos de training vienen provistos con los precios. Por otra parte, no todos las entradas poseen el campo "metros cubiertos" completo, por lo que eliminaremos estos datos de la base cuando queramos predecir esta característica .

5. Cuadrados Mínimos Lineales

CML es un método de optimización que busca encontrar la función que mejor aproxima a un conjunto de datos.
El problema de cuadrados mínimos se formula como

$$\min_{x \in \mathbb{R}^n} ||Ax - b||_2^2$$

donde las columnas de la matriz A están conformadas por las funciones a combinar evaluadas en los puntos de entrada, el vector x contiene los coeficientes que acompañarán a cada término en la combinación lineal buscada, y por último, el vector b contiene los valores de la característica a predecir.

Para resolver esta minimización utilizaremos las ecuaciones normales, quedandónos

$$A^T Ax = A^T b,$$

el sistema equivalente a resolver.

Una cuestión importante a destacar es que, por lo visto en la teórica, el problema de cuadrados mínimos siempre tiene solución.

6. Cálculo del error

Para medir la performance de nuestro predictor utilizaremos el error cuadrático medio, cuya fórmula es la siguiente:

$$RMSE(\hat{f}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2}$$

donde y_i es el valor real a predecir e $\hat{y_i}$ el valor arrojado por el predictor $\hat{f}(x_i)$. N representa la cantidad de datos con los que se cuenta.
Una desventaja de esta métrica es que ponderá más los errores en valores altos de y, antes que los bajos. En nuestro caso esto es un problema ya que queremos que todos los errores se ponderen de la misma forma.
Es por esto, que también consideraremos el error cuadrático medio logarítmico, cuya fórmula es:

$$RMSLE(\hat{f}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y_i} + 1))^2}$$

De todas formas, esta métrica tampoco es perfecta, ya que penaliza en mayor medida predecir un precio menor al real, que el caso contrario. Otra contra que tiene es que no soporta valores negativos de las predicciones (sabemos que en nuestro caso no tienen sentido, pero en ciertas ocasiones nuestro predictor los ha arrojado). Esto no permite aplicarlo en muchas oportunidades dejándonos dos opciones posibles: utilizar únicamente el RMSE o eliminar aquellos datos cuyas predicciones asociadas sean negativas.

En conclusión, ambas métricas tienen fortalezas y debilidades, por lo que terminaremos utilizando ambas para medir la performance de nuestro predictor.

7. Experimentación

7.1. Introducción

En esta sección buscaremos, mediante diversas variantes, reducir el error arrojado por nuestro predictor a la hora de predecir los precios y la cantidad de metros cubiertos de un nuevo inmueble. Para esto utilizaremos técnicas como segmentación y feature engineering.
También iremos variando las características a tener en cuenta con el fin de encontrar el mejor conjunto de ellas considerando el trade-off entre el error obtenido y la complejidad de su combinación.

Salvo que se indique lo contrario, utilizaremos Cross Validation para reportar errores que no estén sesgados por la elección del set de validación (en este caso solo usaremos el RMSE). Esta técnica nos permitirá obtener resultados más robustos.

Por otra parte, analizaremos diferentes formas de tratar con valores faltantes que encontremos en el dataset.

7.2. Predicción de precios

7.2.1. Primera aproximación

Para tener un primer vistazo al problema consideraremos algunos hiperplanos generados por combinaciones de ciertas características númericas que consideramos tenían importancia, es decir que construiremos diferentes modelos basándonos en la intuición que tenemos acerca del problema. Iremos explicando los mismos a medida que los presentamos. El objetivo de este experimento es construir una base sobre la cual podamos trabajar.

Como primer paso, vamos a eliminar todos los datos que contengan algún NaN. Al finalizar este proceso, obtenemos un dataset de 49881 elementos.

Primera combinación

Para construir este modelo utilizaremos todas las variables numéricas a excepción de aquellas que creemos que no tienen una relación con el precio de la vivienda, estas son: la latitud, la logitud, el id y el id de la zona.

Características del modelo: metros totales, metros cubiertos, baños, habitaciones, garages, gimnasio, usos múltiples, piscina, escuelas cercanas, centros comerciales cercanos, antigüedad.

Hipótesis: creemos que utilizar este gran número de categorías resultará en un error bueno. Sin embargo, entendemos que constituye un modelo complejo que, probablemente, no valga la pena utilizar frente a otros más sencillos cuyo error sea similar, siempre y cuando estos existan y logremos encontrarlos a lo largo del trabajo.

Segunda combinación

Ahora que ya vimos qué sucedía si considerábamos una gran cantidad de variables, analizaremos cómo se comporta un modelo basado en una única categoría, en este caso la misma será la cantidad de métros cubiertos, ya que creemos que es la variable numérica más importante.

Características del modelo: metros cubiertos.

Hipótesis: suponemos que el error será más alto que en el experimento anterior ya que creemos que una sola categoría no alcanzará para explicar el precio.

Tercera combinación

Esta vez, tomaremos un modelo con las variables numéricas que consideramos menos importantes a la hora de explicar el precio.

Características del modelo: usos múltiples, piscina, escuelas cercanas, centros comerciales cercanos.

Hipótesis: como estamos considerando características que, a nuestro parecer, no deberían ser tan importantes para explicar los precios de los inmuebles, creemos que el error que arrojará nuestro predictor será notablemente más alto que en los demás casos.

Cuarta combinación

Por último, analizaremos el comportamiento de un modelo basado en las dos variables númericas que consideramos son las más importantes. Nuestro objetivo será ver si hay una mejora significativa respecto de la segunda combinación.

Características del modelo: metros cubiertos, habitaciones.

Hipótesis: creemos que estas dos características son las más decisivas para la correcta predicción de los precios, por lo que esperamos obtener un error bajo. Esperamos que el error se encuentre entre los errores de las combinaciones uno y dos, y que sea más similar al de esta última.

Para tener facilitar una comparación más visual de estos errores, dejamos la figura 1.

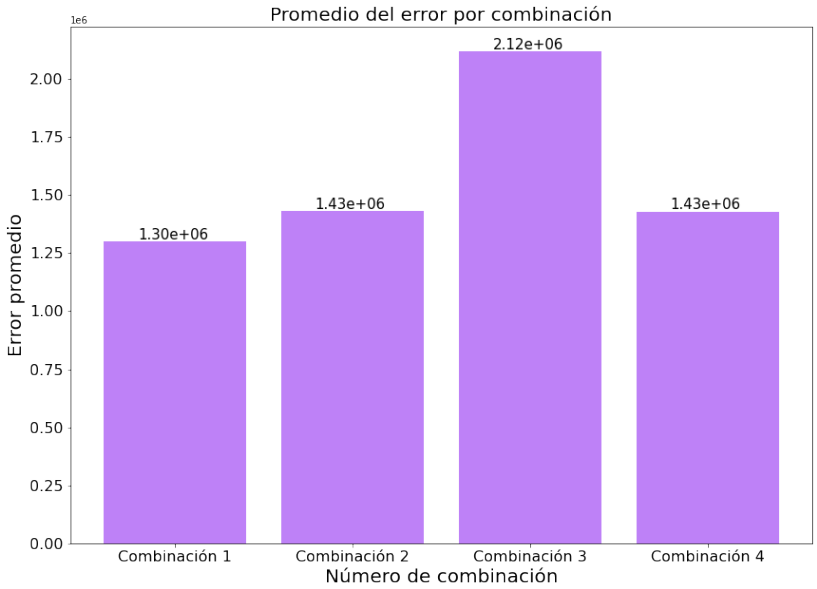


Figura 1: Errores promedio por combinación luego de correr Cross Validation.

Como última conclusión, viendo el gráfico de la figura 1, el error de la primera combinación es más bajo que el de las demás. Además, confirmamos la hipótesis acerca de las categorías de la combinación 3. Los errores de la segunda y de la cuarta combinación son similares y no resultan ser altos a pesar de ser modelos muy sencillos, lo que nos lleva a pensar que existe una combinación que no contenga muchas categorías que nos permite obtener un error relativamente bajo. Intentaremos hallarla en el próximo experimento.

Creemos que al momento de eliminar absolutamente todas las entradas que contengan NaNs, estamos dejando afuera un número demasiado grande de datos, por lo que la capacidad de predicción puede reducirse.
Por esta razón, vamos a eliminar los elementos del dataset que tengan NaNs únicamente en las características que utilizamos en la experimentación anterior, es decir: metros totales, metros cubiertos, baños, habitaciones, garages, gimnasio, usos múltiples, piscina, escuelas cercanas, centros comerciales cercanos y antigüedad.
El dataset ahora tendrá 120510 elementos.

Esperamos que ahora el error baje, ya que estaremos teniendo en cuenta más datos de entrenamiento.

Estos resultados se pueden ver en la figura 2

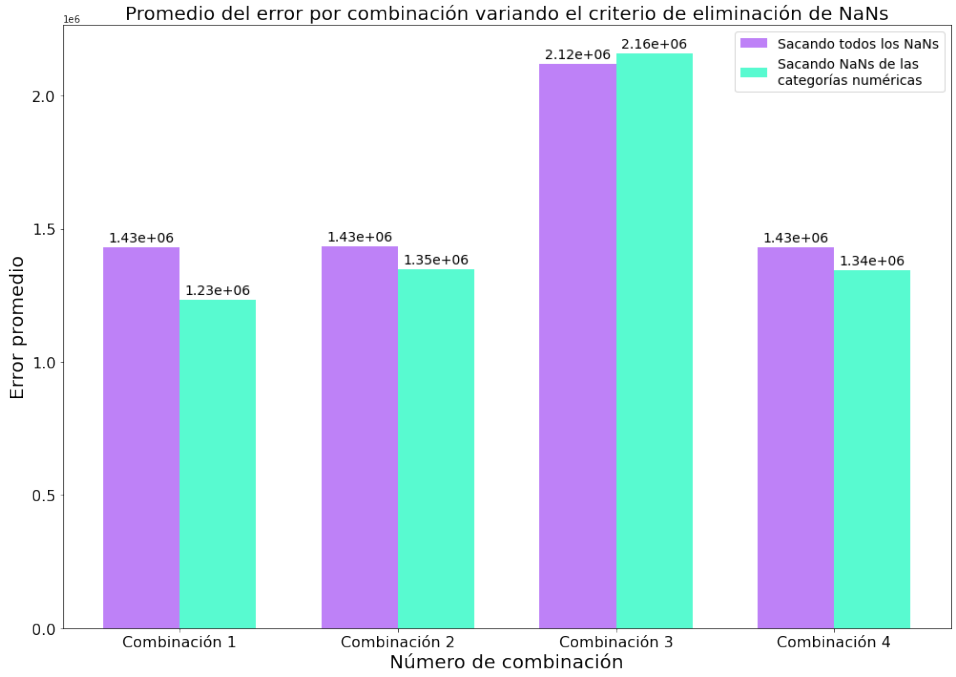


Figura 2: Errores promedio por combinación eliminando los datos con NaNs de dos formas distintas.

Viendo la figura 2 notamos que el error disminuyó en casi todos los casos, corroborándose así nuestra hipótesis.

Con esta información, veremos qué sucede si eliminamos únicamente los datos que contienen NaNs en las categorias utilizadas en cada combinación. Notemos que en cada combinación estaremos trabajando con datasets de distintos tamaños. Estos contienen 120510, 222600, 240000 y 209785 elementos respectivamente.

Creemos que el error debería disminuir nuevamente ya que en todos los casos estamos trabajando con sets de entrenamiento cuyo tamaño es mayor o igual al de los sets utilizados previamente.

En la figura 3 compararemos los resultados de este experimento junto con los obtenidos en los dos anteriores, para poder decidir cuál de estos tres acercamientos es el más adecuado para resolver el problema.

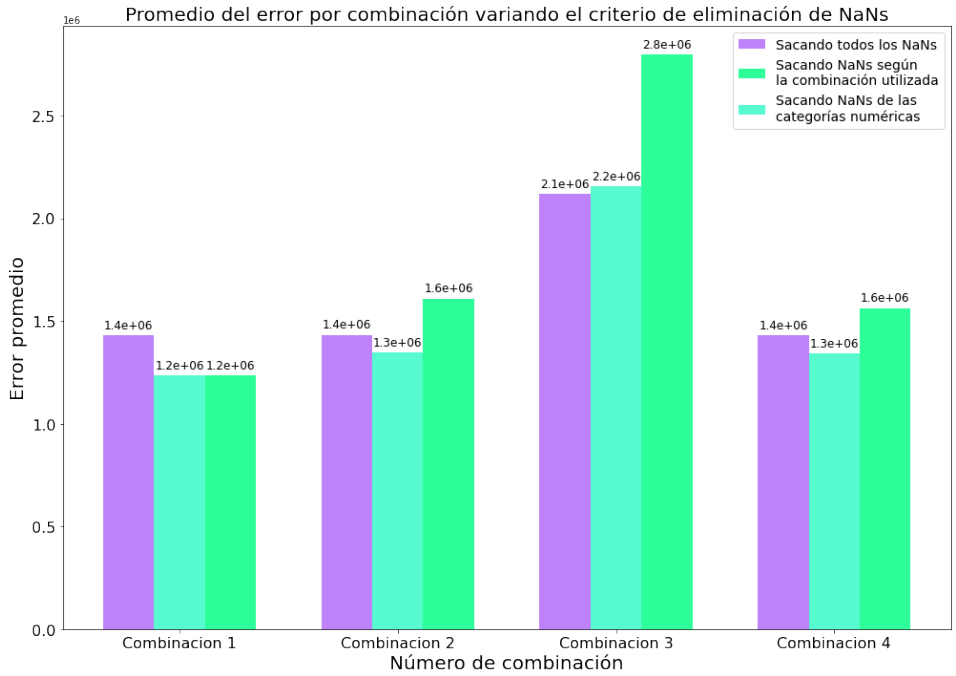


Figura 3: Errores promedio por combinación eliminando los datos con NaNs de tres formas distintas.

Contrario a lo que pensábamos, descubrimos mirando la figura 3 que el error aumentó en la mayoría de los casos. Sospechamos que la razón de este fenómeno es que los datos que antes eliminábamos y ahora conservamos no son confiables. Por ejemplo, estamos teniendo en cuenta datos en los que no se ha cargado la cantidad de metros totales.

Luego de estos experimentos concluímos que la mejor forma de proceder frente a valores faltantes es limpiando aquellos datos que contienen NaNs en alguna de las variable numéricas (las que intervinieron en la primera combinación). Además, notamos que el error correspondiente a las coombinaciones 2 y 4 no es mucho mayor que el error de la combinación 1 a pesar de que esta se basa en un número notablemente más alto que las otras dos.

7.2.2. Mejores combinaciones

En esta sección trataremos de encontrar la mejor combinación de categorías para pasarle a nuestro predictor. En particular, luego de lo visto en la sección 7.2.1, queremos analizar las combinaciones con pocas categorías para ver si logramos hallar alguna que nos genere un modelo cuya performance sea buena comparada con modelos más complejos. Para esto, entrenaremos con todas las combinaciones posibles de menos de 4 elementos y nos quedaremos con la que nos de el error más bajo. Para comparar los errores solo utilizaremos el error cuadrático medio. En esta oportunidad, por cuestiones de complejidad temporal, no haremos Cross Validation, si no que haremos una simple partición del dataset en training y validación.

Sabemos que la combinación con menos error contendrá a la gran mayoría de las características, ya que a mayor cantidad de variables, mayor precisión obtendremos. Sin embargo, hay ocasiones en las que creemos que terminamos tratando con un modelo cuya performance no justifica la complejidad que conlleva. Comprobar esto será el objetivo de este experimento.

Luego de correr el experimento, hicimos un recuento de las características que aparecían en las mejores 50 combinaciones, con el fin de hacernos una idea de qué categorías eran las más importantes a la hora de predecir precios. Los resultados del mismo pueden verse en la figura 4.

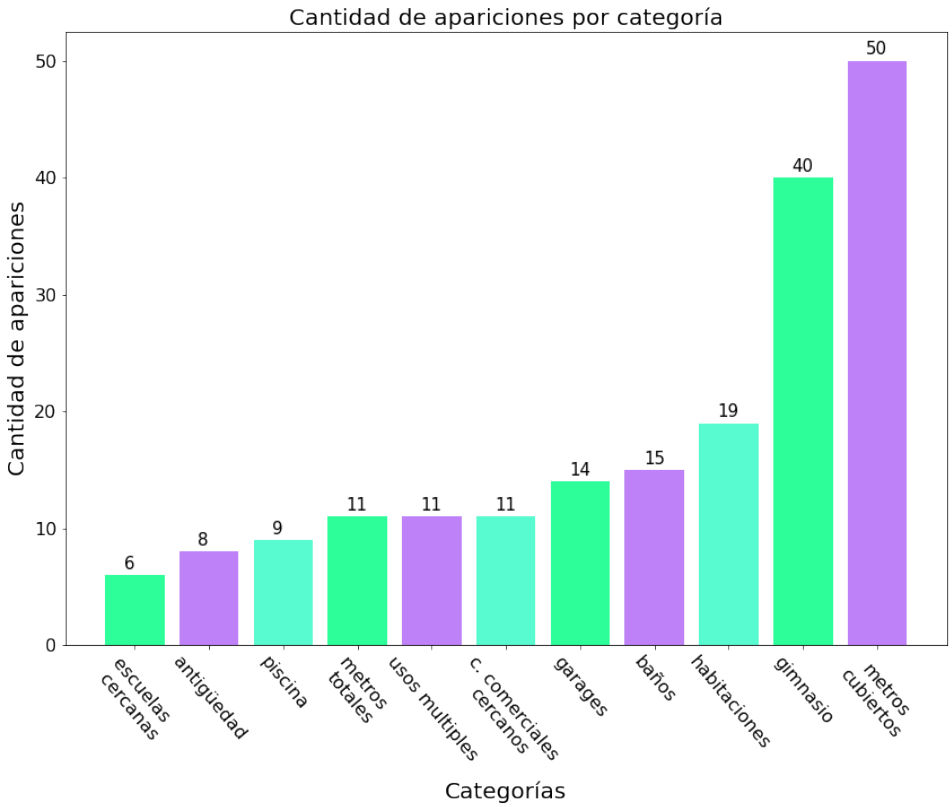


Figura 4: Cantidad de apariciones de cada categoría en las mejores 50 combinaciones.

Analizando la figura 4, vemos que las características más importantes resultan ser: metros cubiertos (la cual apareció en todas las combinaciones ganadoras), gimnasio, habitaciones y baños.

Por esta razón elegiremos las características mencionadas para construir nuestro modelo. Para compararlo con el de mejor error de la seccion 7.2.1, calcularemos el RMSE y el RMSLE de Cross Validation. Para poder utilizar esta última métrica tuvimos que deshacernos de los datos que se correspondían con una predicción negativa. Tomamos esta decisión al observar que estas representaban menos del 0.05 % del total de los avisos de validación. Además, tomando algunos de estos datos, podemos ver que lo más probable es que los mismos hayan sido mal cargados, y por lo tanto, no aporten información confiable. Un ejemplo de esto se puede ver en el cuadro 1.

	Entrada 1	**Entrada 2**
Habitaciones	10	3
Baños	1	1
Metros cubiertos	90	16
Metros totales	100	16
Precio	$550.000	$540.000

Cuadro 1: Ejemplos de datos que llevaron a predicciones negativas.

Viendo el cuadro 1, notamos que en ninguna de las dos entradas la cantidad de habitaciones se condice con la cantidad de metros reportados, además, en el caso de la entrada 1, tampoco con la cantidad de baños ya que sería raro que una casa cuente con diez habitaciones y un único baño.
Por otra parte, nos parece extraño que, siendo propiedades tan distintas, el precio sea tan similar.

Lo visto no hace más que reafirmar nuestra hipótesis acerca de la credibilidad de los datos que resultan en predicciones negativas.

Después de correr Cross Validation para medir la performance del regresor, obtuvimos los siguientes resultados:

	Modelo Complejo	Modelo Simplificado
RMSE	1.23e+6	1.27e+6
RMSLE	0.5383	0.5161

Cuadro 2: Comparción de errores de Cross Validation para dos modelos distintos.

Como podemos ver, el nuevo error cuadrático medio es un poco más alto que el del modelo basado en las 11 variables numéricas, sin embargo, nos permite obtener un modelo mucho más simple para explicar los precios ya que solo se basa en el conjunto más intuitivamente importante. Además, notamos que el RMSLE del modelo simplificado es menor que el del complejo.
Por estas razones, nos quedaremos el regresor basado en las categorías metros cubiertos, habitaciones, baños y gimnasio para los próximos experimentos.

7.2.3. Feature engineering: PBI

El dataset posee información sobre la provincia en la que están ubicados los inmuebles, sin embargo, como esta es una variable categórica, no pudimos incluírla en los experimentos anteriores tan fácilmente, aunque sabemos que es una característica importante que deberíamos tener en cuenta. Es por esto que agregaremos al dataset el PBI correspondiente a la provincia de cada inmueble e incluiremos esta categoría en las variables sobre las que construiremos un nuevo modelo.

Antes de comenzar, filtramos nuevamente el dataset para deshacernos de los datos que tuvieran faltantes en la categoría provincia. Una vez terminado este proceso, trabajaremos con un set de 120502 elementos.
Recordemos que nuestro modelo se basará en las características metros cubiertos, habitaciones, baños, gimnasio y PBI.

Nuevamente utilizaremos Cross Validation para reportar errores más robustos.

Hipótesis: creemos que la variable "provincia" es muy importante, así que al haber encontrado una manera de que nuestro predictor la tenga en cuenta, esperamos ver una mejora significativa en la performance del modelo.

Comparemos los errores obtenidos con los anteriores en el cuadro 3.

	Modelo Complejo	Modelo Simplificado	Modelo Simplificado con PBI
RMSE	1.23e+6	1.27e+6	1.19e+6
RMSLE	0.5383	0.5161	0.5393

Cuadro 3: Comparción de errores de Cross Validation para tres modelos distintos.

Viendo el cuadro 3 confirmamos que el error cuadrático medio del nuevo modelo bajó bastante con respecto a los otros dos. Sin embargo también notamos que el RMSLE aumentó con respecto al modelo simplificado del experimento 7.2.2. No estamos muy seguros del por qué de esto, ya que en un principio pensamos que podría haberse debido al hecho de que el error logarítmico penaliza en mayor medida a las predicciones que son menores al valor real. Pero comparando las predicciones de los modelos comprobamos que esta no era la razón.
No obstante, consideramos que la mejora obtenida respecto del RMSE es lo suficientemente significativo como para adoptar el modelo simplificado con PBI en los próximos experimentos.

7.2.4. Feature engineering: Bueno para Familias

En este experimento queremos comprobar si logramos una mejor performance al considerar una nueva variable que combine algunas de variables ya usadas.
En esta ocasión crearemos el campo "bueno para familias" que agrupa aquellas propiedades que cuenten con los siguientes requisitos:

- tener más de un baño.
- contar con una piscina.

- poseer más de dos habitaciones.
- estar ubicadas cerca de una escuela.

Hipótesis: como estamos agregando información nueva, creemos que debería haber una mejora en el error de este regresor comparado con los anteriores. Sin embargo no sabemos exactamente cuál será la magnitud de la mejora obtenida, pero sospechamos que no será tan significativa como en el experimento de la sección 7.2.3 ya que consideramos que la información agregada en este caso no es tan decisiva.

Veamos los resultados en el cuadro 4. Notar que a partir de ahora estaremos utilizando la siguiente notación:

- Modelo Exp 1: se corresponde con el modelo basado en las once categorías numéricas de los experimentos anteriores.
- Modelo Exp 2: hace referencia al llemado Modelo Simplificado de la sección 7.2.2.
- Modelo Exp 3: se trata del experimento de la sección 7.2.3.
- Modelo Exp 4: modelo generado en este experimento al incorporar la variable "bueno para familias".

	Modelo Exp 1	**Modelo Exp 2**	**Modelo Exp 3**	**Modelo Exp 4**
RMSE	1.23e+6	1.27e+6	1.19e+6	1.19e+6
RMSLE	0.5383	0.5161	0.5393	0.5389

Cuadro 4: Comparción de errores de Cross Validation para modelos distintos.

Luego de analizar los resultados, concluímos que agregar la variable "bueno para familias" no introdujo una mejora tangible con respecto al experimento de la sección 7.2.3 ya que el error cuadrático se mantuvo igual y el logarítmico se decrementó en un 0.14 %.
Es por esto, que consideramos no vale la pena complejizar el modelo incluyendo esta nueva característica al no haber encontrado mejoras relevantes.

7.2.5. Feature engineering: vacacional

Luego de ver que el experimento anterior no introdujo mejoras, intentaremos replicar el experimento anterior utilizando una nueva variable. Esta indicará si una propiedad es adecuada para vacacionar.
Para construirla nos fijamos que la propiedad tuviese piscina o que estuvise ubicada en alguna ciudad o provincia costera.

Hipótesis: si bien creíamos que este experimento nos traería mejoras, luego de ver los resultados de la sección 7.2.3, no estamos tan seguros de la diferencia que pueda hacer.
Igualmente, correremos el experimento para ver si logramos obtener alguna mejoría y así trasladar este nuevo modelo a los experimentos siguientes.

Los resultados se encuentran en el cuadro 5. En este caso, nos referiremos al nuevo modelo como "Modelo Exp 5".

	Modelo Exp 1	**Modelo Exp 2**	**Modelo Exp 3**	**Modelo Exp 4**	**Modelo Exp 5**
RMSE	1.23e+6	1.27e+6	1.19e+6	1.19e+6	1.19e+6
RMSLE	0.5383	0.5161	0.5393	0.5389	0.5389

Cuadro 5: Comparción de errores de Cross Validation para modelos distintos.

Al igual que en la sección anterior, no notamos cambios en los errores reportados.
Tratando de explicar los resultas obtenidos, nos dimos cuenta de que la categoría "bueno para familias" y "vacacional" agrupaban a muy pocas propiedades (6060 de 120502 y 14909 de 120502). Sabiendo esto, tiene sentido que los nuevos regresores se hayan comportado de manera muy similar al del experimento de la sección 7.2.3.

Luego de no haber conseguido mejoras en los últimos experimentos, cambiaremos el enfoque para trabajar con segmentación del dataset. Notar que seguiremos utilizando el modelo obtenido en la sección 7.2.3 ya que a nuestro parecer es el modelo más óptimo.

7.2.6. Segmentación por zona geográfica

Como estamos tratando con un dataset muy grande y variado, querer explicar todos los precios mediante una única función parece ser una aspiración muy grande. Por esta razón, buscaremos armar diferentes predictores cuyo dominio no sea tan amplio y así lograr una mayor precisión.
En este experimento segmentaremos el dataset por regiones. Estas son las siguientes:

- Noroeste: Baja California Sur, Baja California Norte.
- Pacífico: Sonora, Sinaloa.
- Norte: Chihuahua, Coahuila, Nuevo León, Tamaulipas,Durango.
- Occidente: Nayarit, Jalisco, Colima, Michoacán.
- Distrito: Distrito Federal, Hidalgo, Edo. de México.
- Bajío: Aguascalientes, Guanajuato, Querétaro, San Luis Potosí, Zacatecas.
- Golfo: Morelos, Oaxaca, Puebla, Veracruz, Tabasco, Tlaxcala, Guerrero
- Suroeste: Yucatán, Campeche, Quintana Roo, Chiapas

El procedimiento para lograr esto será: en cada fold construiremos un regresor por región, donde cada uno de estos será entrenado por el subconjunto de propiedades del dataset de training que pertenecen a esa zona. Luego, al momento de predecir los datos de validación, miraremos a qué región pertenece cada uno y le asignaremos el regresor correspondiente. Por último calcularemos el error cuadrático medio con todas estas predicciones.

Hipótesis: estamos seguros de que mediante la segmentación por zonas geográficas conseguiremos predicciones más precisas. Sin embargo, no tenemos una noción clara de la magnitud de esta mejoría.

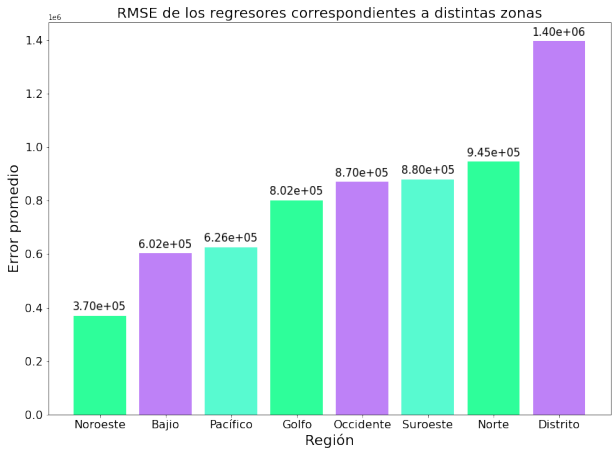
Luego, de correr el experimento los resultados obtenidos fueron los siguientes:

	Modelo Exp 3	**Modelo Regiones**
RMSE	1.20e+06	1.10e+06
RMSLE	0.5394	0.4773

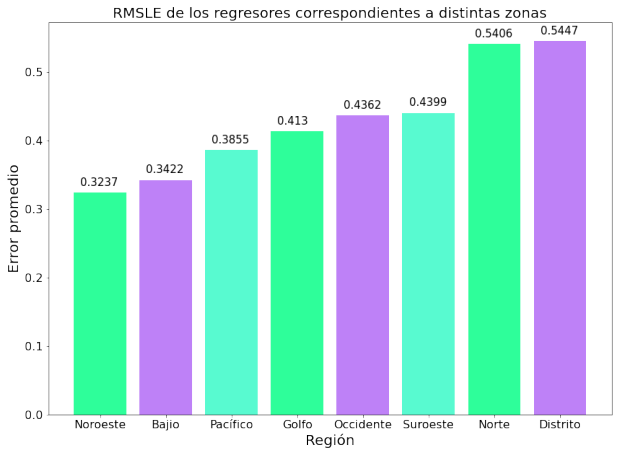
Cuadro 6: Comparción de errores de Cross Validation para dos modelos distintos.

Como podemos ver en el cuadro 6, tanto el error cuadrático como el error logarítmico obtenidos disminuyeron notablemente.
La magnitud de la mejora obtenida superó a la conseguida al incorporar el PBI, el cual, hasta el momento, había sido nuestro mejor avance. Ya que, un salto en el error de 1.23e+6 a 1.19e+6 representa un 3.25 %, mientras que al hacerlo de 1.23e+6 a 1.09e+6, se obtiene una mejora del 11.38 %. Por lo tanto, este fue el experimento más exitoso que realizamos.

Revisando los errores promedio correspondientes a cada zona, notamos que los pertenecientes a la región "Distrito" son mucho mayores a los que corresponden al resto de los regresores. Estas diferencias pueden verse en la figura 5.



(a) Errores cuadráticos de los regresores correspondientes a las distintas zonas.



(b) Errores logarítmicos de los regresores correspondientes a las distintas zonas.

Figura 5

Veamos cuál es la cantidad de datos por región en la figura 6.

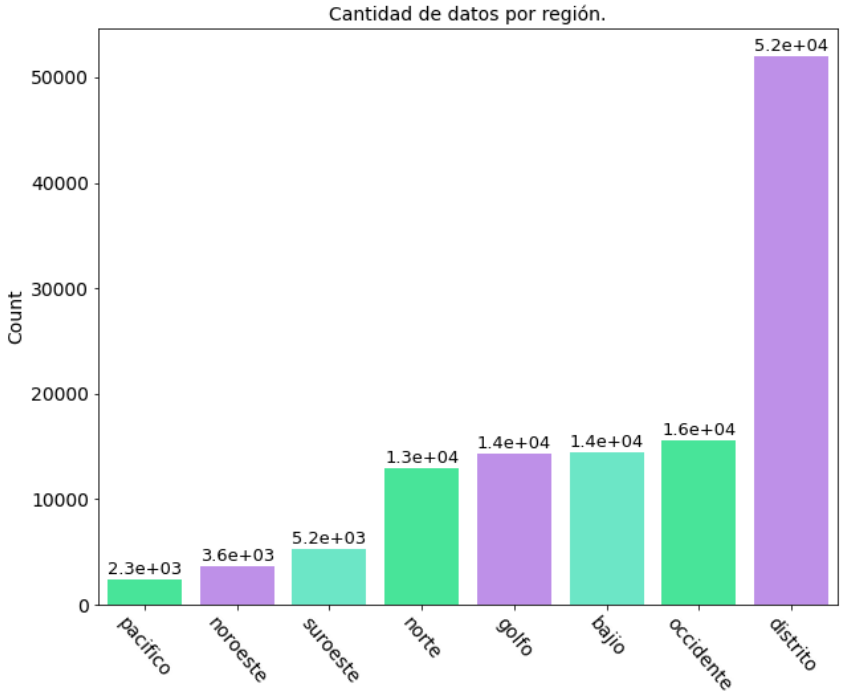


Figura 6: Cantidad de avisos por región.

Viendo la figura 6, notamos que Distrito posee muchísimos más datos que las demás regiones, lo que nos lleva a pensar que estamos en presencia de un dataset muy heterogéneo.
Teniendo esta nueva información, en el próximo experimento trataremos de hallar un nuevo regresor para las propiedades pertenecientes a Distrito con el objetivo de disminuir el error del modelo planteado en el experimento actual.

7.2.7. Segmentación de la región "Distrito" por tipo de propiedad

En esta sección intentaremos disminuir el error de las predicciones de los inmuebles ubicados en la región Distrito. Para esto, segmentaremos esa porción del dataset por el tipo de propiedad.

Viendo que la mayoría de los inmuebles era una casa, un apartamento o una casa en condominio, decidimos quedarnos solamente con estos tipos y agrupar al resto en uno nuevo llamado “Otro”.
La distribución de los inmuebles por tipo de propiedad pueden verse en la figura 7.

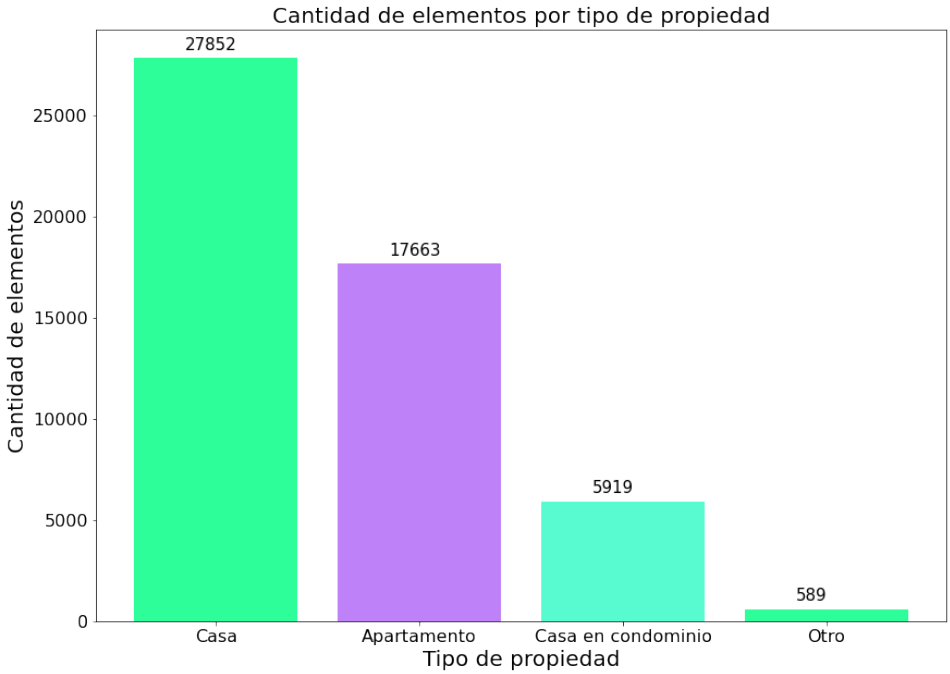
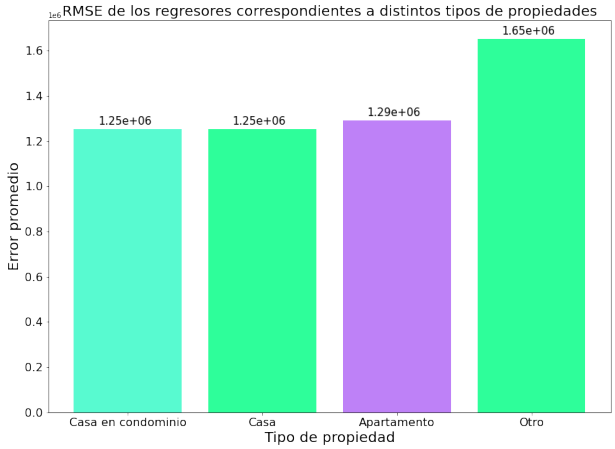


Figura 7: Countplot del tipo de propiedad luego de segmentar el dataset de la región Distrito.

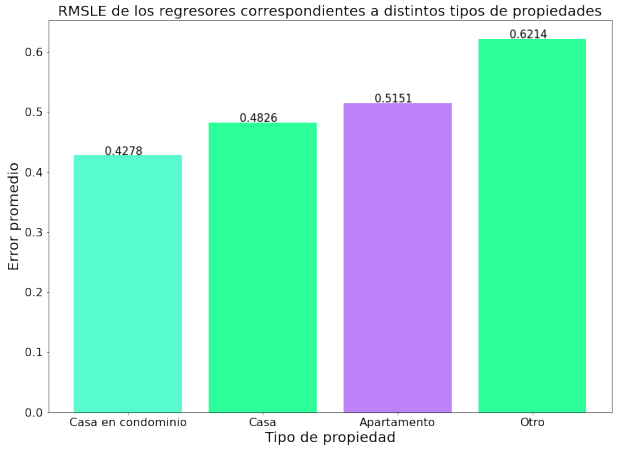
En esta versión, construiremos cuatro nuevos regresores, uno por cada tipo de propiedad.

Hipótesis: esperamos ver una mejoría en el error de las predicciones de los inmuebles situados en la zona Distrito al estar segmentando nuevamente los datos. Creemos que de esta forma conseguiremos entrenar cada regresor sobre datos más homogéneos, lo que nos permitirá ser más precisos a la hora de predecir.

Como primera medida veamos en la figura 8 los errores arrojados por cada regresor para ver si encontramos una mejora respecto del error de las predicciones de Distrito. Recordemos que este valor era $1{,}39e+6$.



(a) Errores cuadráticos de los regresores correspondientes a los tipos de propiedad ubicadas en Distrito.



(b) Errores logarítmicos de los regresores correspondientes los tipos de propiedad ubicadas en Distrito.

Figura 8

Viendo la figura 8 y teniendo en cuenta lo visto en la figura 7, concluimos que la mejoría en el error del predictor de precios de casas tendrá un gran impacto en el error final; mientras que, a pesar de que el error del tipo “Otro” sea

muy alto, hay tan pocas propiedades que pertenecen a este subgrupo que no provocará cambios significativos en el error total. Por lo tanto, creemos que lograremos tener una mejora significativa con respecto al experimento de la sección 7.2.6.

Confirmemos esta hipótesis comparando el error de Distrito del experimento anterior y el del actual en el cuadro 7.

	Error Distrito base	**Error Distrito segmentado**
RMSE	1.40e+06	1.27e+06
RMSLE	0.5447	0.4883

Cuadro 7: Comparción de errores de Cross Validation para las predicciones de Distrito originales y segmentadas por tipo de propiedad.

Luego de haber comprobado que la segmentación por tipo de propiedad sobre los inmuebles de Distrito funciona, reemplazaremos el regresor correspondiente a esta región del experimento 7.2.6 por los obtenidos en el experimento actual. Tenemos confianza en que esta modificación provocará mejoras importantes.

Veamos entonces, cuál es el error final que obtenemos realizando tanto las segmentaciones por región, como la segmentación por tipo de propiedad en la región "Distrito". Además tengamos en cuenta que todos estos regresores fueron construidos teniendo en cuenta las categorias metroscubiertos, habitaciones, baños, gimnasio y PBI. En el cuadro 8, podemos ver un comparación entre el error base que conseguimos en la sección 7.2.1, junto con el mejor error que habíamos conseguido segmentando por región en la sección 7.2.6 y el error final conseguido en este expiremento.

	Modelo Exp 1	**Modelo Regiones**	**Modelo Final**
RMSE	1.23e+06	1.10e+06	1.03e+06
RMSLE	0.5383	0.4773	0.4524

Cuadro 8: Comparción de errores de Cross Validation para las predicciones de tres modelos distintos.

Como podemos ver en el cuadro 8, nuestras hipótesis se corrobaron al haber obtendido una importante mejora en el desempeño del modelo correspondiente al experimento actual con respecto a los regresores anteriores.

7.2.8. Conclusiones

Para conlcuir, creemos que los resultados de la experimentación fueron muy satisfactorios al haber disminuído el error inicial en un 16.26 %.
En este caso, las mejorías más significativas fueron obtenidas al segmentar el dataset. Esto fue a causa de la heterogeneidad del set de datos inicial, la cual hacía imposible poder predecir todos los precios mediante una única función. Esto mismo no solo ocurría en el dataset original, si no que además se podía ver en los inmuebles situados en la región "Distrito", ya que estos constituían la mayor parte del dataset.
De todas formas, no podemos dejar de mencionar la mejora conseguida al añadir el PBI. Creemos que esta se dio al permitir que el regresor tuviese en cuenta la situación económica de la provincia de cada inmueble, lo que es un factor muy importante a la hora de tasar una propiedad.

Por estas razones, consideraremos al regresor construído en la sección 7.2.7 como el modelo final.

7.3. Predicción de metros cubiertos

7.3.1. Primera aproximación

En esta ocasión, tendremos como objetivo construir un regresor que nos permita predecir la cantidad de metros cubiertos que tiene un inmueble con la mayor precisión posible.
Para esto, necesarimente, debemos deshacernos de todos los datos que no vengan provistos de este campo. Luego de este procedimiento, nuestro set de datos está compuesto por 222600 elementos.

Como primer acercamiento, elegiremos cuatro combinaciones de características para construir regresores que luego compararemos. Estas son:

- **Primera combinación:** metros totales, precio, baños, habitaciones, garages, gimnasio, usos múltiples, piscina y antigüedad.

  Hipótesis: nuevamente, pensamos que este regresor será el de menor error ya que condensa más información. Notar que en esta ocasión no incluímos escuelas cercanas ni centros comerciales cercanos, debido a que no sentimos que estuvieran relacionados a la cantidad de metros cubiertos que posee una vivienda.

- **Segunda combinación:** metros totales.

  Hipótesis: creemos que esta es una variable que está súmamente relacionada con la que queremos predecir. Por esta razón, sospechamos que el error será bajo, aunque no tanto como la combinación anterior.

- **Tercera combinación:** precio.

  Hipótesis: como vimos en la experimentación sobre la predicción de precios, esta variable está relacionada con la variable a predecir, sin embargo, también vimos que no es la única característica que interviene, por lo que esperamos que los errores sean relativamente buenos, pero peores que los de la combinación anterior.

- **Cuarta combinación:** metros totales y precio.

  Hipótesis: esperamos que esta combinación arroje muy buenos resultados ya que, intuitivamente, consideramos que son las dos características más importantes. De todas formas, confiamos en que no superará al rendimiento de la primera combinación.

En esta etapa, utilizaremos las mismas estrategias para el manejo de valores mal cargados que en la sección 7.2.

Nuevamente, comenzaremos eliminando todos los datos que contengan NaNs en alguna de sus características, lo que nos reduce tamaño del dataset a 49881 elementos. Los resultados de este experimento pueden verse en la figura 9.

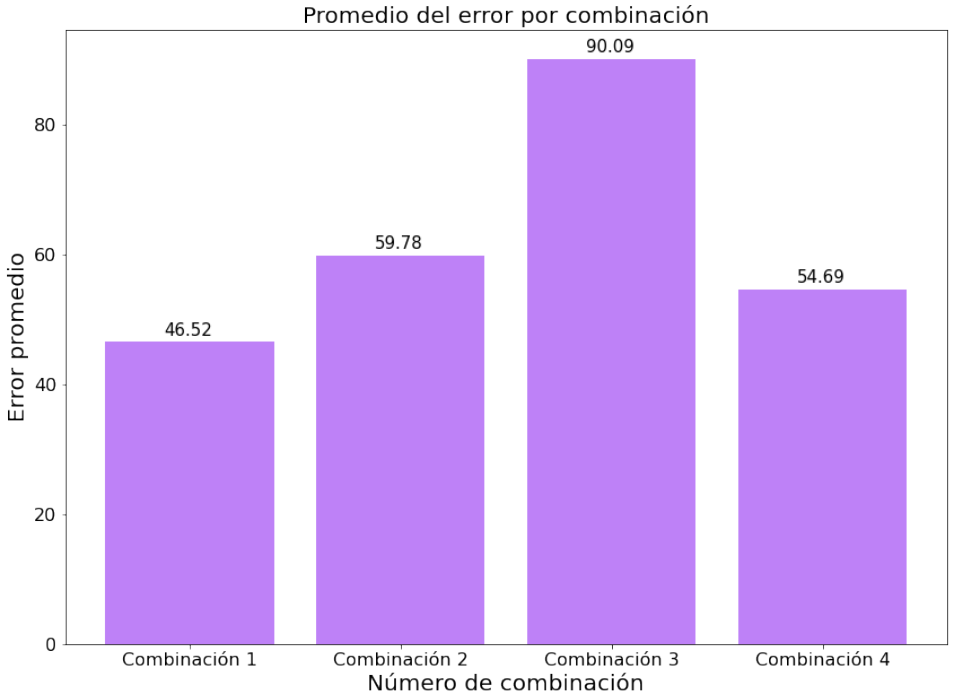


Figura 9: Errores promedio por combinación luego de correr Cross Validation.

Como podemos ver en la figura 9, las hipótesis se cumplieron casi en su totalidad satisfactoriamente. Algo que nos sorprendió fue la pobre performance del regresor costruído sobre la variable "precio". Así mismo, vemos que esta variable puede ser muy útil en combinación con otras, como se puede ver en la combinación 4. Por lo cual no creemos que haya sido erróneo considerar que esta variable explica en cierta medida el valor a predecir.

Ahora, volveremos a correr el experimento manejando de dos maneras distintas los valores faltantes. Veremos si aumentamos la performance eliminando únicamente los datos que posean NaNs en alguno de sus campos númericos,

estos son: metros totales, precio, baños, habitaciones, garages, gimnasio, usos múltiples, piscina y antigüedad. De esta forma trabajaremos con un dataset mayor, el cual está compuesto por 120510 avisos.

También analizaremos qué sucede si sólo nos deshacemos de aquellos datos que tengan valores faltantes en las categorías sobre las que se basa el predictor a utilizar. De esta forma, mientras menos categorías se utilicen, menos entradas son eliminadas. Luego de esto, los distintos datasets resultan tener los siguientes tamaños:

- Primera combinación: 120510.
- Segunda combinación: 171133.
- Tercera combinación: 222600.
- Cuarta combinación: 171133.

Los resultados de este experimento se pueden ver en la figura 10.

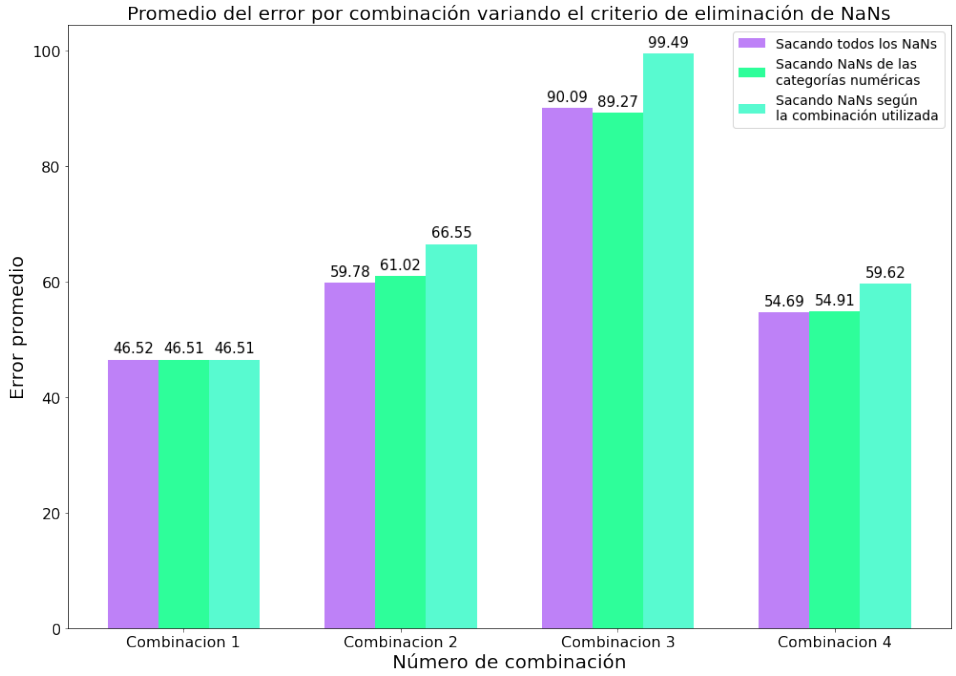


Figura 10: Errores promedio por combinación luego de correr Cross Validation.

Como podemos ver, las primeras dos estrategias resultaron ser las mejores y lograron una performance muy similar. Sin embargo, a diferencia de lo obtenido en la experimentación sobre los precios, el mejor approach para eliminar los NaNs pareciera ser no tener en cuenta ninguna entrada que los contenga sin importar en donde.

7.3.2. Mejores combinaciones

En esta etapa de la experimentación, buscaremos comparar la performance de regresores construídos sobre diferentes combinaciones de categorías con el objetivo de hallar la más óptima, teniendo en cuenta el trade-off entre la simpleza del modelo y el error cometido.
Para esto, volvimos a considerar todas las posibles combinaciones que contengan hasta cuatro categorías y veremos si logramos hallar una combinación que arroje un error similar al de la primera combinación de la sección 7.3.1. En este experimento utilizaremos validación simple.

Veamos los resultados del experimento en la figura 11.

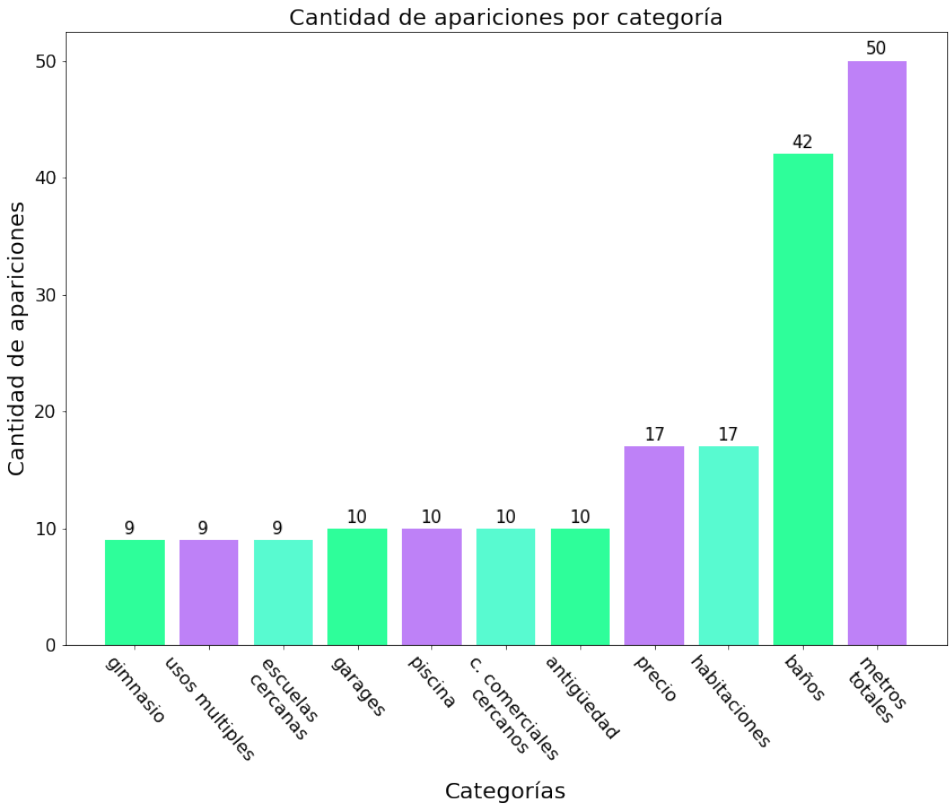


Figura 11: Cantidad de apariciones de cada categoría en las mejores 50 combinaciones.

Como podemos ver en la figura 11, hay cuatro categorías que destacan, estas son: metros totales, baños, habitaciones y precio.
Correremos Cross Validation con la combinación constituída por las características mencionadas en el párrafo anterior y compararemos su error con la mejor combinación del experimento 7.3.1 en el cuadro 9.

	Modelo Complejo	**Modelo Simplificado**
RMSE	46.63	47.10
RMSLE	0.2919	0.2930

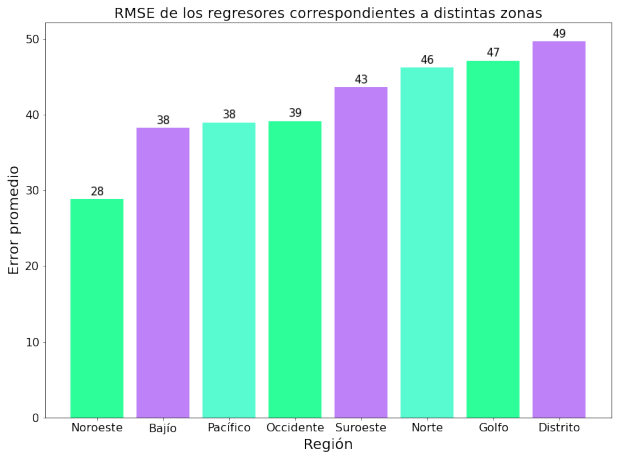
Cuadro 9: Comparción de errores de Cross Validation para dos modelos distintos.

Viendo el cuadro 9, notamos que casi alcanzamos el error obtenido por el mejor modelo del experimento 7.2.2, lo que es un gran avance teniendo en cuenta que utilizamos cinco categorías menos. Cabe aclarar que si bien esperábamos alcanzar valores aceptables para el error, no creíamos que pudiésemos lograr este nivel de performance combinando tan pocas categorías.
De aquí en adelante utilizaremos este nuevo modelo para experimentar con la intención de mejorarlo.

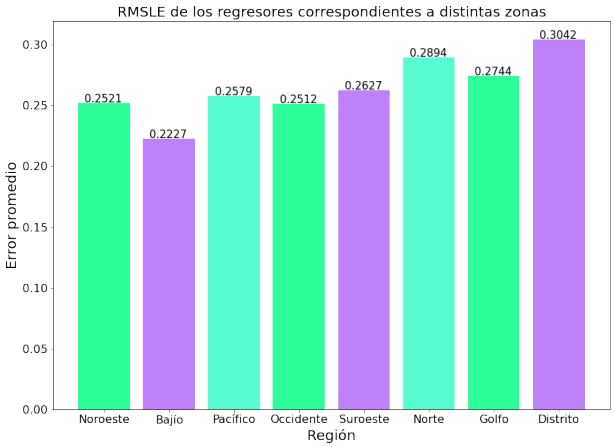
7.3.3. Segmentación por zona geográfica

Teniendo en cuenta que en la experimentación sobre la predicción de precios logramos grandes mejoras al segmentar el dataset, utilizaremos este método como primer estrategia para intentar perfeccionar el modelo construído en la sección 7.3.2.
Repetiremos la segmentación ya utizada, ya que creemos que la región en la que está ubicada un inmueble está relacionada con el tamaño del mismo. Generalmente, las viviendas de las regiones con mayor densidad poblacional suelen ser más pequeñas que las ubicadas en zonas rurales.

Veamos en la figura 12 cuál es el error de cada uno de los predictores (recordar que contamos con un regresor por zona).



(a) Errores cuadráticos de los regresores correspondientes a las distintas zonas.



(b) Errores logarítmicos de los regresores correspondientes a las distintas zonas.

Figura 12

Como podemos ver en la figura 12, no hay una gran diferencia entre los errores de los regresores, por lo que no creemos que haga falta segmentar nuevamente. En el cuadro 10 compararemos el error del modelo que contiene a estos predictores contra el de la sección 7.3.2.

	Modelo Simplificado	**Modelo Región**
RMSE	47.10	45.64
RMSLE	0.2930	0.2806

Cuadro 10: Comparción de errores de Cross Validation para dos modelos distintos.

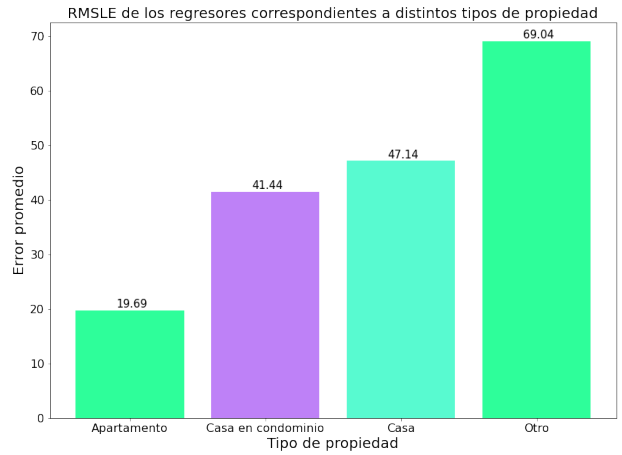
Mirando la tabla 10, notamos que logramos obtener una mejora aunque no creemos que sea muy significativa. Luego de este experimento concluímos que quizás la ubicación no era tan importante a la hora de predecir la cantidad de metros cuadrados de un inmueble.
Por esta razón, intentaremos segmentar nuevamente el dataset original, esta vez por tipo de propiedad.

7.3.4. Segmentación por tipo de propiedad

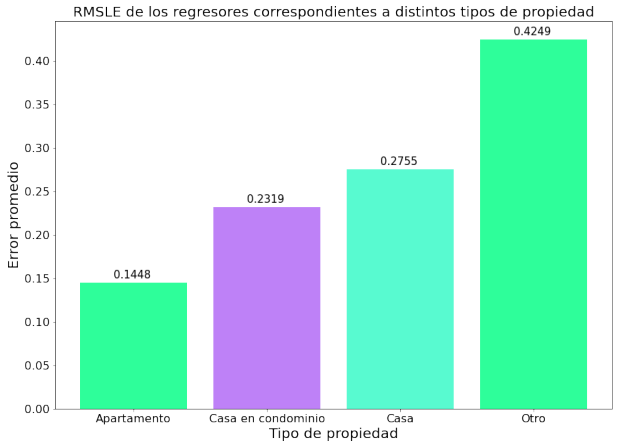
En esta sección veremos si conseguimos mejorar la performance de nuestro modelo si segmentamos nuestros datos por tipo de propiedad.

Hipótesis: creemos que esta segmentación funcionará mejor que la de la sección 7.3.3 ya que, a nuestro parecer, el tipo de propiedad está muy ligado al tamaño de la misma.

Como hicimos anteriormente, comenzaremos viendo los errores de los nuevos regresores por separado. Estos pueden verse en la figura 13.



(a) Errores cuadráticos de los regresores correspondientes a los distintos tipos de propiedades.



(b) Errores logarítmicos de los regresores correspondientes a los distintos tipos de propiedades.

Figura 13

Como pudimos ver en la experimentación de precio, la dominancia del error proviene de las casas y los apartamentos al ser los tipos más poblados. Igualmente, podemos rever esto en la figura 14

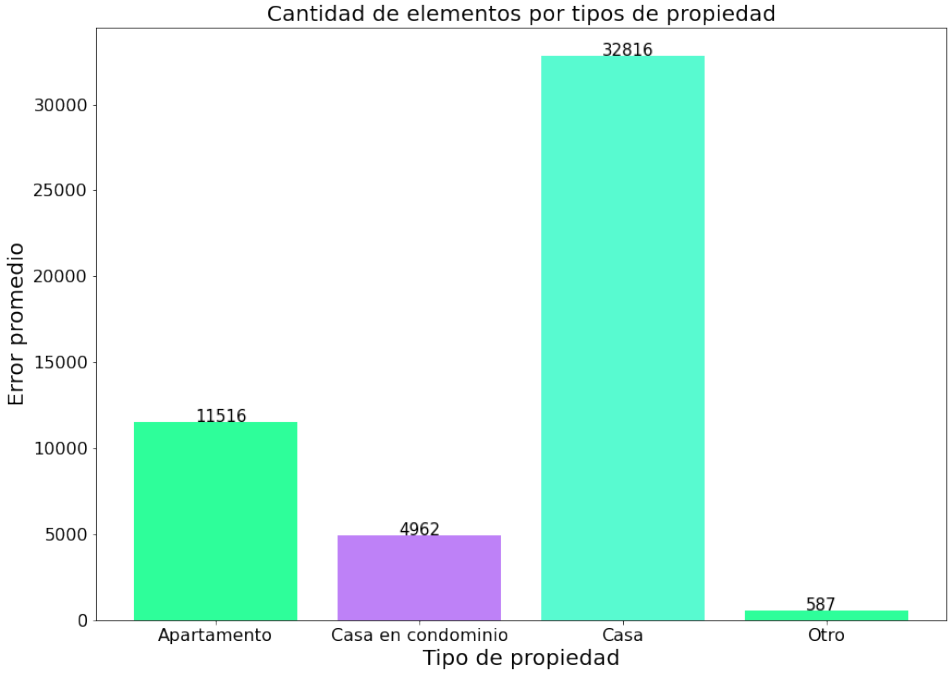


Figura 14: Cantidad de elementos por tipo de propiedad.

Ahora sí, veamos cuál es el error final del modelo que reúne a los cuatro regresores.

	Modelo Simplificado	**Modelo Región**	**Modelo Tipo de Prop.**
RMSE	47.10	45.64	42.17
RMSLE	0.2930	0.2806	0.2495

Cuadro 11: Comparción de errores de Cross Validation para tres modelos distintos.

Como podemos ver en el cuadro 11, segmentar por tipo de propiedad logra mejorar bastante la performance de nuestro modelo. Podemos ver, además, que efectivamente era más importante el tipo de propiedad que la región, ya

que esta nos proporcionó una mejoría más amplia.

Antes de concluír este experimento, veremos si logramos obtener mejores resultados realizando las dos segmentaciones simultáneamente. En este experimento estaremos trabajando con 4 regresores por región (uno por cada tipo de propiedad), lo que, al tener 8 regiones, nos define un modelo conformado por 32 regresores.
Igualmente, viendo que las mejorías no fueron demasiado significativas con las dos segmentaciones, esperamos una leve mejoría, pero ninguna sorpresa.

Comparemos los errores obtenidos contra los de los modelos construídos en las secciones anteriores.

	Modelo Simplificado	**Modelo Región**	**Modelo Tipo de Prop.**	**Modelo Combinado**
RMSE	47.10	45.64	42.17	41.07
RMSLE	0.2930	0.2806	0.2495	0.2410

Cuadro 12: Comparción de errores de Cross Validation para cuatro modelos distintos.

Los resultados de la tabla 12 muestran que combinar las segmentaciones nos permite disminuir el error. Por esta razón adoptaremos este modelo para continuar perfeccionándolo en las próximas secciones.

7.3.5. Feature Engineering

En esta sección, experimentaremos con dos de las tres variantes de feature engineering ya utilizadas. Estas serán "bueno para familias" y "PBI". En este caso, decidimos no usar "bueno para vacacionar" ya que no creemos que esta variable esté relacionada con el tamaño del inmueble.

Como primer medida, para ver los efectos que producen sobre el predictor, las aplicaremos por separado para luego compararlas.

Para empezar, al igual que como hicimos en la experimentación anterior, agregaremos a los datos de entrada un campo que contendrá el PBI correspondiente a la provincia en la que está ubicado cada inmueble. Nuestro objetivo es poder utilizar la información que aporta tanto la ubicación como la situación económica en la que se encuentra la provincia a la que pertenece la propiedad.

Originalmente creíamos que obtendríamos buenos resultados porque pensamos que la provincia en la que se encuentran las propiedades influía en el tamaño de las mismas. Sin embargo, luego de ver los resultados obtenidos en la sección 7.2.6 y teniendo en cuenta que el PBI agrega información sobre las ubicaciones, no esperamos conseguir mejoras significativas.

Los resultados del experimento se encuentran en la siguiente tabla:

	Modelo Combinado	**Modelo PBI**
RMSE	41.67	40.97
RMSLE	0.2410	0.2398

Cuadro 13: Comparción de errores de Cross Validation para dos modelos distintos.

Ahora, agregaremos la categoría "bueno para familias" desarrollada en la sección 7.2.4. La razón por la cual incluiremos esta columna, es que creemos que una propiedad que sea adecuada para una familia debe ser más espaciosa, es decir, tiene una relación con el tamaño del inmueble.
Creemos que esta adición traerá leves mejoras al predictor debido a lo analizado en la sección 7.2.4.

	Modelo Combinado	**Modelo PBI**	**Modelo Bueno para Familias**
RMSE	41.67	40.97	41.02
RMSLE	0.2410	0.2398	0.2414

Cuadro 14: Comparción de errores de Cross Validation para tres modelos distintos.

Como podemos ver en el cuadro 14 ambas estrategias consiguieron disminuir en pequeña medida los errores de los predictores. Como era de esperarse, "bueno para familias" logró resultados levemente mejores que PBI ya que creemos que la primera se encuentra más relacionada con el tamaño de un inmueble.
Es probable que la mejora introducida por "bueno para familias" haya sido chica por la poca cantidad de avisos que condensa.

Antes de concluir, veremos qué sucede si combinamos estas dos estrategias en un solo modelo.
Una vez más esperamos mejoras leves, no creemos que el error disminuya en mayor medida sabiendo que por separado no obtuvieron grandes resultados.

	Modelo Combinado	**Modelo PBI**	**Modelo Bueno para Familias**	**Modelo PBI + Familia**
RMSE	41.67	40.97	41.02	40.94
RMSLE	0.2410	0.2398	0.2414	0.2395

Cuadro 15: Comparción de errores de Cross Validation para cuatro modelos distintos.

Viendo los resultados del cuadro 15, podemos corroborar que, como era de esperarse, la mejora obtenida al combinar las dos estrategias es prácticamente insignificante. De todos modos, este es el mejor resultado que obtuvimos, tanto en el caso del RMSE, como en el del RMSLE. Por lo cual, tomaremos este modelo como el modelo final.

7.3.6. Conclusiones

Consideramos que la experimentación realizada fue exitosa, ya que logramos reducir el error cuadrático medio base conseguido en la sección 7.3.1 de 46.52 a 40.94 en el modelo final, resultando esta mejora en un 12 %.

Nuevamente, los mejores resultados fueron fruto de segmentar el dataset al haber introducido, aproximadamente, el 87 % de las mejoras totales.

7.4. Conclusiones finales

A lo largo del trabajo, pudimos ver que la mejor estrategia para predecir tanto el precio como la cantidad de metros cubiertos de un inmueble, es realizar segmentaciones sobre el dataset para poder construir funciones más específicas y solucionar en cierta medida el problema de la heterogeneidad del set de datos.

Creemos que mediante otros tipos de feature engineering se pueden llegar a conseguir resultados más efectivos, pero nosotros no logramos obtener una combinación de aspectos sobre los inmuebles que resulten en una mejoría significativa.