**HACIENDO DINERO CON ML: PRECIOS HEDONICOS**

Problem Set 3 – GRUPO 6

Víctor Dulio Chique

Víctor Iván Sánchez

Natalia Castro Alarcon

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. INTRODUCCIÓN:**

En el negocio de reventa de propiedades una predicción acertada de los precios basada en los atributos realizar mejores negociaciones de compra y evita pagar precios sobreestimados maximizando los recursos de inversión. En ese sentido, la teoría de precios hedónicos plantea un problema en la economía del equilibrio espacial donde compradores y vendedores están siendo guiados por los precios implícitos existentes en el espacio de características del bien (Rosen (1974)). Las viviendas son productos diferenciados pues sus atributos les proporcionan utilidad los consumidores. En tanto que, los productores(dueños) incurren en costos que dependen de los atributos asignados a estas viviendas. Entonces la interacción en este mercado entre consumidores y productores determinan la senda de equilibrio del precio de la vivienda.

En este trabajo buscamos predecir los precios de viviendas de la localidad de Chapinero en Bogotá basados en los atributos disponibles. Usamos diferentes modelos de aprendizaje automático que buscan obtener el menor error de predicción posible medido con el Mean Absolute Error (MAE). Exploramos diferentes modelos lineales, modelos de regularización, de árboles de decisión, clusters y super learners con el fin de entender mejor el espacio de predicción. Usamos datos de muestra de 38.644 propiedades individuales de Bogotá recuperadas de <https://www.properati.com.co> las cuales contienen atributos adecuados para la labor de predicción. Después de comparar la capacidad predictiva de los modelos encontramos que el modelo con mejor performance a la hora de predecir el precio de las viviendas es el modelo de *Super Learner* que incluyó *Random Forest* y el modelo lineal, dicho modelo obtuvo un MAE en Kaggle de 203.933.663,18751. Encontramos que Super Learner tiene la capacidad de capturar adecuadamente la no linealidad del modelo y por eso su desempeño superior.

**2. DATOS**

**2.1 Descripción de los Datos**

Con la información de las 38.644 propiedades de la ciudad de Bogotá obtenidos evaluamos atributos tales como; precio, numero de baños, superficie, habitaciones, cuartos, el tipo de propiedad (casa o apartamento), latitud, longitud y una descripción general del bien que proporcionan los vendedores de la propiedad. Las anteriores son características comúnmente evaluadas a la hora de comprar el bien, no obstante se precisa i) Completar de forma adecuada los datos faltantes en cada característica y ii) construir algunas variables espaciales de distancia a los parques, centros comerciales y universidades. Lo anterior nos permitirá tener un modelo lo suficientemente robusto en la labor de predicción.

**2.1.1 Limpieza de datos**

El proceso para limpiar la base de datos y lograr obtener una muestra final de datos completa y consistente, fue retador dada la cantidad de valores perdidos en atributos como baños, habitaciones y en particular del área de las propiedades. Solo un 22,1% de las propiedades tenían información de su área. Teníamos el siguiente porcentaje de datos perdidos para cada variable: 79,6% de *superficie total*, 77.8% de datos perdidos de *superficie cubierta*, 26% de *cuartos* y 26% de *baños*. Respecto a los cuartos y baños optamos por usar el algoritmo de aprendizaje automático KNN “K-vecinos más cercanos” (en ingles “K- Nearest Neighboors”) que nos permite estimar los datos faltantes según la información de los vecinos más cercanos en el espacio de características o atributos. En cuanto a los datos de superficie (tanto “total” como “cubierta”) dada la cantidad significativa valores perdidos se opto primero por recuperar la mayor información posible desde la descripción de los datos (esta era una variable alfa numérica que contenía la descripción de las viviendas hecha por los vendedores) y una vez hecho esto imputar los demás valores con KNN, lo anterior para minimizar el sesgo del modelo.

Para recuperar datos de área, aprovechamos los patrones que vienen en la descripción de la propiedad dado que frecuentemente los vendedores que quieren describir el área de su propiedad lo hacen usando patrones ( por ejemplo; metros cuadrados, m2 o mts). Para lo anterior usamos “*Procesamiento del Lenguaje Natural”* el cual nos permite preprocesar texto de modo que este pueda ser usado como insumo para los modelos de aprendizaje de máquinas que vamos a utilizar en este trabajo. Para poder identificar los patrones de área (metros cuadrados, mts o m2) en el texto de la descripción se hizo uso de *expresiones regulares* conocidas como *regex* o *regexp* que son secuencias de caracteres que nos ayudan a conformar patrones de búsqueda[[1]](#footnote-1) Hecho lo anterior, logramos recuperar de la descripción un 30,2% de los datos de área faltantes. Luego, usamos KNN para imputar el resto de los datos perdidos, asi; aprovechamos la información disponible y evitamos una mayor imputación de datos ( evitando añadir sesgo en nuestro modelo). Finalmente limpiamos la variable área de outliers sacando de esta los valores que estaban fuera de los bigotes, es decir las líneas que se determinan como el tercer cuartil + 1.5 veces el rango intercuartílico y el primer cuartil -1.5 veces el rango intercuartílico.

**2.1.2 Variables espaciales de distancia:**

Usando la distancia euclidiana se crearon 4 predictores que reflejan las cercanía de las propiedades a las áreas comerciales, las universidades, los parques y las avenidas principales. Se utilizó Open Street Map y se estimaron las distancias de los apartamentos a cada uno de esos lugares:

***Distancia a parques*:** se identificaron los polígonos de los parques en Bogotá y sus centroides. Se calcularon las distancias desde los apartamentos hasta estos puntos centrales.

***Distancia a centros comerciales*:** de la librería “*building*” se escogió “*commercial*” y se calculó la distancia de los apartamentos a los áreas comerciales: centros comerciales y supermercados.

***Distancia a universidad*:** distancia de cada apartamento a la universidad más cercana.

***Distancia a avenida principal*:** Corresponden a avenidas como la Circunvalar, Carrera 11, Calle 19, Calle 72 o la Calle 92 que se encontraron utilizando la librería “highway” y la opción “secondary”. Se calculó la menor distancia de cada apartamento a esas avenidas.

Map

Description automatically generated

Map

Description automatically generated

Mapa 1: apartamentos y av principales

Mapa 2: Apartamentos y parques

Los mapas 1 y 2 muestran las principales avenidas (en color rojo) y los polígonos que representan los parques en color verde. Podemos observar que la mayoría de las propiedades de nuestra base están concentradas en el Nor-Oriente de la ciudad, donde el precio de las viviendas es de los mas altos de la ciudad, otro factor que identificamos en los mapas es la ausencia de una muestra significativa de observaciones en Chapinero, barrio para el cual vamos a realizar la predicción. Lo anterior, constituye desafíos para la predicción.

**2.2 Estadísticas Descriptivas**

La Tabla 1 muestra las estadísticas descriptivas de las variables que se utilizaron en el modelo

Table

Description automatically generated

Tabla 1: Estadísticas de las variables

Encontramos que las propiedades tienen un precio promedio de aproximadamente $654 millones, la de menor precio cuesta $ 300 millones y la de mayor $1.250 lo cual muestra una alta variabilidad de los precios ( St. Dev aproximado de $ 311 millones). En promedio una propiedad tiene un área de 112 m2, cuenta con aproximadamente 3 cuartos y 3 baños. Esta información es indicadora de que contamos con variables que en promedio describen bien una propiedad de la ciudad de Bogotá. La Gráfica 1, por su parte, nos muestra la relación entre el precio y el área de una propiedad, encontramos una clara relación positiva y por tanto, propiedades con mayor área tienden a tener un mayor precio.

Text

Description automatically generated

Gráfica 1: Precio - Area

Diagram

Description automatically generated with low confidence

Gráfica 2: Precio y distancias

**3. Modelos y Resultados**

El precio de la vivienda (P) es una función de las características estructurales y los atributos del entorno. Así, estas se representan por un vector de características y atributos , donde denota una de las característica o atributos de la vivienda. Y cada vivienda es denotada por , donde el superíndice indica una propiedad distinta con distinto vector de características y atributos . Por lo tanto, el precio de venta de una vivienda es función de las características y atributos de la misma.

En nuestros modelos, el precio dependerá de un conjunto de características estructurales como número de habitaciones, área de la propiedad, número de baños y tipo de vivienda (casa o apartamento), y adicionalmente de un conjunto de características del entorno como la distancia al parque, distancia al centro comercial, distancia a la universidad y cercanía a avenida principal.

**3.1 Random forest (8 predictores):**

Este tipo de modelos busca realizar las predicciones a través de varios árboles de decisión que se construyen a partir de un subconjunto aleatorio de un número determinado de variables predictoras. En este ejercicio de utilizaron 8 de estas variables: bedrooms, bathrooms, property\_type, superficie, distancia\_parque, distancia\_comercial, distancia\_avenida\_principal, distancia\_universidad. Adicionalmente, el modelo de *Random Forest* se entrena sobre un subconjunto de los datos de la variable de interés en este caso: el precio de las casas y apartamentos de Bogotá. El subconjunto de datos de entrenamiento se obtuvo a través de *cross validation* con k=10. Finalmente, se construyó una grilla con 2, 3, 4 ,5 y 8(bagging) predictores aleatorios. La regla de corte fué “varianza” y los diferentes tamaños de nodos: 1,2,3 y 6.

Como se puede ver en la gráfica 3 a medida que se incrementa el número de predictores disminuye el RMSE, sin embargo después de 3 predictores aleatorios comienza a incrementarse. El mejor número de nodos es 1. El cuadro XX muestra el MEA y MAPE dentro y fuera de muestra.

Chart, line chart

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | MAPE | MAE |
| Entrenamiento | 0.051 | 29403419 |
| Testeo | 0.121 | 68925784 |

Tabla 2: MAE y MAPE

Gráfica 3: RMSE

Aunque el MAPE dentro de muestra es bajo no logra mantener el mismo porcentaje en el conjunto de evaluación. Esto se puede deber a *overfitting*. Sin embargo, aunque la variación entre los dos subconjuntos es evidente, una predicción en testeo con un MAPE del 20% es razonable. El MAE en Kaggle utilizando este modelo fué de 230386934.

**3.2 Arboles de decisión (8 predictores):**

Los árboles de decisión dividen los datos en subconjuntos utilizando particiones recursivas binarias. Se utilizaron 8 variables: bedrooms, bathrooms, property\_type, area, distancia\_parque, distancia\_comercial, distancia\_avenida\_principal, distancia\_universidad. Para el conjunto de entrenamiento y evaluación se utilizó *cross validation* con k=10. Se obtuvo el siguiente modelo:

Diagram

Description automatically generated

Gráfica 4: Arbol de decisión 1

|  |  |  |
| --- | --- | --- |
|  | MAPE | MAE |
| Entrenamiento | 0.217 | 122791972 |
| Testeo | 0.218 | 122787377 |

Tabla 3: MAPE y M

El modelo escoge la variable “Superficie” como la más relevante. Para apartamentos con menos de 97m2 predice un precio de $441´000.000. Para las propiedades con un área entre 97m2 y 122 m2 $580´000.000 y las que tienen más de esa área un costo de $824´000.000. Este es un modelo sencillo que puede resultar práctico para tener una visión general de los precios de acuerdo a una variable relativamente fácil de obtener pero que tiene un error de predicción mayor que el modelo de *Random Forest*. Este último sin embargo no es claro en especificar los predictores más importantes pues es escoge 3 de los 8 posibles de forma aleatoria.

**3.3 Modelos de Regularización**

También se han planteado modelos de regularización para contrastarlo con el mejor modelo. El modelo que se planteo es el siguiente, con distintas especificaciones:

La variable dependiente es el precio de la vivienda (casa o apartamento) medido en pesos colombianos de la ciudad de Bogotá. La primera variable predictora es el tipo de propiedad, es decir si es casa o apartamento, no se le hizo ningún tratamiento pues es una dummy; la otra variable es el área de la propiedad medido en metros cuadrados, datos obtenidos de la descripción del anuncio, luego los datos restantes fueron imputados mediante el algoritmo K-Nearest Neighbors (KNN), considerando los seis vecinos más cercanos; la variable habitaciones resulta del valor máximo entre bedrooms y rooms; la variable baños tuvo 10071 valores faltantes y fue imputada mediante KNN con seis vecinos cercano, y las variables de distancia más cercanas al parque, centro comercial, Av. Principal y universidad fueron calculadas usando información geográfica como las coordenadas en formato latitud y longitud.

Adicionalmente en la base train, mediante la función boxplot se detectaron y excluyeron los outliers de la variable dependiente y las variables independientes, considerando como todo valor que está fuera de los bigotes, es decir las líneas que se determinan como el tercer cuartil + 1.5 veces el rango intercuartílico y el primer cuartil -1.5 veces el rango intercuartílico. En la base de datos test, con el propósito de no perder observaciones, en el caso del área los outliers fueron reemplazados por el promedio y dentro del rango 500 y 30 metros cuadrados y los NAs fueron imputados con KNN. En este grupo de modelos se plantean diversas especificaciones como lineal, semilogarítmico y polinómicos. Los resultados de predicción que mejor se ajustan provienen de una especificación lineal, cuyos coeficientes se muestran en la siguiente tabla:

Modelo de Regresión



Tabla 4: Coeficientes modelo de regresión

El grado de penalización de los modelos de regularización está controlado por el hiperparámetro lambda, se recurre a validación cruzada con 10 folds y los óptimos son los siguientes:



Tabla 5: Hiperparametros

El mejor modelo es el Lasso, porque dentro de la muestra tiene la menor métrica RMSE (161362733), frente al OLS (161555081), Ridge (161709161) y Elestic Net (161698822). Los precios predichos con los predictores de la base test se estima los precios de viviendas que muestran la siguiente distribución:

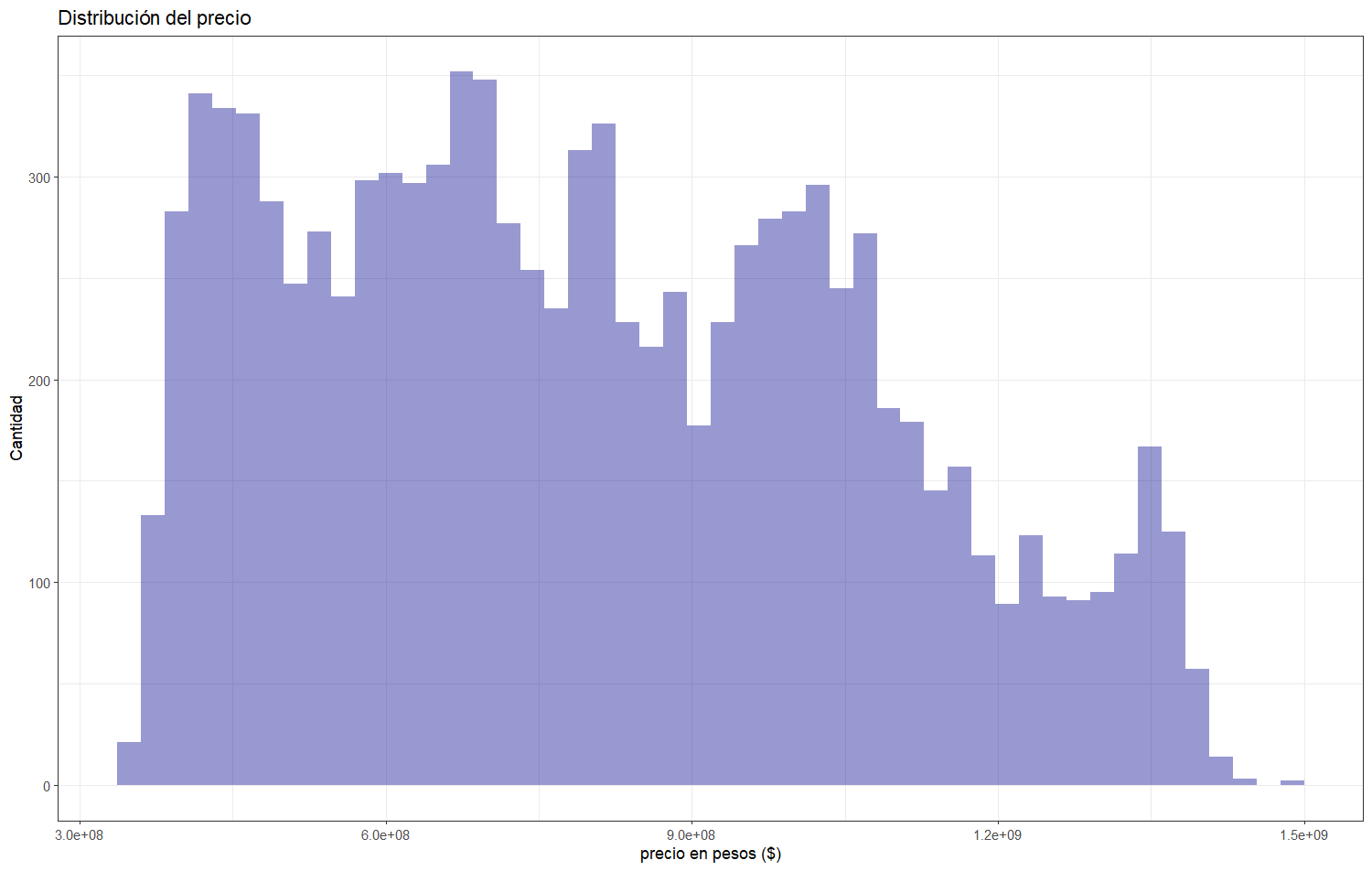


Gráfica 5: Distribución de precios (predichos)

El escore obtenido con dicha predicción del modelo Lasso en kaggle es un MAE de 282445492.70880, que es superado por el modelo Super Learner. Por lo tanto, en este caso, los modelos de regularización no superan las predicciones del modelo Super Learner. Los otros modelos estimados de este grupo son los no lineales, es decir donde el precio está expresado en logaritmos y otro con los predictores en polinomios. Los resultados se muestran en los anexos. El modelo semi-logaritmico, si bien es lo recomendable para reducir la magnitud de la variable dependiente, el problema se da en las predicciones, pues llega a predecir por ejemplo precios por encima de $ 4 mil millones, lo cual empeora la métrica MAE. Por su parte, el modelo con el polinomio de las variables dependientes e interacciones entre ellas, arroja valores predichos del precio negativos para el test, lo que supone un MAE peor fuera de la muestra.

**3.4 Super Learner**

Super Learner es un algoritmo de aprendizaje automático que combina múltiples modelos con el objetivo de mejorar la precisión y desempeño general del modelo. Para aplicar este modelo aprovechamos la muestra final de datos, limpiada e imputada de la forma mas óptima que encontramos y se procedio a estimar el siguiente modelo:



Gráfica 6: Distribución de precios (predichos) para Super Learner

Para la predicción mediante el superlearner incluimos los siguientes Hiper-parametros seleccionados fueron, respecto a los modelos; **Random Forest** y **Regresión Lineal**; y como técnica de optimización usamos **Non Linear Least Squares (NLLS**). La Grafica 6. Muestra la distribución de precios de la predicción, esta distribución nos muestra que la mayoría de propiedades se encuentran contendidas en un rango de precios de entre $574 millones y $1.010 millones aproximadamente, con predicciones máximas de 1.478 millones y mínimas de $338 millones, con un promedio de precios predicho de $ 801 millones aproximadamente. Así mismo el modelo arrojó los siguientes coeficientes:



Tabla 6. Coeficientes de SuperLearner

Los coeficientes indican que Random Forest (RF) se lleva todo el peso dentro de la combinación de ambos modelos, esto se debe a que claramente RF esta capturando las no linealidades que tiene el modelo dada la complidad introducida a este. El modelo nos arrojo un MAE de 203.933.663,18751 resultando en el que mejor predice de todos los cosntruidos en este trabajo.

**3. Conclusiones y Recomendaciones**

Encontramos que un predictor preponderante en la performance del modelo era el área, por lo que fue vital poder recuperar la mayor cantidad de esta información dado que este atributo es de los mas discutidos a la hora de hacer negocios inmobiliarios, por lo que debe siempre incluirse y recuperarse de manera adecuada en los conjuntos de datos.

Asi mismo, en los problemas de predicción es importante explorar diferentes modelos de aprendizaje automatizado pues no es posible conocer la forma exacta de la función de predicción desde un principio. Después de explorar los modelos expuestos se concluye que el *Super Learn*er es el modelo más adecuado para predecir los precios de Chapinero en Bogotá. *Super Learner* permite confirmar la importancia de capturar las no linealidades con modelos que utilizan particiones recursivas binarias al darle el mayor peso al modelo de predicción *Random Forest.* Lo anterior dado que, como se observo en la descripción de los modelos y resultados, cada modelo tiene sus debilidades y fortalezas, es quizá por esta razón que Super Learner, al combinar varios modelos, permite las fortalezas de cada uno y minimizar las debilidades de los otros, y de esta manera tener una visión mas completa del conjunto de datos.

Concluimos que al incluir las variables de latitud, longitud y distancias, los modelos de árboles y en especial *Random Forest* realizan particiones que un modelo lineal sólo podría realizar creando muchas variables *dummies*.

El proceso de depuración de las variables también influye enormemente en una correcta predicción. A través de diferentes intentos en Kaggle se observaron mejoras en la predicción cada vez que se utilizaban métodos de imputación más complejos respecto a los usuales así como la inclusión o no de los *outliers*. De igual manera la creación de las variables espaciales y la obtención de la superficie de las propiedades inciden en una predicción más precisa.

**4. Bibliografía**

Rosen, S. (1974) Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition

Lucas Antonio, Valentina e Ignacio. *“Tutorial de Procesamiento Natural de Texto”.*

Anexos

**Modelo Lasso con variable precio en logaritmos**

train <- Bogota\_train %>%

select(lprice, property\_type, rooms\_tot, bathrooms, distancia\_avenida\_principal,

distancia\_comercial, distancia\_parque, distancia\_universidad, rooms\_bath,

dist\_av2, dist\_com2, dist\_parq2, dist\_univ2)

Lambda optimo: 9.69804377996242e-05"

"Error (mse) de entrenamiento: 0.0874021221908704"Chart

Description automatically generated

"Error (mae) de entrenamiento: 0.239060247990877"

Chart, histogram

Description automatically generated

**Modelo Lasso con polinomios**

OLS RIDGE LASSO ELASTIC\_NET

(Intercept) 591372140.6 591372140.6 591372140.59 591372140.6

property\_typeCasa -30761762.9 -27671041.9 -30468958.28 -30353812.9

`poly(Superficie, 2)1` 91779606.0 59134294.2 89308622.98 88606128.6

`poly(Superficie, 2)2` -85156464.6 -47340141.7 -78480095.63 -76386494.9

rooms\_tot -29503545.3 -17639518.3 -28165170.91 -27587415.6

bathrooms 54106319.6 49748845.1 53703186.13 53654590.6

`poly(Superficie, 2)1:rooms\_tot` 50964122.5 50675645.2 52044514.67 52994058.6

`poly(Superficie, 2)2:rooms\_tot` 19389049.8 2427501.8 18100666.15 16456599.7

`poly(Superficie, 2)1:bathrooms` 10180131.7 36947714.1 11116079.36 10612037.0

`poly(Superficie, 2)2:bathrooms` 5628467.7 -13004017.4 373812.81 0.0

`bathrooms:poly(distancia\_parque, 2)1` 143694.6 615552.4 11128.64 0.0

`bathrooms:poly(distancia\_parque, 2)2` -1261591.7 -1399073.8 -989937.24 -873658.9

`bathrooms:poly(distancia\_comercial, 2)1` -7522750.9 -7529736.1 -7337572.18 -7237881.6

`bathrooms:poly(distancia\_comercial, 2)2` -8690855.3 -8449249.3 -8515198.76 -8446796.2

`bathrooms:poly(distancia\_avenida\_principal, 2)1` 9010529.7 8554239.6 8684796.39 8557105.4

`bathrooms:poly(distancia\_avenida\_principal, 2)2` -2948503.2 -3008093.2 -2752174.13 -2678022.8

`bathrooms:poly(distancia\_universidad, 2)1` -1167006.3 -1371331.4 -942099.92 -846661.4

`bathrooms:poly(distancia\_universidad, 2)2` -4901361.6 -5165554.3 -4726439.11 -4658935.7

OLS RIDGE LASSO EN

RMSE 151007264 151577050 150950062 151591374

Chart, histogram

Description automatically generated

1. Lucas Antonio, Valentina e Ignacio. *“Tutorial de Procesamiento Natural de Texto”.*  [↑](#footnote-ref-1)