Predicción Precio de Venta en Viviendas

Por:
Juan José Toro Villegas
Julián Vargas Uribe
Arthur Jose Mosquera Franco

Materia:
Introducción a la Inteligencia Artificial para las Ingenierías

Profesor:
Raúl Ramos Pollan

Universidad de Antioquia
Facultad de Ingeniería
Medellín 2023

Contenido

Descripción del Problema

Zillow es una compañía tecnológica que opera como un mercado para bienes raíces en línea. La compañía fue fundada en el 2006 por Rich Barton, Lloyd Frink y Spencer Rascoff en Estados Unidos, y tiene como modelo de negocio la venta de anuncios de viviendas en su sitio web y ahora en su aplicativo. Allí puedes comprar, rentar y vender propiedades. (Gutiérrez, 2021. Con información de Wikipedia y Motley Fool).

Los "Zestimates" son valores estimados de viviendas basados en 7,5 millones de modelos estadísticos y de aprendizaje automático que analizan cientos de puntos de datos en cada propiedad. Y, al mejorar continuamente el margen de error promedio (del 14 % al principio al 5 % en la actualidad), Zillow se ha establecido desde entonces como uno de los mercados más grandes y confiables para la información de bienes raíces en los EE. UU. Por tanto, es indispensable mejorar el cálculo y ese margen de error promedio para así contar con una mayor exactitud y en menor tiempo a la hora de hablar de valor de bienes raíces. (Andrew Martin, Bin, Cat N, K Nielsen, Maggie, Wendy Kan, Zillow Prize: Zillow's Home Value Prediction (Zestimate) publicado en Kaggle, 2017).

Dataset

Vamos a usar el dataset de Kaggle de esta competición (https://www.kaggle.com/competitions/zillow-prize-1/data), que tiene 1,048,576 número de muestras y las siguientes columnas:

Feature	Description
'airconditioningtypeid'	Type of cooling system present in the home (if any)
'architecturalstyletypeid'	Architectural style of the home (i.e. ranch, colonial, split-level, etc…)
'basementsqft'	Finished living area below or partially below ground level
'bathroomcnt'	Number of bathrooms in home including fractional bathrooms

'bedroomcnt'	Number of bedrooms in home
'buildingqualitytypeid'	Overall assessment of condition of the building from best (lowest) to worst (highest)
'buildingclasstypeid'	The building framing type (steel frame, wood frame, concrete/brick)
'calculatedbathnbr'	Number of bathrooms in home including fractional bathroom
'decktypeid'	Type of deck (if any) present on parcel
'threequarterbathnbr'	Number of 3/4 bathrooms in house (shower + sink + toilet)
'finishedfloor1squarefeet'	Size of the finished living area on the first (entry) floor of the home
'calculatedfinishedsquarefeet'	Calculated total finished living area of the home
'finishedsquarefeet6'	Base unfinished and finished area
'finishedsquarefeet12'	Finished living area
'finishedsquarefeet13'	Perimeter living area
'finishedsquarefeet15'	Total area
'finishedsquarefeet50'	Size of the finished living area on the first (entry) floor of the home
'fips'	Federal Information Processing Standard code - see https://en.wikipedia.org/wiki/FIPS_county_code for more details
'fireplacecnt'	Number of fireplaces in a home (if any)

'fireplaceflag'	Is a fireplace present in this home
'fullbathcnt'	Number of full bathrooms (sink, shower + bathtub, and toilet) present in home
'garagecarcnt'	Total number of garages on the lot including an attached garage
'garagetotalsqft'	Total number of square feet of all garages on lot including an attached garage
'hashottuborspa'	Does the home have a hot tub or spa
'heatingorsystemtypeid'	Type of home heating system
'latitude'	Latitude of the middle of the parcel multiplied by 10e6
'longitude'	Longitude of the middle of the parcel multiplied by 10e6
'lotsizesquarefeet'	Area of the lot in square feet
'numberofstories'	Number of stories or levels the home has
'parcelid'	Unique identifier for parcels (lots)
'poolcnt'	Number of pools on the lot (if any)
'poolsizesum'	Total square footage of all pools on property
'pooltypeid10'	Spa or Hot Tub
'pooltypeid2'	Pool with Spa/Hot Tub
'pooltypeid7'	Pool without hot tub
'propertycountylandusecode'	County land use code i.e. it's zoning at the county level

'propertylandusetypeid'	Type of land use the property is zoned for
'propertyzoningdesc'	Description of the allowed land uses (zoning) for that property
'rawcensustractandblock'	Census tract and block ID combined - also contains blockgroup assignment by extension
'censustractandblock'	Census tract and block ID combined - also contains blockgroup assignment by extension
'regionidcounty'	County in which the property is located
'regionidcity'	City in which the property is located (if any)
'regionidzip'	Zip code in which the property is located
'regionidneighborhood'	Neighborhood in which the property is located
'roomcnt'	Total number of rooms in the principal residence
'storytypeid'	Type of floors in a multi-story house (i.e. basement and main level, split-level, attic, etc.). See tab for details.
'typeconstructiontypeid'	What type of construction material was used to construct the home
'unitcnt'	Number of units the structure is built into (i.e. 2 = duplex, 3 = triplex, etc...)
'yardbuildingsqft17'	Patio in yard
'yardbuildingsqft26'	Storage shed/building in yard
'yearbuilt'	The Year the principal residence was built

'taxvaluedollarcnt'	The total tax assessed value of the parcel
'structuretaxvaluedollarcnt'	The assessed value of the built structure on the parcel
'landtaxvaluedollarcnt'	The assessed value of the land area of the parcel
'taxamount'	The total property tax assessed for that assessment year
'assessmentyear'	The year of the property tax assessment
'taxdelinquencyflag'	Property taxes for this parcel are past due as of 2015
'taxdelinquencyyear'	Year for which the unpaid propert taxes were due

Objetivo

Desarrollar un algoritmo que haga predicciones sobre los precios de venta futuro de las viviendas, apoyadas en datos de transacciones de bienes raíces que son información pública del año 2016 y 2017.

Métricas

Como métrica de Machine Learning usaremos el MAE (Mean Absolute Error) entre el error de registro previsto y el error de registro real. Esto para facilitar y cuantificar la precisión del modelo, el cual se espera que tenga un porcentaje de acierto alto y que a su vez se vea reflejado en la cantidad de personas que usan la aplicación a la hora de hablar de bienes raíces.

El error de registro definido así:

$$log_{error} = log_{Zestimate} - log_{precio\ venta}$$

Como métrica de negocio, gracias a la utilización del modelo, esperamos un incremento de al menos un 35% en ventas ya que, debido a las predicciones acertadas de la aplicación y página web, los clientes gozarán de un nivel de confianza mayor, a la hora de usar e invertir en Zillow.

Criterio

Lo que esperamos de la implementación del modelo es que este de cifras muy cercanas a la realidad del precio de las viviendas, que los americanos puedan apoyarse también del modelo a la hora de tomar una decisión de comprar, rentar o vender una propiedad y es realmente lo que se quiere al usar Zillow.

La variable que más valor tiene para una organización son las ventas (que tanto incrementan); sin embargo, para un primer momento y como decisión tomada de la junta directiva, la variable que inicialmente evaluará al modelo en los primeros meses será el número de visitas mensuales tanto del aplicativo como de la página web en las ciudades más importantes de EE. UU. Esto no quiere decir que se dejaran de lado el crecimiento de ventas planteado, solo que pasa a un segundo plano en los primeros meses de implementación.

Exploración Descriptiva del Dataset

En esta etapa inicial del proyecto, se importaron los datos de Zillow Prize, los cuales se dividieron en cuatro archivos: Train de los años 2016 y 2017, y properties de los años 2016 y 2017. Para facilitar el análisis, se concatenaron los archivos correspondientes a cada año, creando dos conjuntos de datos, uno para el año 2016 y otro para el año 2017.

train2016

	parcelid	logerror	transactiondate
0	11016594	0.0276	2016-01-01
1	14366692	-0.1684	2016-01-01
2	12098116	-0.0040	2016-01-01
3	12643413	0.0218	2016-01-02
4	14432541	-0.0050	2016-01-02
...	...	...	...
90270	10774160	-0.0356	2016-12-30
90271	12046695	0.0070	2016-12-30
90272	12995401	-0.2679	2016-12-30
90273	11402105	0.0602	2016-12-30
90274	12566293	0.4207	2016-12-30

90275 rows × 3 columns

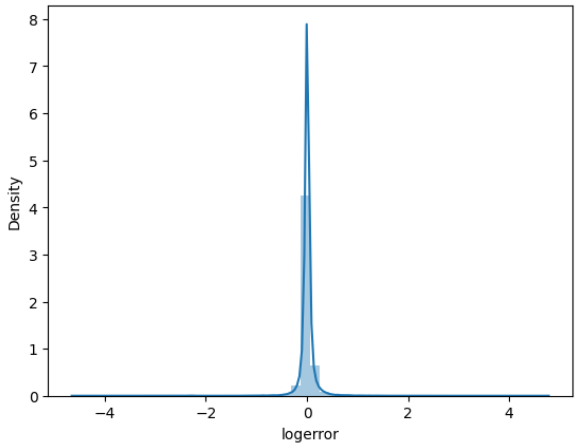
train2017

	parcelid	logerror	transactiondate
0	14297519	0.025595	2017-01-01
1	17052889	0.055619	2017-01-01
2	14186244	0.005383	2017-01-01
3	12177905	-0.103410	2017-01-01
4	10887214	0.006940	2017-01-01
...	...	...	...
77608	10833991	-0.002245	2017-09-20
77609	11000655	0.020615	2017-09-20
77610	17239384	0.013209	2017-09-21
77611	12773139	0.037129	2017-09-21
77612	12826780	0.007204	2017-09-25

77613 rows × 3 columns

Exploración Inicial

Se realizó una exploración inicial de los datos para comprender su estructura y características. Se examinaron las variables, incluyendo el conteo de datos, distribuciones, promedios, correlaciones con la variable objetivo (Log Error), histogramas, desviaciones estándar, valores mínimos y máximos, y tipo de datos de cada variable. En particular, se observó que la variable objetivo "Logerror" se distribuye normalmente con un sesgo bajo de 3,70.



Datos Faltantes

Uno de los desafíos encontrados en el proyecto fue lidiar con los datos faltantes. Se identificó que muchas variables tenían un alto porcentaje de datos faltantes. Se decidió depurar las variables eliminando aquellas columnas que tenían un porcentaje de datos faltantes mayor al 70%. Esto resultó en la eliminación de aproximadamente el 60% de las columnas iniciales. Para manejar los datos faltantes restantes, se optó por llenarlos con el promedio de la respectiva columna.

basementsqft	1282	67.723191
buildingclasstypeid	1282	67.723191
finishedsquarefeet13	1282	67.723191
fireplaceflag	1281	67.670365
architecturalstyletypeid	1280	67.617538
typeconstructiontypeid	1280	67.617538
finishedsquarefeet6	1279	67.564712
pooltypeid10	1276	67.406233
decktypeid	1272	67.194929
poolsizesum	1268	66.983624
pooltypeid2	1260	66.561014
hashottuborspa	1254	66.244057
finishedsquarefeet15	1235	65.240359
taxdelinquencyyear	1227	64.817750
taxdelinquencyflag	1227	64.817750

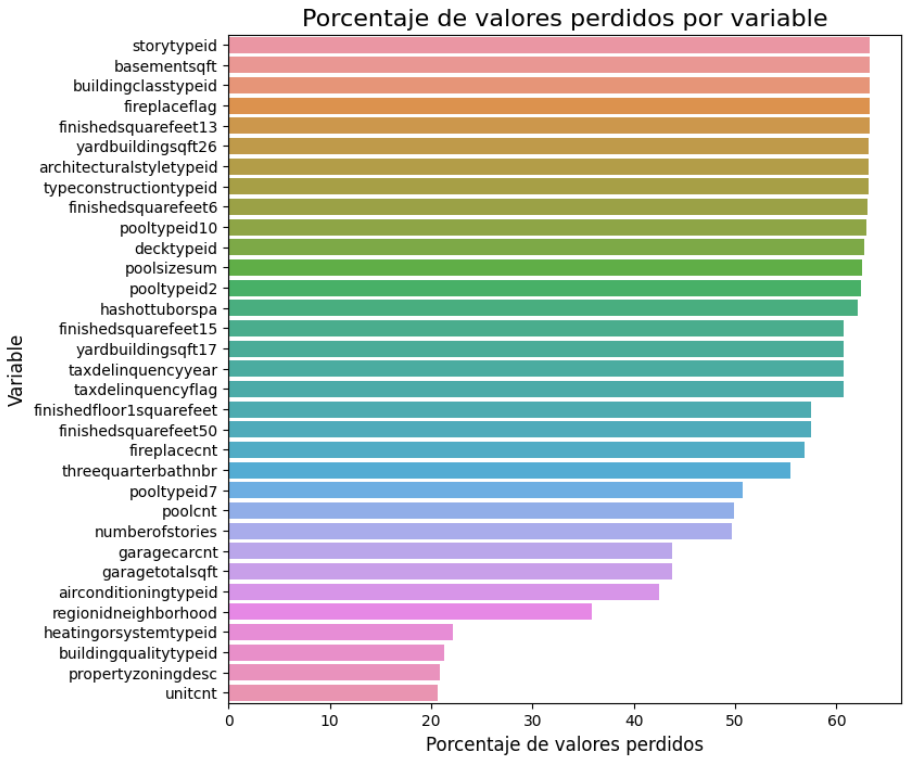


Grafico 1. Barras

El porcentaje de valores faltantes puede ser útil para identificar columnas con una cantidad significativa de datos faltantes. En este caso, las columnas con un porcentaje alto de valores faltantes incluyen architecturalstyletypeid (99.71%), basementsqft (99.95%), buildingclasstypeid (99.98%), decktypeid (99.27%), finishedfloor1squarefeet (92.41%), finishedsquarefeet13 (99.96%),

finishedsquarefeet15 (96.05%), finishedsquarefeet50 (92.41%), finishedsquarefeet6 (99.53%), fireplacecnt (89.36%), garagecarcnt (66.84%), garagetotalsqft (66.84%), hashottuborspa (97.38%), poolcnt (80.17%), poolsizesum (98.93%), pooltypeid10 (98.71%), pooltypeid2 (98.67%), pooltypeid7 (81.50%), regionidneighborhood (60.11%), storytypeid (99.95%), threequarterbathnbr (86.70%), typeconstructiontypeid (99.67%), yardbuildingsqft17 (97.07%), yardbuildingsqft26 (99.89%), numberofstories (77.21%), fireplaceflag (99.75%), taxdelinquencyflag (98.02%), y taxdelinquencyyear (98.02%).

Al comparar estas salidas con las del conjunto de datos train_2016, se observa que hay similitudes en algunas columnas con altos porcentajes de valores faltantes, como architecturalstyletypeid, basementsqft, buildingclasstypeid, decktypeid, finishedfloor1squarefeet, finishedsquarefeet13, finishedsquarefeet15, finishedsquarefeet50, finishedsquarefeet6, fireplacecnt, garagecarcnt, garagetotalsqft, hashottuborspa, poolcnt, poolsizesum, pooltypeid10, pooltypeid2, pooltypeid7, regionidneighborhood, storytypeid, threequarterbathnbr, typeconstructiontypeid, yardbuildingsqft17, yardbuildingsqft26, numberofstories, fireplaceflag, taxdelinquencyflag, y taxdelinquencyyear.

Transformación de Variables

Se observó que la variable objetivo "Logerror" no debía contener valores que fueran cero, ya que esto podría afectar la predicción del modelo. Por lo tanto, se tomaron medidas para asegurarse de que no haya valores cero en la variable objetivo.

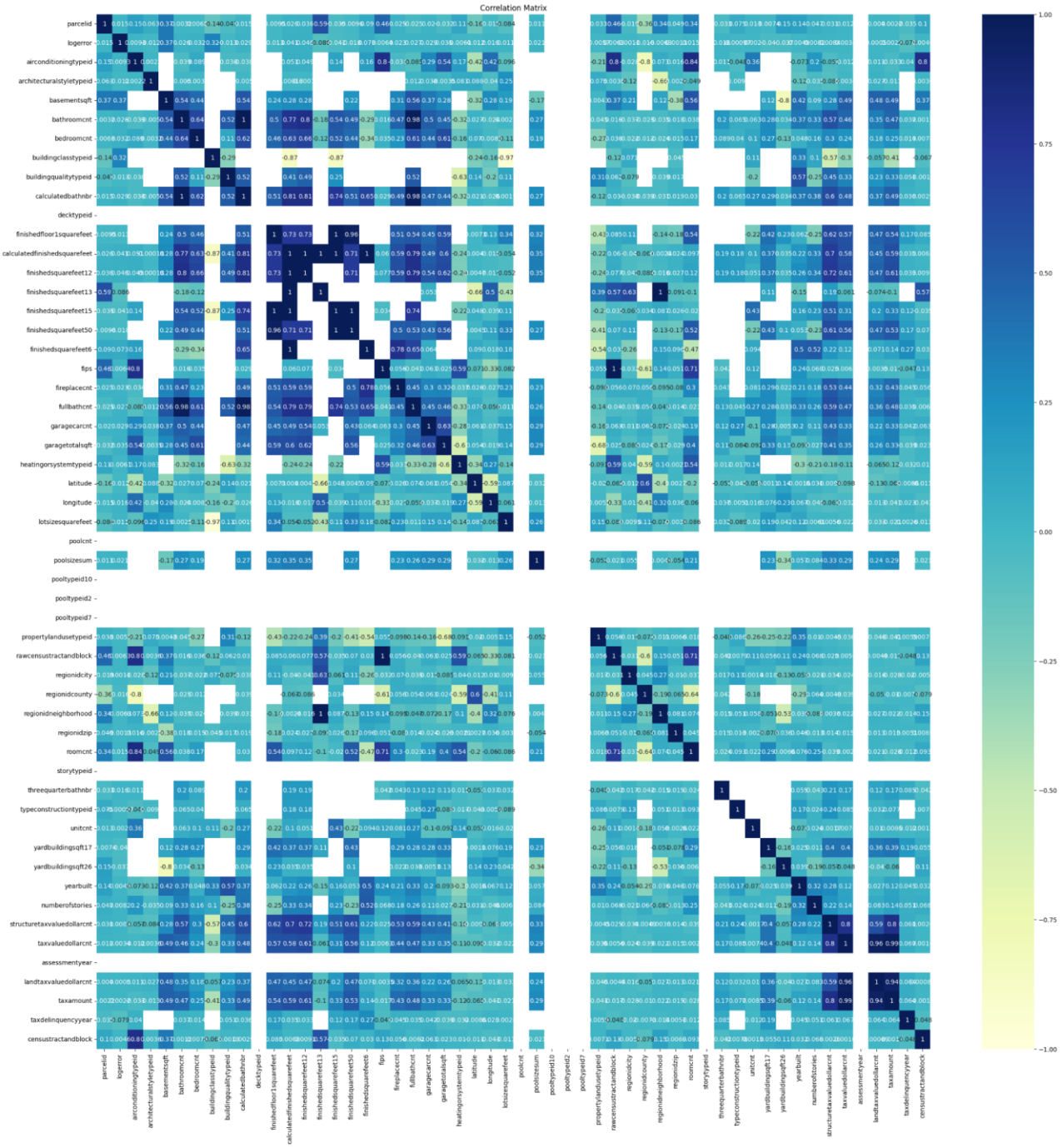
Además, se trabajó en la transformación de variables categóricas (tipo "object") para obtener más información al analizarlas. Estas variables fueron procesadas y transformadas de manera que proporcionen datos más útiles para el análisis. Al finalizar esta etapa, se obtuvo una distribución de tipos de variables en el conjunto de datos del año 2017, con 25 variables de tipo float64 y 9 variables de tipo int64.

Información Limitada del Año 2016

Se notó que el conjunto de datos correspondiente al año 2016 proporcionaba información limitada. Este conjunto presentaba una gran cantidad de datos faltantes en comparación con el año 2017. Esto planteó un desafío adicional, ya que la falta de información puede afectar la calidad de las predicciones a realizar y el desempeño del uso de la base de datos.

Correlación entre las diferentes variables

Al analizar una matriz de correlación, se pueden identificar patrones y relaciones entre variables y la variable regresora. Por ejemplo, si hay una correlación positiva fuerte entre el número de habitaciones y el precio de venta, esto podría indicar que a medida que el número de habitaciones aumenta, el precio de venta tiende a ser más alto. Por otro lado, si hay una correlación negativa entre la ubicación y el precio de alquiler, esto podría sugerir que las propiedades en ubicaciones menos deseables tienen tendencia a tener precios de alquiler más bajos.



Carga de datos:

Se cargó el conjunto de entrenamiento "train_2016" que contiene datos relacionados con propiedades inmobiliarias.
El conjunto de datos se compone de varias columnas que representan diferentes características de las propiedades, como ubicación, tamaño, número de habitaciones, etc.

Se imprimió una descripción básica de los datos para obtener una visión general, incluyendo el número de filas y columnas presentes en el conjunto de datos. Además, se mostró una vista previa de las primeras filas para tener una idea de los valores almacenados en cada columna.
Análisis de valores nulos:

Se realizó una verificación para identificar la presencia de valores nulos en el conjunto de datos. Se generó una lista de las columnas que contienen valores nulos y la cantidad de valores nulos en cada columna. Esta información es útil para comprender la integridad de los datos y determinar si es necesario realizar imputación de valores o eliminar filas/columnas con valores nulos.

Análisis de la columna "logerror":

Se examinó la columna "logerror", que contiene los errores de registro de los precios de las propiedades.
Se proporcionó una descripción estadística de los valores en esta columna, incluyendo el conteo total de valores no nulos, la media, la desviación estándar, el valor mínimo y máximo, así como los percentiles (25%, 50%, y 75%).
Esta descripción estadística nos permite comprender la distribución de los errores de registro y obtener información sobre la dispersión de los valores.
Visualización de la distribución de "logerror":

Se trazó un histograma para visualizar la distribución de los valores de "logerror".
El histograma proporciona una representación gráfica de la distribución de los errores de registro. Permite identificar patrones en la distribución, como si está sesgada hacia un lado o si hay valores atípicos prominentes.

Filtrado de valores atípicos en "logerror":

Se aplicó un filtro para identificar y eliminar los valores atípicos en la columna "logerror" utilizando el método de los percentiles. Se utilizó el rango intercuartílico (IQR) para determinar los límites superior e inferior para identificar los valores atípicos. Los valores por encima del percentil 75 + (1.5 * IQR) o por debajo del percentil 25 - (1.5 * IQR) se consideraron atípicos y se eliminaron del conjunto de datos. Se imprimió el número de valores atípicos identificados y se creó una nueva serie de datos sin esos valores para futuros análisis.

Visualización de la distribución sin valores atípicos:

Se trazó un histograma para visualizar la distribución de los valores de "logerror" después de eliminar los valores atípicos. Esta visualización nos ayuda a comprender cómo se modificó la distribución después de eliminar los valores atípicos y si la distribución se ajusta mejor a una forma esperada.
Identificación de índices con logerror igual a cero

Preprocesamiento de datos:

Limpieza de datos: Se verificó y se manejaron los valores faltantes, se eliminaron registros con datos inconsistentes o errores evidentes. Esto puede incluir técnicas como el reemplazo de valores faltantes, la eliminación de registros incompletos o la imputación de valores faltantes basada en alguna estrategia.
Transformación de variables: Se realizaron transformaciones en las variables para asegurar que estén en la misma escala o para convertirlas en un formato adecuado para los modelos. Esto puede incluir técnicas como la normalización o estandarización de variables numéricas y la codificación de variables categóricas.
Selección de características: Se identificaron las características más relevantes para predecir los precios de Zillow y se eliminaron características irrelevantes o redundantes. Esto puede incluir técnicas como el análisis de correlación, pruebas estadísticas o técnicas de selección de características basadas en modelos.
División de datos: Se dividieron los datos en conjuntos de entrenamiento, validación y prueba, asegurándose de mantener una distribución adecuada de los precios en cada conjunto. Esto permite evaluar el rendimiento del modelo de manera más precisa y evitar el sobreajuste.

Creación de modelos:

Selección del algoritmo: Se seleccionó un algoritmo de regresión adecuado para predecir los precios de Zillow. Algunos algoritmos comunes podrían ser la regresión lineal, regresión de árboles de decisión, regresión de bosques aleatorios, etc.
Configuración del modelo: Se definieron los hiperparámetros del modelo, como la regularización, el número de estimadores en un modelo de bosque aleatorio, etc. Estos hiperparámetros se ajustan para optimizar el rendimiento del modelo.

Entrenamiento del modelo: Se alimentaron los datos de entrenamiento al modelo y se ajustaron los parámetros para que el modelo pueda aprender a predecir los precios. Se utilizó algún algoritmo de entrenamiento adecuado, como el descenso de gradiente estocástico (SGD) o métodos de optimización más avanzados.
Evaluación del modelo: Se utilizaron datos de validación o prueba para evaluar el rendimiento del modelo en términos de métricas de regresión, como el error medio cuadrado (MSE), el error absoluto medio (MAE) o el coeficiente de determinación (R^2).

Esto permite medir qué tan bien se ajusta el modelo a los datos y su capacidad para hacer predicciones precisas.

Ajuste y optimización: Se realizaron ajustes adicionales en el modelo, como la optimización de hiperparámetros utilizando técnicas como la búsqueda en cuadrícula (grid search) o la validación cruzada, para mejorar el rendimiento de las predicciones. Esto implica probar diferentes combinaciones de hiperparámetros y evaluar el rendimiento del modelo en cada caso.

Modelos

Inicial

Nota: En el notebook de Colab, asegurarse de correr los códigos previos, además es fundamental declarar todas las librerías a medida que el notebook avance.

Como se dijo antes como métrica de Machine Learning usaremos el MAE (Mean Absolute Error) entre el error de registro previsto y el error de registro real. Esto para facilitar y cuantificar la precisión del modelo, el cual se espera que tenga un porcentaje de acierto alto y que a su vez se vea reflejado en la cantidad de personas que usan la aplicación a la hora de hablar de bienes raíces.

Tenemos dos datasets, ambos con 35 columnas, sin embargo los datos del año 2016 son 90275 y los del año 2017 son 77613, es decir 12662 datos de diferencia. Por tanto, inicialmente en el código con la variable “min_rows” se calculó el número mínimo de datos entre ambos y posteriormente se calcula un subconjunto de filas en el dataset más grande (2016). Así aseguramos o más bien, igualamos el número de entradas para iniciar el/los modelos posteriores.

Calculamos el MAE, en español Error Absoluto Medio:

```
Mean Absolute Error (MAE): 0.11307266377047162
```

Esto como primera métrica de error, que indica que en promedio, el error absoluto entre las predicciones y los valores reales de los errores de registro es de aproximadamente 0,1130.

Exploración de modelos

Regresión lineal

Se separaron los datos en entradas o características X y en la variable objetivo Y que como dijimos es “logerror”.

```
X_2016 = train_2016.drop('logerror', axis=1)
y_2016 = train_2016['logerror']
X_2017 = train_2017.drop('logerror', axis=1)
y_2017 = train_2017['logerror']
```

Como modelo de Machine Learning adicional, usamos la regresión lineal la cual asume una relación lineal entre las variables de entrada y la variable objetivo. Y es así entonces como calculamos el 2do MAE, “mae_linear”.

```
MAE - Regresión Lineal: 0.07047456750960691
```

Random Forest

Utilizamos la biblioteca “XGBoost” para entrenar el modelo de bosque aleatorio. Creamos objetos “DMatrix” para los conjuntos de entrenamiento y prueba, configuramos los parámetros del modelo y luego entrenamos el modelo utilizando la función “xgb.train()”. Posteriormente, realizamos predicciones en el dataset 2017 y calculamos el MAE-Bosque Aleatorio.

```
MAE - Bosque Aleatorio (XGBoost): 0.07342472678372643
```

Y finalmente comparamos todos los MAE hasta ahora obtenidos, y recopilamos los resultados:

```
# Comparar las variables y encontrar el menor
menor = min(mae, mae_linear, maeBA)
```

```
El menor valor es: 0.07047456750960691 mae linear
```

Y nos damos cuenta de que hasta ahora el MAE de la Regresión Lineal, es el mas bajo hasta ahora con aproximadamente 0,0704.

Validación Cruzada

Importamos “cross_val_score” de la librería “sklearn.model_selection”
Después, se creó el modelo de regresión lineal utilizando “LinearRegression”. A continuación, se realiza la validación cruzada utilizando “cross_val_score” con 5 folds (cv=5) y se especifica la métrica de evaluación como el negativo del MAE. Los scores negativos se usan ya que la función “cross_val_score” maximiza los valores en lugar de minimizarlos.

Finalmente, se calcula el MAE promedio a partir de los scores negativos, y se imprime el resultado como el MAE de validación cruzada, el cual es:

```
MAE - Validación Cruzada: 0.06998102060651418
```

Posteriormente, comparamos el modelo con y sin la validación cruzada:

```
menorr=min(mae_linear,mae_cv)
```

```
0.06998102060651418 MAE con validacion cruzada
```

Y vemos que con 0,0699 logro disminuir el MAE al aplicar la validación cruzada.

Curvas de Aprendizaje

se utiliza la función learning_curve de scikit-learn para calcular las curvas de aprendizaje. Se especifica el modelo de regresión lineal, los datos de características (X) y variable objetivo (y), los tamaños de entrenamiento (train_sizes), la métrica de evaluación como el negativo del MAE (scoring='neg_mean_absolute_error'), y el número de folds de validación cruzada (cv=5).

Luego, se calculan los valores promedio y desviaciones estándar de los scores tanto para el entrenamiento como para la validación. Estos valores se utilizan para plotear las curvas de aprendizaje, donde se muestra cómo evoluciona el MAE en función del tamaño del conjunto de entrenamiento.
El gráfico resultante muestra las curvas de aprendizaje para el conjunto de entrenamiento:

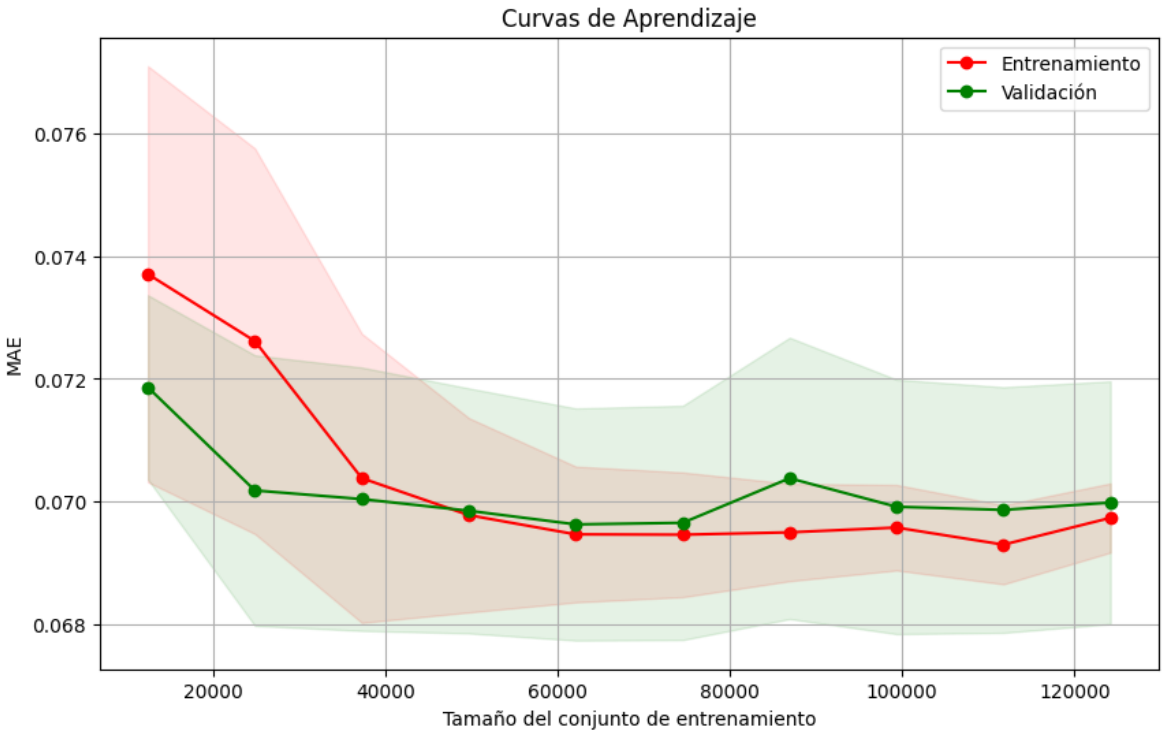


Gráfico 2. Curvas de Aprendizaje

Métodos Supervisados

Usamos el método de regresión Ridge, el cual es un método de regularización que pueden ayudar a mejorar el rendimiento del modelo de regresión lineal al agregar una penalización por la complejidad del modelo.
Se importa la clase Ridge del módulo “linear_model” de la biblioteca “scikit-learn”, creamos y entrenamos el modelo, durante el entrenamiento, el modelo ajusta los coeficientes de la regresión para minimizar la función de pérdida.
Una vez que el modelo está entrenado, utilizamos el método “predict” para hacer predicciones en nuevos datos. Aquí se utilizan las características X_2016 del conjunto de datos de 2016 para predecir los valores correspondientes. Las predicciones se asignan a la variable y_pred_2016 e igualmente para el 2017. Finalmente, se calcula el Error Absoluto Medio (MAE).

```
MAE - Regresión Ridge (2016): 0.06878446828711116
MAE - Regresión Ridge (2017): 0.07047450535006293
```

Métodos No Supervisados
K-means y clustering
Aplica el algoritmo de K-means clustering con 2 clusters a los datos de características X. Luego, utilizamos PCA para reducir la dimensionalidad de X a 2 componentes principales y muestra los resultados de las etiquetas de los clusters y los componentes principales. Teniendo como resultados de la impresión:

```
Etiquetas de los clusters:
[0 0 0 ... 0 0 0]
Componentes principales:
[[-1.22696031e+11 -1.95692434e+06]
 [-1.17730490e+11  1.40732113e+06]
 [-1.19124489e+11 -8.75453799e+05]
 ...
 [ 6.16321530e+11  4.07976265e+06]
 [-1.19424470e+11 -1.99704321e+05]
 [-1.18728471e+11 -1.43347606e+05]]
```

Retos y condiciones de despliegue

Mínimamente para que un modelo sea usado o tenido en cuenta, su desempeño debe mostrar o significar un beneficio para el usuario de este a futuro, sin dejar de lado que variables son las que más peso tienen a la hora de predecir el precio de una vivienda. Se sabe que elegir las características adecuadas para el modelo puede ser un desafío. Se debe realizar un análisis muchos más exhaustivo de las características disponibles y seleccionar aquellas que tengan un impacto significativo en el precio de las viviendas considerando la posibilidad de utilizar características adicionales, como información política que tenga alguna relación según la ubicación ya sea tema de leyes, entre otros y/o datos económicos.

La actualización y mantenimiento del modelo es clave, los precios de las viviendas pueden cambiar con el tiempo debido a factores económicos, cambios en la demanda del mercado u otras condiciones. Por lo tanto, el modelo debe ser actualizado y mantenido regularmente para reflejar los cambios en los datos y asegurar su relevancia y precisión continuas.

Se dirá entonces que el modelo es apto de su uso, si se comprueba que ahorraríamos gastos, tiempo y gestión acompañada de una precisión a la hora de hablar del precio de una vivienda en la plataforma de Zillow.

Conclusiones

- La calidad y disponibilidad de los datos son fundamentales para el desempeño del modelo. Es crucial contar con datos precisos, completos y relevantes para las características de las viviendas.
- La correcta gestión de datos faltantes o inconsistentes es esencial para evitar sesgos o distorsiones en el modelo. Estrategias como la imputación o eliminación de datos pueden ser necesarias.
- La evaluación precisa y exhaustiva del rendimiento del modelo es crucial. Se deben utilizar métricas apropiadas y realizar pruebas en conjuntos de datos de prueba o mediante validación cruzada para evaluar la capacidad de generalización del modelo.
- Es necesario aumentar la complejidad de algunos modelos ya que se presentan problemas de overfitting.
- Dependiendo de la naturaleza del mercado inmobiliario, el modelo puede revelar variabilidad y cambios en los precios de las viviendas a lo largo del tiempo. Esto puede ayudar a comprender las tendencias o patrones estacionales que influyen en los precios de las viviendas.

Bibliografías:

https://www.kaggle.com/competitions/zillow-prize-1/data?select=zillow_data_dictionary.xlsx