Estimación de Precios Competitivos para Propiedades en AirBnB CDMX

Minería y Análisis de Datos

Equipo 5

Raúl Gerardo Reyes Barrón 192129

César Armando Rojas Flores 220019

Rodrigo Alan García Pérez 220211

Emanuel Ortiz Bassoco 130669

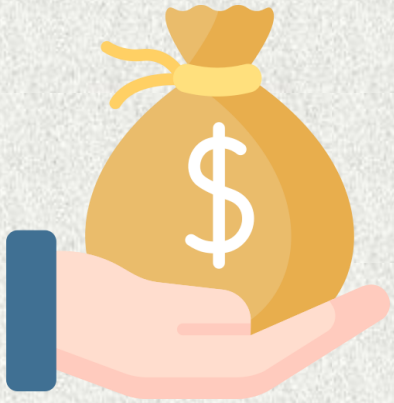
León Felipe Gómez Zarza 111203

Ulises Reyes García 152113

Introducción

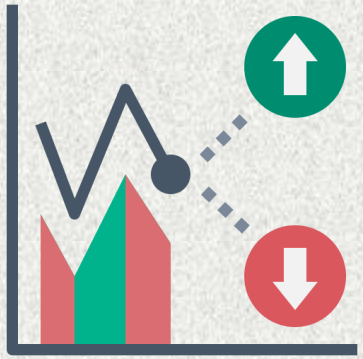
Recordatorio del problema

¿Cuál es nuestro objetivo?



Predecir el precio por noche de un AirBnB listado en CDMX con base en sus características.

¿Cuál es nuestro alcance?



Dar predicción de un precio fijo por noche para una propiedad nueva (sin listar).

Base de datos original

Archivos en formato .csv

26,536 registros (propiedades)

75 posibles variables

- ✓ 34 object
- ✓ 23 integer
- ✓ 18 float

Resultados EDA

Tratamiento de valores nulos y atípicos y transformaciones de variables

Variable	Tipo de Tratamiento	Modificación y explicación:
host_acceptance_rate	**Nulos**	Rellenado con la media
host_response_rate		Rellenado con la media
bedrooms		Rellenado con una constante (0): casos nulos corresponden a estancias sin habitación, como cuartos compartidos o habitaciones dentro de una casa
beds		Rellenado con una constante (1): se parte del supuesto que todos los AirBnB tienen al menos una cama disponible
host_is_superhost		Rellenado con k-vecinos más cercanos
price	**Atípicos**	1) Selección de datos antes del cuantil 999 (para eliminar precios artificialmente elevados). 2) Aplicación de logaritmo natural a la variable objetivo (precio)
host_total_listings_count		Aplicación de transformación Box Cox.

Variables transformadas	Creación y explicación:
Distancia de puntos de interés de la CDMX	Cálculo de distancias hacia puntos de interés usando la fórmula de Haversine (distancia en KM para puntos geográficos) sobre longitud y latitud.
Cuadrática, cúbica y logarítmica	Se hicieron intentos por tener mejores modelos, se intentó hacer transformaciones de distintos tipos a todas las variables numéricas.
Conversión de listas a variables numéricas	Transformación de listas (amenidades) a variable numérica que describe el número de elementos en ella.
One-Hot Encoding	Se aplicó un one-hot encoding a todas las variables de carácter nominal.
Tipo de baño	A partir de la variable original "bathrooms_text" se extrajo información sobre el tipo de baño: privado, compartido o sin especificar.

Resultados EDA

Base resultante tras transformaciones y eliminación de variables

dist_to_Zocalo	bedrooms	room_type_Hotel_room
dist_to_Angel_de_la_Independencia	beds	room_type_Shared_room
dist_to_Parque_México	amenities_n	bathrooms_type_unespecified
dist_to_Bosque_de_Chapultepec	calculated_host_listings_count_entire_homes	bathrooms_type_private
dist_to_Coyoacan_Centro	calculated_host_listings_count_private_rooms	bathrooms_type_shared
dist_to_Aeropuerto_Internacional	calculated_host_listings_count_shared_rooms	alcaldia_Cuauhtémoc
dist_to_Monumento_a_la_Revolucion	accommodates	alcaldia_Cuajimalpa_de_Morelos
dist_to_Museo_Soumaya	description	alcaldia_Miguel_Hidalgo
dist_to_Santa_Fe	neighborhood_overview	alcaldia_Coyoacán
dist_to_Estadio_Azteca	host_location	alcaldia_Alvaro_Obregon
dist_to_UNAM	host_about	alcaldia_Benito_Juárez
dist_to_Centro_Comercial_Perisur	host_is_superhost	alcaldia_Iztacalco
dist_to_Palacio_de_Bellas_Artes	instant_bookable	alcaldia_Tlalpan
dist_to_Auditorio_Nacional	host_verifications_email	alcaldia_La_Magdalena_Contreras
dist_to_Basilica_de_Guadalupe	host_verifications_phone	alcaldia_Iztapalapa
host_since_days	host_verifications_work_email	alcaldia_Venustiano_Carranza
host_since_days	host_verifications_no_verifications	alcaldia_Gustavo_A._Madero
host_since_months	host_response_time_within_an_hour	alcaldia_Xochimilco
bathrooms_numeric	host_response_time_a_few_days_or_more	alcaldia_Azcapotzalco
host_total_listings_count	host_response_time_within_a_day	alcaldia_Tláhuac
host_acceptance_rate	host_response_time_within_a_few_hours	alcaldia_Milpa_Alta
host_response_rate	host_response_time_no_response_time	price
latitude	room_type_Entire_home/apt	
longitude	room_type_Private_room	

✓ 69 variables

31 Numéricas

39 Categóricas

Objetivo

Suposiciones a seguir anteriores

Siguientes pasos de la presentación del 29 de octubre del 2024

- **Análisis de los errores del modelo baseline**, para evaluar si hay subgrupos o segmentos con concentración de error. Iterar el proceso de ingeniería y selección de variables considerando estos resultados.
- Se invertirá **más tiempo en el *tunning* de los hiperparámetros** de los modelos a utilizar para este modelo solo se usó un *grid* de 4 valores para los coeficientes de penalización: *{0.01,0.1,0.5,1.0}*.
- Se optará por otras metodologías que **puedan capturar de mejor forma las relaciones no lineales** de las características, así como, por otras metodologías para la selección de variables.
- **Utilizar el Mean Absolute Percentage Error (MAPE)** para entender el error en términos porcentuales, lo cual puede ser más interpretativo para decisiones de nuestro problema.
- **Utilizar validación cruzada** para evaluar que no tengamos dependencia del split definido actualmente.

Modelo base utilizado: *Ridge*

Feature	Ridge Coefficient
bathrooms_type_shared	-0.402346
alcaldia_Benito_Juárez	-0.204125
bathrooms_type_unespecified	0.102774
host_verifications_work_email	0.116791
host_response_time_no_response_time	0.141735
alcaldia_Miguel_Hidalgo	0.010325
dist_to_Santa_Fe	-0.973997
dist_to_Bosque_de_Chapultepec	0.199611
accommodates	2.241432
calculated_host_listings_count_entire_homes	0.503812
dist_to_Auditorio_Nacional	-1.337561
bathrooms_type_unespecified_squared	0.102774
host_response_time_no_response_time_squared	0.14173

Metric	Training (70%)	Validation(15%)	Test(15%)
RMSE	0.5269	0.5317	0.5239
R^2	0.5001	0.504	0.4984

Hiperparametros evaluados por modelo

- **KNN Regression**
  - Metrica: Manhattan
  - Numero de vecinos: 9
  - Pesos: distancia
- **Red Neuronal**
  - Capa de entrada: 71 neuronas
  - 4 capas ocultas: 40, 20, 10 y 5 neuronas, respectivamente. Activación RELU
  - 1 capa de salida: activación lineal
  - Loss: MAE
- **Random Forest**
  - Número de árboles: 300
  - Número de variables: 17
  - Selección de variable: con reemplazo
  - Tamaño de hoja: 2
  - Métrica de impureza: varianza

Hiperparametros evaluados por modelo

- XGBoost
  - Learning rate: 0.1
  - Número de árboles: 300
  - Selección de variable: con reemplazo
  - Métrica de impureza: varianza
  - Profunidad máxima: 6
- Modelos lineales
  - Alpha Lasso: 0.1
  - Alpha Ridge: 0.1

Tratamiento de los datos

- **Escalamiento (Standard Scaler)**
  - Evitó que variables con mayor escala dominen el modelo.
  - Modelos lineales, KNN y la red neuronal funcionaron mejor con los datos esacaldos. Mejora la convergencia y estabilidad en este tipo de algoritmos.
  - Modelos basados en árboles mejoraron, aunque en menor medida debido a su aqruitectura.
  - Redujo los resultados inconsistentes.
- **PCA**
  - Simplificó los datos al quedarnos solo con las características más relevantes.
  - Redujo tiempos de entrenamiento.
  - Escogimos 10 componentes principales.
- **Feature Selection**
  - Se intentó un proceso de FS, creando diferentes subconjuntos de nuestras variables.
  - El desempeño se vio afectado, por lo que se confirmó que (sin agregar o crear más variables) nuestro conjunto propuesto de datos es el de mejor desempeño.
  - Se optó por entrenar con PCA para capturar la mayor cantidad de varianza explicada.

Tabla Comparativa de Resultados

Datos Sin Escalar

	MSE		MAE		MAPE		R2 Score	
	Train	Test	Train	Test	Train	Test	Train	Test
Ridge	2,244,363	1,071,818,220	576.33	974.45	40.36%	45.46%	50.75%	50.00%
Lasso	1,094,358,750	2,265,570,698	15,334.04	16513.57	1290.60%	1306.66%	-1049.00%	-1047.00%
RLMC	2,243,984	1,066,341,665	576.32	973.4455	40.36%	45.45%	50.73%	49.9%
XGBoost	1,393,578	1,612,675	411.2	452.9	26.30%	28.90%	56.20%	45.60%
Random Forest	612,644	1,511,451	188.46	408.67	9.59%	25.26%	81.00%	49.00%
K-Vecinos	363.47	1,885,655	1.75	535.6	0.11%	43.62%	99.00%	36.00%
Red Neuronal	2,084,678	2,613,546	554.5	569.2	34.9%	35.50%	34.00%	21.00%

Tabla Comparativa de Resultados

Datos Escalados

	MSE		MAE		MAPE		R2 Score	
	Train	Test	Train	Test	Train	Test	Train	Test
Ridge	2,243,990.08	1,066,410,047	576.33	973.46	40.36%	45.5%	50.73%	0.4991
Lasso	2,511,048.79	2,376,455.82	620.156	621.440	44.3%	44.7%	50.7%	44.1%
RLMC	2,243,990.07	1,066,410,047	576.32	973.458	40.36%	45.4%	50.7%	49.9%
XGBoost	702,400	1,402,248	279.4	414.1	17.7%	26.1%	77.9%	52.7%
Random Forest	612,319	1,511,466	188.45	408.72	9.59%	25.26%	80.79%	49.04%
K-Vecinos	363.34	1,526,942.7	1.74	460.21	0.11%	34.05%	0.99	0.48
Red Neuronal	1,685,711	1,740,835	442.3	484.3	27.2%	31.0%	0.47	0.41

Selección de Mejores Modelos

MEJORES MODELOS (con base en métricas anteriormente expuestas):	BASADOS EN ÁRBOLES DE DECISIÓN: **1) XGBoost** **2) Random Forest**
RAZONES PARA EL DESEMPEÑO DE ESTOS MODELOS:	1) En el EDA y durante nuestra primera iteración se observó que el tipo de relaciones entre nuestros datos y nuestra variable explicativa es de tipo NO LINEAL. 2) Se esperaba que estos modelos tuvieran mejor desempeño que los lineales. 3) Los modelos basados en árboles son menos sensibles a diferentes estructuras de datos. 4) Para el otro modelo NO LINEAL (NN) se tuvieron problemas para encontrar la arquitectura adecuada, teniendo comportamientos erráticos con cada combinación de HP.
IMPACTO DE LOS HIPERPARÁMETROS:	Tres hiperparámetros tienen un impacto mayor en el desempeño y tiempo de calibración del modelo: • Número de árboles: Entrenamientos mayores a 500 árboles aumentan mucho el tiempo de ejecución sin mejorar resultados. • Número de variables en split inicial: Valores más grandes a la mitad del número de variables tiran el desempeño del modelo. • Tamaño de la hoja: Se prefieren hojas más chicas a pesar de que esto incrementa el tiempo de entrenamiento.

Ejemplos de éxito y oportunidad

- Se identificaron las mejores predicciones (aquel 5% con el error de predicción más bajo) y las peores predicciones (aquel 5% con el error de predicción más alto) de nuestro modelo.
- Con esa información, se obtuvieron los siguientes datos:

Variable predictora	Tipo de variable	Valor medio de las mejores predicciones	Valor medio de las peores predicciones
room_type_Entire_home/apt	Indicadora	0.528571 (52.8%)	0.814286 (81.4%)
bathrooms_numeric	Numérica	1.484286	2.857143
accommodates	Numérica	2.588571	6.502857
bedrooms	Numérica	1.354286	3.291429
bathrooms_type_shared	Indicadora	0.257143 (25.7%)	0.042857 (4.3%)

- Con esto podemos observar que nuestro modelo es bueno para predecir precios de AirBnB's sencillos (1 a 4 personas, apr), pero tiene bajo desempeño en AirBnB's grupales o de gran tamaño (4 a 6 personas, aprox.)

Conclusiones

Los mejores modelos son capaces de predecir correctamente

- La base de datos contiene cerca de **27,000** inmuebles listados en **Airbnb en la Ciudad de México**, con **75 variables**.
- El objetivo del proyecto fue obtener un modelo que sea capaz de **predecir el precio correcto** para una vivienda con base en el resto de las variables.
- Con el EDA y la ingeniería y refinación de variables, **algunas** variables fueron **eliminadas**, otras fueron **inducidas** de la manera más lógica y consistente con la variable.
- Se hicieron pruebas con distintos modelos una vez que se tuvo la base de datos ya limpia y procesada: **XG Boost, redes neuronales, regresión de mínimos cuadrados, vecinos más cercanos, random forest, Ridge y Lasso**.
- Tomando las métricas de **MSE, MAPE, MAE y R2**, concluimos que los mejores modelos son XG Boost y Random Forest.
- En particular, los modelos **basados en árboles** (XGBoost y Random Forest) fueron los mejores en métricas clave como el error cuadrático medio (MSE) y el error absoluto medio (MAE). Además, se mostraron **robustos frente a datos no lineales y complejas relaciones entre variables**.

Conclusiones

Los mejores modelos son capaces de predecir correctamente

- Para **Random Forest**, el número óptimo de árboles fue 300. El tamaño de la hoja y el número de variables en los splits iniciales influyeron directamente en el rendimiento.
- En el caso de **XGBoost**, una profundidad máxima de 6 y un learning rate de 0.1 ofrecieron el mejor equilibrio entre precisión y tiempo de entrenamiento.
- **Random Forest** presentó un rendimiento **consistente** tanto en el set de entrenamiento como en el de prueba, superando a los modelos lineales en MSE, MAE y MAPE.
- **XGBoost** demostró ser competitivo, logrando un **balance óptimo** entre precisión y capacidad de generalización.
- Los modelos lineales como Ridge y Lasso evidenciaron limitaciones al capturar relaciones no lineales en los datos.
- El modelo predice con precisión reservas de Airbnbs para dos personas, pero su desempeño disminuye notablemente en propiedades grupales para más de cinco personas, indicando que **no captura adecuadamente** factores clave de estas características.

Líneas futuras

- Se puede considerar el crear 2 modelos:
  - Uno para AirBnB's sencillos (para parejas o pocas personas) o de calidad estándar
  - Uno para AirBnB's grupales o de calidad premium
- Explorar a mayor profundidad alternativas diferentes de hiperparámetros para XGBoost y Random Forest
  - Esto ya que los resultados muestran comportamientos interesantes ligados probablemente a la forma en la que cada uno de estos dos modelos se construye
  - Una alternativa sería probar con métodos más robustos y exhaustivos de búsqueda de hiperparámetros (Hyperopt, por ejemplo)
- Incorporar técnicas de ingeniería de variables más complejas, como por ejemplo modelar ubicaciones y trayectos a puntos de interés/transporte público con base en grafos
- Incorporar elementos de estacionalidad y otras características temporales (autoregresiones, por ejemplo) que capturen de mejor manera la fluctuación de los precios a lo largo del tiempo, y el valor que le da el huésped a una propiedad dependiendo la época del año.