Predicción de Precios en alojamientos listados en Airbnb: Nueva York 2019

Sergi Sanz Orellana

Abstract— Since its launch in 2008, Airbnb has transformed the online accommodation landscape, focusing its offer on home stays and unique tourist experiences. This study focuses on the analysis of Airbnb data in New York during 2019. The main purpose is to discover the most efficient predictive model for estimating accommodation prices. In addition, an exploratory data analysis will be carried out to investigate the correlation between price and other relevant features, and a comparison of the results obtained will be made.

Keywords— Price prediction, Airbnb, Linear Regression, Random Forest Regression, Sklearn, New York City, Dataset AB_NYC_2019 of Kaggle

1 INTRODUCCIÓN

Este proyecto ha sido realizado para la asignatura de Aprendizaje Computacional de Ingenieria Informática de la UAB. El objetivo de este, es predecir el precio por noche de los alojamientos listados en Airbnb de la Ciudad de Nueva York en el 2019.

El dataset utilizado es de Kaggle, llamado New York City Airbnb Open Data [1], contiene un excel llamado AB-NYC-2019, con 48.895 alojamientos y 16 columnas de información sobre cada uno:

Features	Nands
id	0
name	16
host_id	0
host_name	21
neighbourhood_group	0
neighbourhood	0
latitude	0
longitude	0
room_type	0
price	0
minimum_nights	0
number_of_reviews	0
last_review	10052
reviews_per_month	10052
calculated_host_listings_count	0
availability_365	0

Tabla. 1 Tabla con la suma de nands que hay en el dataset.

Partiendo de este punto vamos a explicar como se ha desarrollado este trabajo y las relaciones más relevantes encontradas.

2 ANÁLISIS EXPLORATORIO DE DATOS

2.1 Tratamiento de Nands

Para el tratamiento de los nands de esta base de datos, no ha sido muy complicado, debido a que tenia muy pocas variables con nands. Eliminé las columnas de 'name', 'host_name', 'last_review' ya que observando la matriz de correlación no influenciaban prácticamente en el precio.

Para las otras dos variables con nands, me di cuenta de que estaban relacionadas 'last_review' y 'reviews_per_month', lo que quería decir que, si no había obtenido una última reseña, todavía no tenia ninguna reseña por lo tanto las reseñas por mes eran igual a 0. Así que 'reviews_per_month' la rellené de 0 las filas que no tenían dato.

2.2 Codificar variables categóricas y normalización

Una vez con el dataset sin problemas, empecé a codificar las variables categóricas con OneHotEncoder de la librería sklearn [2].

Al codificar 'neighbouhood' y 'neighbouhood_group', nos encontramos con más de 200 columnas nuevas para el atributo 'neighbouhood' y solo 5 columnas para el atributo 'neighbouhood_group'. Por lo que decidí quedarme con la codificación del segundo y eliminar del dataset la variable 'neighbouhood' ya que 'neighbouhood_group' nos daba la misma información de manera más general y fácil para trabajar.

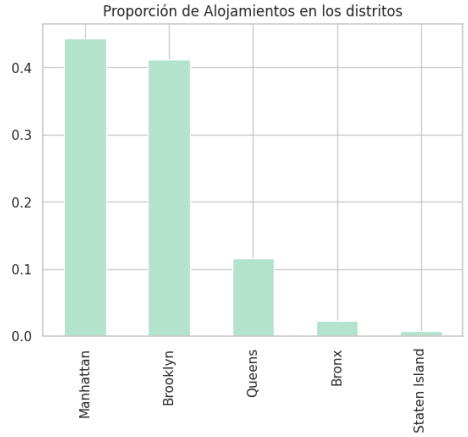

Ilustración 1: Distribución del alojamiento segun el barrio.

Otra variable importante que codificar fue 'room type' que se podía dividir en 3 tipos de habitación:

Entire home/apt; Private room; Shared room;

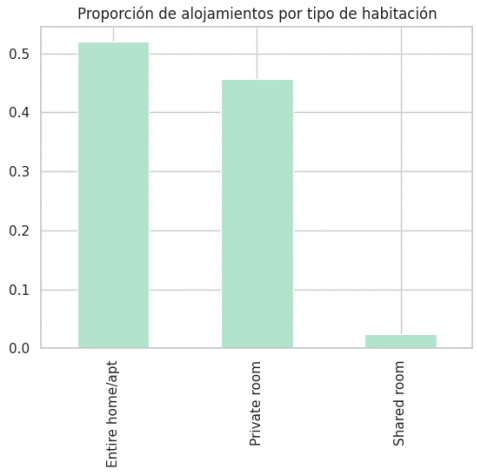

Ilustración 2: Distribución tipo de alojamiento.

Para normalizar el dataset apliqué un StandardScaler, también de la librería de sklearn [2], en aquellas variables que lo requerían y así ya quedó normalizado.

2.3 Matriz de Correlación

Una vez con el dataset limpiado y preparado, ahora si se podía aplicar una matriz de correlación para observar que categorías estaban más influenciadas a la hora de determinar el precio. Y este fue el resultado:

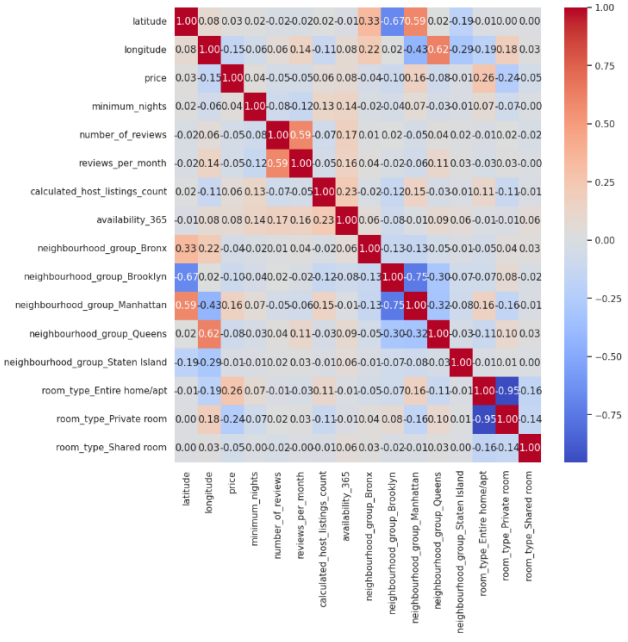

Ilustración 3: Matriz de correlación.

Observando esta imagen vemos que el precio viene influido en mayor medida por el tipo de alojamiento, si es un apartamento entero o una habitación privada. Seguido de en la ubicación de que barrio de encuentra y por la longitude del mismo (coordenadas en la Tierra).

3 EXPERIMENTOS Y MODELOS DE REGRESSIÓN

Como veremos más adelante, tenemos muy malos resultados utilizando el 100% del dataset debido a que tenemos los precios de alojamiento muy mal distribuidos y tenemos muy pocos datos de entrenamiento para precios del alojamiento por encima de los 200 dólares. Por eso decidí separar el dataset i trabajar con cerca del 80% de los datos filtrando los alojamientos por debajo de los 200$.

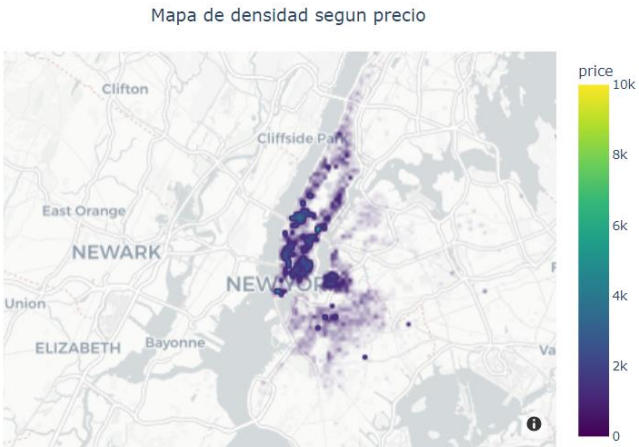

Ilustración 4: Mapa interactivo creado en el notebook.

3.1 Todos los precios

En esta sección trabajamos con todas las filas de nuestro dataframe lo que nos producirá muy malos resultados. Algo mejor con el Random Forest, pero no significante.

3.1.1 Regresión Lineal

Separamos en train y test dejando un 30% para test. Aplicamos LinearRegression()[2] y observamos los resultados tanto del mae, mse, rmse y r2_square:

Métrica	Valor
MAE	75.26
MSE	55073.80
RMSE	234.67
R2 SQUARE	0.09

3.1.2 Regresión Lineal con Random Forest

Separamos en train y test dejando un 30% para test. Aplicamos RandomForestRegressor()[2] y observamos los resultados tanto del mae, mse, rmse y r2_square:

Métrica	Valor
MAE	65.64
MSE	35541.35
RMSE	188.52
R2 SQUARE	0.12

3.2 Precios < 200$

Ahora trabajando con el 80% de los datos filtrados por el precio inferior a 200$ mejoran mucho los resultados.

3.2.1 Regresión Lineal

Al ver que tiene un mejor resultado miro como de bien esta el r2 score tanto en el train como en el test, no encontramos overfitting:

Métrica	Valor
MAE	24.60
MSE	977.14
RMSE	31.25
R2 SQUARE train	0.49
R2 SQUARE test	0.51

3.2.2 Regresión Lineal con Random Forest

Pruebo a hacer una regressión con RandomForestRegressor() [2] para ver que tal:

Métrica	Valor
MAE	21.97
MSE	839.36
RMSE	28.97
R2 SQUARE train	0.94
R2 SQUARE test	0.58

Encontramos leves mejoras en todas las métricas, pero nos damos cuenta de que nuestro modelo esta padeciendo de overfitting ya que hay diferencia entre la r2 del train (muy buena) y del test (no tan buena).

3.2.3 Cross Validation para mitigar Overfitting

Para mitigar el overfitting, empiezo haciendo una feature importance del modelo actual que dispongo:

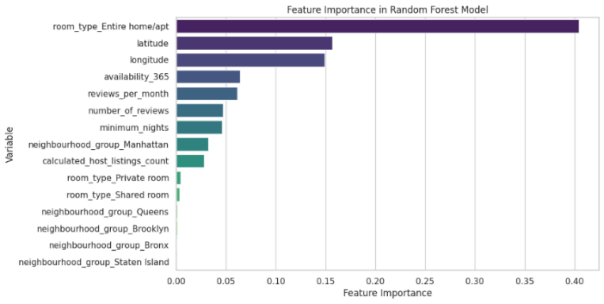

Ilustración 5: Gráfica feature importance.

Seguimos haciendo Cross Validation con 3-fold y 100 iteraciones por cada 1 para encontrar los mejores parámetros. Una vez tenemos los mejores parámetros entrenamos al modelo y vemos los resultados obtenidos:

Métrica	Valor
MAE	24.60
MSE	977.14
RMSE	27.36
R2 SQUARE train	0.83
R2 SQUARE test	0.62

Conseguimos mitigar algo la diferencia y mejorar el modelo.

4 Predicciones y resultados

4.1 Predicciones con todos los precios

Ejemplos de predicciones de precio con el 100% del dataset:

Real	Pred
89	157.50
30	48.69
470	302.70
800	248.36
48	40.97

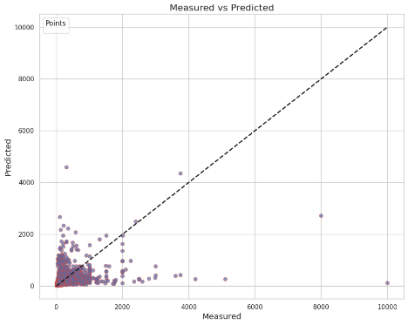

Ilustración 6: Resultados comparados con los datos reales.

Con esta gráfica me di cuenta que la mayoría de los precios de los alojamientos eran inferior a los 1000$ y que la mayoría estaban centrados por debajo de los 200$.

4.2 Predicciones precios < 200$

Ejemplos de predicciones de precio con el 80% del dataset:

Real	Pred
100	108.07
80	81.26
63	70.81
81	76.26
55	55.78

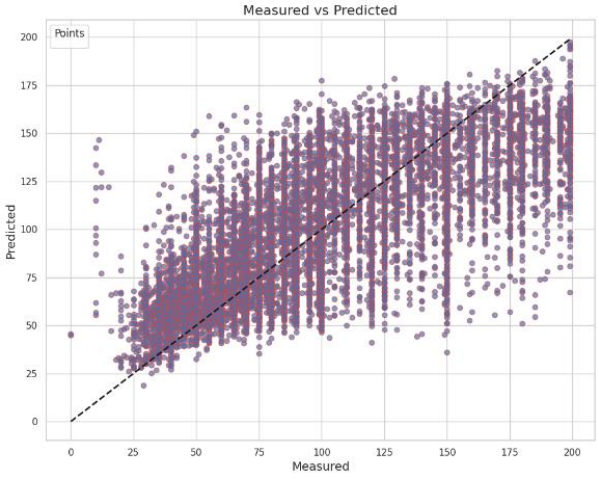

Ilustración 7: Resultados comparados con los datos reales.

Vemos como se agrupa el 80% de los alojamientos en un intervalo de 200$ y que las predicciones de nuestro modelo son muy cercanas a la realidad.

5 CONCLUSIONES

5.1 Comparación RMSE de los modelos

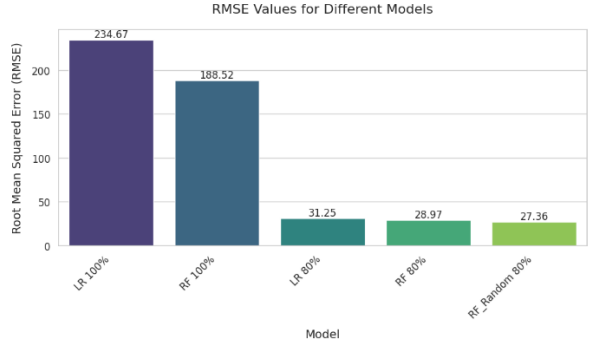

Ilustración 8: Comparación del RMSE con todos los modelos utilizados.

5.2 Comparación R2 SCORE de los modelos

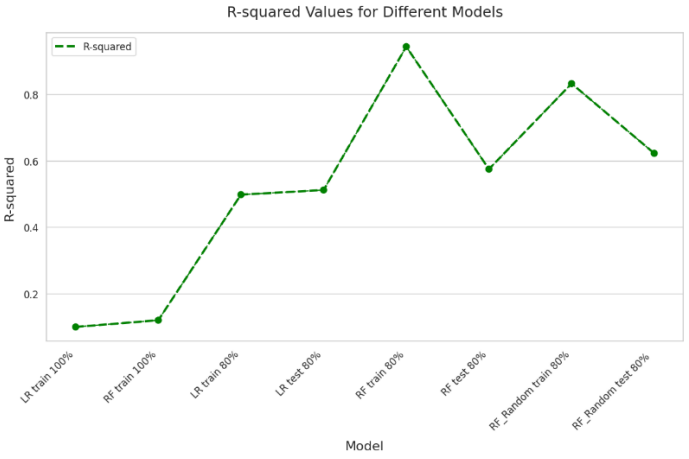

Ilustración 9: Comparación del r2 score de cada modelo.

-- Los alojamientos de este dataset tienen grandes diferencias en los precios. Separar el conjunto de datos por el precio es útil para el análisis.

-- Los modelos de predicción de precios no han funcionado bien con todo el conjunto de precios.

-- En cambio, aplicando una restricción en el precio inferior a 200$ (80% del conjunto de datos) podemos considerar que nuestro modelo ha mejorado bastante, dando una buena predicción.

-- La mejor puntuación con el 100% de los datos ha sido de 0.12 del r^2 score.

-- La mejor puntuación ha sido de un r^2 score de 0.58 para el random forest con el 80% del conjunto de datos, pero nos hemos encontrado en un caso de overfitting ya que en el train tenemos un r^2 score de 0,94.

--Finalmente, haciendo Cross Validation, hemos podido mitigar un poco el overfitting y pasar de un r^2 score train a 0.83 y un r^2 score test de 0.62.

-- Gracias a este análisis, hemos obtenido una comprensión más profunda de los factores clave que influyen en el precio de un alojamiento en NYC en la plataforma de Airbnb.

-- Estoy contento con los resultados, pese a las complicaciones y la inexperiencia, estoy satisfecho con la finalización y aprendizaje de este proyecto.

BIBLIOGRAFIA

[1] https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data Kaggle dataset of DGOMONOV.
[2] https://scikit-learn.org/stable/ Documentación de sklearn.
[3] https://github.com/sergisanzorellana/Price-Prediction-NYC-Dataset-of-Kaggle Enlace al código jupyter notebook.