Trabajo final grupal: Airbnb Price

Introducción y objetivos

Durante el desarrollo del siguiente trabajo se llevarán a la práctica diferentes técnicas y conceptos aprendidos durante la cursada. En este caso, partiendo de un dataset con un listado de 19.309 publicaciones con 29 variables/dimensiones que muestran características de las propiedades, se busca entrenar un modelo que logre predecir el precio por noche de los hospedajes para algunas ciudades dentro de USA.

Descripción del dataset

El dataset a disposición contiene información de 19.309 publicaciones de la plataforma Airbnb, en donde se observan 29 dimensiones diferentes que caracterizan los diferentes tipos de propiedades. Entre ellas podemos encontrar tipo de propiedad, amenities, políticas de cancelación, ubicación geográfica, entre otras y el precio por noche, que es la variable que buscaremos poder predecir.

En primer lugar se inició con la limpieza del Data Frame, en donde evaluamos la presencia de valores faltantes y el tipo de dato correspondiente a cada variable (**fig.1**). En esta primera acción pudimos notar la presencia de una gran cantidad de NaNs, los cuales, en caso de su eliminación, generaría la pérdida de más de 4.000 registros. De esta forma optamos por diferentes técnicas, para completar estos valores con datos reales calculados. En el caso de la variable con mayor cantidad de nulos, **'host_response_rate'**, se la analizó en relación con las variables 'property_type', 'room_type' y 'city', calculando el promedio de host_response_rate según las 3 variables por separado y logrando crear un promedio segmentado (**fig.2**) con el cual reemplazamos los valores faltantes del dataset original. Luego repetimos este mismo caso con la segunda variable con mayor cantidad de nulos, **'review_scores_rating'**, también en relación a las variables 'property_type', 'room_type' y 'city'.

En el caso de las siguientes tres variables en la tabla de nulos, 'last_review', 'first_review' y 'thumbnail_url' se decidió no afectarlas, ya que los valores que las mismas contienen son irrelevantes o redundantes para el modelo que buscará predecir el precio de un hospedaje. Y para las variables **'bathrooms'**, **'beds'** y **'bedrooms'** buscamos los valores estadísticos según 'property_type' y 'room_type' para también poder reemplazar los NaNs por valores representativos.

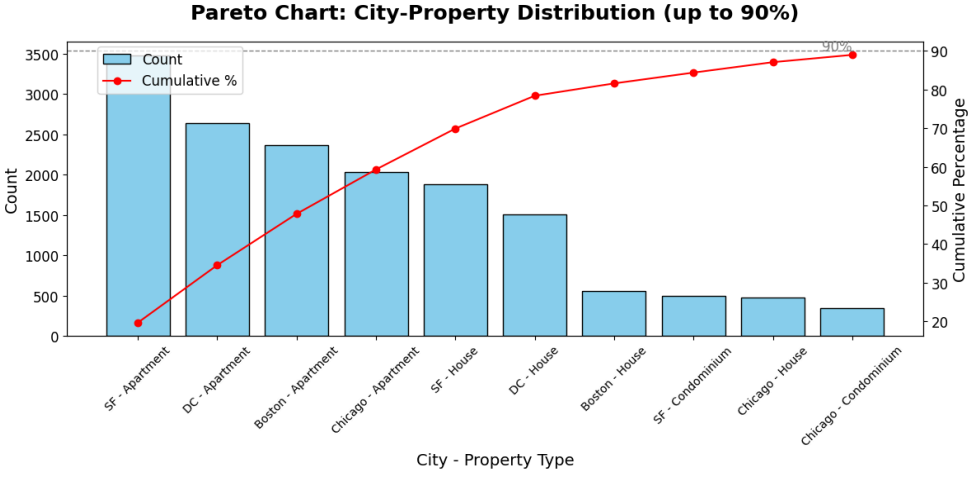
Para la variable **'neighbourhood'**, se buscó algún método que nos permitiese completar los NaNs con valores, sin necesidad de incurrir a la eliminación de registros, como pudiera ser un join & merge de pandas (símil buscarV en excel) buscando los zip codes y reemplazando el barrio asociado a este. Sin embargo, en este paso descubrimos que los zip codes en Estados Unidos, pueden identificar a varias ciudades distintas y en concordancia, varios barrios distintos (**fig.3**). Por esta razón, la solución final fue eliminar la totalidad de los nulos existentes en la variable.

Finalmente post limpieza de datos, el dataset inicial de 19.309 registros pasó a ser de 17.713, es decir, durante esta etapa se eliminaron un 8,27% de los datos originales para continuar las siguientes etapas con un set adecuado.

Análisis exploratorio de datos

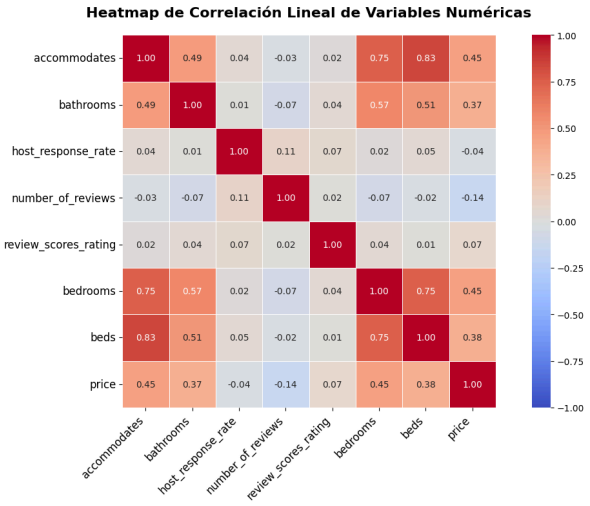
Para el análisis exploratorio de datos, con el dataset limpio, buscamos entender el comportamiento de las variables más importantes del dataset, su relación entre sí y entender cuales son las más relevantes para los análisis predictivos posteriores. Si bien en este paso se realizó un análisis extensivo, desarrollaremos aquellos resultados que nos parecieron más importantes para el entendimiento de dataset.

En primer lugar representamos, mediante un diagrama pareto, como se distribuye la combinacion Ciudad - Tipo de Propiedad, en donde podemos ver, por ejemplo, que el 20% de las muestras se concentra en Apartments en SF, pudiéndose observar también que el 59% de los valores, es decir más de la mitad del dataset corresponde a Apartments.



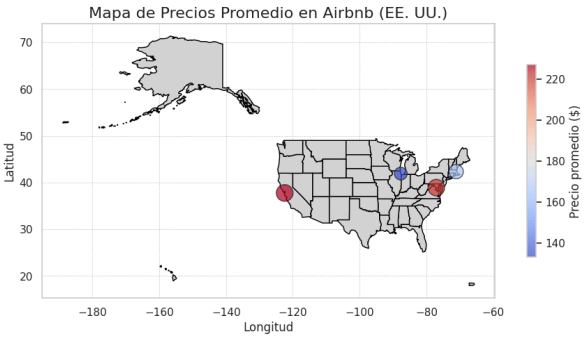
En segundo lugar se buscó entender para cada una de las 4 ciudades, que cantidad de barrios asociados presentan, qué cantidad de reservas tienen, siendo que cada muestra corresponde a una reserva, y que porcentaje del dataset ocupan en consecuencia, lo cual se puede ver en la tabla adjunta.

	city	neighbourhood_count	reservation_count	neighbourhood_percentage	reservation_percentage
0	Boston	34	3440	12.2%	19.4%
1	Chicago	80	3056	28.8%	17.3%
2	DC	116	4877	41.7%	27.5%
3	SF	55	6340	19.8%	35.8%

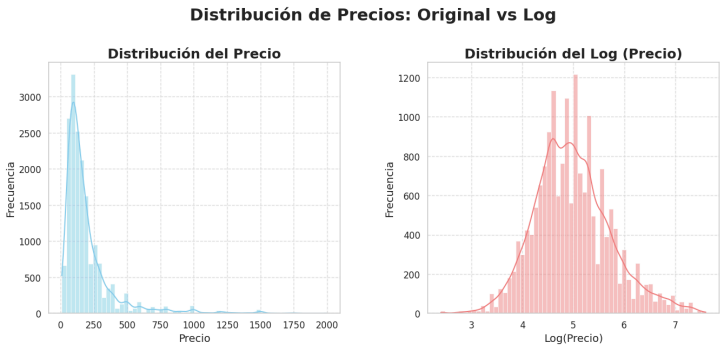


Luego pasamos a analizar la correlación lineal existente entre las diferentes variables numéricas del dataset, para entender si existe alguna correlación entre variables, interesándonos por aquellas que más se correlacionan con la variable precio, variable a predecir posteriormente. En este caso, representado con un heatmap, podemos ver como existe alta correlación entre las variables 'accommodates' - 'beds' y 'accommodates' - 'bedrooms', pero ninguna variable correlaciona linealmente con el precio.

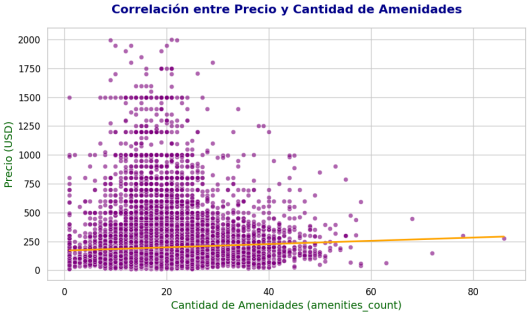
Por otro lado, analizamos la distribución de precios en las diferentes regiones analizadas del dataset, donde encontramos que la ciudad con los precios más altos es SF.



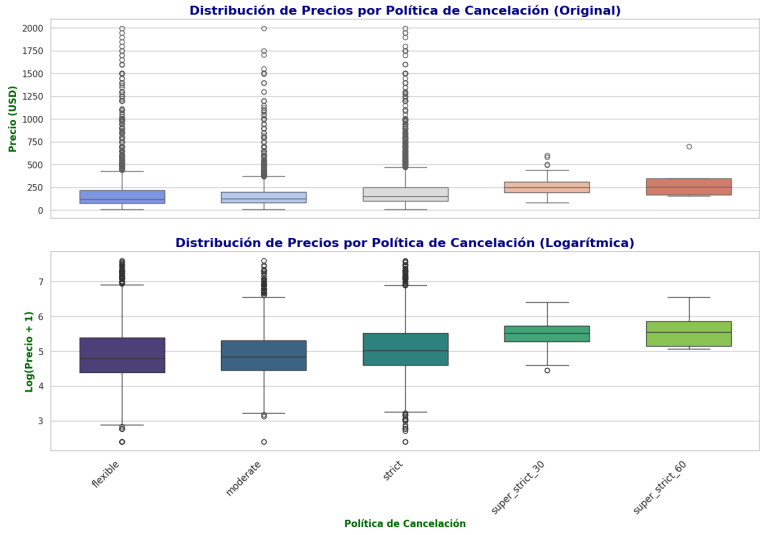
También graficamos la distribución de la variable precio para entender su comportamiento, en donde junto con un boxplot, se pueden ver sus principales indicadores y que tenemos valores muy variados, con un mínimo de 10 USD y un máximo de 1.999 USD.



Durante este análisis también creíamos que el precio podía tener cierta correlación lineal con la cantidad de amenities, lo cual graficamos y descubrimos que en la realidad no tienen ningún tipo de correlación.



Finalmente, mediante el siguiente boxplot logramos entender que el precio por noche varía en gran parte según la política de cancelación, encontrando precios más altos/caros para políticas de cancelación más estrictas.



Materiales y métodos

Para el desarrollo del modelo buscamos aplicar métodos como get dummies para transformar variables categóricas a numéricas y así usarlas más adelante en el análisis de regresión. En este caso utilizamos get_dummies de pandas para las variables categóricas city, room_type, property_tpe, cancellation_policy, bed_type. A su vez transformamos los valores “t” y “f” de las variables 'host_has_profile_pic'y 'host_identity_verified' a True y False con .replace y para las variables ‘host_response_rate’ y ‘review_scores_rating’ convertimos la primera de porcentaje a valor entre 0 y 1 y la segunda de su rango original (0-100) a un valor entre 0 y 1.

Una vez preparado el dataset definimos la variable dependiente (y) a predecir, en este caso ‘log_price’ y el conjunto de variables dependientes (x), conformado por todas las variables restantes exceptuando ‘neigbourhood’, 'id', 'zipcode', 'latitude', 'longitude', 'price' y 'log_price', esta última por ser la variable dependiente. Luego dividimos el set de datos entre entrenamiento (70%) - prueba (30%) y escalamos el conjunto x tanto de entrenamiento como el de prueba con StandardScaler().

Finalmente aplicamos dos modelos iniciales para la predicción del precio por noche, iniciando con un modelo de regresión lineal [LinearRegression()] y luego, para el segundo modelo, utilizamos Support Vector Regressor (SVR) en donde mediante grid search seleccionamos los mejores hiperparametros, dentro de las siguientes opciones:

```
'kernel': ['linear', 'rbf']        # Dos posibles valores para el parámetro "kernel"
 'C': [0.1, 1, 10]                 # Tres posibles valores para "C"
'gamma': [0.01, 0.1, 1]            # Tres posibles valores para "gamma"
```

Posteriormente, para un nuevo experimento, se aplicó PCA al dataset con las dummies creadas, buscando proyectar las features originales en un espacio de menor dimensión, y volvimos a entrenar ambos modelos para comparar los resultados obtenidos.

Experimentos y resultados

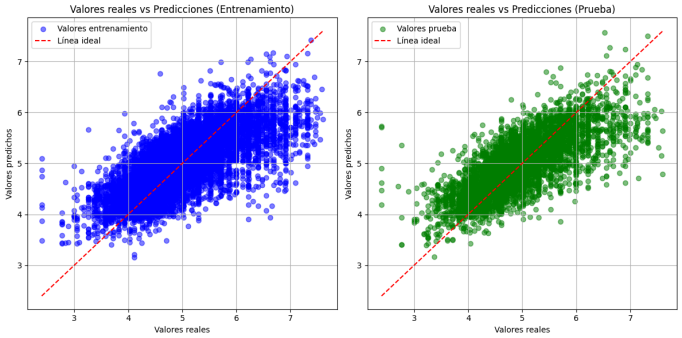
Como resultados podemos ver que en el primer experimento, con un modelo de regresión lineal, los resultados con el log_price son los siguientes:

- Raiz cuadrada del Error cuadrático medio (RMSE) en el conjunto de entrenamiento: 0.5006
- Raiz cuadrada del Error cuadrático medio (RMSE) en el conjunto de prueba: 0.5064

y sin logaritmos en la variable price:

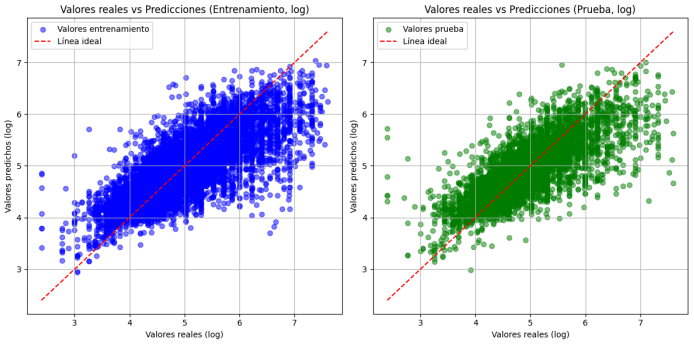
- RMSE en entrenamiento (sin log): 163.65
- RMSE en prueba (sin log): 165.76
- R^2 en el conjunto de entrenamiento: 0.5383
- R^2 en el conjunto de prueba: 0.5494

con lo cual, podemos ver que la performance en regresión del modelo no es tan buena teniendo en cuenta que el R^2 ideal es 1.



Por otro lado, se puede observar que con Support Vector Regressor, con los mejores hipeparámetros {'C': 1, 'gamma': 0.01, 'kernel': 'rbf'} los resultados fueron levemente superiores:

- Raíz del error cuadrático medio (RMSE) en el test set con SVR (log): 0.4927
- Raíz del error cuadrático medio (RMSE) en el test set con SVR (sin log): 162.2723
- Coeficiente de determinación (R^2) en el test set con SVR: 0.5735



Luego vemos que aplicando reducción de la dimensionalidad, en donde hacemos una combinación lineal de las features originales proyectándose en un espacio de menor dimensión, vemos que con 10 componentes solo podemos explicar el 43% de la variabilidad (**fig.4**), por lo tanto el PCA no resulta muy eficiente, esto se puede ver ya que probando nuevamente entrenar con LinearRegression, los resultados son peores:

con logaritmo de la variable price:

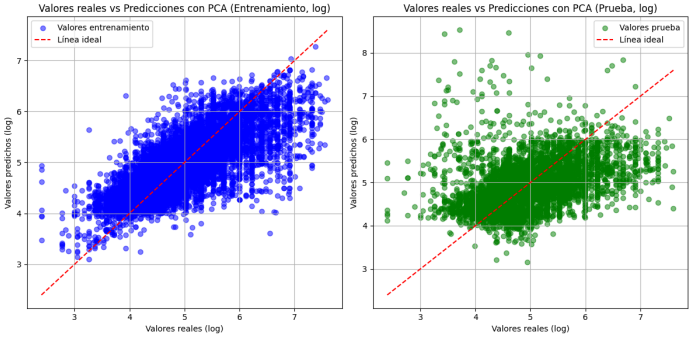
- Raiz cuadrada del Error cuadrático medio (RMSE) en el conjunto de entrenamiento: 0.5269
- Raiz cuadrada del Error cuadrático medio (RMSE) en el conjunto de prueba: 0.6614

y sin logaritmos en la variable price:

- RMSE en entrenamiento (sin log): 171.31
- RMSE en prueba (sin log): 10458.84
- R^2 en el conjunto de entrenamiento: 0.4885
- R^2 en el conjunto de prueba: 0.2314

Y, entrenando nuevamente con Support Vector Regressor, vemos el mismo caso donde los resultados empeoran respecto a la prueba anterior sin reducción de la dimensionalidad:

- Raíz del error cuadrático medio (RMSE) en el test set con SVR (log): 0.7492
- Raíz del error cuadrático medio (RMSE) en el test set con SVR (sin log): 259.8831
- Coeficiente de determinación (R^2) en el test set con SVR: 0.0138



Discusión y conclusiones

Luego del desarrollo del siguiente trabajo, podemos concluir que la reducción de la dimensionalidad no fue funcional en el experimento ya que teníamos una gran cantidad de variables categóricas, pero ninguna lograba explicar la variabilidad del modelo con claridad.

A su vez, el modelo previo al PCA tampoco vio reflejado una buena performance por lo tanto la capacidad de predicción del mismo no fue la mejor. Esto se puede explicar dado que el dataset estudiado, presenta una gran cantidad de variables pero ninguna está altamente relacionada con el precio por noche del alojamiento, por lo tanto, la predicción del mismo presenta dificultades aun asi el mejor modelo fue el de Support Vector Regressor(SVR) sin PCA donde se obtuvo un R^2 de 58%

Referencias

https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html
https://scikit-learn.org/stable/modules/decomposition.html
https://scikit-learn.org/stable/modules/svm.html#regression
https://python-data-science.readthedocs.io/en/latest/index.html

Gráficos Adjuntos

Tabla resumen de NaNs:

	Columna	Tipo de datos	Valores no nulos	Porcentaje no nulos (%)	Valores nulos	Porcentaje nulos (%)
0	host_response_rate	object	15013	77.75%	4296	22.25%
1	review_scores_rating	float64	15175	78.59%	4134	21.41%
2	last_review	object	15355	79.52%	3954	20.48%
3	first_review	object	15355	79.52%	3954	20.48%
4	thumbnail_url	object	16907	87.56%	2402	12.44%
5	neighbourhood	object	17851	92.45%	1458	7.55%
6	zipcode	object	19084	98.83%	225	1.17%
7	bathrooms	float64	19274	99.82%	35	0.18%
8	beds	float64	19285	99.88%	24	0.12%
9	bedrooms	float64	19292	99.91%	17	0.09%
10	host_since	object	19306	99.98%	3	0.02%
11	host_has_profile_pic	object	19306	99.98%	3	0.02%
12	host_identity_verified	object	19306	99.98%	3	0.02%
13	longitude	float64	19309	100.0%	0	0.0%
14	number_of_reviews	int64	19309	100.0%	0	0.0%

fig.1

	city	property_type	room_type	Promedio host_response_rate (%)
0	Boston	Apartment	Entire home/apt	96.66
1	Boston	Apartment	Private room	96.89
2	Boston	Apartment	Shared room	91.95
3	Boston	Bed & Breakfast	Entire home/apt	100.00
4	Boston	Bed & Breakfast	Private room	98.67
...	...	...	...	...
152	SF	Timeshare	Private room	100.00
153	SF	Townhouse	Entire home/apt	96.85
154	SF	Townhouse	Private room	100.00
155	SF	Treehouse	Entire home/apt	100.00
156	SF	Villa	Private room	100.00

151 rows × 4 columns

fig.2

index	zipcode	0	1	2
12	02122	Dorchester - Boston	Downtown Crossing - Boston	
13	02124	Dorchester - Boston	Mattapan - Boston	
14	02125	Dorchester - Boston	Roxbury - Boston	
15	02126	Dorchester - Boston	Mattapan - Boston	Hyde Park - Chicago

fig.3

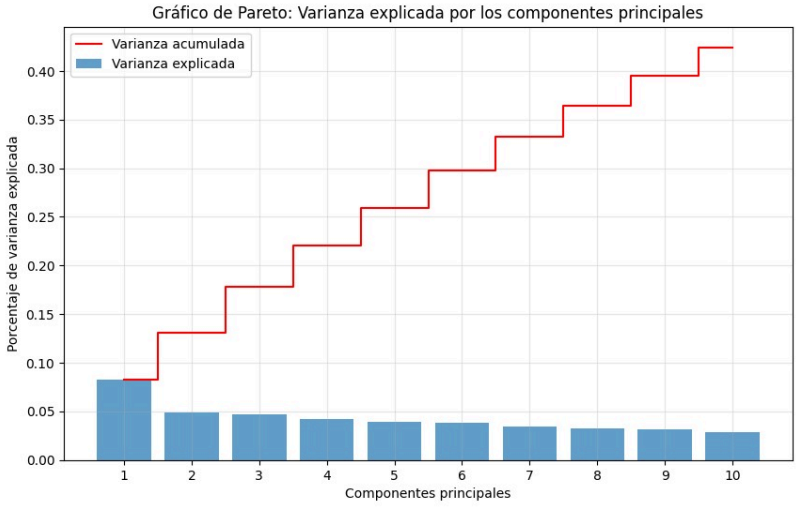


fig.4