Universidad de
SanAndrés

Segundo Trabajo Práctico

Ciencias de Datos

2 de Octubre 2024

Profesor Ignacio Spiousas

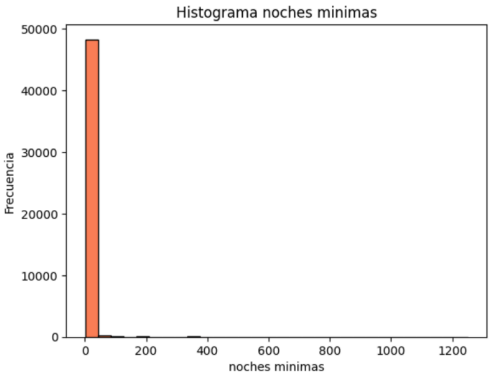
Zahira Chaia

Martina Lucas

Parte 1: Descripción de la Base de Datos y Proceso de Carga

Como primer paso, procedimos a cargar la base de datos proporcionada, titulada “Base Airbnb NY”, la cual contiene 16 variables relacionadas con los oferentes de Airbnb en la ciudad de Nueva York. Al inicio, el conjunto de datos contenía un total de 48.905 observaciones. Primero, identificamos las filas duplicadas utilizando la función *df.duplicated()*, que generó una serie booleana donde se marcan como True aquellas filas que se han repetido. Posteriormente, contamos la cantidad de duplicados (10) y los filtramos con *drop_duplicates()*.

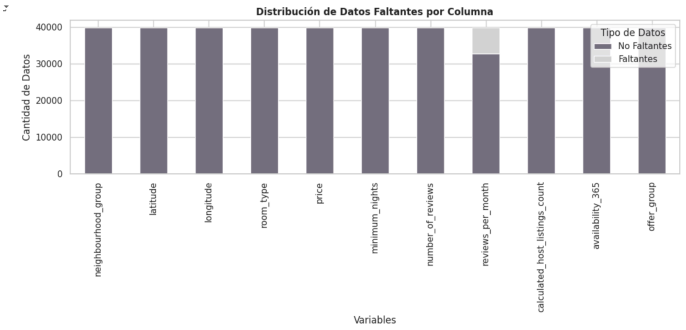
Luego, eliminamos las columnas no relevantes para el análisis, como "id", "name", "host_id", "host_name", "neighbourhood", "last_review". Las variables con las que trabajaremos son las siguientes: “neighbourhood_group” (ubicación del alojamiento), “latitude” (latitude), “longitude” (longitud), “room_type” (tipo de habitación), “price” (precio por noche), “minimum_nights” (número mínimo de noches requeridas), “number_of_reviews” (número total de reseñas del anuncio), “reviews_per_month” (número de reseñas por mes), “calculated_host_listings_count”: (cantidad de anuncios de un determinado anfitrión) y, “availability_365” (número de días al año en los que la unidad está disponible).



A continuación, realizamos un análisis exploratorio de los datos (EDA) con el objetivo de identificar errores obvios, comprender mejor los patrones y detectar valores atípicos (IBM, n.d.). Para cada variable calculamos la media, desviación estandar y los valores mínimos y máximos con la funciona *describe()*. Posteriormente, para observar la distribución de los datos, generamos un histograma para cada columna. A partir de este análisis, observamos que en muchas columnas la gran mayoría de los datos se concentraban dentro de un rango acotado y unos pocos valores eran inusualmente altos o bajos (Puede observarse la figura 1 a modo de ejemplo). Para cada variable donde esto ocurría, decidimos eliminar una cantidad igual o menor al 0.1% de los datos. Antes de llevar esto a cabo, es importante considerar que esta reducción en la varianza está acompañada de un aumento en el sesgo (Singh, 2018). En este caso en particular, tras analizar los datos, concluimos que era razonable llevar a cabo esta acción para mejorar la calidad de nuestro análisis.

El siguiente paso fue la identificación y eliminación de *outliers*. Para hacerlo creamos un Data Frame con las columnas numéricas y sin la columna "minimum_nights" ya que una gran cantidad de unidades no cuentan con un mínimo de noches requeridas y no resultaba tan relevante para el análisis. Posteriormente, aplicamos una transformación logarítmica a la columna "price" para normalizar su distribución. Aunque evaluamos aplicando transformaciones logarítmicas a otras variables, estas no parecían mejorar significativamente el modelo. Calculamos el rango intercuartílico (IQR) para establecer los límites que definen los outliers. Utilizamos este método porque el IQR es eficaz para identificar valores atípicos al considerar la dispersión de los datos (R-bloggers, 2020). Definimos que cualquier valor que se encuentre a más de 3 veces el IQR por debajo del primer cuartil o por encima del tercer cuartil se clasifica como un outlier. Identificamos un total de 5,943 outliers. Decidimos eliminarlos, ya que estos podrían distorsionar los resultados y sesgar las conclusiones del análisis.

Para abordar los datos faltantes en nuestro conjunto, comenzamos por identificarlos y visualizarlos utilizando un gráfico de barras apiladas (Figura 2). Observamos que todos los datos faltantes provenían de la columna "reviews_per_month", que presentaba un total de 9,570. Al analizar la naturaleza de los datos faltantes y considerando la gran cantidad de datos ausentes en una misma columna, concluimos que estos no pueden considerarse como Missing Completely At Random (MCAR). Considerando esto, decidimos optar por la imputación múltiple de datos, utilizando *IterativeImputer* para estimar los valores ausentes basándonos en otras variables. Una vez imputados los datos, los reemplazamos en la columna original. La imputación es especialmente beneficiosa cuando se presentan datos faltantes en variables independientes, particularmente si esos faltantes son de naturaleza no aleatoria (Nguyen, 2021). Además, es recomendable considerando que nuestro objetivo final es la predicción, donde la imputación puede reducir el error estándar al incluir información de otras variables (Rubin, 1996).



Continuamos transformando las variables categóricas "neighbourhood_group" y "room_type" en variables numéricas. En un primer momento, probamos utilizando *frequency encoding* para renombrar las variables. Esta técnica consiste en reemplazar cada categoría, como los barrios o tipos de cuarto, por su frecuencia relativa en la base de datos (Neural Ninja, 2023). El objetivo de esta decisión era proporcionar al modelo información sobre la cantidad de Airbnbs en cada barrio o de cada tipo de habitación, algo relevante considerando que la oferta está vinculada al precio. Luego de implementar el modelo, evaluamos ésta decisión y decidimos utilizar otra técnica para renombrar las variables categóricas.

Por segunda vez renombramos las variables categóricas, ahora dándoles un valor ordinal basado en el precio promedio. Luego de calcular el precio promedio de cada categoría las ordenamos y las reemplazamos manualmente. En el caso de "neighbourhood_group" otorgamos a las variables valores ascendentes de 0 a 4, donde Bronx recibe el valor más bajo debido a su precio promedio y Manhattan el más alto. Para la variable "room_type" realizamos el mismo procedimiento, renombrando con 0 el tipo de cuarto con un menor precio promedio (“Shared room”) y 2 el de precio más alto ("Entire home/apt"). El objetivo de ésta técnica es evitar asignarle a las variables un número de forma arbitraria y otorgarle al modelo información sobre los precios de cada categoría, algo relevante considerando que es la variable que queremos predecir.

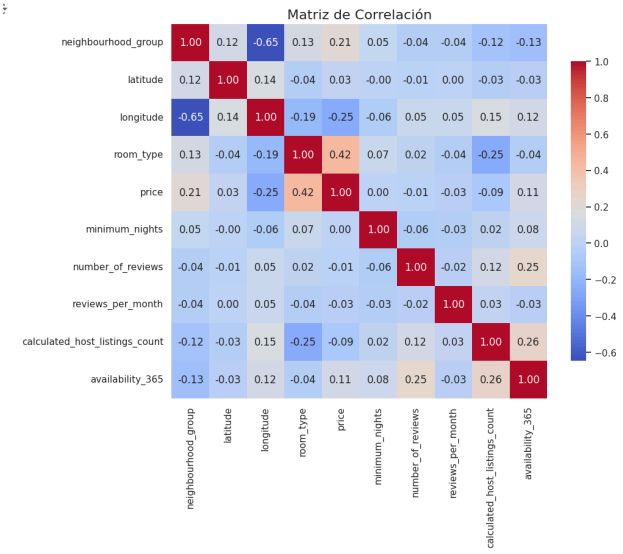
Comparamos el rendimiento del modelo utilizando esta técnica con el *frecuency encoding* y observamos resultados bastante similares. Por esta razón, decidimos utilizar este segundo método, ya que utilizar números enteros simplifica el análisis y facilita el manejo y visualización de los datos.

Por último, creamos una nueva columna llamada "offer_group" para reflejar la cantidad de oferentes por cada "neighbourhood_group". Primero, contamos cuántas veces aparece cada categoría en la variable "neighbourhood_group". Luego, renombramos las columnas de este conteo para mayor claridad. Finalmente, utilizamos la función merge para combinar esta información con el Data Frame

original, de modo que cada fila en incluya la cantidad de oferentes correspondiente a su grupo de vecindario. Esto nos proporciona una mejor comprensión del número de oferentes en cada área.

Parte 2: Gráficos y visualizaciones

Ejercicio 2



Una vez completada la limpieza de los datos, generamos una matriz de correlación (Figura 3), que es una tabla que muestra los coeficientes de correlación de Pearson entre todas las posibles parejas de variables. Este análisis nos permite entender las relaciones entre las distintas columnas de datos con las que estamos trabajando. El coeficiente de correlación, que varía entre -1 y 1, indica la fuerza y la dirección de la relación lineal entre dos variables.

Recordemos que cuando renombramos las variables categóricas les dimos un valor ordinal ascendente según el precio promedio de cada neighbourhood group y room type. El número asignado a cada barrio y tipo de cuarto aporta información sobre el precio de la unidad, por lo que incluirlas en la matriz de correlación puede aportar información relevante sobre la relación de dichas variables categóricas transformadas con otras variables de interés.

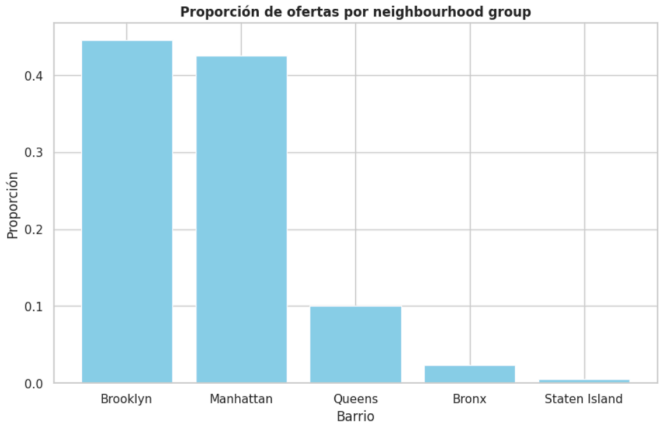
A partir de la matriz de correlación no parecerían observarse relaciones muy fuertes entre la mayoría de las variables. La correlación positiva más fuerte parecería ser entre *room_type* y *price* (0.42), lo que indica que los tipos de habitaciones a los que se les asignó un número mayor tienden levemente a tener precios más altos. Esto tiene sentido, teniendo en cuenta que el criterio con el cual se asignó un orden a cada uno de los tipos de habitaciones. No obstante, considerando que utilizamos el mismo criterio para renombrar a la variable categórica *neighbourhood_group,* nos llamó la atención que no se observa una relación positiva más bien débil entre dicha variable y *price* (0.21). Por otro lado, observamos una correlación negativa moderada (0.65) entre las variables *longitud* y *neighbourhood_group*. Esto indica que, a medida que cambia el grupo de barrio, hay una tendencia de variación en la longitud geográfica. Además, existe una correlación negativa débil (-0.25) entre *longitude* y *price*, lo que sugiere que a medida que cambia la longitud, el precio se modifica ligeramente. Además, existe una correlación positiva moderada (0.26) entre la variable *calculated_host_listings_count* y *availability_365*. A partir de esto podemos concluir que aquellas personas más activas en la plataforma (con más anuncios publicados) tienden a ofrecer propiedades con mayor disponibilidad.

Ejercicio 3

A continuación, calculamos la proporción de oferentes por grupo barrial, dividiendo la cantidad de veces que aparece cada barrio por el total de filas de la variable *neighbourhood*_group. Estas se pueden observar en la siguiente tabla.

Brooklyn	0.445353
Manhattan	0.425118
Queens	0.100565
Bronx	0.023318
Staten Island	0.005646

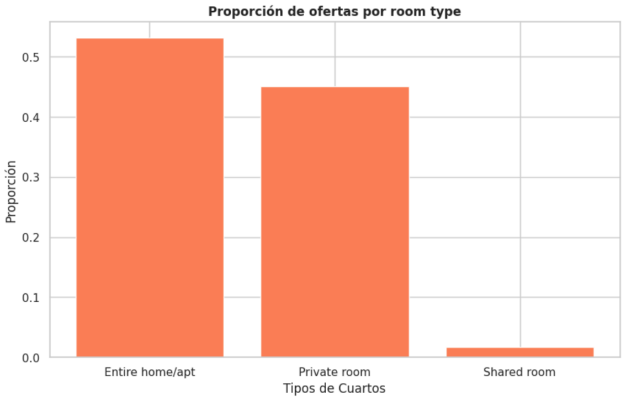
Además, realizamos un gráfico de barras (Figura 4) que visualiza estas proporciones de manera más clara, permitiendo una comparación directa entre los distintos barrios. Este gráfico muestra las diferencias en la proporción de oferentes por cada barrio, destacando la importancia de Manhattan y Brooklyn, que juntos representan la mayor parte de los oferentes.



Luego calculamos la proporción de oferentes por tipo de cuarto, dividiendo la cantidad de veces que aparece cada tipo de unidad por el total de filas de la variable room_type. Estas se pueden observar en la siguiente tabla:

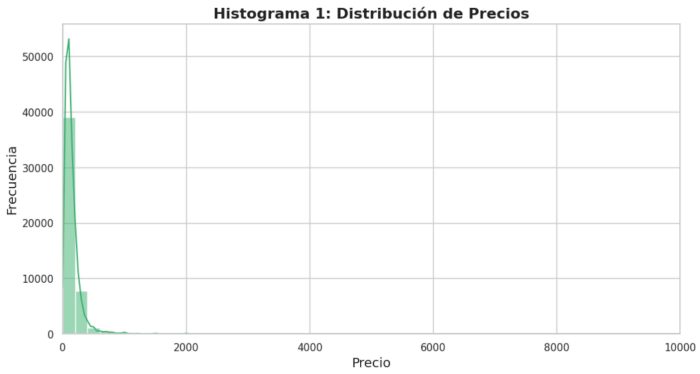
Entire home/apt	0.531329
Private room	0.451274
Shared room	0.017397

Nuevamente, realizamos un gráfico de barras para visualizar las proporciones (Figura 5). En él se ve claramente que la mayoría de unidades son casas o departamentos completos, luego habitaciones privadas y una proporción mucho menor de habitaciones compartidas.

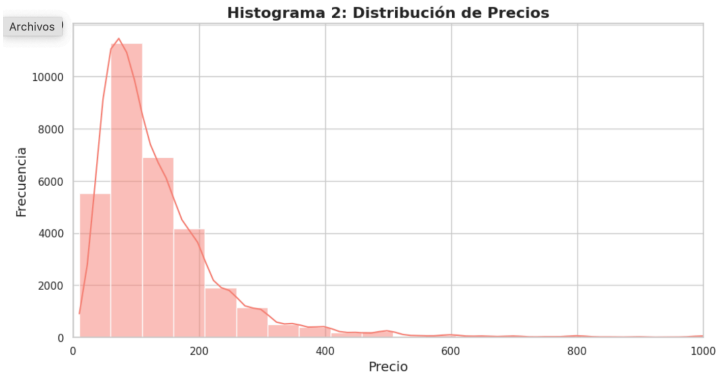


Ejercicio 4

Comenzamos realizando un histograma que incluía todos los precios del data frame llamado "df", para evaluar la distribución original de los precios, antes de filtrar los outliers y valores extremadamente altos (Figura 6). En este gráfico observamos que el precio de la gran mayoría de las unidades es menor a 2000 USD, no obstante, unas pocas alcanzan un costo de hasta 10.000 USD. El precio promedio del Data Frame original es de 152.7 USD.



Luego, realizamos un segundo histograma a partir del Data Frame "limpio" llamado "df_sin_outliers" (Figura 7). Además, luego de analizar el primer histograma, redujimos el rango del eje X a un máximo de 1000 para mejorar la visualización. Aún luego de haber reducido el rango de



precios, se puede observar que la gran mayoría de las unidades se concentran en un rango de precios de entre 0 y 400 USD. Con el Data Frame filtrado, el precio promedio es de 139.8, el precio máximo es de 2500 USD (este fue el criterio que establecimos al realizar el EDA) y el precio mínimo es de 10 USD (notar que eliminamos 11 unidades con precio igual a 0 USD).

Anteriormente habíamos calculado el precio promedio por neighbourhood_group y room_type; se pueden observar en las siguientes tablas.

Manhattan	174.078685
Brooklyn	120.511924
Staten Island	98.691892
Queens	95.982398
Bronx	80.617801

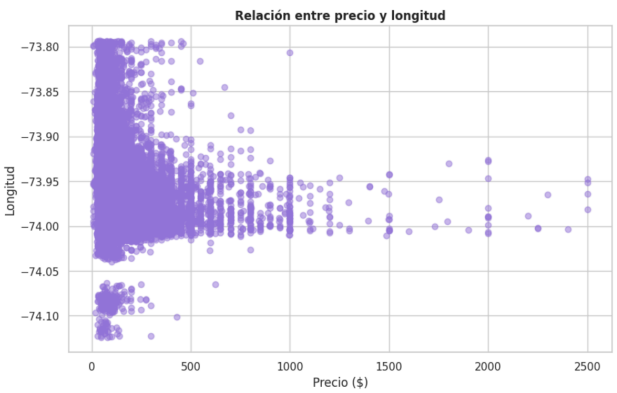
Entire home/apt	191.801597
Private room	81.116529
Shared room	71.756140

En el caso de los barrios, observamos que Manhattan tiene un precio promedio significativamente más alto, seguido por Brooklyn. Luego, los Stanten Island y Queens no presentan precios bastante similares. Por último, el barrio de Bronx tiene precios significativamente más bajos. Con respecto a los tipos de cuartos, las casas o departamentos completos tienen un precio promedio mucho más alto, en comparación con los cuartos privados o compartidos.

Ejercicio 5

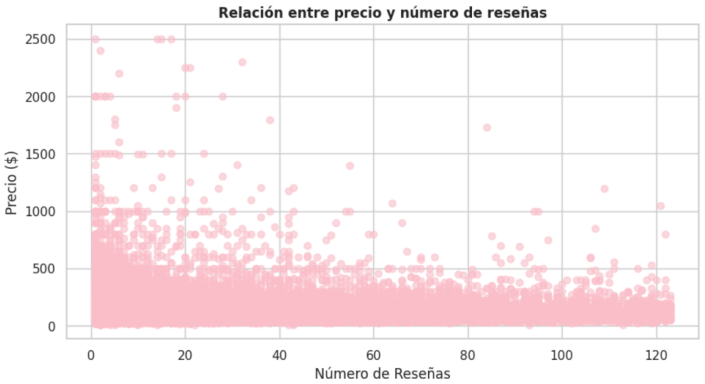
A continuación, realizamos scatter plots (diagramas de dispersión) para representar gráficamente la relación entre algunas variables e identificar tendencias generales. Para elegirlas tuvimos en cuenta la matriz de correlación realizada anteriormente. Además, nos enfocamos principalmente en analizar la variable “price”, ya que es la que buscamos predecir con el modelo de regresión en la parte III.

Comenzamos graficando las variables *longitude* y *price* (Figura 7) y observamos que la gran mayoría de unidades con precios relativamente altos se encuentran en un rango determinado de longitud (entre -73.9 y -74.05 aproximadamente). Es interesante destacar que Manhattan y Brooklyn (los barrios que tienden tener precios más altos) se encuentran aproximadamente en un punto longitudinal de -73.96, mientras que Bronx y Queens



(los barrios que tienden tener precios más bajos) se encuentra en un punto de -73.8 aproximadamente.

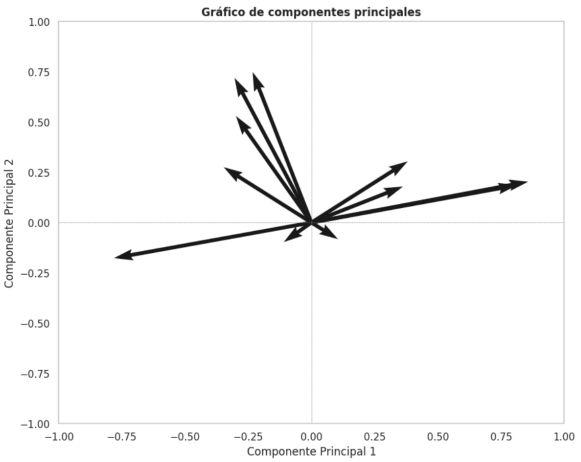
Continuamos graficando las variables number_of_reviews y price (Figura 8), para determinar si existe alguna relación entre el precio y la tendencia a dejar reseñas. Observamos que la mayoría de los puntos se agrupan en la parte baja del gráfico, con precios que están por debajo de los 500 USD, independientemente del número de reseñas. La cantidad de reseñas parece ir de 0 a 120 y no parecerían seguir una tendencia definida en relación al precio.



Ejercicio 6

A continuación, utilizamos el análisis de componentes principales para graficar las variables en dos dimensiones (Figura 9). Mediante este análisis identificamos que los dos primeros componentes explican el 39.52% de la varianza total. Esto significa que, aunque se está capturando una parte considerable de la variabilidad de los datos, aún queda un gran porcentaje de varianza no explicado por estos dos componentes.

Los loadings indican cuánto contribuye cada variable original a cada componente principal. En el gráfico, se representan como flechas que indican la relación y contribución de cada variable a los componentes principales. Observamos que dos variables relacionadas a la ubicación (*neighbourhood_group* y *longitude*) tienen un valor alto que sugiere que la ubicación está fuertemente relacionada con el primer componente. La variable *neighbourhood_group* tiene un valor alto positivo (0.857), por lo que parece contribuir al PC1 y *longitude* tiene un valor alto negativo (-0.781), es decir que está inversamente relacionada con PC1. En relación con *neighbourhood_group, offer_group* también tiene un valor alto (0.814), indicando que esta variable tiene una fuerte contribución al primer componente. Respecto al segundo componente, las variable*s number_of_reviews, reviews_per_month y availability_365* tienen valores positivos altos (0.748, 0.719 y 0.529), lo que significa que estas variables están fuertemente asociadas con el mismo. Nos parece importante mencionar que la variable price tiene un valor positivo pequeño en relación con ambos componentes (0.380 y 0.304 respectivamente), lo que indica que tiene una contribución moderada.



Parte 3: Predicción

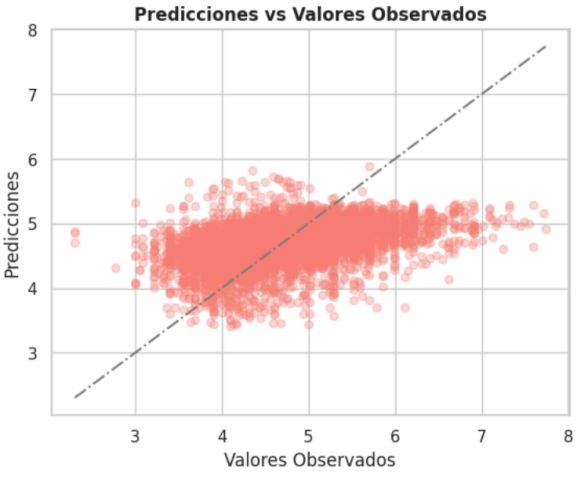
En primer lugar, utilizamos el comando *train_test_split*, asignando el 70% de los datos a la base de entrenamiento y el 30% a la base de prueba, con una semilla de 201 para garantizar la reproducibilidad de los resultados (Nguyen, 2020). La separación de los datos en conjuntos de entrenamiento y prueba es crucial en el desarrollo de modelos de machine learning. El conjunto de

entrenamiento se utiliza para ajustar el modelo, mientras que el conjunto de prueba sirve para evaluar su rendimiento en datos no vistos. Esto permite detectar problemas como el sobreajuste, donde un modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos (Bishop, 2006). Luego, definimos "price" como nuestra variable dependiente, que es la que buscamos predecir, y el resto de las variables como las independientes, que serán utilizadas para hacer las predicciones. Nuevamente, probamos aplicando transformaciones logaritmos a variables como “minimun_nights” y “number_of_reviews”. Como esto no generó un cambio significativo en el modelo, decidimos conservar el dataframe con las variables originales y únicamente le aplicamos logaritmo a la variable “price”.

Finalmente, creamos un modelo de regresión lineal utilizando LinearRegression. Luego, entrenamos el modelo con los datos de entrenamiento (X_train y y_train) y realizamos predicciones sobre el conjunto de prueba (X_test), generando el vector de predicciones y_pred. Para evaluar el rendimiento del modelo, calculamos el error cuadrático medio (MSE), obtuvimos un valor de 0.35, lo que indica que, en promedio, las predicciones del modelo se desvían en 0.35 unidades del valor real. Además, el coeficiente de determinación R^2 fue de 0.19, lo que sugiere que aproximadamente el 19% de la variabilidad en el precio se puede explicar por las variables independientes incluidas en el modelo. Estos resultados indican que, aunque el modelo proporciona información, su capacidad predictiva es limitada.

Por otro lado, calculamos los coeficientes del modelo. El intercept tendría un valor de aproximadamente -501.4, indicando el valor predicho de la variable dependiente (“price”) cuando todas las variables independientes son iguales a 0. El coeficiente de la variable “Latitud” es uno de los más altos, con un valor de -6.19, indicando que por cada unidad que aumenta la longitud, el precio disminuye en promedio 6.19 unidades. El coeficiente de la latitud es de 1.19, lo que significa que por cada unidad adicional en la latitud, el precio aumenta en promedio 1.19 unidades. Una mayor cantidad de alojamientos de anfitrión se relaciona con una ligera reducción del precio (-0.1025), aunque con un impacto casi insignificante. Por último, por cada aumento de una unidad de “number of reviews” el precio aumenta en promedio 0.0072 unidades; aunque el efecto es chico, un mayor número de reseñas parece estar asociado con precios más altos. Las demás variables tienen coeficientes mucho menores, con efectos mínimos en el precio.

Además de entrenar y evaluar el modelo de regresión lineal, visualizamos la relación entre los valores observados y las predicciones generadas por el modelo mediante un gráfico de dispersión (Figura 9). En este gráfico, los valores reales del conjunto de prueba se representan en el eje X, mientras que las predicciones del modelo se encuentran en el eje Y. Para facilitar la interpretación, incluimos una línea discontinua de referencia, que indica cómo se deberían distribuir los valores observados si las predicciones fueran perfectas. Al observar el gráfico vemos que, si bien los valores observados parecen seguir una tendencia lineal positiva, estos no se alejan de la línea de referencia. Esto sugiere que el modelo no es muy preciso en sus predicciones.



Bibliografía

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

IBM. (n.d.). *¿Qué es el análisis exploratorio de datos?* Recuperado el October 1, 2024, de https://www.ibm.com/topics/exploratory-data-analysis

Neural Ninja. (2023, 12 de junio). *Frequency encoding: counting categories for representation*. Recuperado el October 1, 2024, de https://letsdatascience.com/frequency-encoding/

Nguyen, M. (2020). *A guide on data analysis*. Bookdown. https://bookdown.org/mike/data_analysis/

R-bloggers. (2020, 19 de enero). *How to remove outliers in R*. R-bloggers. https://www.rbloggers.com/how-to-remove-outliers-in-r/

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association, 91*(434), 473–489.

Singh, S. (2018, May 21). *Understanding the bias-variance tradeoff*. Towards Data Science. https://towardsdatascience.com