Tecnológico de Monterrey

**Curso:**

Gestión de proyectos de plataformas tecnológicas (Gpo 201)

**Actividad:**

Actividad 3 (Regresión Logística)

**Elaborado Por:**

Sofia Villar García A01737544

**Fecha:** 14/10/2025

**Introducción**

En este proyecto se aplicaron diferentes modelos de regresión logística para analizar relaciones entre variables del conjunto de datos de *Inside Airbnb*, el cual contiene información detallada de alojamientos, anfitriones y sus características.

La regresión logística es una herramienta estadística que permite predecir resultados binarios o dicotómicos, es decir, variables que solo pueden tomar dos valores (por ejemplo: *sí/no*, *0/1*, *barato/caro*). A diferencia de la regresión lineal, este método no busca una relación numérica directa, sino la probabilidad de que algo ocurra en función de otras variables.

El objetivo principal fue identificar relaciones significativas entre variables y evaluar la capacidad predictiva de los modelos a través de métricas como precisión, exactitud y sensibilidad, además de aplicar métodos de reponderación de clases para equilibrar los datos en los casos donde las categorías estaban desbalanceadas.

1. **Métricas de evaluación:**
    - **Precisión:** mide qué tan exactas son las predicciones positivas del modelo.
    - **Exactitud:** mide el porcentaje total de aciertos (positivos y negativos).
    - **Sensibilidad:** mide la capacidad del modelo para reconocer correctamente los casos verdaderos.

Estas métricas permiten entender **qué tan confiable es el modelo** y si existen errores sistemáticos.

2. **Reponderación de clases:**

    En varios casos las categorías estaban muy desbalanceadas (por ejemplo, muchos anfitriones verificados y pocos no verificados). Por eso se aplicó este método, que asigna más peso a los ejemplos minoritarios, ayudando a que el modelo los detecte mejor aunque baje un poco la exactitud general.

    También se probaron técnicas de oversampling, duplicando algunos ejemplos minoritarios para equilibrar la cantidad de datos en cada clase.

**Resultados e interpretación**

Se construyeron 13 modelos de regresión logística, de los cuales 11 fueron normales y 2 reponderados.
Las métricas principales (precisión, exactitud y sensibilidad) se organizaron en una tabla general (ver archivo Excel compartido), y cada modelo se analizó individualmente para entender sus fortalezas y debilidades.

**Tabla de todos los coeficientes**

https://tecmx-my.sharepoint.com/:x:/g/personal/a01737544_tec_mx/EeBdG-bRl0ZHtutC_sBhTw4BniHfLKBMO6x6xYX_WVmwEQ?e=Vt8W0P

**Caso 1 — source (bathrooms, beds, price)**

- En este primer modelo los resultados fueron casi perfectos. La precisión y la sensibilidad estuvieron muy cerca de 1 tanto para *previous scrape* como para *city scrape*, y la exactitud fue de 0.9996. Esto significa que el modelo prácticamente no se equivoca al decir de qué fuente viene cada dato. Es tan preciso que parece que se aprendió de memoria el patrón de los datos, lo que podría indicar un poco de sobreajuste.
  Como el resultado es tan alto, puede que alguna variable, como el precio o el número de camas, esté muy relacionada con la fuente, así que el modelo no está entendiendo una relación general, sino una coincidencia. Aun así, el desempeño me parece excelente.

**Caso 2 — instant_bookable (host_acceptance_rate, price, minimum_minimum_nights)**

- En este modelo, la exactitud fue de 0.74, lo que indica que el modelo acierta más de la mitad, pero no es perfecto. Se nota que funciona mejor cuando predice que una propiedad no tiene reserva instantánea, mientras que le cuesta más acertar cuando sí la

tiene. Esto puede pasar porque hay más ejemplos de alojamientos sin reserva instantánea, o porque las variables que usamos no explican tan bien esa diferencia.

Considero que el modelo está "jugando a lo seguro", y casi siempre dice que no hay reserva instantánea. Para mejorar, habría que equilibrar los datos o agregar información sobre las políticas de los anfitriones o su rapidez de respuesta.

**Caso 3 — host_is_superhost (estimated_occupancy_l365d, review_scores_communication, review_scores_value)**

- Aquí el modelo intenta predecir si un anfitrión es superhost o no. Los resultados muestran que identifica mejor a los que no son superhost, pero tiene muchos errores con los que sí lo son. Esto significa que al modelo le cuesta reconocer a los superhosts reales, y además confunde a algunos anfitriones normales con superhosts.

  Probablemente esto pasa porque las variables que usamos no reflejan del todo las características de un superhost. Faltan cosas como el tiempo en la plataforma, las cancelaciones o la rapidez al responder. En general, el modelo funciona, pero aún necesita más información para ser confiable.

**Caso 4 — host_identity_verified (bathrooms, availability_eoy, availability_90)**

- En este modelo, el resultado muestra que el modelo casi siempre dice que los anfitriones sí tienen la identidad verificada, y prácticamente nunca predice lo contrario. La exactitud parece buena (0.91), pero en realidad eso pasa porque la mayoría de los datos son de personas verificadas. O sea, el modelo no aprendió a distinguir, solo repite la clase más común.

  Este es un caso claro de desbalance de datos, donde hay muchos de un tipo y pocos del otro. Para mejorar, habría que tener más ejemplos de anfitriones no verificados o darles más peso al entrenar el modelo.

**Caso 5 — host_identity_verified (reponderado)**

- Después de aplicar el método reponderado, los resultados mejoraron mucho. Ahora el modelo ya logra distinguir entre verificados y no verificados, aunque la exactitud bajó

un poco (a 0.70). Considero que no es malo, porque significa que ahora reconoce mejor ambos grupos y no solo el más grande.

Este modelo ya es más equilibrado, aunque todavía tiene margen de mejora. Podría ser más preciso si se agregan variables más específicas sobre la verificación, como si el anfitrión subió documentos o si tiene algún tipo de validación extra.

**Caso 6 — host_has_profile_pic (host_total_listings_count, host_listings_count, calculated_host_listings_count)**

- Este modelo intenta predecir si el anfitrión tiene foto de perfil. Sin embargo, el modelo casi siempre dice que sí tiene, porque la mayoría de los anfitriones efectivamente la tienen. Por eso la exactitud es alta (0.97), pero en realidad no está aprendiendo nada nuevo. Solo está repitiendo lo que más aparece.

  Esto significa que el modelo no sirve para detectar a los que no tienen foto. Se necesitarían más datos equilibrados o variables que de verdad estén relacionadas con tener o no tener una imagen de perfil.

**Caso 7 — host_has_profile_pic (reponderado)**

- Con el método reponderado, el modelo mejora bastante para detectar a los anfitriones que no tienen foto de perfil. Sin embargo, al hacerlo, se equivoca más con los que sí la tienen, y por eso la exactitud general baja a 0.53.

  Aun así, este cambio es positivo si lo que buscamos es no dejar pasar ningún caso sin foto. El modelo ahora reconoce mejor los dos grupos, aunque comete más errores. Es un buen intercambio si el objetivo es detectar perfiles incompletos o sospechosos.

**Caso 8 — beds (accommodates, bedrooms, price)**

- El modelo predice bastante bien los alojamientos con **menos de 3 camas**, pero se confunde con los que tienen más. Esto se debe a que hay muchos más alojamientos pequeños, así que el modelo "aprende" a clasificar casi todo como pequeño.
  Su exactitud general es alta (0.88), pero tiene un sesgo hacia las propiedades con pocas camas. Para mejorar, sería bueno incluir variables sobre el tamaño real del lugar, como los metros cuadrados o el número de habitaciones.

**Caso 9 — number_of_reviews (number_of_reviews_ltm, number_of_reviews_ly, reviews_per_month)**

- En este modelo, el modelo acierta muy bien con los alojamientos que tienen pocas reseñas, pero casi nunca acierta con los que tienen muchas. Esto muestra que el modelo está desbalanceado, porque los alojamientos con pocas reseñas son la mayoría.
  Esto puede ser un problema si se quiere identificar cuáles son los más populares, porque el modelo no los detecta bien. Para mejorarlo, sería bueno ajustar los valores de corte o agregar información sobre el tiempo activo del anuncio.

**Caso 10 — availability_30 (availability_eoy, availability_60, availability_90)**

- Este modelo tiene muy buen rendimiento. Predice con gran precisión la disponibilidad tanto en la primera mitad del mes como en la segunda, aunque le va un poco mejor en la primera. Su exactitud es de 0.93, lo cual es excelente.
  Este modelo es útil porque sí logra captar los patrones de ocupación, así que podría servir para planificar reservas o predecir cuántos alojamientos estarán disponibles según la fecha. Es de los más confiables de todos los casos.

**Caso 11 — price (accommodates, minimum_minimum_nights, minimum_nights)**

- Aquí el modelo trata de predecir si un alojamiento es caro o barato. Los resultados son bastante equilibrados, con una exactitud del 81%. Acierta un poco mejor con los alojamientos baratos, aunque también reconoce bien los caros.
  Esto puede pasar porque los alojamientos más caros suelen tener más variación en sus precios, mientras que los baratos son más parecidos entre sí. El modelo funciona bien,

pero se podría mejorar si se agregan datos sobre la ubicación o la calidad del alojamiento.

**Caso 12 — bathrooms (accommodates, availability_365, bedrooms)**

- En este modelo, el modelo detecta muy bien los alojamientos con menos de 2 baños, pero se equivoca con los que tienen más. De nuevo, esto se debe a que hay muchos más ejemplos de alojamientos pequeños, así que el modelo tiende a clasificar todo como si fuera pequeño.

  Aunque su exactitud total es de 0.75, en realidad le cuesta con las propiedades grandes o de lujo. Se podría mejorar agregando datos sobre el tamaño, tipo de inmueble o rango de precio.

**Caso 13 — accommodates (availability_365, bedrooms, calculated_host_listings_count_entire_homes)**

- Este modelo busca predecir cuántas personas puede recibir un alojamiento. Funciona bien con los alojamientos para pocos huéspedes, pero se complica con los que admiten a muchos. Tiene una buena exactitud general (0.88), pero es evidente que le cuesta más con los casos grandes.

  Esto pasa porque hay menos alojamientos de gran capacidad o porque son más variados. Aun así, el modelo describe bastante bien los patrones generales, y se podría mejorar si se agregan datos del tipo de propiedad o su superficie.

**Conclusión de los casos**

En general, los modelos funcionan muy bien para los casos más comunes, como alojamientos pequeños, anfitriones verificados o propiedades con pocos baños. Pero cuando hay menos ejemplos (como alojamientos grandes o caros), el modelo se confunde más. Cuando se usa el método reponderado, la exactitud puede bajar, pero eso no significa que el modelo sea peor; al contrario, significa que ahora reconoce mejor los casos que antes ignoraba. Los resultados muestran que la regresión logística es una herramienta útil para entender los patrones más

comunes, pero que necesita datos más equilibrados y variables más completas para representar bien los casos menos frecuentes.

**Comparación general entre los modelos**

Al comparar todos los casos, se observaron los siguientes patrones:

- Los modelos con mejor desempeño fueron los que tenían relaciones más directas entre las variables, como:
    - *source – (bathrooms, beds, price)*
    - *availability_30 – (availability_eoy, availability_60, availability_90)*
      Estos lograron una exactitud superior al 90%, mostrando que hay una fuerte relación entre esas variables.
- En cambio, los modelos con desbalance de clases (por ejemplo, *host_identity_verified* o *host_has_profile_pic*) presentaron exactitudes aparentemente altas, pero realmente solo aprendieron la clase más común.
  En esos casos, el modelo no sirve para distinguir correctamente entre categorías.
- Los modelos reponderados mejoraron notablemente la sensibilidad y el equilibrio entre clases, aunque su exactitud bajó un poco. Esto no significa que sean peores; al contrario, ahora el modelo reconoce mejor los casos minoritarios.
- En la mayoría de los modelos, las variables relacionadas con el tamaño o popularidad de los alojamientos (*beds, number_of_reviews, accommodates*) mostraron sesgos hacia los valores bajos. Esto demuestra que los datos están dominados por propiedades pequeñas, por lo que se necesita más información de alojamientos grandes o premium para mejorar la generalización.

**Conclusiones generales**

1. La regresión logística resultó ser una herramienta efectiva para analizar relaciones binarias dentro del conjunto de datos de Airbnb. Permite entender qué factores influyen en características como la verificación del anfitrión, la reserva instantánea o la disponibilidad del alojamiento.
2. Los modelos más confiables fueron aquellos con variables directamente relacionadas (como precios o disponibilidad), mientras que los más débiles fueron los que dependían de variables más subjetivas (como la verificación o el estado del perfil).
3. La reponderación de clases fue una técnica clave para corregir el sesgo hacia las clases mayoritarias. Aunque la exactitud bajó, los modelos se volvieron más justos y representativos.
4. En todos los casos, las métricas mostraron que la calidad de los datos influye mucho más que el tipo de modelo. Para mejorar los resultados, se recomienda:
    - Equilibrar las categorías antes de entrenar el modelo.
    - Incluir variables adicionales que aporten más contexto (como ubicación, tipo de alojamiento o fecha de registro).
    - Validar los modelos con nuevos datos para comprobar su capacidad de generalización.
5. Se logró cumplir con los objetivos del ejercicio: aplicar la regresión logística, comprender las métricas, usar reponderación cuando fue necesario, y generar interpretaciones comparativas para cada análisis.

Después de analizar todos los modelos, se puede concluir que los métodos de regresión logística y reponderación de clases permiten entender de manera clara las relaciones entre variables binarias dentro del conjunto de datos de Airbnb.

El análisis comparativo muestra que los modelos sin balance tienden a favorecer las categorías más grandes, mientras que los reponderados ofrecen resultados más equitativos y realistas.