



INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Actividad 3

José Francisco Vera Jimenez

Introducción

Para desarrollar la actividad se trabajó con diez casos distintos, cada uno con una variable objetivo particular y un conjunto de variables explicativas elegidas según su relevancia y su posible relación con el fenómeno estudiado. En algunos de estos casos fue necesario realizar ajustes adicionales, como el balanceo de clases o la aplicación de técnicas de sobremuestreo, con el propósito de mejorar la capacidad predictiva del modelo frente a datos desbalanceados. En cada escenario se evaluó el rendimiento del modelo mediante métricas de precisión, exactitud, sensibilidad y F1, lo que permitió comparar de manera objetiva los resultados obtenidos con y sin ajustes.

El proceso comenzó con la limpieza y preparación de los datos, tarea indispensable para garantizar la coherencia de las variables y evitar errores durante el entrenamiento del modelo. Se implementaron funciones específicas para estandarizar porcentajes, convertir precios a valores numéricos, codificar variables booleanas y dividir los conjuntos de entrenamiento y prueba de forma estratificada. Este procedimiento buscó no solo mejorar la calidad del dataset, sino también asegurar la reproducibilidad de los resultados y la comparabilidad entre los distintos modelos generados.

Caso 1: `instant_bookable ~ price + reviews_per_month`

En este primer modelo se buscó predecir si un alojamiento contaba con la opción de reserva instantánea, considerando como variables predictoras el precio del hospedaje y la cantidad de reseñas por mes. Tras limpiar los datos y estandarizar las variables, se observó un importante desbalance entre clases, por lo que se incorporó el parámetro `class_weight='balanced'` para ajustar los pesos del modelo.

El desempeño obtenido mostró una exactitud del 74.6 %, con una sensibilidad de 0.37 y un F1 de 0.30. Aunque la precisión fue baja (0.25), el modelo logró detectar con mayor frecuencia los casos positivos después del ajuste, lo que demuestra que el balanceo ayudó a compensar la diferencia entre clases, aunque a costa de una ligera reducción en la exactitud general.

Caso 2: `host_is_superhost ~ host_acceptance_rate + host_response_rate`

El segundo caso se centró en predecir si un anfitrión era superhost, utilizando como variables explicativas sus tasas de aceptación y respuesta. Ambas variables se transformaron a formato numérico mediante la función `clean_percent()`. Al igual que en el caso anterior, se detectó un fuerte desbalance de clases, por lo que se aplicó

sobremuestreo (oversampling) con RandomOverSampler únicamente en el conjunto de entrenamiento.

Los resultados mostraron una exactitud del 60.7 %, una precisión de 0.36, una sensibilidad de 0.85 y un F1 de 0.51. En comparación con el modelo sin ajuste, el incremento de la sensibilidad fue significativo, lo que indica que el oversampling permitió al modelo identificar con mucha mayor eficacia a los anfitriones superhost, aun cuando la precisión disminuyó ligeramente.

Caso 3: high_availability ~ price + availability_30 + availability_90

En este escenario se generó una variable binaria denominada high_availability, que tomaba el valor de 1 cuando la disponibilidad en 30 días era mayor a 15 o en 90 días mayor a 45. El objetivo fue estimar qué tan predecible era la alta disponibilidad de un alojamiento en función del precio y de las disponibilidades mensuales.

El modelo obtuvo un desempeño sobresaliente, con una precisión de 0.95, exactitud de 0.96, sensibilidad de 0.95 y un F1 de 0.95. Esto sugiere una relación clara entre las variables, ya que las propiedades con alta disponibilidad mostraron un comportamiento consistentemente distinguible. Este caso se considera uno de los más sólidos dentro del ejercicio.

Caso 4: host_has_profile_pic ~ host_identity_verified

Este caso evaluó la relación entre el hecho de que un anfitrión tuviera foto de perfil y si su identidad estaba verificada. Ambas variables fueron codificadas como binarias mediante encode_bool(). A pesar de la simplicidad del modelo, los resultados fueron excelentes: una exactitud de 95.5 %, una sensibilidad de 1.00, una precisión de 0.95 y un F1 de 0.97.

El modelo predijo casi perfectamente los casos positivos, confirmando que la mayoría de los anfitriones verificados también cuentan con una fotografía, lo que puede interpretarse como una política común dentro de la plataforma.

Caso 5: property_type_bin (Entire home=1) ~ accommodates + bathrooms

En este modelo se transformó la variable property_type en una variable binaria que diferenciara entre viviendas completas y otros tipos de alojamiento. Las variables predictoras fueron la capacidad del alojamiento y el número de baños.

Los resultados mostraron un desempeño alto, con exactitud del 85.9 %, precisión de 0.86, sensibilidad de 0.97 y F1 de 0.91. El modelo logró identificar con gran exactitud los alojamientos que correspondían a viviendas completas, lo que evidencia que la combinación de estas dos variables estructurales resulta muy efectiva para este tipo de clasificación.

Caso 6: room_type_bin (Private room=1) ~ price + accommodates

Aquí se modeló la probabilidad de que el alojamiento fuera una habitación privada, en función de su precio y capacidad. Tras limpiar los datos, el modelo arrojó una exactitud del 88 %, con precisión de 0.78, sensibilidad de 0.38 y F1 de 0.51.

Aunque la exactitud global fue buena, la baja sensibilidad muestra que el modelo tuvo dificultades para identificar correctamente los casos positivos, probablemente porque la variable precio presenta una alta variabilidad y no discrimina adecuadamente entre habitaciones privadas y otros tipos de alojamiento.

Caso 7: host_identity_verified ~ host_is_superhost + host_total_listings_count

En este caso se intentó predecir si un anfitrión tenía su identidad verificada, utilizando como predictores el estatus de superhost y el número total de propiedades administradas.

El modelo arrojó una precisión de 0.88, exactitud de 0.88, sensibilidad de 1.00 y un F1 de 0.93. Estos resultados reflejan una fuerte relación entre las variables, ya que los superhosts suelen estar verificados y contar con más propiedades, lo cual refuerza la coherencia de la base de datos y la calidad de este modelo.

Caso 8: review_scores_rating_bin (>90) ~ reviews_per_month + price

El octavo caso buscó clasificar las propiedades con una puntuación de reseñas superior a 90 en función de sus reseñas mensuales y su precio. Sin embargo, la variable objetivo presentó muy poca variación, volviéndose prácticamente monoclasa tras la limpieza.

Como consecuencia, el modelo no logró distinguir adecuadamente entre categorías, mostrando valores nulos en precisión, sensibilidad y F1, y una exactitud apenas superior al 60 %. Este resultado demuestra que la falta de variabilidad en la variable dependiente impide que el modelo aprenda patrones útiles.

Caso 9: availability_365_bin (>200) ~ price + accommodates + bathrooms

En este caso se modeló la probabilidad de que una propiedad tuviera más de 200 días disponibles al año, a partir de su precio, capacidad y número de baños. El modelo mostró un desempeño deficiente, con precisión de 0.50, exactitud de 0.64, sensibilidad de 0.005 y F1 de 0.01.

El resultado indica una clara dificultad para identificar las propiedades con alta disponibilidad anual, lo que puede deberse a un fuerte desbalance entre clases o a la falta de relación directa entre las variables seleccionadas y el fenómeno que se intenta predecir.

Caso 10: high_demand (number_of_reviews_ltm>0) ~ price + accommodates

El último modelo se enfocó en estimar la probabilidad de que un alojamiento tuviera alta demanda, medida a través de la existencia de reseñas recientes. Dado que muchas propiedades carecían de reseñas, el conjunto de datos resultó desbalanceado, por lo que se aplicó el parámetro `class_weight='balanced'`.

El modelo ajustado alcanzó una precisión de 0.62, exactitud de 0.53, sensibilidad de 0.48 y F1 de 0.54. Aunque la exactitud se redujo ligeramente, el modelo fue más equilibrado y logró detectar un mayor número de casos positivos que la versión sin ajuste, demostrando la utilidad de los métodos de balanceo en situaciones de desigualdad de clases.

Discusión y análisis comparativo de resultados

Una vez obtenidos los resultados individuales de cada modelo, se procedió a realizar un análisis comparativo que permitiera comprender el comportamiento general de la regresión logística en diferentes contextos dentro del mismo conjunto de datos. La comparación se centró especialmente en los efectos de los ajustes aplicados como el balanceo de clases o el sobremuestreo y en cómo estos influyeron en las métricas de desempeño, principalmente en la sensibilidad y el F1 score, que son las más representativas en escenarios con clases desiguales.

El primer aspecto relevante fue constatar que los modelos sin ajuste tendieron a presentar una exactitud alta pero una sensibilidad muy baja. Esto significa que, aunque acertaban con frecuencia al predecir la clase mayoritaria, prácticamente ignoraban los casos pertenecientes a la clase minoritaria. Este patrón se observó con claridad en los primeros ensayos del caso 1 (reservas instantáneas) y el caso 2 (superhost), donde la sensibilidad

inicial era cercana a cero. Sin embargo, tras aplicar el ajuste correspondiente (`class_weight='balanced'` o `RandomOverSampler`), el comportamiento del modelo cambió de manera significativa: la sensibilidad aumentó considerablemente, reflejando una mejor capacidad de detección de la clase minoritaria, aunque la exactitud global se redujo levemente.

El caso 3 y el caso 5 se destacaron por obtener los mejores resultados sin necesidad de ajustes adicionales. En ambos, las variables predictoras se mostraron sólidamente asociadas a la variable dependiente, lo que permitió alcanzar valores de F1 superiores a 0.9. Esto sugiere que la regresión logística puede comportarse de manera óptima cuando existe una relación clara entre las variables explicativas y la categoría de salida, y cuando las proporciones entre clases son relativamente equilibradas.

En contraste, el caso 8 y el caso 9 ilustraron situaciones en las que la técnica no logró un rendimiento aceptable. En el primero, el problema surgió de una variable objetivo prácticamente monoclasa, lo que impidió al modelo distinguir entre categorías. En el segundo, aunque las variables estaban numéricamente bien definidas, su relación con el fenómeno de disponibilidad anual no resultó suficientemente informativa, lo que se tradujo en una sensibilidad cercana a cero. Estos dos casos evidencian que la regresión logística no puede superar las limitaciones intrínsecas del dataset y que el preprocesamiento, así como la selección de variables, juegan un papel determinante en la validez del modelo.

El caso 10, por su parte, representó un ejemplo intermedio: si bien el modelo balanceado sacrificó exactitud (pasando de 0.73 a 0.52), logró aumentar la capacidad de detección de casos de alta demanda y obtuvo un F1 más equilibrado (0.54). Este comportamiento demuestra que el balanceo de clases no necesariamente mejora todos los indicadores simultáneamente, sino que permite un intercambio más justo entre la capacidad de clasificación general y la identificación correcta de las categorías menos representadas.

En términos generales, el análisis comparativo evidenció que las técnicas de ajuste mejoraron el comportamiento del modelo frente al desbalance, especialmente en contextos donde la clase positiva era poco frecuente. Asimismo, se observó que la exactitud, aunque útil, puede ser un indicador engañoso cuando las clases no están equilibradas, ya que puede aparentar un buen rendimiento sin reflejar la verdadera capacidad del modelo para reconocer los casos relevantes. Por esta razón, métricas como la sensibilidad y el F1 resultaron más adecuadas para la interpretación de los resultados en este estudio.

Finalmente, al revisar todos los casos, se puede concluir que la regresión logística mostró un desempeño sólido y coherente con las características de los datos. Los mejores resultados se concentraron en escenarios donde las variables predictoras representaban de forma directa un comportamiento estructural del alojamiento o del anfitrión, mientras que los resultados más débiles se relacionaron con variables derivadas o altamente desbalanceadas. El trabajo permitió así comprender tanto el potencial como las limitaciones de esta técnica en la clasificación binaria de datos reales.

Interpretación de la tabla comparativa

CASO 1: `instant_bookable ~ price + reviews_per_month`

	Métrica	SIN AJUSTE	CON AJUSTE
0	Precisión	0.364	0.259
1	Exactitud	0.848	0.746
2	Sensibilidad	0.017	0.372
3	F1	0.032	0.305

CASO 10: `high_demand_ltm ~ price + accommodates`

	Métrica	SIN AJUSTE	CON AJUSTE
0	Precisión	0.584	0.623
1	Exactitud	0.584	0.529
2	Sensibilidad	0.998	0.487
3	F1	0.737	0.546

CASO 2: `host_is_superhost ~ host_acceptance_rate + host_response_rate (Oversampling)`

	Métrica	SIN AJUSTE	CON AJUSTE
0	Precisión	0.000	0.365
1	Exactitud	0.759	0.607
2	Sensibilidad	0.000	0.852
3	F1	0.000	0.511

Para integrar los resultados de todos los modelos se elaboró una tabla comparativa que reunió las métricas de los tres casos en los que se aplicaron ajustes de balanceo o sobremuestreo: el caso 1 (`instant_bookable`), el caso 2 (`host_is_superhost`) y el caso 10 (`high_demand`). En esta tabla se contrastaron los valores de precisión, exactitud, sensibilidad y F1 tanto en los modelos sin ajuste como en los modelos ajustados, con el propósito de observar de forma global los efectos que las técnicas de corrección de desbalance tuvieron sobre el rendimiento.


Al analizar los datos, se identificó un patrón consistente: los modelos sin ajuste tendieron a ofrecer mayor exactitud y precisión, mientras que los modelos con ajuste destacaron por mayor sensibilidad y un F1 más equilibrado. Esto indica que, al redistribuir la importancia de las clases o al aumentar los ejemplos de la clase minoritaria, el modelo logra detectar mejor los casos positivos, aun cuando ello implique una reducción en la proporción total de aciertos. Por ejemplo, en el caso 2, el uso del oversampling permitió elevar la sensibilidad de 0.0 a 0.85, mostrando un cambio radical en la capacidad del modelo para reconocer anfitriones superhost, lo que antes era prácticamente inexistente.

De igual manera, el caso 1 demostró cómo el balanceo de clases puede mejorar el reconocimiento de reservas instantáneas, pasando de una sensibilidad mínima a un valor más representativo (0.37). Aunque la exactitud bajó ligeramente, el modelo se volvió más útil desde el punto de vista interpretativo, al no centrarse únicamente en la clase predominante. En el caso 10, el ajuste produjo un efecto más moderado, pero aun así se observó una mejor relación entre las métricas, especialmente en el F1, que combina precisión y sensibilidad.

Esta tabla comparativa sirvió para visualizar de manera resumida el comportamiento de la regresión logística bajo diferentes condiciones de equilibrio de clases, confirmando que no existe una única configuración óptima para todos los casos, sino que el tipo de ajuste debe seleccionarse con base en la naturaleza de los datos y el objetivo del modelo. En situaciones donde el interés principal es identificar correctamente los casos positivos (como en el caso de los superhosts o la alta demanda), el sacrificio de exactitud es aceptable si se gana capacidad de detección. Por el contrario, en contextos donde ambas clases están equilibradas o el objetivo es mantener un desempeño general alto, los modelos sin ajuste pueden resultar suficientes.

Análisis de la tabla general de los diez casos

Además de la comparación entre los modelos ajustados y no ajustados, se elaboró una tabla general con los resultados de los diez casos trabajados. Esta tabla permitió observar de manera conjunta cómo varía el comportamiento de la regresión logística dependiendo del tipo de variable, el equilibrio entre clases y la relación existente entre las variables independientes y la dependiente.

 TABLA FINAL COMPARATIVA (3 ajustes: SIN vs CON)

	Caso	Métrica	SIN AJUSTE	CON AJUSTE
0	1. instant_bookable	Precisión	0.364	0.259
1	1. instant_bookable	Exactitud	0.848	0.746
2	1. instant_bookable	Sensibilidad	0.017	0.372
3	1. instant_bookable	F1	0.032	0.305
4	2. host_is_superhost	Precisión	0.000	0.365
5	2. host_is_superhost	Exactitud	0.759	0.607
6	2. host_is_superhost	Sensibilidad	0.000	0.852
7	2. host_is_superhost	F1	0.000	0.511
8	10. high_demand	Precisión	0.584	0.623
9	10. high_demand	Exactitud	0.584	0.529
10	10. high_demand	Sensibilidad	0.998	0.487
11	10. high_demand	F1	0.737	0.546

Al revisar las métricas, se nota una diferencia clara entre los modelos. Algunos alcanzaron un desempeño muy alto, especialmente los casos 3, 4, 5 y 7, donde la exactitud y el F1 superaron el 90 %. En estos escenarios las variables utilizadas describían con claridad el fenómeno a predecir, por lo que el modelo logró distinguir con facilidad entre las clases. En el caso de la disponibilidad y del tipo de propiedad, por ejemplo, las relaciones eran evidentes y los datos estaban bien definidos, lo que facilitó que la regresión logística captara los patrones sin necesidad de ajustes adicionales.

En cambio, los casos 8 y 9 mostraron el extremo opuesto. En el primero, la variable de puntuación de reseñas prácticamente no tenía variación, lo que hizo imposible que el modelo aprendiera algo significativo. En el segundo, la disponibilidad anual no tenía una correspondencia clara con las demás variables, y eso se reflejó en una sensibilidad casi nula. Ambos ejemplos muestran que el modelo no falla por sí mismo, sino porque los datos no le ofrecen una señal clara que seguir.

Los demás casos, como el 1, 2, 6 y 10, se ubicaron en un punto intermedio. Los resultados fueron aceptables, pero dependieron mucho de los ajustes aplicados para mejorar el reconocimiento de la clase minoritaria. En estos ejemplos, las técnicas de balanceo o de sobremuestreo ayudaron a que el modelo dejara de favorecer siempre a la clase dominante, aunque eso implicó perder un poco de exactitud.