INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

CHRISTOPHER AGUILAR MÉNDEZ
BLANCA FLOR VISCA COCOTZIN
ÁNGEL FERNANDO RUÍZ VÁSQUEZ

31 DE MARZO DE 2025
INGENIERÍA DE SOFTWARE I
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Análisis de Regresión Logística en Cuatro Ciudades

1. Introducción

En este reporte se analiza la regresión logística aplicada a datos de alojamiento en cuatro ciudades diferentes. Se busca evaluar cómo diversas variables independientes afectan ciertas variables objetivo y comparar el desempeño del modelo entre las distintas ciudades.

2. Definición y Justificación de las Variables

2.1 Variables Objetivo (Dependientes)

- host_is_superhost: Indica si el anfitrión es "superhost", lo que puede influir en la demanda.
- host_has_profile_pic: Señala si el anfitrión tiene foto de perfil, afectando la confianza del huésped.
- host_identity_verified: Indica si la identidad del anfitrión está verificada, influyendo en la credibilidad.
- has_availability: Muestra si el alojamiento está disponible, clave para analizar ocupación.
- instant_bookable: Indica si la reserva es inmediata, factor que facilita la elección del huésped.
- price: Precio del alojamiento, evaluado en relación con el promedio.
- property_type: Tipo de propiedad, diferenciando entre unidad completa y otras opciones.
- accommodates: Capacidad de huéspedes, influyendo en la demanda.
- room_type: Tipo de habitación ofrecida, afectando la preferencia del huésped.
- review_scores_rating: Puntuación general de reseñas, clave en la percepción de calidad.

2.2 Variables Independientes

- host_response_rate: Tasa de respuesta del anfitrión, influyendo en la confianza del huésped.
- number_of_reviews: Cantidad de reseñas, relacionada con la reputación del alojamiento.
- review_scores_rating: Influye en otras métricas de calidad y demanda.
- review_scores_communication: Evalúa la comunicación del anfitrión.
- availability_365: Días disponibles en un año, clave para analizar ocupación.
- minimum_nights: Mínimo de noches por reserva, afectando la flexibilidad.
- maximum_nights: Máximo de noches permitidas, determinante en la ocupación.
- availability_30: Disponibilidad en los próximos 30 días, métrica de corto plazo.
- accommodates: También se usa para evaluar su impacto en otras métricas.

- bedrooms: Número de habitaciones, influyendo en el precio y demanda.
- bathrooms: Cantidad de baños, afecta la comodidad y precio.
- price: Factor clave en la demanda y accesibilidad.
- review_scores_cleanliness: Puntuación de limpieza, importante en la satisfacción del huésped.
- review_scores_value: Relación calidad-precio, influye en la decisión de los huéspedes.
- instant_bookable: También se evalúa como factor que afecta disponibilidad y demanda.

2.3 Variables Convertidas a Dicotómicas

Para facilitar el análisis en la regresión logística, algunas variables se han convertido en binarias:

- price (Arriba/Abajo del promedio)
- property_type (Entire vs. otros)
- accommodates (Arriba/Abajo del promedio)
- room_type (Apt vs. otros)
- review_scores_rating (Arriba/Abajo del promedio)

3. Proceso General de la Regresión Logística

3.1 Carga y Preprocesamiento de Datos

- Importación de librerías y lectura del dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.special as special
from scipy.optimize import curve_fit
import seaborn as sns
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

3.2 Creación de los dataframes con las variables a utilizar

- Se crean dos dataframes para cada tipo de variable, las dependientes y las independientes:

```
df_Dep = df[['host_is_superhost', 'host_has_profile_pic',
             'host_identity_verified', 'has_availability',
             'instant_bookable', 'price', 'property_type',
             'accommodates', 'room_type', 'review_scores_rating']].copy()
```

```
df_Ind = df[['host_response_rate', 'number_of_reviews',
             'review_scores_rating', 'review_scores_communication',
             'availability_365', 'minimum_nights', 'maximum_nights',
             'availability_30', 'accommodates', 'bedrooms',
             'bathrooms', 'beds', 'price', 'review_scores_cleanliness',
             'review_scores_value', 'instant_bookable']].copy()
```

3.3 Conversión de variables a variables dicotómicas

Por la naturaleza de las variables elegidas tenemos dos tipos de conversiones dicotómicas de los cuales podemos observar sus implementaciones:

- Por promedio:

```
df_Dep['price'] = df_Dep['price'].apply(lambda x: 1 if x > df_Dep['price'].mean() else 0)
df_Dep['price'] = df_Dep['price'].replace({1: 'higher', 0: 'lower'})
df_Dep['price'].value_counts()
```

```
price
lower     12872
higher     5864
Name: count, dtype: int64
```

- Por categoría:

```
# Convertimos la variable property_type a dicotómica con Entire rental unit y el resto other
df_Dep['property_type'] = df_Dep['property_type'].apply(lambda x: 1 if x == 'Entire rental unit' else 0)
df_Dep['property_type'] = df_Dep['property_type'].replace({1: 'entire', 0: 'other'})
df_Dep['property_type'].value_counts()
```

```
property_type
entire    11288
other      7448
Name: count, dtype: int64
```

3.4 Variables independientes usadas por variable dependiente

Variables Dependientes	Variables Independientes
host_is_superhost	host_response_rate, number_of_reviews, review_scores_rating
host_has_profile_pic	host_response_rate, number_of_reviews, review_scores_communication
host_identity_verified	host_response_rate, number_of_reviews, availability_365
has_availability	minimum_nights, maximum_nights, availability_30, availability_365
instant_bookable	host_response_rate, number_of_reviews, review_scores_rating

price (Arriba/Abajo promedio)	accommodates, bedrooms, bathrooms, number_of_reviews, review_scores_rating
property_type (Entire vs. otros)	accommodates, bedrooms, price, review_scores_rating
accommodates (Arriba/Abajo promedio)	bedrooms, bathrooms, beds, price, number_of_reviews
room_type (Entire vs. otros)	accommodates, bedrooms, price, instant_bookable
review_scores_rating (Alto/Bajo)	number_of_reviews, review_scores_cleanliness, review_scores_communication, review_scores_value

3.5 Entrenamiento del Modelo (Ejemplo usando la variable host_is_superhost)

- Declaramos las variables dependientes e independientes para la regresión Logística

```
Vars_Indep = df_Ind[['host_response_rate', 'number_of_reviews', 'review_scores_rating']]
Vars_Dep = df_Dep['host_is_superhost']
```

- Redefinimos las variables

```
X = Vars_Indep
y = Vars_Dep
```

- Dividimos el conjunto de datos en la parte de entrenamiento y prueba

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=None)
```

- Se escalan todos los datos

```
escalar = StandardScaler()
```

- Para realizar el escalamiento de las variables "X" tanto de entrenamiento como de prueba, utilizamos el método "fit_transform"

```
X_train = escalar.fit_transform(X_train)
X_test = escalar.transform(X_test)
```

- Definimos el algoritmo a utilizar

```
from sklearn.linear_model import LogisticRegression
algoritmo = LogisticRegression()
```

- Entrenamos el modelo

```
algoritmo.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression()
```

- Realizamos una predicción

```
y_pred = algoritmo.predict(X_test)
y_pred
```

```
array(['f', 'f', 'f', ..., 'f', 'f', 't'], dtype=object)
```

- Verificamos la matriz de confusión

```
from sklearn.metrics import confusion_matrix
matriz = confusion_matrix(y_test, y_pred)
print('Matriz de Confusión:')
print(matriz)
```

```
Matriz de Confusión:
[[3233  270]
 [1460  658]]
```

4. Comparación de Resultados Entre Ciudades

A continuación, se presentan los coeficientes de Precisión, Exactitud y Sensibilidad obtenidos para cada ciudad:

TOKIO

VARIABLE	Precisión	Exactitud	Sensibilidad
HOST_IS_SUPERHOST	70.9%	69.2%	92.2%
HOST_HAS_PROFILE_PIC	98.8%	98.8%	0.0%
HOST_IDENTITY_VERIFIED	96.9%	96.9%	0.0%
HAS_AVAILABILITY	99.9%	99.9%	0.0%
INSTANT_BOOKABLE	75.0%	75.0%	0.0%
PRICE	70.8%	77.2%	91.7%
PROPERTY_TYPE	64.0%	62.1%	28.1%
ACCOMMODATES	79.0%	84.0%	92.1%
ROOM_TYPE	82.2%	88.2%	0.06%
REVIEW_SCORES_RATING	84.2%	82.6%	77.0%

AMSTERDAM

VARIABLE	Precisión	Exactitud	Sensibilidad
HOST_IS_SUPERHOST	75.6%	76.5%	100%
HOST_HAS_PROFILE_PIC	99.0%	99.0%	0.0%
HOST_IDENTITY_VERIFIED	98.3%	98.3%	0.0%
HAS_AVAILABILITY	99.6%	99.6%	0.0%
INSTANT_BOOKABLE	46.6%	83.8%	98.2%
PRICE	29.4%	61.8%	97.4%
PROPERTY_TYPE	76.5%	77.3%	38.3%
ACCOMMODATES	86.9%	88.3%	89.0%
ROOM_TYPE	83.5%	81.7%	52.0%
REVIEW_SCORES_RATING	84.2%	84.0%	69.0%

COPENHAGUE

VARIABLE	Precisión	Exactitud	Sensibilidad
HOST_IS_SUPERHOST	59.3	88.1	98.8
HOST_HAS_PROFILE_PIC	59.3	88.1	12.3
HOST_IDENTITY_VERIFIED	88.4	88.4	0.0%
HAS_AVAILABILITY	98.0	98.0%	0.0%
INSTANT_BOOKABLE	40.0	91.0%	99.8
PRICE	60.8%	59.7	81.0
PROPERTY_TYPE	91.8	91.7	12.7
ACCOMMODATES	87.0	86.9	90.9
ROOM_TYPE	93.1	93.0	27.6
REVIEW_SCORES_RATING	37.2%	90.2%	99.7%

CDMX

VARIABLE	Precisión	Exactitud	Sensibilidad
HOST_IS_SUPERHOST	66.1%	68.4%	87.2%
HOST_HAS_PROFILE_PIC	98.2%	98.2%	0.0%
HOST_IDENTITY_VERIFIED	95.5%	95.5%	0.0%
HAS_AVAILABILITY	96.1%	96.1%	0.0%
INSTANT_BOOKABLE	58.8%	61.0%	96.8%
PRICE	62.9%	78.4%	92.8%
PROPERTY_TYPE	56.7%	64.1%	88.9%
ACCOMMODATES	78.6%	81.2%	83.4%
ROOM_TYPE	86.1%	77.0%	0.76%
REVIEW_SCORES_RATING	87.3%	86.8%	72.6%

5.Conclusiones y Observaciones

5.1 Interpretación General de los Resultados

Los resultados obtenidos mediante la regresión logística muestran que ciertos factores tienen un impacto significativo en la predicción de variables clave dentro del mercado de alojamientos en distintas ciudades. En general, variables como `host_is_superhost`, `accommodates` y `review_scores_rating` muestran una precisión y exactitud relativamente altas en la mayoría de las ciudades, lo que indica que son factores predictivos importantes.

Sin embargo, la sensibilidad varía significativamente entre las variables y las ciudades. Por ejemplo, `instant_bookable` presenta una alta sensibilidad en algunas ciudades (como Copenhague y Ámsterdam) pero baja precisión y exactitud, lo que sugiere que el modelo detecta muchas instancias positivas, pero también tiene una alta tasa de falsos positivos. Esto podría significar que la variable está influenciada por otros factores no considerados en el modelo.

Otra observación relevante es que la variable `host_has_profile_pic` tiene una precisión y exactitud muy altas en todas las ciudades, pero su sensibilidad es 0 % en casi todos los casos. Esto indica que el modelo casi nunca predice correctamente la ausencia de una foto de perfil, lo que sugiere que esta variable no tiene suficiente variabilidad o que su influencia en la regresión logística es mínima.

5.2 Comparación entre Ciudades y Análisis de Diferencias

Al comparar los resultados entre ciudades, se pueden identificar varias diferencias clave:

1. **Tokio:**
   - Presenta una precisión y exactitud altas en variables como `host_is_superhost` (70.9% y 69.2%, respectivamente) y `accommodates` (79% y 84%).
   - La sensibilidad de `host_is_superhost` y `price` es notablemente alta (92.2% y 91.7%), lo que indica que el modelo predice correctamente los casos positivos en estas variables con mayor frecuencia.
   - Sin embargo, variables como `host_has_profile_pic`, `host_identity_verified` y `has_availability` tienen sensibilidad de 0 %, lo que sugiere que el modelo no predice bien las clases minoritarias en estos casos.
2. **Ámsterdam:**
   - Presenta una alta sensibilidad en `host_is_superhost` (100%) y `instant_bookable` (98.2%), pero baja precisión en instant_bookable (46.6%), lo que podría indicar un alto número de falsos positivos.
   - La variable price tiene una precisión muy baja (29.4%) en comparación con otras ciudades, lo que podría sugerir que los precios en Ámsterdam tienen una mayor variabilidad o que otros factores influyen más en esta variable.
   - Variables como `host_has_profile_pic` y `host_identity_verified` tienen alta precisión pero sensibilidad 0 %, similar a lo observado en Tokio.
3. **Copenhague:**
   - Se observa un desempeño irregular con variaciones notables entre las métricas. Por ejemplo, `host_is_superhost` tiene una sensibilidad muy alta (98.8%) pero precisión relativamente baja (59.3%).
   - `review_scores_rating` tiene una de las sensibilidades más altas (99.7%), pero una precisión muy baja (37.2%), lo que indica que el modelo tiende a sobreestimar las calificaciones positivas.
   - `instant_bookable` también muestra alta sensibilidad (99.8%) pero baja precisión (40%), similar a Ámsterdam.
4. **Ciudad de México (CDMX):**
   - Se encuentra un desempeño intermedio en la mayoría de las variables, con valores de precisión, exactitud y sensibilidad más equilibrados.
   - `price` tiene una mejor precisión (62.9%) y exactitud (78.4%) en comparación con Ámsterdam.

- `property_type` tiene una baja precisión (56.7%) pero una alta sensibilidad (88.9%), lo que indica que el modelo tiende a predecir muchos casos positivos, pero con menor certeza.

5.3 Conclusión

Los resultados muestran que la efectividad de la regresión logística varía entre ciudades debido a factores específicos de cada mercado de alojamiento. Tokio y CDMX presentan un desempeño más equilibrado, mientras que en Copenhague y Ámsterdam hay mayor variabilidad en los valores de sensibilidad y precisión, indicando posibles problemas con la distribución de los datos o la influencia de otros factores no considerados en el modelo.