31 marzo 2025

Reporte. Regresión logística

RAMÍREZ DORANTES PAULINA

PALACIOS FERNÁNDEZ PAOLA

LÓPEZ MARTÍNEZ PÉREZ MIRIAM FLORENCIA

ÍNDICE

1 Introducción

2 Justificación de las variables

3 Desarrollo general

4 Resultados. Ciudad de México

5 Resultados. Bolonia

6 Resultados. Florencia

7 Conclusiones

8 Conclusiones

INTRODUCCIÓN

El análisis de bases de datos de la plataforma de airbnb en distintas ciudades, nos permite identificar patrones y tendencias clave que pueden resultar de beneficio para nuestros clientes.

El objetivo principal fue evaluar la influencia de diversas variables en la probabilidad de que una propiedad sea señalada con una demanda alta.
A través de este enfoque, buscamos identificar factores determinantes en la popularidad de alojamientos y explorar posibles diferencias entre los mercados de las ciudades analizadas: Florencia, Ciudad de México y Bolonia.

JUSTIFICACIÓN

Variable	Justificación
host_response_time	Puede indicar qué tan rápido responde un anfitrión, que tan bueno es y si es recomendado.
host_response_rate	Una tasa alta sugiere un anfitrión más comprometido, lo que podría influir en la decisión del huésped.
host_acceptance_rate	Un anfitrión que acepta más reservas puede atraer más huéspedes y mejorar la tasa de éxito del anuncio.
host_is_superhost	Los Superhosts suelen ofrecer mejor servicio, se analiza que tan probable es que un host se vuelva un Superhost
host_identity_verified	Un anfitrión verificado genera más confianza, lo que puede llevar a más reservas o mejores calificaciones.
room_type	Categorizarla entre privado(Entire) y compartido(hotel, shared y room) es fácil de trabajar.
has_availability	Desde cuanto cuesta, si esta disponible también puede influir se se puede reservar ahí.
instant_bookable	Puede que las atenciones y permisiones del host se tomen en cuenta.
host_listings_count	Un anfitrión con muchas propiedades puede tener más experiencia, lo que podría impactar la calidad del servicio y la satisfacción del huésped.
accommodates	Siendo que se puede contar como “grupo grande (mas de 2)” y “grupo chico (2 o menos)”.
beds	La cantidad de camas afecta la comodidad del alojamiento para diferentes tipos de viajeros.
price	Un precio alto (mas de $1080) o bajo(menos de $1080) puede afectar la tasa de reserva según la percepción de valor y el presupuesto del huésped.
minimum_nights	Estadías mínimas largas pueden limitar la cantidad de huéspedes interesados.
availability_365	Si es que esta disponible, tendría que saber si conviene tenerlo en cuenta en el plazo de un año.
number_of_reviews	Más reseñas pueden significar más confianza para los nuevos huéspedes, lo que aumenta la probabilidad de reserva.
review_scores_rating	Una calificación alta sugiere mejor experiencia del huésped y puede aumentar la demanda del alojamiento.
reviews_per_month	Una mayor frecuencia de reseñas indica un alojamiento con más actividad y demanda constante.

DESGLOSE

Variable independiente	Variable independiente
instant bookable	host_is_superhost, host_response_time, price
has_availability	availability_365 , minimum_nights, price
host_is_superhost	number_of_reviews, reviews_scores_rating,host_response_rate
host_identity verified	host_response_time,host_acceptance_rate, host_listings_count
host_response time	host_response_rate, host_is_superhost, number_of_reviews
review_scores rating	host_is_superhost, number_of_reviews, reviews_per_month
accommodates	room_type, price, beds
price	room_type, accommodates, review_scores_rating
room_type	accommodates, beds, price
availability 365	price, accommodates, room_type

DESARROLLO GENERAL

1.IMPORTACIÓN DE BIBLIOTECAS

- sklearn.matrics, mide que tan bien un modelo de regresión se ajusta los datos
- sklearn.model_selection: divide los datos en conjunto de entrenamiento y conjunto de prueba.
- sklearn.model_preprocessing: escala los datos para que tenga media 0 y desviación estándar 1.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.special as special
import scipy.optimize as curve_fit
import seaborn as snsfrom sklearn.metrics
import r2_scorefrom sklearn.model_selection
import train_test_splitfrom sklearn.preprocessing
import StandardScaler
```

2. CARGAR ARCHIVOS DESDE SEABORN

Según el nombre del csv que se maneje, y si es necesario eliminar columnas innecesarias, y mostrar los primeros cinco resultados del dataset.

```
#Cargamos el archivo csv
dm = pd.read_csv("Datos_Mexico.csv")
dm = dm.drop([ 'Unnamed: 0',], axis=1)
dm.head(5)
```

3. CONVERTIMOS STRING A TIPOS NUMÉRICOS PARA EL ANÁLISIS.

- Para la variable "instant_bookable" era necesario convertir las siguientes variables a números:

```
#Convertimos Los booleanos a números
dm['instant_bookable'] = dm['instant_bookable'].replace({'f': 0, 't': 1, 'False': 0})

#Convertimos Los booleanos a números
dm[ 'host_is_superhost'] = dm['host_is_superhost'].replace({'f': 0, 't': 1, 'False': 0})
```

Además, para usar la variable *"price"*, los euros los convertimos en pesos mexicanos:

valor_euro = 18.50
*dm["price_mxn"] = dm["price"] * valor_euro*

- Para poder hacer la regresión logística teníamos que convertir la variable "has_availability" en binario:

```
#Convertimos Los booleanos a números
dm['has_availability'] = dm['has_availability'].replace({'f': 0, 't': 1, 'False': 0})
```

Solo las ciudades de Bolonia y Florencia hacen la conversión de “price” antes de establecerla como una variable independiente.

- Para poder hacer la regresión logística teníamos que convertir la variable “host_is_superhost” en binario, como se mostró nteriormente
- Para usar la variable “host_response_time” se categorizó y la variable “host_is_superhost” se convirtió en binario:

```
#Categorizamos la variable ''host_response_time'', donde 1 = ''rápido'' y 0 = ''lento''
dm[''host_response_time''] = dm[''host_response_time''].replace({
'within an hour': 1,
'within a few hours': 1,
'within a day': 1,
'a few days or more': 0,
'Sin tiempo': 0,'
Desconocido': 0
})
```

- Para la variable “price” era necesario categorizarla, solo las ciudades de Bolonia y Florencia convirtieron la moneda antes de categorizar la variable:

```
#Primero convertimo los euros a pesos
valor_euro = 18.50dm[''price_mxn''] = dm[''price''] * valor_euro

#Categorizamos la variable ''price'' en 0=''barato'' y 1=''caro''
median_price = dm[''price_mxn''].median()
dm[''precio''] = (dm[''price_mxn''] > median_price).astype(int)

#Categorizamos la variable ''room type'', donde 1 = ''privado'' y 0 = ''compartido''
dm[''room_type''] = dm[''room_type''].replace({
'Entire home/apt': 1,
'Private room': 0,
'Shared room': 0,
'Hotel room': 0
})
```

- Para la variable “host_identity_verified” se necesitaba categorizar la variable independiente “host_response_time” para poder usarla
- Para las variables “room_type” y “availability_365” era necesario convertir “price” a pesos mexicanos y categorizar “room_type” para poder usarla:
- Para analizar la variable “accommodates” se categorizó, al igual que la variable independiente “room_type”, y se convirtió “price” en pesos:

```
#vamos a categorizar la variable “accommodates” en 0 = pequeño y 1 = grande
median_value = dm[”accommodates”].median()
dm[”accommodates_binary”] = (df[”accommodates”] >median_value).astype(int)
```

- Para analizar la variable “review_scores_rating” se categorizó, y también era necesario convertir la variable independiente “host_is_superhost” en binario:

```
#Categorizamos 0 "baja calificación" y 1 "alta calificación"
median_rating=dm["review_scores_rating"].median()
dm["review_rating"] =(dm["review_scores_rating"]>median_rating).astype(int)
```

4. DECLARACIÓN DE LAS VARIABLES

Tenemos 10 modelos, 1 variable dependiente por cada 3 variables independientes.

```
# Declaración de variables para 'instant_bookable'
Vars_Indep_1 = dm[['host_is_superhost', 'host_response_time', 'price_mxn']]
Var_Dep_1 = dm['instant_bookable']

# Declaración de variables para 'has_availability'
Vars_Indep_2 = dm[['availability_365', 'minimum_nights', 'price']]
Var_Dep_2 = dm['has_availability']

# Declaración de variables para 'host_is_superhost'
Vars_Indep_3 = dm[['number_of_reviews', 'review_scores_rating', 'host_response_rate']]
Var_Dep_3 = dm['host_is_superhost']

# Declaración de variables para 'host_response_time'
Vars_Indep_4 = dm[['host_is_superhost', 'host_response_rate', 'number_of_reviews']]
Var_Dep_4 = dm['host_response_time']

# Declaración de variables para 'host_identity_verified'
Vars_Indep_5 = dm[['host_response_time', 'host_acceptance_rate', 'host_listings_count']]
Var_Dep_5 = dm['host_identity_verified']

# Declaración de variables para 'availability_365'
Vars_Indep_6 = dm[['price', 'accommodates', 'room_type']]
Var_Dep_6 = dm['availability_365']

# Declaración de variables para 'room_type'
Vars_Indep_7 = dm[['price', 'accommodates', 'beds']]
Var_Dep_7 = dm['room_type']

# Declaración de variables para 'accommodates'
Vars_Indep_8 = dm['room_type', 'price', 'beds']]
Var_Dep_8 = dm['accommodates']

# Declaración de variables para 'review_scores_rating'
Vars_Indep_9 = dm[['host_is_superhost', 'number_of_reviews', 'reviews_per_month']]
Var_Dep_9 = dm['review_scores_rating']

# Declaración de variables para 'price'
Vars_Indep_10 = dm['room_type', 'accommodates', 'review_scores_rating']]
Var_Dep_10 = dm['price_mxn']
```

5. DECLARAN X Y Y

X, representa arreglo matricial por eso la mayúscula
y, representa un arreglo vectorial

```
#Dividimos el conjunto de datos en la parte de entrenamiento y prueba:
X_1 = Vars_Indep
Y_1 = Var_Dep_1
```

6. DIVIDEN EL CONJUNTO DE LOS DATOS

Le da al entrenamiento un 30% y prueba de 70%, en el test_size = 0.3 y el random_state, para fijar una semilla aleatoria, en este caso no, por eso None.

```
#Dividimos el conjunto de datos en la parte de entrenamiento y prueba:
X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(X_1, y_1, test_size=0.3, random_state=None)
```

7. ESCALAR DATOS

Estandariza los datos para que todas las variables tengas la misma escala.
fitt_transform(X_train): calcula la media y desviación estándar de X_traing y luego escala los datos.
transform(X_test): Aplica la misma transformación a X_test usando la media y desviación de X_train.

```
# Se escalan todos los datos
escalar_1 = StandardScaler()

# Realizamos el escalamiento de las variables "X" tanto de entrenamiento como de prueba
X_train_1 = escalar_1.fit_transform(X_train_1)
X_test_1 = escalar_1.transform(X_test_1)
```

8. DEFINICIÓN DEL ALGORITMO A UTILIZAR

Modelo para problemas de clasificación

```
frrom sklearn.linear_model import LogisticRegression algortimo_1 = LogisticRegression()
```

9. ENTRENA EL MODELO

Calcula los coeficientes de cada variable para minimizar la función de costo.

```
algoritmo_1.fit(X_train_1, y_train_1)
```

10. PREDICCIONES

Predicen las clases para los datos de prueba (X_text)

```
y_pred_1 = algoritmo_1.predict(X_test_1)
y_pred_1
```

11. MATRIZ DE CONFUSIÓN

Evalúa el rendimiento del modelo comparando las etiquetas reales(y_test) con las predichas (y_pred).

```
from sklearn.metrics import confusion_matrix

matrix_1 = confusion_matrix(y_test_1, y_pred_1)
print('Matriz de Confusión de "instant_bookable":')
print(matrix_1)
```

12. MÉTRICAS DE EVALUACIÓN

Se calculan métricas para determinar la eficacia del modelo.

- Precisión

La precisión es una métrica de evaluación que nos dice: qué porcentaje de las predicciones positivas del modelo son realmente correctas; es decir, calculamos la precisión del modelo comprobando los valores reales (y_test_1) con las predicciones obtenidas (y_pred_1)

```
from sklearn.metrics import accuracy_score

exactitud_1 = accuracy_score(y_test_1, y_pred_1)
print('Exactitud del modelo "instant_bookable":')
print(exactitud_1)
```

- Exactitud

La exactitud es la métrica más común para evaluar modelos de clasificación, mide el porcentaje de predicciones correctas sobre el total de ejemplos; es decir, calculamos la exactitud del modelo comparando las predicciones (y_pred_1) con los valores reales (y_test_1)

```
from sklearn.metrics import accuracy_score

exactitud_1 = accuracy_score(y_test_1, y_pred_1)
print('Exactitud del modelo "instant_bookable":')
print(exactitud_1)
```

- Sensibilidad

La sensibilidad o la tasa de verdaderos positivos, mide qué porcentaje de los casos positivos reales fueron correctamente identificados por el modelo. Si tiene una alta sensibilidad (cercana a 1), el modelo detecta la mayoría de los positivos reales, si tiene una baja sensibilidad entonces, el modelo ignora muchos positivos reales

```
from sklearn.metrics import recall_score

sensibilidad_1 = recall_score(y_test_1, y_pred_1, average="binary", pos_label=1)print('Sensibilidad del modelo
"instant_bookable":')print(sensibilidad_1)
```

RESULTADOS

CDMX

Variable dependiente	Matriz de confusión	Precisión del modelo	Exactitud del modelo	Sensibilidad del modelo	Variable dicotómica
Instant bookable	[4239 594] [2512 630]	0.514705 88235	0.6105329 1536	0.2005092 2978	1 = se puede reservar instantáneamente 0 = no se puede reservar instantáneamente
Has availability	[167 150] [352 7306]	0.9798819 7424	0.9370532 9153	0.95403499 608	1 = tiene disponibilidad 0 = no tiene
Host is superhost	[4392 675] [1494 1414]	0.676878 88942	0.728025 07836	0.4862448 4181	1 = el anfitrión es superhost 0 = el anfitrión no es superhost
Host response time	[933 472] [327 6243]	0.929709 60536	0.89981191 222	0.95022831 050	1 = el tiempo de respuesta es rápido 0 = el tiempo de respuesta es lento
Host identify verified	[13342 4470] [3505 14307]	0.761942 80236	0.7761340 6692	0.8032225 4659	t = el anfitrión está verificado f= el anfitrión no está verificado
Availability 365	[13 3036] [32 4894]	0.6171500 6305	0.6152978 0564	0.99350385 708	1 = alta disponibilidad (>180 días) 0 = baja disponibilidad (<180 días)
Room type	[1956 846] [899 4274]	0.834765 625	0.78119122 257	0.82621302 919	1 = privado, es decir Entire home/apt 0 = compartido, es decir Private room, Shared room, Hotel room

RESULTADOS

CDMX

Variable dependiente	Matriz de confusión	Precisión del modelo	Exactitud del modelo	Sensibilidad del modelo	Variable dicotómica
Accommodates	[4239 594] [2512 630]	0.88613720466	0.8510344827 5	0.7980005126 8	1 = grande, es decir, la capacidad de alojamiento es mayor a la media (+ de 2 personas)
Review scores rating	[167 150] [352 7306]	0.69982905982	0.6516614420 0	0.5186217380 2	1 = alta calificación, es decir, las calificaciones van del 1 al 5, la calificación es alta si es mayor a la media (> 4.8) 0 = baja calificación, es decir, la calificación es baja si es menor a la media (< 4.8)
Price	[4392 675] [1494 1414]	0.65604122924	0.6323510971 7	0.7525728049 0	0 = "barato", es decir, el precio es menor a la media (< $1093) 1 = "caro", es decir, el precio es mayor a la media (> $1093)

BOLONIA

RESULTADOS

Variable dependiente	Matriz de confusión	Precisión del modelo	Exactitud del modelo	Sensibilidad del modelo	Variable dicotómica
Instant bookable	[434 283] [275 461]	0.6121297602256 7	0.6159669 6490020 64	0.60529986052 99861	1 = se puede reservar instantáneamente 0 = no se puede reservar instantáneamente
Has availability	[0 6] [0 1447]	0.99655 8843771 5073	0.995870 61252580 87	0	1 = tiene disponibilidad 0 = no tiene
Host is superhost	[808 87] [297 261]	0.73122 1719457 0135	0.735719 2016517 55	0.9027932960 893855	1 = el anfitrión es superhost 0 = el anfitrión no es superhost
Host response time	[0 57] [0 1396]	0.960770 81899518 24	0.960770 818995182 4	1.0	1 = el tiempo de respuesta es rápido 0 = el tiempo de respuesta es lento
Host identify verified	[0 79] [0 1374]	0.878871 3007570 543	0.865629 731589814 2	0	t = el anfitrión está verificado f= el anfitrión no está verificado
Availabilit y 365	[848 0] [605 0]	0.583620 0963523 744	0.583620 09635237 44	1.0	1 = alta disponibilidad (>180 días) 0 = baja disponibilidad (<180 días)
Room type	[65 259] [9 1120]	0.878378 37837837 84	0.8196834 13626978 7	0.99202834366 69619	1 = privado, es decir Entire home/apt 0 = compartido, es decir Private room, Shared room, Hotel room

BOLONIA

Variable dependiente	Matriz de confusión	Precisión del modelo	Exactitud del modelo	Sensibilidad del modelo	Variable dicotómica
Accommodates	[396 112] [135 810]	0.7457627118644068	0.8300068823124 57	0.7795275590551181	1 = grande, es decir, la capacidad de alojamiento es mayor a la media (+ de 2 personas)
Revie scores rating	[599 134] [302 418]	0.664816870144284 1	0.699931176875430 2	0.8171896316507503	1 = alta calificación, es decir, las calificaciones van del 1 al 5, la calificación es alta si es mayor a la media (> 4.8) 0 = baja calificación, es decir, la calificación es baja si es menor a la media (< 4.8)
Price	[4 113] [2 1334]	0.666666666666666 6	0.920853406744662	0.03418803418803419	0 = "barato", es decir, el precio es menor a la media (< $1093) 1 = "caro", es decir, el precio es mayor a la media (> $1093)

RESULTADOS

FLORENCIA

Variable dependiente	Matriz de confusión	Precisión del modelo	Exactitud del modelo	Sensibilidad del modelo	Variable dicotómica
Instant bookable	[297 1161 [207 2152]	0.649562330214307 3	0.6416033534189154	0.9122509537939805	1 = se puede reservar instantáneamente 0 = no se puede reservar instantáneamente
Has availability	[0 7] [0 3810]	0.99816609903065 23	0.998166099030652 3	1.0	1 = tiene disponibilidad 0 = no tiene
Host is superhost	[1540 679] [397 1201]	0.638829787234042 6	0.718103222425989	0.7515644555694618	1 = el anfitrión es superhost 0 = el anfitrión no es superhost
Host response time	[65 373] [5 3374]	0.92045369629059 43	0.900969347655226 6	0.9985202722699024	1 = el tiempo de respuesta es rápido 0 = el tiempo de respuesta es lento
Host identify verified	[0 143] [0 3674]	0.965136023054855	0.962536023054755	1.0	t = el anfitrión está verificado f= el anfitrión no está verificado
Availability 365	[0 127] [0 3690]	0.966727796698978 2	0.936726796698978 2	1.0	1 = alta disponibilidad (>180 días) 0 = baja disponibilidad (<180 días)
Room type	[15 635] [6 3161]	0.832718651211801 9	0.792067068378307 6	0.998105462582886	1 = privado, es decir Entire home/apt 0 = compartido, es decir Private room, Shared room, Hotel room

FLORENCIA

Variable dependiente	Matriz de confusión	Precisión del modelo	Exactitud del modelo	Sensibilidad del modelo	Variable dicotómica
Accommodates	[1071 185] [345 2216]	0.92294877134527 28	0.86114749803510 62	0.8652869972666927	1 = grande, es decir, la capacidad de alojamiento es mayor a la media (+ de 2 personas)
Revie scores rating	[1539 422] [666 1190]	0.7382133995037 221	0.6416033534189154	0.6411637931034483	1 = alta calificación, es decir, las calificaciones van del 1 al 5, la calificación es alta si es mayor a la media (> 4.8) 0 = baja calificación, es decir, la calificación es baja si es menor a la media (< 4.8)
Price	[0 143] [0 3674]	0.9625360230547 55	0.9625360230547 55	1.0	0 = "barato", es decir, el precio es menor a la media (< $1093) 1 = "caro", es decir, el precio es mayor a la media (> $1093)

CONCLUSIÓN

Podemos demostrar la utilidad que tiene la regresión logística para predecir variables clave para nuestro análisis, se muestran grandes diferencias entre los países analizados.

Cada uno es un diferente mercado, varían desde precios hasta modo de vida, los tres son presentados desde escalas completamente diferentes, siendo que ambas ciudades italianas no tienen el aforo de lo que vendría viendo CDMX.

Por eso mismo, se nota una gran diferencia en los resultados de Bolonia y Florencia, que parecen tener resultados más exactos con respecto a los resultados mostrados de CDMX.

Las métricas que se obtuvieron proporcionan datos de utilidad para futuras investigaciones y optimizaciones, como el uso de técnica de balanceo de datos.
En resumen, este tipo de regresión ofrece datos prácticos para nuestros clientes, las industrias de este ámbito.