REGRESIÓN LOGÍSTICA

Ingeniería de Software

Romero Lagunes Jossette Yeraldin
Ruiz Santiago María Fernanda
Pestaña Marquez Valeria

ÍNDICE

REGRESIÓN LOGÍSTICA

INTRODUCCIÓN

El presente proyecto aborda el análisis de datos mediante el uso de Regresión Logística, una técnica estadística ampliamente utilizada para modelar la relación entre variables independientes y una variable dependiente dicotómica. El estudio se centra en evaluar diversas variables relacionadas con alojamientos en diferentes regiones (México, Malta, Quebec y Victoria), con el objetivo de predecir características clave como el estatus de superanfitrión, la verificación de identidad del anfitrión, la disponibilidad de reservas instantáneas, entre otras.

A lo largo del proyecto, se siguen metodologías rigurosas que incluyen la preparación de los datos, la conversión de variables a formatos dicotómicos, la división de conjuntos de entrenamiento y prueba, y la evaluación de modelos mediante métricas como precisión, exactitud y sensibilidad. Los resultados obtenidos proporcionan datos valiosos sobre el comportamiento de estas variables en cada región, lo que puede ser útil para la toma de decisiones en el sector de alojamientos turísticos.

DEFINICIÓN DE LAS VARIABLES

INDEPENDIENTES

Variable	Definición	# Caso
accommodates	Número de personas que el alojamiento puede recibir.	5, 6 y 7
availability_30	Días disponibles para reservar en los próximos 30 días.	7
availability_60	Días disponibles para reservar en los próximos 60 días.	2
availability_90	Días disponibles para reservar en los próximos 90 días.	2
availability_365	Días disponibles para reservar en los próximos 365 días.	2, 3 y 10
bedrooms	Número de habitaciones en el alojamiento.	6 y 8
calculated_host_listings_count	Número total de anuncios activos calculados para el anfitrión.	4 y 9
calculated_host_listings_count_entire_homes	Número de anuncios de viviendas completas del anfitrión.	4
calculated_host_listings_count_private_rooms	Número anuncios de habitaciones privadas del anfitrión.	5
number_of_reviews	Total, de reseñas recibidas por el anuncio.	1 y 10
number_of_reviews_ltm	Cantidad de reseñas recibidas en el último año.	1, 3 y 9
price	Precio por noche del alojamiento.	6 y 8
reviews_per_month	Promedio de reseñas recibidas por mes para el alojamiento.	9
review_scores_value	Puntuación basada en la relación calidad precio del alojamiento	7

DEFINICIÓN DE LAS VARIABLES

DEPENDIENTES

Variable	Definición	# Caso
host_is_superhost	Indica si el anfitrión tiene estatus de superanfitrión.	1
host_identity_verified	Indica si la identidad del anfitrión ha sido verificada.	2
bathrooms_text	Descripción textual sobre los baños (compartidos o privados).	3
instant_bookable	Indica si el alojamiento se puede reservar de forma instantánea.	4
room_type	Tipo de espacio ofrecido (hab. privada, hab. Compartida, ect.).	5
host_response_time	Tiempo promedio que tarda el anfitrión en responder a un mensaje	6
minimum_nights	Número mínimo de noches requeridas para reservar.	7
property_type	Tipo de propiedad (apartamento, casa, etc.)	8
host_response_rate	Porcentaje de mensajes respondidos por el anfitrión.	9
maximum_nights	Número máximo de noches permitidas para reservar.	10

DESCRIPCIÓN DE LAS VARIABLES

DICOTÓMICAS POR NATURALEZA

Variable	Etiquetas	
host_is_superhost	f: 0	t: 1
host_identity_verified	f: 0	t: 1
instant_bookable	f: 0	t: 1

CONVERTIDAS A DICOTÓMICAS

Variable	Etiquetas iniciales	Etiquetas finales
room_type	Entire home/apt Private room Shared room Hotel room	Entire No Entire
host_response_time	within an hour within a few hours within a day a few days or more Desconocido	Rapida No rapida
bathrooms_text	0-49 bath 0-49 private bath 0-49 shared bath 1.5-14.5 baths 1.5-14.5 shared baths Half-bath Private half-bath Shared half-bath	Limitados No limitados
minimum_nights	1-62	Alto Bajo
maximum_nights	1-1825	Alto Bajo
host_response_rate	-%, 0% - 100%	Respuesta alta Respuesta baja

Variable	Etiquetas iniciales	Etiquetas finales
property_type	Barn Boat Camper/RV Castle Entire bungalow Entire cabin Entire chalet Entire condo Entire cottage Entire guest suite Entire home Entire loft Entire rental unit Entire serviced apartment Entire townhouse Entire vacation home Entire villa Private room in bed and breakfast Private room in bungalow Private room in casa particular Private room in condo Private room in cottage Private room in guest suite Private room in guesthouse Private room in home Private room in hostel Private room in nature lodge Private room in rental unit Private room in serviced apartment Private room in townhouse Private room in villa Room in aparthotel Room in bed and breakfast Room in boutique hotel Room in hotel Room in serviced apartment Shared room in home Shared room in hostel Tent Tiny home	Entire PLace No Entire

PROCESO GENERAL

PASO 1

CARGA DE LAS LIBRERIAS Y EL ARCHIVO CSV

Se importan las bibliotecas necesarias (pandas, numpy, scikit-learn, etc.) y se carga el conjunto de datos desde un archivo CSV, eliminando columnas innecesarias.

```python
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
```

```python
df=pd.read_csv('Mexico_limpio.csv')
df = df.drop(['Unnamed: 0'], axis=1)
```

PASO 2

CONVERSIÓN DE LAS VARIABLES DEPENDIENTES A DICOTÓMICAS

Las variables categóricas se transforman en variables dicotómicas para que sean compatibles con el modelo de regresión logística.

- Variables como room_type (que originalmente tenía categorías como "Entire home/apt", "Private room", etc.) se simplifican a dos clases: "Entire" y "No Entire".
- Para host_response_time, las respuestas se agrupan en "Rápida" (si el anfitrión responde en menos de un día) y "No rápida" (para respuestas tardías o desconocidas).
- En bathrooms_text, se estandarizan descripciones textuales (ej. "0-49 private bath", "Half-bath") según su tipo (privado/compartido).
- Cada variable dependiente queda convertida, permitiendo al modelo interpretar relaciones entre características de los alojamientos y las variables objetivo.

```python
df['room_type']= df['room_type'].replace(["Hotel room", "Shared room", "Private room"], "No Entire")
df['host_response_time']=df['host_response_time'].replace(["within an hour", "within a few hours", "within a day"], "Rapida")
df['host_response_time']=df['host_response_time'].replace(["a few days or more", "Desconocido"], "No rapida")
entire_place_keywords = [
    'Entire', 'Boat', 'Campsite', 'Castle', 'Dome',
    'Earthen home', 'Farm stay', 'Holiday park',
    'Hut', 'Tiny home', 'Tower', 'Shipping container', 'Tent'
]
def categorize_property(property_type):
    if any(keyword in property_type for keyword in entire_place_keywords):
        return 'Entire Place'
    else:
        return 'No Entire'
df['property_type'] = df['property_type'].apply(categorize_property)
def clean_and_convert(value):
    if value == '-' or value is None:  # Tratamos -% como valor bajo
        return -1
    try:
        return float(value.replace('%', ''))
    except ValueError:
        return None

df['host_response_rate_num'] = df['host_response_rate'].apply(clean_and_convert)
def categorize_response_rate(value):
    if value < 50:
        return 'Respuesta baja'
    else:
        return 'Respuesta alta'
df['host_response_rate'] = df['host_response_rate_num'].apply(categorize_response_rate)
def categorize_minimum_nights(value):
    if value < 2:
        return 'Bajo'
    else:
        return 'Alto'
```

PASO 3

VERIFICACIÓN DE VALORES

Se revisa la integridad de los datos, asegurando que no haya valores nulos o inconsistentes que puedan afectar el modelo.

```
unico = np.unique(df['property_type'])
unico
```

```
ray(['Entire Place', 'No Entire'], dtype=object)
```

```
unico = np.unique(df['host_response_rate'])
unico
```

```
rray(['Respuesta alta', 'Respuesta baja'], dtype=object)
```

PASO 4

DEFINICIÓN DE VARIABLES INDEPENDIENTES Y DEPENDIENTES

Se seleccionan las variables predictoras (independientes) y las variables objetivo (dependientes) para el análisis.

```
Vars_Indep1 = df[['number_of_reviews', 'number_of_reviews_ltm']]
Vars_Indep2 = df[['availability_365', 'availability_90', 'availability_60']]
Vars_Indep3 = df[['availability_365', 'number_of_reviews_ltm']]
Vars_Indep4 = df[['calculated_host_listings_count', 'calculated_host_listing
Vars_Indep5 = df[['calculated_host_listings_count_private_rooms', 'accommoda
Vars_Indep6 = df[['accommodates', 'price', 'bedrooms']]
Vars_Indep7 = df[['review_scores_value', 'availability_30']]
Vars_Indep8 = df[['accommodates', 'price', 'bedrooms']]
Vars_Indep9 = df[['number_of_reviews_ltm' ,'calculated_host_listings_count',
Vars_Indep10 = df[['number_of_reviews', 'availability_365']]
```

```
Var_Dep1 = df['host_is_superhost']
Var_Dep2 = df['host_identity_verified']
Var_Dep4 = df['instant_bookable']
Var_Dep3 = df['bathrooms_text']
Var_Dep5 = df['room_type']
Var_Dep6 = df['host_response_time']
Var_Dep7 = df['minimum_nights']
Var_Dep8 = df['property_type']
Var_Dep9 = df['host_response_rate']
Var_Dep10 = df['maximum_nights']
```

PASO 5

REDEFINICIÓN DE LAS VARIABLES

Se organizan formalmente las variables independientes (predictoras) y las variables dependientes (objetivo) para cada modelo de regresión logística que se entrenará.

```
X1= Vars_Indep1
y1= Var_Dep1
X2= Vars_Indep2
y2= Var_Dep2
X3= Vars_Indep3
y3= Var_Dep3
X4= Vars_Indep4
y4= Var_Dep4
```

PASO 6

DIVISIÓN DEL CONJUNTO DE DATOS

Los datos se dividen en conjuntos de entrenamiento (70%) y prueba (30%) para evaluar el rendimiento del modelo.

```
X1_train, X1_test, y1_train, y1_test = train_test_split( (constant) X2: Series andom_state=None)
X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y2, test_size=0.3, random_state=None)
X3_train, X3_test, y3_train, y3_test = train_test_split(X3, y3, test_size=0.3, random_state=None)
X4_train, X4_test, y4_train, y4_test = train_test_split(X4, y4, test_size=0.3, random_state=None)
X5_train, X5_test, y5_train, y5_test = train_test_split(X5, y5, test_size=0.3, random_state=None)
X6_train, X6_test, y6_train, y6_test = train_test_split(X6, y6, test_size=0.3, random_state=None)
X7_train, X7_test, y7_train, y7_test = train_test_split(X7, y7, test_size=0.3, random_state=None)
X8_train, X8_test, y8_train, y8_test = train_test_split(X8, y8, test_size=0.3, random_state=None)
X9_train, X9_test, y9_train, y9_test = train_test_split(X9, y9, test_size=0.3, random_state=None)
X10_train, X10_test, y10_train, y10_test = train_test_split(X10, y10, test_size=0.3, random_state=None)
```

PASO 7

ESCALA DE DATOS

Las variables independientes se estandarizan para asegurar que todas tengan la misma escala, lo que mejora el rendimiento del modelo.

```
escalar1 = StandardScaler()
escalar2 = StandardScaler()
escalar3 = StandardScaler()
escalar4 = StandardScaler()
escalar5 = StandardScaler()
escalar6 = StandardScaler()
escalar7 = StandardScaler()
escalar8 = StandardScaler()
escalar9 = StandardScaler()
escalar10 = StandardScaler()
```

PASO 8

ESCALAMIENTO DE LAS VARIABLES "X"

Estandarizar las variables independientes (X) para que todas contribuyan equitativamente al modelo de regresión logística, eliminando sesgos por diferencias en escalas numéricas.

```
X1_train = escalar1.fit_transform(X1_train)
X1_test = escalar1.transform(X1_test)
X2_train = escalar2.fit_transform(X2_train)
X2_test = escalar2.transform(X2_test)
X3_train = escalar3.fit_transform(X3_train)
X3_test = escalar3.transform(X3_test)
X4_train = escalar4.fit_transform(X4_train)
X4_test = escalar4.transform(X4_test)
```

PASO 9

DEFINICIÓN DEL ALGORITMO AUTILIZAR

Se inicializa el modelo de regresión logística para cada variable dependiente.

```
from sklearn.linear_model import LogisticRegression
algoritmo1 = LogisticRegression()
algoritmo2 = LogisticRegression()
algoritmo3 = LogisticRegression()
algoritmo4 = LogisticRegression()
algoritmo5 = LogisticRegression()
algoritmo6 = LogisticRegression()
algoritmo7 = LogisticRegression()
algoritmo8 = LogisticRegression()
algoritmo9 = LogisticRegression()
algoritmo10 = LogisticRegression()
```

PASO 10

ENTRENAMIENTO DEL MODELO

El modelo se entrena con los datos de entrenamiento para aprender la relación entre las variables.

```
algoritmo1.fit(X1_train, y1_train)
algoritmo2.fit(X2_train, y2_train)
algoritmo3.fit(X3_train, y3_train)
algoritmo4.fit(X4_train, y4_train)
algoritmo5.fit(X5_train, y5_train)
algoritmo6.fit(X6_train, y6_train)
algoritmo7.fit(X7_train, y7_train)
algoritmo8.fit(X8_train, y8_train)
algoritmo9.fit(X9_train, y9_train)
algoritmo10.fit(X10_train, y10_train)
```

PASO 11

PREDICCIÓN

Se realizan predicciones utilizando el conjunto de prueba para evaluar el rendimiento del modelo.

```
y1_pred = algoritmo1.predict(X1_test)
y1_pred
```

PASO 12

VERIFICACIÓN DE LA MATRIZ DE CONFUSIÓN

Se genera una matriz de confusión para visualizar los aciertos y errores del modelo.

```
from sklearn.metrics import confusion_matrix
matriz1 = confusion_matrix(y1_test, y1_pred)
print('Matriz de Confusión: ')
print (matriz1)
```

PASO 13

MÉTRICAS DE EVALUACIÓN

Se calculan métricas como precisión, exactitud y sensibilidad para determinar la eficacia del modelo.

- Precisión

```
from sklearn.metrics import precision_score
precision1 = precision_score(y1_test, y1_pred, average="binary", pos_label="t")
print('Precision del modelo:')
print(precision1)
```

- Exactitud

```
from sklearn.metrics import accuracy_score
exactitud1 = accuracy_score(y1_test, y1_pred)
print('Exactitud del modelo:')
print(exactitud1)
```

- Sensibilidad

```
from sklearn.metrics import recall_score
sensibilidad1 = recall_score(y1_test, y1_pred, average="binary", pos_label="f")
print('Sensibilidad del modelo:')
print(sensibilidad1)
```

RESULTADOS DE MÉXICO

Variables	Matriz de Confusión	Precisión de Modelo	Exactitud del Modelo	Sensibilidad del Modelo
host_is_superhost	[[4184 715] [1809 1267]]	0.6392	0.6835	0.8540
host_identity_verified	[[0 329] [0 7646]]	0.9587	0.9587	0.0
bathrooms_text	[[0 1170] [0 6805]]	0.0	0.8532	1.0
instant_bookable	[[4281 567] [2250 877]]	0.6392	0.6467	0.8830
room_type	[[4833 366] [765 2011]]	0.8633	0.8581	0.7244
host_response_time	[[6 1439] [0 6530]]	0.8194	0.8195	0.0041
minimum_night	[[315 2811] [361 4488]]	0.4659	0.6022	0.9255
property_type	[[5184 0] [2791 0]]	0.6500	0.6500	0.0
host_response_rate	[[7659 0] [316 0]]	0.9603	0.9603	0.0
maximum_night	[[5920 14] [2022 19]]	0.7454	0.7447	0.0093

RESULTADOS DE MALTA

Variables	Matriz de Confusión	Precisión de Modelo	Exactitud del Modelo	Sensibilidad del Modelo
host_is_superhost	[[2476 207] [866 234]]	0.5306	0.7163	0.9228
host_identity_verified	[[0 42] [0 3741]]	0.9888	0.9888	0.0
bathrooms_text	[[0 390] [0 3393]]	0.0	0.8969	1.0
instant_bookable	[[783 840] [429 1731]]	.5306	0.6645	0.4824
room_type	[[2529 155] [138 961]]	0.9482	0.3225	0.8744
host_response_time	[[0 495] [0 3288]]	0.8691	0.8691	0.0
minimum_night	[[2406 23] [1336 18]]	0.6429	0.6407	0.0132
property_type	[[2677 0] [1106 0]]	0.7076	0.7076	0.0
host_response_rate	[[3681 0] [102 0]]	0.9730	0.9730	0.0
maximum_night	[[2849 5] [928 1]]	0.7543	0.7533	0.0010

RESULTADOS DE QUEBEC

Variables	Matriz de Confusión	Precisión de Modelo	Exactitud del Modelo	Sensibilidad del Modelo
host_is_superhost	[[253 102] [164 177]]	0.6344	0.6178	0.7126
host_identity_verified	[[0 31] [0 665]]	0.9554	0.9554	0.0
bathrooms_text	[[0 44] [0 652]]	0.0	0.9367	1.0
instant_bookable	[[320 76] [196 104]]	0.6344	0.6091	0.8080
room_type	[[597 0] [99 0]]	0.8577	0.8577	0.0
host_response_time	[[0 87] [0 609]]	0.875	0.875	0.0
minimum_night	[[494 0] [202 0]]	0.7097	0.7097	0.0
property_type	[[595 0] [101 0]]	0.8548	0.8548	0.0
host_response_rate	[[692 0] [4 0]]	0.9942	0.9942	0.0
maximum_night	[[523 6] [167 0]]	0.7579	0.7514	0.0

RESULTADOS DE VICTORIA

Variables	Matriz de Confusión	Precisión de Modelo	Exactitud del Modelo	Sensibilidad del Modelo
host_is_superhost	[[429 127] [196 439]]	0.7756	0.7287	0.7715
host_identity_verified	[[0 80] [0 1111]]	0.9328	0.9328	0.0
bathrooms_text	[[0 71] [0 1120]]	0.0	0.9403	1.0
instant_bookable	[[902 8] [265 16]]	0.7756	0.7707	0.9912
room_type	[[1006 0] [185 0]]	0.8446	0.8446	0.0
host_response_time	[[0 233] [0 958]]	0.8043	0.8043	0.0
minimum_night	[[871 0] [320 0]]	0.7313	0.7313	0.0
property_type	[[991 0] [200 0]]	0.8320	0.8320	0.0
host_response_rate	[[1166 0] [25 0]]	0.9790	0.9790	0.0
maximum_night	[[456 189] [315 231]]	0.5914	0.5768	0.4230

CONCLUSIÓN

El proyecto demuestra la utilidad de la Regresión Logística para predecir variables clave en el contexto de alojamientos turísticos, destacando diferencias significativas entre regiones como México, Malta, Quebec y Victoria. Los resultados revelan que variables como host_identity_verified y host_response_rate presentan altos niveles de precisión en todas las regiones, mientras que otras, como bathrooms_text, muestran limitaciones debido a la naturaleza de los datos.

Las métricas obtenidas proporcionan una base sólida para futuras investigaciones y optimizaciones, como la inclusión de más variables predictoras o el uso de técnicas de balanceo de datos para mejorar la sensibilidad en casos desequilibrados. En conclusión, este estudio no solo valida la aplicabilidad de la regresión logística en el análisis de alojamientos, sino que también ofrece datos prácticos para la industria turística y plataformas de hospedaje.