

DEMANDA EN HOTELES DE BOOKING

Ramiro Monroy Ramos
Estudiante de Ingeniería de Sistemas
Universidad de Antioquia

INTRODUCCIÓN

A lo largo de este documento se realiza un análisis y exploración de una base de datos con el fin de depurarla y prepararla lo mejor posible para así lograr predecir el mejor modelo para la variable de salida

I. DESCRIPCIÓN DEL PROBLEMA.

El aprendizaje de máquina se ha convertido en una herramienta esencial para todas las industrias y la hotelera no es la excepción; ya que permite predecir con mayor precisión la probabilidad de cancelación de reservas. En este artículo, se estudiará una base de datos de reservas de hotel y se aplicarán técnicas de aprendizaje de máquina para predecir qué reservas son más propensas a ser canceladas.

La base de datos utilizada en este estudio contiene información sobre las reservas de dos hoteles, en la que se plantea abordar un problema de clasificación en el cual requiere seleccionar las reservas en las que se han realizado cancelaciones, y las reservas que no han sido canceladas, con el fin de predecir cuáles reservas son las más opcionadas a ser canceladas. Entre las variables que detalla esta base de datos encontramos detalles sobre la fecha de llegada y días de permanencia, número de adultos y niños, tipos de comida, entre otros factores.

El objetivo final de este estudio es ayudar a los hoteles a optimizar su gestión de reservas y a mejorar su rentabilidad mediante la identificación temprana de reservas que tienen una alta probabilidad de ser canceladas. Espero que este estudio sea de gran utilidad para la industria hotelera y para cualquier persona interesada en el uso del aprendizaje de máquina para la toma de decisiones empresariales.

En la base de datos Hotel Booking Demanda cuenta con 119391 muestras y 32 características y contiene datos tomados el artículo Hotel Booking Demand Datasets [1]. Entre las características de la base de datos se optó como variable de salida 'is_canceled' la cual hace referencia si la reserva realizada en un hotel ha sido cancelada o no.

A. Variables de la base de datos:

- 'hotel': indica a qué hotel pertenece la reserva.
- 'is_canceled': valor que indica si una reserva fue cancelada (1) o no (0).
- 'lead_time': número de días que hay entre la fecha de la reserva y la fecha revista de entrada.
- 'arrival_date_year': año de la fecha de llegada.
- 'arrival_date_month': mes de la fecha de llegada.
- 'arrival_date_week_number': número de la semana de la fecha de llegada.
- 'arrival_date_day_of_month': día del mes de la fecha de llegada.
- 'stays_in_weekend_nights': número de noches en fin de semana (sábado y domingo) que se han reservado.
- 'stays_in_week_nights': número de noches reservadas entre semana.
- 'adults': número de adultos.
 - 'children': número de niños.
 - 'babies': número de bebés.
- 'meal': tipo de comida reservada. (Undefined/SC, BB – Cama y desayuno, HB – desayuno y otra comida, FB – desayuno, almuerzo y cena).

- *'country'*: indica el país de origen de los clientes. las categorías representadas en el formato ISO 3155-3:2013.
- *'market_segment'*: marca la designación del mercado. (TA – Agente de viajes, TO – Operador de tour).
- *'distribution_channel'*: indica cuál fue el canal de distribución. Es una variable categórica. (TA – Agente de viajes, TO – Operador de tour).
- *'is_repeated_guest'*: valor que indica si el nombre de la reserva forma parte de un cliente repetido (1) o no (0).
- *'previous_cancellations'*: número de reservas previas que fueron canceladas por el cliente que está reservando.
- *'previous_bookings_not_cancelled'*: número de reservas previas que no fueron canceladas por el cliente que está reservando.
- *'reserved_room_type'*: código del tipo de habitación que fue reservada. (AG).
- *'assigned_room_type'*: código del tipo de habitación que se asigna a la reserva.
- *'booking_changes'*: número de cambios hechos en la reserva desde que reservó hasta que se hizo el check-in o la cancelación.
- *'deposit_type'*: indicador de si el cliente dejó un depósito para garantizar la reserva. (No deposit – Ningún depósito fue realizado, Non Refund – Un depósito fue hecho en el valor del costo total de la estadía, Refundable – un depósito fue hecho con un valor menor al valor total de la estadía).
- *'agent'*: identificador que muestra que agencia de viajes ha realizado la reserva.
- *'company'*: identificador de la compañía que hizo la reserva o que hizo el pago.
- *'days_in_waiting_list'*: número de días que la reserva estuvo en la lista de espera antes de ser confirmada por el cliente.
- *'customer_type'*: el tipo de cliente. Puede ser de 4 tipos diferentes: Contract – cuando la reserva tiene algún tipo de contrato asociado, Group – cuando la reserva está asociada a un grupo, Transient – cuando la reserva no está asociada a un grupo o contrato, Transient-party – cuando la reserva es transitoria.
- *'adr'*: media del precio pagado por noche.
- *'required_car_parking_spaces'*: número de plazas de parking requeridas por el cliente.
- *'total_of_special_request'*: número de requisitos pedidos por el cliente.
- *'reservation_status'*: se muestra el estado de la reserva. Puede ser: Canceled – la reserva fue cancelada, Check-out – el cliente se registró, pero también partió, No-Show – el cliente no se registró e informó al hotel la razón.
 - *'reservation_status_date'*: fecha en la que la última modificación de la *'reservation_status'* fue hecha.

II. ANTECEDENTES.

En esta sección se hablará de un artículo relacionado con la base de datos o modelos que se usarán. El artículo "Modelo dinámico de predicción y ajuste de los precios de las habitaciones de hoteles". utiliza una técnica de aprendizaje supervisado llamada Random Forest para predecir los precios de las habitaciones de hoteles. La metodología de validación utilizada fue la validación cruzada, que se realiza dividiendo el conjunto de datos en varias partes y probando el modelo en cada parte, para evaluar su capacidad de generalización.

En cuanto a los resultados, el modelo propuesto en el artículo demostró ser capaz de predecir los precios de las habitaciones de los hoteles con una precisión del 90%. Además, se encontró que las variables más influyentes en la predicción del precio de la habitación son la ubicación, la temporada y la categoría del hotel. El artículo también propone una estrategia de ajuste de precios para ayudar a los hoteles a maximizar sus ingresos.

III. EXPERIMENTO.

La base de datos a estudiar contiene información de dos hoteles el cual uno está en la capital portuguesa y el otro es un complejo turístico, entre las características que incluye están el momento en que la reserva fue hecha, tiempo de la estadía, número de las personas a hospedarse como niños, adultos, y bebés, número de parqueos disponibles fechas de llegada e ida entre otras. la base de datos cuenta con 1193191 muestras.

Como metodología de este proyecto se utilizó para ejecutar el dataset es Jupiterlab de Anaconda, para el lenguaje Python, implementando diferente librerías como Numpy, Pandas para el procesamiento de datos; Scikit-learn para realizar el aprendizaje supervisado y no supervisado; para la visualización de los datos Matplotlib y Seaborn.

A. Preparación y limpieza de datos

En la preparación de los datos se crea un pipeline, y se limpia country por fuera del pipeline para que no quede agregado en el dataset final como categórico; se eliminan las columnas que no tienen sentido para el entrenamiento, además la característica se elimina porque solo existen 6797 valores del total (119390) lo que corresponde a un porcentaje de faltante del 94.3%, la característica “reservation_status” se elimina porque es una variable de salida de igual manera con la característica reservation_status_date se elimina porque es una fecha.

B. Instalación e importación de bibliotecas y paquetes.

Dentro de las librerías importadas encontramos las de análisis de datos, las de visualización de los datos y las librerías de selección de modelos.

```
# Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder, LabelEncoder
from sklearn import svm
from sklearn.model_selection import KFold, StratifiedKFold
from sklearn.preprocessing import MinMaxScaler
```

Figura 1. Importación de librerías.

C. Cargar la base de datos.

En cuanto a la lectura de datos se carga la base de datos de tipo .csv en la variable “db” la cual muestra los datos como forma de verificar que la base de datos carga correctamente.

```
# Cargar la base de datos
db = pd.read_csv('hotel_bookings.csv')
db.info()
```

Figura 2. Carga de la base de datos.

con db.info() nos podemos verificar el tamaño de la base de datos en la que se observan que se tienen 119390 muestras y 32 características, además de las etiquetas de cada columna con su respectiva información.

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   hotel                                     119390 non-null  object
1   is_cancelled                             119390 non-null  int64
2   lead_time                                119390 non-null  int64
3   arrival_date_year                        119390 non-null  int64
4   arrival_date_month                      119390 non-null  object
5   arrival_date_week_number                119390 non-null  int64
6   arrival_date_day_of_month               119390 non-null  int64
7   stays_in_weekend_nights                 119390 non-null  int64
8   stays_in_week_nights                   119390 non-null  int64
9   adults                                  119390 non-null  int64
10  children                                119386 non-null  float64
11  babies                                  119390 non-null  int64
12  meal                                    119390 non-null  object
13  country                                 118902 non-null  object
14  market_segment                          119390 non-null  object
15  distribution_channel                    119390 non-null  object
16  is_repeated_guest                       119390 non-null  int64
17  previous_cancellations                  119390 non-null  int64
18  previous_bookings_not_cancelled         119390 non-null  int64
19  reserved_room_type                      119390 non-null  object
20  assigned_room_type                      119390 non-null  object
21  booking_changes                         119390 non-null  int64
22  deposit_type                            119390 non-null  object
23  agent                                   103050 non-null  float64
24  company                                 6797 non-null   float64
25  days_in_waiting_list                   119390 non-null  int64
26  customer_type                           119390 non-null  object
27  adr                                     119390 non-null  float64
28  required_car_parking_spaces             119390 non-null  int64
29  total_of_special_requests               119390 non-null  int64
30  reservation_status                     119390 non-null  object
31  reservation_status_date                 119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB

```

Figura 3. Tipos de datos de cada etiqueta.

D. Análisis exploratorio de datos



Figura 4. Diagrama de barras respecto al recuento de reservas canceladas y no canceladas.

De acuerdo con el análisis realizado se puede observar que El 37% de las muestras corresponden a la clase Cancelada - 1 y el 63% a No Cancelada - 0. La base de datos se puede considerar levemente desbalanceada por lo que se recomienda utilizar el método de validación StratifiedKfold.

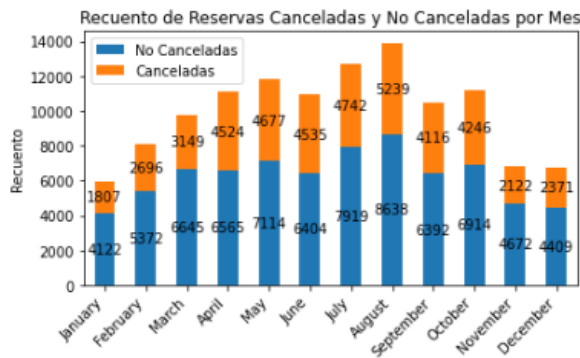


Figura 4. Diagrama de barras respecto al recuento de reservas canceladas y no canceladas por mes.

Se puede observar que la proporción de reservaciones canceladas y no en cada mes es similar. El mes en el que más se realizan viajes es agosto, julio y mayo.

III. MODELOS DE PREDICCIÓN.

Antes de iniciar con cada uno de los modelos se hace la preparación de datos utilizando pipelines y transformadores en el contexto de entrenamiento de un modelo de aprendizaje automático. A continuación, se describen las principales funciones definidas en el código: `procesar_con_onehotencoder(db)`: Esta función realiza la transformación de los datos utilizando `OneHotEncoder` para las variables categóricas y `OrdinalEncoder` para el mes de llegada. También se utiliza `SimpleImputer` para rellenar los valores nulos en las características "children" y "agent" con ceros. Luego, se aplica el preprocesamiento utilizando un `ColumnTransformer` y un `Pipeline`. El resultado transformado se devuelve.

`procesar_con_labelencoder(db)`: Esta función es similar a la anterior, pero en lugar de usar `OneHotEncoder`, se utiliza `LabelEncoder` para las variables categóricas. El resto del preprocesamiento es el mismo que en `procesar_con_onehotencoder(db)`.

`procesar_con_ordinalencoder_estandarizado(db)`: Esta función realiza el mismo preprocesamiento que `procesar_con_labelencoder(db)`, pero además aplica una estandarización a los datos utilizando `MinMaxScaler`.

Después se aplica una de las funciones de preprocesamiento (en este caso, '`procesar_con_ordinalencoder_estandarizado`') al conjunto de datos '`datosX`' para obtener los datos preprocesados '`X`' y el vector objetivo '`Y`'. Y se define una función '`classification_error`' para calcular el error de clasificación entre las etiquetas predichas y las etiquetas reales.