Organización de Datos (75.06)
Trabajo Práctico 1: Reservas de Hotel. Checkpoint 2.
Cuatrimestre, año y grupo: 2ºC 2023
Grupo: 11 - Los Pandas
Corrector: Mateo
Fecha de entrega: 5 de octubre de 2023
Datos de los integrantes:

Labollita, Francisco	103456	flabollita@fi.uba.ar
Mundani Vegega, Ezequiel	102312	emundani@fi.uba.ar
Otegui, Matías Iñaki	97263	motegui@fi.uba.ar

Introducción

Para este checkpoint se implementó el dataset detallado en el checkpoint anterior. A modo de recordatorio, se creó una columna que indica si la habitación solicitada coincide con la asignada, también se eliminaron las filas con datos faltantes en *country*, *market_segment* y *distribution_channel*. Las siguientes columnas fueron transformadas a variables booleanas (0 si no tenían ningún valor, 1 si su valor era mayor a 0): *required_car_parking_spaces*, *days_in_waiting_list*, *babies*, *previous_cancelations*, *special_requests previous_bookings_not_canceled*, *booking_changes*.

Una vez modificado el dataset original, para poder entrenar el árbol de decisión, se crearon dummies para las variables cualitativas: *hotel*, *arrival_date_month*, *meal*, *country*, *market_segment*, *distribution_channel*, *assigned_room_type*, *deposit_type*, *customer_type* y *agent*.

Con hacer dummies se refiere a crear una nueva columna por cada posible valor de esa categoría. El dataframe utilizado para entrenar al modelo cuenta con 499 columnas.

Cabe aclarar que al dataframe construido a partir del .csv de test se lo modificó para tener las mismas 499 columnas anteriores y en el mismo orden.

Construcción del modelo

Optimización de hiperparámetros

Se optimizó la profundidad, el criterio, el alfa, la cantidad mínima de muestras para hacer un split y la mínima cantidad de muestras para que sea una hoja. Se optó por optimizar parámetro por parámetro y luego se corroboraron estos resultados al hacer un random search. No se decidió comparar todos los parámetros contra todos debido al alto costo computacional que hubiese requerido, ni tampoco probar valores al azar. El costo de realizar una comparación todos contra todos sería de $O(n^4)$, siendo n la cantidad de posibles parámetros que se tomarían por cada hiperparámetro.

Primero se probaron 5 valores (siempre equiespaciados) entre 1 y 25 para la profundidad, se obtuvo que la ideal era 16. Luego se volvió a realizar la búsqueda entre 12 y 20 tomando 4 valores, y por último de 15 a 18, obteniéndose como mejor aproximación una profundidad de 16. Este método, similar a una búsqueda binaria, fue repetido para los otros, excepto para el alfa, donde se realizó una búsqueda donde los posibles valores estaban equiespaciados logarítmicamente, debido a la naturaleza del mismo.

Finalmente, se concluyó que los hiperparámetros que daban una mejor predicción eran alfa=$3,80 \times 10^{-5}$, criterio=Gini, profundidad máxima = 16, mínimo de muestras para ser hoja = 1 y mínimo de muestras para hacer split = 2.

Cabe aclarar que se utilizó siempre la misma semilla al momento de realizar los folds, para obtener siempre los mismos resultados al correr dos veces la misma predicción.

K-fold Cross Validation

Se utilizó el método de K-fold Cross validation. Se probaron distintas cantidades de folds y se obtuvo que la que mejor predecía era con 16 folds.

Métrica utilizado

Para ver qué combinación de hiperparámetros predecía mejor, se utilizó el F1 Score, debido a que como se trataba de un dataset bastante equilibrado se pensó que sería la más significativa. Otra razón fue que este era el parámetro utilizado para calificar al modelo en la competencia de Kaggle.

Mejora desde el modelo inicial al final

Antes de optimizar los hiperparámetros el F1 score era 0,801. Al haber optimizado la cantidad de folds y los hiperparámetros se llegó a un F1 score de 0,834.

Visualizaciones

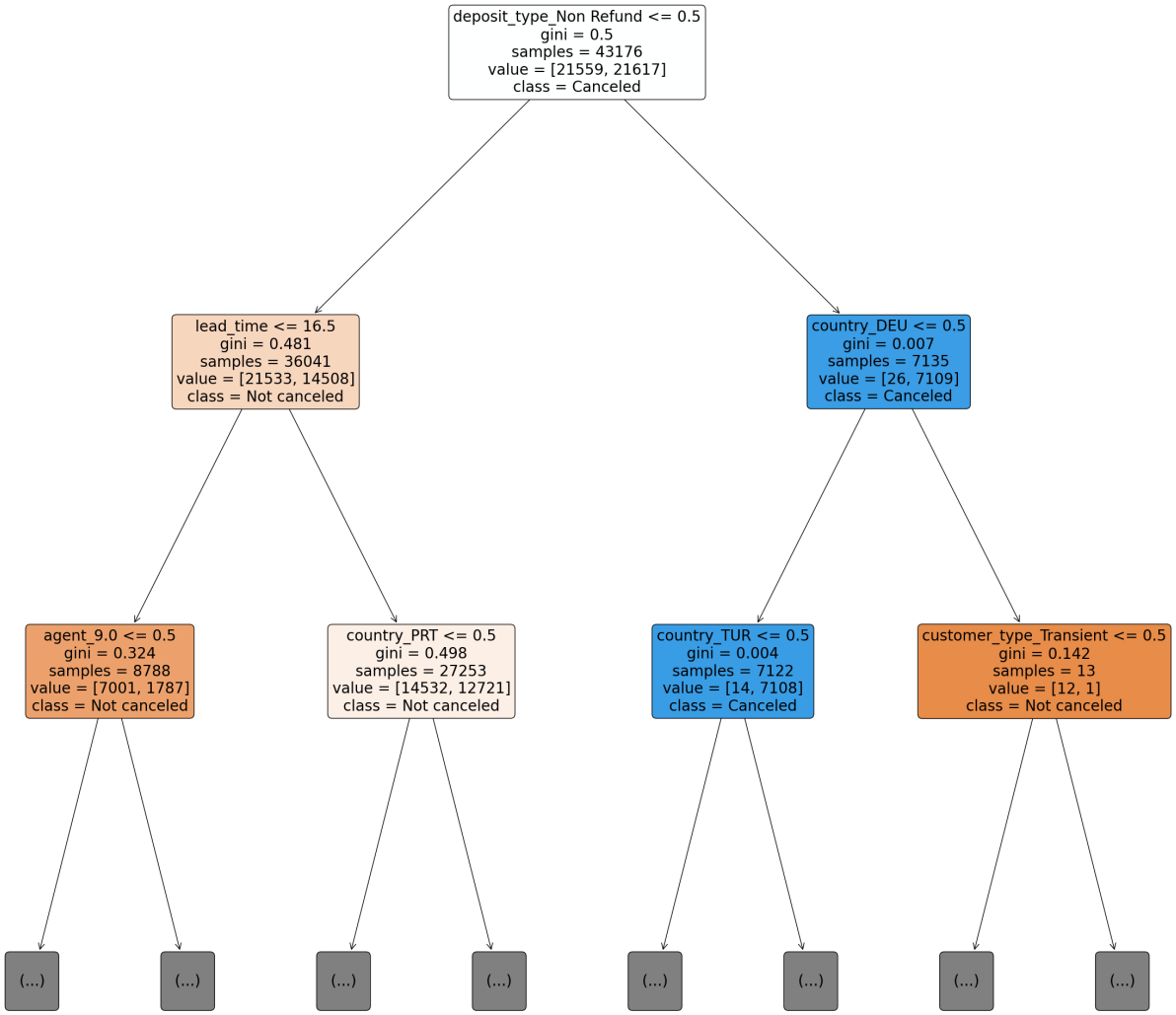

Figura 1: Mejor árbol de decisión obtenido.

Analizando el gráfico del árbol, se ve que el nodo más importante es *deposit_type_Non Refund*, que es uno de los dummies generados para la variable *deposit_type*. Esto se corresponde con lo visto en el checkpoint 1 dado a que la gran mayoría que tenía este tipo de depósito tendía a cancelar.

Se ve que primero se analiza si el depósito es de tipo *Non refund*. En caso de serlo, se fija si la persona viene del país con código DEU. Si lo es se fija si el tipo de cliente es *Transient*, si no, se fija si el código de país es TUR.

Por otro lado, si el tipo de depósito no era *Non_refund*, se fija con cuánta anticipación se hizo la reserva. En caso de haberse hecho con más de 16,5 días de anticipación se analiza si proviene del país con código PRT, si no verifica si la reserva fue hecha por el agente número 9.

El agente 9 fue el que más reservas manejó de los que no tenían ese campo indefinido, por lo que tiene sentido que aparezca entre los primeros nodos. Para el código de país PRT (Portugal) y DEU (Alemania) pasa algo parecido, solo que estos valores tenían más reservas que los que tenían código de país indefinido. El tipo de cliente *Transient* es el que más aparecía en las reservas.

Cabe aclarar que para las variables booleanas, se tomaron solo como valores 1 y 0 y es por eso que el umbral de varios nodos es 0,5.

Resultados

Modelo	F1 Score	Precision	Recall	Accuracy	Kaggle
Mejor GridSearch	0,842	0,813	0,873	0,836	0,834
Mejor Random	0.842	0.813	0.873	0.836	-
Sin optimizar	0.821	0.851	0.793	0.827	0,801

Cuadro 1: Tabla de resultados

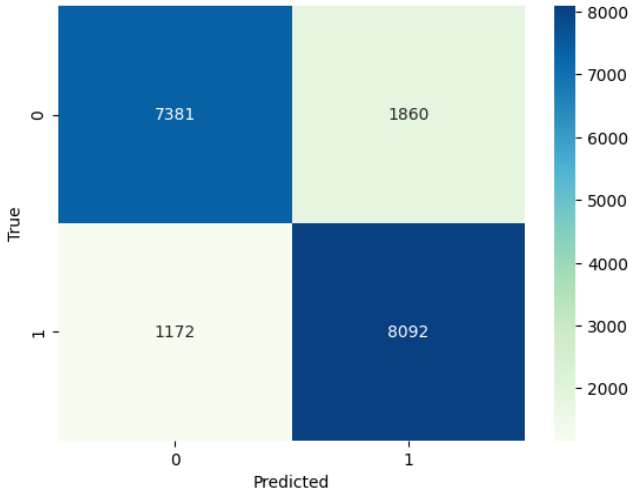

Figura 2: Matriz de confusión del mejor modelo obtenido.

En la siguiente matriz, se puede observar que de todos los valores predichos, 8.092 fueron reservas predichas como canceladas que efectivamente lo fueron, 1.860 fueron falsos positivos, 1.172 fueron falsos negativos y 7.381 fueron predichas como no canceladas y efectivamente lo fueron. Analizando brevemente el heatmap parecería que el modelo entrenado realmente ayuda a predecir si una reserva será o no cancelada.

Distribución de tareas

Dado a que se utilizó la técnica de desarrollo pair-programming y acordamos evitar trabajar solos, todos los integrantes tuvieron participación en todas las partes del checkpoint.

Integrante	Tareas
Labollita, Francisco	Todo
Mundani Vegega, Ezequiel	Todo
Otegui, Matías Iñaki	Todo