Informe TP1: Reservas de Hotel

Integrantes:
Agama Avila, Arely - 105829
Martinez, Selene Anahi - 100439
Meichtri, Melany - 102330

Checkpoint 2:

A. .

Construir árboles de decisión y optimizar sus hiperparámetros mediante k-fold
Cross Validation para obtener la mejor performance. ¿Cuántos folds utilizaron?
¿Qué métrica consideran adecuada para buscar los parámetros?

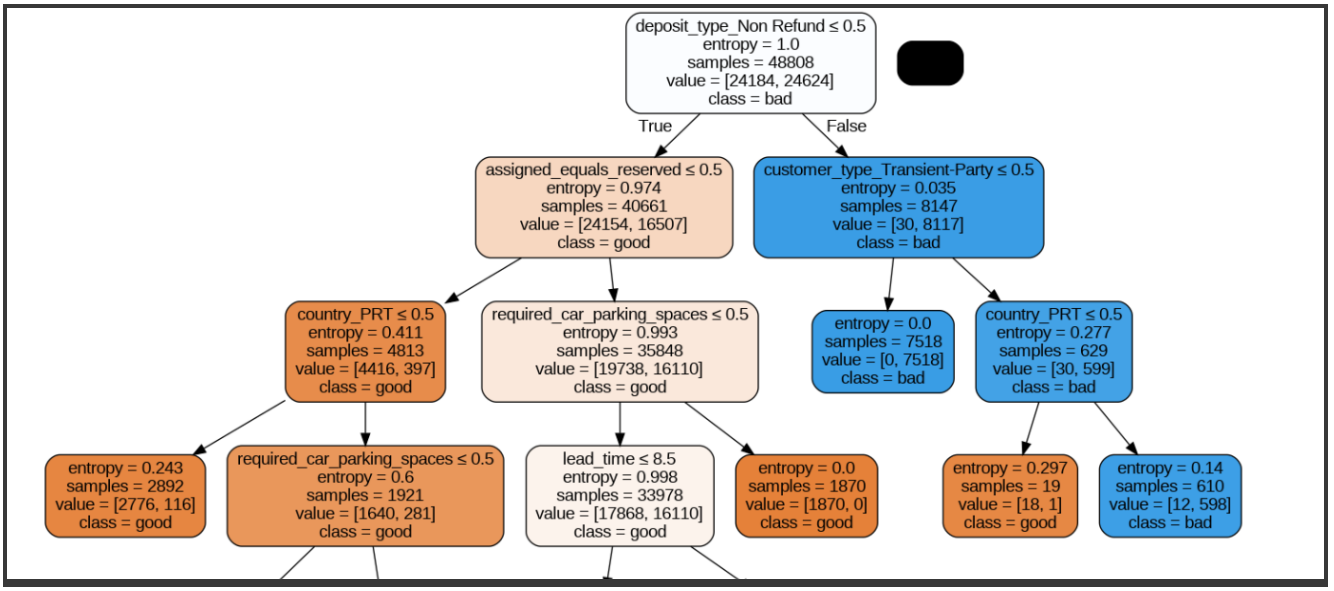
Primero decidimos dividir en 80/20 el dataset.Utilizamos el 80% para , es decir 48808 de 61010 filas de nuestro dataset para entrenar y el 20% (12202) para testear.
Con Cross validation lo que hacemos es dividir nuestros datos de entrenamiento para entrenar y para validar. Lo bueno de esto es que se entrena en datos distintos buscando así el mejor modelo.

...

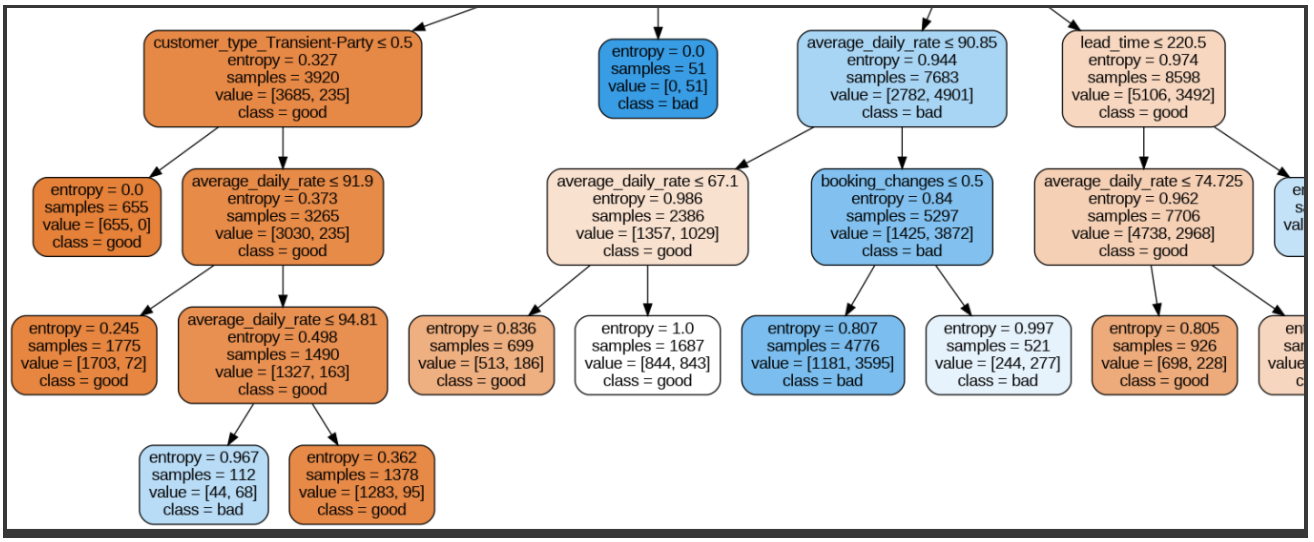
En un principio lo separamos en 20 folds y luego subimos a 45/40 . Elegimos f1 score ya que es una métrica que tiene en cuenta tanto el recall como la precisión, descartamos el accuracy ya que si el árbol está desbalanceado nos puede arrojar un buen accuracy pero el árbol no sirve para hacer una buena predicción

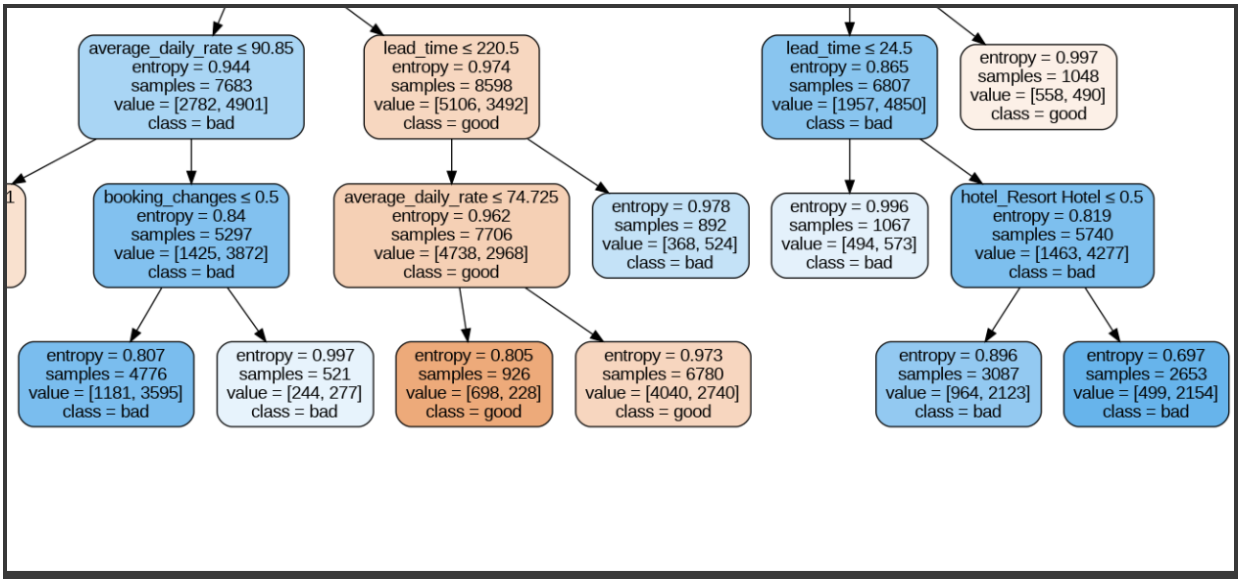
B. Graficar el árbol de decisión con mejor performance

Raíz y hoja derecha



Hojas





C. Analizar el árbol de decisión seleccionado describiendo los atributos elegidos, y
decisiones evaluadas (explicar las primeras reglas obtenidas).

ÁRBOL 1

Elegimos las variables que estaban más relacionadas con el target en nuestro heatmap del CH1.

Transformamos las siguientes variables categóricas para poder utilizarlas en el árbol utilizando one-hot-encoding:

a. meal
b. hotel

```
features1= ['hotel_Resort
Hotel','meal_FB','meal_HB','meal_SC','lead_time', 'adults',
'children', 'babies',
'previous_cancellations','days_in_waiting_list']
```

Como eran 10 variables, decidimos que nuestro árbol (1, 8) de profundidad .
Al tener tantas observaciones, decidimos aumentar nuestra cantidad de folds, pero no sabíamos cuánto, así que probamos con 20.
Del primer árbol obtuvimos que las features más representativas para clasificar eran lead_time y previous_cancellations. Las reglas para este árbol fueron lead_time <= 17.5 y después clasificó por previous_cancellations >0.5.
Nuestro mejor parámetro obtenido solo tuvo profundidad de 2, por lo que, en los siguientes árboles, decidimos cambiar el mínimo para que sea un mejor clasificador.

ARBOL 2

Decidimos utilizar únicamente variables cuantitativas para comenzar a elegir cuales brindan una mejor predicción.

```
features2=
['lead_time','previous_cancellations','booking_changes','previous_bookings_not_canceled', 'stays_in_week_nights','days_in_waiting_list']
```

Estabamos priorizando encontrar los mejores features, por lo que vimos que en el árbol N2 nos dio lead_time, previous_cancellations, booking_changes y prevous_booking_not_canceled. Estas 2 últimas, decidimos agregarlas al siguiente árbol por ser consideradas buenas features para predecir.

Continuamos intercambiando variables cuantitativas hasta que llegamos a la conclusión que debíamos hacer clustering con algunas variables, como por ejemplo, country dado que en CH1 notamos que Portugal, principalmente entre otros países, tenía mayor relación con el target.

También hicimos clustering de assigned_room_type y reserved_room_type. Tomamos la decisión de dividirlos en 6 clusters. Luego creamos una variable llamada assigned_equals_reserved la cual nos indica si coincide el tipo de habitación.

Luego aplicamos one hot encoding en esas variables cualitativas, también en hotel, meal ,deposit_type, customer_type y country.

ÁRBOL 9

En el árbol 9 pudimos confirmar que las variables cualitativas mejoran mucho la predicción .

```
[279] features9= ['deposit_type_Non Refund','lead_time','average_daily_rate','assigned_equals_reserved','previous_cancellations',
      'required_car_parking_spaces','booking_changes','stays_in_week_nights','adults','hotel_Resort Hotel','total_of_special_requests',
      'customer_type_Group', 'customer_type_Transient','customer_type_Transient-Party','country_PRT', 'country_GBR',
      'country_FRA','country_ESP','country_DEU']
```

Reglas

```
|--- deposit_type_Non Refund <= 0.50
|   |--- assigned_equals_reserved <= 0.50
|   |   |--- country_PRT <= 0.50
|   |   |   |--- class: 0
|   |   |--- country_PRT >  0.50
|   |   |   |--- required_car_parking_spaces <= 0.50
|   |   |   |   |--- class: 0
|   |   |   |--- required_car_parking_spaces >  0.50
```

```
|   |   |   |   |--- class: 0
|   |--- assigned_equals_reserved >  0.50
|   |   |--- required_car_parking_spaces <= 0.50
|   |   |   |--- lead_time <= 8.50
|   |   |   |   |--- country_PRT <= 0.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- country_PRT >  0.50
|   |   |   |   |   |--- class: 0
|   |   |   |--- lead_time >  8.50
|   |   |   |   |--- country_PRT <= 0.50
|   |   |   |   |   |--- customer_type_Transient <= 0.50
|   |   |   |   |   |   |--- previous_cancellations <= 0.50
|   |   |   |   |   |   |   |--- customer_type_Transient-Party <= 0.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- customer_type_Transient-Party >  0.50
|   |   |   |   |   |   |   |   |--- average_daily_rate <= 91.90
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- average_daily_rate >  91.90
|   |   |   |   |   |   |   |   |   |--- average_daily_rate <= 94.81
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- average_daily_rate >  94.81
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- previous_cancellations >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- customer_type_Transient >  0.50
|   |   |   |   |   |   |--- total_of_special_requests <= 0.50
|   |   |   |   |   |   |   |--- average_daily_rate <= 90.85
|   |   |   |   |   |   |   |   |--- average_daily_rate <= 67.10
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- average_daily_rate >  67.10
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- average_daily_rate >  90.85
|   |   |   |   |   |   |   |   |--- booking_changes <= 0.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- booking_changes >  0.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- total_of_special_requests >  0.50
|   |   |   |   |   |   |   |--- lead_time <= 220.50
|   |   |   |   |   |   |   |   |--- average_daily_rate <= 74.72
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- average_daily_rate >  74.72
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- lead_time >  220.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- country_PRT >  0.50
|   |   |   |   |   |--- previous_cancellations <= 0.50
|   |   |   |   |   |   |--- booking_changes <= 0.50
|   |   |   |   |   |   |   |--- lead_time <= 24.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- lead_time >  24.50
|   |   |   |   |   |   |   |   |--- hotel_Resort Hotel <= 0.50
```

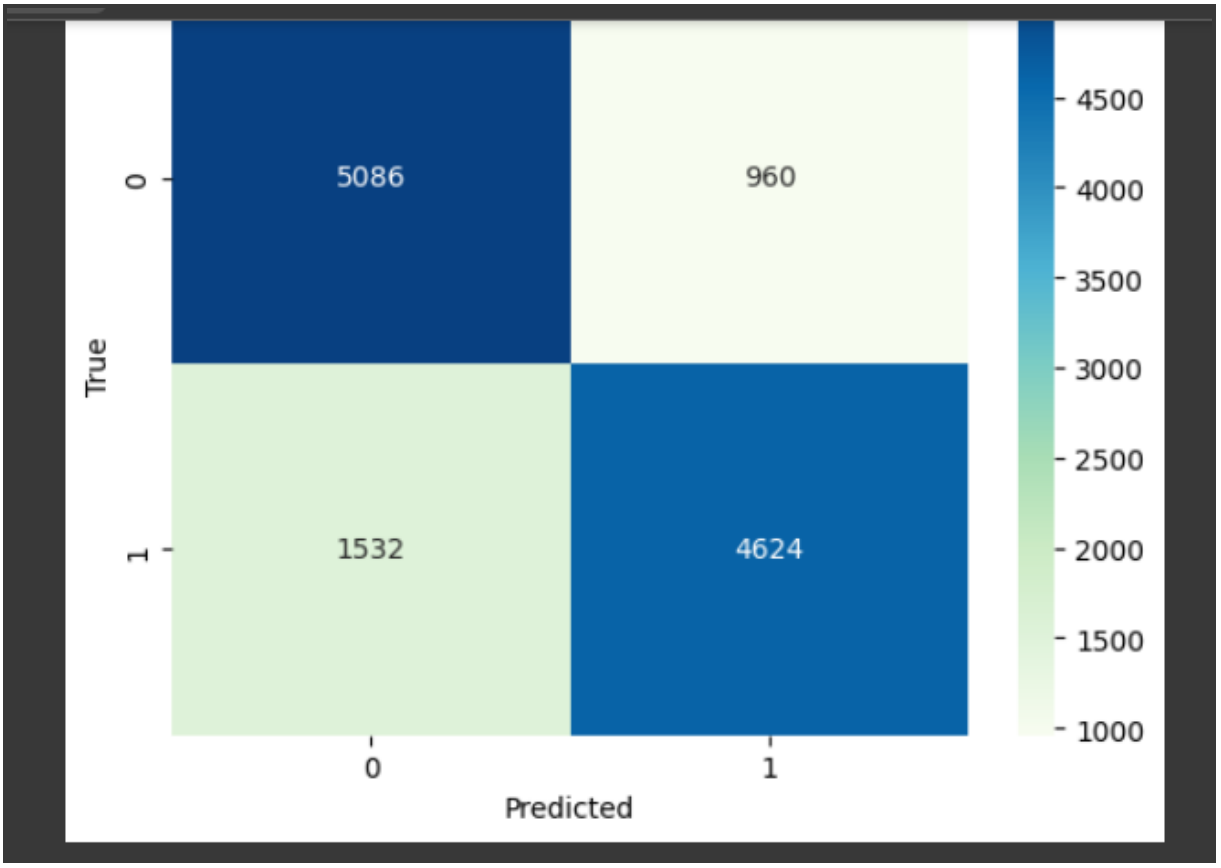
```
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- hotel_Resort Hotel >  0.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- booking_changes >  0.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- previous_cancellations >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- required_car_parking_spaces >  0.50
|   |   |   |   |--- class: 0
|--- deposit_type_Non Refund >  0.50
|   |--- customer_type_Transient-Party <= 0.50
|   |   |--- class: 1
|   |--- customer_type_Transient-Party >  0.50
|   |   |--- country_PRT <= 0.50
|   |   |   |--- class: 0
|   |   |--- country_PRT >  0.50
|   |   |   |--- class: 1
```

Vemos que la primera regla por la que decide clasificar fue por deposit_type Non Refund que es una dummy de los tipos de depósito. Decide separar por un umbral establecido en este caso con 0.50.
Luego continua con assigned_equals_reserved, despues con country_PRT que tiene mucho sentido dado que los hoteles están ubicados en Portugal

Vemos que toma esas decisiones dado que los atributos más considerados en este árbol fueron:

```
hotel_Resort Hotel: 0.004064312174447036
customer_type_Transient-Party: 0.008417469018179847
booking_changes: 0.011756899399145369
total_of_special_requests: 0.030410535601930806
previous_cancellations: 0.033226819142557186
average_daily_rate: 0.03718664594185886
lead_time: 0.07234795917454122
required_car_parking_spaces: 0.08332980247837111
country_PRT: 0.08969252926235621
assigned_equals_reserved: 0.09841909682357101
customer_type_Transient: 0.10525721878643467
deposit_type_Non Refund: 0.4258907121966065
```

D. Evaluar la performance del modelo en entrenamiento y validación, explicar todas las métricas y mostrar la matriz de confusión.




```
Accuracy: 0.7957711850516309
Recall: 0.751137102014295
Precision: 0.828080229226361
f1 score: 0.7877342419080067
```

Mejoramos el recall, mantuvimos la precisión y mejoramos el accuracy

ACCURACY = cantidad de aciertos / total

9710/12202 = 0.795771

Mide la cantidad de casos que mi modelo acierta. Se podría considerar una buena métrica, pero es engañosa si las clases están desbalanceadas, por ello es preferible usar otras métricas.

RECALL =

$TP/\ (TP\ +\ FN)$

4624/4624+1532 = 0.75113

Nos indica la proporción de casos correctamente clasificados como positivos frente a casos falsos negativos, nos permite visualizar cuantos fallo en clasificar. Es decir, que ahora nuestro último puede clasificar correctamente como positivos otros casos que antes tomaba como negativos

PRECISIÓN =

$TP/\ (TP\ +\ FP)$

4624/4624+960 = 0.8280802

Nos indica la proporción de casos correctamente clasificados como positivos frente a falsos positivos, podemos visualizar cuantos clasificó como positivos incorrectamente. Se mantiene la cantidad de falsos positivos identificados.

F1-SCORE = 2*(Precision * Recall) / (Precision+recall)

=0.787734

Combina el recall y la precisión, es la mejor métrica para tomar ya que nos da una mejor idea de cual es el árbol más preciso.