Checkpoint 2 - Grupo 05

Introducción

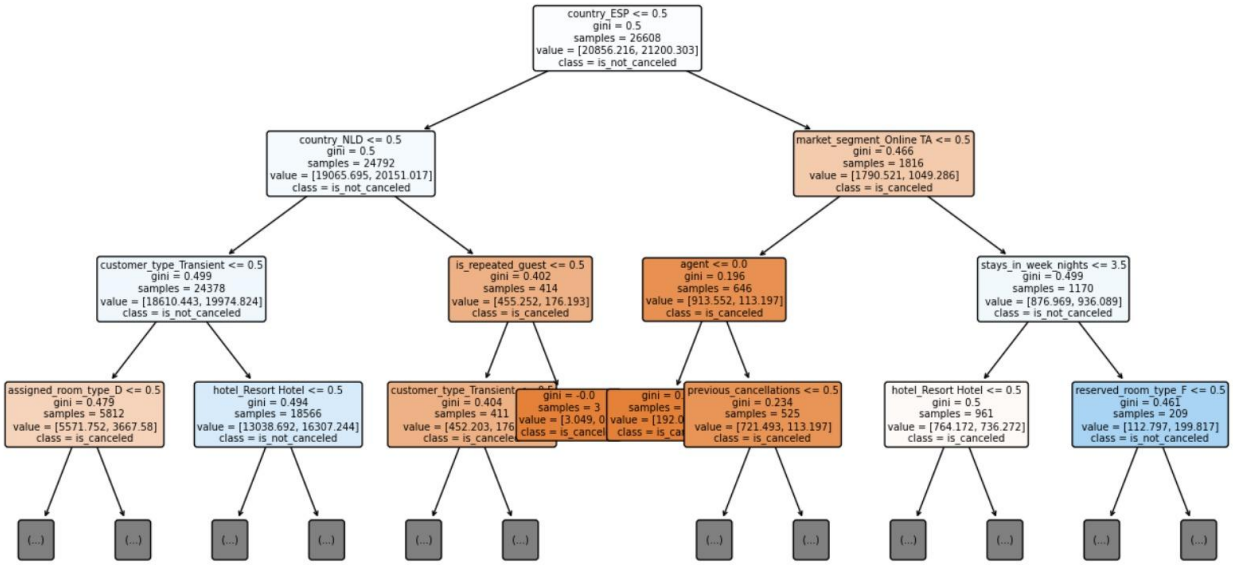
Realizamos modificaciones en ambos datasets (el de train que obtuvimos luego del chp1 y el test) para poder aplicar nuestros modelos de árboles. A hotels_test (al igual que habíamos hecho con hotels_train) concatenamos el año, el mes y el día en única variable. Convertimos el arrival date a date time para luego pasarlo a un valor numérico y luego (tanto a train como a test) a un int calculando una diferencia de días desde cierta fecha que usamos como referencia. Convertimos child de float a int y juntamos children y babies en una misma columna.
Para poder manejar los valores NaN de Agent y Company los convertimos al valor -1.
Luego, preparamos los datos mediante One-Hot Encoding para representar características categóricas de manera binaria, permitiendo a los modelos de árboles de decisión trabajar con ellas.
Realizamos nuestro proceso de modelado y optimización utilizando dos enfoques principales: Random Search y Grid Search para la búsqueda de hiperparametros, ambos aplicados en conjunción con Stratified K-Fold Cross-Validation.

Construcción del modelo

El mejor modelo que encontramos fue el segundo random search realizado.

- Optimizamos los hiperparámetros utilizando RandomizedSearchCV. En estos se incluyeron:
  - n_estimators: El número de árboles en el bosque.
  - max_depth: La profundidad máxima de los árboles.
  - min_samples_split: El número mínimo de muestras requeridas para dividir un nodo interno.
  - min_samples_leaf: El número mínimo de muestras requeridas en un nodo hoja.
  - class_weight: El peso de las clases en el modelo.
  - max_features: El número máximo de características a considerar para la división en cada nodo.
  - max_leaf_nodes: El número máximo de nodos hoja en un árbol.
  - random_state: Una semilla aleatoria para reproducibilidad.
- Usamos K-fold Cross Validation con 10 folds (n_folds = 10) en la validación cruzada.
- Consideramos el F1-Score como la métrica adecuada para buscar los hiperparámetros. Esta métrica es apropiada para problemas de clasificación en los que se busca un equilibrio entre precisión y recall.

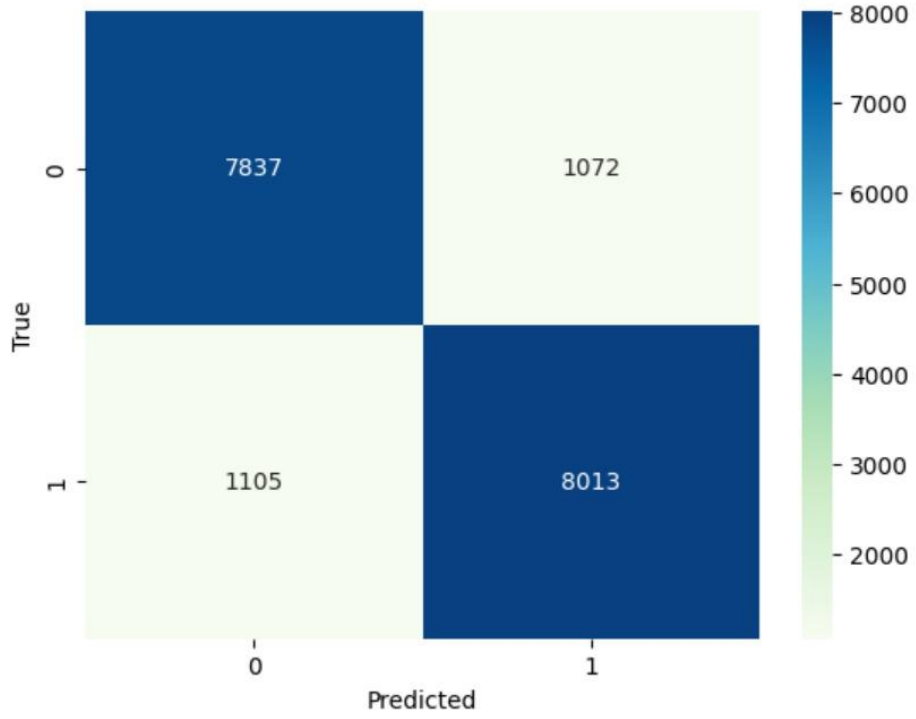
- El primer modelo de Random Search que hicimos nos dió con una métrica de 0.835819344108934 de F1 y el último modelo (el mejor) con 0.8804043289567653.



Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle
Random Search	0.8358193441089334	0.7943297441469992	0.8818819916648388	0.8247628557164254	0.84258
Nuevo Random Search	0.8804043289567653	0.8820033021463951	0.8788111427944725	0.8792367005047984	0.87775
Grid Search	0.8359139561467319	0.7942338072669827	0.8822110111866638	0.8248183280634603	0.81975

Matriz de Confusion



En la matriz se puede observar que hay 8013 casos en los que el modelo predijo que se cancelaba y efectivamente era así. 1105 casos en los que predijo que no se cancelaba pero en realidad eran reservas canceladas. 7837 casos en los que el modelo dijo que no se iba a cancelar y era así. Y 1072 casos que se predijo que se iban a cancelar y no fue así.
Se observa que la mayoría de las reservas fueron predichas correctamente.

Tareas Realizadas

Integrante	Tarea
Iara Jolodovsky	Modelos de Random Search, Preprocesamiento (One Hot) y Reporte
Tomas Vainstein Aranguren	Modelo de Random Search y modelo de Grid Search
Martin Abramovich	Modelo de Random Search y modelo de Grid Search