Checkpoint 2 - Grupo 03

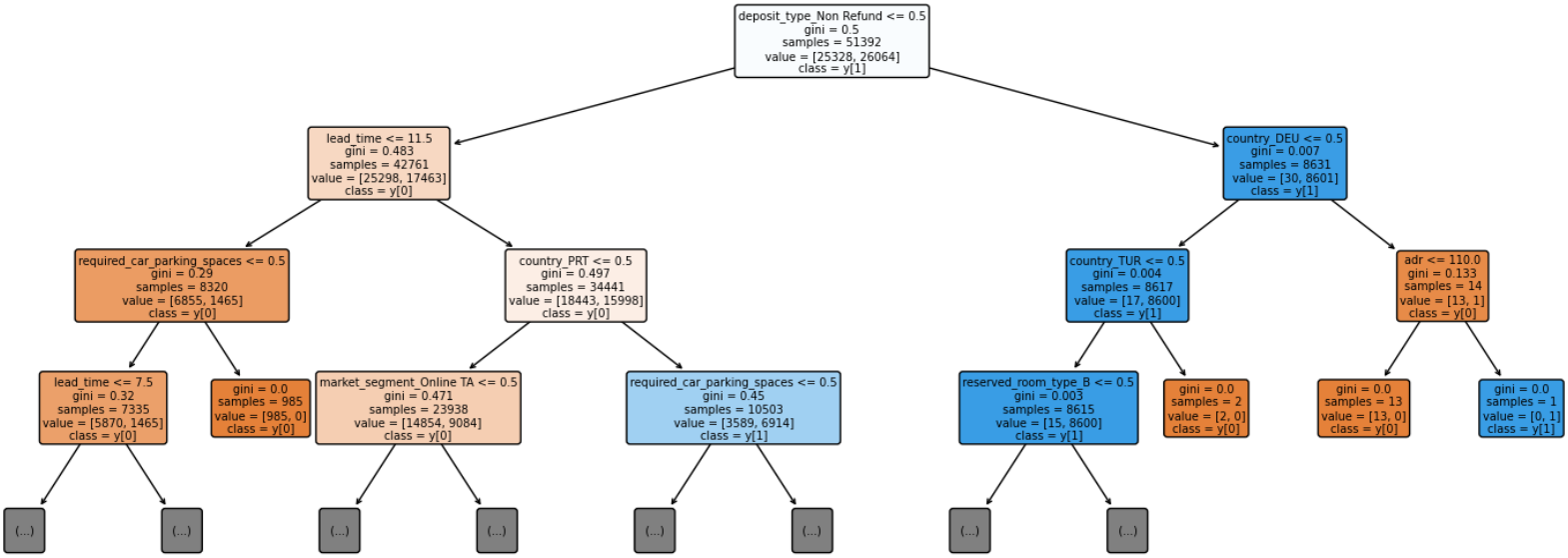
Introduccion

En un momento consideramos hacer recortes de filas de valores *outliers* con tal de subir el *F1-Score*, pero en los intentos de Kaggle rendía cada vez peor. Lo dejamos tal cual como venía del Checkpoint 1 al dataset, para evitar *overfitting*.
Para poder computar todas las variables cualitativas, se crearon variables *dummies* en su lugar, y se modificó el dataset de testeo para hacer coincidir las columnas en este sentido.
Por lo demás, utilizamos *K-Fold Cross Validation* y, donde el tiempo nos lo permitió, realizamos la búsqueda mediante *Grid Search*, priorizándolo frente al *Random Search*.

Construcción del modelo

- Optimizamos los hiperparámetros ***criterion***, ***max_depth***, y ***min_samples_split***.
- Hemos hecho uso de *K-fold Cross Validation*, con un total de hasta **12 folds** en la mejor iteración.
- Nosotros hicimos caso al ***F1-Score***, que media entre el *Recall* y *Precision* y además es el utilizado en la competencia de Kaggle.
- Los primeros intentos fueron adivinanzas al azar, varias con puntajes cercanos a *0.82* o a veces menores a *0.7*.

Fig. 1: Árbol de decisión: Mejor Iteración



Cuadro de Resultados

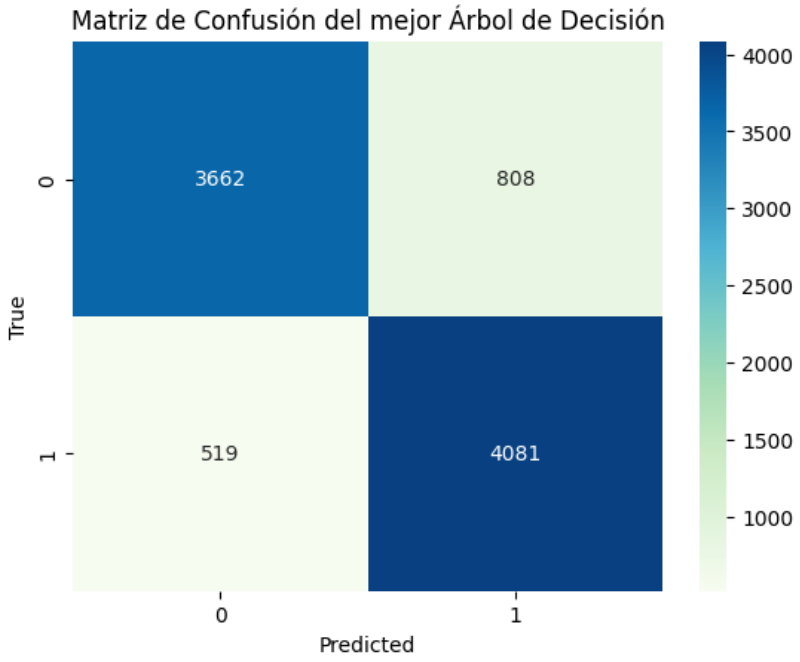
Modelo	F1-Score	Precision	Recall	Accuracy	Kaggle
modelo_1	0.8601539	0.8347310	0.8871739	0.8536935	0.84687
modelo_2	0.8601538	0.83473102	0.8871739	0.8536935	0.84344
modelo_3	0.8604109	0.8389095	0.8830435	0.8546858	0.84564

Matriz de Confusion

El mejor intento nos dejó con una matriz de confusión como la que se observa; en este intento los *splits* fueron distribuidos como:

- 15% Test
- 85% Train

Razón por la cual se observan anotaciones con números relativamente bajos.



Tareas Realizadas

Integrante	Tarea
Franco Lighterman Reismann	Creación de Árboles Optimización de Hiperparámeros Armado de Informe
Marcos García Neira	Optimización de Hiperparámeros Armado de Informe
Martín Andrés Maddalena	Optimización de Hiperparámeros Armado de Informe