Checkpoint 2 - Árbol de decisión

Análisis previo

Antes de empezar el análisis, filtramos el dataset, borrando algunas filas que contenían datos no coherentes con lo reflejado en el dataset de test y columnas que consideramos que no aportan información valiosa. Luego pasamos a preparar los datos, donde usamos one hot encoding para poder transformar las variables categóricas a variables dummies, y min-max scaling para normalizar las variables cuantitativas. Al hacer esto llegamos a la conclusión de que teníamos que agregar un dato en la columna que indicaba la habitación alquilada, ya que al eliminar los outliers en en checkpoint anterior se eliminaron los casos en los que se reservaban la habitación de tipo P, llevando a que no coincidan las columnas de los dos datasets. Por lo tanto la restauraremos estando siempre con valor igual a 0.

Después, pasamos a buscar los mejores hiperparámetros mediante K-Fold Cross Validation. En cuanto los folds utilizados decidimos usar 10, ya que teniendo en totalidad 47k de datos al dividirlo en 10 particiones quedamos con 4.7K de datos para entrenar, lo cual nos pareció un número sensato que no llevaría ni a errores de predicción muy similares la mayoría del tiempo ni a un tiempo muy alto de consumo. La métrica usada para elegir los mejores hiperparámetros fue el F1 Score.

Entrenamiento y Predicción

Teniendo los mejores hiperparámetros, pasamos a entrenar el árbol, logrando un F1 Score de `0.764191754215591`. Graficamos los atributos de clasificación en orden de importancia, y observamos que los mejores fueron: *deposit_type_Non_Refund*, *lead_time*, y *total_of_special_requests*, entre otros.

Luego mostramos el árbol generando un gráfico, y procedimos a evaluar el performance del árbol en el conjunto de evaluación, describiendo las métricas y mostrando la matriz de confusión.