Universidad de Buenos Aires
Facultad De Ingeniería
Año 2023 - 1er Cuatrimestre

Organización de Datos (75.06 /95.58)
Trabajo Practico 1: Reservas de Hotel

CHECKPOINT 2: Árboles de decisión
FECHA: 28/04/23

INTEGRANTES:

Gonzalez,Tomás <togonzalez@fi.uba.ar>	#108193
Moreno del Ruvo, Valentina <vmdr18@gmail.com>	#107948
Pol, Juan Manuel <jpol@fi.uba.ar>	#108448

Introducción
El objetivo de este segmento fue generar predicciones de la variable target *is_canceled* a partir de la creación y analisis de arboles de decisiones, y la optimización de sus hiperparámetros.

Transformación de variables categóricas
Decodificamos variables **dummies** a partir de cada valor de variables categóricas como *agent* o *meal* de manera que los arboles puedan trabajar con estas. Para el caso de la variable *company*, decidimos pasarla a la binaria *no_company*, que refiere a si hay o no una compañia responsable de la reservación.

Elección de variables
Dada la cantidad de valores unicos de las variables *agent* y *country*, decidimos generar 2 arboles (uno de baja profundidad y otro de mayor pero podado) para 3 casos distintos: **Sin ninguna de las dos**, **Solo con *country***, **Solo con *agent***. Y elegir la variable con la que mejor predicción logra el arbol.
Una vez probados todos los casos anteriormente descriptos, concluimos que el mejor caso era el dataset **solo con *country*(podado)** (priorizando el *f-score*). Además, notamos que el valor 'portugal' era el único que incidía significativamente en el arbol, por lo que optamos por dejar una sola variable *is_portugal* que refiera a si provienen de portugal o no.

Obtención del modelo
Una vez elegido el conjunto de variables para entrenar nuestro modelo, pasamos a la busqueda de hiperparámetros, utilizando **Random search** para obtener las combinaciones posibles al azar, y luego **10-fold CV** para medir dichas combinaciones. Nos quedamos con la combinación con mayor desempeño.

Testeo del modelo
Una vez que tenemos el mejor modelo, lo entrenamos con los datos de *train* para luego 'testearlo' con los datos de *test*. Obtenemos los resultados y subimos el *score*, de 0.81017, a la competencia de Kaggle.