

Checkpoint 2 - Grupo 39

Introduccion

Breve comentario de técnicas exploradas, pruebas realizadas y si efectuaron nuevas modificaciones sobre el dataset. Cualquier implementación realizada por el equipo se debe detallar en esta sección.

Iniciamos haciendo un preprocesamiento a el dataset de test para que tenga las mismas columnas que nuestro dataset de train. Esto consistió en eliminar algunas columnas, y transformar las columnas cualitativas de ambos datasets a variables dummies.

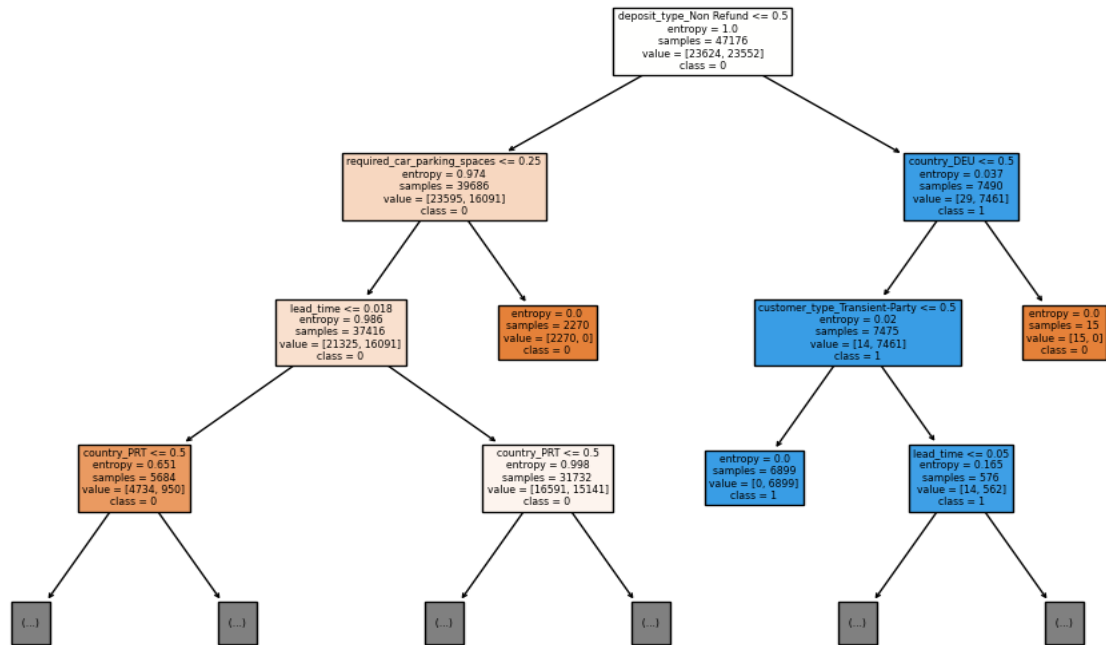
Primero utilizamos RandomizedSearchCV para buscar hiperparametros y subimos la predicción a kaggle para tener un primer acercamiento del puntaje con los datos de test. Luego queríamos ver cuanta diferencia había con un GridSearchCV. Y luego al GridSearchCV le aplicamos normalización y balanceo en sus datos y comparar. Tambien revisamos si podíamos mejorar el entrenamiento según el parámetro scoring.

Construcción del modelo

Los hiperparametros de nuestro mejor modelo fueron encontrados por GridSearchCV. Verificamos que con 7 folds devuelve el mejor score y los mejores hiperparametros que encontró son `ccp_alpha = 0`, `criterion = entrophy`, `max_depth = 5`.

Utilizamos f1-score para entrenar los modelos ya que es la mas consistente y es la que utiliza kaggle.

En el modelo de randomSearch, la predicción con la partición de datos de test del data set de train da 0.478 y en el mejor modelo devuelve 0.789.



El árbol es demasiado grande para mostrar completo por lo que mostramos una Imagen representativa del árbol.

La columna mas importante por la que empieza es si el tipo de deposito es reembolsable lo cual tiene mucho sentido en términos si las personas cancelan la reserva o no.

Se puede ver también según el tiempo de espera, si el cliente hizo una reserva para una fiesta o según el país de donde es el cliente.

Cuadro de Resultados

Modelo	F1-Test	Presicion Test	Recall Test	Accuracy	AUC-ROC	Kaggle
Random	0.4785	0.9978	0.3147	0.655	0.657	0.7852
GridSearch	0.7899	0.6844	0.9337	0.7502	0.7492	0.7957
GS_balance_normalizacion	0.8867	0.8638	0.9110	0.8829	0.8828	0.7624

Mejor modelo: GridSearch

Vemos que GS_balance_normalizacion tiene muy buenos scores con la partición de test de nuestro dataset de train(mejores que nuestro mejor modelo elegido), pero cuando predecimos con el dataset de test devuelve un score menor.

Una razón de esto puede ser que el modelo este haciendo overfitting, por lo que al predecir con datos nunca vistos no es muy bueno.

Tareas Realizadas

Integrante	Tarea
Agustin Ezequiel Sanchez Decouflet	Preprocesamiento de Datasets Preprocesamiento para entrenar RandomizedSearchCV Busqueda hiperparametro Scoring Matriz de confusiones
Cattaneo Ariadna Antonella	Preprocesamiento de Datasets Preprocesamiento para entrenar GridSearchCV Busqueda hiperparametro Scoring Graficos de arbol
Franco Dario Mazzaro	Preprocesamiento de Datasets Preprocesamiento para entrenar GridSearchNormalizacionBalanceo Busqueda hiperparametro Scoring Armado de reporte