Checkpoint 2 - Grupo 26

Introduccion

Empezamos limpiando los dataset de entrenamiento que obtuvimos en el chp1 y el test. Para el test, decidimos aplicarles las mismas transformaciones que hicimos para el train. Luego, tuvimos que aplicar ciertas transformaciones de variables a ambos datasets, para que al momento de entrenar los modelos, no hubiese ningún problema.

Una vez con los datasets limpios, nos pusimos a entrenar 3 modelos diferentes cada uno del grupo, tratando de variar lo más posible, así podríamos explorar resultados diferentes.

Finalmente, se eligió el modelo con la métrica **f1 score** más alta, el cual resultó ser el modelo 1.

Construcción del modelo

El mejor predictor lo elegimos en función de la puntuación de kaggle, que vendría a ser el **modelo 1**. Para este modelo, decidimos optimizar sus hiperparametros, que fueron:

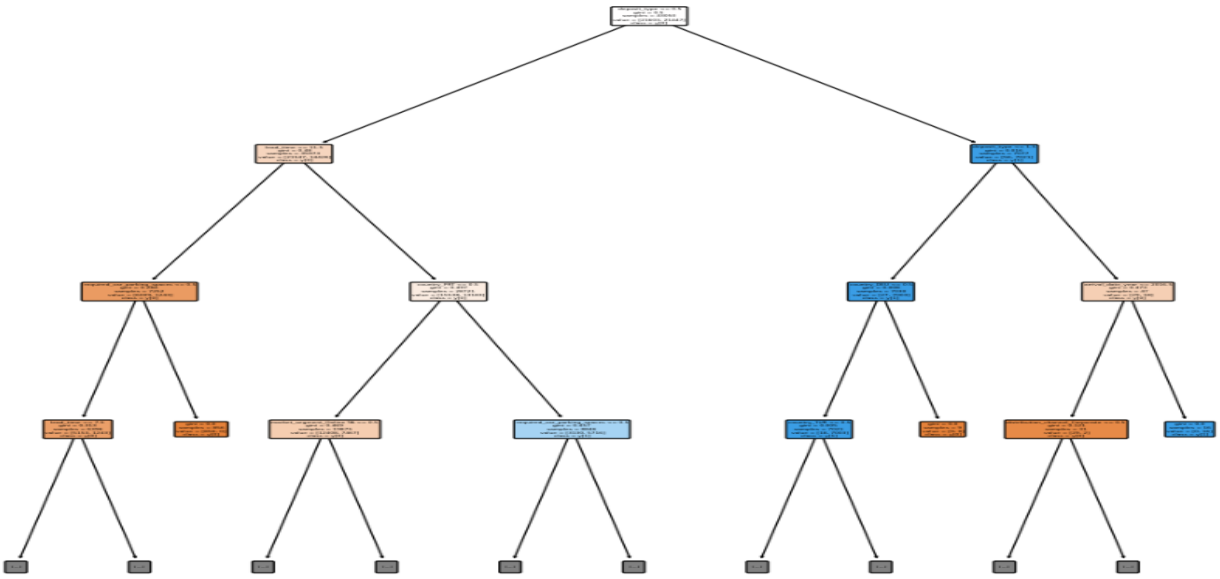
- max_depth: controla la profundidad máxima del árbol.
- Min_samples_leaf: especifica el número mínimo de muestras requeridas en un nodo hoja del árbol.
- Ccp_alpha: controla la complejidad del árbol utilizando el enfoque de poda por complejidad de costo.
- Min_samples_split: especifica el número mínimo de muestras requeridas para realizar una división en un nodo interno del árbol.

Para optimizar estos parámetros, usamos el método "**Random Search Cross Validation"**, en donde consideramos que, la métrica adecuada para decidir si el parámetro fue optimizado o no, es la de **F1 score**. Decidimos usar 10 folds, para que el entrenamiento del modelo no tarde mucho.

Para los primeros entrenamientos del modelo 1, la métrica empezó con un valor de 0.85335. Por cada entrenamiento que hacíamos, empezaba a aumentar a 0.85423, 0.85594. Finalmente, con el último entrenamiento, llegó a 0.85527.

Grafico del árbol

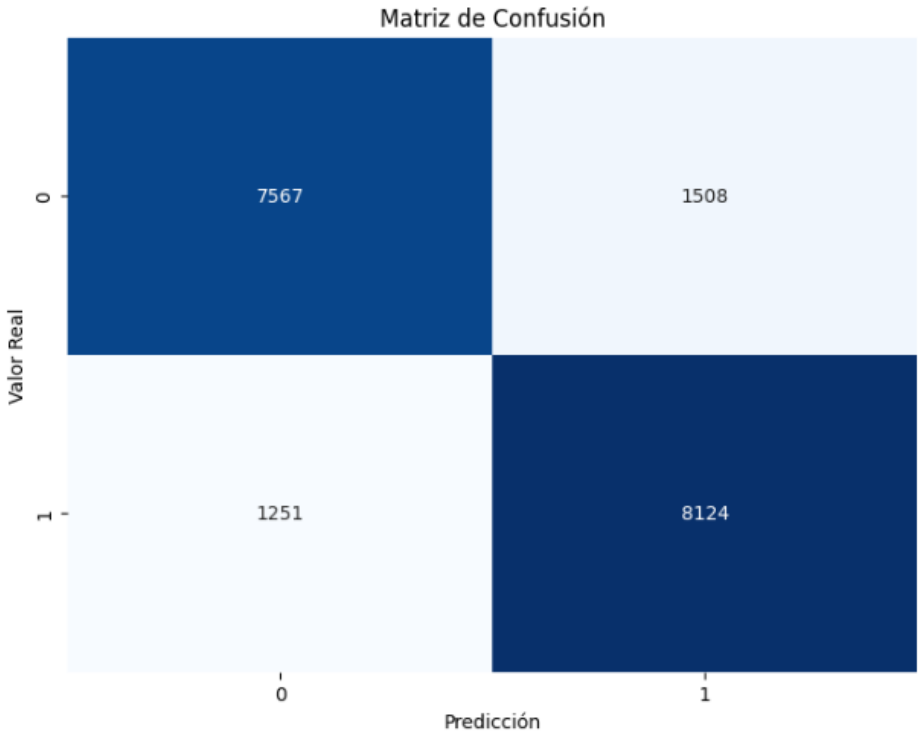
Gráficas solo una parte del árbol del modelo 1, ya que al contener tantos nodos, no entraría en árbol completo::



Cuadro de Resultados

Modelo	F1-Test	Presicion Test	Recall Test	Metrica X	Kaggle
1	**0.85527**	**0.83996**	**0.87114**	**0.85018**	**0.84833**
2	0.85528	0.84257	0.86837	0.85067	0.84519
3	0.85478	0.84342	0.86645	0.85040	0.84684

Matriz de Confusion



A través de esta matriz, vemos lo siguiente:

- Verdaderos positivos: casos en donde el modelo predijo de forma correcta cuando una reserva es cancelada.
- Verdaderos negativos: casos en donde el modelo predijo de forma correcta cuando una reserva no es cancelada.
- Falsos positivos: casos en donde el modelo predijo incorrectamente si la reserva fue cancelada
- Falsos negativos: casos en donde el modelo predijo de forma incorrecta si la reserva no fue cancelada

Tareas Realizadas

Integrante	Tarea
Carbajal Robles, Kevin Emir	Reporte Limpieza de los datasets Entrenamientos de modelos
Vallcorba, Agustin	Reporte Limpieza de los datasets Entrenamientos de modelos
Garcia, Nicolas	Reporte Limpieza de los datasets Entrenamientos de modelos