75.06 / 95.58

Organización de datos

Reporte TP N°1: Reservas de Hotel

Clasificación - Entrenamiento y Predicción: Checkpoint N°2

Alen Davies Leccese: 107084

Luca Mauricio Lazcano: 107044

Abril 2023

1 Introducción

Para esta parte del trabajo, se utiliza como base el dataset procesado durante la primera etapa del trabajo: `hotels_procesado.csv`. Este dataset se encuentra limpio de valores nulos, y se le quitaron los valores que consideramos atípicos. Además, contiene una nueva columna `arrival_date`, de tip `object`, con la fecha formada a partir de las demás columnas que contenían esta información, y las columnas `company` y `agent` son de tipo `object`, ya que se imputaron los valores nulos.

En primer lugar, se cargaron los datasets a utilizar en esta etapa: `hotels_procesado.csv` cargado como `df_train`, y `hotels_test.csv`. De más está aclarar que estos son los datasets de entrenamiento y de prueba respectivamente.

Luego, al dataset de entrenamiento se le eliminan las columnas de `id` y `arrival_date`. `id`, porque no aporta información del problema, sólo permite identificar cada registro; y `arrival_date` por el doble motivo de ser de tipo objeto, y porque la información "temporal" ya está convenientemente presente en otras columnas.

Además, convertimos las columnas `company` y `agent` a valores numéricos, para ser uti-

lizadas en el entrenamiento. Se habían imputado los valores nulos como "Sin compania" y "Sin agente" respectivamente, y se transforman al número cero, convirtiendo la columna a tipo entero.

Finalmente, se quita la columna `reservation_status_date` del dataset de prueba. Como ya se explicó anteriormente, esta columna es un dato que se registra luego de cancelada o no la reserva, por lo que está totalmente relacionada con el "target" `is_canceled`, y lo filtraría al modelo si se usase en el entrenamiento. También se tratan los datos nulos de `company`, `agent` y `country`.

De esta forma, se tienen los datasets listos para ser usados en el entrenamiento del modelo y pruebas de predicción.

2 Hiperparámetros

Para el entrenamiento de los árboles de decisión, se optó por realizar "one hot encoding" a las variables categóricas. Originalmente, el "encoding" se realizó con la funciónn `get_dummies()`, pero al momento de realizar la predicción, nos encontramos con un error, donde las columnas del dataset de entrenamiento y prueba no eran iguales. Esto se debía a que, al realizar el encoding, se crearon varias columnas con variables que aparecen en el dataset de entrenamiento pero no en el de test, y viceversa. Esto sucede cuando, por ejemplo un país, aparece de forma muy infrecuente. Era necesario considerar esto de alguna forma. Se optó por agregar a cada dataframe las columnas faltantes respecto al otro.

A continuación, se divide el dataset de entrenamiento para entrenar el modelo y testearlo.

Para la búsqueda de los hiperparámetros óptimos para entrenar el modelo, y que resulten en el mejor poder de predicción posible, contemplamos dos alternativas principales: "Random Search" y "Grid Search". Ambos, dados intervalos para los hiperparámetros, probarán distintas combinaciones, devolviendo los mejores hiperparámetros que encuentre. Lo hacen de forma distinta: "Random Search", como su nombre lo indica, prueba combinaciones aleatorias dentro del espacio de posibilidades; mientras que "Grid Search" lo hace de forma metódica, probando cada combinación posible del espacio de posibilidades. Mientras que "Random Search" es más rápido y puede proveer hiperparámetros satisfactorios, "Grid Search" tenderá a producir mejores resultados. Probamos ambos, y considerando la mejora en los resultados, nos quedamos con "Grid Search". Los rangos probados y los hiperparámetros obtenidos de esta forma están debidamente detallados en el notebook. Es importante notar que se puede probar un espacio mayor de posibilidades, pero el tiempo que tomaría crece de forma exponencial. En una de las primeras

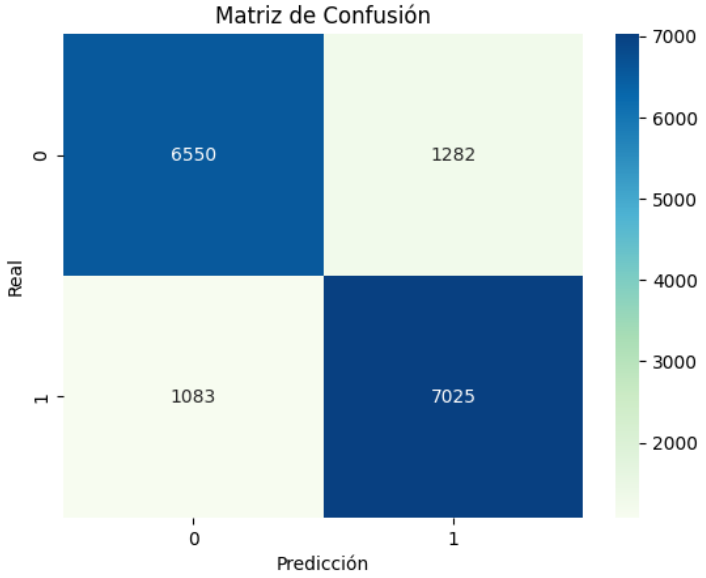
pruebas, probamos rangos de hiperparámetros bastante exhaustivos, y pasada la media hora de búsqueda, consideramos la posibilidad de agregar algo que permita "loguear" el progreso, y así tener una noción de cuánto tiempo tardaría en encontrar los hiperparámetros. Esto se logra agregando `verbose=n`, con n = 2, a los parámetros de `GridSearchCV`. Encontramos que, a promedio de 0,7 segundos por combinación de hiperparámetros a probar, la búsqueda que intentamos tardaría... 27 días. Incluso con una computadora con mayor capacidad de procesamiento, estas cifras son irrealistas y no resultaría práctica la reproducción de los resultados. Nos quedamos finalmente con una búsqueda cuyo tiempo es de unos pocos segundos, y provee resultados aceptables.

En resumen, se utilizaron 10 folds, maximizando el "F1 score", ya que este es el utilizado por Kaggle para evaluar las submissions.

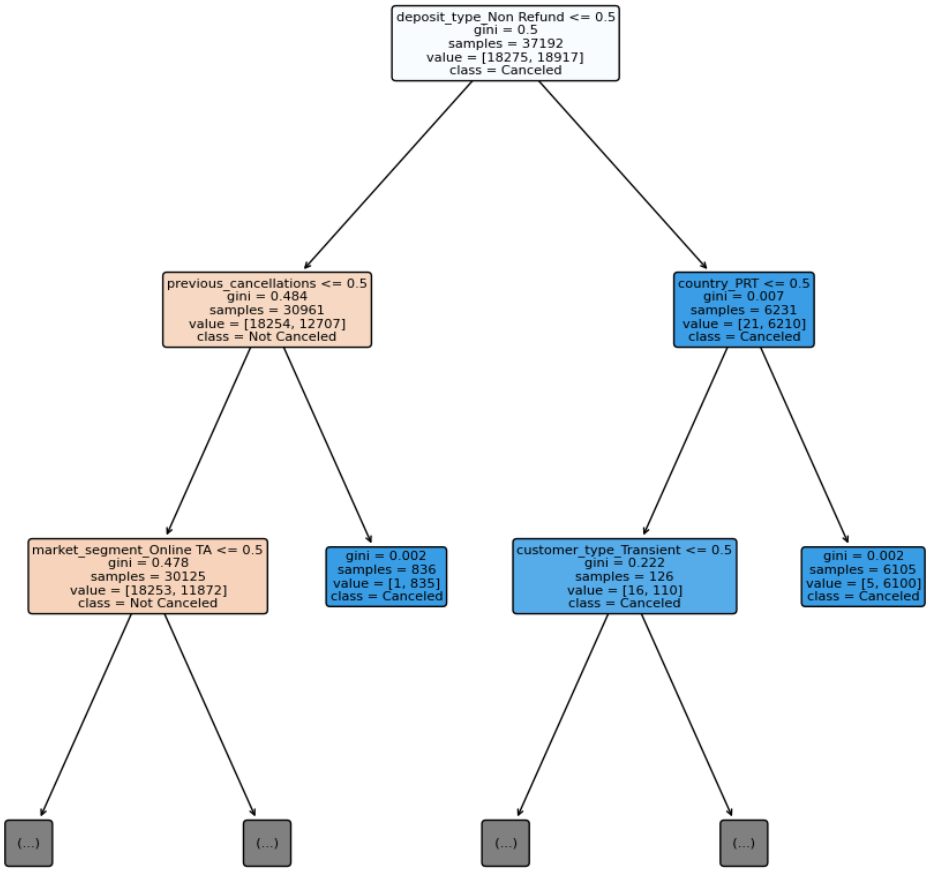
3 Árbol de decisión

Con los hiperparámetros encontrados, se crea el `DecisionTreeClassifier`, y se crea el modelo fiteando la fracción del dataset utilizada para entrenamiento. Luego se realiza una predicción con la fracción utilizada para test. La mejor predicción encontrada acierta el 85,2% de las veces para el dataset de prueba, y 87,4% para el de entrenamiento, con un "Cross Validation Score" de 85,1%, sugiriendo que el modelo generaliza bien para todo el dataset, y está levemente overfiteado.

Se provee un gráfico de la matriz de confusión resultante.



Se provee también un gráfico de los primeros tres niveles del árbol de decisión.



Se puede apreciar que la característica más importante es el tipo de depósito "Non refund". Le sigue el "lead_time" y justo después "country_PRT". Si bien esta última no es la métrica más importante, sugiere que si se quisiese utilizar el modelo para predecir reservas canceladas en hoteles de otro lugar del mundo, ser de nacionalidad portuguesa sería "perjudicial", ya que el modelo es bastante "prejuicioso". Esto se explica ya que el dataset está recogido de hoteles de Portugal. Realizamos pruebas sin considerar el "country", es decir, dropeándola del dataset de entrenamiento, pero la performance empeoró. Finalmente decidimos conservar esta variable, ya que asumimos que el dataset de evaluación está recogido originalmente del mismo dataset del que proviene el de entrenamiento.

Por último, se realizó la submission a Kaggle, con el archivo `submission11.csv`, obteniendo un puntaje levemente menor al conseguido en el notebook.

4 Conclusiones

Esto da por concluida esta etapa del trabajo. Luego de bastantes pruebas de hiperparámetros, bastante tiempo de procesamiento y esperanza por obtener el mejor poder de predicción posible, obtuvimos resultados que consideramos aceptables. Después de todo, se pueden predecir las reservas canceladas con una certeza del 85% (aproximadamente, veremos los resultados de las pruebas privadas en Kaggle).

Durante el trabajo, revisamos decisiones tomadas en la anterior etapa del trabajo, adaptamos el dataset a nuevos requerimientos, pero dentro de todo creemos que las decisiones en la primera etapa del trabajo fueron bastante acertadas. No tuvimos demasiado overfitting, aunque, como mencionamos, restan ver los resultados de las pruebas privadas.