UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
Maestría en explotación de datos y descubrimiento del conocimiento

Año 2020 - 1er Cuatrimestre

APRENDIZAJE AUTOMÁTICO

TRABAJO PRÁCTICO Nº 1
TEMA: Algoritmos de generación de árboles de decisión
FECHA: 17 de junio de 2020

INTEGRANTES:

Paez Press, Ximena
<ximena.paezpress@gmail.com>
Villalba, Ricardo Demián
<rikivillalba@gmail.com>
Massaro Rocca, Patricio
<patomassaro@gmail.com>

***Resumen*—Los algoritmos de generación de árboles de decisión han sido ampliamente utilizados para construir modelos de predicción y otras aplicaciones, destacando por su semejanza con el razonamiento humano y la facilidad para entender las reglas que lo construyen. El objetivo de este trabajo consiste en analizar en detalle estos algoritmos, aplicándolos a la predicción de la cancelación de una reserva de hotel utilizando un set de datos de reservas proveída por la cátedra.**

En primer lugar, se aplicaron distintos técnicas de preprocesamiento para mejorar la calidad del conjunto de datos, luego se usaron técnicas de aprendizaje automático para entrenar un modelo y posteriormente ajustarlo para maximizar el rendimiento de la predicción. Por otro lado, este tipo de algoritmos requieren de un grupo de hiperparámetros cuyo valor es una incógnita sin una respuesta trivial. A lo largo del trabajo se evaluaron los efectos que los cambios en este grupo pueden traer al resultado final.

I. Introducción

El aprendizaje de árboles de decisión es uno de los métodos de inferencia inductiva más usados. Este método realiza aproximaciones a funciones de valores discretos, siendo representadas por un conjunto de reglas *if-then* que facilitan la comprensión humana. Gracias a características como robustez frente a ruido y tolerancia a valores nulos en los conjuntos de entrenamiento, han sido ampliamente usados en tareas como diagnóstico médico o calificación de riesgo crediticio [1]. Estos algoritmos poseen un conjunto de parámetros que deben ser seleccionados cuidadosamente para evitar comportamientos indeseados y performance bajas. El objetivo del trabajo es estudiar las particularidades de la implementación de estos algoritmos, repasando también conceptos comunes a la mayoría de los problemas de aprendizaje automático como división de conjuntos de datos en entrenamiento y prueba o validación cruzada.

En la sección II, se presenta una descripción detallada de los datos, obtenidos de la plataforma *Kaggle* [2], y las transformaciones que formaron parte del preprocesamiento de los mismos. Por otro lado, se presenta un análisis de asociación con la variable objetivo. A lo largo de la sección III se explican las técnicas utilizadas a lo largo del trabajo para abordar los objetivos del mismo. Los resultados y análisis pertinentes se presentan en la sección IV. Las conclusiones pueden verse dentro de la sección V.

II. Datos

Al comenzar con el análisis exploratorio, se detectó que el conjunto de datos presenta las reservas realizadas en dos hoteles denominados *City Hotel* y *Resort Hotel* en una proporción de 66.4 % y el 33.6 % respectivamente. Las mismas acontecen a lo largo de los años 2015, 2016 y 2017.

Las reservas poseen un tipo de depósito asociado que indica el estado de pago de la misma, siendo posible adoptar tres distintos valores *No Deposit* (88 %), *Non Refound* (12 %) y *Refundable* (0 %). Adicionalmente, las mismas tienen asignado un tipo de habitación reservada (con 10 valores posibles) y un tipo de habitación efectivamente asignado (con 12 valores posibles). Los huéspedes pertenecen a 117 países distintos y son agrupados en una de 4 posibles categorías.

En lo que respecta a las variables del tipo numéricas, son de notoria atención las relacionadas al tiempo antelación con el cual se realiza la reserva, cantidad de cancelaciones previas, números de cambios realizados, precio promedio abonado por noche, número de pedidos especiales realizados y si se trata de un huésped recurrente.

Más detalle del análisis exploratorio realizado sobre los elementos del conjunto de datos se encuentra en la tabla IV, incluida en el anexo.

Otro hallazgo realizado durante esta actividad, y que vale la pena resaltar, se trata de la aparente duplicidad de reservas en la fuente de datos. En primer lugar, se analizó la posibilidad de eliminar estos registros, pero se decidió descartar este procedimiento debido a la granularidad que representa cada línea en el conjunto de datos. Al no poseer un nivel de detalle más atómico (como por ejemplo un ID de huésped), se consideró que era posible que existieran varias reservas con iguales características en caso que una agencia de viaje reservara varias habitaciones para un contingente de pasajeros o una familia ocupara más de un cuarto.

II-A. Tratamiento de valores faltantes

Para continuar con el análisis exploratorio, se buscaron identificar aquellos atributos que contaran con datos faltantes. El estudio arrojó los siguiente resultados:

- *company*: presenta un 94 % de valores nulos
- *agent*: presenta un 13 % de valores nulos
- *country*: presenta un 0.4 % de valores nulos
- *children*: presenta 4 casos de valores nulos

Se decidió dar tratamiento a los mismos de diversas maneras. En el caso de *company*, y debido a la gran presencia de valores nulos, se decidió no incorporar este feature en la construcción del árbol y, por ende, desestimar esta columna del conjunto de datos.

En el caso de *agent*, se procedió a asignar un valor ficticio generado por el equipo de desarrollo. Esta decisión fue motivada por el hecho que esta columna posee 333 valores distintos y adoptar el valor más frecuente podía implicar efectos negativos en el posterior entrenamiento del árbol.

Finalmente, para tratar los casos donde no se poseía un valor informado para la variable *children*, se procedió a completar los mismos con el valor 0, asumiendo que en este caso la reserva no cuenta con niños.

II-B. Análisis de correlaciones

Se estudió la correlación que existen entre las variables numéricas y categóricas con la variable target a predecir *is_canceled*.

Respecto a las variables del tipo numérico, se listan en la tabla I las 4 que poseen una correlación más elevada y su correspondiente coeficiente.

variable	correlación
lead_time	0.293123
total_of_special_requests	0.234658
required_car_parking_spaces	0.195498
booking_changes	0.144381

Cuadro I: Correlación entre variables numéricas y variable target *is_canceled*

Se puede observar en la figura 1 que aquellas reservas que se realizan con mayor antelación son más proclives a sufrir cancelaciones, mientras que lo contrario sucede con

la cantidad de pedidos especiales que posee la misma, que incrementan en las reservas que efectivamente se llevan a cabo, como puede observarse en la figura 2.

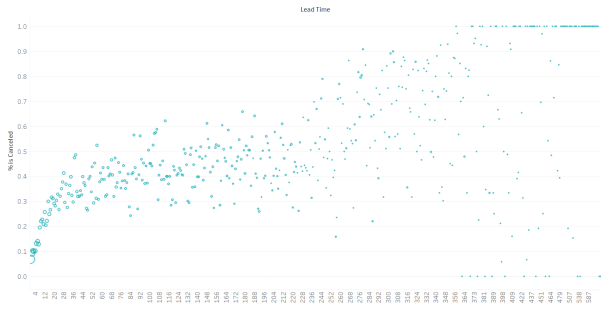

Figura 1: Porcentaje de reservas canceladas en base a la antelación de la misma

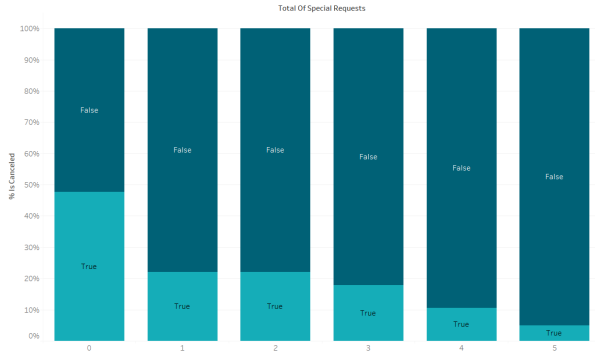

Figura 2: Porcentaje de reservas canceladas en base a la cantidad de pedidos especiales que recibe

En cuanto a la correlación que se pudo capturar entre la variable objetivo y los features categóricos, encontramos el detalle en la tabla II. Para cuantificar la asociación se utilizó el coeficiente V de asociación de Cramer [3], basado en el estadístico χ^2. Como características más vinculadas vemos las variables *reservation_status* y *deposit_type*.

Variable	Cramer
reservation_status	0.999996
deposit_type	0.481464

Cuadro II: Correlación entre variables categóricas y variable target *is_canceled*

Respecto a la correlación que existe con *reservation_status*, se explicará en mayor detalle su tratamiento en la subsección II-C. En cuanto al tipo de depósito asociado a la reserva, es llamativo descatar que, contrario al sentido común, una gran parte de las reservas que no poseen reembolso en caso de cancelación se encuentren canceladas. Más detalle puede observarse en la figura 3.

II-C. Eliminación de features y generación de dummies

Realizando un análisis de los features que se podían remover de la construcción del árbol, se procedió a eliminar en primer lugar el campo *reservation_status*. Para tomar tal decisión se tuvo en cuenta que la misma posee estricta correlación con la variable a precedir, dado que todas las reservas canceladas poseen únicamente los estados *Canceled* o *No Show*, mientras que todas las reservas que efectivamente acontecieron poseen el estado *Check Out*.

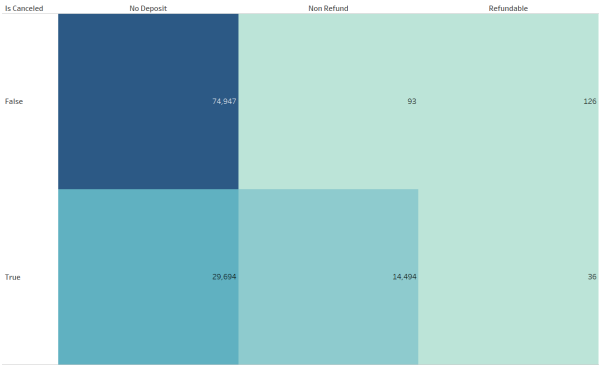

Figura 3: Relación entre reservas canceladas y tipo de depósito

En segunda instancia, se tomó la decisión de eliminar del estudio a las variables relacionadas con temporalidad pasada, dado que las mismas no ayudarían a modelar comportamiento futuro. Por este motivo se procedió a descartar del conjunto de datos los features *reservation_status_date* y *arrival_date_year*.

Finalmente, y para evitar que el árbol sobre-ajustara debido a la gran cantidad de valores únicos que podían tomar estas variables, se concretó eliminar las variables *agent* y *country*.

Para aquellas variables categóricas que se conservaron en el modelo (*hotel, meal, market_segment, distribution_channel, reserved_room_type, assigned_room_type, customer_type y deposit_type*) se utilizó la técnica de construcción de dummies para poderlas incorporar en el análisis.

II-D. Construcción de variables

Como último paso antes de comenzar con la elaboración del árbol de decisión, se procedió a transformar a los meses que originalmente poseían características de atributos descriptivos a una variable del tipo ordinal.

III. Metodología

III-A. Métrica de performance

Se decidió utilizar el *recall* como métrica de performance. La decisión se fundamentó en la necesidad planteada de detectar la mayor cantidad de cancelaciones posibles, sin importar los falsos positivos. El equivalente en términos de fβ-score es utilizando un β que tienda a infinito.

III-B. Conjuntos de entrenamiento y prueba

Se separó el set de datos en conjuntos de entrenamiento y prueba, el primero se utilizó para todos los procedimientos de entrenamiento y ajuste de parámetros. Luego, para verificar el comportamiento de modelo, se utilizó el conjunto de prueba.

III-C. Búsqueda de Hiperparámetros óptimos

Una vez definida la métrica a maximizar y luego de preprocesar los datos, se realizó un procedimiento de *Grid Search*, iterando sobre distintas combinaciones de hiperparámetros del árbol para buscar la que optimice la performance. Para minimizar el sesgo del modelo, se utilizó validación

cruzada durante la búsqueda. Los parámetros encontrados fueron los utilizados durante el resto del trabajo.

III-D. Costo de complejidad

El parámetro de costo de complejidad (α), permite realizar una poda del arbol, disminuyendo su profundidad y haciendo un equilibrio entre ajuste y capacidad de generalización. De esta manera, se puede evitar sobre ajustar el algoritmo a los datos de entrenamiento [1] En primer lugar, con propósitos exploratorios, se utilizó un algoritmo de Grid Search con todos los hiperparámetros fijos excepto α. Finalmente, se iteró a lo largo de una lista de valores de α y se graficaron los efectos en la performance sobre el conjunto de entrenamiento y de prueba. Al encontrar una configuración que obtenía mejor performance respecto del obtenido mediante Grid search, se compararon las matrices de confusión.

III-E. Importancia de los descriptores

A través de un algoritmo de eliminación recursiva se obtuvo la importancia de los descriptores. Se utilizaron los 3 más significativos para entrenar un nuevo árbol de decisión y se evaluó su performance para compararla con el árbol original. Esto permite ver como se comporta un árbol entrenado con una cantidad menor de variables y por lo tanto, mas simple.

IV. Resultados

IV-A. Balance del conjunto de datos

Se puede observar que las reservas canceladas representan un 37 % del total del dataset, lo cual implica que el conjunto de datos se encuentra moderadamente desbalanceado. Como consecuencia, utilizar sólo la precisión como métrica de performance del modelo no sería adecuado (i.e. un árbol que clasificara todos los casos como no cancelados aún tendría una precisión del 63 %). Se considera, además, que interesa identificar **todas** las posibles cancelaciones, es decir, no subestimar ninguna posible cancelación aún a costo de predecir cancelaciones que no ocurran. Se busca maximizar la sensibilidad (*recall*) del modelo a fin de minimizar los falsos negativos, aunque ello implique penalizar la precisión y aumentar la proporción de falsos positivos.

IV-B. Determinación de hiperparámetros óptimos

La búsqueda de hiperparámetros se realizó mediante GridSearch, obteniéndose los siguientes resultados:

ccp_alpha	0.0,
criterion	'gini',
max_depth	33.0

IV-C. Entrenamiento sobre conjuntos aleatorios

Con los parámetros obtenidos, se realizó una serie de entrenamientos sobre 50 conjuntos aleatorios de entrenamiento y validación, en proporción 80/20 por cada muestreo y se obtuvieron las métricas de performance de cada entrenamiento (en este caso *recall*). En la figura 4 se muestra la distribución de la performance sobre los conjuntos de entrenamiento y validación.

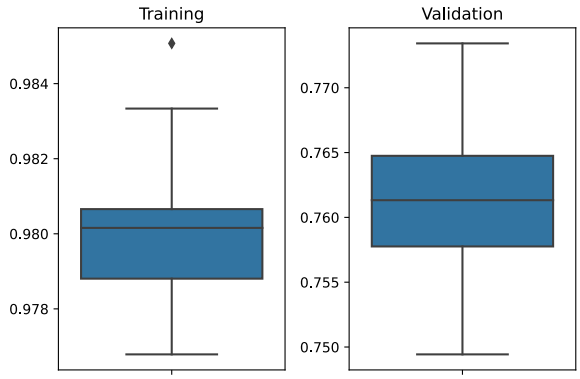

Figura 4: Performance para 50 conjuntos de entrenamiento/validación

IV-D. Validación cruzada

Se realizó validación cruzada de 50 iteraciones utilizando los índices obtenidos mediante *StratifiedKFold*. En la figura 5 se muestra la distribución de la performance sobre los conjuntos de validación.

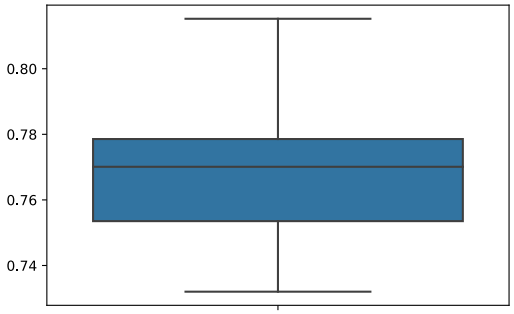

Figura 5: Performance con validación cruzada de 50 iteraciones

IV-E. Evaluación de modelos con distintos valores de α

Para este procedimiento se tomó una serie de distintos valores de α y en cada caso se utilizó validación cruzada de 10 iteraciones. Los valores de performance (*recall*) para entrenamiento y validación, y la profundidad de cada serie se muestra en las figuras 6 y 7. Una observación es que el rango de valores de α considerado se escogió con valores deliberadamente pequeños (<0.00025) de modo de poder representar las zonas de performance y profundidad más sensibles a este parámetro. Como puede observarse, la profundidad de los árboles disminuye rápidamente con incrementos muy pequeños de α.

Interesa comparar el desempeño del árbol con poda con mejor performance con respecto a un árbol sin poda (en este caso corresponde al árbol del modelo óptimo). La figura 8 muestra las tablas de contingencia del árbol sin poda y del mejor árbol con $\alpha > 0$, en este caso $\alpha > 0{,}000005$, siendo la performance de sensiblidad ligeramente superior en el árbol sin poda.

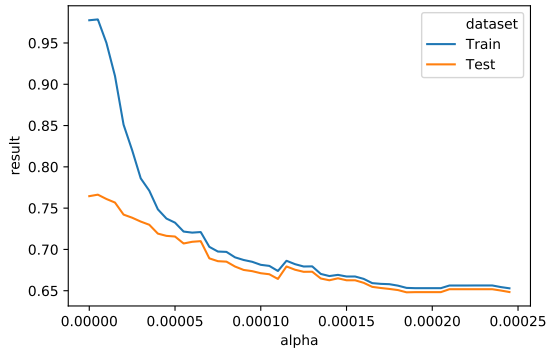

Figura 6: Performance para distintos α con validación cruzada

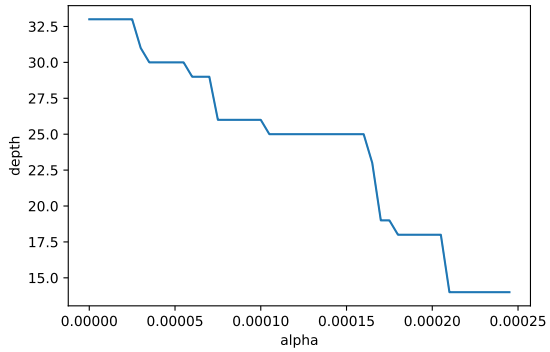

Figura 7: Profundidad para distintos α con validación cruzada

IV-F. Ranking de descriptores

Se utilizó la técnica de eliminación recursiva para obtener la lista de descriptores según su relevancia. El algoritmo permite determinar el número de descriptores relevantes deseados (en nuestro caso tres) y devuelve una lista: a los tres principales descriptores seleccionados les asigna un orden de importancia igual a 1 y al resto un orden sucesivo por orden de importancia decreciente (cuadro III).

descriptor	importancia
lead_time	1
adr	1
deposit_type_Non Refund	1
arrival_date_day_of_month	2
arrival_date_week_number	3
stays_in_week_nights	4
total_of_special_requests	5

Cuadro III: Importancia de los descriptores

Por otra parte se grafican en la figura 9, a modo de comparación, los valores de importancia arrojados por un árbol con los parámetros óptimos descriptos previamente.
Se entrenó un modelo utilizando los 3 atributos mas relevantes y los mismos hiperparámetros. La performance obtenida fue mucho menor, con un valor de *recall* de 0,4.

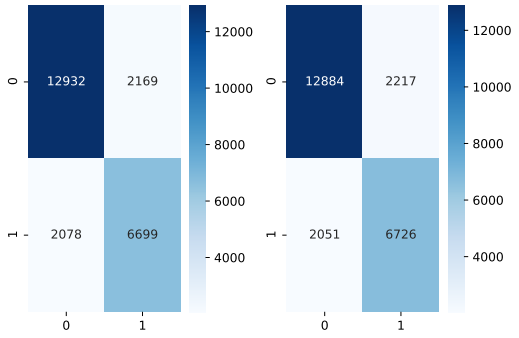

Figura 8: Tabla de contingencia para $\alpha = 0$ y $\alpha = 0{,}000005$

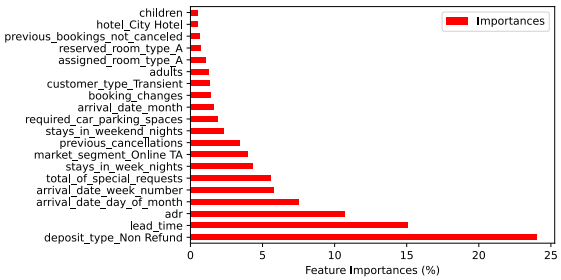

Figura 9: Importancias devueltas por DecisionTreeClassifier

IV-G. Árboles obtenidos

Un resumen de los árboles obtenidos para los mejores parámetros y para los mejores tres descriptores se muestran en las figuras 10 y 11 respectivamente.

V. Conclusiones

En el transcurso de este trabajo se aplicaron conceptos teóricos de árboles de decisión, haciendo uso de las librerías especializadas disponibles en *Python*. Con estas herramientas se logró entrenar un modelo maximizando la métrica de performance elegida en base a la problemática planteada.
El modelo determinado mediante la búsqueda de hiperparámetros arroja un árbol con una profundidad muy alta (33 ramas) que aparenta estar sobreajustado. Esto se evidencia al comparar la performance de los conjuntos de entrenamiento y validación, que muestra diferencias de hasta 20 %. Contrario a lo esperado, la poda del árbol no tuvo mejores resultados. Entre las posibles explicaciones se puede señalar la existencia de registros duplicados en el conjunto de datos. Un posible trabajo futuro es que éstos se mantengan siempre en uno de los dos conjuntos utilizados.
Al entrenar un árbol con los tres descriptores más importantes la performance cayó significativamente. Esto implica que no existe un conjunto acotado de atributos que puedan predecir la variable objetivo a través de un árbol de decisión. Otra evidencia de ello es el análisis de las correlaciones, en donde no fue posible hallar asociaciones fuertes.

Referencias

[1] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
[2] Kaggle. Hotel Bookings. https://www.kaggle.com/jessemostipak/hotel-booking-demand, 2020.
[3] Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

VI. Anexo: Tabla de variables y gráficos de árboles

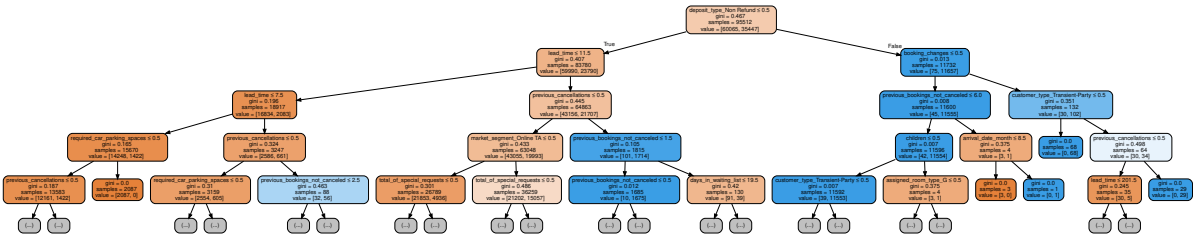

Figura 10: Árbol para los mejores parámetros

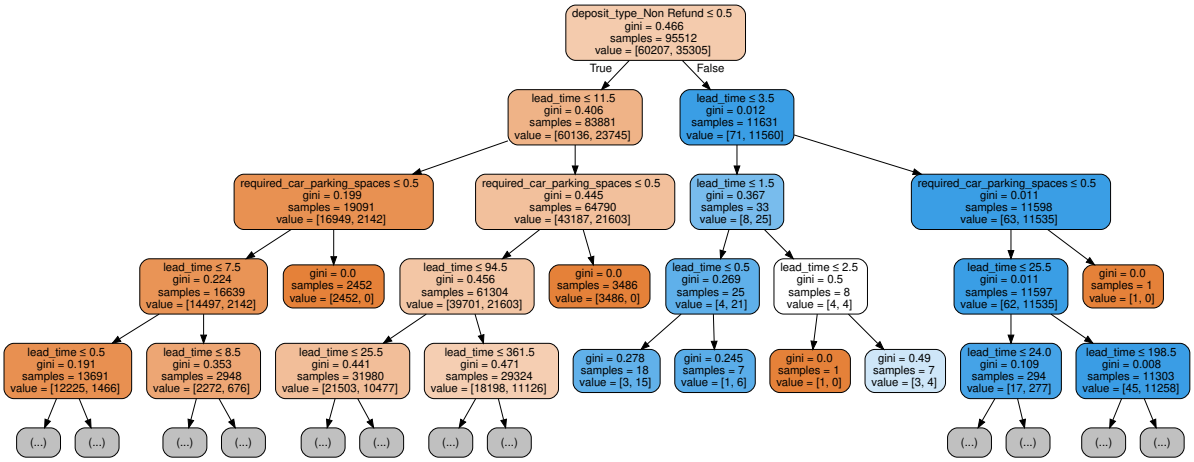

Figura 11: Árbol para los tres descriptores más importantes

Variable	Tipo de dato	Nulos %	Valores Unicos	Valores únicos %
hotel	categórica	0.00 %	2	City Hotel –>0.664461 Resort Hotel –>0.335539
is_canceled	bool	0.00 %	2	0 –>0.629584 1 –>0.370416
lead_time	numerica	0.00 %	479	...
arrival_date_year	numerica	0.00 %	3	2015 –>0.184237 2016 –>0.474973 2017 –>0.340791
arrival_date_month	categórica/ordinal	0.00 %	12	...
arrival_date_week_number	numerica	0.00 %	53	...
arrival_date_day_of_month	numerica	0.00 %	31	...
stays_in_weekend_nights	numerica	0.00 %	17	...
stays_in_week_nights	numerica	0.00 %	35	...
adults	numerica	0.00 %	14	...
children	numerica	∼0	5	...
babies	numerica	0.00 %	5	...
meal	categórica	0.00 %	5	BB –>0.773180 HB –>0.121141 SC –>0.089203 Undefined –>0.009791 FB –>0.006684
country	categórica	0.40 %	117	...
market_segment	categórica	0.00 %	8	Online TA –>0.473046 Offline TA/TO –>0.202856 Groups –>0.165935 Direct –>0.105587 ....
distribution_channel	categórica	0.00 %	5	TA/TO –>0.819750 Direct –>0.122665 Corporate –>0.055926
is_repeated_guest	bool	0.00 %	2	0 –>0.968088 1 –>0.031912
previous_cancellations	numerica	0.00 %	15	0 –>0.945691 1 –>0.050683 ...
previous_bookings_not_canceled	numerica	0.00 %	73	0 –>0.969679 1 –>0.012916 ...
reserved_room_type	categorica	0.00 %	10	A –>0.720278 D –>0.160826 E –>0.054737 ...
assigned_room_type	categorica	0.00 %	12	A –>0.620261 D –>0.212095 E –>0.065382 ...
booking_changes	numerica	0.00 %	21	0 –>0.848597 1 –>0.106382 ...
deposit_type	categorica	0.00 %	3	No Deposit –>0.876464 Non Refund –>0.122179 Refundable –>0.001357
agent	numerica/categorica	13.00 %	333	...
company	numerica/categorica	94.00 %	352	...
days_in_waiting_list	numerica	0.00 %	128	...
customer_type	categorica	0.00 %	4	Transient –>0.750591 Transient-Party –>0.210436 Contract –>0.034140 Group –>0.004833
adr	numerica	0.00 %	8879	...
required_car_parking_spaces	numerica	0.00 %	5	0 –>0.937884 1 –>0.061839 ...
total_of_special_requests	numerica	0.00 %	6	0 –>0.588977 1 –>0.278298 2 –>0.108627 ...
reservation_status	categorica	0.00 %	3	Check-Out –>0.629584 Canceled –>0.360307 No-Show –>0.010110
reservation_status_date	fecha	0.00 %	926	

Cuadro IV: Análisis de variables del set de datos