UNIVERSIDAD LA SALLE CHIHUAHUA

EXPEDIA HOTEL RECOMMENDATIONS

Inteligencia Artificial
6to Semestre
Ingeniería en Tecnologías de la Información y Telecomunicaciones.
Profesor: Normando Ali Zubia Hernández

Victoria Guadalupe Molina Pineda
Matricula:6147

Examen Semestral

Planear las vacaciones de tus sueños, o incluso una escapada de fin de semana, puede ser un asunto abrumador. Con cientos, incluso miles, de hoteles para elegir en cada destino, es difícil saber cuál se adaptará a sus preferencias personales. ¿Deberías ir con un viejo standby con esas pastillas de menta que te gustan, o arriesgarte a un nuevo hotel con un moderno bar en la piscina?

Actualmente, Expedia utiliza parámetros de búsqueda para ajustar sus recomendaciones de hotel, pero no hay suficientes datos específicos del cliente para personalizarlos para cada usuario. En esta competencia, Expedia desafía a Kagglers a contextualizar los datos de los clientes y predecir la probabilidad de que un usuario se aloje en 100 grupos de hoteles diferentes

Column name	**Description**	**Data type**
date_time	Timestamp	string
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia .co.jp, ...)	int
posa_continent	ID of continent associated with site_name	int
user_location_country	The ID of the country the customer is located	int
user_location_region	The ID of the region the customer is located	int
user_location_city	The ID of the city the customer is located	int
orig_destination_distance	Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated	double
user_id	ID of user	int
is_mobile	1 when a user connected from a mobile device, 0 otherwise	tinyint
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int
channel	ID of a marketing channel	int

Column name	Description	Data type
srch_ci	Checkin date	string
srch_co	Checkout date	string
srch_adults_cnt	The number of adults specified in the hotel room	int
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int
srch_rm_cnt	The number of hotel rooms specified in the search	int
srch_destination_id	ID of the destination where the hotel search was performed	int
srch_destination_type_id	Type of destination	int
hotel_continent	Hotel continent	int
hotel_country	Hotel country	int
hotel_market	Hotel market	int
is_booking	1 if a booking, 0 if a click	tinyint
cnt	Numer of similar events in the context of the same user session	bigint
hotel_cluster	ID of a hotel cluster	int

Se pretende trabajar con un conjunto de datos de 37, 670, 293 datos del cual se extraerá 1, 000, 000 con el método de reducción Sample.

Al tener un conjunto de datos tan grande, la laptop no pudo correrlo por lo que un profesor de La Universidad La Salle Chihuahua proporciono lo necesario para trabajar con una Workstation. Si bien esta computadora es más rápida, pero tampoco pudo cargar el conjunto de datos tan grande. Es aquí cuando entra la reducción de datos; fue necesario crear una división en el conjunto de datos para ir reduciéndolo.

No se pudo obtener un chunck de un 1, 000,000 porque la computadora no lo corría, la siguiente decisión por tomar fue hacer dos chuncks de 500,000 cada uno para luego convertirlo en archivo .csv y unirlos. Después de todo este procedimiento la computadora continuó sin correr el conjunto de datos por lo que se decidió trabajar con un conjunto de datos de 500,000 datos solamente.

Una vez más el conjunto de datos fue muy pesado para el equipo a la hora de trabajar con él, al momento de llegar a los modelos de entrenamiento y aunque se corrieran uno por uno la computadora no los pudo.

La decisión final ha sido trabajar con un conjunto de 300,000 datos que si bien también se atora en los modelos, pero eso es debido a la cantidad de categorías o divisiones que se encuentran dentro del target del proyecto.

Datos vacíos

Para la realización de esta parte del proyecto se tienen dos opciones: trabajar con la media de los datos para trabajar con los datos vacíos o trabajar con el target y la dimensión con datos vacíos.

Al iniciar el proyecto se realizó un análisis de datos vacíos, de esta forma se llegó al porcentaje de datos vacío que contiene el conjunto de datos.

EL proyecto cuenta con un 1.44% de datos vacíos en total. Contenidos todos en la dimensión: *orig_destination_distance.*

Luego se aplicó el relleno de datos vacíos en cuanto a la media de los datos y el porcentaje disminuyó a 0.010191082802547772%

Después se aplicó relleno por target de proyecto y el porcentaje es menos del normal y mayor al relleno por media. El porcentaje quedó de 0.7506634819532908%

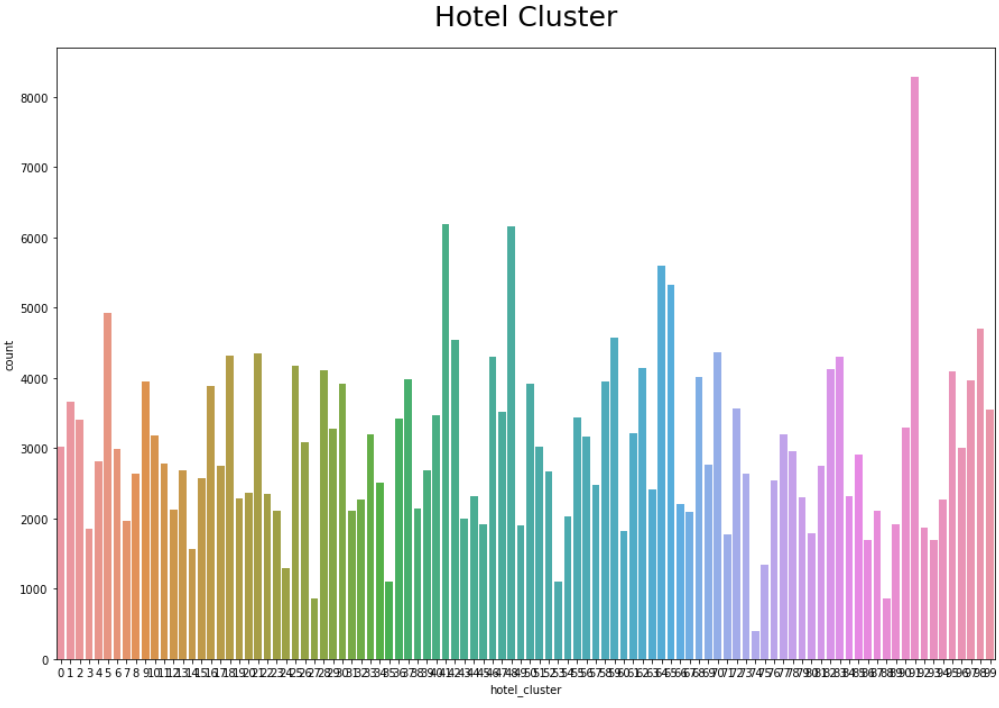
Análisis de dimensiones

Al tener resuelto el asunto de los datos vacíos se procede al análisis por características. En este caso contamos con 24 de las cuales 3 son en formato de fecha y hora, por lo que es difícil trabajar con estas. Lo que se realizó para trabajar de forma amigable con dos de estas tres características fue una separación de datos, de esta forma obtuve tres nuevas columnas por característica (Año, mes y día). Para el análisis de estas fueron utilizadas sólo las columnas de *Mes* y *Día*.

Iniciamos analizando el grafico destinado al target. El target del proyecto es llamado *hotel_cluster* haciendo referencia a los distintos grupos hoteleros con los que cuenta *Expedia*.

El target cuenta con un rango de 1 a 100 en cuanto a grupos hoteleros, es por ello que la grafica podrá verse un poco saturada.

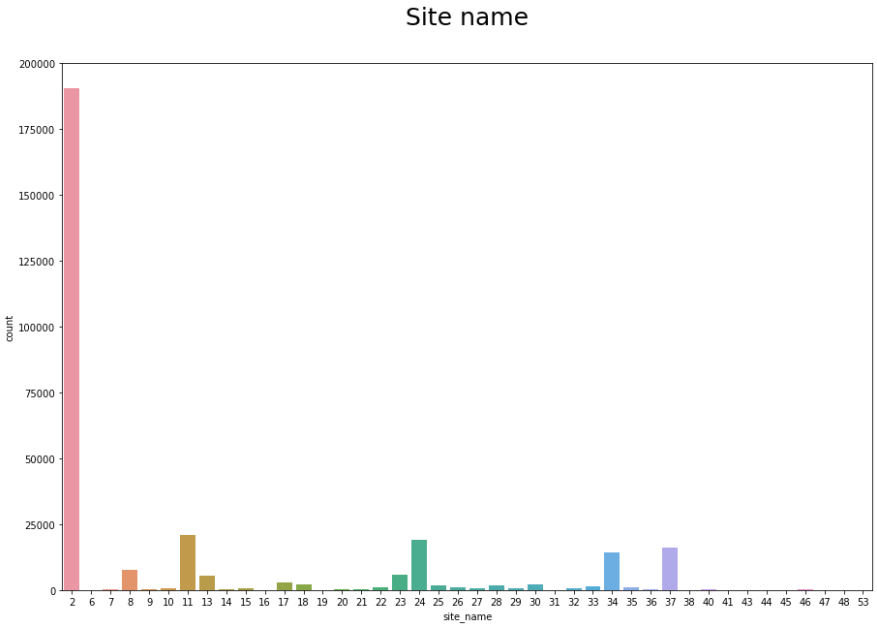
Aún saturada puede apreciarse que se tienen 100 grupos hoteleros con los cuales contar a la hora de tomar una decisión.



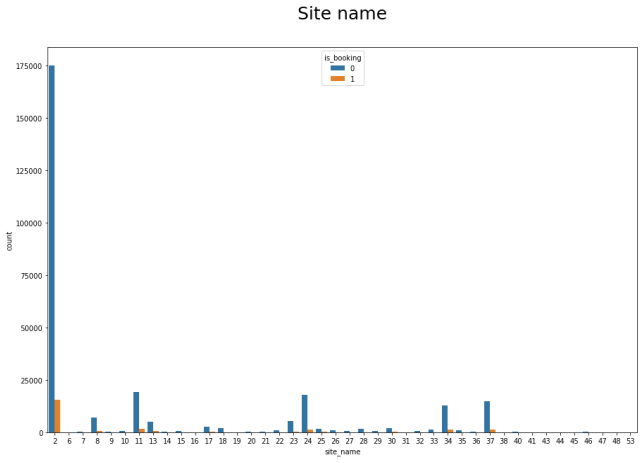
La característica siguiente dentro de este proyecto es *date_time* no pudo ser graficada por su formato de fecha y hora juntos. Además de no ser tan importante para el análisis de este proyecto, esta característica brinda la fecha y hora en que un usuario entra a la página de Expedia para ver que proporciona.

La característica *site_name* es el identificador del punto de compra de Expedia (es decir, Expedia.com, Expedia.co.uk, Expedia.co.jp, ...). No se especifica cuáles son los sitios con los que contamos en esta característica, pero es fácil apreciar que el sitio “2” es el que contiene gran cantidad de ventas registradas.

Dentro de los que reflejan popularidad están los sitios denominados con los números: 11, 24, 37, 34, 23, 8.



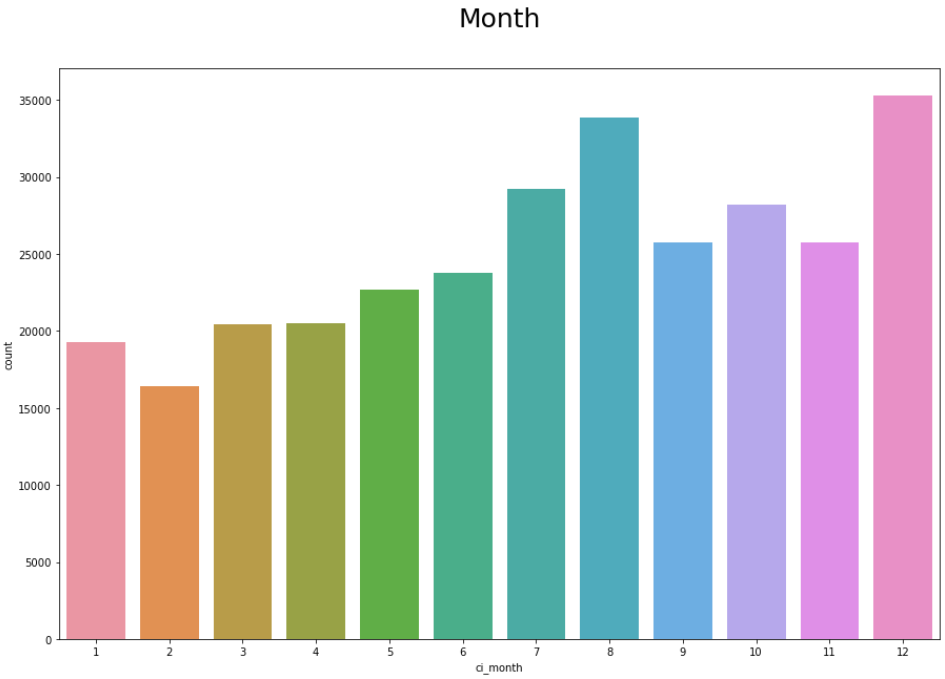
Fue realizada otra grafica de esta característica comparándola con la característica *is_booking* para detectar si existe una relación que sea de ayuda.



is_booking es una característica que contiene dos valores: 1 se reservo y 0 no se reservó. Con esto podremos apreciar de mejor forma si las reservaciones hechas fueron concretadas en compra por uno de estos sitios que proporciona Expedia.

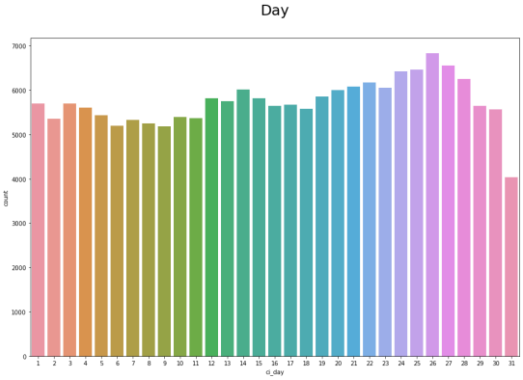
Esta grafica nos muestra que en el sitio más popular, el sitio designado por “2” es el que nos muestra más ventas concretadas y gracias a la característica de *is_booking* es fácil saber ahora que no todas las ventas son reservadas con antelación. Se cuenta con más registros de ventas sin reservación previa.

Luego de esta contamos con la característica *srch_ci* que es la característica que nos muestra la fecha de registro de la persona que concretó la compra. La decisión de separar los datos de esta característica fue con el fin de poder apreciar en que fechas (meses) se registran más reservan dentro de Expedia.

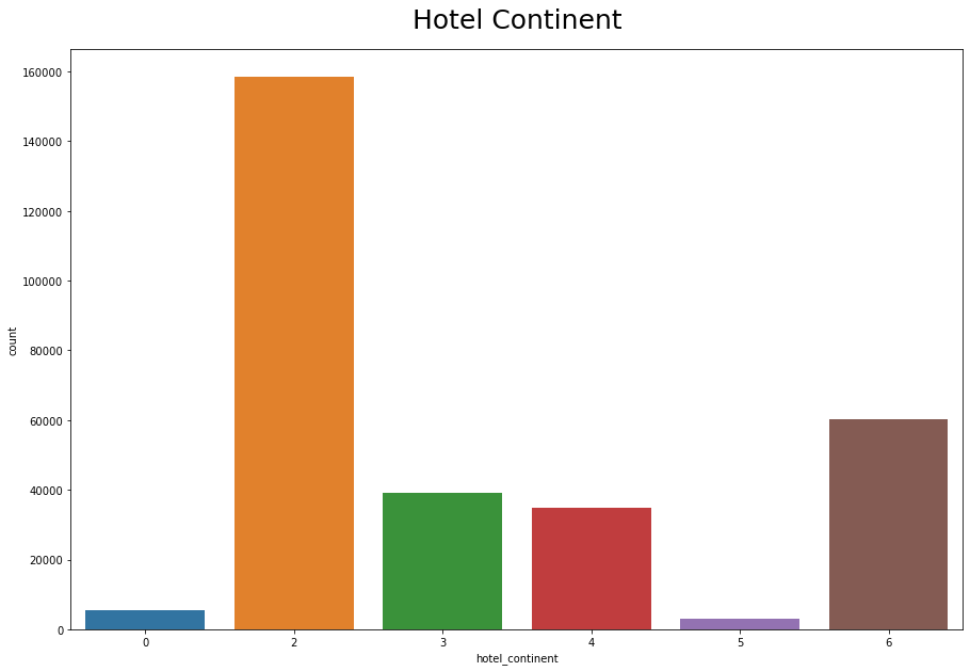


Llegamos a la conclusión que todo el mundo sabe, es más común tener reservaciones en los meses de verano e invierno por el tema de las vacaciones.

También fue realizada una gráfica para ver qué días del mes son los más frecuentes a la hora de reservar.

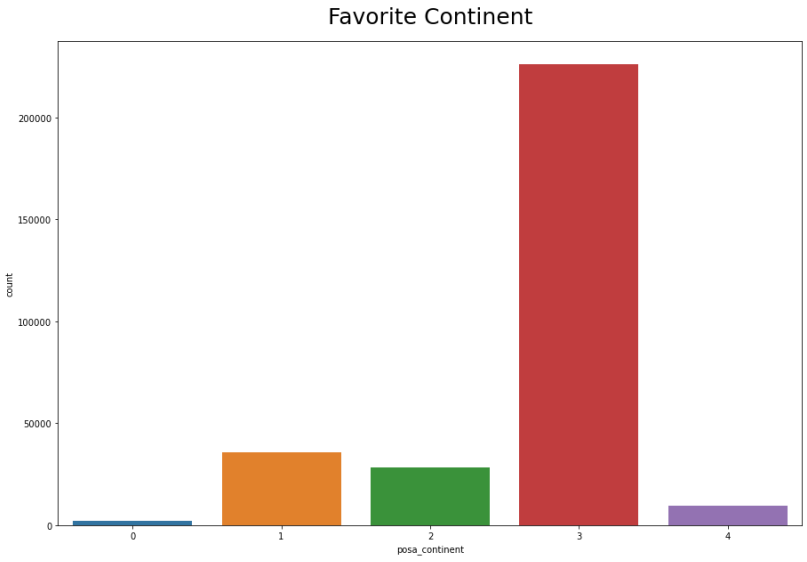


Al graficar esta característica *hote_continent* nos topamos con seis barras en lugar de cinco y esto es porque en la columna de numero 0 se encuentran todos aquellos hoteles que dentro de sus datos no cuentan con el campo de continente así que no han sido asignados.



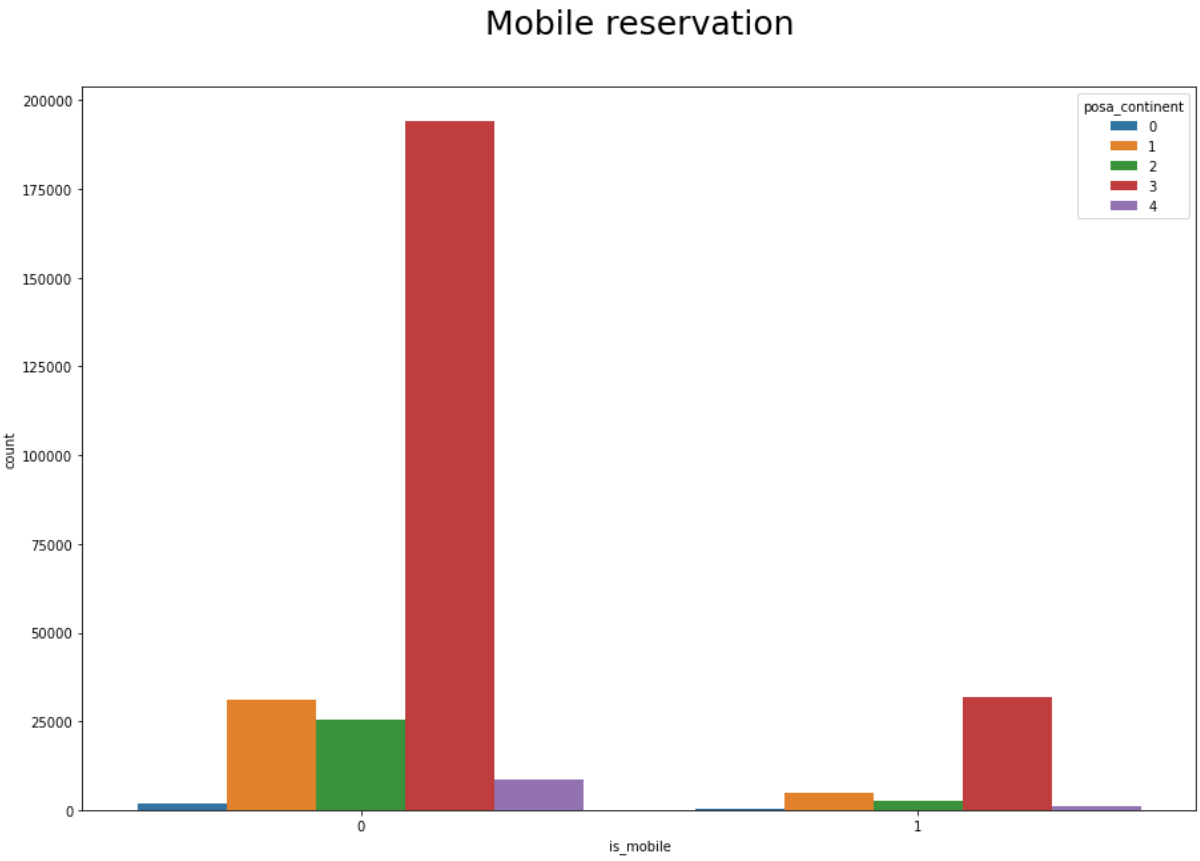
Los continentes dos y seis son los de mayor cantidad de registros, sin embargo no se nos ha proporcionado cual es el nombre del continente para cada número.

Después tenemos la característica *posa_continent* que va de la mano con esta. Esta característica nos proporciona en que continente se registraron más ventas concretadas por *site_name*.



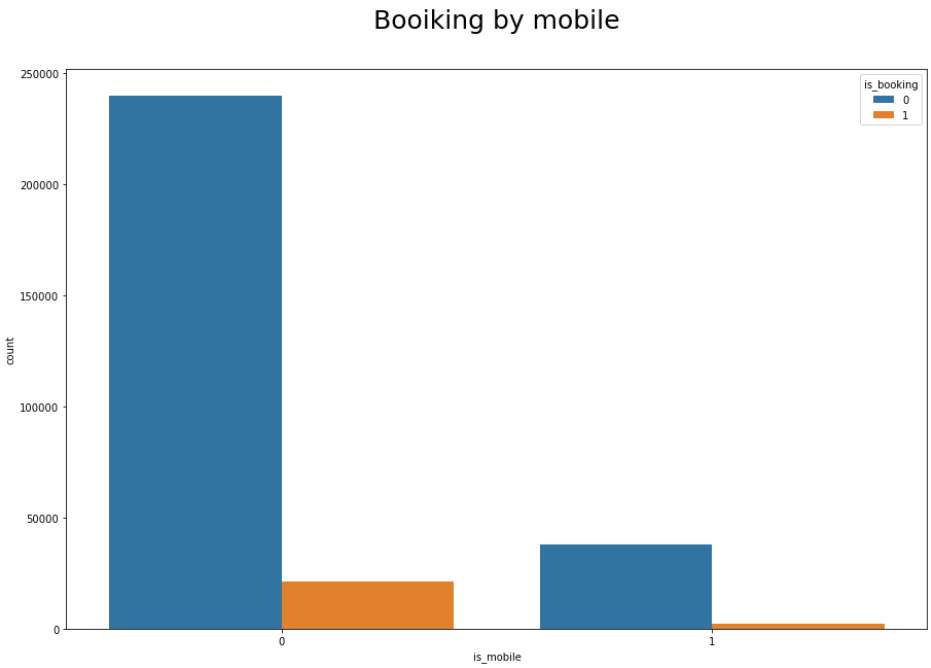
Con esta grafica vemos que en la grafica pasada todos los registro pudieron haber sido reservaciones, pero no ventas concretas para el continente “2”. Sin embargo el continente “3” contaba con menos cantidad, pero fueron ventas realizadasa. En este caso se toma como el continente favorito por los compradores.

La característica de *is_mobile* nos indica con un 1 si el cliente se conectó desde un dispositivo móvil y con un 0 de ser lo contrario. En esta grafica comparamos por continentes cuantos clientes se conectan desde su móvil.

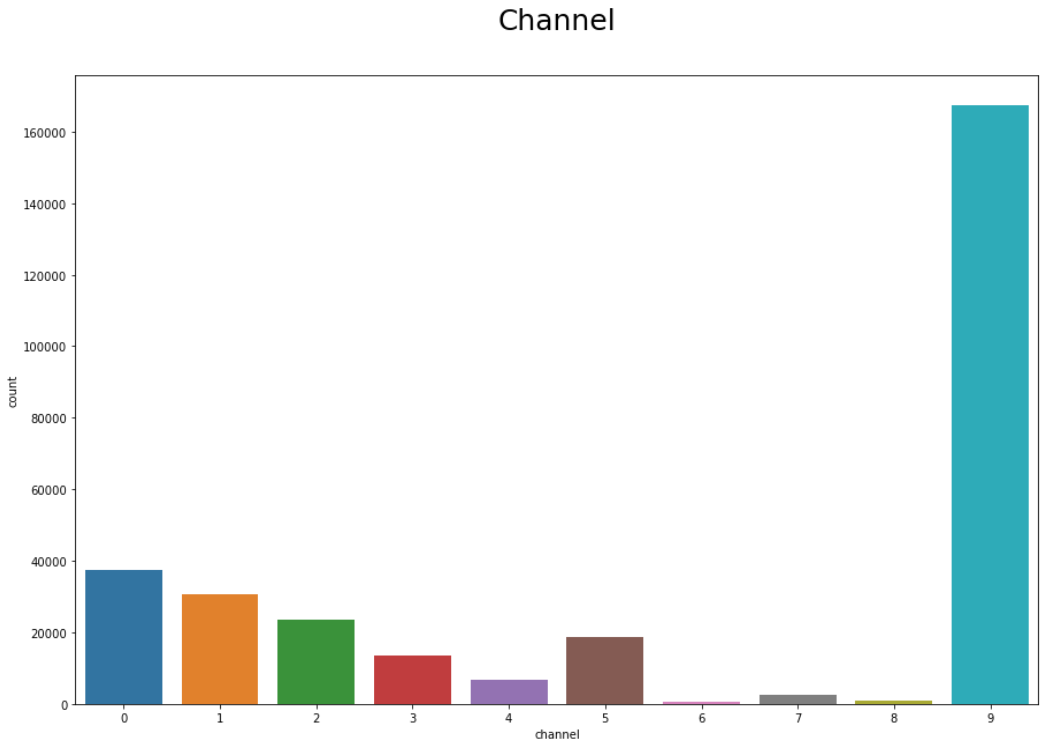


Los datos que arrojó esta grafica es que la mayoría de los clientes no se conecta desde un dispositivo móvil y se muestra que por todos los continentes gana el 0.

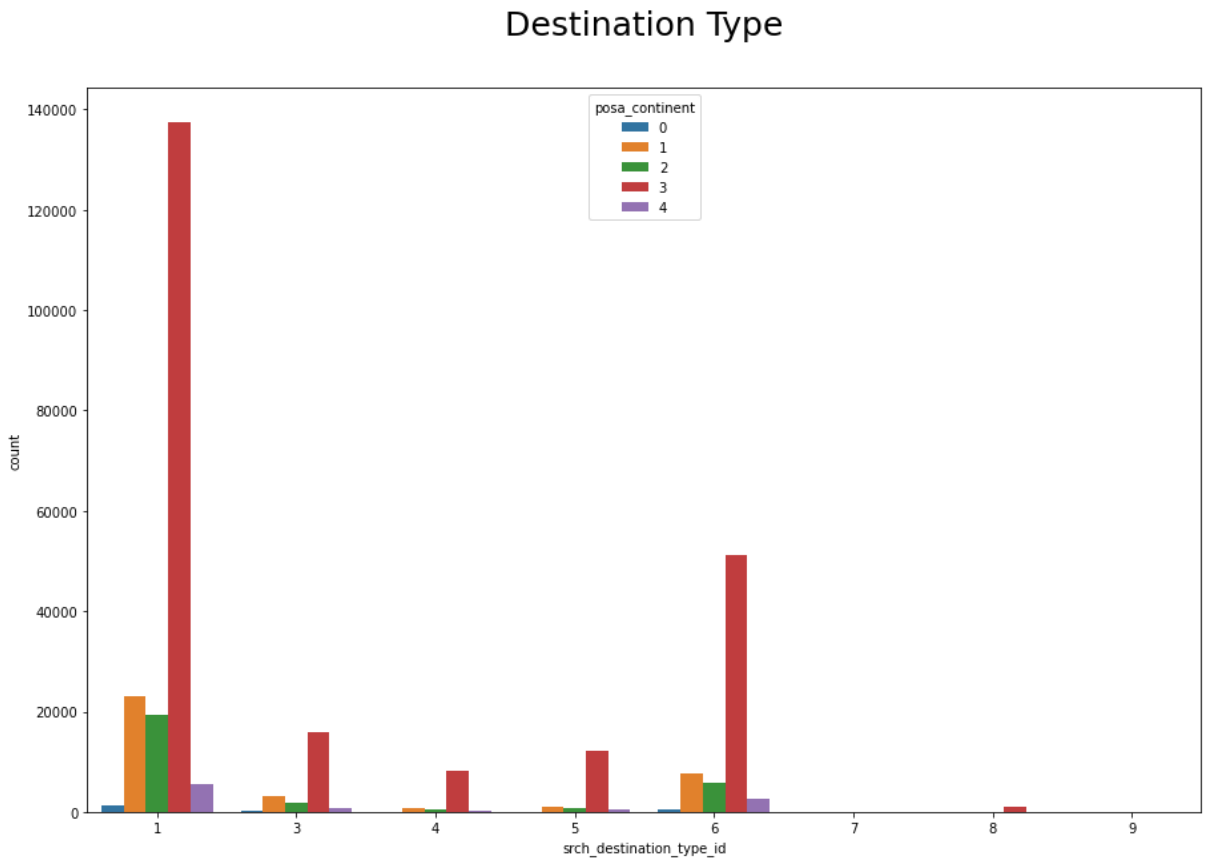
Las características de *is_mobile* y *is_booking* fueron unidas en esta grafica para ver si las reservaciones eran hechas desde un dispositivo móvil y fue lo contrario. Son muchos los registros que no fueron reservados por un móvil.



Una de las características que tiene este conjunto es cannel, la cual nos muestra el canal de mercadotecnia por el que se da a conocer entre los clientes. No se especifica en ningún lugar que tipo de canal o el nombre, pero es muy evidente que el mejor canal de mercadotecnia con el que cuenta Expedia es el número 9.

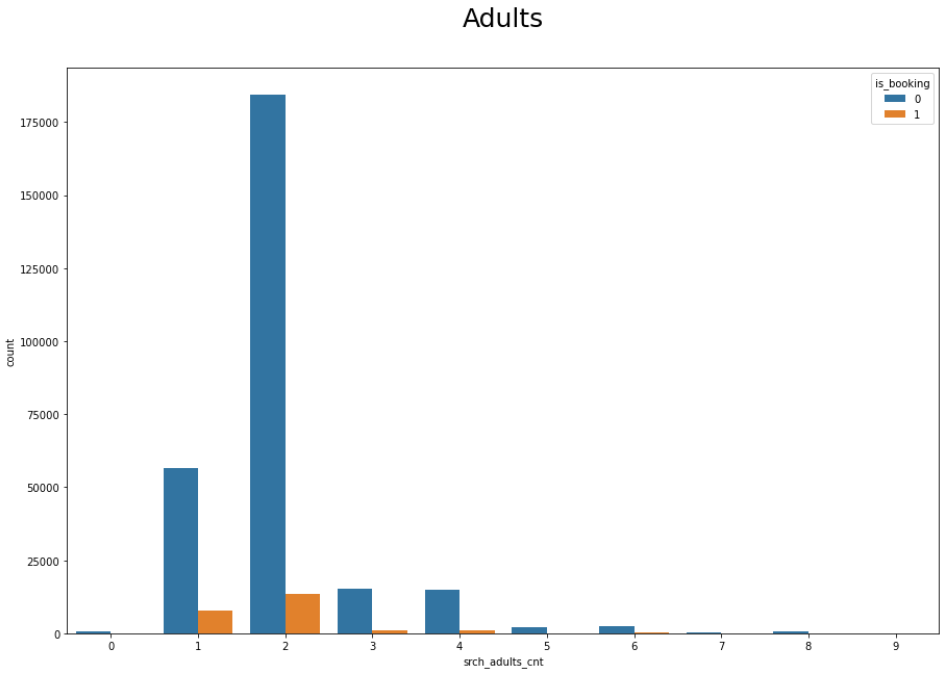


Al igual que en casos pasados, en esta grafica tampoco nos dan los tipos de destinos que maneja Expedia, pero si nos los dan por número. Lo que se hizo en esta grafica fue comparar el tipo de destino con *posa_continent* que es la característica que está relacionada directamente con *site_name* que nos dice si la compra fue concretada o no.

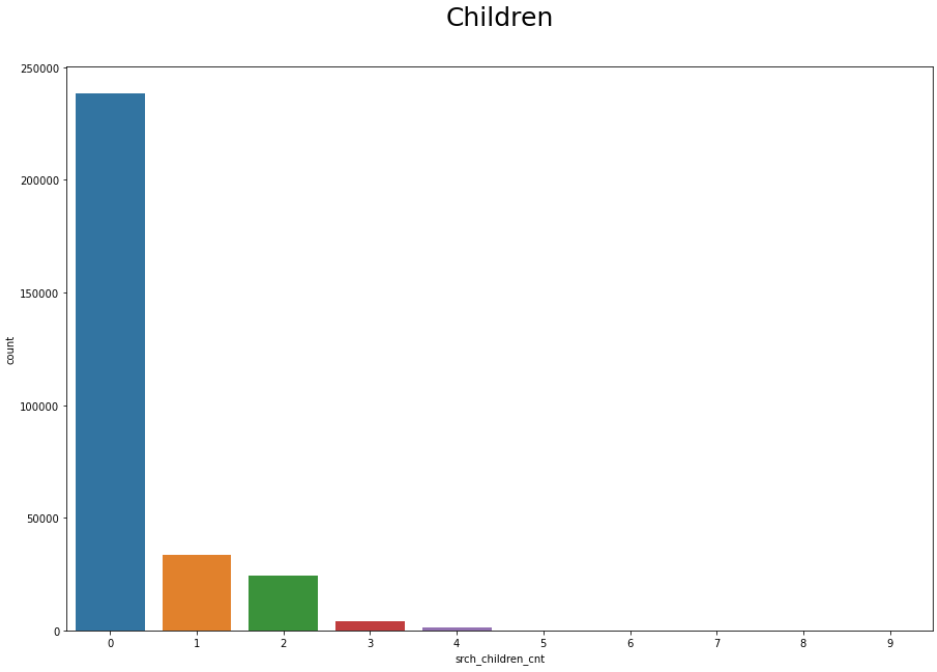


Podemos notar que el destino 1 es el más visitado por todos los continentes, exclusivamente por el continente favorito, el número 3. Seguido de este está el destino número 6 que también tiene suficientes visitas.

Como ocupación extra, Expedia registra cada adulto o niño además de las personas confirmadas. Debido a esto se realizaron dos graficas para ver que tan común es que los clientes paguen por un cupo extra ya sea para un adulto como para un pequeño.



Son muy pocos los que reservan con cupo extra y suelen ser de uno a dos adultos extra.

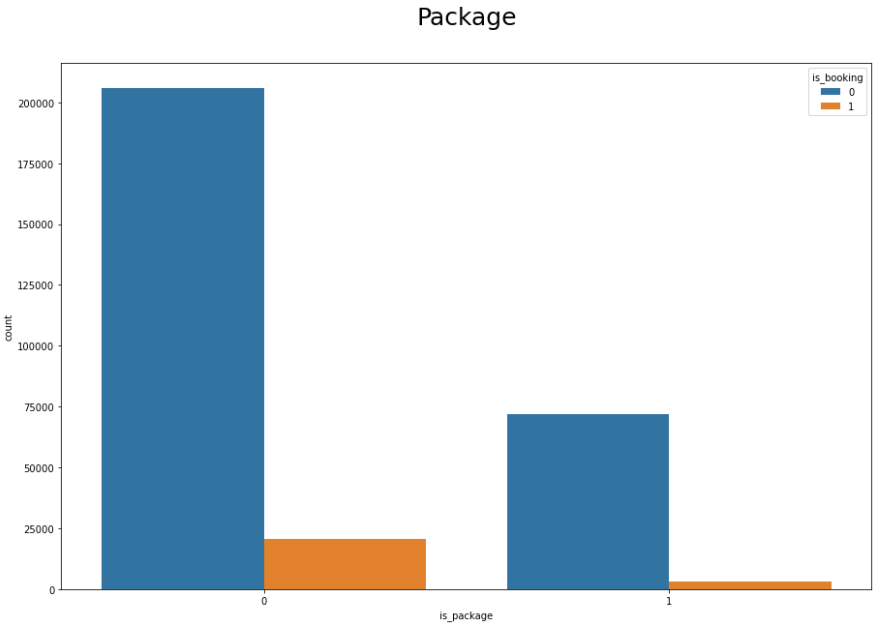


Respecto a los niños es común que varíe de uno a tres niños, el porcentaje va en escalerita.

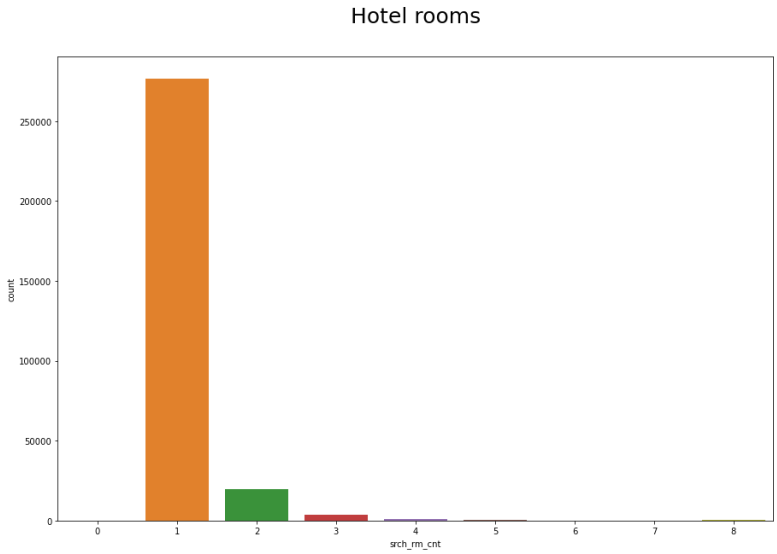
Como es común, hay personas que al reservar unos días en un hotel lo hacen en un paquete que tenga los vuelos o algo más. Esta característica nos muestra esa información; 1 es un paquete y 0 no lo es.

Se está comparando con *is_booking* para visualizar si fueron más los registros reservados en paquete o no.

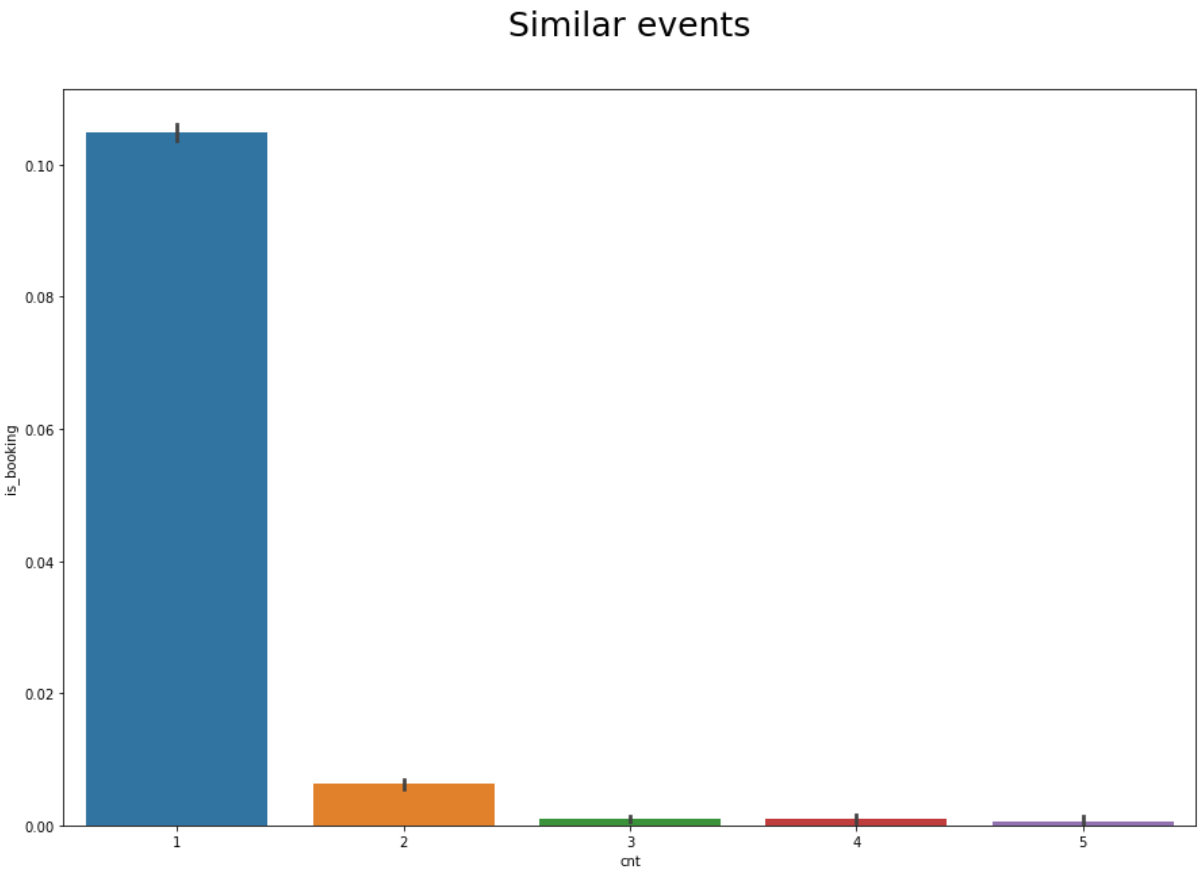
Son más los registrados como no reservados en paquete.



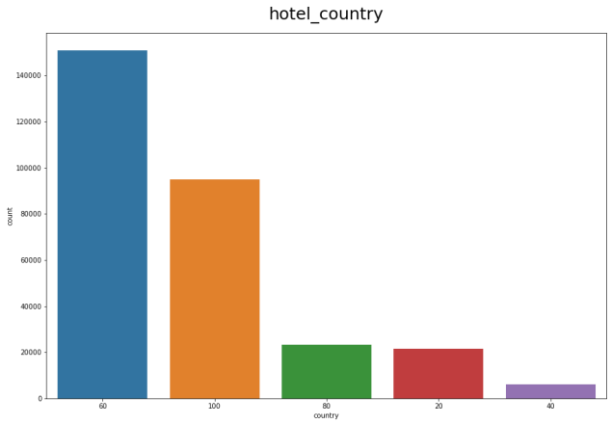
En la siguiente grafica se nos muestra la cantidad de habitaciones que reservan por venta. Es bastante común se pague sólo una a varias.



Se cuenta con una característica que nos muestra el número de eventos similares en el contexto de la misma sesión de usuario. No se sabe cuáles sean estos eventos similares, pero se tienen localizados por número en la gráfica.



El evento similar más reservado es el número 1 ganando por gran cantidad a los otros 4 eventos que se tienen.

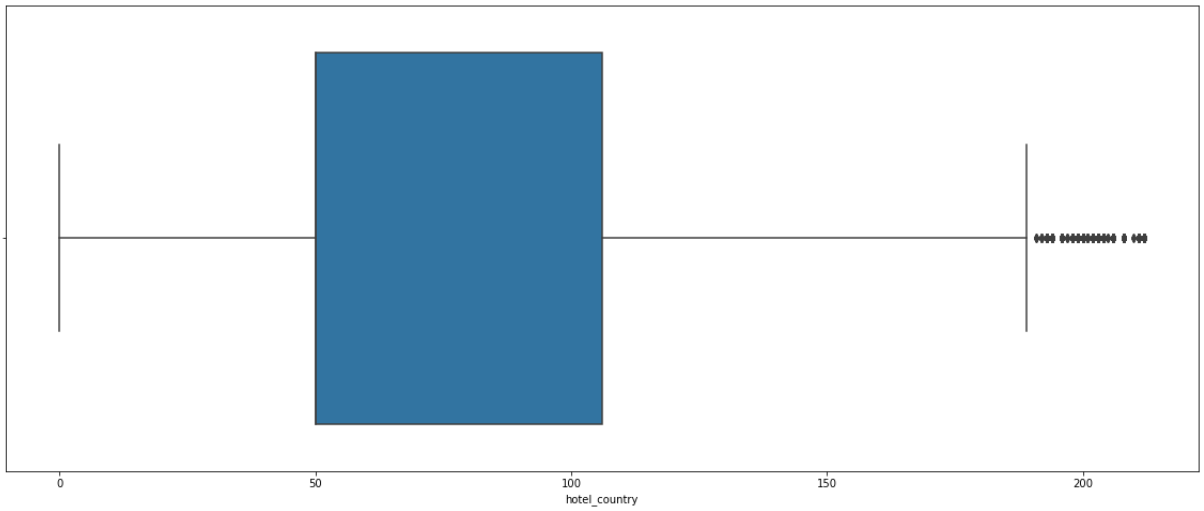


La característica de *hotel_country* fue dividida en cinco rangos para visualizarla de mejor manera. Esta nos da el id del país en que se encuentran los hoteles con los que cuenta Expedia.

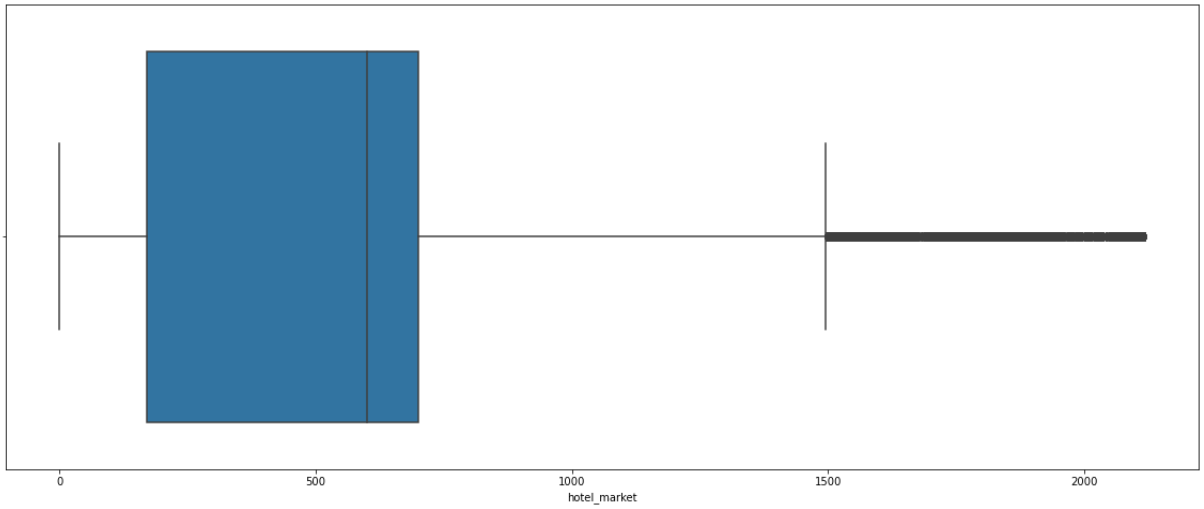
Outliers

Para identificar los datos outliers que se tenían en el conjunto de datos se fueron checando característica por característica, tomando en cuenta que no son tantas.

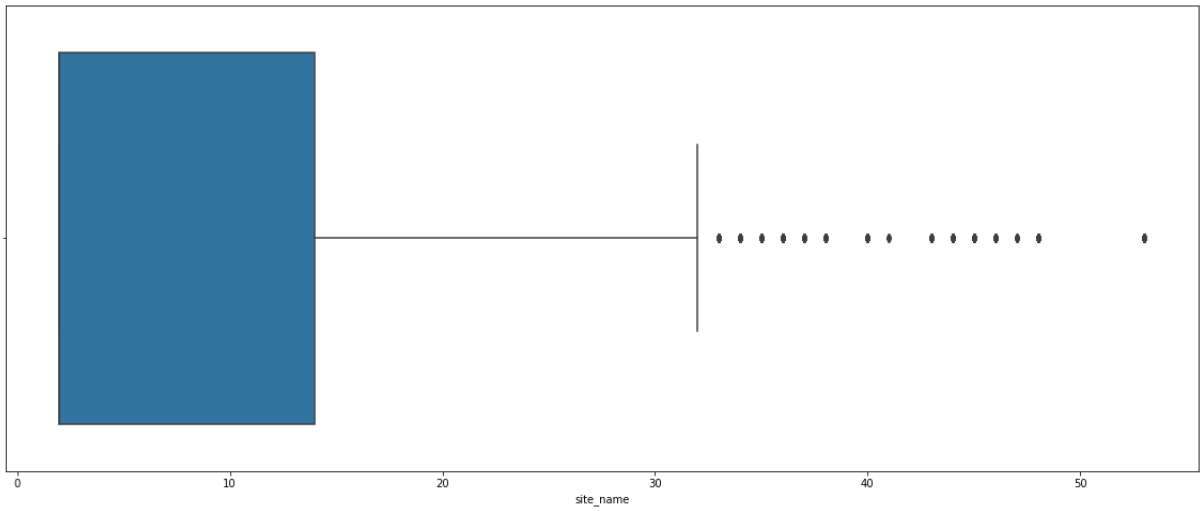
La primera en la que se notaron datos outliers fue en *hotel_country* y en este caso se decidió borrarlo porque eran pocos y no dejaban leer bien los datos que arrojaba dicha característica.



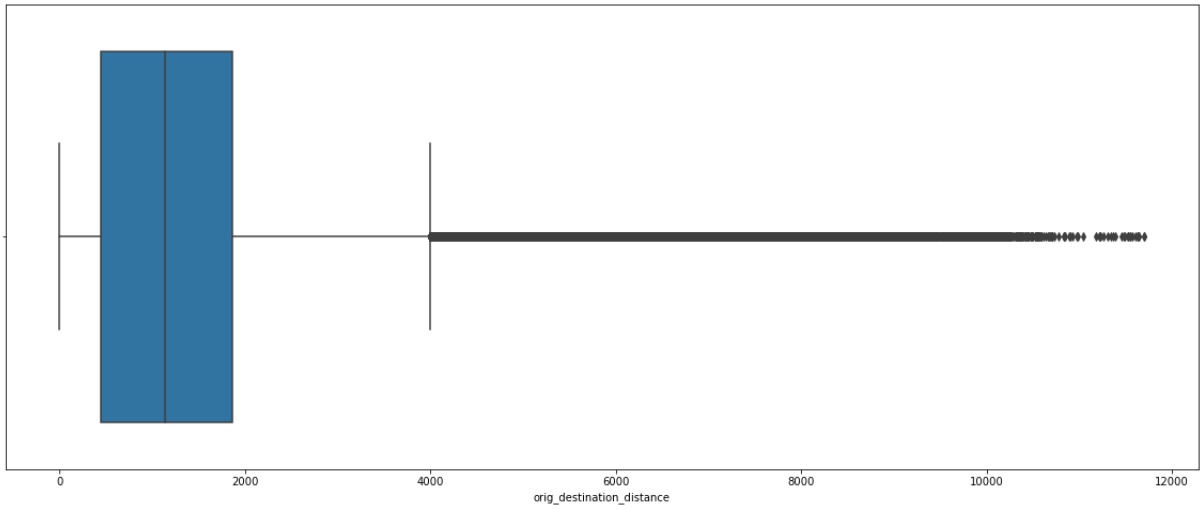
Esta característica muestra el mercado hotelero y al tener tantos id's es imposible poder tener una lectura correcta de los datos y más por estos outliers. También fueron eliminados.



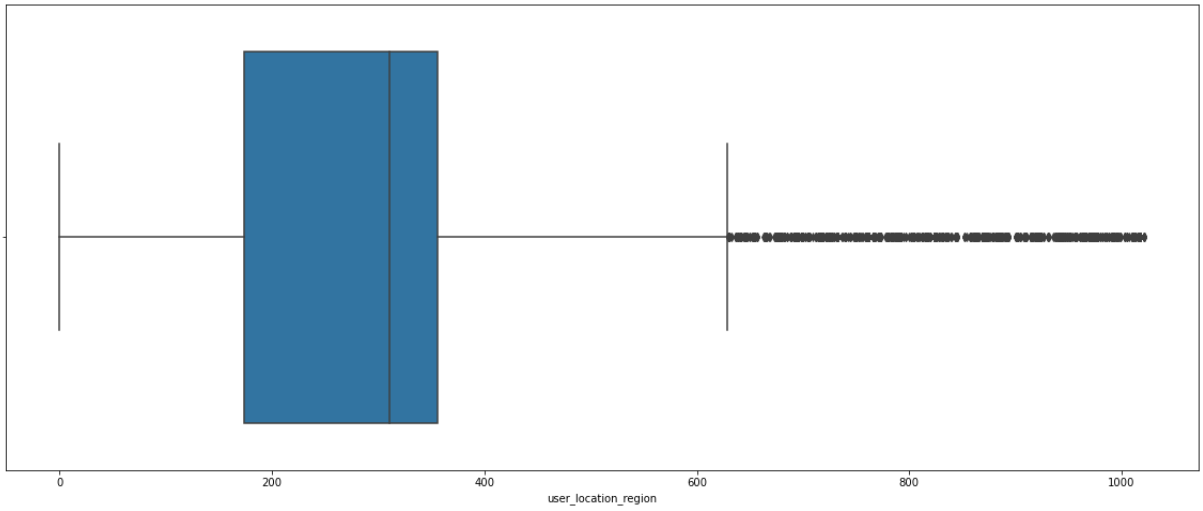
En site_name existían muy pocos outliers y que si influían al ver los datos así que fueron eliminados.



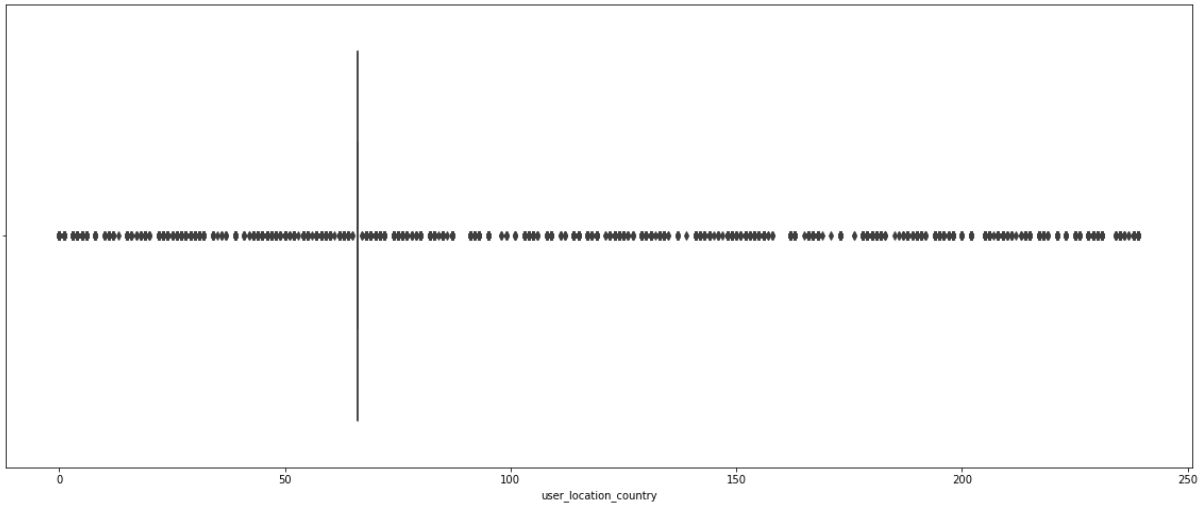
En esta característica aparecen muchos outliers, el dilema es que no son outliers. Al tratarse de distancias es muy fluctuante. Sin un rango límite por medio de Expedia es imposible controlar esto. Inclusive esta característica destaca por tener datos vacíos y es porque hubo hoteles de los cuales no se guardó el dato de distancia. Se probo eliminándolos en una iteración y en otra no.



Esta característica muestra la región del comprador y al tener tantos id’s es imposible poder tener una lectura correcta de los datos y más por estos outliers, y fueron eliminados.



En esta característica son tantos id’s que están totalmente esparcidos y por eso mismo se ve esta forma la grafica. Estos no fueron eliminados y se dejaron así.



Antes de la eliminación de los datos Outliers se contaba con un conjunto de datos de 301440 datos y luego de la eliminación el conjunto cuenta solamente con 244833.

Reducción de dimensionalidad

La reducción de dimensionalidad es otra de las grandes aplicaciones de los algoritmos. El objetivo de este tipo de algoritmo es convertir un dataset de una cierta dimensión en otro con una menor dimensión. Los motivos para desear realizar esta reducción son variados:

- Puede eliminar ruido presente en el dataset original
- Los resultados pueden ser más fácilmente interpretables

Se utilizarán*: PCA(0.90)* y *Attribute subset selection with trees*.

Al realizar las primeras iteraciones se presentó un aspecto que lo cambió todo. El hecho que el target con el que estemos trabajando tenga 99 clases distintas hizo que el entrenamiento fuera difícil y complicado de entender para el algoritmo. Salieron iteraciones donde el score más alto fue un 0.13%

Otro aspecto fue que al comenzar con la reducción de dimensionalidad no se pudo ejecutar *Attribute subset selection with trees* por falta de memoria.

Iniciando con las iteraciones

En esta parte incluí el método Gaussian Naive Bayes (GaussianNB). Puede realizar actualizaciones de los parámetros del modelo. Fue elegido por que al investigar se obtuvo como respuesta que se puede usar para abordar problemas de clasificación a gran escala para los cuales el conjunto de entrenamiento completo podría no encajar en la memoria.

Primera iteración:

Es esta iteración se trabajó con:

- Eliminación de datos vacíos por media.
- Eliminación de outliers.
- PCA(tomó una dimensión).
- Z-Score
- Modelo *Bagging Classifier*

En esta primera iteración se trabajó con estos aspectos porque creí que era el camino acertado. Al eliminar datos y rellenar datos vacíos estuve trabajando con un conjunto completo y parejo. PCA fue mi única opción por problemas de memoria y Z-Score es el mejor método de normalización.

El mejor modelo fue *Bagging Classifier* con un 0.13%. Resultó un porcentage pobre, es un algoritmo que no serviría de nada. Kappa Statistic y F1 Measure lo comprueban.

####################---Bagging Classifier---####################
Model Score: %s 0.13890151084857436
Confusion Matrix:
[[153 3 0 ... 0 0 0]
[0 615 0 ... 0 0 0]
[0 0 10 ... 10 5 0]
...
[9 0 2 ... 50 12 1]
[8 0 0 ... 17 125 0]
[5 2 0 ... 10 2 10]]

Evaluation report

accuracy			0.14	35212
macro avg	0.14	0.11	0.09	35212
weighted avg	0.13	0.14	0.09	35212

Kappa Statistic: 0.11697453656311041
Cross Validation: 0.13890151084857436

Segunda iteración:

Es esta iteración se trabajó con:

- Eliminación de outliers.
- PCA(tomó una dimensión).
- Min-Max
  Modelo *Random Forest Classifier*

####################---Random Forest Classifier---####################
Model Score: %s 0.18069748949221856
Confusion Matrix:
[[80 6 0 ... 0 1 0]
 [0 613 0 ... 0 0 0]
 [0 0 5 ... 10 1 0]
 ...
 [3 0 1 ... 50 9 0]
 [1 1 1 ... 22 63 0]
 [2 1 1 ... 3 2 1]]
Evaluation report

accuracy			0.16	35212
macro avg	0.12	0.08	0.06	35212
weighted avg	0.11	0.12	0.06	35212

Kappa Statistic: 0.09255464878481556
Cross Validation: 0.17069748949221856

En esta segunda iteración se trabajó sin rellenar los datos vacíos y con la normalización Min-Max para probar una nueva combinación de herramientas.

Al igual que la iteración pasada el score es muy pobre y este algoritmo no sería de mucha ayuda. F1-Measure está por debajo del porcentaje que se toma como bueno.

Para las siguientes iteraciones se tomó la decisión de generar rangos en el target para poder trabajar de una forma más sencilla a la hora de entrenar el algoritmo. Los rangos son 5 en los que están divididos las cantidades para tener un equilibrio.

Tercera iteración:

Es esta iteración se trabajó con:

- Rellenado de datos vacíos por media.
- Eliminación de outliers.
- PCA(tomó una dimensión).
- Z-Score
- Modelo *Bagging Classifier*

####################---Bagging Classifier---####################

Model Score: %s 0.46221428442568235
Confusion Matrix:

[[116 5 5283 439 21]
[24 22 4299 589 16]
[108 9 21571 2269 80]
[58 13 9879 3771 59]
[15 2 6126 516 141]]

Evaluation report

accuracy			0.46	5543
macro avg	0.44	0.24	0.21	55431
weighted avg	0.45	0.46	0.36	55431

Kappa Statistic: 0.09717897660651731
Cross Validation: 0.46221428442568235

Se puede notar lo mucho que ayudó el implementar rangos dentro del target, pero aún así se puede ver que el algoritmo no es muy fiable. Es muy pobre y el f1-score sigue por debajo de lo deseado.

Cuarta iteración:

Es esta iteración se trabajó con:

- Rellenado de datos vacíos por media.
- Eliminación de outliers.
- PCA(tomó una dimensión).
- Min-Max
- Modelo Random Forest Classifier

####################---Random Forest Classifier---####################

Model Score: %s 0.45583864118895967
Confusion Matrix:

[[63 1 7428 427 0]
[7 1 5981 717 8]
[13 0 29081 2930 6]
[4 0 15122 5152 10]
[1 0 7812 541 55]]
Evaluation report

accuracy			0.46	75360
macro avg	0.58	0.24	0.19	75360
weighted avg	0.53	0.46	0.35	75360

Kappa Statistic: 0.08679145934786403
Cross Validation: 0.45583864118895967

Si bien esta combinación de herramientas no mejoró el algoritmo, sino al contrario, bajó el porcentaje y sigue por debajo del requerimiento de f1-score. En esta iteración se cambio el tipo de normalización utilizada.

Quinta iteración:

Es esta iteración se trabajó con:

- Rellenado de datos vacíos por media.
- PCA(tomó una dimensión).
- Min-Max
- Modelo Random Forest Classifier

####################---Random Forest Classifier---####################

Model Score: %s 0.4578954352441614
Confusion Matrix:

[[64 0 7374 423 0]
[18 0 5995 717 6]
[15 0 29255 2839 5]
[10 1 15218 5132 6]
[2 0 7710 514 56]]
Evaluation report

accuracy			0.46	75360
macro avg	0.47	0.24	0.19	75360
weighted avg	0.48	0.46	0.35	75360

Kappa Statistic: 0.08820101895300558
Cross Validation: 0.4578954352441614

En esta iteración dejé de lado la eliminación de datos outliers para ver que tanto cambiaba o ayudaba al algoritmo, el resultado fue similar a los pasados y f1-score nos lo comprueba.

Se realizaron más de 30 iteraciones, muchas de estas fallidas a causa de la falta de memoria del equipo y otras no brindaban nada al algoritmo. En estas cinco iteraciones fueron de gran ayuda Kappa Statistic, Cross Validation y F-Measure para comprobar en cada una de estas que el algoritmo era muy pobre.

Conclusión.

Este proyecto requirió de muchas horas de trabajo, mucha investigación y mucha prueba y error.
Fue muy difícil trabajar con un conjunto de datos tan grande ya que los equipos no lo soportaban, además de tantas trabas que esto ocasionó en el proceso de análisis de tablas hasta el momento de entrenar los modelos.

Otro aspecto que paraba mucho el trabajo era el target que a mi parecer estaba mal establecido, el grupo hotelero no lo veo tan importante a la hora de recomendar el siguiente destino. Sería mejor que el *site_name* que es la característica que dice si la compra fue concretada o no, incluso *is_booking* que te dice si el cliente reservó una habitación. Muchas de las dimensiones están muy lejos de poder ayudar al objetivo del proyecto.

Además, leyendo comentarios en la competencia en Expedia me di cuenta que mis procedimientos no estaban mal ya que a la mayoría de los participantes les corría de 0.3% a 0.18% score. Incluso el resultado final que arroja Expedia es de 0.18584% utilizando *Random Forest Classifier*.

La mejor combinación de este trabajo con el conocimiento previo sería la de la tercera iteración con 0.46% de score y con *Bagging Classifier* como modelo.
Tomando en cuenta la opinion de Expedia y siguiendo este proyecto la major combinación sería la segunda iteración con un score de 0.18% con *Random Forest Classifier*.