

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



**SISTEMA DE RECOMENDACIÓN DE HOTELES
SIMILARES**

TESIS

QUE PARA OBTENER EL TÍTULO DE
MAESTRO EN CIENCIA DE DATOS

PRESENTA

FELIPE GERARD VALDÉS

ASESOR: [Si me dejan] NADIA DEL VILLAR

MÉXICO, D.F.

2015

“Con fundamento en el artículo 21 y 27 de la Ley Federal del Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada “**SISTEMA DE RECOMENDACIÓN DE HOTELES SIMILARES**”, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la Biblioteca Raúl Bailleres Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una contraprestación”

Felipe Gerard Valdés

FECHA

FIRMA

Índice general

Prólogo	IV
1. Análisis previo	5
1.1. Sistemas actuales	5
1.2. Diagnóstico	11
2. Sistema de recomendación de hoteles	13
2.1. Modelo propuesto	13
2.2. Precio	14
2.3. Distancia: filtro estático	16
2.4. Similitud de hoteles	17
2.5. Distancia: filtro dinámico	28
2.6. Precios dinámicos	30
2.7. Consideraciones prácticas	31
2.8. Juntando todo	31
2.9. Comentarios	32
3. Implementación	33
3.1. Información	33
3.2. Sistema de respaldo	35
4. Desempeño	37
5. Conclusión	38
5.1. Trabajo futuro	38
5.2. Comentarios finales	38

Índice de figuras

2.1.	Ejemplos de similitud de servicios para 4 categorías distintas. El hotel verde se parece al negro porque no le faltan muchos servicios que el negro sí tiene. En contraste, al hotel morado le faltan muchos servicios.	21
2.2.	Ejemplo univariado. Distancia entre x y y usando valor absoluto (- -) y pérdida <i>hinge</i> (—). La pérdida <i>hinge</i> no penaliza el caso en que $y > x$	22
2.3.	Ejemplos de similitud de perfil para 4 categorías distintas. El hotel verde se parece al negro porque aunque le faltan servicios, tiene aproximadamente la misma proporción de sus servicios en cada categoría. El hotel azul tiene muchos servicios, pero tiene un enfoque distinto (en este caso es un hotel de súper lujo muy enfocado a playa, mientras que el negro no tiene playa).	24
2.4.	La elección de α determina la dirección en la que avanza la similitud total al variar la de servicios y la de perfil.	26
2.5.	El filtro debe ser más laxo cuando hay pocos hoteles similares cercanos (izquierda) y más estricto cuando hay más (derecha).	29

Índice de tablas

Prólogo

Las agencias de viajes han migrado la mayor parte de su negocio a las ventas en línea. Eso implica que junto con sus precios y la calidad de su servicio, la experiencia de usuario en sus páginas web es una parte crucial si quieren conseguir una ventaja competitiva. Dado que la mayor parte de las agencias empezaron antes de la era del Internet, el cambio al mundo digital ha sido paulatino y por necesidad, más que rápido y por gusto. Dado que el negocio consiste en *vender* viajes, la primera prioridad que se tiene al hacer una página es que sea operativa. La segunda prioridad normalmente es que sea agradable a la vista y aceptablemente fácil de usar. Estos dos principios tienen mucho sentido y de hecho los portales web actuales de las agencias de viajes los siguen razonablemente bien. Sin embargo, la era del Internet es mucho más que un punto de venta en cada casa.

Sería imprudente pensar que la experiencia de comprar en casa vía electrónica es exactamente igual que comprar en un quiosco de una agencia. La diferencia clave es el conocimiento que tienen los vendedores, que está ausente en una página web. La razón por la que esto es importante es que los clientes muchas veces no saben bien lo que quieren, y mucho menos lo que hay disponible. Entonces, suponiendo que se tiene resuelta la parte estática de la página (i.e. que sea funcional, utilizable y agradable), la tercera prioridad debería ser sustituir de algún modo al vendedor. Para ello es necesario hacer que la página sea dinámica y que tome ciertas decisiones automáticamente, de modo que pueda ayudarle al cliente a encontrar (y por lo tanto a comprar) la mejor opción disponible.

Usualmente, el primer componente de las páginas web que se busca que sea inteligente es el orden en el que aparecen los hoteles cuando se busca un destino en particular, también llamado *Sort Order*. La razón de esto es simple: dado que se sabe a dónde quiere ir el cliente, hay que darle las mejores opciones. Aquí el término “mejores” es, desde luego, desde el punto de vista

del cliente, pero en última instancia lo que se busca es aumentar las ventas y la rentabilidad de la empresa. Es común que en esta parte haya filtros adicionales, como el rango de precio y la categoría del hotel (es decir, sus estrellas) que permitan al cliente refinar su búsqueda. Si bien esto funciona bastante bien, también es cierto que en un destino con muchas opciones puede seguir siendo demasiado vago y, además, obliga al cliente a ser su propio asesor. La pregunta ahora es qué más se puede hacer para incorporar información y así ayudar a cerrar ventas.

Conocer el destino del viaje del cliente es cercano a lo mínimo que se puede saber sobre él. Sin embargo, cuando se mete a la página de un hotel en particular nos proporciona bastante más información. En particular, nos da una idea de su presupuesto, de la categoría que espera que tenga su hotel y también algo de información geográfica más detallada que sólo una ciudad o zona. A menos que apliquemos un cuestionario, no tenemos otra manera de obtener más datos sobre lo que el cliente quiere, así que a lo más que podemos aspirar es a recomendar hoteles similares al que el cliente vio en primera instancia. La esperanza aquí es que si el cliente casi encontró lo que quería en la primera opción, probablemente le guste algún hotel similar cercano. Si aún así no compra, queremos que siga y siga viendo alternativas hasta que encuentre una lo suficientemente bueno. Un sistema con estas características podría jugar el papel de asesor y, de hecho, los vendedores humanos hacen algo muy similar: escuchan al cliente y proponen una opción, reciben retroalimentación y vuelven a proponer hasta que se cierra una venta o el cliente se retira.

Ahora bien, para que el sistema que hemos delineado tenga éxito, es necesario que tenga varias características. Entraremos en detalles a lo largo de la tesis, pero hay ciertas propiedades básicas que deberá tener si planea sustituir a un vendedor:

- Deberá *pensar lo más parecido a un humano*, es decir, tener un concepto de similitud y cercanía entre dos hoteles lo más parecido posible al de una persona.
- Deberá ser *inteligente y flexible* al recomendar. Es importante que proponga alternativas viables sin caer en repeticiones. Imagine el lector que un vendedor insiste en una opción que el cliente no aprueba.
- Deberá ser *empático y atento*, o también, tomar en cuenta el presupuesto y los cambios que el cliente proponga (por ejemplo, si el cliente

repentinamente está más interesado en hoteles más económicos que antes, se deberán ajustar las recomendaciones).

- Deberá ser *ingenioso* cuando las circunstancias sean difíciles y recomendar siempre algo, aún cuando haya poco de dónde escoger.

Aunque el lector podría pensar que los adjetivos utilizados en los puntos anteriores son ambiciosos, en realidad corresponden a funciones muy básicas y perfectamente evaluables en el contexto que nos interesa. Esperamos que al final de esta lectura lo habremos convencido de que nuestro sistema en efecto es capaz de sustituir razonablemente bien a un vendedor.

En el presente trabajo desarrollaremos paso a paso un *modelo de recomendación de hoteles similares* con la filosofía descrita anteriormente, hecha a la medida especialmente para las necesidades de la empresa Best Day Travel (de aquí en adelante sólo “Best Day”). La inquietud que tuvo Best Day por mejorar la sección de “Hoteles Similares” en su página fue la que llevó a la concepción de este proyecto, por lo que el objetivo siempre fue no sólo mejorar con respecto a lo que se tenía, sino también con respecto a la competencia. Debido a la naturaleza tan aplicada del proyecto, incluimos un capítulo entero dedicado a temas técnicos relacionados con la implementación particular en el sistema actual de Best Day. A continuación describimos brevemente la estructura de la tesis.

En el primer capítulo describiremos el *status quo* tanto de la competencia como de Best Day con el fin de aprender de lo que ya hay y así poder mejorarlo. Buena parte del proceso de aprendizaje provino de esta parte.

En el segundo capítulo describiremos a detalle el modelo que construimos, incluyendo las propiedades que deberá tener para que se comporte como deseamos y también qué medidas tomamos para solucionar cada una de las dificultades.

El tercer capítulo tiene información técnica necesaria para la implementación del modelo en el escenario real de Best Day. Incluye el tratamiento de la información y las acciones a tomar en presencia de datos de mala calidad o faltantes. En pocas palabras, es la liga entre el capítulo 2 y el mundo real.

En el capítulo cuatro hablaremos del desempeño del modelo, desde dos puntos de vista. El primero es desde el punto de vista del negocio, es decir cómo se comporta la página antes y después de implementar el modelo propuesto. El segundo es desde el punto de vista del cliente, es decir si las recomendaciones tienen sentido y por lo tanto si hemos logrado hacer el

trabajo un asesor humano. En esta segunda parte también incluiremos un análisis de importancia de los hoteles, que resulta naturalmente del modelo que elegimos.

Finalmente, concluiremos en el quinto capítulo, donde discutiremos algunos puntos importantes sobre el sistema y sus implicaciones y algunas ideas para trabajo futuro.

Capítulo 1

Análisis previo

Antes de desarrollar el modelo de recomendación de hoteles similares, es conveniente analizar el *status quo*. Comenzaremos con un análisis de la competencia del que trataremos de sacar información valiosa de qué hay o no hay en el mercado y tomar ideas, posteriormente haremos un análisis del sistema de Best Day que pretendemos mejorar con el fin de entender en dónde hay áreas de oportunidad y finalmente sacaremos conclusiones de lo aprendido.

1.1. Sistemas actuales

En esta sección analizaremos la manera en que la competencia de Best Day recomienda hoteles y luego veremos qué se hace en Best Day. Los ámbitos que nos interesan son

1. La posición dentro de la página de las recomendaciones.
2. La forma en la que se eligen las recomendaciones.

Enfatizaremos en los principales competidores, de Best Day, es decir: Booking, Price Travel, Expedia y Despegar. También veremos si podemos aprender algo de TripAdvisor, que aunque no es un competidor directo, vive de recomendar hoteles y de su comunidad de comentarios. Las empresas competidoras están ordenadas de menor a mayor relevancia del sistema de recomendación que tienen.

Price Travel (pricetravel.com)

Pusimos a Price Travel en primer lugar en la lista porque no tienen un sistema de recomendación de hoteles similares. Cuando se busca un destino, se presentan los hoteles en el Sort Order determinado por su sistema, pero ya que se entra a la página de un hotel en particular (independientemente de cómo se llegó a esa página), no se recomienda hoteles similares ni cercanos. De hecho, la única indicación de otros hoteles es en la pestaña de mapa, en la que se pueden ver todos los cercanos en un mapa interactivo. Dado que no aprendimos nada de este competidor, procederemos al siguiente.

Booking (booking.com)

Booking es un competidor muy importante de Best Day, aunque también tiene un mercado más grande y mayor cantidad de opciones en destinos fuera de Latinoamérica. Antes de entrar en los puntos de interés general, cabe mencionar que algo que hace booking que no hacen los demás es que cuando se busca un hotel en la página con la barra inteligente, en lugar de ir directamente a su página, se muestra en el primer lugar de una lista, dando la posibilidad de elegir algún otro hotel. Ahora sí pasemos al tema en cuestión:

■ Hoteles populares por destino

1. Se encuentran del lado izquierdo, abajo de la barra de búsqueda y fechas. Es una posición de prominencia media-baja, puesto que no es visible inmediatamente sin bajar en la página.
2. Cuando se entra a la página (no al hotel en particular) por primera vez y se selecciona un hotel, aparece una sección de hoteles populares en el mismo destino del hotel buscado. El tema es que siempre aparece la misma selección para cada destino, independientemente del hotel que se haya elegido, pues la popularidad de los otros hoteles no tiene relación con el hotel elegido.

■ Búsquedas recientes

1. Se encuentran en el mismo lugar que los hoteles populares por destino, pero los sustituyen una vez que se ha visitado al menos un hotel anteriormente.

2. La información de los hoteles visitados se va almacenando en una *cookie*, de modo que para los hoteles subsecuentes, en lugar de los hoteles populares por destino, aparece un listado de unos pocos hoteles recientemente buscados por el usuario.

Al parecer en Booking sólo están interesados en lo que el cliente ha buscado y utilizan los hoteles populares por destino para no dejar vacía la sección para los clientes nuevos. El problema con los hoteles populares por destino es que no varían según el hotel buscado (siempre y cuando estén en el mismo destino), mientras que el de las búsquedas recientes es que no promueven que el cliente explore la página y encuentre la mejor opción para sus necesidades. Entraremos en detalles más adelante.

Despegar (despegar.com)

Despegar es un competidor más directo y con un mercado más parecido al de Best Day que Booking, aunque está más enfocado en vender boletos de avión que cuartos de hotel. En este caso hay más de una estrategia y en más de un lugar de la página de cada hotel, por lo que las describiremos por separado. Cabe mencionar que no usan todas las estrategias para todos los hoteles, por lo que sospechamos que tienen información incompleta y sólo muestran lo que tienen disponible.

■ Búsquedas recientes

1. Se encuentra en la izquierda abajo del mapa. Igual que en booking.com, se tiene que bajar en la página para verlo, por lo que su relevancia es media-baja.
2. Son los hoteles buscados recientemente por el usuario. Curiosamente, no siempre nos aparecieron al hacer búsquedas, incluso después de haber visto varios hoteles.

■ Hoteles cercanos

1. Se encuentra directamente debajo de las búsquedas recientes.
2. Se listan los hoteles más cercanos al hotel en buscado, sin ningún criterio de precio, categoría o estilo del hotel.

■ Hoteles populares por destino

1. Se encuentra directamente abajo de los hoteles cercanos.
2. Para cada destino muestra una lista de los hoteles más populares (presumiblemente los más vendidos). La lista es la misma para todos los hoteles en un destino dado y de hecho contiene al hotel buscado en su posición correspondiente (es decir ni se omite ni aparece hasta arriba).

■ **Otros usuarios también buscaron...**

1. Aparece en la parte inferior de la página, abajo de prácticamente todo. Su relevancia es muy baja y de hecho se omitió en los primeros análisis de la competencia porque no lo vimos.
2. En la parte inferior de la página también aparece una sección de hoteles que fueron visitados por los demás usuarios que han visto el hotel en cuestión. Este análisis no es complejo, pero puede ser pesado computacionalmente si no se utilizan métodos apropiados de *Market Basket Analysis* como el algoritmo *A Priori*, por ejemplo.

En Despegar tuvieron muchas ideas para recomendar, pero todos los criterios están separados y no colaboran entre sí. Es como tener una tabla ordenada por varios campos distintos. Los hoteles recomendados con base en lo que han visto otros clientes creemos que están desperdiciados en donde están, puesto que sólo los clientes que vayan hasta abajo de la página los verán. El sistema funciona razonablemente bien y permite movilidad hacia arriba en la categoría del hotel. El problema es que es claramente afectado por la popularidad de los hoteles, de modo que tiende a atorarse en aglomerados de hoteles populares y también a subir de categoría. El resultado es que después de algunos pocos *clicks*, casi invariablemente se llega a los hoteles más caros de la zona, sin posibilidad de explorar otros.

Expedia (expedia.com)

En Expedia se le da una prioridad bajísima a las recomendaciones, puesto que están posicionadas en un lugar casi imposible de ver. Sin embargo, tienen un sistema diferente a los demás y por eso lo pusimos en esta posición en la lista de sistemas competidores.

■ **Búsquedas recientes**

1. Se encuentran en el centro, casi hasta abajo, después de toda la información del hotel. La prioridad es muy baja.
2. Es igual que en las demás páginas. Aparecen los últimos cuatro hoteles buscados distintos del actual (en caso de repetir una búsqueda).

■ Hoteles recomendados

1. Se encuentran directamente abajo de las búsquedas recientes.
2. Utilizan un método desconocido para recomendar hoteles similares que permite movilidad progresiva entre categorías. No estamos seguros si tiene alguna restricción de precios.

El sistema de recomendación de Expedia nos gustó bastante y de hecho creemos que está siendo desaprovechado. Permite explorar los hoteles disponibles hacia arriba y hacia abajo en la escala de precio y categoría y llegar a opciones completamente nuevas. Es probable que esté acotado por destino, aunque eso no es tan grave. Algo que no nos gustó tanto es que tal vez es algo rígido y después de buscar un poco se puede atorar. A diferencia del algoritmo de Booking, esto no necesariamente lleva a los hoteles más caros de la zona.

Trip Advisor (tripadvisor.com)

Como decíamos arriba, Trip Advisor no es la competencia directa de Best Day, pero es un sitio muy popular de evaluación de hoteles y creemos que podemos aprender de ellos.

■ Hoteles similares

1. Se encuentran en la derecha, abajo de la foto principal, pero al lado de otra información importante, por lo que el impacto es medio.
2. No sabemos cómo hacen las recomendaciones, pero no son por distancia estricta y sí permiten cierta movilidad de la categoría.

El sistema de TripAdvisor también funciona razonablemente bien, pero parece estar más basado en la calificación de los usuarios que en la categoría del hotel. Si bien ese enfoque tiene sentido en el contexto de su página, Best Day no tiene información tan detallada como ellos en cuanto a lo que opinan

los usuarios y además el enfoque es distinto. Algo que tampoco nos gustó es que el sistema es bastante rígido y se puede quedar atorado fácilmente en ciertos grupos de hoteles.

Best Day (bestday.com.mx)

Finalmente, analizaremos lo que se hace actualmente en Best Day. Del mismo modo que con la competencia, comenzaremos con una breve descripción:

■ Hoteles similares

1. Los hoteles similares están abajo de la liga al mapa del lado derecho. Como en la mayoría de los competidores, no se ve inmediatamente al entrar en la página, por lo que tiene prominencia media-baja.
2. La manera de recomendar hoteles es por categoría (estrellas) y por destino, es decir, se toman los hoteles de las mismas estrellas y en el mismo destino que el hotel buscado y se presentan en el Sort Order del momento. No se toma en cuenta el precio ni la distancia (salvo por el hecho de encontrarse en el mismo destino) de las opciones.

■ Búsquedas recientes

1. Se encuentran directamente debajo de los hoteles similares.
2. Similarmente a la competencia, aparece una lista de los hoteles que el usuario visitó recientemente. Las búsquedas recientes permanecen entre sesiones dentro de una misma computadora.

Al parecer las búsquedas recientes son (casi) un estándar en la industria, por lo que es confortante saber que Best Day es competitiva en este ámbito. Por otro lado, la calidad de las recomendaciones es muy variable. En zonas homogéneas y con mucha oferta para una categoría dada, las recomendaciones pueden ser razonablemente buenas. Sin embargo, en destinos donde hay hoteles muy distintos pero con la misma categoría, como podrían ser un hotel citadino y uno de playa, las recomendaciones dejan mucho que desear. Además, si no hay o casi no hay hoteles bajo este criterio, no se muestra nada, o casi nada. Más adelante expondremos estos temas con más detalle.

1.2. Diagnóstico

En los siguientes párrafos resumiremos lo que aprendimos del análisis de la competencia y del estado actual de Best Day. En primer lugar, nos dimos cuenta de que poner una sección con las búsquedas recientes es algo muy deseable y relativamente fácil de implementar. De hecho, es un estándar en la industria, pues todos los competidores lo tienen, a excepción de Price Travel. Sin embargo, recomendar hoteles que el cliente ya vio no cuenta realmente como recomendaciones y debería manejarse en un plano separado. Posiblemente se podría utilizar información de los hoteles visitados en las recomendaciones, pero por ahora proponemos dejar intacta esta sección.

Quitando las búsquedas recientes de en medio, podemos analizar los diversos sistemas de recomendación. Un punto que pensamos que es sumamente importante es el precio. Hasta donde pudimos ver, ninguna página tiene miramiento alguno con el costo de los hoteles recomendados. El resultado es que pueden recomendar hoteles del doble o incluso triple de precio que el original. Creemos que esto no promueve que el cliente explore más alternativas, ya que la mayoría basa su decisión fuertemente en el precio de los hoteles. Propondremos una forma simple de lidiar con este problema sin perder flexibilidad en las recomendaciones.

Otro factor que nos pareció mejorable es el uso de la información geográfica. Parece ser que la opción más popular es únicamente utilizar la información incluida en el destino y, de hecho, parece que sólo Despegar tiene una sección de hoteles más cercanos. Si bien utilizar el destino como punto de partida es una opción razonable, creemos que se podría aprovechar mejor la información si se utilizan las coordenadas geográficas, dado que en la vida real lo que realmente importa es la proximidad de los hoteles; los destinos son particiones más o menos arbitrarias que no deberían afectar las recomendaciones. Por ejemplo, se puede recomendar un hotel en Cancún dado que se buscó un hotel en la zona norte de la Riviera Maya, o uno en Playa del Carmen a una persona que buscó algo en Cozumel.

Un tercer problema con el que nos encontramos en las recomendaciones es que en la mayoría de los casos eran estáticas. Es decir, son idénticas para ciertos grupos de hoteles. Esto pasa en todas las opciones que vimos que utilizan el destino (Booking, Despegar, Best Day) como criterio, pues no dependen también del hotel. Esto en sí mismo puede ser un problema en destinos amplios, además de que el usuario agota las recomendaciones rápidamente, volviéndolas inútiles.

En cuarto lugar, nos dimos cuenta que en varias de las páginas se han dado cuenta de que quieren recomendar hoteles vía varios criterios, pero muchas veces no los integran. La estrategia tiende a ser usar un criterio para ordenar a la vez, en lugar de aprovechar varios simultáneamente. Esto contribuye al tercer punto, pues hay poco cambio en las recomendaciones de hotel a hotel.

En quinto lugar, notamos que en general los sistemas que sí recomendaban de forma más sofisticada resultaban rígidos y/o se podían quedar atorados fácilmente en grupos aislados de hoteles. Esto hace que el sistema se vuelva inútil una vez que se llegó a uno de estos grupos absorbentes. El único que nos pareció decente en este sentido es el de Expedia, aunque sí es algo más rígido de lo que nos gustaría. Otro punto sobre los sistemas en general, pero de los de hoteles similares en particular, es que en ninguna página (salvo tal vez Trip Advisor) aparecen en posiciones de importancia alta. En el mejor de los casos aparecen una página abajo de la principal, al lado, y en los peores casos están hasta abajo. Creemos que están desaprovechando mucho el potencial de dichos sistemas.

Finalmente, un problema particular del sistema de Best Day es que no toma en cuenta de ninguna forma el perfil del hotel. Como mencionamos arriba, no es lo mismo un hotel de playa de cuatro estrellas que un hotel de ciudad de cuatro estrellas, y si por alguna razón ambos están en el mismo destino, como puede suceder en ciudades cerca del mar, por ejemplo, el sistema de recomendación por destino y estrellas puede fallar catastróficamente. Tenemos ejemplos de hoteles de negocios en Cancún para los que se recomienda hoteles todo incluido en la zona hotelera. Esto no es viable y claramente es importante que el algoritmo que desarrollemos tome en cuenta el perfil de los hoteles, sobre todo porque busca recomendar hoteles *similares*.

Capítulo 2

Sistema de recomendación de hoteles

En el capítulo anterior analizamos los sistemas de recomendación que tienen actualmente tanto los competidores como Best Day y expusimos los principales puntos débiles que encontramos. Ahora podemos utilizar el conocimiento obtenido para generar un sistema nuevo que tenga un mejor desempeño y propiedades más deseables que los que vimos. Primero describiremos punto por punto qué nos gustaría que hiciera el sistema y posteriormente dedicaremos una sección para profundizar en cómo resolvimos cada uno de los problemas.

2.1. Modelo propuesto

Para resolver los problemas expuestos en el capítulo anterior proponemos las siguientes medidas:

1. Poner una cota superior al **precio** de los hoteles recomendados, como un porcentaje del precio del hotel buscado.
2. Recomendar hoteles razonablemente **cercanos** al buscado, donde el criterio de cercanía varía según la densidad de hoteles en la zona.
3. Las recomendaciones deben tener un **perfil** o filosofía similar al original.
4. Las recomendaciones deben ser hoteles no necesariamente de las mismas estrellas, pero sí deben tener una cantidad similar de **servicios**

disponibles.

Antes de entrar a una descripción detallada de cada medida, explicaremos brevemente cómo colaborarán a que nuestro sistema de recomendación tenga las propiedades que buscamos.

Siempre y cuando dos hoteles sean genuinamente similares a los ojos de una persona, la vasta mayoría de los clientes siempre preferirá el más económico. Acotando el precio esperamos que rara vez aparezcan hoteles más allá de las posibilidades del cliente en las recomendaciones. En cuanto a la distancia entre hoteles, creemos que utilizar la distancia al hotel en cuestión es mucho más razonable que usar el mismo criterio del destino para todos, puesto que promoverá hoteles cercanos además de parecidos en otros sentidos.

Esperaríamos que las similitudes de perfil y servicios trabajen conjuntamente con el criterio de cercanía para obligar a que las recomendaciones varíen de hotel a hotel, incluso dentro de un mismo destino. La ventaja de esto es que se puede explorar muchas opciones sin tener que buscar nada explícitamente, puesto que las recomendaciones abren nuevas alternativas por sí solas. Es importante que demos suficiente flexibilidad al método para que no sea rígido y no se estanque en grupos reducidos de hoteles.

Como último punto, no queremos utilizar estos criterios de forma aislada, pues cada uno tiene diversas faltas. Lo que queremos es integrarlos todos en un sistema poderoso que evite que el cliente tenga que pensar demasiado y que genuinamente le proponga alternativas interesantes. Para ello generaremos una medida de similitud entre los hoteles y mostraremos los más parecidos que estén a una distancia y precio razonables del hotel original. En las siguientes secciones describiremos cada uno de los criterios con mayor nivel de detalle.

Para facilitar la exposición le daremos nombres a los hoteles que van superando cada etapa, hasta llegar a las recomendaciones para cada hotel.

Definición 2.1 (Conjunto de hoteles). *Denotamos el conjunto de todos los hoteles por \mathcal{H} y al número de hoteles por N , es decir $N = |\mathcal{H}|$.*

2.2. Precio

Dado que el precio es un factor determinante en la elección de un hotel para la mayoría de los clientes, es importante respetar sus posibilidades eco-

nómicas a la hora de recomendar nuevas opciones. Es razonable pensar que dados dos hoteles idénticos, un cliente siempre elegirá el más barato, salvo tal vez unos pocos clientes de súper lujo. Por esta razón, decidimos no imponer ninguna cota inferior al precio de los hoteles recomendados, siempre y cuando cumplan los criterios de similitud que describiremos más adelante. Por otro lado, como puede ser desalentador que las sugerencias contengan opciones demasiado caras, optamos por poner una cota superior al precio de los hoteles. Como en este paso queremos reflejar los precios “reales” de los hoteles, utilizaremos su precio de largo plazo para generar el filtro, como se explica a continuación.

R 1 (Filtro de precio de largo plazo). Sea $\gamma \geq 0$ y sea

$$p_i^D = \frac{\text{Importe total del hotel } i \text{ en el periodo } D}{\text{Total de noches del hotel } i \text{ en el periodo } D}$$

el precio de largo plazo del hotel i en el periodo D . Entonces definimos la cota superior de precio para i por

$$p_i^{sup} = (1 + \gamma)p_i^D,$$

es decir, se excluyen del análisis para i todos los hoteles j con $p_j^D > p_i^{sup}$. Los hoteles restantes están contenidos en el conjunto

$$\mathcal{H}_i^1 = \{j \in \mathcal{H} : p_j^D > p_i^{sup}\}$$

El parámetro γ es esencialmente el porcentaje por encima del precio de i que se permitirá que tengan las recomendaciones. Por ejemplo, si $\gamma = 0,3$, entonces se ignoran todas las opciones que sean más de 30 % más caras que i . El periodo D debe ser suficientemente largo para que refleje los precios promedio de un hotel, pero no tanto como para que se vea demasiado afectado por la inflación y otros factores. Una buena elección es por ejemplo un año o dos años, pues así además se toma en cuenta los cambios de precios en las diversas temporadas del año. Cabe notar que en esta versión del modelo, tanto γ como D son los mismos para todo hotel i .

En pruebas empíricas nos gustó el desempeño de γ alrededor de 0.3, ya que permite suficiente flexibilidad en los precios y las categorías sin proponer opciones excesivamente costosas.

2.3. Distancia: filtro estático

Antes de entrar en criterios dinámicos de distancia y en similitudes de hoteles, es importante pensar si su cálculo es factible. El tema es que si bien en teoría sería suficiente calcular las similitudes de los hoteles y luego quedarnos con los cercanos, en la práctica esto es complicado. Cuando se quiere conocer la distancia entre cualesquiera dos hoteles de entre, supongamos, N opciones, es necesario almacenar $(N^2 - N)/2$ números (una de las porciones triangulares de la matriz de distancias, quitando la diagonal). Para N incluso moderadamente grande, la cantidad de información a guardar se vuelve prohibitiva. Para empeorar las cosas, los criterios de similitud que desarrollaremos más adelante no son simétricos, por lo que tendríamos que almacenar toda la matriz (menos la diagonal), lo que resultaría en $N^2 - N$ números.

El problema con lo anterior es que además la gran mayoría de los cálculos se desperdiciarían, ya que en nuestro contexto no tiene mucha utilidad guardar la similitud entre un hotel en Veracruz y otro en Las Vegas. Una manera de resolver este problema es ignorar hoteles que seguro no recomendaremos. Dado que queremos que el sistema regrese hoteles similares y cercanos, es razonable ignorar los que estén más lejos que una cantidad fija suficientemente grande en cualquier destino.

Hay diversas fórmulas para calcular la distancia geodésica entre dos puntos, unas más sofisticadas que otras. Sin embargo, como en este caso estamos interesados en hoteles cercanos, basta con la versión que supone que la Tierra es esférica:

Definición 2.2 (Distancia geodésica). *Definimos la distancia geodésica (en kilómetros) entre dos puntos $x = (x_{long}, x_{lat})$ y $y = (y_{long}, y_{lat})$ por la distancia mínima que hay que recorrer por la superficie terrestre para llegar de uno a otro, suponiendo que la tierra es esférica. Si definimos $R =$ radio promedio de la Tierra $\approx 6371km$, $\Delta_{long} = y_{long} - x_{long}$ y $\Delta_{lat} = y_{lat} - x_{lat}$, entonces*

$$d_{geo}(x, y) = 2R \arcsin \left\{ \sin^2(\Delta_{lat}/2) + \cos(x_{lat}) \cos(y_{lat}) \sin^2(\Delta_{long}/2) \right\}$$

Ya con una noción de distancia geográfica establecida, podemos proceder con el filtro simple propuesto.

R 2 (Criterio de distancia estática). *Sea $d_{geo}(i, j)$ la distancia entre los hoteles i y j , en kilómetros, y sea $r_{outer} > 0$. Entonces excluimos de las recomendaciones a todos los hoteles más lejanos de i que r_{outer} . Los hoteles que*

pasan a la siguiente etapa están contenidos en el conjunto

$$\mathcal{H}_i^2 = \{j \in \mathcal{H}_i^1 : d_{geo}(i, j) \leq r_{outer}\}$$

El círculo centrado en i de radio r_{outer} define lo que denominamos **cerca exterior**, fuera de la cuál ignoramos a todos los hoteles; denominamos **candidateos iniciales** a los hoteles contenidos en \mathcal{H}_i^2 .

Buscamos que \mathcal{H}_i^2 sea mucho más pequeño que \mathcal{H}_i^1 , ya que como la cantidad de los números a almacenar crece cuadráticamente, entonces el número de comparaciones que deberemos hacer se reducirá masivamente y por lo tanto problema será factible computacionalmente.

Es importante hacer una buena elección de r_{outer} para mantener el cálculo manejable pero sin ignorar hoteles interesantes. Un criterio que en la práctica funcionó bien es $r_{outer} = 30$ km. En las zonas muy densas este tope nunca se alcanzará, como veremos más adelante, pero en las zonas ralas (y por ende con menos tráfico), difícilmente representará un viaje de más de media hora entre las opciones.

2.4. Similitud de hoteles

El filtro de cercanía y el límite de precio son esenciales para el algoritmo de recomendación. Sin embargo, no capturan nada sobre la naturaleza misma del hotel. Para ello utilizaremos información sobre los servicios disponibles en cada hotel, puesto que incluyen información más detallada sobre la categoría y el perfil que las estrellas por sí solas. Buscaremos desarrollar una medida de similitud entre hoteles que sea más informativa y utilizarla para ordenar las recomendaciones que estén a una distancia y precio razonables.

Lo primero que tenemos que hacer al diseñar la similitud entre dos hoteles es pensar qué entendemos por “hoteles similares”. Una idea es utilizar los servicios tal cuál en un vector binario s_j para cada hotel j , y por ejemplo utilizar como medida de similitud la cantidad de servicios en que coinciden dos hoteles i y j :

$$sim(i, j) = \sum_k 1 - |s_{ik} - s_{jk}|$$

Después de algunas pruebas con distancias de este estilo nos dimos cuenta de tres problemas fundamentales que tiene este enfoque, independientemente de la medida particular que utilicemos:

1. Es una técnica muy ruidosa y no permite que servicios (casi) sustitutos contribuyan a la similitud entre los hoteles.
2. No se toma en cuenta cuál de los hoteles tiene más servicios.
3. No necesariamente se captura la filosofía de los hoteles.

Elaboremos más en estos puntos. El primero tiene que ver con el hecho de que los hoteles son más que la suma de sus servicios: son experiencias completas. Es decir, si el hotel A tiene fútbol, snorkel y tenis, probablemente se parezca más al hotel B que tiene basquetbol, kayak y squash, que al hotel C, que sólo tiene fútbol, puesto que aunque A y B no tienen ningún deporte en común, tienen una selección que será suficientemente buena en la mayoría de los casos. Sin embargo, cualquier similitud basada en los vectores s_j diría que A se parece más a C que a B. La solución a este problema es aglomerar los servicios en categorías dentro de las cuales los servicios sean aproximadamente equivalentes y comparar la cantidad de cada categoría.

El segundo punto es sumamente importante porque hay un factor fundamentalmente asimétrico a la hora de comparar dos hoteles. Esto se debe a que un cliente que ve un hotel con pocos servicios probablemente no le molestará que una recomendación tenga más servicios, siempre y cuando el precio y el perfil del hotel sean similares. De hecho, probablemente preferirá una mayor cantidad de servicios si se respeta el precio.

El tercer punto surge de que este tipo de medidas no capturan explícitamente la similitud entre los perfiles de dos hoteles. Si bien, mientras más servicios coincidan, probablemente haya una mayor coincidencia, una similitud más o menos chica no necesariamente implica que dos hoteles sean fundamentalmente distintos. Consideremos dos hoteles de playa, pero uno de tres y otro de cinco estrellas. Es razonable pensar que sus perfiles no son tan lejanos entre sí como lo serían con un hotel citadino, incluso a pesar de que sus servicios probablemente diferirán más entre sí que entre el de cinco estrellas y el citadino.

Una última observación es que aunque es más fácil pensar en similitudes entre los hoteles, matemáticamente es más práctico trabajar con distancias. La razón es que ya hay una noción formal de lo que es una distancia, lo que nos permitirá utilizar matemáticas bien difundidas en lugar de inventar nociones propias:

Definición 2.3 (Distancia). *Sea A un conjunto no vacío. La función $d : A \rightarrow \mathbb{R}^+ \cup \{0\}$ es una distancia si y sólo si tiene las siguientes propiedades*

para cualesquiera $a, b, c \in A$:

$$d(a, b) \geq 0 \text{ (positividad)} \quad (2.1)$$

$$d(a, b) = 0 \text{ si y sólo si } a = b \text{ (distancia mínima)} \quad (2.2)$$

$$d(a, b) = d(b, a) \text{ (simetría)} \quad (2.3)$$

$$d(a, b) \leq d(a, c) + d(c, b) \text{ (desigualdad del triángulo)} \quad (2.4)$$

Una ventaja inmediata aparte de facilitar la comprensión es que si decidimos violar algunas de las propiedades de las distancias sabremos exactamente qué estamos haciendo. Aparte de eso, este enfoque no tiene ningún inconveniente si se construye una distancia acotada, pues se puede invertir al final para generar una medida de similitud.

A continuación describiremos con más detalle cómo tomamos en cuenta estas observaciones y cómo las integramos para construir nuestra medida de similitud

Categorización de los servicios

Dado que el enfoque servicio a servicio tiene problemas independientes de la distancia que elijamos, será el primero que abordaremos. Una forma que en la práctica tuvo buenos resultados fue agrupar los servicios en distintas categorías. Dentro de cada una de ellas suponemos que los servicios son sustitutos perfectos, esto es, son perfectamente intercambiables. Si bien este supuesto podría parecer algo aventurado, tuvo muy buenos resultados en la práctica. Comenzaremos por describir la información disponible y luego la forma en que agrupamos en categorías.

Definición 2.4 (Conjunto de servicios). *Definimos \mathcal{S} como el conjunto de todos los servicios y $n_s = |\mathcal{S}|$.*

Es conveniente tener un solo objeto que contenga la información de todos los hoteles y sus respectivos servicios. Elegimos una matriz para facilitar la notación.

Definición 2.5 (Matriz de servicios). *Definimos la matriz de servicios S de $n_s \times N$ como sigue:*

$$s_{ij} = \begin{cases} 1 & \text{si el hotel } j \text{ tiene el servicio } i \\ 0 & \text{en otro caso} \end{cases}$$

Las columnas de S corresponden a los vectores de características de los hoteles:

$$S = [s_1 s_2 \dots s_N],$$

es decir, s_j corresponde al vector de características del hotel j . Por practicidad, denotamos los renglones de S por $z_i^T = [s_{i1}, s_{i2}, \dots, s_{iN}]$.

Para agrupar los servicios tomamos los renglones de S como individuos y los aglomeramos utilizando un aglomerado jerárquico con el criterio de Ward. Tenemos entonces n_s “observaciones” cada una con N “características”. Como en este caso sí importa tener grupos interpretables, asistimos al método manualmente. El resultado es que el algoritmo dio los grupos básicos, pero utilizamos conocimiento humano para arreglar las inconsistencias. Durante el proceso, se elige un número de categorías y se le da un nombre apropiado a cada una. Con lo que acabamos entonces es con un mapeo de cada servicio a su categoría correspondiente. Formalizamos estos conceptos en la siguiente definición:

Definición 2.6 (Categorías de servicios). Denotamos por \mathcal{C} al conjunto de todas las categorías de servicios, de cardinalidad n_c . Definimos la categoría a la que pertenece cada servicio vía la función $\kappa : \mathcal{S} \rightarrow \mathcal{C}$, es decir si el servicio s pertenece a la categoría c , entonces $\kappa(s) = c$.

Ya teniendo el mapeo de los servicios crudos a las categorías, es conveniente generar una nueva versión de S . Dado que dentro de cada categoría supondremos que los servicios son intercambiables, basta comparar el número de servicios en cada categoría que tiene cada hotel, de modo que resulta natural la siguiente definición.

Definición 2.7 (Matriz de categorías). Definimos la matriz de categorías C de $n_c \times N$ como sigue:

$$c_{ij} = \sum_{\kappa(z)=i} s_{zj}$$

Es decir, c_{ij} resume el número de servicios que tiene el hotel j en la categoría i . Similarmente al caso de S , denotamos las columnas de C por c_j , de modo que

$$C = [c_1 c_2 \dots c_N]$$

De este modo, podemos utilizar las columnas de C para comparar los diversos hoteles. En este caso, por ejemplo $\|c_i - c_j\|_1 = \sum_k |c_{ki} - c_{kj}|$ resulta

en una distancia entre los hoteles i y j que permite intercambiar los servicios en cada categoría. En las siguientes secciones describiremos qué distancia decidimos utilizar.

Similitud de servicios

Dado que el problema principal ha sido tratado, podemos proceder a definir una distancia entre los hoteles vía sus vectores característicos c_j . Inicialmente no es tan descabellado utilizar algo como la norma 1, $\|\cdot\|_1$, pero no nos da la asimetría que queremos. Retomando los tres problemas fundamentales, sí es importante tomar en cuenta que es distinto que el hotel buscado tenga más servicios que el recomendado (y por lo tanto la recomendación no es buena), a que lo inverso suceda (en cuyo caso la recomendación es de mejor calidad). En el gráfico (2.1) ilustramos lo que sucede. Lo que buscaremos

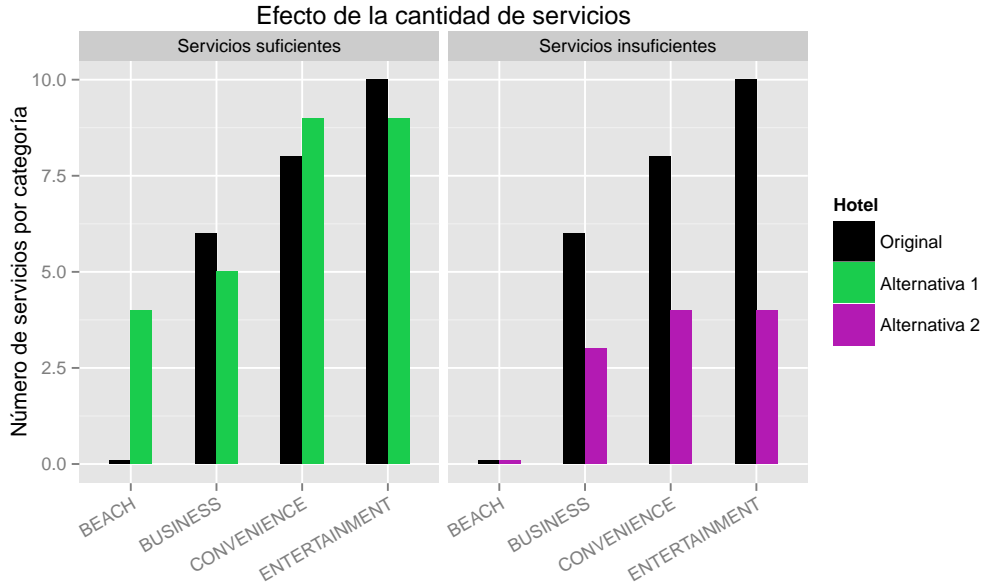


Figura 2.1: Ejemplos de similitud de servicios para 4 categorías distintas. El hotel verde se parece al negro porque no le faltan muchos servicios que el negro sí tiene. En contraste, al hotel morado le faltan muchos servicios.

entonces es penalizar los servicios que le falten a un hotel a la hora de buscar candidatos. Para ello definimos la siguiente función:

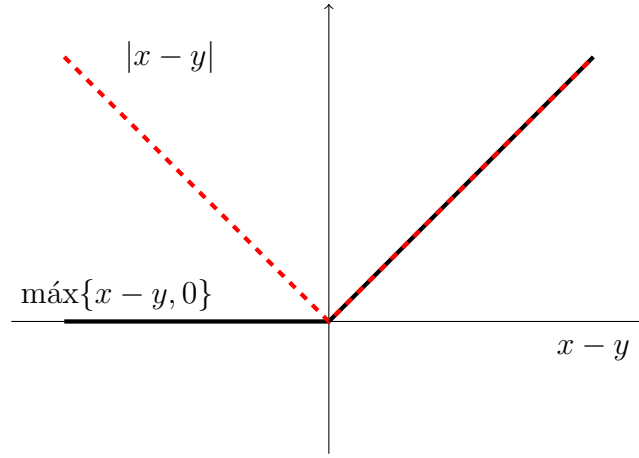


Figura 2.2: Ejemplo univariado. Distancia entre x y y usando valor absoluto (---) y pérdida *hinge* (—). La pérdida *hinge* no penaliza el caso en que $y > x$.

Definición 2.8 (Pseudodistancia *hinge*). Sean $x, y \in \mathbb{R}^n$. Definimos la pseudodistancia *hinge* de x a y como sigue:

$$d_h(x, y) = \sum_{k=1}^n \text{máx}\{x_k - y_k, 0\}$$

Esta función de pérdida no cumple la propiedad (2.3) porque no es simétrica, como es evidente en la figura (2.2). Por lo tanto, se trata de una *pseudodistancia*. Esta propiedad de $d_h(\cdot, \cdot)$ es algo positivo porque refleja justamente la asimetría de la relación entre los servicios del hotel buscado y los recomendados. Aplicando esta pseudodistancia a los vectores característicos de dos hoteles c_i y c_j obtenemos

$$d_h(c_i, c_j) = \sum_{k=1}^{n_c} \text{máx}\{c_{ki} - c_{kj}, 0\}$$

La lógica detrás de esta cantidad es la siguiente: la resta $c_{ki} - c_{kj}$ representa la diferencia en número de servicios que el hotel i tiene sobre el j en la categoría k . Dado nuestro supuesto de que los servicios son perfectamente intercambiables, tenemos que $c_{ki} \geq c_{kj}$ si y sólo si el hotel i es “mejor” en la categoría k , y por lo tanto j incurrirá en una penalización de ese número por

su falta de servicios. En el caso en que sea peor, es decir, cuando $c_{ki} \leq c_{kj}$, no se penaliza a j y el máximo hace que se sume 0 a la distancia. En conclusión, $d_h(c_i, c_j)$ = cantidad de servicios que j no tiene pero i sí, tomando en cuenta los servicios en cada categoría como sustitutos perfectos.

Para etapas posteriores, cabe mencionar que dado que el número de servicios y de categorías es finito, hay una cota máxima a la distancia *hinge* entre dos hoteles, digamos M_h , que además coincide con el número total de servicios n_s (el caso se da cuando un hotel no tiene servicios y el otro tiene todos los servicios disponibles). Si bien $d_h(\cdot, \cdot)$ tiene una interpretación muy conveniente, cambiarla de escala es más útil para el modelo y hace que sea casi inmediato ver si una distancia es grande o chica, independientemente de la cantidad de servicios en el sistema. Para ello simplemente la dividimos entre M_h , de modo que queda entre 0 y 1, lo que también permite definir una similitud asociada:

Definición 2.9 (Pseudodistancia *hinge* normalizada y similitud de servicios). Sean $x, y \in \mathbb{R}^n$. Definimos la pseudodistancia *hinge* normalizada de x a y como sigue:

$$d_{hn}(x, y) = \frac{1}{n_s} d_h(x, y)$$

Definimos la similitud de servicios (o basada en *hinge*) entre x e y como sigue:

$$sim_h(x, y) = 1 - d_{hn}(x, y)$$

Similitud de perfil

Si bien la pseudodistancia *hinge* captura muy bien la falta de servicios de los hoteles candidatos, no es capaz de diferenciar adecuadamente los perfiles de los hoteles. En la gráfica (2.3) ilustramos este hecho con dos ejemplos. Una manera de modelar el perfil de un hotel es interpretando el número de servicios en cada categoría no como un número absoluto, sino como una *proporción* de los servicios disponibles. Es decir, tener m servicios en la categoría k es mucho o poco dependiendo del número total de servicios que tiene el hotel. Como en este caso cualquier diferencia de perfil afecta, necesitamos una distancia simétrica, que definimos a continuación.

Definición 2.10 (Divergencia y similitud de perfil). Sean $x, y \in \mathbb{R}$. Definimos

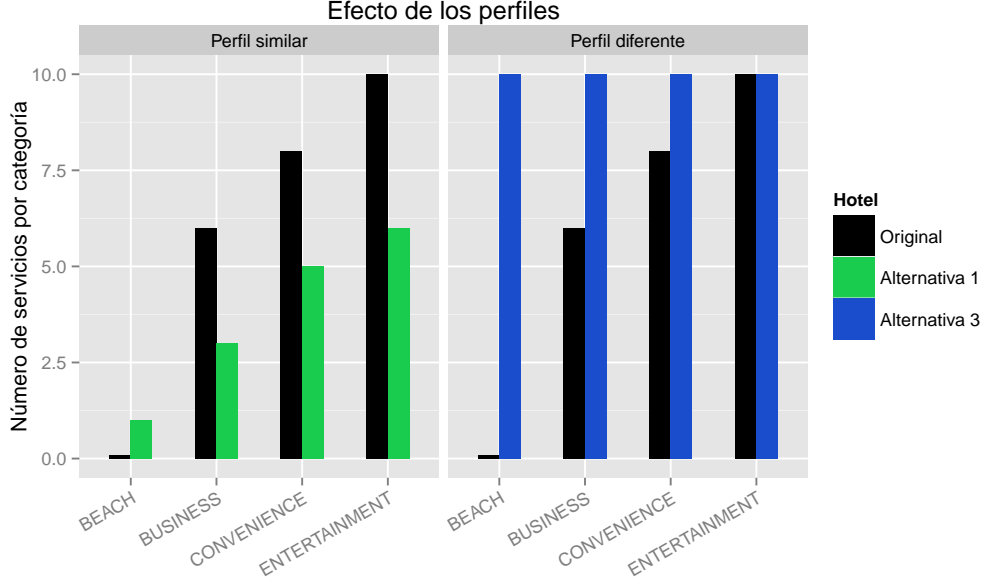


Figura 2.3: Ejemplos de similitud de perfil para 4 categorías distintas. El hotel verde se parece al negro porque aunque le faltan servicios, tiene aproximadamente la misma proporción de sus servicios en cada categoría. El hotel azul tiene muchos servicios, pero tiene un enfoque distinto (en este caso es un hotel de súper lujo muy enfocado a playa, mientras que el negro no tiene playa).

la divergencia entre x e y como sigue:

$$d_{div}(x, y) = \frac{1}{2} \sum_{k=1}^n \left| \frac{x_i}{\sum_j x_j} - \frac{y_i}{\sum_j y_j} \right|,$$

es decir, la distancia absoluta entre los vectores normalizados por su suma. Definimos también la similitud de perfil (o basada en la divergencia) como

$$sim_{div}(x, y) = 1 - d_{div}(x, y)$$

Como $d_{div}(\cdot, \cdot)$ siempre está entre 0 y 1, La ventaja de esta distancia (se deja como ejercicio al lector probar que en efecto cumple todas las propiedades) es que primero quita la escala y luego compara los vectores en un espacio equitativo. Cabe mencionar que bajo este criterio, el hotel verde de

la gráfica (2.3) es prácticamente idéntico al negro, incluso a pesar de que es un hotel de mucho menor categoría. El punto es que gastan sus recursos de manera similar.

Dado que los vectores normalizados suman 1 y los estamos restando, es fácil ver que por el factor de $1/2$ que incluimos en la definición, la divergencia nunca puede pasar de 1 y por lo tanto la similitud que definimos en efecto tiene sentido.

Similitud completa

En las secciones anteriores resolvimos parcialmente los tres puntos críticos. Los criterios de similitud de servicios y de perfil se basan en el agrupamiento de los servicios para capturar parte de las similitudes entre los hoteles. Sin embargo, como explicamos en cada sección, ambas tienen fallas. La idea es entonces integrar las dos medidas en un criterio que sí tome en cuenta todo. El problema es cómo hacerlo sin perder información y dándole un peso apropiado a cada una.

De este punto en adelante trabajaremos nuevamente con similitudes, porque son más cómodas para razonar y porque además son equivalentes a sus distancias asociadas. Dado que tanto la similitud de servicios como la de perfil están en escalas equiparables, una manera razonable de combinarlas es promediándolas. Sin embargo, podrían tener diferentes niveles de variabilidad, por lo que es prudente permitir la posibilidad de utilizar una ponderación distinta de $1/2$.

Definición 2.11 (Similitud entre hoteles). *Sea $\alpha \in (0, 1)$ y sean i, j hoteles en \mathcal{H} . Definimos la similitud con parámetro α entre i y j como sigue:*

$$sim_{\alpha}(c_i, c_j) = \alpha sim_h(c_i, c_j) + (1 - \alpha) sim_{div}(c_i, c_j)$$

Cabe mencionar que como la similitud entre hoteles no es simétrica porque está basada en parte en la similitud de servicios. Es decir, no necesariamente se cumple que $sim_{\alpha}(c_i, c_j) = sim_{\alpha}(c_j, c_i)$.

Ya habiendo establecido la forma de combinar las dos medidas de similitud, el siguiente paso es elegir un valor apropiado de α , que de preferencia sea óptima con respecto a algún criterio. Para ello, hay que notar que aunque ambas similitudes están entre 0 y 1, podrían tener niveles de variación y medias distintos, por lo que utilizar $\alpha = 1/2$ podría no ser una buena idea. Ilustramos este hecho en la figura (2.4). En el ejemplo, $\alpha = 1/2$ probable-

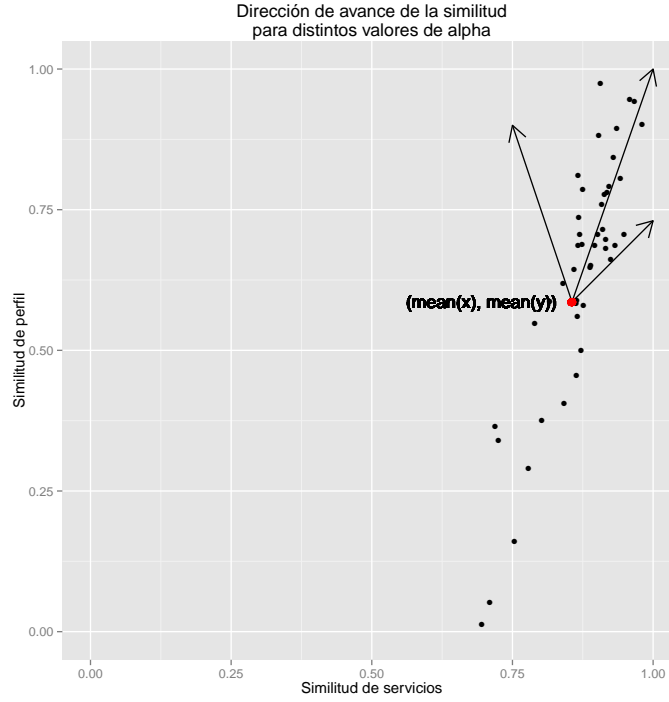


Figura 2.4: La elección de α determina la dirección en la que avanza la similitud total al variar la de servicios y la de perfil.

mente sería similar a avanzar en la flecha pequeña de la derecha, pero es claro que avanzando en la dirección de la de en medio estaríamos tratando ambas similitudes de manera más equitativa. Dado que lo que queremos es identificar hoteles similares, una buena manera de decidir el valor de α es utilizando la que maximice la varianza explicada por la similitud completa. Proponemos utilizar un Análisis de Componentes principales para calcular la α óptima.

Como queremos calcular el valor óptimo de α para las recomendaciones que haremos, tenemos que hacer el análisis con todas (o una muestra suficiente de) las recomendaciones de todos los hoteles. La forma de hacerlo es como sigue:

1. Establecer los parámetros de los filtros estáticos de precio y distancia.
2. Para cada hotel i , generamos \mathcal{H}_i^2 .

3. Para todo $j \in \mathcal{H}_i^2$, calculamos

$$\begin{aligned} h_{ij} &:= \text{sim}_h(c_i, c_j) \\ d_{ij} &:= \text{sim}_{div}(c_i, c_j) \end{aligned}$$

4. Insertamos cada pareja (h_{ij}, d_{ij}) como un renglón en una matriz A , con dos columnas y número de renglones igual a $\sum_i |\mathcal{H}_i^2|$. Es decir, los renglones de A pueden ser representados como en la figura (2.4).
5. Corremos Componentes Principales sobre A y obtenemos el vector propio asociado a la primera componente principal, digamos

$$\vec{a} = (a_h, a_d)$$

6. El vector \vec{a} define la dirección deseada. Para que sus componentes sumen 1 y por lo tanto definan valores aceptables para α y $1-\alpha$, dividimos sus componentes entre la suma, de modo que:

$$\begin{aligned} \alpha^* &:= \frac{a_h}{a_h + a_d} \\ 1 - \alpha^* &:= \frac{a_d}{a_h + a_d} \end{aligned}$$

De este punto en adelante únicamente utilizaremos la similitud completa utilizando α^* . La siguiente definición nos ayudará a facilitar la exposición.

Definición 2.12 (Score). *La similitud completa que resulta de utilizar α^* se denomina score, es decir:*

$$\text{score}(i, j) := \text{sim}_{\alpha^*}(c_i, c_j)$$

El *score* condensa todas las propiedades que buscábamos en una medida de similitud y que no tenga los tres problemas expuestos al principio de esta sección. Claramente no presentará la volatilidad de las medidas basadas en servicios crudos porque está basada en categorías. Sin embargo, este mismo hecho genera potencial de enmascaramiento si las categorías no están bien elegidas. Esto se debe a que si son heterogéneas y no consisten de servicios que sean sustitutos más o menos cercanos, puede haber falsos positivos y falsos negativos al calcular cualquier medida basada en ellos. En cuanto al punto (2), suponiendo que las categorías en efecto son apropiadas, la parte de

hinge del *score* garantizará hasta cierto punto que no se recomiende hoteles con demasiado pocos de los servicios de interés. Finalmente, el hecho de que hay una colaboración de la similitud basada en la divergencia evita que se recomiende hoteles con un perfil muy distinto al original.

Suponiendo que el *score* es una medida satisfactoria de similitud, podemos proseguir a los siguientes componentes del sistema de recomendación, que tomarán en cuenta aspectos físicos y temporales que son imposibles de capturar utilizando únicamente información de los servicios de los hoteles.

2.5. Distancia: filtro dinámico

El concepto de “hotel similar” que buscamos en este caso tiene que ver no sólo con que los hoteles sean en sí mismos similares, sino que además estén cerca. La dificultad es que la noción de lo que significa la palabra “cerca” varía de lugar a lugar. Por ejemplo, en una zona con una alta densidad de hoteles, como Reforma Centro en la Ciudad de México, un hotel es considerado cercano si está a, digamos, no más de 1 ó 2 kilómetros. Una opción en el Aeropuerto Benito Juárez no es aceptable y difícilmente alguien diría que dicha opción está cerca de Reforma Centro (aunque en realidad está a 15 km). Sin embargo, en una zona despoblada como la Riviera Maya, en la que todo el transporte es por carretera, 15 kilómetros son 10 minutos en coche, y de hecho hay muchos hoteles que no tienen vecinos a menos de varios kilómetros de distancia. La moraleja de lo que acabamos de exponer es que la distancia se percibe distinto según la zona, por lo que debemos utilizar algún criterio adaptativo.

Una primera idea para incorporar la distancia en el modelo fue meterla de alguna manera en el *score*, por ejemplo, como un tercer término en el promedio. Sin embargo, encontramos un argumento que sugiere que esto podría no ser la mejor idea: Supongamos que buscamos recomendaciones para el hotel A y tenemos dos candidatos, B y C. Dada la densidad de la zona, ambos están “cerca” de A, pero resulta que B está a la mitad de distancia de A que C. Además, supongamos que C se parece un poco más a A que B. La pregunta es entonces: ¿deberíamos recomendar a B sobre C aunque se parece menos y dado que ambos están cerca? Creemos que la respuesta a este ejercicio es claramente que no, que si ambos están cerca, es preferible el hotel más similar. Esta manera de pensar nos llevó a la conclusión de que incorporar la distancia de manera suave (i.e. en el promedio) tal vez no sea

la mejor alternativa y que posiblemente la podríamos utilizar en la forma de un filtro dinámico. Para ello hay que seguir dos pasos:

1. Definir qué entendemos por hoteles cercanos.
2. Pasar a la siguiente ronda del modelo los hoteles que cumplen dicho criterio, ordenados por *score*, es decir, por qué tan parecidos son al hotel en cuestión.

El paso crítico para este método es la definición de cercanía. Una primera idea es, por ejemplo, elegir los 20 ó 30 hoteles más cercanos al hotel en cuestión. Sin embargo, este enfoque ingenuo tiene un problema grave: si un hotel es único en su categoría en la zona (por ejemplo es de 5 estrellas en un pueblo), entonces posiblemente los 20 hoteles más cercanos a él no se parezcan en lo absoluto. En estos casos debería permitirse una mayor flexibilidad de búsqueda en favor de obtener mejores recomendaciones. En la figura (2.5) ilustramos lo que sucede y a continuación describimos el proceso formalmente.

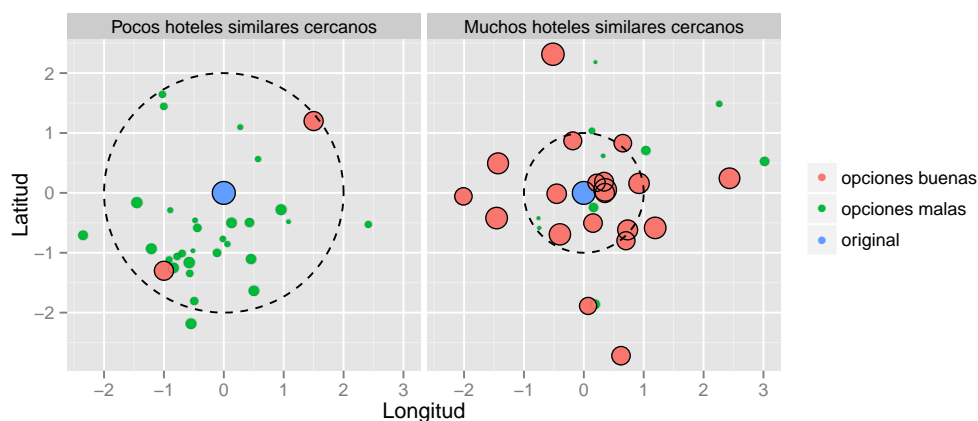


Figura 2.5: El filtro debe ser más laxo cuando hay pocos hoteles similares cercanos (izquierda) y más estricto cuando hay más (derecha).

R 3 (Distancia: filtro dinámico). *La función de acumulación de score al alejarse del hotel i está dada por $F_i : \mathbb{R} \rightarrow \mathbb{R}$, es decir:*

$$F_i(d) = \sum_{j | d_{geo}(i,j) < d} score(i, j)$$

*Para una cantidad predefinida a acumular, digamos $w_{inner} > 0$, definimos la **cerca interior** como el círculo centrado en el hotel i y con radio dado por*

$$r_{inner} = \arg \min_d \{F_i(d) \geq w_{inner}\},$$

es decir, la mínima distancia que garantice que se acumula al menos w_{inner} de score dentro de la cerca interior. Los hoteles que pasan a la siguiente ronda son entonces los que estén en la cerca interior:

$$\mathcal{H}_i^3 = \{j \in \mathcal{H}_i^2 : d_{geo}(i, j) \leq r_{inner}\}$$

2.6. Precios dinámicos

Un punto que hasta el momento no hemos tratado son las posibles promociones que los hoteles puedan ofrecer. Si bien el objetivo principal de este sistema es recomendar hoteles similares y no necesariamente descuentos, es deseable que éstos tuvieran algún impacto. Para ello notemos que el filtro estático de precio toma en cuenta el promedio de los precios en el largo plazo. Eso quiere decir que si un hotel baja sus precios temporalmente, podría ser un candidato para otros hoteles más baratos, pero el promedio impide que esto suceda. Una forma de suavizar este problema es filtrar con los precios en la fecha de la búsqueda en lugar de los promedios. Es decir que, si bien debemos calcular α y r_{inner} con \mathcal{H}_i^2 tal como lo hemos descrito hasta el momento, conviene utilizar

$$\mathcal{D}^* = \text{período para el cual se busca reservar} \quad (2.5)$$

para calcular los precios en lugar del período histórico largo \mathcal{D} .

R 4 (Precio dinámico). *Dadas α y r_{inner} obtenidas en R1, R2 y R3, denotamos las **recomendaciones finales** por*

$$\mathcal{R}_i = \{j \in \mathcal{H} : p_j^{D^*} > p_i^{sup}, d_{geo}(i, j) \leq r_{inner}\},$$

y los hoteles se presentarán ordenados por score, de mayor a menor.

2.7. Consideraciones prácticas

Una última consideración que hay que tener son los casos extremos. Un caso es que la zona del hotel buscado tenga muy pocas opciones dentro de la cerca exterior y el otro es cuando el hotel es muy barato para la zona, por lo que el filtro de precio le impide ver suficientes de los demás hoteles.

Para aliviar el primer problema el procedimiento es simple: como no queremos que se queden vacías las recomendaciones, ignoramos todo lo anterior y regresamos un número pequeño predefinido de hoteles cercanos, ordenados por distancia, dado que hay muy pocas opciones. Si por otro lado sí hay suficientes hoteles dentro de la cerca exterior pero el filtro de precio es demasiado estricto, entonces se ignora por completo este filtro.

R 5 (Casos extremos). Sea rec_{min} el mínimo número de recomendaciones que se desea.

(I) Si $|\mathcal{H}_i^2| < rec_{min}$, entonces redefinimos

$$\mathcal{H}_i^1 = \mathcal{H}$$

(II) Si aún después del paso anterior $|\mathcal{H}_i^2| < rec_{min}$, entonces ignoramos todas las reglas y recomendamos los hoteles en

$$\mathcal{R}_i^{ex} = \{j \in \mathcal{H} : j \text{ es uno de los } rec_{min} \text{ hoteles más cercanos a } i\}$$

Además, ignoramos el score y los presentamos ordenados por distancia al hotel i .

(III) Las dos reglas anteriores aplican de manera análoga para \mathcal{H}_i^{1*} y \mathcal{H}_i^{2*} .

2.8. Juntando todo

El procedimiento descrito anteriormente es largo y puede ser difícil de asimilar en su conjunto, de modo que presentamos aquí una versión resumida:

1. Sean i un hotel, D un periodo largo y D^* el periodo en el que se desea reservar.
2. Calculamos α y r_{inner} como está descrito en R1, R2 y R3.

3. Checamos los casos extremos descritos en R5. Si ninguno aplica, continuamos; si resulta el caso (II), entonces recalculamos el paso anterior y continuamos; si resulta el caso (I), recomendamos con \mathcal{R}_i^{ex} y terminamos.
4. Generamos las recomendaciones finales como está descrito en R4 y las presentamos ordenadas por *score*.

2.9. Comentarios

Antes de terminar el capítulo, hay un punto que nos gustaría resaltar y es por qué no decidimos simplemente basar la distancia en la norma 1. La respuesta corta es que es muy exigente. Aunado a ello, no nos permite ponderar fácilmente las contribuciones del perfil y de los servicios. Es posible que haya alguna manera de alterar esta distancia para capturar la asimetría de la disponibilidad de servicios. De ninguna forma afirmamos que no haya alternativas al menos tan buenas como la nuestra, pero dado que obtuvimos resultados satisfactorios, nos quedamos con el método presentado anteriormente. Investigar nuevas medidas de distancia sería un buen tema de trabajo futuro.

Capítulo 3

Implementación

En este capítulo describiremos algunos detalles adicionales al modelo matemático que son necesarios para la implementación del sistema de recomendación en el mundo real. Hablaremos de qué fuentes obtendremos los datos necesarios para el modelo y de qué haremos en los casos en los que la calidad no sea la necesaria para correr el modelo.

3.1. Información

En esta sección describiremos el qué, el cómo y el dónde de la información que utilizamos para el sistema de recomendación. Para ello hay que notar que necesita dos insumos para funcionar. El primero es información a nivel hotel, como son los nombres, los detalles geográficos y los precios. El segundo consiste de la información de los servicios del hotel. En los siguientes dos apartados describiremos cada uno con mayor detalle.

Información de hoteles

La tabla de información de hoteles debe tener por lo menos las siguientes columnas para que el modelo funcione:

- Clave del hotel.
- Latitud y longitud.
- Precio promedio por noche en alguna moneda unificada (USD) y en algún período razonablemente largo, digamos uno o dos años.

Opcionalmente se puede agregar el volumen de noches y la utilidad bruta en el mismo periodo para hacer algunos análisis, aunque el modelo estrictamente no los necesita.

Para obtener esta tabla es necesario notar que no hay un catálogo de precios, por lo que los tuvimos que estimar mediante las reservaciones que se hicieron en el periodo analizado. Es decir, el precio promedio por noche en el periodo D (en USD) de un hotel dado es:

$$\text{precio promedio} = \frac{\text{Suma de importes (USD) de las reservas en el periodo } D}{\text{Suma de noches reservadas en el periodo } D}$$

Para acceder fácilmente esta información generamos un procedimiento en SQL Server que recibe una fecha de inicio, hace D = de la fecha proporcionada hasta hoy, y finalmente regresa la tabla con la información de los hoteles. Para la primera versión decidimos utilizar un periodo de 2 años, con el fin de evitar que los hoteles poco vendidos no tuvieran ningún precio.

Información de servicios

Las matrices S y C , que contienen la información de los servicios y las categorías respectivamente, son bastante ralas. Por esta razón es conveniente almacenarlas como relaciones con un renglón por cada elemento distinto de cero de la matriz correspondiente. Como S es binaria, la almacenaremos en una tabla con dos columnas, clave del hotel y clave del servicio, mientras que C requiere tres: clave del hotel, clave de la categoría y número de servicios en la categoría.

Dado que en la base de datos de Best Day ya está la tabla con los servicios en el formato deseado, basta filtrar los hoteles con información incompleta o de zonas que no nos interesen y ya tenemos la primera tabla. Para obtener la segunda fue necesario primero correr el aglomerado en R utilizando la primera tabla y crear a mano la relación entre los servicios y las categorías que elegimos. Posteriormente bastó pegarle a cada servicio su categoría correspondiente y sumarizar la tabla de servicios utilizando la clave del hotel y la clave de la categoría como llaves; los conteos dentro de cada grupo corresponden a los valores c_{ij} . Nuevamente, generamos un procedimiento de SQL Server para poder obtener esta información con una simple llamada a una función en lugar de una consulta completa.

A continuación describiremos un tema sobre los servicios que es importante en el contexto de la recomendación de hoteles similares.

Plan de alimentos

La decisión de exactamente qué características del hotel cuentan como servicios es subjetiva hasta cierto punto. Hay algunos que no se incluyeron por ejemplo por razones administrativas. Entre ellos están los planes de alimentos, es decir, la cantidad de comidas que incluyen los hoteles. El plan de alimentos puede ser muy importante en algunas regiones vacacionales en donde haya hoteles de tipo “todo incluido” mezclados con hoteles que sólo incluyen desayuno, o incluso nada. La razón es que los precios de dos hoteles del mismo nivel de lujo podrían diferir bastante si uno es todo incluido y el otro no. Por ello es importante incorporar esta información de algún modo en el modelo.

Si bien hay diversas formas en que se podría tomar en cuenta el plan de alimentos en el sistema de recomendación, uno que nos pareció razonable, fácil de implementar y que además funciona bien, es generar una categoría de alimentos en la matriz C . La idea es que si dos hoteles tienen el mismo plan de alimentos, tendrán más servicios en común y por lo tanto favoreceremos que se recomienden entre sí, mientras que si difieren, se tendrán que parecer mucho para que salgan en las recomendaciones del otro. Cada hotel puede tener muchos contratos con Best Day y puede vender cuartos con y sin todo incluido, pero en general, es razonable pensar que si un hotel tiene la opción de todo incluido, es parte de la filosofía del hotel. Por ello, para cada hotel elegimos el plan de alimentos que más comidas incluidas tuviera. El resultado es que podemos agregar hasta cuatro servicios por hotel, uno si incluye desayuno, uno si comida, uno si cena y otro si incluye las bebidas. En la matriz C esto se refleja en una nueva categoría que puede tomar cualquier valor entre cero y cuatro.

3.2. Sistema de respaldo

Uno de los grandes temas que surgen en cualquier modelo matemático que surja del análisis de datos es la calidad de la información. El modelo descrito en el capítulo anterior presupone no sólo que se tiene la información ya en el formato adecuado, sino que además está limpia. La norma en el mundo real es que esto no es así, por lo que será necesario agregar componentes al sistema de recomendación para que funcione incluso cuando la información no sea perfecta. A continuación enlistaremos algunos de los principales pro-

blemas que pueden aparecer en este contexto y las posibles soluciones que les podemos dar:

- El hotel no tiene ninguna reservación en el periodo analizado para obtener los precios: Utilizamos el modelo pero ignoramos los precios por completo.
- El hotel tiene mal la información de servicios: Tendremos que utilizar un sistema de recomendación alternativo, como por ejemplo el antiguo.
- El hotel no tiene coordenadas o las coordenadas son incorrectas: No hay nada que podamos hacer salvo conseguirlas de alguna fuente.

La primera dificultad no es tan grave porque podemos utilizar el modelo sin problemas, sólo que ignorando los precios. En realidad es lo mejor que podemos hacer y dado que de cualquier forma tiene que ser un hotel muy poco popular, no importa mucho que las recomendaciones no sean óptimas en estos casos.

Los otros dos puntos son más complicados de resolver porque implican que de plano no podemos utilizar el sistema de recomendación. Una forma de mitigar esto es utilizar el sistema de recomendación antiguo de Best Day que está descrito en el capítulo 1, es decir, tomar los hoteles en el mismo destino y de las mismas estrellas y recomendarlos en el *Sort Order* del momento. La ventaja de esto es que únicamente requiere el destino y las estrellas del hotel para funcionar, por lo que en la mayoría de los casos podremos evitar no recomendar nada.

Si también falla el sistema de recomendación antiguo no hay mucho que podamos hacer, pues o bien ni siquiera tenemos el destino o estrellas del hotel, o no hay ninguna alternativa en el mismo destino. Si detectamos una zona o algún hotel con esta clase de problemas lo que sí podríamos hacer en estos casos es detectar la información faltante o de mala calidad y completarla o limpiarla a mano, según la necesidad.

Capítulo 4

Desempeño

Casos de éxito, incluso en zonas ralas.

Google Analytics

Red de interconexiones, incluyendo PageRank.

Capítulo 5

Conclusión

5.1. Trabajo futuro

Porcentaje dinámico en forma de ‘U’ para el tope de precio.
Otras métricas

5.2. Comentarios finales

Por qué no distancia absoluta?
No es personalizado, sino de hoteles similares.
[Ver presentación para qué sí/no es o podría ser]