



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**POSICIONAMIENTO DE PRODUCTOS NUEVOS EN LAS CATEGORÍAS DE
FALABELLA.COM UTILIZANDO LA PLATAFORMA GOOGLE CLOUD
(GCP)**

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

ALINSON GUILLERMO ROBERTO LINCOPÁN ROJAS

PROFESOR GUÍA:
MARCOS ORCHARD CONCHA

MIEMBROS DE LA COMISIÓN:
CESAR AZURDIA MEZA
AQUILES MARTINEZ BERRIOS

Este trabajo ha sido parcialmente financiado por:
FALABELLA RETAIL SA.

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: **ALINSON GUILLERMO ROBERTO LINCOPÁN ROJAS**
FECHA: 2021
PROF. GUÍA: MARCOS ORCHARD CONCHA

POSICIONAMIENTO DE PRODUCTOS NUEVOS EN LAS CATEGORÍAS DE FALABELLA.COM UTILIZANDO LA PLATAFORMA GOOGLE CLOUD (GCP)

El ecommerce es una de las áreas de comercio que viene en constante aumento desde la aparición del internet, el cual tuvo un importante incremento debido a la pandemia que actualmente afecta al mundo. Situándose en Chile, Falabella es el actual líder en cuanto a desarrollo y avances tecnológicos en lo que respecta al comercio en línea, al punto de contar con modelos inteligentes que realizan la tarea de ordenar algunos productos en la página de Falabella.com

Para realizar el ordenamiento diario de los productos, el modelo utiliza atributos de producto y atributos de interacción de los clientes con dichos productos, ante este panorama es que aparece un problema, el tratar con productos nuevos, ya que no cuentan con datos de interacción y como resultado no son correctamente ordenados.

El trabajo de esta memoria abarca dicho problema, se encarga de detectar los productos nuevos, para luego asignarles una posición. Dicha posición se basa en la relación que tienen los productos nuevos con sus pares históricos.

El trabajo se divide en 3 partes principales, la exploración de las fuentes, la representación de los productos, la selección y evaluación del modelo. La exploración se realiza mediante consultas de SQL en BigQuery, para la representación de los productos se evalúa el desempeño de 2 embeddings (Word2Vec y Universal Sentence Encoder), y para las métricas del modelo se utilizan KPI's de ecommerce, como la conversión y la venta por visita.

El Universal Sentence Encoder fue el embedding que tuvo mejor desempeño, por sobre el Word2Vec. La evaluación con los KPI's estuvo enfocada en las categorías que presentaron mayor conversión y mayor venta de productos en boletas. Se analizan categorías en general y en particular, resultando en que no hay una relación clara entre la distancia semántica con la distancia por desempeño. Otras observaciones (por categoría generalmente), permiten definir que enfoque y que indicador utilizar, por ejemplo se presenta el caso de smartphones donde mediante los resultados se observa que para las 2 semanas no hay resultado concluyente, mientras que para la misma categoría, pero en un periodo de 4 semanas se recomienda el uso de la conversión. Se presenta además los costos asociados del modelo en Google Cloud Platform.

Por ultimo este trabajo no se pudo llevar a completitud debido a temas logísticos internos de la empresa, pero realiza recomendaciones para mejorar el desempeño, como también la incorporación de otras áreas para mejorar la experiencia del usuario, y a su vez, el valor de Falabella Retail SA.

*Dedicado a todos aquellos que perdieron un ser querido por la pandemia.
Consuelo a su alma.*

Agradecimientos

Agradezco principalmente a mi familia, en primer lugar a mi madre (Pamela) por estar siempre a mi lado en las buenas y en las malas, en segundo lugar a mi padre (Alinson) por siempre apoyarme e impulsarme a seguir adelante, a mi hermano (Christopher) por ser un pilar de apoyo para nunca rendirme, a mi hermana (Bárbara) por entregarme su amor y asumir el rol de madre en varias ocasiones y finalmente a mi otra hermana (Tamara) por aguantar a tan pesado hermano y por entregar su cariño incondicional.

Le doy las gracias a Daniela por ser la mejor amiga que uno puede pedir, por su cariño y apañe durante estos 20 años de amistad. Agradezco también a mis amigos del colegio, porque representaron un apoyo increíble cuando comencé esta carrera. También agradezco a todos los amigos que hice durante estos 7 años, siempre me hicieron sentir acompañado. Y están también los “*Brrritos*”, les agradezco por invitarme todos esos viernes a compartir disfrutando con un trago.

Agradezco a Falabella, por darme la oportunidad de realizar mi trabajo de memoria en su empresa, en especial al equipo de PLP, que desde el comienzo se mostró dispuesto a apoyarme en mis labores. Agradecimiento especial a Aquiles quien fue mi mentor y consejero en la mayoría de los problemas que tuve durante el año. Agradezco de igual manera al profesor Marcos y al profesor Cesar, por su apoyo y críticas en el desarrollo de mi memoria.

Un agradecimiento especial a “*La Guarida del Gato*”, por darme horas de diversión en el mundo de los videojuegos online, y ser un apoyo al momento de pasar por malos momentos. Siempre dispuestos a escuchar y prestar ayuda cuando estaba a su alcance.

Finalmente gracias a mi por escribir este documento y gracias a ti por leerlo.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Descripción del problema	2
1.3. Objetivos	2
1.3.1. Objetivo general	2
1.3.2. Objetivo específico	3
1.4. Resumen de la metodología	3
2. Marco Teórico y Estado del Arte	4
2.1. Marco Teórico	4
2.1.1. Sistemas de Recomendación	4
2.1.1.1. Sistemas de recomendación colaborativos	4
2.1.1.2. Sistemas de recomendación por contenido	6
2.1.1.3. Sistemas de recomendación por conocimiento	6
2.1.1.4. Sistemas de recomendación híbridos	7
2.1.1.5. Métricas de evaluación de los sistemas de recomendación	9
2.1.2. Métricas de evaluación de desempeño en ecommerce (KPI)	11
2.2. Estado del Arte	13
3. Metodología	14
3.1. Exploración de las fuentes	14
3.2. Selección del <i>embedding</i>	15
3.3. Cálculo de la distancia coseno e iteración diaria	16
3.4. Evaluación	17
3.5. Plataforma	18
3.6. Pruebas a realizar	20
4. Resultados y análisis	22
4.1. Resultados de la exploración de las bases de datos	22
4.2. Resultados de la selección del <i>embedding</i>	22
4.3. Resultados del modelo	23
5. Conclusión y trabajo futuro	29
5.0.1. Conclusión	29
5.0.2. Trabajo futuro	30
Bibliografía	31

Anexo A. Resultados

34

A.1. Distribuciones

34

Índice de Tablas

4.1.	Cantidad de Match	23
4.2.	Gasto mensual estimado	28

Índice de Ilustraciones

2.1.	Ejemplos de tipos de ranking	5
2.2.	Taxonomía resumen de los sistemas de recomendación híbridos	9
3.1.	Esquema del funnel	15
3.2.	BigQuery	19
3.3.	Notebooks	19
3.4.	JupyterLab	20
3.5.	Jobs	20
3.6.	Storage	21
4.1.	Distribución espacial de productos más cercanos	24
4.2.	Distribución espacial agrupada de productos más cercanos	25
4.3.	Distancia promedio de CR para la categoría smartphones (2 semanas)	26
4.4.	Distancia promedio de CR para smartphones (4 semanas)	27
A.1.	Distancia promedio de CR para la categoría secadoras	34
A.2.	Venta por visita promedio para la categoría secadoras	35
A.3.	Distancia promedio RPV para la categoría smartphones	36

Capítulo 1

Introducción

1.1. Motivación

El comercio en línea o *ecommerce* se refiere a todo tipo de transacciones realizadas electrónicamente o a través de internet¹. Dichas transacciones se pueden agrupar en 4 categorías: empresa a consumidor (B2C por sus siglas en inglés), empresa a empresa (B2B), consumidor a consumidor (C2C) o consumidor a empresa (C2B). En el resto de la memoria se tratará solamente del tipo B2C, por su naturaleza llega a más público y es, quizás el tipo de transacción más común hoy en día².

Durante los últimos años en Chile se ha visto un aumento significativo del *ecommerce*, según una presentación publicada por la Cámara de Comercio en Santiago, el número de empresas presentes del comercio en línea se duplicó en cantidad desde el año 2016 al 2019, llegando a un total de 9.236[1]. De igual manera la cantidad de ventas del tipo B2C presenta un crecimiento evidente, cerrando el año 2019 con un total de 6.000 MMUS (miles de millones de dolares)[2].

Dentro de los mayores exponentes del B2C se encuentra el retail, el cual también se ve afectado por el comercio en línea, el año pasado se registró una contribución de 2.7% puntos porcentuales en el crecimiento de esta industria (porcentaje aportado por el *ecommerce* del total del crecimiento). El impacto que tiene esta actividad es aún mayor, en el mismo periodo llegó a un 6% del total de la penetración que tuvo el retail, con proyecciones a un 10% para el año 2020[1]. Demostrando así que el comercio *online* va en constante aumento. La penetración de mercado guarda relación en como una empresa mantiene y atrae nuevos clientes, sin la necesidad de lanzar nuevos productos o servicios³.

A pesar de este favorable escenario para el *ecommerce* lamentablemente no se aplican las mismas reglas que el comercio tradicional (compras que requieren de interacción humana), hay desafíos que se deben sortear, siendo el mayor de ellos: la transformación digital. El pilar fundamental del comercio en línea se basa en la experiencia de compra *online*, la transformación digital apunta a ello, a que las empresas emprendan un cambio de estrategia, de manera

¹ <https://www.shopify.com/encyclopedia/what-is-ecommerce>

² <https://www.hollingsworthllc.com/different-types-of-e-commerce/>

³ <https://rockcontent.com/es/blog/penetracion-de-mercado/>

que se orienten en mejorar la experiencia de los clientes a través de las tecnologías digitales para ofrecer soluciones más eficaces e innovadoras[3].

Y sin lugar a dudas, algo que favoreció la transformación e impulsó aun más el desarrollo del *ecommerce* fue la pandemia del *covid-19*. Dadas las características del virus, las interacciones entre personas se vieron limitadas, los comercios que dependían de la presencia de clientes se vieron obligados a cerrar momentáneamente para mantener el avance del virus bajo control. Como consecuencia, algunos sectores de la industrias se vieron obligados a cambiar su modelo de negocios: los restaurantes optaron por el *delivery*, los cines comenzaron a arrendaban las películas *online* y las ventas de productos se trasladaron a la venta web. Este aumento del comercio en línea se vio reflejado en los 9.423 MMUSD que recaudo este tipo de transacciones (hasta noviembre del 2020[4]).

1.2. Descripción del problema

Cuando se trata de clientes el principal problema no es captar su atención para que compre, si no de entregarle una experiencia de venta satisfactoria, de manera que la próxima vez que quiera comprar vuelva a hacerlo en el mismo sitio. Una forma de entregar una experiencia de compra agradable es mediante los sistemas de recomendación. Los sistemas de recomendación se basan en los gustos o interacciones de los clientes para recomendar una futura compra, por lo tanto, mientras mayor interacción tengo un producto con los usuario más fácil es identificar su posición en los productos recomendados (un producto que tenga baja o nula interacción se encontrará en las posiciones finales).

Ante esta situación y el aumento del comercio en línea se presenta el problema de la recomendación de productos nuevos, cuando un nuevo producto es agregado al catálogo este no tiene visitas, nunca fue agregado al carro, ni menos fue comprado (ni en tienda ni mediante la página web), como resultado el recomendador no puede atribuirle una posición, por lo que se le designa un aproximado. Esto no sería un problema si los productos agregados fueran pocos, pero considerando una empresa como Falabella, las categorías de productos son muchas y revisar cada listado para definir la posición de un producto nuevo entre un universo de productos ya disponibles no es fácil e incurre en un gran gasto para la empresa, donde se destina personal a dicho posicionamiento, sin mencionar que tal decisión genera conflicto con los modelos ya existentes.

1.3. Objetivos

1.3.1. Objetivo general

Como solución a este problema y objetivo general de esta memoria, se propone diseñar un modelo que sea capaz de identificar los productos nuevos, luego atribuirles la información requerida del recomendador según la similitud semántica que tienen con sus pares históricos.

1.3.2. Objetivo específico

1. Determinar la base para el catálogo de productos, donde se puedan diferenciar claramente los productos históricos de los productos nuevos, junto a ello definir el criterio para diferenciar productos históricos de nuevos.
2. Determinar cuales serán los atributos semánticos requeridos para ver la cercanía entre productos (título y marca son obligatorios).
3. Determinar la mejor técnica de procesamiento natural de lenguaje para ver la similitud entre productos.
4. Evaluar el desempeño del recomendador considerando las variaciones anteriores.
5. Implementar el modelo en el recomendador general y evaluar el desempeño.

1.4. Resumen de la metodología

Se especifica la metodología de trabajo a seguir para cumplir con el objetivo general y secundarios de la memoria.

1. La elección de la base de datos del catálogo productos de *Falabella.com*, al mismo tiempo se debe definir el criterio que diferencie los productos históricos de los nuevos. Dada la cantidad de datos también se debe definir la especificidad de los productos.
2. Definido el criterio es necesario explorar el *funnel*, estos son los datos que albergan la interacción que tuvieron los clientes en la página web (productos vistos, comprados, agregados al carro, entre otros).
3. Después de la exploración se deben definir los atributos de productos a utilizar
4. Continuando con la exploración, se debe elegir el *embedding* a utilizar, a priori se tiene en consideración *Word2Vec* y *Universal Sentence Encoder*.
5. Se debe definir una métrica de evaluación acorde al problema, de esta forma se tiene una forma de probar y mejorar los modelos de forma iterativa.
6. Por último se analizan los resultados obtenidos para los casos propuestos.

Capítulo 2

Marco Teórico y Estado del Arte

2.1. Marco Teórico

2.1.1. Sistemas de Recomendación

El continuo crecimiento del comercio web ha permitido el desarrollo y mejora de los sistemas de recomendación, basándose en el trabajo de Resnick[5], el autor Ricci[6] define a los sistemas de recomendación como las herramientas o software capaces de proveer sugerencias de productos, o servicios, que pueden ser de uso para el usuario. Los sistemas de recomendaciones estaban enfocados principalmente a los clientes que era incapaces de tomar decisiones debido a una falta de experiencia para evaluar múltiples opciones[5], pero al día de hoy se utilizan para mejorar la experiencia del usuario. Existen diversas formas de recomendar productos o servicios, siguiendo el desarrollo de Aggarwall[7] se distinguen 5 tipos: colaborativo, por contenido, por conocimiento, demográfico e híbrido.

Antes de caracterizarlos es importante mencionar algunos conceptos. El primero de ellos es *rating*, que según el diccionario de Cambridge¹ corresponde a una medida de calidad o éxito de un producto o servicio cuando se compara con similares, por ejemplo: la película A tiene 5 estrellas, mientras que la película B tiene 3, luego se puede afirmar que A tiene mejor rating que B.

2.1.1.1. Sistemas de recomendación colaborativos

Los sistemas colaborativos, también conocidos como filtros colaborativos (CF con sus siglas en inglés), utilizan la información del *rating* que generan los usuarios para realizar sus recomendaciones. La idea base detrás del filtro colaborativo es que las evaluaciones de los productos no calificados pueden ser imputadas, debido a que existe una alta correlación de *ratings* entre usuarios y productos. A modo de ejemplo, sean Pedro y Anita 2 usuarios de Netflix, ambos tienen preferencia por las películas de terror, Pedro ha visto “El Resplendor”, “Psicosis” y “En la Alta Hierba”, todas las calificó de forma similar; Anita en cambio solo ha visto las últimas 2 películas y el *rating* fue similar al de Pedro, por lo tanto, de forma segura se puede recomendar “El Resplendor” a Anita dado el parecido que tiene con Pedro. Los filtros colaborativos se encargan de analizar estas relaciones, ya sean producto a producto,

¹ <https://dictionary.cambridge.org/es-LA/dictionary/english/rating>

usuario a usuario o una mezcla de ambos.

Las recomendaciones enfocada en usuarios, buscan encontrar las características que presenta un conjunto de usuarios, de manera que cuando aparezca un usuario X nuevo, se le pueda asignar a un grupo ya definido para la recomendación. La recomendación enfocada en ítems se encarga en determinar un set de ítems similares, de manera que cuando X visite un producto Z perteneciente al set, se le puedan recomendar similares. A estos 2 tipos de recomendación se les conoce como algoritmos de filtro colaborativo basados en la vecindad, son los más simples y fáciles de implementar pero poco escalables. Otro tipo de filtro colaborativo es el basado en modelos, estos incluyen todo lo relacionado con las técnicas de *machine learning* y *data mining*, en el contexto de modelos predictivos. En esta categoría los modelos parametrizan características e interacciones, las que luego son optimizadas para realizar las recomendaciones correspondientes. Estos modelos son más difíciles de implementar pero su naturaleza los hace más escalables.

Al tener que considerar todo tipo de interacciones, los sistemas de recomendación se ven obligados a utilizar una escala de *rating* adecuada, estas pueden ser de distintos tipos: binarias o por intervalo. Dependiendo del tipo de sistemas es que algunas escalas son mejores que otras, por ejemplo, una escala binaria (me gusta o no me gusta) se puede encontrar en plataformas de video como Youtube. Los *ratings* de tipo intervalos son más frecuentes, pueden ser numéricos, simbólicos (estrellas por ejemplo) o semánticos (Poco conforme, Conforme, Muy Conforme). Un último tipo de *rating* menos visible es el unario, que sólo indica gusto pero no disgusto, esta forma calificar es particularmente común cuando se utilizan *ratings* implícitos.



Figura 2.1: Ejemplo de rating por intervalo simbólico, calificación de 5 estrellas presente en plataformas como Netflix, Amazon o Google Play Store. Imagen de [7].

En el párrafo anterior se mencionaron tanto los *ratings* explícitos como los implícitos, las calificaciones explícitas corresponden a las primeras 2 de ellas, donde el usuario **explícitamente** califica el producto o un servicio (por ejemplo otorgando estrellas según su grado de satisfacción, Figura 2.1). Pero también está el *rating implícito*, donde el usuario no califica el producto, si no que se va infiriendo su preferencia según sus acciones. Por ejemplo un usuario A ve películas de ciencia ficción y al termino de cada película las puntúa con estrellas, el

recomendador sabe el tipo de contenido a recomendar; sea ahora una persona B que también ve películas de ciencia ficción, pero que no las califica, de forma implícita se puede saber que le gustan ese tipo de filmes.

2.1.1.2. Sistemas de recomendación por contenido

El filtro colaborativo mencionado anteriormente solo considera la interacción entre ítems, usuarios o entre ambos, no considera el contenido de los ítems. El filtro por contenido se encarga de captar los atributos específicos de los productos o servicios para realizar las recomendaciones de lo que ha gustado previamente. Un ejemplo de este filtro sería con la música, si un usuario A escucha música de categoría rock contemporánea, al momento que se lancen 2 discos, uno de categoría rock y otro de categoría folclore, se recomendará aquel que tiene el contenido similar a los escuchados previamente. Al contrario del filtro colaborativo, el filtro por contenido no considera la interacción existente entre usuarios con gustos similares, **se enfoca en cada usuario** y en el o los atributos que le gustaron en particular.

Los filtros por contenido dependen de 2 cosas en particular, una fuente de datos que sea capaz de describir en alguna manera los atributos del objeto y una caracterización de los perfiles del usuarios. El perfil de los usuarios se debe construir según las restricciones y características del funcionamiento del negocio, no hay restricción, pudiendo utilizar tanto la interacción explícita como la implícita. Como se mencionó con anterioridad, las preferencias de otros usuarios no juega ningún rol en este tipo de recomendaciones, sin embargo se puede considerar la utilización de los similares para ayudar la construcción de los perfiles nuevos con baja interacción.

La presencia de interacción de usuarios similares permite atacar el problema de la partida lenta (*cold start* en inglés). Este problema es típico de los filtros colaborativos, se refiere a la aparición de un usuario o productos nuevo, al no poseer interacción se hace difícil de realizar una recomendación adecuada. En el caso de los filtros por contenido la partida lenta no es mayor problema cuando se trata de los ítems, debido a que la extracción de los atributos no se ve afectada, sin embargo si es un problema cuando se trata de usuarios nuevos ya que no hay como extraer información inicial.

El uso del filtro sigue la misma metodología del proceso KDD presentado por Cios[11]. En primer lugar se deben analizar los datos disponibles, hacer una exploración y limpieza de los mismos. El segundo lugar corresponde a seleccionar las características importantes de los ítems a ser utilizadas por el recomendador, junto con la definición del *target*. En tercer lugar está el aprendizaje de los perfiles de los usuarios (que corresponde a utilizar algún modelo de machine learning). El cuarto y último paso corresponde a la evaluación y mejora del recomendador (sobre la evaluación se detalla mas adelante).

2.1.1.3. Sistemas de recomendación por conocimiento

Tanto los sistemas de recomendación por contenido y colaborativos requieren una cantidad no menor de datos para poder funcionar correctamente, se presenta la *cold start*, entre

otros problemas, pero mas allá de los problemas, hay un caso donde los recomendadores no funcionan bien: productos altamente específicos. Ejemplos de este tipo son autos, viajes de turismo, servicios financieros, etc., lo que presentan en común es un elevado valor y una baja o nula presencia de *rating*. Para estos casos existe el filtro por conocimiento, que se encarga de atender la explicitad de lo solicitado por los usuarios.

Si bien suena sencillo atender específicamente lo que requieren los usuarios, en la práctica no es tarea sencilla poder recomendarlo lo buscado, debido a que existen muchas combinaciones de lo solicitado, o que cierta combinación no exista. Ante esto la mejor forma de solucionar el problema es mediante un correcto guiado del cliente para concretar su recomendación, es importante que esta guía sea interactiva, donde el usuario pueda modificar rápidamente sus preferencias y que los resultados se adecuen a lo solicitado. En ese sentido es crucial tener un alto conocimiento del sector, y así ser capaces de codificar la información relevante dentro de parámetros que permitan guiar la recomendación. Por ejemplo si se quieren recomendar casas, es importante tener conocimiento de piezas, precio de venta, metros cuadrados construidos, etc., sólo así se pueden crear los filtros necesarios para dar la opción al usuario de expresar lo requerido.

Los sistemas de recomendación por conocimiento pueden ser categorizados según la metodología de interacción de los usuarios y el correspondiente conocimiento para facilitar la interacción. Primero está la recomendación por restricciones, en este caso el usuario define los límites de lo que busca (ej. precio máximo, cantidad de habitaciones, potencia del motor, tamaño de pantalla, etc.), todos son atributos que pueden quedar restringidos a rangos o valores específicos, pero que el usuario no considerará si no está dentro de lo especificado. El segundo es la recomendación por caso, consiste en que el usuario explícitamente pide ciertas características, luego el recomendador muestra una selección de ítems para que el usuario haga un juicio de valor, si no se encuentra lo solicitado puede volver a realizar un pedido siendo más específico, los ítems no selectos cuentan como casos a considerar para el siguiente resultado, el ejemplo más claro de este es un buscador de producto. Es importante mencionar que en cualquiera sea la categoría debe existir una interacción continua entre el usuario y el sistema de recomendación, con un feed-back inmediato.

A pesar que la experiencia es altamente personalizada, la mayoría de estos sistemas de recomendación no guarda la información de los clientes de forma permanente, se trabaja en formato de sesión. Esto conlleva una ventaja: no sufre por la partida lenta al no depender de la data histórica. Se podría almacenar la información de los usuarios, pero debido a la principal utilización de este recomendador está enfocado en productos poco vendidos, de mayor precio y altamente específicos, es poco probable que un mismo usuario realice múltiples consultas en el tiempo.

2.1.1.4. Sistemas de recomendación híbridos

Al comienzo del capítulo se hablo de 5 tipos de recomendadores: por contenido, colaborativo, por conocimiento, demográficos e híbridos. Las recomendaciones por demografía no son recomendaciones por si solas, sino que pueden ser utilizadas en cualquiera de los mencionados anteriormente, estas recomendaciones se hacen según localidad y tiempo (no es lo

mismo recomendar a la gente de Iquique en verano que a la gente de Valdivia en invierno). Aclarado el punto de los recomendadores demográficos, el último de estos sistemas consiste en el recomendador híbrido, y como su nombre lo indica, corresponde a la utilización de uno o más de los filtros mencionados con anterioridad en este capítulo. La utilización de múltiples métodos se hace para cubrir las falencias que presentan por separado, aprovechando además todo el potencial de los datos disponibles.

Hay 3 formas de crear los sistemas híbridos: diseño de ensamble, diseño monolítico y sistemas mixtos. El diseño por ensamble resulta de la combinación *off-the-shelf* de los modelos solitarios, mezclados en una única salida, por ejemplo combinar el *rating* del filtro colaborativo con el *rating* del filtro por contenido en una salida única, la combinación de los resultados puede igualitaria o con ponderación de pesos. Existen 2 tipos de ensambles: secuencial, donde la predicción de una de las partes depende de otra; y paralelo, donde los algoritmos trabajan de forma independiente. El diseño monolítico es una integración utilizando las distintas fuentes de datos de los recomendadores, de esta forma se pueden ver fácilmente los datos de interacción junto a los de contenido o a los de especificidad. Por último el diseño mixto, este es muy similar al diseño por ensamble, utiliza los distintos tipos de recomendadores como cajas negras, pero la recomendación final muestra los ítems de cada filtro particular como un conjunto de recomendaciones (no asigna pesos, sino que el resultado final son todas las recomendaciones).

Los diseños dan paso a la categorización de los filtros híbridos, dentro de este grupo se pueden distinguir: por peso, por cambio, cascada, aumento de *features*, combinación de características, meta-nivel y mixtos. Los híbridos por peso caen dentro del diseño por ensamble paralelo, donde la recomendación final se define según un ponderado de los sistemas participantes. Los híbridos por cambio (*switching* en inglés) también son del tipo ensamble, estos están enfocados en el desarrollo del recomendador, por ejemplo si la base de datos es pequeña comienza como un filtro por contenido y a medida que los *ratings* crecen se cambia por un filtro colaborativo, este tipo de recomendación evalúa cual sistema tiene mejor desempeño en que punto. Los híbridos por cascada también son de la familia por ensamble, aquí el recomendador va afinando la recomendación entregada por uno previo, por ejemplo el primer recomendador es por conocimiento, este pasa a uno por contenido para finalizar a uno colaborativo. El aumento de *features* es similar al de cascada, sólo que en este caso la salida de un recomendador no es la entrada directa del siguiente, el resultado se transforma en otro atributo más a considerar para el siguiente recomendador. La combinación de características o *feature combination* es la definición basal del diseño monolítico, donde la data de cada recomendador es unificada. En el caso del meta-nivel, se tiene que el modelo de un filtro se utiliza como entrada de otro recomendador, un ejemplo sería modificar el filtro colaborativo para que utilice la información de contenido para realizar segmentación de clientes. Finalmente el mixto tiene la misma definición que su diseño, donde la recomendación final es mostrar los resultados de todos los sistemas en solitario.

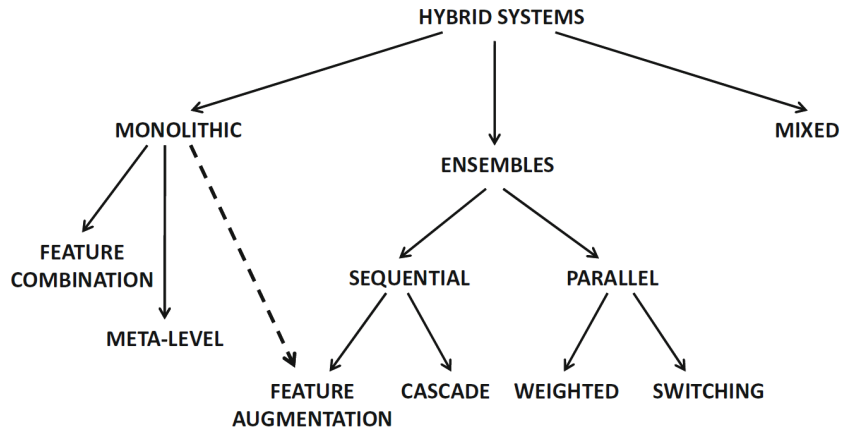


Figura 2.2: Taxonomía resumen de los sistemas de recomendación híbridos. Imagen de [7].

2.1.1.5. Métricas de evaluación de los sistemas de recomendación

Para asegurar que las recomendaciones sean las correctas y que los modelos vayan en mejora de la experiencia del usuario es que es necesario tener métricas para evaluar dicho desempeño, respecto a estas Aggarwal, las divide en 3 grupos: estudios de usuario, evaluación *online* y evaluación *offline*, donde las 2 últimas utilizan la data disponible. Para el caso de los estudios de usuario, se reclutan personas para que interactúen con el sistema de recomendación, realizando tareas específicas, dichas actividades son registradas por los modelos para realizar inferencias sobre los gustos del usuario. La ventaja que presenta es que se pueden simular bastantes escenarios, donde se le pueden pedir diversas tareas a los participantes, entre las desventajas está el hecho que los participantes están conscientes del estudio lo cual puede afectar sus decisiones, el gasto que se incurre en reclutar a los participantes y el sesgo existente al seleccionar los participantes.

La evaluación *online* también utiliza la interacción de los usuarios con la diferencia que no hay sujetos de pruebas seleccionados, y las interacciones ocurren en un sistema completamente desarrollado y operativo comercialmente. Los métodos de desempeño utilizados en este caso reciben el nombre de *A/B testing*, donde se mide directamente el impacto del sistema en el usuario final. La idea básica del *A/B testing* es comparar 2 algoritmos de la siguiente forma:

1. Dividir a los usuarios en 2 grupos A y B
2. Usar un algoritmo en el grupo A y otro algoritmo en el grupo B por un periodo de tiempo, manteniendo las otras condiciones (por ejemplo la ubicación de los usuarios al momento de la compra) entre los 2 grupos lo mas similar posible.
3. Al final del proceso se comparan las métricas de desempeño de los 2 grupos, para determinar la selección del algoritmo.

Entre las ventajas de este método es que se pueden probar varios algoritmos en datos reales con usuarios que no están sesgados por una prueba, ni por selección (salvo que el motivo sea para cierto segmento de clientes), también está la reducción de costos al no tener que modificar ni crear una plataforma para el testeo, ni tampoco tener que pagar a los usuarios para las pruebas. La mayor desventaja radica en el hecho que requiere de una plataforma

funcional con una gran participación de usuarios, por lo que no funcionan en fases iniciales de un proyecto o comercio, otro de los problemas es que no hay control sobre los usuarios, por lo que se deben filtrar solo para analizar las interacciones deseadas. Como último contra esta el tiempo, los resultados y análisis *online* de los algoritmos solo se pueden ver al término del tiempo definido, lo que retrasa el *deployment* en caso que ningún algoritmo alcance estándares deseados.

En la evaluación *offline* se utiliza la data histórica, se pueden hacer los mismos experimentos que en la evaluación *online* (*A/B testing*). La gran ventaja es que no se requiere de la gran interacción de los clientes ya que todo está almacenado en el histórico, otra ventaja es que se pueden hacer más pruebas debido a la baja dependencia del tiempo (en este caso el tiempo es solo el que toma en procesar los datos). La gran desventaja es que no se puede analizar al actual propensión del usuario a reaccionar a los sistemas en el futuro, es decir, si en este instante se hacen pruebas de algoritmos y por desempeño se deciden cambiar, las interacciones del nuevo modelo no se verán reflejadas hasta en un futuro. Sobre este mismo punto, este tipo de análisis no permite capturar las tendencias actuales, por lo que suelen dejar de lado ciertas características como *serendipity* y *novelty*.

Cualquiera sea la elección de los tres métodos seleccionados anteriormente, se deben definir las métricas al momento de evaluar, además del *accuracy*, en los sistemas de recomendación hay otros factores a considerar como: la diversidad, novedad, robustez, escalabilidad, entre otros. El *accuracy* o precisión es de las métricas más utilizadas en los sistemas de recomendación, cuantifica la diferencia entre la predicción del modelo con el verdadero valor, si se tratara de un problema binario, el error se cuantifica como la cantidad de aciertos sobre la cantidad de casos totales. En casos donde se presenten valores numéricos, la precisión se puede cuantificar como un error, como por ejemplo *MSE* o *RMSE*. La decisión de los target depende de los requerimientos del sistema, por ejemplo, se puede dar énfasis a los productos que generen mayor ganancia por sobre los que no.

El *coverage* o cobertura se refiere a la capacidad del modelo de cubrir una cantidad de producto al momento de realizar la recomendación, por ejemplo, se puede dar el caso que el algoritmo tenga buena precisión pero siempre descarta cierto grupo de productos, dicho esto se debe tener en cuenta el *trade-off* entre la precisión y la cobertura. Un ejemplo común de este comportamiento se observa cuando el usuario tiene poca interacción, donde el listado de recomendación es corto y poco variado (a modo de visualización, al ingresar a una plataforma de películas *online*, las primeras recomendaciones serán basadas en las similitudes de los primeros *filmes* observados, lo cual mostrará un catálogo poco variado). La cobertura se puede dar en 2 ámbitos, en el espacio de usuarios o de productos: para el espacio de usuarios es la mínima cantidad de perfiles que debe ser construido para un usuario antes de hacer recomendaciones para ese usuario, y en el espacio de productos se mide la fracción de ítems para los cuales el ranking de al menos n usuarios puede ser predicho. Una definición análoga a la cobertura de producto es la cobertura de catálogo, que se entiende como la fracción de ítems que son recomendados al menos a 1 usuario:

$$CC = \frac{|\cup_{u=1}^m T_u|}{n} \quad (2.1)$$

Donde n corresponde al total de productos, T_u es el listado de los top- k ítems recomendados al usuario u , $u \in \{1 \dots m\}$.

Novelty o novedad en un sistema de recomendación, mide la tendencia del recomendador para entregar recomendaciones novedosas, de las cuales el usuario pudiera no estar consciente de su existencia. Productos novedosos, a menudo, incrementan la habilidad del usuario en percibir mejor aquello que le gusta o disgusta, mejorando finalmente el desempeño del recomendador. La idea básica para decir que un sistema es novedoso, se sustenta en que la recomendación realizada es más probable de ser elegida en el futuro que en el presente.

Serendipity se refiere al término de “descubrimiento fortuito”, es por tanto una medida de sorpresa en la recomendación. Se diferencia con *novelty* en que para la anterior se requiere que el usuario no este consciente de la recomendación, mientras que para el caso de *serendipity* la recomendación debe ser inesperada, generando un mayor impacto en la recomendación. La forma típica de medición en los sistemas *online*, es mediante un *feed-back* directo del usuario, para el caso de los métodos *offline* se utilizan 2 algoritmos, el recomendador que se quiera medir y un recomendador primario (que generalmente es sólo por contenido), luego la fracción top- k ítems que fueron correctamente recomendados y que no estuvieron en el recomendador primario define la medida de *serendipity* del recomendador.

La noción de *diversity* o diversidad implica que el listado de recomendaciones debe ser lo mas diversificado posible. La diversidad puede ser medida en términos de la similitud de contenido presente en las recomendaciones, los ítems se representan en un espacio vectorial común, donde se calcula la distancia entre pares de productos (dentro del listado recomendado), la lista que tenga el menor valor promedio de distancia será considerado la más diversa.

Por último está la escalabilidad del modelo, debido al alto tráfico de información en línea, y que el tamaño de los datos a procesar va en aumento diario, se hace necesario tener en consideración modelos que sean capaces de llevar a cabo las tareas de manera efectiva y eficiente. Entre los factores que se utilizan para cuantizar la escalabilidad está: el tiempo de entrenamiento, el tiempo de predicción y la cantidad de memoria utilizada.

2.1.2. Métricas de evaluación de desempeño en ecommerce (KPI)

Según con Brudan[17], los *Key Performance Indicators* (KPI según sus siglas en inglés), corresponden a indicadores que ayudan a cuantificar el progreso hacia un objetivo, proveyendo de visibilidad a empleados, equipos y departamentos sobre el rendimiento del negocio, lo que les permite evaluar la situación para tomar las acciones adecuadas. La función que cumplen los KPI's es la evaluar el desempeño que lleva la empresa o institución.

Las 2 características principales que estos deben cumplir son:

- Específicos y diferenciados según las actividades de la empresa (no se pueden utilizar los mismos indicadores en tienda física que en la página web, pueden haber coincidencias)
- Deben estar claramente definidos y deben ir en lineamiento con la misión y visión de la empresa

Para ver que tipo de información se quiere reportar con los KPI's es que en la empresa se deben considerar los siguientes aspectos[18]: primero el análisis debe ser claro y balanceado, en segundo lugar no se pueden tener una gran cantidad de KPI's porque esto dificulta la interpretación y desvía la atención importante y tercero, los análisis deben proveer de la información suficiente para comprender como desarrollar y mejorar el negocio y su presencia en el mercado.

Los KPI's no son algo reciente, pero si que tuvieron que agregarse algunos para poder satisfacer las necesidades del *ecommerce*. A la hora de elegir cualquiera de estos indicadores, hay que ver cual área se quiere monitorear: ventas, tráfico o satisfacción del cliente. Cualquiera de estas 3 va en pos de aumentar las ventas, sólo que se diferencian en el camino a seguir. Algunos de estos KPI's son: *conversion rate* (CR) o conversión, *revenue per visit* (RPV) o venta por visitante, *click-through rate* (CTR), *checkout abandonment rate* (o *drop rate*), entre otros.

La conversión es una de la métricas mas utilizadas en el *ecommerce*, que cuantifica el ratio de órdenes vendidas por sobre la cantidad de visitas recibidas, este indicador está enfocado en 2 puntos, llegar a nuevos clientes y mantener clientes antiguos. En el segmento *online* es muy común que exista alto tráfico con respecto a la venta, por lo que este valor suele rondar entre el 2% y 5%. La estrategia directa para mejorar este valor, es mejorar el *marketing*, ya sea con campañas, ofertas, etc.

$$CR = \frac{n^{\circ} \text{ ordenes(boletas)}}{n^{\circ} \text{ visitantes}} \quad (2.2)$$

La venta por visita indica en promedio cuanto es la ganancia que se tiene de un producto en particular, se calcula como el total vendido por sobre la cantidad de visitas distintas recibidas. Este indicador es útil para comprender el comportamiento que tienen los clientes con respecto a ciertos productos, lo que permite potenciar la venta de otros productos, por ejemplo si un producto "a" tiene alta venta y otro producto similar "b"no, al primer producto se le puede subir el precio a "a" para inducir al cliente a comprar "b".

$$RPV = \frac{\text{venta total(de un producto)}}{n^{\circ} \text{ visitantes}} \quad (2.3)$$

El *click-through rate* (CTR) o ratio de clic, es una medida que indica el impacto que genera un producto, en términos prácticos, indica el avance en clics que tiene un producto hasta llegar a ser comprado. Este indicador es altamente utilizado para ver la efectividad que tienen ciertas campañas, también se ocupa para estudiar el comportamiento que tienen los clientes dentro de la página web. Se calcula como la cantidad de clics por visitantes distintos (también esta la variación donde se calcula como la cantidad de clics por visita). Importante mencionar que la cantidad de clics se define hasta una cierta profundidad, para el caso de las campañas bastaría que el cliente sólo haga 1 clic (ver la campaña), mientras que en procesos de compra se puede analizar desde que entra a la página hasta que llega lo más específico de cada producto.

$$CTR = \frac{n^{\circ} \text{ de clics}}{n^{\circ} \text{ visitantes}} \quad (2.4)$$

2.2. Estado del Arte

El problema de esta memoria esta enfocado en la caracterización de los productos nuevos, en este sentido, y dada la naturaleza del problema, no se han encontrado papers o publicaciones que aborden el problema de forma particular. Pese a esto, hay trabajos de investigación en la misma industria de los cuales se podrían obtener beneficios, como ocurre con el trabajo de Krishnan y Amarthaluri[16] en WalmartLabs.

WalmartLabs es la sección de inteligencia desarrolladora de Walmart, se encarga de crear soluciones que vayan en mejora de una experiencia de compra. Uno de los problemas que tienen actualmente viene del **Marketplace**, a medida que aumenta la cantidad de vendedores, también lo hace la cantidad de productos, por lo que es necesario poder clasificarlos correctamente para mantener una experiencia de usuario. Los autores comparan un seria de algoritmos *Deep Learning* para determinar cual modelo entrega mejores resultados, paralelo a eso proponen la utilización de atributos estructurados. Los atributos para realizar la clasificación son del tipo texto, los que incluyen título, descripción, tamaño, color, entre otros, estos se definieron según la exploración de datos. Para el uso de los datos, estos se concatenan, pasan por un *embedding*, por un modelo que trabaje con texto y realizan la clasificación, su aporte está en la estructuración del contenido, de esta forma todas los productos quedan con una presentación común, lo que permite a los modelos enfocarse en los términos relevantes de los productos. Este aporte permite una mejora de desempeño de los modelos en un 3%, la mejora no es substancial, pero el costo computacional adicional es mínimo.

Capítulo 3

Metodología

En este capítulo se presenta todo el flujo de trabajo a realizar para lograr cumplir de manera metódica el objetivo principal y secundarios asociados. En primer lugar se realizará una exploración de las fuentes disponibles, estas incluyen tanto el catálogo de productos como los datos de interacción de los clientes en la página de Falabella.com. En segundo lugar sigue la definición del *embedding* a utilizar, para esto se harán pruebas entre el *Universal Sentence Encoder* y el *Word2Vec*, la evaluación es una métrica que se discutirá más tarde. En tercer lugar está la mejora de la métrica de evaluación, para este cometido se utilizarán los datos de interacción de los productos dentro de la página web.

3.1. Exploración de las fuentes

La exploración de las fuentes no es una labor que esté completamente definida, esta es particular de cada problema y el análisis varía según la composición de los datos. Las tablas a analizar se encuentran alojadas en la nube, en la plataforma de *Google Cloud*, las tablas que se deben analizar son de 2 tipos, una de catálogo de producto y otra de interacción. La exploración de las tablas se realiza con la herramienta de *Bigquery* que es nativa de la plataforma de Google, esta herramienta maneja tablas de datos mediante el lenguaje SQL.

Considerando la naturaleza de la tabla del catálogo, lo primero que se analiza es cuanta información de la página web se refleja en la base de datos. Como se pretende atacar el problema mediante procesamiento de texto, es necesario que la tabla cuente con la mayor información de texto disponible, respecto a esto se observa que se cuenta con el título, la marca, el tamaño, el color y las 5 especificaciones principales. Hay presencia de más columnas, pero estas son las principales para el modelo. Otras columnas que deben ser mencionadas y que son importantes son el sku, la categoría, la fecha de aparición del producto y el estado de publicación. Esta tabla recibe el nombre de *product feed* e incluye la información por día de los productos trabajados en Falabella.com

El análisis del funnel es distinto, todo el proceso de interacción esta dividido en múltiples tablas, que reflejan la seguidilla de pasos para llevar a cabo la compra: desde ver el producto, hacer clic para ver los detalles, agregar al carro y finalizar la compra con el pago. La exploración es para comprender el funcionamiento de la sucesión de eventos, de como validar los datos de interacción, de realizar cruces entre las tablas, entre otros. Como base de estas

pruebas se utiliza una *query* desarrollada por Aquiles (*data scientist* de Falabella). Con la base estudiada se realizan *requests* de prueba para revisión de resultados, de las pruebas se obtiene la comprensión de los indicadores de interacción: vistas del producto, venta total, cantidad de órdenes y unidades totales vendidas.

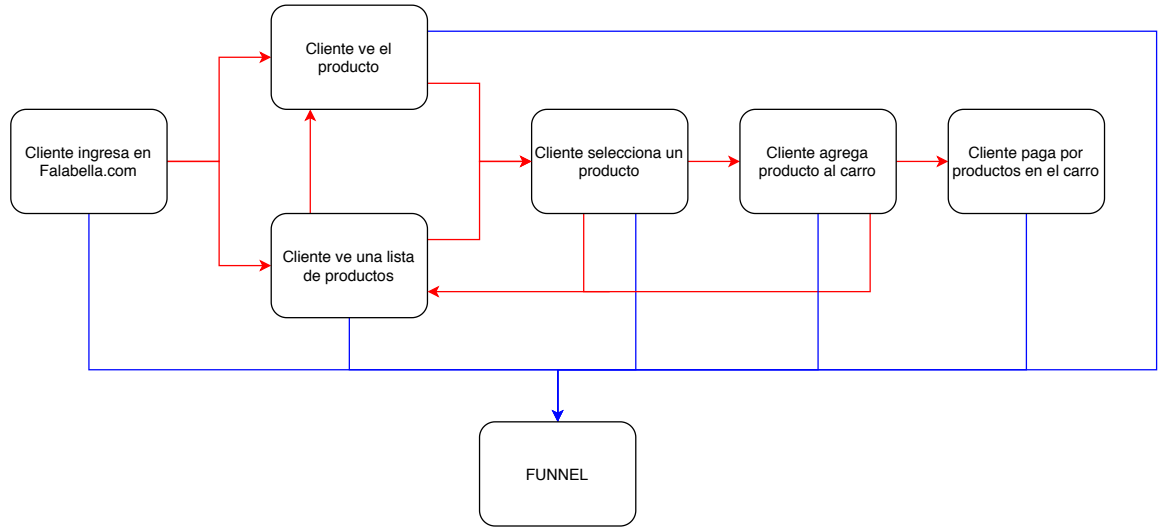


Figura 3.1: Esquema general del funcionamiento del funnel. Las flechas rojas indican el flujo general que realiza un cliente al momento de realizar una compra en la página de Falabella. Comienza ingresando en la página, selecciona algún producto, lo agrega al carro y paga. Las flechas azules muestran las acciones monitoreadas que existen en el funnel

La definición de producto nuevo no es directa, esta debe ir acorde a las reglas del negocio, pero más allá, debe ser bien definida y no prestarse para equivocaciones. Dentro de los productos nuevos que se presentan en Falabella hay 2 grandes diferenciaciones: productos nuevos como tal (se incorporan hace poco al mercado) y productos que llevan tiempo en el mercado pero son nuevos en Falabella, al primer tipo se le denominará novedoso. En primera instancia no se hará diferenciación para el uso, esto puede cambiar a medida se avance con el modelo. Se considera como producto nuevo todo aquel que lleve presente en el catálogo online menos de 28 días, dicho esto hay un par de casos a considerar, por ejemplo, cambio de códigos, devoluciones o productos despublicados que han vuelto a aparecer.

3.2. Selección del *embedding*

La selección del *embedding* es una parte importante, este se encarga de la representación del texto a utilizar, y por consiguiente tiene un peso relevante al momento de calcular las distancias entre los productos. Los *embeddings* a considerar son el *Universal Sentence Encoder*

y el *Word2Vec*. Para la realización de la prueba se ignoran los datos de interacción mientras que del catalogo de utiliza el título, marca, tamaño y subclase, además se utiliza el esquema de atributos estructurados planteados por Krishnan[16].

El catalogo a utilizar esta a nivel de sku padre, es decir no considera las variaciones de color, tipo o género, esto reduce la cantidad de datos a procesar y la vez agrega generalidad, se seleccionan como nuevos aquellos que lleven 28 días o menos en el catalogo entre el 15 de Junio y el 13 de Julio del año 2020 (4 semanas de interacción), el preprocesamiento es el mismo, se consideran sólo los productos publicados dejando el último título presente en la página. El texto a utilizar debe estar en minúsculas y sin tilde, el separador de atributos es no convencional (en caso que los títulos contengan separadores comunes, como comas, guiones o puntos).

Para la selección del *embedding* se utilizan un total de 55.000 productos de los cuales 32.000 corresponden a históricos y 23.000 son considerados como nuevos. El primer paso es agrupar todos los atributos del producto a un solo *string*, tanto para los productos históricos como nuevos. El texto de cada producto pasa por los 2 *embeddings*, quedando representados en vectores de dimensión 512 (esta la dimensión de salida por defecto del *Universal Sentence Encoder*, se escoge el mismo largo para el *Word2Vec*). Luego se calcula la distancia coseno entre los vectores de los productos nuevos con los históricos, para no exceder la capacidad de memoria RAM del equipo, sólo se almacenan los 5 productos históricos más cercanos.

La decisión final es una métrica de conteo, al no tener un *target* definido, se opta por lo simple, donde a un producto nuevo se espera, le asocien la interacción de productos pertenecientes a la misma categoría o al menos una categoría cercana. Dicho esto, se consideran 2 niveles de cercanía, la clase y la subclase, siendo la subclase el nivel mas específico y por ende más cercano, a modo de ejemplo, se tiene un *smartphone* y un chip de telefonía *Claro*, ambos son de la clase “telefonía”, pero el celular cae en la subclase “prepago”, mientras que el chip cae en la subclase “plan”. El conteo de clase y subclase se define como: se considera un *match* de clase y subclase si, al menos 1 de los 5 productos más cercanos elegidos (históricos) pertenece a la misma categoría que el producto nuevo. Finalmente aquel *embedding* que tenga mayor número de *matches* será el elegido.

3.3. Cálculo de la distancia coseno e iteración diaria

Seleccionado ya el *embedding*, se hace un nuevo cálculo de la distancia coseno con la totalidad de los productos entre el 15 de Junio y el 13 de Julio de 2020, al igual que en el caso de la selección del *embedding*, se utiliza el título más reciente. El *string* a vectorizar esta compuesto por el título, la marca y el color del producto. Para la vectorización se utilizará aquel que tuvo mejor rendimiento en la prueba anterior, una vez vectorizados los productos, se procede a calcular la distancia coseno entre todos los productos, registrando la id de los 1000 productos más cercanos. Se crea una tabla que almacena la id del producto nuevo, las id de los 1000 productos más cercanos junto a su distancia correspondiente, esta tabla luego se exporta al *Storage* para ser utilizada en la operación siguiente.

El proceso de interacción diaria utiliza 3 tablas, el catálogo, la tabla de las distancias coseno y la tabla del *funnel*. Antes de realizar el cruce de las tablas, se hace una corrección a la tabla del *funnel*. En la actividad del párrafo anterior, el cálculo de la distancia coseno se hizo con los productos de todo el periodo, sin embargo, al tratarse de productos nuevos, hay productos que fueron apareciendo a medida que transcurrían los días, por lo que se da el caso que un producto *a* tiene como más cercanos productos que aparecieron en días posteriores. Esta situación no tiene sentido analizarla en un sistema *online*, puesto que la interacción que se le asignaría al producto *a* podría estar basada en un producto actual inexistente y aunque se tratase de un sistema *offline* también carece de sentido, dado que se tendría que indexar como interacción nula. Dicho esto, es importante que la interacción de los productos más cercanos sea solo la del pasado, luego la corrección realizada en el *funnel*, no es otra más que filtrar por la fecha del día. Para dejar más clara la situación anterior:

1. Primero se elige una fecha (entre el 15 de Junio y el 13 de Julio).
2. Segundo se filtra el catálogo y *funnel* según el día, de manera que sólo se pueda tener la interacción de los productos presentes.
3. Tercero se cruza el catálogo con el *funnel* filtrado y con la tabla de distancias, quedando en operación sólo los productos con interacción presente.
4. Cuarto se elige un producto del catálogo, se cruza con la tabla de distancias, se extraen los 1000 productos más cercanos y se busca la interacción que tuvieron ese día en la tabla del *funnel*, guardando la interacción de los 100 más cercanos (dividido en 2 partes, el *top 5* y el resto).
5. Se repite el cuarto paso hasta recorrer todos los productos del día.
6. Se repiten los pasos anteriores hasta haber recorrido los 28 días correspondientes.

El resultado que se obtiene es una tabla que tiene los productos del día, los 5 más cercanos con su respectiva interacción del día y los restantes 95 también con su interacción, se puede dar el caso que el producto cercano estuviera publicado un día pero que no tuviera presencia en el *funnel*, para dicho caso se asume interacción nula, indexándola con un 0. Dicho sea de paso, la interacción rescatada del *funnel* son, las impresiones que tuvo un producto, las visitas que recibió, la cantidad de veces que se compró y el total en ventas, estas son las interacciones requeridas para evaluar el desempeño según las métricas de negocio mencionadas en el capítulo 2. La tabla de interacción también se exporta al *Storage* para ser utilizada en la fase final.

3.4. Evaluación

El último paso de todo el trabajo corresponde a la evaluación del modelo, que estará basada en los KPI's del capítulo anterior. Una empresa, ya sea extranjera o nacional, pequeña o grande, siempre buscará lo mismo, generar ganancia o utilidades, y la incorporación de productos nuevos también se realiza con la finalidad de generar ganancias (si un producto no genera ganancias, es preferible sacarlo del mercado a gastar dinero para tenerlo almacenado). Asumiendo que todo producto va a generar ganancia, es importante la ubicación del producto en el listado, por lo tanto, a modo de evaluación para determinar la posición del

producto nuevo, se va a medir que tan alejado está el producto de sus 100 más cercanos semánticamente, tal distancia se calcula con CR, CTR y RPV. De esta forma se puede ver la relación que hay entre similitud semántica y económica.

Para realizar la evaluación, basta con importar la tabla de interacciones de la actividad anterior, se debe definir un periodo de análisis y calcular el CR, CTR y RPV de todos los productos. Se pueden analizar 2 casos, tanto a nivel de producto como a nivel de clase, para el análisis a nivel de categoría, se pretende ver un histograma de distancias de cada KPI, mientras que en el análisis de producto se pueden ver casos algunos casos particulares. La totalidad de los experimentos está detallado en la sección 3.6.

3.5. Plataforma

Las plataforma principal a utilizar corresponde *Google Cloud Platform* (GCP), es una plataforma de *cloud computing* que provee Google, la selección de esta plataforma por sobre otra es debido a que Falabella ya utiliza de sus servicios, por lo que la adición de tareas a un proyecto, no requiere de infraestructura extra ni tiempo de integración. GCP permite que un mismo usuario pueda tener acceso a múltiples proyectos, por lo que es fundamental tener claros los permisos y restricciones que tiene cada participante en cada proyecto, estos permisos incluyen también las herramientas a las cuales se tiene acceso.

GCP tiene muchas herramientas, todas desarrolladas para realizar diversas tareas, sin embargo, sólo se dará énfasis es aquellas utilizadas en esta memoria de título, dichas herramientas son: *BigQuery*, *AI Platform* y *Storage*. *BigQuery* es la herramienta que tiene Google para administrar y consultar las bases de datos, la cual es restrictiva a la plataforma, es decir, no se pueden hacer consultas a bases que no estén alojadas en *BigQuery*. El flujo de la información comienza con seleccionar el proyecto, luego se selecciona el conjunto de datos y luego la tabla (esto para subir y administrar los datos), en el caso que se quiera hacer una consulta, se hace mediante SQL en la pestaña para redactar una consulta.

Es en *BigQuery* donde se realiza la exploración de las fuentes y donde se determinan los datos que se utilizaran en el modelo. Luego está *AI Platform*, y como su nombre lo indica es la plataforma dedicada para las labores de inteligencia artificial, también ubicada dentro del menú principal de GCP. De esta sección se ocupan 2 herramientas: *Notebooks* y *Jobs*, *Notebooks* permite crear un entorno virtual con las especificaciones de CPU y RAM requeridas (también se puede incorporar el uso de GPU), el entorno virtual creado abre una pestaña del tipo *JupyterLab*, el uso principal que tiene es la de pruebas y prototipos de código. *Jobs* también es una herramienta para códigos, pero que se enfoca principalmente en la ejecución y como tal, carece de interfaz para la escritura y desarrollo de modelos. La ejecución de un *trabajo* se puede realizar mediante 3 formas, la interfaz de *Jobs*, mediante la consola de GCP o conectando la cuenta de Google al computar y mediante la consola de equipo remoto, en todos los casos el código se ejecuta en la nube. Una diferencia importante al momento de utilizar ambas herramientas es que *Notebooks* cobra por el tiempo que se mantiene la máquina virtual encendida, mientras que en *Jobs* se cobra por tiempo de máquina operativa, esta diferencia es importante, pues la maquina de *Notebooks* son recursos reservados al usuario y cobra a pesar que no se esté utilizando, y las máquina de *Jobs* son volátil por lo que al terminar la ejecución se elimina todo dato almacenado (incluida la máquina), por este motivo

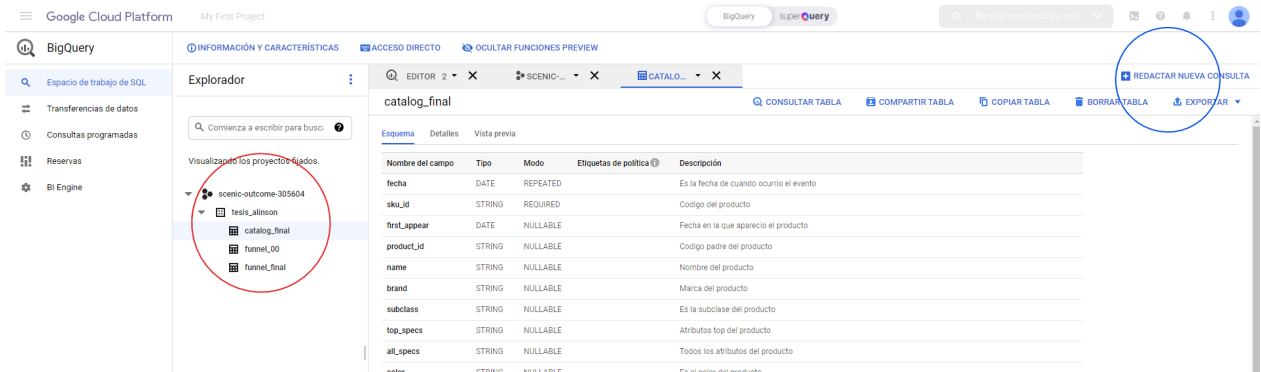


Figura 3.2: Ejemplo de como se ve la herramienta *BigQuery*, en el circulo rojo se encuentra el desglose mencionado para subir y administrar datos, en este caso *scenic-outcome-305604* es el nombre del proyecto, *tesis_alinson* es el nombre del conjunto de datos y lo último corresponde a las tablas. Para realizar una consulta se debe hacer clic sobre el circulo azul, lo que despliega una pestaña nueva para realizar la *query* correspondiente en SQL.

es que se utiliza *Storage*.

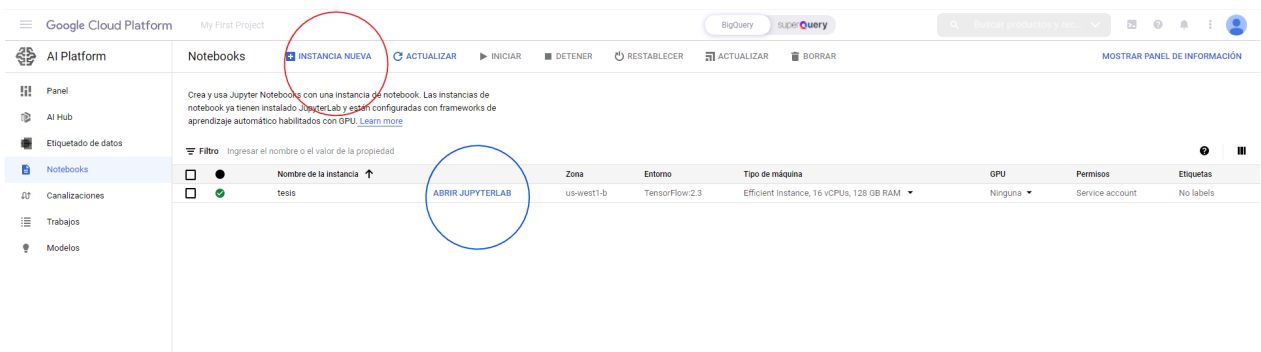


Figura 3.3: Ejemplo de como se ve la herramienta de *Notebooks*, encerrado en rojo esta la opción que permite crear un entorno nuevo. Y una vez creada la máquina, en azul está como acceder a ella, las especificaciones de la máquina están especificadas en las columnas adyacentes.

Storage es la última de las herramientas que se utilizan para el desarrollo de esta memoria y como su nombre lo indica se utiliza para el almacenamiento. *BigQuery* también es para almacenar pero solo en formato de bases de datos, mientras que en *Storage* se pueden almacenar datos de cualquier tipo. Para el uso de *Storage* se deben crear un *Bucket*, donde la única restricción importante es el nombre (este debe ser único, por lo que no pueden existir 2 *buckets* con el mismo nombre dentro de la plataforma de GCP), una vez creado el *depósito* se pueden agregar carpetas, subcarpetas y archivos a gusto.

Con respecto a que actividades se realizan en que herramienta, como se menciono con anterioridad, todo el análisis de las bases de datos se realiza en *BigQuery*. Son 3 las actividades principales que se realizan con el modelo, la primera de ellas es el cálculo de las distancias,

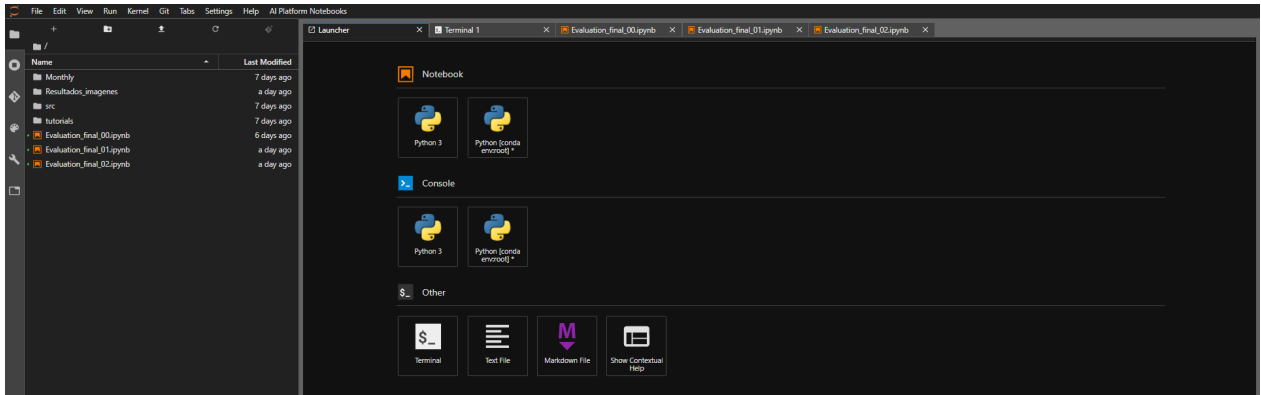


Figura 3.4: Ejemplo de como se ve la ventana de *JupyterLab* al iniciar la máquina virtual desde *Notebooks*.

ID de trabajo	Tipo	HyperTune	Parámetros de HyperTune	Métrica objetivo	Fecha de creación	Tiempo transcurrido	Registros	Etiquetas
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214727	Entrenamiento de código personalizado	No			22 feb. 2021 21:47:38	3 h 47 min	View Logs	
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214702	Entrenamiento de código personalizado	No			22 feb. 2021 21:47:15	3 h 36 min	View Logs	
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214628	Entrenamiento de código personalizado	No			22 feb. 2021 21:46:41	3 h 17 min	View Logs	
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214557	Entrenamiento de código personalizado	No			22 feb. 2021 21:46:11	3 h 42 min	View Logs	
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214527	Entrenamiento de código personalizado	No			22 feb. 2021 21:45:41	3 h 40 min	View Logs	
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214458	Entrenamiento de código personalizado	No			22 feb. 2021 21:45:12	3 h 47 min	View Logs	
models_n1_highmem_8_0_closer_products_per_day_01_20210222_214429	Entrenamiento de código personalizado	No			22 feb. 2021 21:44:43	3 h 36 min	View Logs	

Figura 3.5: Ejemplo de como es la herramienta de *Jobs*, en rojo está el histórico de trabajos realizados, haciendo clic en el nombre se despliega una ventana con el gasto incurrido. En azul están los *logs* del proceso, haciendo clic se despliegan en una ventana aparte.

la segunda es el cálculo de la interacción diaria y la tercera es el análisis de resultados. De estas actividades, toda la prueba de código y evaluación se realiza en *JupyterLab*, mientras que el el cálculo de las distancias y la interacción diaria se hace mediante *Jobs* (se paraleliza el trabajo y se reducen costos).

3.6. Pruebas a realizar

Lo último que queda por detallar son las pruebas a realizar, dichas pruebas corresponden a la última parte del trabajo detallado con anterioridad, sólo consideran la fase de evaluación, posterior la iteración diaria. Siguiendo el argumento que lo más importante para la empresa es la ganancia, se van tomar el *top 10* de categorías en 2 casos: las primeras 10 categorías en conversión y las primeras 10 en cantidad de ventas de producto por boleta distinta (a modo de visualización, sea un producto p que tiene 100 ventas de unidades, no es lo mismo que una persona compre los 100 a que 10 personas compren 10 cada uno, el segundo caso tiene mayor peso pues tiene mayor cobertura de clientes, y es precisamente este al que se refiere como cantidad de venta por boleta). Para ambos casos se realizarán las siguientes pruebas:

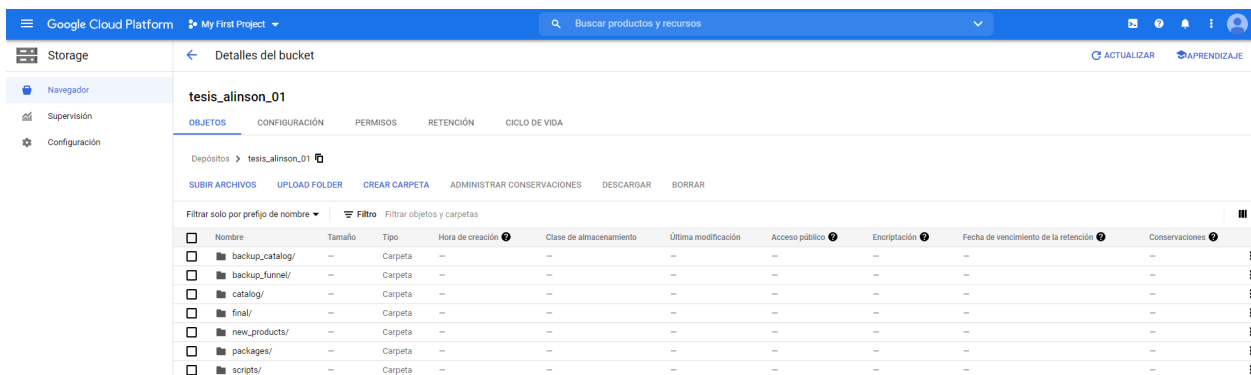


Figura 3.6: Ejemplo de como se ve la herramienta *Storage*

1. Del *top* 10 de categorías, se va a filtrar por los primeros 10 productos en desempeño según conversión por cada categoría, a cada producto se le va a calcular la distancia promedio (según KPI) con sus 5 más cercanos. Se ordena la lista según conversión y distancia.
2. Del *top* 10 de categorías, se genera un gráfico de distribución de distancias y por KPI, diferenciando entre la distancia de los primeros 5 productos con los restantes 95 cercanos (cola).

Importante mencionar que el tiempo total considerado es la interacción de 4 semanas, y con la finalidad de ver el impacto temporal es que también se analizarán los casos anteriores para periodos de 2 y 3 semanas.

Capítulo 4

Resultados y análisis

En este capítulo se presentan los resultados obtenidos, con sus respectivos análisis. En primer lugar se presentan los resultados obtenidos de la exploración de las bases de datos, sus respectivas limitaciones y consideraciones para ser utilizados en los modelos. En segundo lugar se presentan los resultados de la selección del *embedding* entre el *Word2Vec* y el Universal Sentence Encoder. Y en tercer lugar están los resultados obtenidos del modelo, se muestran resultados visuales para los casos propuestos en la metodología (resultados por producto y por categoría).

4.1. Resultados de la exploración de las bases de datos

La exploración en la base de datos fue un proceso de iteración, enfocado primeramente en cuantizar y calificar la información disponible, dado que se trata de datos confidenciales de Falabella, solo se presentan algunas de las acciones. Explorando el catálogo, se encontró un total de **459.891** productos publicados entre el 15 de Junio y el 13 de Julio, dentro de las anomalías encontradas, estaba el caso de productos con 2 o más sku padres o productos con 2 o más categorías padres asociadas, otra anomalía a considerar era el caso de los productos que le cambiaban el sku durante este periodo. El total de productos que presentaba alguno de estos casos ascendía a 1.808, que representa cerca del 0.4 % del número inicial, dado su bajo porcentaje, se deciden eliminar del registro, quedando con un total de **458.083** productos para las siguientes operaciones. Respecto a los resultados y consideraciones del *funnel*, sólo mencionar que se aplicó un filtro por tipo de lista, es decir, se ignoraron las interacciones que venían como resultado del buscador, considerando solamente las pertenecientes a las categorías de la página web.

4.2. Resultados de la selección del *embedding*

Como se había mencionado en la sección de metodología, la selección del *embedding* se realiza mediante un conteo de aciertos de clase y subclases de los productos nuevos con respecto de sus pares cercanos históricos, el resultado de esto se puede observar en la Tabla 4.1, donde se observa que tanto para la clase y la subclase, el *Universal Sentence Encoder* tiene una mayor cantidad de aciertos. Esto se debe a como están contruidos y entrenados,

para el caso del *Word2Vec*, se requiere de un diccionario y está basado en las distintas palabras utilizadas en Falabella, mientras que el Universal Sentence Encoder corresponde a una red neuronal entrenada para ser utilizada en *transfer learning*[19] de otras aplicaciones de NLP, luego es menos restrictivo.

Tabla 4.1: Comparativa entre el *Word2Vec* y el *Universal Sentence Encoder*

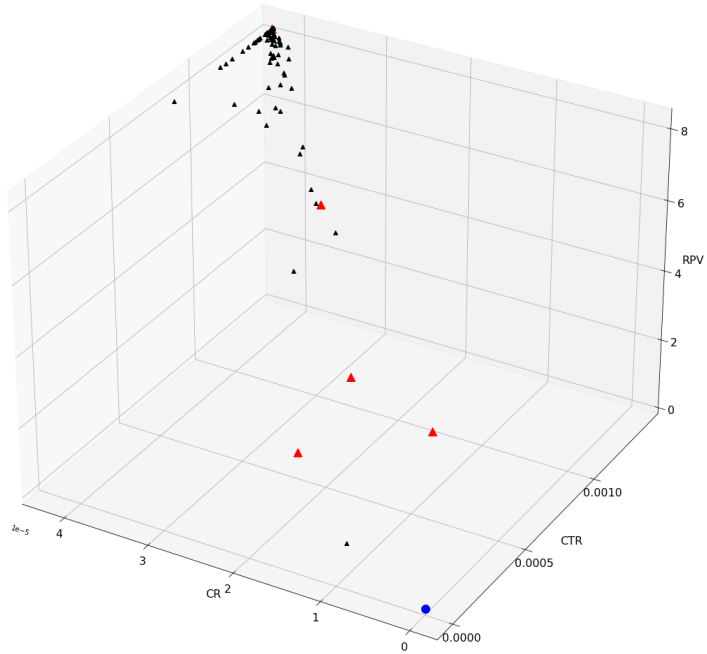
	Word2Vec	Universal Sentence Encoder
Clase	15.706	17.018
Subclase	14.212	15.904

Definido ya el *embedding* y según lo comentado en la metodología, serían 4 los casos a analizar, tanto para el nivel de producto como el nivel de categoría. Se decide elegir las primeras 10 categorías superiores en conversión y 10 superiores en ventas por boleta para reducir el tiempo de computo, y también está el hecho que a la empresa le es de mayor interés analizar los productos que más venden por sobre la cola (la cola igual guarda valor, pero no se harán análisis de dichas categorías). Se entenderá por cola a todos los productos cercanos que no sean parte de los primeros 5, en este caso abarca desde el sexto al centésimo producto cercano, y esta definición de cola sólo aplica a este caso puntual.

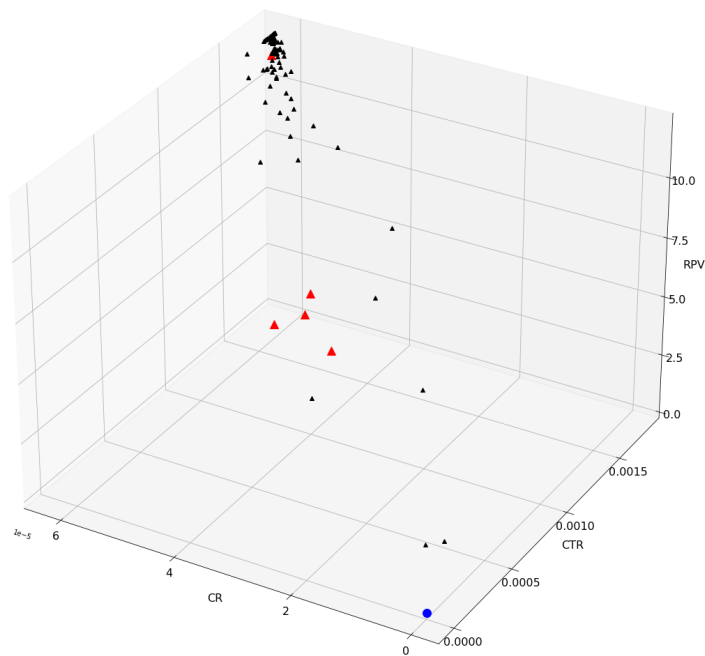
4.3. Resultados del modelo

De los resultados por producto obtenidos y para las variaciones realizadas, se observa que no existe una clara relación entre la distancia semántica utilizada y la distancia obtenida por los indicadores de rendimiento, para ninguno de los casos donde se consideren las primeras 10 categorías ordenadas por cantidad de venta, ni para el caso donde se analizan los *top* 10 productos de las categorías ordenadas por conversión. De igual forma se analizan las interacciones de los productos cercanos para periodos de 2, 3 y 4 semanas, como resultado de esto mismo se observa que la distribución de los productos tiende a mantenerse en el tiempo y que los productos de la cola tienden a acercarse al producto nuevo a medida que aumenta el lapso de tiempo. A modo de ejemplo se presentan la Figura 4.1, donde se distingue que la distribución de los productos cercanos no varió mayormente en consideración a este punto. Dentro de los casos que no se rigen a esta regla se encontraron principalmente perfumes, la distribución de cercanos se vio afectada debido a la baja disponibilidad que tenían en el periodo.

El caso particularmente llamativo, es donde las categorías y productos se ordenan según la conversión, en este caso se encontró que varios productos de la categoría lavadora, en promedio, la interacción de los 5 más cercanos coincidía con la cercanía según KPI, pero dado que se analizó sólo el *top* de la categoría, no es información suficiente para afirmar que este comportamiento se replique para el resto de productos.



(a) Lavadora Samsung con 2 Semanas de interacción



(b) Lavadora Samsung con 4 Semanas de interacción

Figura 4.1: Se muestra la distribución espacial que tienen los productos más cercanos. El punto azul corresponde a la referencia y se trata del producto nuevo, los 5 triángulos rojos corresponden a los 5 productos más cercanos semánticamente, los triángulos negros representan desde el sexto al centésimo producto más cercano. En el caso (a) se considera una interacción de 2 semanas y en el caso (b) una interacción de 4 semanas.

La Figura 4.1, muestra un ejemplo de como se distribuyen los productos más cercanos en el espacio de los KPI's, en los ejes se encuentra la conversión (CR), el *click-through rate* (CTR) y la venta por visita (RPV). En dicha imagen se observa que los productos más cerca-

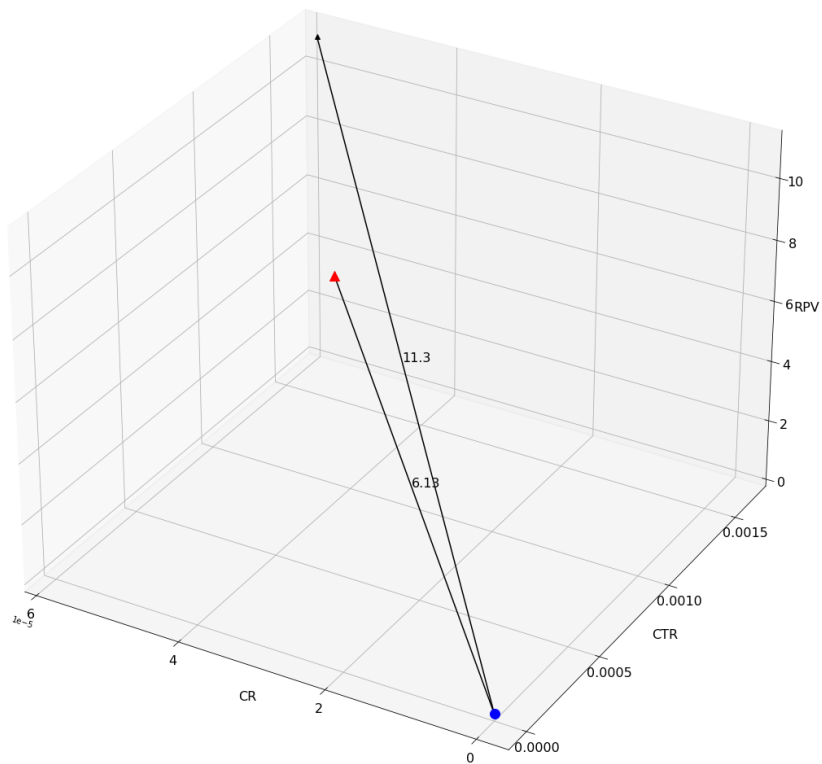


Figura 4.2: Lavadora Samsung con 4 semanas de interacción. Corresponde al mismo caso que la Figura 4.1 (b), donde ahora todos los puntos se representan con su respectivo centro, en rojo está el centro de los 5 productos más cercanos y en negro esta la cola. En azul se mantiene el producto de referencia.

nos semánticamente, corresponden también a los más cercanos en cuanto a rendimiento, sin embargo este resultado es particular. Sólo a modo de visualización, se presenta la Figura 4.2, donde ahora están los centros correspondientes, con sus respectivas distancias del producto nuevo. El periodo que se muestra en dicha imagen corresponde a 4 semanas.

Los casos particulares permiten hacerse una idea del comportamiento que podrían tener ciertos productos de ciertas categorías y siguiendo ciertos supuestos, lo cual no es representativo a la hora de trabajar con una gran cantidad de productos, por lo que no aplica para el catálogo de Falabella.com. Dicho esto es que está la segunda prueba de datos, en este caso se van a evaluar las primeras 10 categorías ordenadas según conversión y según cantidad de ventas por boleta (mismos criterios de categorías que en los casos particulares).

De los resultados observados, hay 2 aspectos importantes a mencionar, primero, para las categorías que fueron ordenadas según conversión, se da mucho el caso que presentan pocos productos, esto desencadena en que no hay un patrón claro para las distribuciones, a pesar de esto el evento que mayor ocurrencia fue que los productos más cercanos semánticamente eran los más lejanos en cuanto a rendimiento (comportamiento esperado en categorías específicas de baja impresión). No es de extrañar, y el ejemplo de esto es la categoría *secadoras*, donde no se superan los 30 productos, cuando un cliente entra a navegar en Falabella.com lo hace

con la intención de cotizar o comprar, en el caso de las secadoras al ser de uso específico, el cliente tiene claro cuales son sus requerimientos lo que se traduce en tiempos cortos de navegación y decisiones claras. Que la categoría sea específica también indica poco tráfico de público general, disminuyendo las impresiones se aumenta la conversión, a su vez la poca cantidad de productos induce la aparición de productos de otras categorías, categorías cuyo rendimiento puede estar muy alejado del producto nuevo. Pese a esto, en este tipo de categorías se tiene el mayor rendimiento en cuanto a venta por visita. El ejemplo gráfico de estos casos se pueden observar en la Figura A.1 y en la Figura A.2 ubicadas en el Anexo.

El segundo aspecto está relacionado con las categorías que fueron ordenadas según la cantidad de venta por boletas, la mayor diferencia con los casos anteriores es que estas categorías presentan una mayor interacción de los productos, hay 2 casos que son los más frecuentes, el primero de ellos, es que la distribución de distancias no varía mucho entre los primeros 5 con respecto a la cola, y el segundo caso es que el *top* si está mas cerca que la cola. El primero de los casos se puede encontrar en la categoría de *smartphones*.

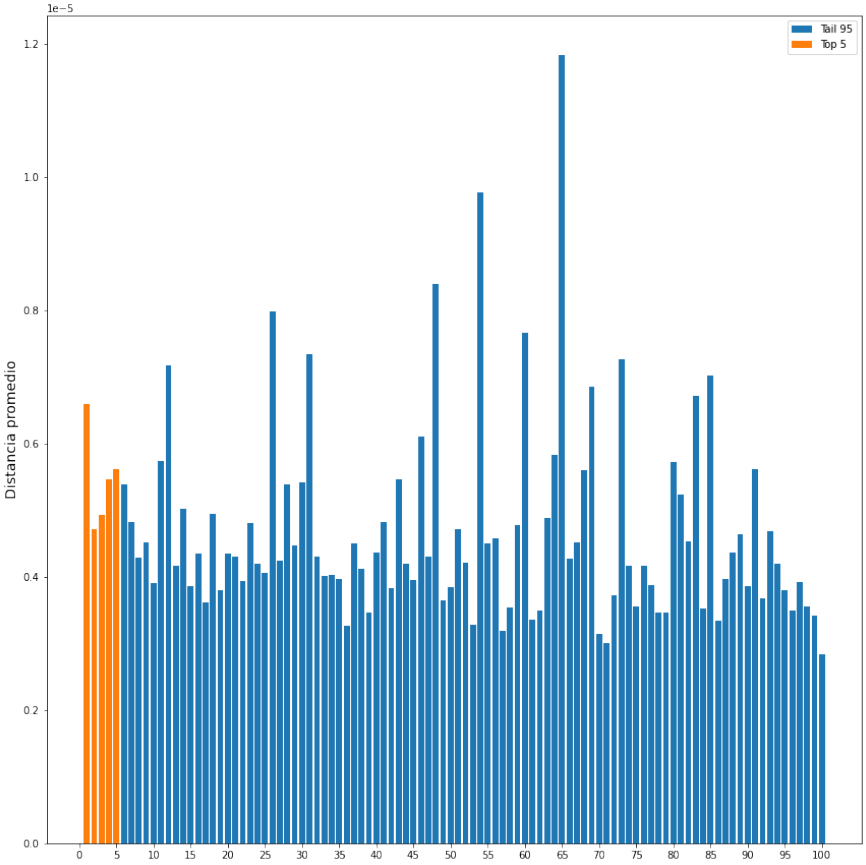


Figura 4.3: Distribución de distancia promedio de la categoría *smartphones*, según conversión. Los números indican la posición de cercanía semántica de los productos, donde 1 corresponde al más cercano y 100 al más lejano. El tiempo corresponde al periodo de 2 semanas

De la figura anterior se observa que la distancia de conversión del producto con sus cercanos no tiene tendencia a disminuir o aumentar a medida se desplaza de los más cercanos a los más lejanos, este comportamiento puede conducir a que un producto nuevo, de posible

alta venta, quede en un lugar muy cercano a la cola (lo que va a disminuir su exposición y como tal su utilidad) o que el producto nuevo, de posible baja venta, quede entre los primeros lugares (lo que se traduce en disminuir la calidad de la experiencia del usuario), cualquiera fuera la situación es preferible el segundo de los casos, ya que el aumento de impresiones dejará que el modelo de Falabella le corrija la posición. En este caso también se considerará otro KPI para ver como actuar con la categoría (revisar Figura A.3 en Anexos). El segundo caso, menos frecuente, donde el *top* está más cerca de la cola, se puede encontrar en la categoría *audífonos bluetooth*, respecto a este comportamiento, no se observó una relación clara con las otras categorías, sólo hecho que todas correspondían a la línea de tecnología.

Con respecto a la variación de la distribución en el tiempo, acotar que no hay una resolución clara, se dieron varios casos: casos donde la cola inicialmente estaba más lejos, pero que al tomar un mayor periodo de tiempo todas las posiciones tenían un rendimiento parecido, casos donde el *top* estaba más lejos y que al tomar las 4 semanas pasaba a estar más cerca. También está el caso donde no hubo mayor variación de la distribución al pasar de 2 a 4 semanas. Un ejemplo visible de mejora de rendimiento se puede observar en la categoría de *smartphones*.

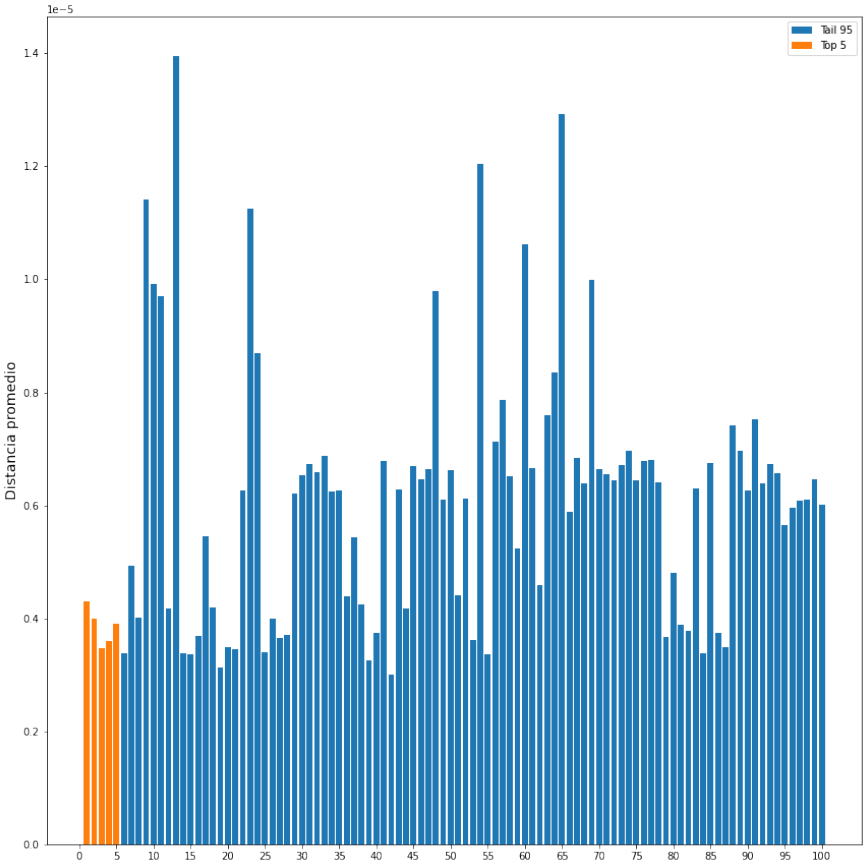


Figura 4.4: Distribución de distancia promedio de la categoría *smartphones*, según conversión. Los números indican la posición de cercanía semántica de los productos, donde 1 corresponde al más cercano y 100 al más lejano. El tiempo corresponde al periodo de 4 semanas

Comparando la Figura 4.3 con la figura anterior se observa que la distancia promedio de los 5 productos más cercanos disminuye con respecto al periodo de 2 semanas, para la

cola ocurre que algunas posiciones se acercan, otras se alejan y también hay algunas que no sufren variación. Este caso de mejora indica que si se quisiera usar el modelo en la categoría de *smartphones* es recomendable un periodo de interacción de 4 semanas por sobre uno de 2 semanas.

El último análisis que originalmente no estaba contemplado es el gasto en que se incurre en realizar todo este proceso, la plataforma de GCP entrega un desglose detallado de los gastos y un estimado del gasto restante del mes. El reporte de gastos se realiza por herramienta utilizada, y realizar todo este proceso 1 vez, costó **226 USD**. Pensando en dejar este proceso operativo para la empresa, se deben descontar los gastos de prototipo y resultados, se deben mantener los gastos de ejecución de los *Jobs*, el gasto de almacenamiento en *Storage* y los gastos en las consultas de *Bigquery*. Pensando que el cálculo de la distancia coseno es lo más caro, se recomienda realizar de 1 a 2 veces por semana, de igual manera el cálculo de interacción diaria. Un gasto mensual aproximado se presenta en la siguiente tabla:

Tabla 4.2: Gasto mensual estimado

Operación	Gasto por operación (USD)	Cantidad a la semana (USD)	Gasto mensual (USD)
Cálculo distancias coseno	100	2	400
Iteración diaria	48	2	384
Consulta de queries	0.47	20	37.6
Almacenamiento Storage	$\sim 10 + 5^*$	-	$10 + 5^*$
Almacenamiento BigQuery	$\sim 2 + 5^*$	-	$2 + 5^*$
Total estimado			$843.6 + 10^{**}$

De la tabla anterior corresponde aclarar 2 puntos, (*) lo que viene después del signo más (+) es el extra estimado que se agregará mensualmente debido al aumento de la base de datos de *BigQuery* y de *Storage*. (**) **843.6 USD** es el gasto que se tendrá al primer mes después de tener el modelo implementado, sumándole 10 USD en gastos cada mes. Importante mencionar que estos gastos son fijos, cualquier operación de prueba incurre en gasto extra, a pesar de esto mejoras en el modelo pueden traer rebaja en los precios.

Capítulo 5

Conclusión y trabajo futuro

5.0.1. Conclusión

El presente trabajo de título consistió en la realización de un modelo el cual fuera capaz de suplir una de las falencias actuales del modelo de ranking en Falabella.com, dicho problema está acotado a la implementación del modelo, que toma los datos de interacción de los clientes que navegan en la página web con los productos. Luego el trabajo se desarrolla en la búsqueda de un modelo que fuera capaz de identificar los productos nuevos y atribuirles los datos de interacción para poder ser utilizados en el recomendador. Para realizar esta labor es que se siguen una serie de pasos, que comienza por la exploración de las fuentes disponibles, para ver el tipo y calidad de los datos disponibles, sigue con la selección el *embedding*, paso fundamental para poder trabajar con el texto disponible, continua con la selección de las métricas para cuantificar el desempeño del modelo, para culminar con la implementación del modelo.

Los *embeddings* obtenidos de los algoritmos, *Word2Vec* y *Universal Sentence Encoder*, fueron los que se compararon mediante un métrica de conteo de clases y subclases acertadas. Como resultado se tuvo que el *Universal Sentence Encoder* fue el que tuvo mayor cantidad de aciertos, resultado esperable dada la forma en como estaba creado (mediante una red neuronal entrenada para ser usada como *transfer learning* en aplicaciones de texto). Una de las ventajas de usar el *Universal Sentence Encoder* es que no requiere de crear un diccionario de las palabras a utilizar, ya que si se usara el *Word2Vec* se tendría un error cada vez que apareciera una palabra no registrada previamente. Algo que no se utilizó, pero que podría ayudar a mejorar el rendimiento (en cuanto a hacer *match* de la clase), es entrenar el *embedding* para los productos de Falabella.com.

Con respecto a las pruebas realizadas, estas se enfocaron particularmente en revisar las categorías que tuvieran mayor conversión, por lo que se vieron 2 casos, las categorías que vendieron más unidades de productos en distintas boletas y las categorías que estrictamente fueron las *top* en conversión promedio (todo esto en el periodo de 1 mes, que abarcaba desde mediados de Junio a mediados de Julio del año pasado). Para todas estas categorías se analizaron tanto a nivel general como particular, del caso particular no se pueden obtener conclusiones consistentes debido a que el comportamiento de los productos era muy variado entre categorías e intra categorías, este tipo de análisis se podía realizar cuando la categoría no presentaba una gran cantidad de productos, pero considerando que la cantidad

de categorías en Falabella es gigante, esta aproximación se recomienda realizar cuando se tenga un interés particular en alguna categoría, por ejemplo alguna campaña de ropa. En el caso general, los resultados fueron similares, se concluye que no hay una relación clara entre la distancia semántica y la distancia por rendimiento, ya que se encontraron todos los posibles escenarios. En el caso de las categorías que tienen alta conversión y baja cantidad de productos, por lo general estas son poco variadas y no pasan de 1 página (en cantidad de productos), por lo que no se recomienda utilizar ningún modelo, ya que el cliente tiene pocas opciones lo que no dificulta su elección. Para el caso de las categorías que tienen alta cantidad de productos y alta cantidad de ventas en boleta, se recomienda la utilización del modelo, porque a pesar de mostrar un error en la diferencia de rendimiento, si se consideraba la venta por visita promedio (Figura A.3), el modelo posicionaría a los productos nuevos entre las primeras posiciones, aumenta la exposición y dejando que el modelo operativo actual se encargase del nuevo ranking. Otra recomendación es revisar cual métrica es la mejor para realizar la evaluación, el análisis se enfocó en la conversión (venta sobre impresiones), pero en algunos casos se podría utilizar la venta por visita (por ejemplo en productos exclusivos).

Originalmente no estaba planificado, pero el hecho que todo este proceso tome 850 USD aproximadamente, es un tema a considerar, este valor corresponde aproximadamente a 630.000 pesos chilenos, que corresponde a casi un tercio de sueldo de 1 ingeniero¹, luego si muchos de estos ingenieros se dedican a realizar el trabajo de posicionamiento manual de los productos nuevos, a la empresa le es más beneficioso destinar dichos trabajadores a la mejora del modelo de productos (el hecho que tengan el conocimiento sobre productos nuevos, les permite desarrollar de mejor manera el modelo).

Para culminar, se cumplió gran parte del objetivo general y de los objetivos secundarios, el punto que no se pudo cumplir fue el de atribuir los datos de interacción necesarios (en los productos nuevos) para ser utilizados en el recomendador de Falabella, al igual que tampoco se pudo evaluar el desempeño del recomendador, todo esto debido a la logística llevada a cabo en la empresa.

5.0.2. Trabajo futuro

Como trabajo futuro se recomienda la realización de más pruebas y también de más análisis, en esta memoria sólo se abordaron las primeras 10 categorías en venta (por unidades), pero se podían también haber analizado por venta total, de igual forma, la cola no aporta tanta ganancia como los productos *top*, pero una vista de su comportamiento pudo obtener resultados más concluyentes, sobretodo a nivel de categoría. Respecto a las métricas de rendimiento, se pudieron utilizar en conjunto con otras que también estaban presentes en la literatura (*drop rate* por ejemplo, que indica el ratio de visitantes que botan objetos del carro antes de pagar), esto permitiría medir el impacto que generan ciertos productos.

Otra prueba recomendada consiste en incrementar la cantidad de atributos a utilizar al momento de construir los *embeddings*, la utilización del título y marca es muy genérico para categorías pequeñas donde la diferenciación no siempre está en el título. Por lo que incluir color, dimensiones, talla, departamento, entre otros puede aportar a diferenciar de mejor

¹ Según una noticia de T13 <https://bit.ly/3rvzAYC>

manera los productos. Considerando este aumento de información al momento de realizar el *embedding*, también se recomienda disminuir el tamaño de salida del *Universal Sentence Encoder* con la finalidad de reducir costo computacional.

Además de manera de hacer que la interacción de los productos con los clientes sea más selectiva, se podría considerar trabajar en conjunto con el área de segmentación de clientes, de esta manera la posición del producto nuevo también dependería del tipo de cliente. Esta acción va en lineamiento que el recomendador deje de ser global y sea personalizado (por cliente).

Bibliografía

- [1] G. Lever, Centro Economía Digital CCS. Tendencias del Comercio Electrónico en Chile. presentado en EISUMMIT 2019. [En línea]. Disponible en: <https://bit.ly/2A5gEds>
- [2] Cámara de Comercio de Santiago (2020, Marzo). Informe Económico. Santiago. [En línea]. Disponible en: <https://bit.ly/3emoz4C>
- [3] Cámara de Comercio de Santiago, PMG Chile (2019, Junio). Desafíos en la evolución de las empresas hacia la transformación digital. Santiago. [En línea]. Disponible en: <https://bit.ly/2WZVvdv>
- [4] C. Lever, Cámara de Comercio de Santiago. Perspectivas económicas del sector comercio. Santiago. [En línea]. Disponible en: <https://bit.ly/2ZKGHju>
- [5] P. Resnick, H. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, pp.56-58, Mar. 1997. [En línea]. Disponible en: <https://doi.org/10.1145/245108.245121>
- [6] F. Ricci, L. Rokach, B. Saphira, “ Introduction to Recommender Systems Handbook,” *Recommender Systems Handbook*, Springer, Boston, MA, 2010, pp. 1-13 [En línea]. Disponible en: <https://doi.org/10.1007/978-0-387-85820-3-1>
- [7] C. Aggarwall, *Recommender Systems: The textbook*, Yorktown Heights, NY: Springer, 2016.
- [8] Y. Koren, “The Bellkor solution to the Netflix grand prize”, *Netflix prize documentation*, 81, 2009. <https://bit.ly/3dOjwg9>
- [9] Y. Hu, Y. Koren, C. Volinsky, “Collaborative filtering for implicit feedback datasets”, *IEEE Int. Conf. on Data Mining*, pp. 263–272, 2008.
- [10] D. Bridge, M. Goker, L. McGinty, B. Smyth, “Case-based recommender systems”, *The Knowledge Engineering Review*, 20(3), pp. 315–320, 2005.
- [11] K. Cios, R. Swiniarski, W. Pedrycz, L. Kurgan, *The Knowledge Discovery Process. In: Data Mining*, Boston, MA: Springer, 2007.
- [12] R. Kohavi, R. Longbotham, D. Sommerfield, R. Henne, “Controlled experiments on the Web: survey and practical guide”, *Data Mining and Knowledge Discovery*, 18(1), pp. 140–181, 2009.
- [13] J. Konstan, S. McNee, C. Ziegler, R. Torres, et al., “Lessons on applying automated recommender systems to information-seeking tasks”, *AAAI Conference*, pp. 1630–1633, 2006.
- [14] T. Murakami, K. Mori, R. Orihara. 2008. Metrics for evaluating the serendipity of recommendation lists. *New frontiers in artificial intelligence*. [En línea]. pp.40-46. Disponible

en: <https://bit.ly/3kmXiUa>

- [15] B. Smyth, P. McClave, “Similarity vs. Diversity”, *Case-Based Reasoning Research and Development*, pp. 347–361, 2001.
- [16] A. Krishnan, A. Amarthaluri. “Large Scale Product Categorization using Structured and Unstructured Attributes,” *arXiv preprint: arXiv:1903.04254*, 2019.
- [17] A. Botezatu, “ Ce sunt indicatorii cheie de performanta”, 2012, [En línea]. Disponible en: <https://bit.ly/3rzfD32>
- [18] M. Muntean, D. Tarnaveanu, A. Raluca. “E-commerce Performance. Shopping Cart Key Performance Indicators”, *Informatica Economică*, vol. 20, n° 1, 2016.
- [19] D. Cer, Y. Yang, S. Kong, et al., “Universal Sentence Encoder”, 2018. [En línea]. Disponible en: <https://arxiv.org/abs/1803.11175>
- [20] T. Mikolov, K. Chen, G. Corrado, J. Dean, “Efficient Estimation of Word Representations in Vector Space”, 2013. [En línea]. Disponible en: <https://arxiv.org/abs/1301.3781>
- [21] P. Sitikhu, K. Pahi, P. Thapa, S. Shakya, “A Comparison of Semantic Similarity Methods for Maximum Human Interpretability”, Department of Electronics and Computer Engineering, Tribhuvan University, 2019. [En línea]. Disponible en: <https://arxiv.org/pdf/1910.09129.pdf>
- [22] S. J. Pan, Q. Yang, “A Survey on Transfer Learning”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010. [En línea]. Disponible en: <https://bit.ly/3tVP4Gy>

Anexo A

Resultados

A.1. Distribuciones

En esta sección se presentan resultados de situaciones mencionadas en la sección de Resultados.

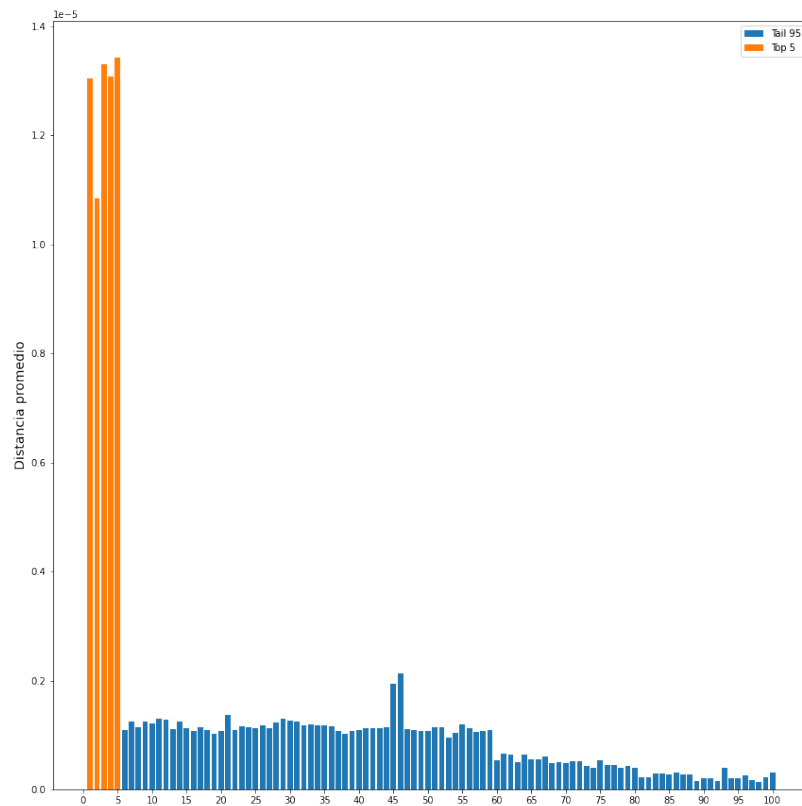


Figura A.1: En esta figura está la distribución de distancias promedio según la cercanía del producto, para la categoría secadoras. El indicador corresponde a la conversión. Los números en el eje de las abscisas indican la posición de cercanía, donde 1 es el producto más cercano y 100 es el más lejano. El tiempo corresponde a un periodo de 4 semanas

En la figura anterior se observa el comportamiento mencionado en los Resultados, al

tratarse de una categoría con pocos productos, el *top 5* más cercano está más alejado en rendimiento que la cola. Parte de este comportamiento se justifica en la baja presencia de productos en la categoría, luego no todos los productos presentes en el listado (100 más cercanos) pertenecen a la categoría y la decisión de compra del cliente ya está definida por parámetros previos, por lo que la compra será específica.

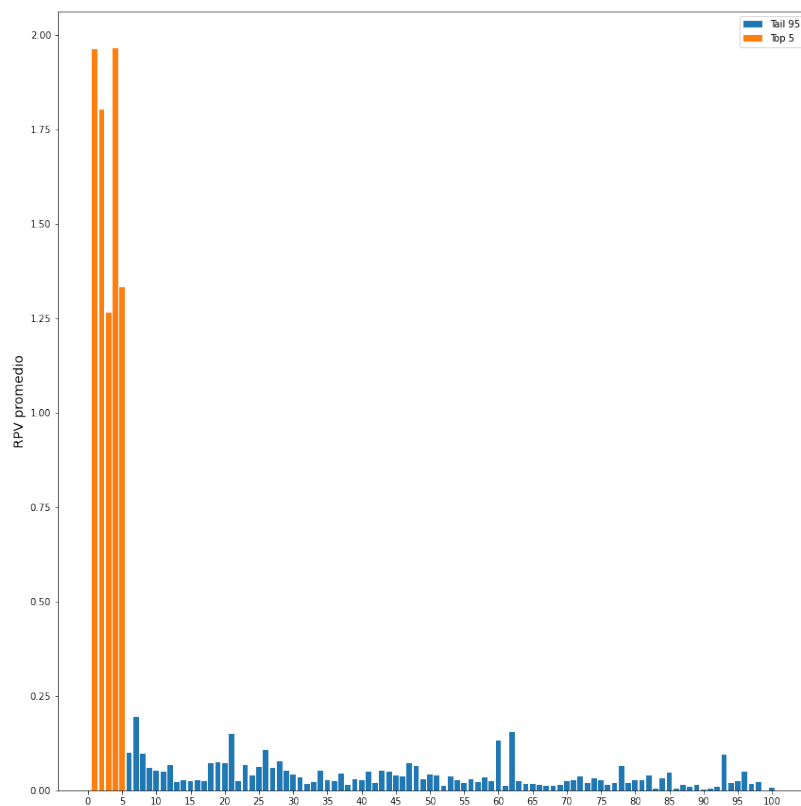


Figura A.2: En esta figura está la distribución de la venta por visita promedio de la categoría secadoras. El tiempo corresponde a un periodo de 4 semanas

En la Figura A.2 se observa que los primeros 5 productos tienen una venta por visita muy superior a la cola, por lo que a pesar de encontrarse alejado según conversión (Figura A.1), este resultado no deja de ser valioso pues indica que el *top 5* trae consigo una ganancia, luego si la métrica para indexar fuera puramente el RPV, los productos nuevos serían posicionados entre los primeros lugares de la categoría, lo que aumentaría su exposición, permitiendo que el modelo principal de Falabella tome menos tiempo en ordenarlo.

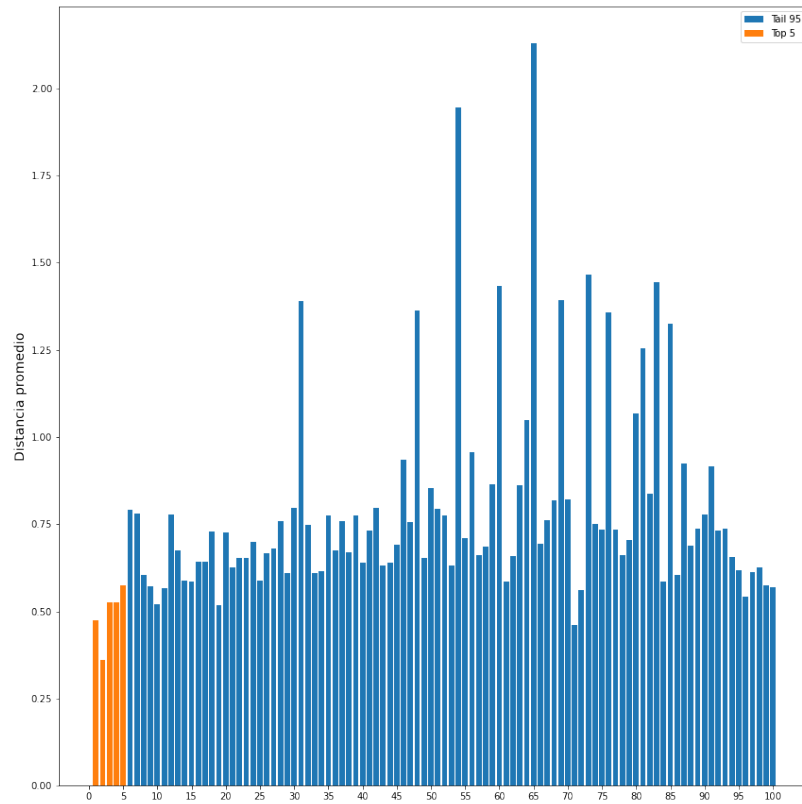


Figura A.3: Distancia promedio de RPV para la categoría de *smartphones*. El tiempo corresponde al periodo de 2 semanas

En la Figura A.3 está la distancia promedio de la venta por visita de los productos con sus 100 mas cercanos para la categoría de *smartphones*. En este caso ocurre una situación similar que en la Figura 4.3, donde la distribución no presenta un comportamiento concluyente, luego el uso de este indicador entregará resultados similares que al usar la conversión.