



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

ANÁLISIS DE BUSCADOR VISUAL PARA TIENDAS DE E-COMMERCE

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ANTONIO RODRIGO MONTALVA VARGAS

PROFESOR GUÍA:
JUAN MANUEL BARRIOS NÚÑEZ

MIEMBROS DE LA COMISIÓN:
IVÁN ANSELMO SIPIRÁN MENDOZA
LUIS EMILIO ANTONIO MATEU BRULE

SANTIAGO DE CHILE
2022

Resumen

En el presente documento se expone el trabajo realizado durante el desarrollo y experimentación de algoritmos para tres tareas relacionadas con el uso de imágenes de consultas recibidas por un motor de búsqueda visual de tiendas de e-commerce. La primera tarea consiste en la detección de comportamientos erróneos del motor de búsqueda en que este recupera sólo productos que no tienen relación al resultado esperado (*Zero-Hit*). La segunda tarea consiste en el etiquetado de imágenes de consultas recibidas por una tienda, en base a palabras presentes en su catálogo. Por último, la tercera tarea consiste en la generación de una visualización más estructurada que sirva de resumen de búsquedas visuales recibidas en un periodo de tiempo.

El problema a resolver nace de la necesidad de los dueños de tiendas que emplean al buscador, de conocer cómo es que sus clientes utilizan este servicio. Esperando que este análisis adicional sobre las búsquedas realizadas en su tienda pueda llegar a generar información a partir de la cual el dueño pueda llegar a generar acciones que beneficien el funcionamiento de su negocio.

La solución propuesta consiste, por un lado, en el desarrollo de algoritmos de clasificación que analizan las respuestas entregadas por el motor de búsqueda para determinar si esta corresponde a un *Zero-Hit* o la categoría a la que corresponde la imagen de consulta en caso contrario. Por otro lado, la solución también contempla el desarrollo de algoritmos que emplean diferentes herramientas de clusterización para la generación de grupos de imágenes de consulta, que pueden ser utilizados como resúmenes de las búsquedas recibidas por una tienda. Para cada uno de los algoritmos propuestos en este trabajo, se experimentó y probó con diferentes alternativas, de manera que la solución que se propone consta de aquellos experimentos más exitosos.

Respecto a los resultados, se concluye que se lograron cumplir parcialmente todos los objetivos planteados. Esto pues, se logró desarrollar un algoritmo que logra identificar con un 63.8 % de precisión los *Zero-Hit* de un mes de búsquedas, otro algoritmo que logra etiquetar en categorías con un 48.1 % de *Precision@5* en un mes de búsquedas y, por último, un algoritmo que, utilizando los resultados previos, logra seleccionar e identificar con una categoría a clústeres de imágenes de consulta similares. Gracias a estos tres resultados, la compañía detrás del motor búsquedas puede enriquecer más su servicio con información adicional que puede potenciar los negocios de sus clientes.

Dedicado a mi madre y padre.

Agradecimientos

Gracias a todos los involucrados, directa e indirectamente, en que haya logrado hacer este trabajo. En especial gracias al apoyo incondicional de mi familia, madre, padre y hermanos. Agradecer al Bryan y al Machuca por acompañarme en gran parte de mi aventura universitaria. Por último, agradecer al team ATR por las noches de ocio y conversación durante el último año.

Tabla de Contenido

1. Introducción	1
1.1. Contexto y motivación	1
1.2. Problema a Resolver	2
1.3. Solución propuesta	2
1.4. Objetivos	3
1.5. Metodología	3
2. Marco Teórico	4
2.1. Redes Neuronales Artificiales	4
2.1.1. Perceptrón	4
2.1.2. Convolución	5
2.1.3. Redes Neuronales Convolucionales	5
2.1.4. ResNet-152	6
2.2. Descriptores Visuales	7
2.2.1. Deep Feature	7
2.3. Algoritmos de Clusterización	7
2.3.1. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)	8
2.3.2. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)	9
2.4. Métricas de evaluación	11
2.4.1. Mutual Information (MI)	11
2.4.2. Adjusted Mutual Information (AMI)	12

2.4.3.	Rand Index (RI)	13
2.4.4.	Adjusted Rand Index (ARI)	13
2.4.5.	Precision y Recall	14
2.4.6.	Precision at k (P@k)	14
2.4.7.	Intersection Over Union (IOU)	15
2.5.	Distancia de Levenshtein	15
2.6.	Term Frequency - Inverse Document Frequency (TF-IDF)	15
2.7.	Reducción de Dimensionalidad	16
2.7.1.	Principal Component Analysis (PCA)	16
2.7.2.	t-distributed stochastic neighbor embedding (t-SNE)	17
2.7.3.	Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)	18
3.	Recopilación de Antecedentes	19
3.1.	Estado del arte	19
3.1.1.	Clasificación	19
3.1.2.	Deep Features	21
3.1.3.	Visualización de espacios de alta dimensión	21
4.	Descripción e Implementación de la Solución	22
4.1.	Descripción General de la Solución	22
4.2.	Implementación	22
4.2.1.	Conjuntos de datos	22
4.2.2.	Etiquetado de búsquedas y detección de Zero-Hits	23
4.2.3.	Clusterización de etiquetas	26
4.2.4.	Clusterización de descriptores	26
4.2.5.	Selección de clústeres	27
5.	Experimentos y Análisis de Resultados	29

5.1. Diseño de Experimentos y Evaluación	29
5.1.1. Datos para la evaluación	29
5.1.2. Experimentos	30
5.2. Resultados y Análisis	32
5.2.1. Detección de Zero-Hits	32
5.2.2. Etiqueta de búsquedas visuales	33
5.2.3. Clusterización	33
5.2.4. Selección de clústeres	34
6. Conclusiones y Trabajo Futuro	38
Bibliografía	41
Anexo A	42
Anexo B	44

Índice de Tablas

4.1. Muestra de 5 resultados más cercanos para una consulta	23
4.2. Ejemplo de similitud de palabras utilizando FuzzyWuzzy	24
4.3. Muestra de los cinco resultados más cercanos para una consulta y sus etiquetas	25
4.4. Muestra de los cinco resultados más cercanos para una consulta clasificada como Zero-Hit	25
4.5. Ejemplo de la función de distancia propuesta entre pares de etiquetas distintas	26
5.1. Resultados experimentos detección de Zero-Hits	32
5.2. Resultados experimentos etiqueta de búsquedas	33
5.3. Resultado experimentos de clusterización	34

Índice de Ilustraciones

1.1.	Ejemplo de entrada y salida para las dos formas de búsqueda visual.	1
2.1.	Diagrama de un perceptrón con cinco señales de entrada	5
2.2.	Ejemplo de convolución	5
2.3.	Ejemplo de CNN	6
2.4.	Bloque residual de una ResNet	7
2.5.	Ejemplo de clústeres detectados por DBSCAN	9
2.6.	Ejemplo de distancia de alcance mutuo	10
2.7.	MST y Dendrograma calculados por HDBSCAN	10
2.8.	Selección de clústeres	11
2.9.	Precision y Recall	14
4.1.	Diagrama de selección de clúster	28
5.1.	Muestra de imágenes de consultas durante Agosto 2021	30
5.2.	Muestra de uno de los clústeres de búsqueda seleccionados para el conjunto de datos Tienda 1	35
5.3.	Muestra de uno de los clústeres de búsqueda seleccionados para el conjunto de datos Tienda 2 - Noviembre 2021	36
5.4.	Muestra de uno de los clústeres de búsqueda seleccionados para el conjunto de datos Tienda 2 - Diciembre 2021	37
1.	Muestra de imágenes de consultas en Tienda 2 durante Noviembre 2021	42
2.	Muestra de imágenes de consultas en Tienda 2 durante Diciembre 2021	43

3.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021	44
4.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021	45
5.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021	45
6.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021	46
7.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021	46
8.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021	47
9.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021	48
10.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021	48
11.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021	49
12.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021	49
13.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021	50
14.	Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021	50

Capítulo 1

Introducción

1.1. Contexto y motivación

El e-commerce -también conocido como comercio electrónico, o comercio en línea- ha probado ser el mecanismo de compra de productos más popular y resiliente. Dada las facilidades que provee, incluso en las circunstancias actuales del COVID-19 que afectan a todo el mundo, es el único método de compra que no se ha visto afectado negativamente. Por el contrario, múltiples estudios como los realizados por Fareeha[1] y PayU[3], muestran como este tipo de comercio ha experimentado un crecimiento durante el desarrollo de la pandemia.

Impresee es una empresa dedicada a la investigación aplicada de la inteligencia artificial, que se ha especializado en la implementación de servicios de búsqueda avanzada de productos para tiendas en línea. Los tipos de búsqueda que ofrece Impresee son: búsqueda por texto, búsqueda visual (por fotografía) y búsqueda creativa (por dibujo). Estas últimas dos permiten encontrar dentro de un catálogo, los productos más similares a una imagen o a un sketch¹ dado por el usuario. En la Figura 1.1 se observan los dos tipos de búsquedas visuales que ofrece Impresee que constan de una imagen de entrada (fotografía o sketch de dos productos respectivamente), y el resultado del buscador, que corresponde a las imágenes más parecidas del catálogo de productos de la tienda.



Figura 1.1: Ejemplo de entrada y salida para las dos formas de búsqueda visual.

¹Dibujo realizado a mano por el usuario al momento de la consulta

Impresee entrega a sus clientes (tiendas de e-commerce) un reporte periódico en el cual le entrega distintas métricas del uso del servicio de buscador que les provee. La principal motivación de este trabajo es enriquecer este reporte con mayor información.

Este trabajo, permitirá organizar las imágenes de búsquedas de cada tienda para que puedan ser inspeccionadas de manera más estructurada, según parecido.

1.2. Problema a Resolver

Dentro de los reportes de búsquedas visuales que genera Impresee, para cada uno de sus clientes, sólo se ofrecen la totalidad de imágenes consultadas en los sitios de cada tienda, junto con las respuesta entregadas por el buscador para cada consulta. Pese a que esta información se encuentra ordenada por ventanas de tiempos, resulta difícil inspeccionar un periodo prolongado de tiempo y más aún cuando la tienda recibe demasiadas búsquedas. Extraer más información de las consultas podría ayudar a categorizar las diferentes imágenes que buscaron los usuarios de las tiendas y así otorgarle mayor valor al buscador.

En lo recién mencionado, se identifican principalmente dos problemas a resolver. Por un lado, se busca categorizar imágenes de consultas en función del catálogo de cada tienda. El segundo problema es la identificación de búsquedas cuya respuesta por parte del buscador no coincide con la clase o tipo del producto consultado.

La categorización de imágenes de consulta tiene como objetivo conocer los productos que más buscan los visitantes de una tienda. Estos podrían encontrarse o no en el catálogo de la tienda y es útil saber a qué clase pertenecen en ambos casos. Esto último debido a que permitiría obtener información respecto a los productos más populares dentro de la tienda y también de los productos más populares que no se ofrecen.

El término *Zero-Hit* viene de la búsqueda por texto y corresponde a aquel caso en que no hay palabra que coincida con la consulta. En este trabajo se busca una identificación de casos análogos para consultas visuales. Esta identificación no es trivial ya que es común que exista parecido en las formas y colores de dos imágenes que contienen objetos distintos, en este caso, un buscador visual determinará que las imágenes tienen una distancia muy pequeña entre ellas y por lo tanto serán consideradas como iguales. Lograr identificar estas situaciones es algo particularmente importante para Impresee ya que sirve como evaluación de su propio producto.

1.3. Solución propuesta

La solución propuesta consiste en, primeramente, desarrollar algoritmos para la detección de *Zero-Hits* y para la etiqueta en base a una palabra que esté presente en los nombres de los productos del catálogo de una tienda, de las imágenes de consulta. También se propone, con las búsquedas ya etiquetadas, desarrollar un algoritmo que las agrupe según su parecido visual, si pertenecen a categorías similares.

1.4. Objetivos

Objetivo General

El objetivo general de esta memoria consiste en la implementación de uno o más algoritmos que logren las tareas de estructurar conjuntos de datos de consultas visuales. Esta estructuración debe aportar información útil para la administración de tiendas de e-commerce. Así mismo, debe de ser útil para Impresee de tal forma que sirva para evaluar y/o conocer el rendimiento de su producto (motor de búsqueda visual).

Objetivos Específicos

1. Identificar casos de búsquedas que corresponden a *Zero-Hits*.
2. Implementar un algoritmo de etiquetado por palabra de imágenes de consultas, para generar una categorización entre términos que se encuentren más frecuentemente en los nombres de los productos que posee un catálogo de tienda.
3. Implementar un algoritmo que agrupe conjuntos de búsquedas similares y además pueda identificarlos con una palabra que corresponda a su categoría.

1.5. Metodología

La metodología utilizada en el desarrollo de este trabajo consistió en un desarrollo, experimentación y evaluación iterativa de cada parte de la solución propuesta.

En particular, para el desarrollo, se experimentaron con diferentes alternativas para resolver una misma tarea, de tal forma de optar finalmente por la que entregó el mejor resultado. Por otro lado, la evaluación, dada la naturaleza práctica del trabajo, se realizó de manera manual junto a miembros de Impresee.

Capítulo 2

Marco Teórico

En el presente capítulo se describirán diferentes conceptos, técnicas y algoritmos que se mencionan o utilizan a lo largo de este trabajo. En general corresponden a temas variados pero el principal foco son, redes neuronales, clasificación de imágenes, técnicas de reducción de dimensionalidad y algoritmos de clusterización.

2.1. Redes Neuronales Artificiales

Las redes neuronales artificiales (ANN) son sistemas computacionales inspirados por la red neuronal biológica que constituye el cerebro de los animales. Una ANN está basada en una colección de unidades o nodos conectados, llamados neuronas artificiales o perceptrón.

A través de aristas (edges), con pesos, una ANN conecta diferentes componentes (capas) de neuronas artificiales que terminan por conformar un sistema adaptativo cuyo propósito consiste en, mediante un algoritmo, ajustar los pesos de cada conexión del sistema, para alcanzar el mejor resultado en alguna tarea dada.

2.1.1. Perceptrón

El perceptrón o neurona artificial corresponde a la unidad base de las redes neuronales artificiales. El perceptrón recibe una o más entradas (representando la excitación de las dendritas de una neurona biológica) y los suma para producir una salida (o activación, representando la acción de una neurona). Usualmente cada entrada es ponderada separadamente por un peso y la salida es pasada por una función no lineal, también conocida como función de activación.

En la Figura 2.1 se puede observar un diagrama que muestra más gráficamente el funcionamiento de un perceptrón que recibe cinco señales de entrada x_i y produce una salida y . Cada uno de los pesos x_i es ponderado separadamente por su peso w_i correspondiente. Finalmente, la salida corresponde a la suma de las entradas ponderadas, pasadas por una

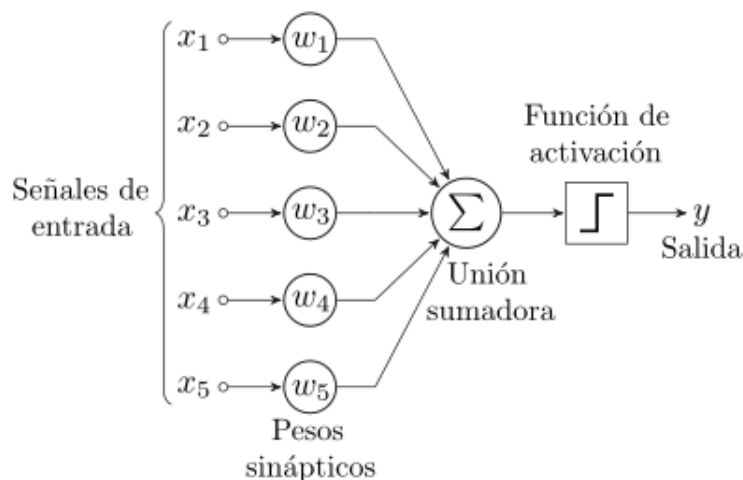


Figura 2.1: Diagrama de un perceptrón con cinco señales de entrada

función de activación f , obteniéndose: $y = f(\sum_{i=1}^n w_i x_i)$.

2.1.2. Convolución

En procesamiento de imágenes una convolución es una operación entre matrices, una grande (imagen) sobre la cual se está haciendo la operación y otra pequeña, cuadrada, llamada Kernel. Comúnmente las dimensiones de un Kernel son 1×1 , 3×3 , 5×5 ó 7×7 y su objetivo es aplicar un filtro sobre la imagen, analizando zonas del tamaño del Kernel en la imagen. Dependiendo de los parámetros utilizados en esta operación, el resultado podría disminuir el tamaño de la imagen de entrada, logrando así resumir su contenido.

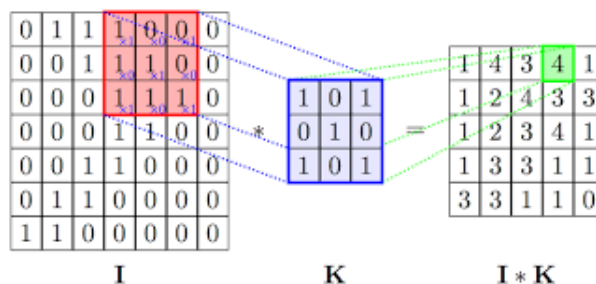


Figura 2.2: Ejemplo de convolución

En la Figura 2.2 se puede observar cómo se calcula una celda de la matriz resultante (verde), mediante la operación de una vecindad de la imagen (rojo) con el Kernel (azul).

2.1.3. Redes Neuronales Convolucionales

En deep learning, una red neuronal convolucional (CNN) es una clase de red neuronal artificial, comúnmente aplicada para analizar imágenes visualmente.

Para el análisis de imágenes, esta clase de red, recibe como entrada, comúnmente, un vector de dos dimensiones (imagen). Para procesar este tipo de entrada, consta de capas convolucionales que aplican una operación de convolución sobre todos los píxeles en la zona receptiva (tamaño del kernel) generando un único valor. Para este tipo de capa, los parámetros (pesos) a ajustar para aprender alguna tarea determinada, son los valores presentes en los kernel de las capas en cuestión.

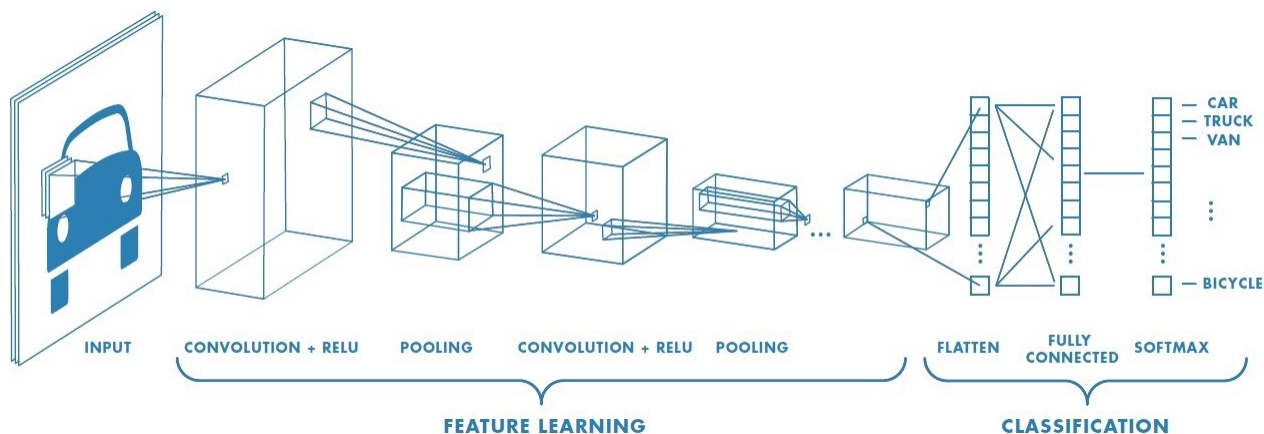


Figura 2.3: Ejemplo de CNN

En la Figura 2.3 se puede observar una arquitectura clásica de CNN para clasificación. En las capas convolucionales, se van extrayendo características disminuyendo progresivamente el tamaño de la entrada, hasta convertirlo en un vector que pasa por una ANN más clásica.

2.1.4. ResNet-152

Las ResNet (*Residual Networks*) son una arquitectura de CNN presentadas en la competencia *ImageNet Large Scale Visual Recognition Challenger* (ILSVRC) durante 2015, donde obtuvo el primer lugar en la tarea de clasificación de imágenes. En esta competencia, esta arquitectura, alcanzó un error en el top 3 de 3,6 % en la clasificación del conjunto de imágenes ImageNet[13].

Esta arquitectura es importante, dado a que mediante la implementación de conexiones residuales entre las diferentes capas convoluciones de la red, se logra un aprendizaje para redes muy profundas. Previo a este trabajo, acumular muchas capas en redes convolucionales disminuía el rendimiento, pues el sistema (pesos de las capas) se vuelven más difíciles de optimizar. En la Figura 2.4 se puede observar un bloque residual de una ResNet, en el cual la salida corresponde a la suma entre el paso de la entrada del bloque (x) por dos capas convolucionales (F) con la entrada misma (x).

En particular, la ResNet-152[7], cuenta con 152 capas convolucionales y fue la arquitectura que ganó la competencia ILSVRC previamente mencionada.

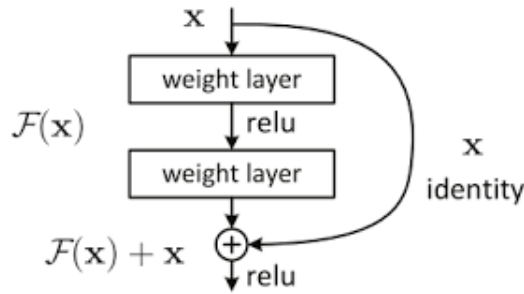


Figura 2.4: Bloque residual de una ResNet

2.2. Descriptores Visuales

En procesamiento de imágenes, los descriptores visuales son descripciones de características visuales presentes en el contenido de imágenes o videos. Los descriptores de imágenes, comúnmente, corresponden a vectores que resumen la información de las formas, colores o texturas presentes en la imagen.

2.2.1. Deep Feature

Un Deep Feature corresponde a la respuesta concisa de un nodo o capa dentro de un modelo jerárquico de red neuronal a una entrada dada. La salida o respuesta de esta capa, es relevante para la salida final del modelo. En particular, es ‘profunda’ (*deep*), dado que la capa que genera esta salida, está muy cerca de la salida final del modelo.

Para el caso particular de este trabajo, un Deep Feature corresponderá a la salida (vector) de última capa *Fully Connected* de un modelo, previo a la capa de clasificación. Por ejemplo, en la Figura 2.3, un Deep Feature podría ser la salida de cualquiera de las primeras dos capas del bloque de clasificación.

2.3. Algoritmos de Clusterización

Los algoritmos de clusterización tienen como objetivo agrupar objetos, ubicados en un espacio, de tal manera que todos los objetos pertenecientes a un mismo grupo tengan características similares. En general estos algoritmos se pueden categorizar por el modelo de clusterización, entre los que destacan los clústeres jerárquicos, basados en centroides y basados en densidad, entre otros.

Los algoritmos utilizados en este trabajo contemplan clusterizaciones jerárquicas y por densidad. La clusterización jerárquica está basada en la idea de que objetos más cercanos se encuentran más relacionados que aquellos objetos más lejanos. A partir de esta idea se genera una estructura jerárquica basada en las distancias entre los elementos. Por otro lado, en la clusterización basada por densidad, los clústeres están definidos como áreas de densidad más

alta que en el resto de elementos. Una característica importante que comparten ambos tipos es que pueden detectar clústeres no convexos.

2.3.1. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN [6] es un algoritmo de clusterización desarrollado en 1996 por científicos de la Universidad de Munich. Este algoritmo agrupa puntos juntos en un clúster si cumplen con las condiciones de ser al menos una cantidad mínima de puntos que se encuentren lo suficientemente cerca, según una función de distancia. Bajo este criterio para formar clústeres, este algoritmo marca como ruido aquellos puntos que no cumplan las condiciones para ser agrupados con otros puntos.

Tanto la cantidad mínima de puntos (*minPoints*) como la distancia mínima (*eps*) son parámetros de entrada que recibe el algoritmo.

1. ***eps***: Especifica cuan cerca deben estar los puntos, unos de otros, para ser considerados un clúster. Significa que si la distancia entre dos puntos es menor o igual a este valor (*eps*), entonces estos puntos serán considerados vecinos.
2. ***minPoints***: Es la cantidad mínima de puntos necesarios para formar una región densa (clúster). Por ejemplo, si se tiene como parámetro *minPoints* igual a cinco, entonces se necesitarán al menos cinco puntos para formar un clúster.

La estimación de cada uno de los parámetros a utilizar es una tarea manual, pero se ofrecen las siguientes directrices para lograrlo.

1. ***eps***: Si *eps* es muy pequeño, una gran porción de los puntos no serán clusterizados. Esto debido a que la condición para que dos puntos sean vecinos, será más difícil y por ende, lograr encontrar grupos de vecinos que tengan al menos *minPoints* puntos, será más improbable. Por otro lado, si el valor es muy alto, los diferentes clústeres se mezclarán y la mayoría de los puntos se encontrarán en el mismo clúster. En general este valor depende de cada conjunto de datos utilizados pero, en general, un valor pequeño favorecerá la detección de clústeres distintos por parte del algoritmo.
2. ***minPoints***: Como regla general, un mínimo de *minPoints* puede ser derivado del número de dimensiones D del conjunto de datos, con $minPoints \geq D + 1$ donde *minPoints* debe ser al menos 3. Esto último, debido a que si fuera 1, entonces por definición todos los puntos serían un clúster y si fuera 2, entonces el resultado del algoritmo sería lo mismo que un clúster jerárquico con enlace simple y el dendrograma cortado a la altura *eps*.

Por último, cabe mencionar que este algoritmo es capaz de detectar clústeres de diferentes formas y tamaños. En la Figura 2.5 se pueden observar algunos resultados de la aplicación de este algoritmo.

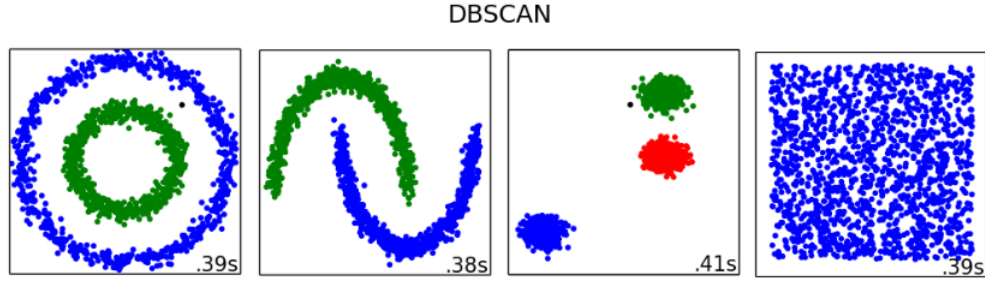


Figura 2.5: Ejemplo de clústeres detectados por DBSCAN

2.3.2. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

HDBSCAN [12] surge como algoritmo de clusterización durante el 2013, como mejora al algoritmo DBSCAN. Este algoritmo utiliza conceptos similares a DBSCAN para definir clústeres, pero recibe como entrada únicamente un parámetro, *minPoints*, que es la cantidad mínima de puntos para definir un clúster.

Inicialmente, al igual que DBSCAN, identifica zonas densas centradas en puntos con respecto a *minPoints*, pero cambia de métrica de distancia a distancia de alcance mutuo (d_{reach}). Esta nueva distancia d_{reach} entre dos puntos a y b está dada por:

$$d_{reach}(a, b) = \max\{core_k(a), core_k(b), d(a, b)\}$$

donde $core_k(x)$ corresponde a la distancia del punto x a la distancia del k -ésimo vecino más cercano de x , con $k = minPoints$, y $d(a, b)$ corresponde a la métrica de distancia original. En la Figura 2.6 se puede observar dos ejemplos de d_{reach} entre ambos puntos. En la primera imagen se puede observar como la distancia d_{reach} entre los puntos azul y verde corresponde a la distancia al núcleo del punto verde, pues es la mayor. Por el contrario, en la segunda imagen, la distancia d_{reach} entre el punto verde y rojo, corresponde a la distancia original $d(verde, rojo)$ entre ellos, ya que esta es mayor a las distancias $core_k(x)$ correspondientes a cada uno de los dos puntos.

Posteriormente, en contraste a DBSCAN, HDBSCAN computa un árbol mínimo cobertor (MST), sobre un grafo en el cual los nodos serán los puntos y las aristas la distancia de alcance mutuo entre los puntos del conjunto de entrada. Posteriormente, se va construyendo un dendrograma iterativamente mediante la eliminación de las aristas del MST, en orden decremental de pesos. Cada componente del dendrograma serán las componentes conexas que vayan quedando en el MST y su distancia (altura) será el peso de la arista que se remueve. El ruido corresponderá a las componentes que vayan quedando sin ninguna otra arista. En la Figura 2.7, se observa en primera instancia un MST de ejemplo, junto con su dendrograma asociado.

Por último, la altura a la que se corta el dendrograma para obtener el resultado de este algoritmo, corresponde a un problema de optimización. Para esto se condensa el dendrograma, sólo permitiendo que las divisiones de un nodo ocurran siempre que ambos nodos hijos sigan

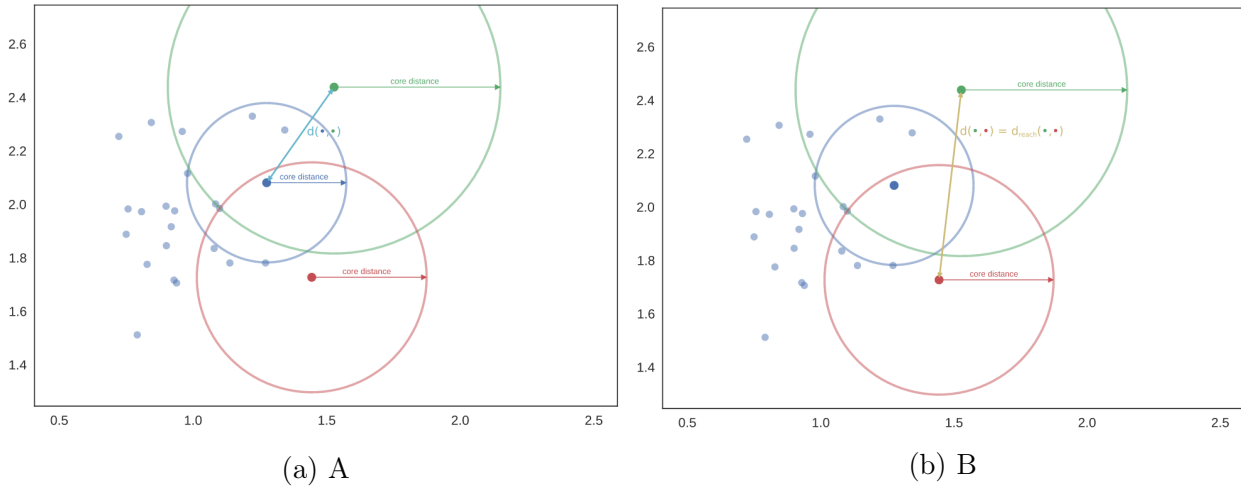


Figura 2.6: Ejemplo de distancia de alcance mutuo

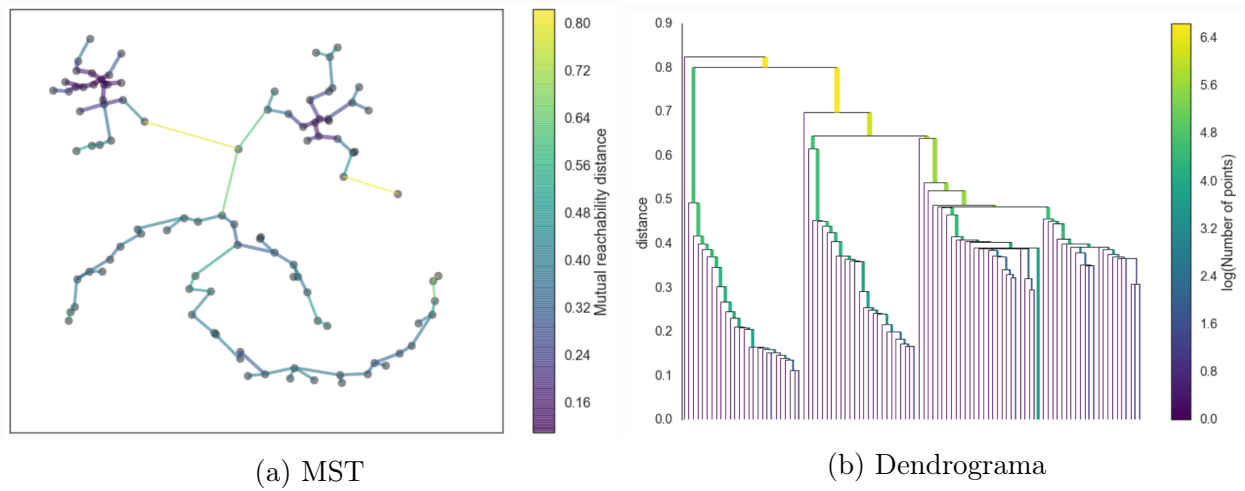


Figura 2.7: MST y Dendrograma calculados por HDBSCAN

teniendo al menos $minPoints$ puntos, en caso contrario el nodo mantiene toda la información de los hijos. En la primera imagen de la Figura 2.8, se puede observar el dendrograma previamente mostrado, en esta forma.

Finalmente, los clústeres seleccionados serán aquellos que, como se puede ver en la segunda imagen de la Figura 2.8, tengan la mayor cantidad de altura, en el dendrograma, sin dividirse.

Cabe mencionar que este algoritmo de clusterización posee las mismas cualidades que el algoritmo DBSCAN, pero con la ventaja de aceptar densidades variables en los clústeres detectados.

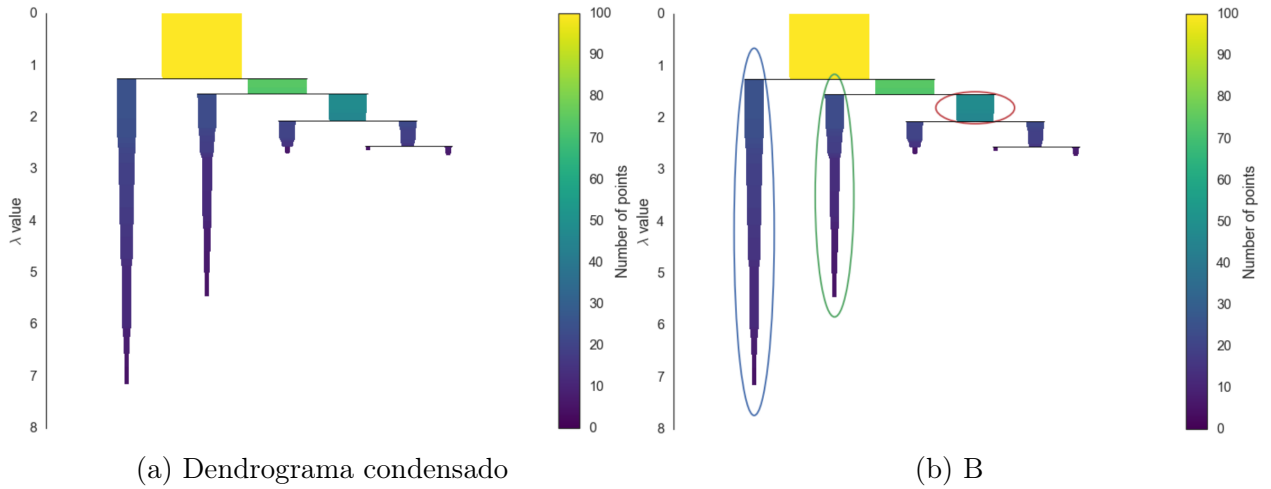


Figura 2.8: Selección de clústeres

2.4. Métricas de evaluación

En el desarrollo de este trabajo se utilizan diferentes métricas de evaluación para diferentes tareas. A continuación se describen diferentes métricas utilizadas para principalmente evaluar tareas de clustering y clasificación.

2.4.1. Mutual Information (MI)

Mutual Information (MI) es una métrica basada en teoría de la información y utilizada para comparar clústeres.

Dado un conjunto S de N elementos $S = \{s_1, s_2, \dots, s_N\}$, se considera dos particiones de S , $U = \{U_1, U_2, \dots, U_R\}$ con R clústeres y $V = \{V_1, V_2, \dots, V_C\}$ con C clústeres. Se cumple:

$$U_i \cap U_j = \emptyset = V_i \cap V_j$$

para todo $i \neq j$, y:

$$\cup_{i=1}^R U_i \subseteq S$$

$$\cup_{j=1}^C V_j \subseteq S$$

La MI de la superposición de clústeres entre U y V puede ser representada en la forma de una matriz de contingencia $M = [n_{ij}]_{j=1 \dots C}^{i=1 \dots R}$, de tamaño $R \times C$, donde n_{ij} denota al número de objetos en común que tienen los clústeres U_i y V_j . Esto es, $n_{ij} = |U_i \cap V_j|$.

Si se toma un elemento al azar de S , la probabilidad de que ese elemento sea parte del clúster U_i es:

$$P_U(i) = \frac{|U_i|}{N}$$

La entropía asociada a la partición U es:

$$H(U) = - \sum_{i=1}^R P_U(i) \log P_U(i)$$

Donde $H(U)$ es no-negativa y toma valor 0 sólo cuando no hay incertidumbre respecto a la determinación de pertenencia de un elemento a un clúster, es decir, cuando hay únicamente un clúster. $H(V)$ se calcula de manera análoga.

Luego, la MI entre dos particiones se calcula como:

$$MI(U, V) = \sum_{i=1}^R \sum_{j=1}^C P_{UV}(i, j) \log \frac{P_{UV}(i, j)}{P_U(i)P_V(j)}$$

donde $P_{UV}(i, j)$, denota la probabilidad de que un elemento pertenezca a ambos clústeres U_i en U y V_j en V :

$$P_{UV}(i, j) = \frac{|U_i \cap V_j|}{N}$$

Finalmente MI es una cantidad no-negativa acotada superiormente por las entropías $H(U)$ y $H(V)$. Cuantifica la información compartida por dos clústeres y por ello, es utilizada como métrica de evaluación de clústeres.

2.4.2. Adjusted Mutual Information (AMI)

AMI es una variación de la métrica *mutual information*. Corrige el efecto de concordancia entre clústeres debido al azar y también acota el valor de similitud entre 0 y 1, cuando originalmente no es un intervalo constante y, más aún, tiende a ser mayor cuando las dos particiones (clusterizaciones) tienen mayores números de clústeres. Finalmente se calcula como:

$$AMI(U, V) = \frac{MI(U, V) - E(MI(U, V))}{avg(H(U), H(V)) - E(MI(U, V))}$$

donde, MI es la métrica de Mutual Information, E es el valor esperado y H la entropía.

2.4.3. Rand Index (RI)

Rand Index (RI) o Rand Mesasure es una métrica utilizada para medir la similitud de dos clusterizaciones.

Dado un conjunto de n elementos $S = \{o_1, o_2, \dots, o_n\}$ y dos particiones de S a comparar, $X = \{X_1, X_2, \dots, X_r\}$, partición de S en r sub-conjuntos, y $Y = \{Y_1, Y_2, \dots, Y_s\}$, una partición de S en s sub-conjuntos, se define lo siguiente:

1. a , el número de pares de elementos en S que se encuentran en el mismo sub-conjunto en X y en el mismo sub-conjunto en Y .
2. b , el número de pares de elementos en S que se encuentran en diferentes sub-conjuntos en X y en diferentes sub-conjuntos en Y .
3. c , el número de pares de elementos de S que se encuentran en el mismo sub-conjunto en X y diferentes sub-conjuntos en Y .
4. d , el número de pares de elementos de S que se encuentran en diferentes sub-conjuntos en X y en el mismo sub-conjunto en Y .

Luego, el Rand Index se calcula como:

$$R = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}}$$

2.4.4. Adjusted Rand Index (ARI)

ARI es un ajuste sobre RI que establece una línea base entre las similitudes esperadas por todas las comparaciones de pares entre clústeres, especificadas por un modelo aleatorio. Originalmente RI corresponde valores entre 0 y 1, donde 0 son clústeres sin ninguna coincidencia entre pares de puntos mientras que 1 son clústeres idénticos. Sin embargo, ARI entrega una métrica que compara con respecto a un modelo aleatorio, por lo que 1 significa clústeres idénticos, 0 que los clústeres coinciden tanto como una clusterización aleatoria y, por último, valores negativos significan que la coincidencia es peor a una clusterización aleatoria. Se calcula como:

$$ARI(U, V) = \frac{RI(U, V) - E(RI(U, V))}{\max(RI(U, V)) - E(RI(U, V))}$$

donde RI es la métrica Rand Index y E es el valor esperado.

2.4.5. Precision y Recall

Precision es una métrica de evaluación que corresponde a la fracción de ejemplos correctamente seleccionados como positivos (verdaderos positivos), sobre todos los seleccionados como positivos, por un modelo (verdaderos positivos + falsos positivos).

Por otro lado el *Recall*, es otra métrica de evaluación que corresponde a la fracción de ejemplos correctamente seleccionados como positivos (verdaderos positivos), sobre todos los ejemplos que son realmente positivos (verdaderos positivos + falsos negativos).

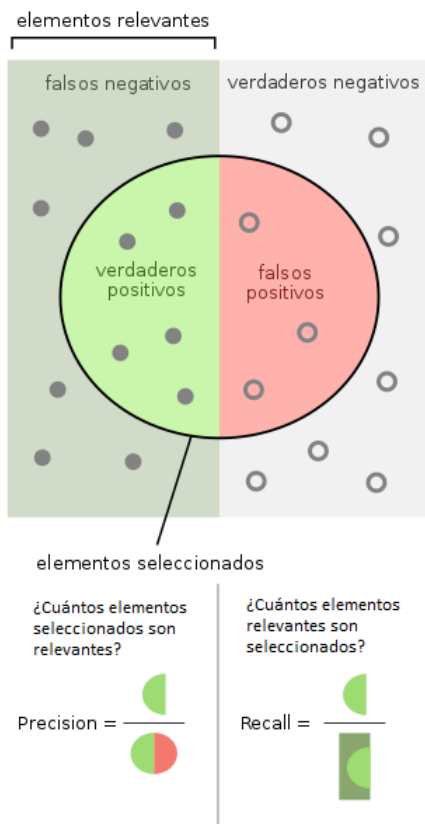


Figura 2.9: Precision y Recall

Fuente: Wikipedia

2.4.6. Precision at k (P@k)

En tareas de clasificación, donde un modelo o algoritmo devuelve más de una respuesta ordenada por alguna métrica de confianza, se puede estudiar que la respuesta correcta se encuentre en los *top k* resultados entregados. La cantidad *k* es variada y depende de cuan flexible se quiere ser con el resultado y también depende de la cantidad de respuesta que entrega el modelo, comúnmente son valores pequeños como 1, 3 o 5.

2.4.7. Intersection Over Union (IOU)

Intersection Over Union (IOU) es una métrica utilizada para comparar similitud entre dos conjuntos de elementos.

Si se tiene un conjunto de n palabras $W = \{w_1, w_2, \dots, w_n\}$, y dos subconjuntos de W , $A = \{a_1, a_2, \dots, a_r\}$ y $B = \{b_1, b_2, \dots, b_c\}$

El IOU de A y B corresponderá a:

$$IOU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

2.5. Distancia de Levenshtein

La distancia de Levenshtein es una métrica para calcular cuan diferentes son dos palabras. En otras palabras, mide el número de ediciones a realizar sobre una palabra para obtener la otra. Estas ediciones pueden ser inserciones, eliminaciones o sustituciones de caracteres.

La definición formal de esta distancia entre dos cadenas de caracteres a y b es:

$$lev_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{si } \min(i, j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1, j) + 1 \\ lev_{a,b}(i, j-1) + 1 \\ lev_{a,b}(i-1, j-1) + 1_{a_i \neq b_j} \end{cases} & \text{si no} \end{cases}$$

donde $1_{a_i \neq b_j}$ denota 0 cuando los caracteres i de a y j de b son iguales y 1 en cualquier otro caso. Los casos dentro de la llave del *min*, corresponden a las eliminaciones, inserciones y sustituciones de caracteres, en ese orden.

Particularmente, para este trabajo, se utiliza esta métrica como criterio para agrupar palabras similares. Un ejemplo de esto sería la agrupación de una palabra en su forma singular y plural, esto último siendo particularmente útil para casos en que la forma plural de una palabra no es solamente la misma palabra con una letra s añadida al final, como por ejemplo *lápiz*.

2.6. Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF es una métrica que expresa cuán relevante es una palabra para un documento de una colección de documentos. En análisis de textos, se utiliza principalmente como ponderador de términos, para así filtrar palabras frecuentes que no aportan información.

Más específicamente, TF-IDF es un producto de dos medidas, frecuencia de término y frecuencia inversa de documento.

Por un lado, la frecuencia de término corresponde a la frecuencia normalizada de aparición de un término t en un documento d :

$$tf(t, d) = \frac{f(t, d)}{\max\{f(t, d) : t \in d\}}$$

Por otro lado, la frecuencia inversa de documento corresponde a una medida de si un término t , es común, dentro de una colección de documentos D :

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

donde $|D|$ es la cantidad de documentos en la colección y $|\{d \in D : t \in d\}|$ es el número de documentos donde aparece el término t .

Finalmente, el TF-IDF se calcula como:

$$TF-IDF(t, d, D) = tf(t, d) \times idf(t, D)$$

2.7. Reducción de Dimensionalidad

En aprendizaje automático y estadística, la reducción de dimensionalidad es el proceso de reducción del número de variables aleatorias de un conjunto de datos. Esta reducción de variables, naturalmente, viene con un costo asociado de la precisión de los datos, pues se está perdiendo información. Sin embargo, estos métodos buscan minimizar esta pérdida de información, con el objetivo de volver los datos más manejables.

2.7.1. Principal Component Analysis (PCA)

PCA es un método de reducción de dimensionalidad, mediante la transformación de un conjunto de datos con muchas componentes (dimensiones) en uno con unas pocas componentes que contienen la mayor parte de la información original.

Sea X un conjunto de n elementos de p dimensiones (variables), la matriz $p \times p$ de covarianzas entre cada una de las p variables de X se define como:

$$C(X) = [c_{ij}]_{i=1 \dots p}^{j=1 \dots p}$$

con $c_{ij} = cov(dim_i, dim_j)$. Luego, los valores y vectores propios de la matriz $C(X)$ son:

$$\{(\lambda_1, \vec{v}_1), (\lambda_2, \vec{v}_2), \dots, (\lambda_p, \vec{v}_p)\}$$

Las k componentes serán los k pares (λ_i, \vec{v}_i) que tengan los mayores k valores propios. El P porcentaje de información que mantienen estas k componentes está dado por:

$$P = \frac{\sum_{z=1}^k \lambda_z}{\sum_{i=1}^p \lambda_i}$$

La transformación a este nuevo espacio compuesto de k componentes está dada por la siguiente matriz de transformación, donde cada uno de los \vec{v}_i son los vectores propios de las k componentes seleccionadas, puestos como columnas:

$$T = [\vec{v}_1 \mid \vec{v}_2 \mid \dots \mid \vec{v}_k]$$

Finalmente, para transformar un vector de Y de p dimensiones, a este espacio, basta con calcular:

$$Y' = T^T Y$$

2.7.2. t-distributed stochastic neighbor embedding (t-SNE)

t-SNE[16] es un método estadístico para la visualización de datos de alta dimensión, mediante un mapeo punto a punto de los datos en un mapa de 2 o 3 dimensiones.

Dado un conjunto de N elementos de alta dimensión x_1, \dots, x_N , t-SNE primero calcula las probabilidades p_{ij} proporcionales a la similitud de los objetos x_i y x_j , para $i \neq j$ de la siguiente manera:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

Luego, como indican los autores del paper de t-SNE, "*La similitud del punto x_j a el punto x_i es la probabilidad condicional, $p_{j|i}$, de que x_i elija a x_j como su vecino, si los vecinos fueran elegidos en proporción a su densidad de probabilidad bajo una Gaussiana centrada en x_i .*" Por lo tanto, se define:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

donde se cumple que $p_{ij} = p_{ji}$, $p_{ii} = 0$ y $\sum_{ij} p_{ij} = 1$.

t-SNE apunta a aprender un mapeo d -dimensional y_1, \dots, y_N (con $y_i \in \mathbb{R}^d$), que refleje las similitudes p_{ij} de la mejor manera posible.

Cabe mencionar que t-SNE no preserva ni las distancias ni densidades del espacio original, en el espacio de baja dimensión. Sólo preserva, hasta cierto punto, los vecinos más cercanos de cada punto. Por eso último, no es recomendado utilizar el resultado de t-SNE para efectuar análisis y otros experimentos, pues no se está viendo los datos en su forma más real. Sin embargo, en este trabajo se experimentó igualmente con esta técnica de visualización, como técnica de reducción de dimensión, para contrastar con otros métodos.

2.7.3. Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)

UMAP [9] es una técnica de reducción de dimensiones que puede ser utilizada para visualización, al igual que t-SNE, pero también como método de reducción no lineal de dimensiones.

El algoritmo que utiliza UMAP consiste, esencialmente, de dos fases. Inicialmente, construye una representación topológica difusa, del conjunto de datos en el espacio de alta dimensión. Por otro lado, la segunda fase consiste, simplemente, en optimizar la representación en baja dimensión de los datos, de tal manera que la representación topológica de este último espacio, sea lo más parecida a la representación inicial.

En particular para la representación topológica inicial, se consideran los vecinos más cercanos de cada punto, generándose elementos topológicos que representan vecindades y sus conexiones. Para la segunda fase, se consideran los pesos del elemento topológico $w(e)_h$ en alta dimensión y su peso correspondiente en el espacio de baja dimensión $w(e)_l$. La optimización se hace minimizando una función de entropía cruzada que relaciona ambas variables.

El resultado de UMAP, al igual que el de t-SNE, no preserva la densidad de los datos en alta dimensión. Sin embargo, cabe mencionar que experimentos[4], demuestran que preserva de mejor manera la estructura global de los datos que t-SNE. Por esto último, es que se considera, en este trabajo, experimentar con esta herramienta como pre-procesamiento para otros análisis como clusterización, en datos de alta dimensión.

Capítulo 3

Recopilación de Antecedentes

3.1. Estado del arte

En esta sección se presentarán diferentes trabajos y estudios orientados a la clasificación de imágenes, extracción de atributos y visualización de espacios de alta dimensión. Esto debido a que la tarea más complicada del trabajo propuesto en esta memoria, consiste en categorizar y visualizar las imágenes que se buscan en una tienda.

3.1.1. Clasificación

La clasificación consiste en una suerte de taxonomía para determinar la etiqueta o clase a la que pertenece un objeto. Para esta tarea existen principalmente dos técnicas diferentes, una es la clasificación con descriptores visuales y la segunda es la clasificación con redes neuronales.

Clasificación con Descriptores Visuales

Los descriptores visuales, o de imagen, corresponden a un vector que representa las características visuales contenidas dentro de una imagen. La clasificación de imágenes que hace uso de estos descriptores busca determinar la clase de una imagen según alguna característica presente en su descriptor visual. Este tipo de clasificación suele tener un buen rendimiento, sin embargo, depende mucho de ajustes en el algoritmo de clasificación que aplican únicamente para el conjunto a clasificar. Debido a esto último, trabajos que hacen uso de técnicas de este tipo, son difíciles de aplicar en otras tareas de clasificación cuyos datos a clasificar no tengan similitudes con los datos con que fue inicialmente propuesto el trabajo.

Tomasik et al. [14] utilizan un clasificador *k nearest neighbours* sobre un dataset de imágenes de productos de Amazon. Los descriptores utilizados fueron *bag of visual words* de descriptores *SIFT*. Pese a que obtienen buenos resultados (hasta un 80% de precisión en algunas clases), se ve que, en general, las imágenes que se clasifican son imágenes de catálogo

con un fondo blanco. Debido a esto, se espera que su rendimiento para la aplicación propuesta sea muy por debajo de lo indicada en la investigación citada, al considerar que se utilizaría con imágenes que difícilmente compartirían un solo fondo uniforme y, por el contrario, se trataría de productos con diferentes tipos de detalles y grados de interferencia visual.

Por otro lado Oyewole et al. [10] extrae descriptores de colores de alta dimensionalidad de las imágenes, para posteriormente reducir su dimensión con un algoritmo similar a *PCA*. Finalmente comparan los rendimientos de diferentes arquitecturas para clasificar utilizando estos descriptores, obteniendo como resultado que un *Ensemble de Support Vector Machines* logre resultados similares a una arquitectura de *Red Neuronal* con precisiones de 84% y 87.2% respectivamente. Pese a que se obtienen resultados bastantes buenos, cabe mencionar que ocurre algo similar con el dataset del trabajo previamente revisado de Tomasik et al. [14], que consiste en un dataset de imágenes que comparten características similares analogables a las encontradas en un catálogo, al presentarse pulcramente el objeto a clasificar y por lo tanto se espera que los resultados no obtengan el mismo grado de logro para imágenes menos controladas.

Clasificación con Redes Neuronales

La clasificación con redes neuronales es más reciente que la clasificación con descriptores. La mayor diferencia con respecto al previamente mencionado método de clasificación es que los trabajos que emplean este tipo de clasificación, pueden ser aplicados en otras tareas de clasificación sin grandes ajustes en el modelo como tal, debido a que las redes neuronales, durante su entrenamiento, aprenden del dataset que se les provee. Sin embargo, este tipo de clasificación requiere una cantidad considerablemente mayor de recursos, en tanto es computacionalmente más caro en comparación a la clasificación con descriptores y además requiere de datos de entrenamientos que deben estar correctamente etiquetados de antemano.

Actualmente los trabajos que buscan resolver el problema de categorizar productos de tiendas con redes neuronales, tratan de emplear no solo la información visual (imagen) de los productos, sino que también información textual como lo es el título y descripción del producto. Así, en el trabajo de Wirojwatanakul et al. [18], evalúan el rendimiento de clasificadores para imágenes, títulos y descripciones de los productos por separado, para posteriormente crear un clasificador modal que utilice las salidas de estos tres clasificadores. En este trabajo la clasificación a nivel de imagen se hace mediante una ResNet [7] pre-entrenada en el dataset ImageNet [13]. En particular, esta investigación sirve para evidenciar una alternativa a la categorización de productos que contempla la combinación de diferentes clasificadores que toman en consideración distintas entradas, siendo la imagen del producto una de ellas. Por último, una de las conclusiones interesantes de este paper, es que un clasificador que hace uso únicamente del título del producto, supera al rendimiento de un categorizador que usa solo la imagen del producto.

Rajput et al. [11] utilizan una categorización de prendas de vestuario hindú, a través de únicamente la imagen del producto. Los modelos que se prueban en este trabajo son diferentes versiones de la arquitectura ResNet [7] y entrena con datos de algunos datasets de productos de tiendas como lo es Atlas [15]. El resultado de este trabajo es notable, pues

alcanza precisiones sobre el 80% en los datasets de test. Sin embargo, la tarea a resolver es un poco más acotada con respecto a la clasificación del dataset ImageNet [13], pues solo son 15 categorías diferentes en las que se deben clasificar las prendas, y por lo tanto es esperable que modelos utilizados en tareas de clasificación con muchas más clases, se desempeñen de buena forma.

3.1.2. Deep Features

Como se mencionó en una de las secciones previas, un descriptor visual corresponde a un vector que almacena características presentes dentro de una imagen. En el último tiempo, tareas de clasificación de imágenes han comenzado a utilizar *deep features*, como alternativa a descriptores calculados de manera manual. Estos *deep features* son creados por una *Red Neuronal Convolutiva* o *CNN* y corresponden a representaciones intermedias de una imagen. Mas en detalle, un *deep feature* es la entrada a la capa previa de la red que termina por clasificar una imagen, en alguna clase. Para que este descriptor sea significativo, se debe extraer de una red que ya ha sido entrada para clasificar imágenes.

En general, el mayor beneficio de los *deep features* es contar con descriptores altamente semánticos, y poder usarlos en otros tipos de algoritmos clásicos que utilicen descriptores visuales para alguna tarea. Un ejemplo es el trabajo de Wieczorek et al. [17] que emplea *deep features* para la recuperación de imágenes. En este trabajo, se utiliza una red con arquitectura ResNet [7], entrenada en datasets como DeepFashion [8], tras lo cual calcula centroides en el espacio de estos descriptores para posteriormente recuperar una nueva imagen en el subconjunto de descriptores que contiene al centroide que está más cerca de él.

3.1.3. Visualización de espacios de alta dimensión

Dado que los descriptores visuales se caracterizan por tener alta dimensionalidad, es que es importante encontrar una forma particular de visualizarlos para poder observar los resultados que se obtienen. En los últimos años han surgido técnicas novedosas en esta área como lo son t-SNE [16] y UMAP [9], para las cuales se obtienen visualizaciones que preservan de mejor manera la estructura local de las vecindades de puntos en el espacio resultante, que con PCA. En estos trabajos, se presentan visualizaciones de diferentes tipos de descriptores, entre ellos descriptores visuales, haciendo uso de cada una de las técnicas.

Capítulo 4

Descripción e Implementación de la Solución

4.1. Descripción General de la Solución

El trabajo consiste en el desarrollo de algoritmos que permitan ordenar conjuntos de búsquedas visuales parecidos, para facilitar su posterior inspección visual. El resultado pretende caracterizar las búsquedas de tal manera que, por ejemplo, un dueño de tienda pueda averiguar cuál categoría es la que está siendo más buscada por sus clientes, así como también contar con la posibilidad de visualizar de manera más estructurada, agrupando todas aquellas búsquedas que se refieran a un cierto tipo de producto. Es por ello que la solución desarrollada consta de: clasificación de búsquedas en base corresponde a un *Zero-Hit* o no, cálculo de una etiqueta textual de búsquedas en función de los resultados entregados por el buscador, clusterización con el algoritmo **HDBSCAN** de las búsquedas utilizando tanto sus descriptores visuales, como también sus etiquetas textuales previamente calculadas. Posteriormente, se evalúan los diferentes escenarios de clusterización con el objetivo de seleccionar subconjuntos de búsquedas, bien identificados, que permitan mostrar al dueño de la tienda la actividad de sus clientes.

4.2. Implementación

4.2.1. Conjuntos de datos

Los datos a utilizar para el desarrollo de las diferentes partes de la solución constan de un conjunto de imágenes de consultas recibidas por una tienda durante un mes de uso del buscador. A parte de la imagen como tal, también se tienen los descriptores *DeepFeatures*, utilizados por el buscador, de cada una de las imágenes. Estos descriptores son vectores numéricos de 2048 componentes y corresponden a la salida de penúltima capa *Fully-Connected* de una red neuronal **ResNet-152**.

Por último, también se cuenta con los resultados, entregados por el buscador, en respuesta a cada una de las consultas visuales recibidas. Estas respuestas constan de los títulos y categorías de un número variado de productos, que son los vecinos más cercanos, del catálogo de una tienda, a la imagen de consulta.

En la Tabla 4.1 se puede observar una imagen de consulta de muestra junto con los 5 títulos y categorías (separados por una coma) de los productos del catálogo más parecidos.


Imagen de consulta	Top 5 resultados
	<ul style="list-style-type: none"> • Silla Gamer Ergonómica Ajustable Reclinable, Oficina • Nombre: Silla Gamer Escritorio Regulable Negro-Rojo, Oficina • Nombre: Silla Gamer Forks Plus Rojo Escritorio, Oficina • Nombre: Silla Escritorio Gamer Butaca Reclinable + 2 Almohadillas Negro y Rojo, Oficina • Nombre: Silla Gamer Escritorio Regulable, Oficina

Tabla 4.1: Muestra de 5 resultados más cercanos para una consulta

Para omitir búsquedas idénticas realizadas por los mismos usuarios, se filtran las consultas visuales, según el **MD5 checksum**, provisto por la librería **hashlib** de Python, de las imágenes de búsqueda. De esta manera, sólo se considera una única búsqueda por imagen. Esto para evitar clústeres redundantes en los análisis posteriores.

4.2.2. Etiquetado de búsquedas y detección de Zero-Hits

Se propone etiquetar las búsquedas por imagen en función de los resultados del buscador. Para ello se obtienen las palabras (unigramas) y pares de palabras (bigramas) más frecuentes entre los títulos de los cinco mejores resultados del buscador, para cada búsqueda. Estas palabras o pares de palabras de ser significativos, sirven para etiquetar una búsqueda, en base al resultado otorgado por el buscador. A continuación se explica en más detalle, paso por paso el proceso.

1. Limpieza de texto

Para evitar ruido y que no se agrupen correctamente palabras que deberían ser iguales, se remueven todos los diacríticos y símbolos presentes en el texto. También, se reemplazan las letras mayúsculas por minúsculas. Por último, también se remueven palabras con largo menor que 3 caracteres, que generalmente no aportan información relevante del producto.

2. Diccionario de frecuencias y agregación de palabras similares

Se genera un diccionario de frecuencias de palabras y pares de palabras presentes en el texto ya limpio, que contiene los títulos y categorías de los 5 productos más cercanos a la búsqueda, según el buscador visual.

Un caso que favorece a la dispersión de frecuencias de palabras muy parecidas es el caso de la forma plural del nombre de ciertos productos, como por ejemplo ‘Mueble’ y ‘Muebles’ o casos más complejos como ‘Farol’ y ‘Faroles’. Para evitar esto se realiza un análisis de similitud entre cada una de palabras del diccionario de frecuencias para fusionar las más parecidas y tengan una mayor frecuencia. En la Tabla 4.2 se observan similitudes entre algunas palabras post-procesadas del catálogo de la tienda. La métrica para determinar si dos palabras corresponden a palabras similares, es la distancia de Levenshtein disponible en la librería **FuzzyWuzzy**, donde si dos palabras tienen un parecido *token_set_ratio* mayor a 85 % entonces suman sus frecuencias mutuamente, para potenciar ambas apariciones de la palabra.

Palabra 1	Palabra 2	% Similitud
higienico	planta	13
repisa	repisas	92
figuras	medianos	40
fotografias	fotografica	91
utensilios	impurezas	32
plantas	planeta	86
plastico	plastica	88

Tabla 4.2: Ejemplo de similitud de palabras utilizando FuzzyWuzzy

3. IDF y filtro de etiquetas

El primer filtro para determinar si una etiqueta, del diccionario de frecuencias, es buena para identificar a una consulta, es que se encuentre presente en la mayor cantidad posible de los 5 resultados más cercanos según el buscador. Por ello, se considera una palabra si esta se encuentra al menos cuatro veces en los cinco títulos o categorías.

Por otro lado, es importante asegurarse la etiqueta aporte información útil para poder diferenciar las diferentes consultas. Para ello también es importante considerar su frecuencia total en los títulos del catálogo de la tienda a la que pertenecen. Las palabras o pares de palabras con frecuencia x que tengan un *IDF* (**Inverted document frequency**) que cumpla que $1,4 < IDF(x) < 2,9$, con $IDF(x) = \log_{10}(\frac{D_t}{D_f})$, donde D_t corresponde a la cantidad total de documentos (productos del catálogo) y D_f a la cantidad de documentos que contienen a la palabra o etiqueta en cuestión.

Este último filtro sobre el *IDF* funciona como filtro de **Stop Words** y valida que etiquetas aporten información en tiendas con un catálogo de aproximadamente 10 mil productos. Los rangos de palabras que quedan posterior al filtro, serán aquellas que aparecen mencionadas entre 10 y 300 veces en el catálogo de una tienda.

Las etiquetas seleccionadas se ordenan de manera descendente con un *rank* que corresponde al producto entre la frecuencia local de la etiqueta entre los 5 títulos más cercanos y el *IDF* de la etiqueta en el catálogo $rank(x) = local_freq(x) \cdot IDF(x)$.

Finalmente, en la Tabla 4.3 se muestra una imagen de consulta, junto con los textos de los títulos de sus cinco productos más parecidos sin procesar (previo a la limpieza), y por último con las etiquetas que se extraen de esos títulos y categorías.


Imagen de consulta	Top 5 resultados	Etiquetas calculadas
	<ul style="list-style-type: none"> ● Kit Cocina Mueble 8 Puertas 4 Gavetas, Cocina ● Mueble Cocina 7 Puertas 2 Cajones Lily Louro Freijo, Muebles ● Mueble Rack Tv Madera Cajones, Living ● Mueble Cocina Compacta Sara 4 Puertas 1 Cajon Blanco, Cocina ● Mueble Tv Rack 2 Puertas Home Theater, Living 	<ul style="list-style-type: none"> ● muebles, 14.940 ● mueble, 11.848 ● puertas, 7.827

Tabla 4.3: Muestra de los cinco resultados más cercanos para una consulta y sus etiquetas


Imagen de consulta	Top 5 resultados
	<ul style="list-style-type: none"> ● Cubre Asiento Térmico 12v Calienta Asiento, Accesorios ● Mochila Mini Bolso Escolar Diseño Canvas Moose, Bolsos ● Huincha Cinta Métrica 5 Metros 16mm, Herramientas y Construcción ● Capa Para Mochila Tamaño De 55 - 80lt Nautika, Artículos de Camping ● Pechera Delantal Asado Wayu Parrilla Cocina 70 x 80 cm, Fiestas Patrias

Tabla 4.4: Muestra de los cinco resultados más cercanos para una consulta clasificada como Zero-Hit

Por último, las búsquedas cuyos resultados no generen ninguna etiqueta según los criterios establecidos por este algoritmo, serán considerados **Zero-Hits**. Un ejemplo es la consulta presente en la Tabla 4.4.

4.2.3. Clusterización de etiquetas

Dado que el algoritmo para la etiqueta de búsquedas genera, dependiendo del caso, múltiples etiquetas para cada una de las consultas, se propone un método para agrupar aquellas etiquetas que corresponden a productos similares. Esto para no omitir etiquetas que pueden contener información relevante para identificar una búsqueda. Para ello se propone una función de distancia entre dos etiquetas, para poder agrupar aquellos productos que poseen etiquetas cercanas entre sí.

1. Función de distancia entre etiquetas

Se propone la distancia entre las etiquetas x e y calculadas de dos búsquedas diferentes $D(x, y) = 1 - f(x, y)$, donde $f(x, y) = \frac{x \cap y}{x \cup y}$ corresponde a la intersección sobre la unión de las diferentes etiquetas que tienen cada una de las etiquetas presentes en x e y . Calculando la distancia sobre todas las etiquetas se genera una matriz de distancias de etiqueta de todas las consultas del conjunto, donde M_{ij} es la distancia $d \in [0, 1]$ entre las búsquedas i y j . En la Tabla 4.5 se pueden observar algunas etiquetas y las distancia calculada entre ellas, usando la función de distancia $D(x, y)$.

Etiqueta 1	Etiqueta 2	Distancia
silla, sillas, oficina	silla, sillas, taburete, taburetes, (silla, taburete), (sillas, taburetes), (taburetes, silla)	0.75
silla, sillas, oficina	mueble, muebles, rack, living, (mueble, rack)	1.0
mueble, muebles, rack, living, (mueble, rack)	silla, sillas, taburete, taburetes, (silla, taburete), (sillas, taburetes), (taburetes, silla)	1.0

Tabla 4.5: Ejemplo de la función de distancia propuesta entre pares de etiquetas distintas

2. Algoritmo de clusterización

Posteriormente se pueden clusterizar las consultas, en base a la matriz de distancia entre sus etiquetas. El algoritmo utilizado en la clusterización, es uno que pueda detectar ruido (puntos dispersos que no pertenecen a ningún clúster) y no considerarlos en los clústeres y que, además, sea capaz de identificar cualquier tipo de forma como lo es HDBSCAN en la implementación que provee la librería `hdbscan` de Python.

4.2.4. Clusterización de descriptores

Un acercamiento diferente para agrupar aquellas búsquedas similares, es clusterizar los descriptores visuales que representan cada imagen de búsqueda. Se proponen dos clusterizaciones de este tipo, la primera es una clusterización en los descriptores de 2048 componentes directamente y otra en un espacio de menor dimensión (2 componentes). La reducción dimensión tiene como objetivo mejorar la clusterización provista por el algoritmo HDBSCAN, al costo de perder algo de información. La reducción de dimensionalidad se hizo con la técnica UMAP.

1. Reducción de dimensionalidad con UMAP

Para la reducción de dimensionalidad se determinó que los mejores parámetros para transformar mediante UMAP (con su librería **umap** en Python), los vectores de dimensión 2048 a un espacio de dos dimensiones son $n_neighbours = 30$ y $min_dist = 0$.

2. Algoritmo de clusterización

Para clusterizar los descriptores en tanto el espacio de dimensión reducida de 2 componentes, como también en el espacio de alta dimensionalidad, se utiliza el algoritmo HDBSCAN en la implementación que provee la librería **hdbscan** en python.

4.2.5. Selección de clústeres

Ninguno de los resultados obtenidos de las clusterizaciones por etiquetas y por descriptores visuales (ambos casos) es suficientemente buena por sí sola, por ello, se propone usar ambos resultados para determinar cuáles clústeres son más útiles.

Para esto se consideran ambos escenarios de clusterización de descriptores visuales y se elige uno de ellos, comparándolo con el resultado de clusterización de etiquetas. Para la selección se consideran los siguientes pasos.

1. Condición de selección

Se considera válido un clúster si las búsquedas que lo componen se encuentran contenidas en al menos un 40% en algún clúster y de etiquetas descrito en la Sección 4.2.3.

El escenario de clusterización por descriptores elegido, será aquel que tenga mayor cantidad de clústeres útiles, según la definición recién mencionada.

2. Nombre del clúster

Para la selección de nombre del clúster x de descriptores, se consideran las etiquetas correspondientes al clúster y de etiquetas que mejor contiene a x , según la restricción descrita en el punto anterior. El título corresponde a las etiquetas más frecuentes entre las búsquedas visuales que coinciden en ambos clústeres x e y .

En la Figura 4.1 se puede observar un diagrama que muestra de manera más gráfica la selección e identificación de los clústeres de búsquedas visuales. A los costados se ven las dos alternativas de clusterización de descriptores, y al medio la clusterización de etiquetas. Las flechas indican la correspondencia de clúster de descriptor con clúster de etiquetas, según la condición descrita en el punto previo. En este ejemplo, la clusterización de descriptores realizada en el espacio de dos componentes tiene una mayor cantidad de correspondencias (3) en comparación al espacio de alta dimensión (2) y por ello es seleccionada. En la primera columna, de clústeres en alta dimensión, se puede ver como el último clúster no tiene ninguna correspondencia, ya que ningún clúster de etiquetas contiene al menos un 40% de sus búsquedas. Por último, los nombres de cada clúster seleccionado (de la columna de la derecha) serán calculados con las etiquetas de los clústeres que le corresponden (columna central).

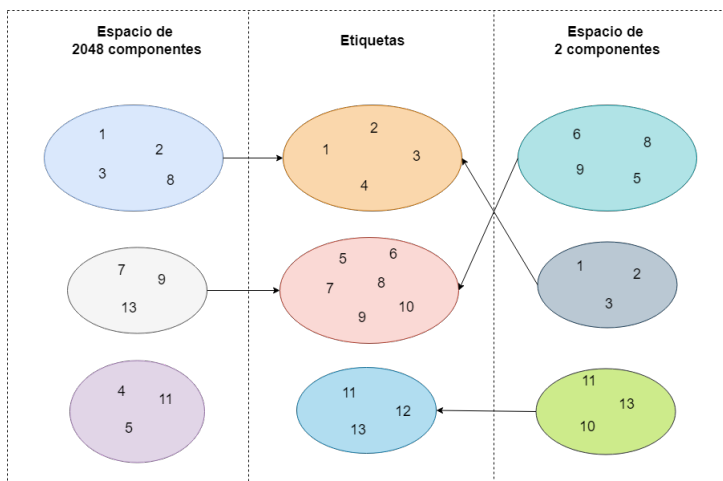


Figura 4.1: Diagrama de selección de clúster

Capítulo 5

Experimentos y Análisis de Resultados

5.1. Diseño de Experimentos y Evaluación

Para la evaluación de las diferentes herramientas implementadas, se necesita comparar con un *ground-truth*, por lo que se etiquetaron manualmente cada una de las consultas visuales del conjunto de estudio. También para otro análisis posterior, de clusterización, se agruparon y posteriormente etiquetaron, de la misma manera las etiquetas más relacionadas, del *ground-truth* recién mencionado, con el objetivo de generar un *ground-truth* de clústeres. Este último etiquetado contempló, la agrupación de etiquetas como ‘escritorio’ y ‘mesa’, para generar un clúster de productos más general, como por ejemplo ‘muebles’, que podría contener ambas etiquetas.

5.1.1. Datos para la evaluación

Para la evaluación de las diferentes partes de la solución propuesta, se cuenta con tres conjuntos de datos correspondientes a búsquedas visuales, realizadas en dos tiendas, clientes de Impresee, en diferentes periodos. La Tienda 1 se especializa en vender diferentes tipos de productos importados, mientras que la Tienda 2 vende productos para bebés.

1. Datos Tienda 1 - Agosto 2021

Este conjunto de datos consta de 270 imágenes de consultas, recibidas por la tienda durante el mes de Agosto del 2021. El catálogo consta de más de 8 mil productos y en la Figura 5.1 se pueden ver una muestra de las imágenes de consulta recibidas. Para cada imagen del conjunto se tienen los descriptores, *deep-feature* y los productos (nombre y categoría) más cercanos del catálogo de la tienda según el motor de búsqueda de Impresee.

Adicionalmente para este conjunto de datos, se etiquetó de manera manual, una única etiqueta (a partir del conjunto de palabras más repetidas en los nombres de productos del catálogo) para cada imagen de consulta. También se agruparon aquellas imágenes

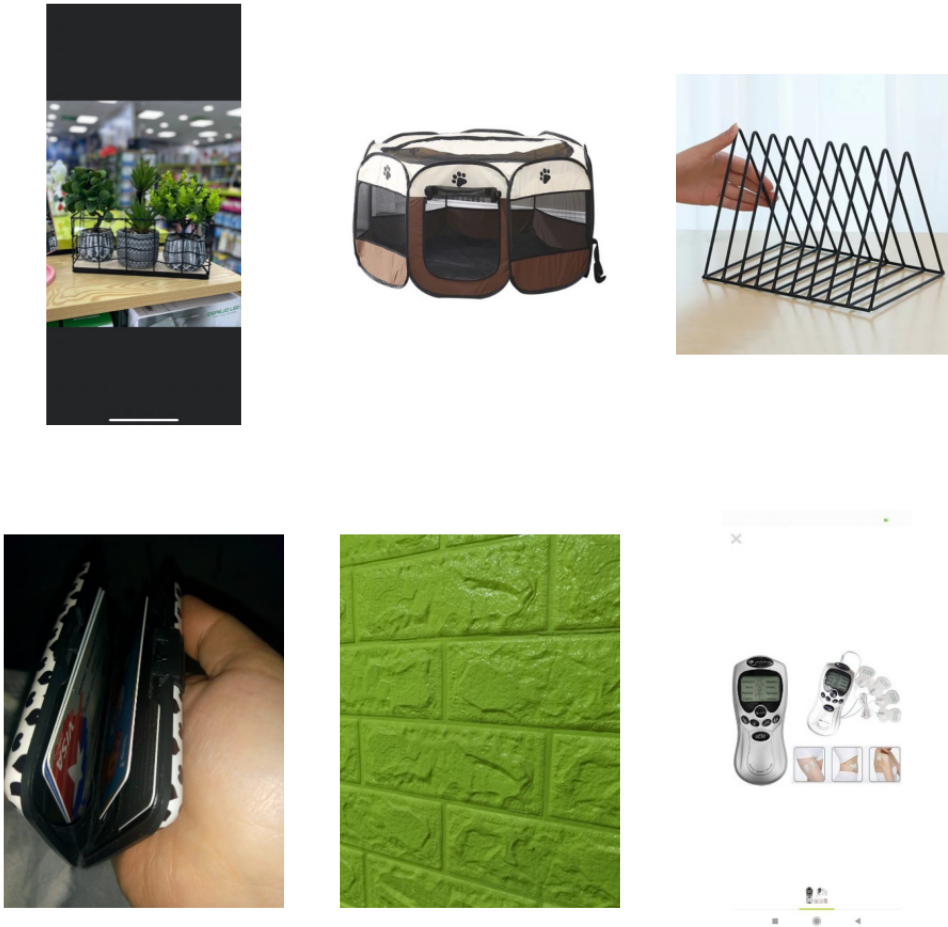


Figura 5.1: Muestra de imágenes de consultas durante Agosto 2021

que corresponden a etiquetas similares. Estos dos datos adicionales de cada consulta, será utilizado para evaluar el algoritmo de etiqueta y los análisis de clusterización.

2. Datos Tienda 2 - Noviembre 2021

Para este conjunto se tienen 112 imágenes de búsquedas, junto con sus descriptores y respuestas del buscador para cada una de las consultas. En la Figura 1 del Anexo A, se puede observar una muestra de las imágenes. Cabe mencionar que el tamaño del catálogo de esta tienda es de aproximadamente 2600 productos.

3. Datos Tienda 2 - Diciembre 2021

Este conjunto contiene datos similares al último descrito, pero este consta de 136 búsquedas visuales. En la Figura 2 del Anexo A, se puede ver una muestra de las imágenes.

5.1.2. Experimentos

Para cada parte de la solución propuesta en este trabajo, se exploraron diferentes alternativas que llevaron a elegir la implementación descrita en el capítulo anterior. Estas alternativas van desde variaciones en los algoritmos propuestos para ciertas partes, como también en la exploración de herramientas sustitutas como por ejemplo en el algoritmo utilizado para la

reducción de dimensionalidad de descriptores. A continuación, se describirán los diferentes experimentos junto con su respectiva forma de evaluación, realizados en cada componente de la solución implementada.

Etiqueta de búsquedas visuales y detección de Zero-Hits

1. Experimentos

Se realizaron cuatro experimentos que corresponden a dos algoritmos diferentes para la selección de etiqueta, y también variando el tamaño del subconjunto, entre 5 y 10, de productos más cercanos considerados a cada una de las búsquedas.

El primer algoritmo es el indicado en la Sección 4.2.2. El segundo algoritmo es similar al anterior pero separa del texto donde se analizan las frecuencias de palabras, las categorías asociadas a cada título. De esta forma se genera el mismo el mismo análisis de frecuencias locales e *IDF* sobre los títulos sin categorías de los productos más cercanos y también sobre solo las categorías de los productos. Esto para disminuir que palabras de los títulos sean potenciadas por nombre de categorías que aparecen pocas veces y viceversa.

2. Evaluación

Dado que se tiene solo una etiqueta de *ground-truth* para cada búsqueda por la complejidad de generar manualmente múltiples etiquetas para diferentes imágenes de consulta, se utiliza como métrica de evaluación: *precision@k*. Esta métrica considera un etiquetamiento correcto si este contiene la etiqueta del *ground-truth* entre las *k* primeras etiquetas calculadas por el algoritmo, con $k = 1, 3, 5$. Cabe mencionar que este experimento sólo se realizó con el primer conjunto de datos descrito en el Capítulo 5.1.1, dado que es el único conjunto con el que se tienen un *ground-truth* para comparar.

Clusterización

1. Experimentos

Para los experimentos de clusterización, tanto en el espacio de etiquetas como en el espacio de descriptores visuales (alta y baja dimensión), se utilizaron los algoritmos de clusterización **DBSCAN** y **HDBSCAN**. En particular para los descriptores visuales en el espacio de baja dimensión, se experimentó la clusterización con 2 alternativas más (**PCA** y **t-SNE**) a la propuesta en la solución de este trabajo (**UMAP**).

2. Evaluación

La evaluación de los clústeres se hizo de dos maneras, por un lado con las métricas ARI (adjusted rand index) y AMI (adjusted mutual information) y por otro lado de manera visual, con visualizaciones de los clústeres obtenidos. En particular, este experimento se realizó en el primer conjunto de datos descrito en Capítulo 5.1.1, ya que sólo este conjunto cuenta con un *ground-truth* con el que compararlo.

Selección de clústeres

1. Experimentos

Los experimentos para esta sección correspondieron a variaciones de las selecciones de los algoritmos de clústeres. Alternativamente al criterio de selección descrito en la Sección 4.2.5, se propone también un acercamiento inverso, en el cual se seleccionan clústeres de etiquetas resultantes de la Sección 4.2.3 y que estén contenidos en algún porcentaje en los clústeres de descriptores resultantes de la Sección 4.2.4. Para ambos escenarios de selección se experimenta con diferentes porcentajes de pertenencia entre los diferentes clústeres.

2. Evaluación

Dado que este resultado es el que se espera sea de utilidad para el dueño de la tienda, beneficiario final de este trabajo, se evalúa únicamente de manera visual. Cabe mencionar que esa la forma en que será utilizado este resultado. Dado que la evaluación consta de solo inspeccionar visualmente los resultados, este experimentó se realizó en los tres conjuntos de datos descritos en 5.1.1.

5.2. Resultados y Análisis

A continuación, se muestran y comentan los resultados obtenidos para cada uno de los experimentos descritos en la sección previa de este capítulo.

5.2.1. Detección de Zero-Hits

Primeramente, para la detección de Zero-Hits en el conjunto de datos de la Tienda 1 se obtuvieron los siguientes resultados.

Experimento	Algoritmo	Respuestas consideradas	Precision	Recall
1	Solución	5	63.8 %	20.2 %
2	Alternativo	5	74.1 %	16.6 %
3	Solución	10	18.9 %	68.7 %
4	Alternativo	10	37.9 %	23.4 %

Tabla 5.1: Resultados experimentos detección de Zero-Hits

Como se puede observar en la Tabla 5.1, donde están los resultados de detección de *Zero-Hits* para las variaciones propuestas para el algoritmo de etiquetado, el algoritmo elegido, considerando 5 respuestas, cuenta con una de las mayores precisiones detectadas y el mayor *recall*. La precisión nos indica que el 63,8 % de los *Zero-Hits* detectados, efectivamente lo es, mientras que el *Recall* indica que se detectaron un 20,2 % de todos los *Zero-Hits* que hay en el conjunto de datos.

De este resultado se puede observar que en general, mientras más *Zero-Hits* se detectan (mayor *recall*), más falsos positivos se encuentran (menor precisión). El algoritmo de la solución es el que logra el mayor *recall* entre los dos algoritmos con mayor precisión y por ello fue elegido.

5.2.2. Etiqueta de búsquedas visuales

Los experimentos de algoritmos de etiqueta de búsquedas visuales, en el conjunto de datos de la Tienda 1, dejaron los siguientes resultados.

Experimento	Algoritmo	Respuestas consideradas	P@1	P@3	P@5
1	Solución	5	31.9 %	48.1 %	49.6 %
2	Alternativo	5	21.4 %	42.0 %	46.9 %
3	Solución	10	24.3 %	36.6 %	39.2 %
4	Alternativo	10	16.6 %	33.8 %	39.2 %

Tabla 5.2: Resultados experimentos etiqueta de búsquedas

En la Tabla 5.2 se pueden ver los resultados de las otras variaciones de algoritmos de etiquetas, de las cuales se pueden observar que ninguna se comporta de mejor manera que la propuesta en la solución de este trabajo.

Respecto a este resultado, se puede observar que el algoritmo responde para sólo un 31,9 % su mejor etiqueta en primera posición, pero que, en general, para un 50 % de todas las búsquedas visuales ubica una etiqueta similar dentro de las 5 primeras etiquetas. Este resultado, no evalúa a cabalidad la capacidad del algoritmo, dado que este depende de los títulos que el buscador responde y es difícil lograr un *ground-truth* tan flexible, que represente el uso real, para evaluarlo. Sin embargo, este experimento evidencia igualmente la funcionalidad del algoritmo, aunque también que existe margen de mejora.

Por último, es evidente que el algoritmo de la solución, supera en cada una de las métricas a las alternativas propuestas. Esto confirmando, la elección de solución para resultado previo de detección de *Zero-Hits*, que hace uso del mismo algoritmo.

5.2.3. Clusterización

En la Tabla 5.3 se pueden observar todos los resultados de la experimentación de clusterización, tanto sobre el espacio de descriptores visuales en alta y baja dimensión, como también sobre el espacio de etiquetas. Para cada caso, se evaluaron los rendimientos de los algoritmos de clústerización propuestos, HDBSCAN y DBSCAN.

El ruido corresponde a la cantidad de búsquedas visuales que no pudieron ser identificadas dentro de ningún clúster según los algoritmos de clusterización.

En general, se puede observar que la mejor precisión al detectar clústeres se alcanza en el espacio de alta dimensión al costo de detectar como ruido, la mayoría de las búsquedas visuales. Por ello, la solución propuesta contempla una comparación entre los resultados de los experimentos 1 y 2 de la Tabla 5.3. Esto último permite revisar la posibilidad de considerar una mayor muestra (menor ruido) en el espacio de dos dimensiones de UMAP y si este resultado no es lo suficientemente bueno, entonces utilizar el espacio original que es más preciso pero más pequeño (mayor ruido).

N° Exp	Espacio	Algoritmo	ARI	AMI	% ruido
1	2048 (raw)	HDBSCAN	0.414	0.522	59.2
2	2 (UMAP)	HDBSCAN	0.332	0.388	18.6
3	2 (PCA)	HDBSCAN	0.001	0.103	22.8
4	2 (t-SNE)	HDBSCAN	-0.02	0.0003	5.7
5	2048 (raw)	DBSCAN	1.0	1.0	100.0
6	2 (UMAP)	DBSCAN	-0.014	0.0001	2.1
7	2 (PCA)	DBSCAN	0.001	0.103	22.8
8	2 (t-SNE)	DBSCAN	1.0	1.0	100.0
9	Etiquetas	DBSCAN	0.549	0.539	53.5
10	Etiquetas	HDBSCAN	0.204	0.294	11.4

Tabla 5.3: Resultado experimentos de clusterización

5.2.4. Selección de clústeres

En las Figuras 5.2, 5.3 y 5.4 se muestran algunos de los clústeres de búsquedas visuales seleccionados para cada uno de los conjuntos de datos propuestos. La totalidad de los resultados se pueden revisar en el Anexo B.

Como se puede observar de estos resultados, en general, estos tienen sentido e inspeccionando visualmente, sólo hay una cantidad mínima de clústeres ruidosos erróneamente agrupados. Así mismo, la etiqueta o nombre calculado para cada uno de los clústeres tiene relación con las imágenes observables en cada uno de ellos. Por último, este resultado se considera de utilidad, ya que ofrece una cantidad de clústeres totalmente manejable para que una persona, con mera inspección visual pueda determinar cuáles de estos realmente se les puede ofrecer a los dueños de tiendas usuarias de Impresee.

Visualización de búsquedas para clúster 'mueble'



Figura 5.2: Muestra de uno de los clústeres de búsqueda seleccionados para el conjunto de datos Tienda 1

Visualización de búsquedas para clúster maderas

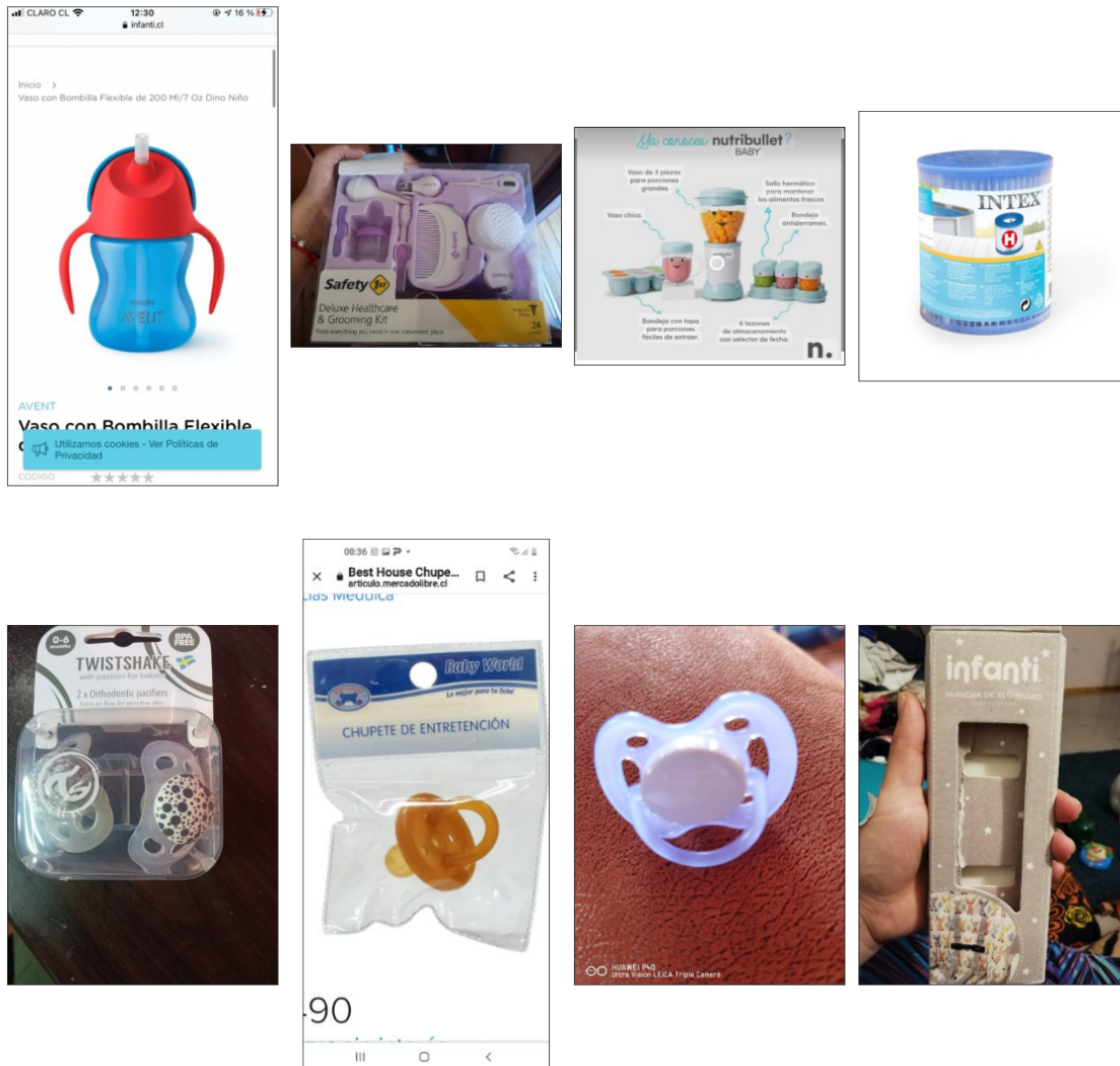


Figura 5.3: Muestra de uno de los clústeres de búsqueda seleccionados para el conjunto de datos Tienda 2 - Noviembre 2021

Visualización de búsquedas para clúster sillas silla

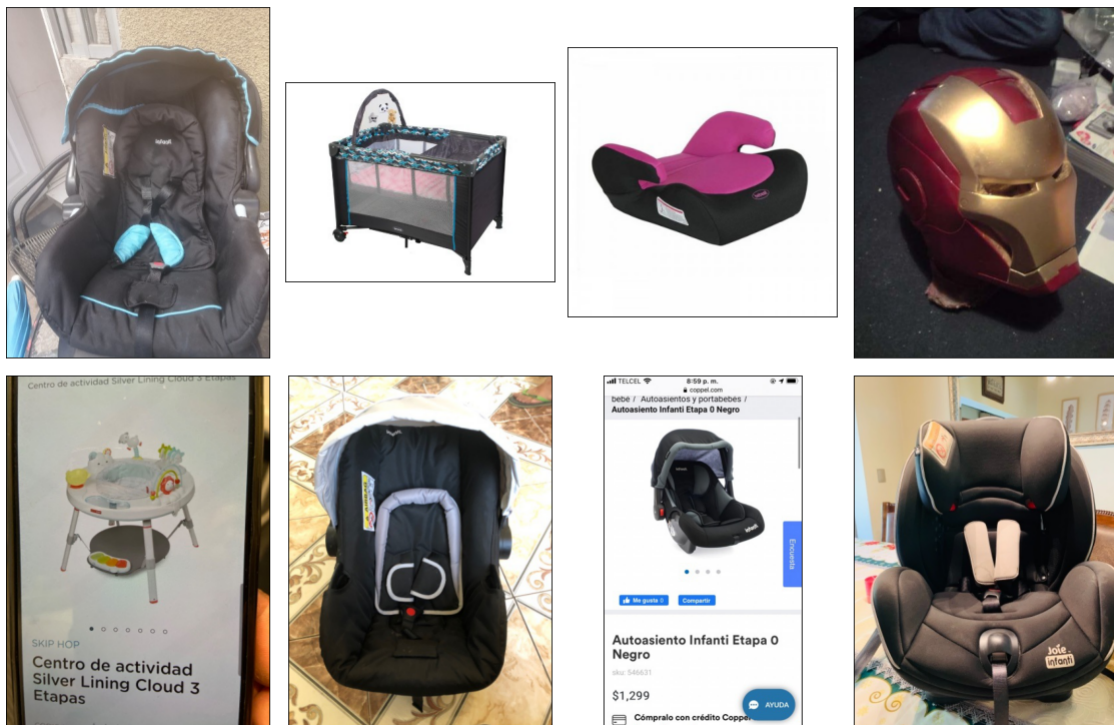


Figura 5.4: Muestra de uno de los clústeres de búsqueda seleccionados para el conjunto de datos Tienda 2 - Diciembre 2021

Capítulo 6

Conclusiones y Trabajo Futuro

Conclusiones

Respecto al trabajo realizado, se puede concluir que se cumplieron parcialmente cada uno de los objetivos propuestos. En primera instancia, se pasó de no poder reconocer *Zero-Hits* a lograr detectar el 20,2% de los casos de búsquedas corresponden a este caso, con una precisión del 63,5%. De la misma forma, se logró calcular etiquetas categóricas para imágenes de consultas, que alcanzan una *Precision@5* del 49,6%. Por último, se implementó un algoritmo que logra agrupar e identificar con un nombre, distintos conjuntos de imágenes de búsquedas, que se refieren a productos similares.

La consecuencia más importante y destacable de este trabajo, es la posibilidad de enriquecer los reportes de usos del buscador que entrega Impresee a sus clientes. Ahora será posible ofrecerles, a los administradores de tiendas, una visualización más ordenada en la que se encuentran grupos identificados de búsquedas visuales similares. Este resultado detecta en su gran mayoría grupos bien identificados de búsquedas similares, sin embargo, hay un pequeño porcentaje de grupos que son incorrectamente agrupados o identificados. Si se buscara entregar un resultado perfecto a los administradores de tiendas, el resultado ofrecido por este trabajo podría no ser suficiente, sin embargo, deja una base sólida sobre la cual se podrían incluso filtrar de manera manual aquellas detecciones erróneas. Finalmente, este resultado se sigue considerando como positivo dada la situación previa a este trabajo, en la que no se ofrecía ninguna información adicional más allá de las imágenes de consultas recibidas por las tiendas.

Trabajo Futuro

Pese a que, en general, los resultados de este trabajo fueron positivos, a continuación, se proponen algunas mejoras que podrían implementarse en el futuro para incrementar aún más la utilidad de la solución propuesta.

En el corto plazo, algo importante que debe realizarse, consta de evaluar los resultados de este trabajo con otros clientes (tiendas) de Impresee. En general, las pruebas contempladas en la realización de este trabajo corresponden a dos tiendas diferentes pero que comparten la característica de ser tiendas con catálogos variados, que reciben consultas visuales variadas. De existir una tienda muy especializada que reciba, por ejemplo, fotos o imágenes de consulta muy similares, estas podrían llegar a no ser clusterizables. En este último caso mencionado, el resultado de este trabajo podría no comportarse de la manera esperada.

Dado la naturaleza de este trabajo, el cual constó de ir seleccionando los resultados que mejor se comportaban con los datos con que se contó, es que es un trabajo perfectible en general. Con una mayor cantidad de tiempo se podría explorar con mayores cambios en diferentes partes de la solución. La parte de la solución que potencialmente tiene más margen de mejora es la detección de *Zero-Hits* como también el etiquetado de búsquedas. Experimentar con más variaciones en estos algoritmos, podría incrementar la calidad de clústeres de etiquetas y, eventualmente, incrementar los resultados que utilizan la clusterización de etiquetas.

Por último, como otra alternativa al etiquetado de búsquedas visuales, se podría experimentar con una clasificación de búsquedas visuales, independientes del buscador. Para esto, podría entrenarse una red neuronal clasificadora de todo tipo de búsquedas visuales, con imágenes del catálogo de todas las tiendas de Impresee, para hacerlo lo más general posible. Contar con un método de etiquetado de búsquedas visuales mejorado e independiente del buscador, podría enriquecer los resultados posteriores, de este trabajo, que hacen uso de este mismo. Esto último, podría verse reflejado en una selección final de clústeres con incluso menor ruido que podría permitir hacer uso de este resultado de manera directa.

Bibliografía

- [1] Charts: How the coronavirus is changing e-commerce. <https://www.digitalcommerce360.com/2021/02/19/e-commerce-during-coronavirus-pandemic-in-charts/>. Visitada: 2021-07-10.
- [2] Dimensionality reduction for data visualization: Pca vs tsne vs umap vs lda. <https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29>. Visitada: 2021-07-10.
- [3] E-commerce latinoamérica: En tiempos del covid-19 2020. https://latam.payu.com/reporte-covid?utm_source=landingpage&utm_medium=referral&utm_campaign=latam-pr-publisher---reporte_impacto_covid_v3-datadriven_content_e-commerce_covid_impact. Visitada: 2021-07-10.
- [4] tsne vs. umap: Global structure. <https://towardsdatascience.com/tsne-vs-umap-global-structure-4d8045acba17>. Visitada: 2021-07-10.
- [5] Umap: Uniform manifold approximation and projection for dimension reduction. <https://umap-learn.readthedocs.io/en/latest/>. Visitada: 2021-07-10.
- [6] Martin Ester, Hans-Peter Kriegel, Joerg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Int. Conf. Knowl. Discovery and Data Mining*, 1996.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.
- [8] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [9] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2020.
- [10] S.A. Oyewole and O.O. Olugbara. Product image classification using eigen colour feature with ensemble machine learning. *Egyptian Informatics Journal*, 18, 2019.
- [11] Pranjal Singh Rajput and Shivangi Aneja. Indofashion : Apparel classification for indian ethnic clothes. *arXiv:2104.02830*, 2021.

- [12] Campello R.J.G.B., Moulavi D., and Sander J. Density-based clustering based on hierarchical density estimates. *Pei J., Tseng V.S., Cao L., Motoda H., Xu G. (eds) Advances in Knowledge Discovery and Data Mining.*, 2013.
- [13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, , and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015.
- [14] Brian Tomasik, Phyo Thiha, and Douglas Turnbull. Tagging products using image classification. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, pages 792–793, New York, NY, USA, 2009. ACM.*, pages 197–204, 2009.
- [15] Venkatesh Umaashankar, Girish Shanmugam S, and Aditi Prakash. Atlas: A dataset and benchmark for e-commerce clothing product categorization. *arXiv:1908.08984*, 2019.
- [16] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9, 2016.
- [17] Mikolaj Wicczorek, Barbara Rychalska, and Jacek Dabrowski. On the unreasonable effectiveness of centroids in image retrieval. *arXiv:2104.13643*, 2021.
- [18] Pasawee Wirojwatanakul and Artit Wangperawong. Multi-label product categorization using multi-modal fusion models. *arXiv:1907.00420*, 2019.
- [19] Tom Zahavy, Alessandro Magnani, Abhinandan Krishnan, and Shie Mannor. Is a picture worth a thousand words? a deep multi-modal fusion architecture for product classification in e-commerce. *arXiv:1611.09534*, 2016.

Anexo A



Figura 1: Muestra de imágenes de consultas en Tienda 2 durante Noviembre 2021

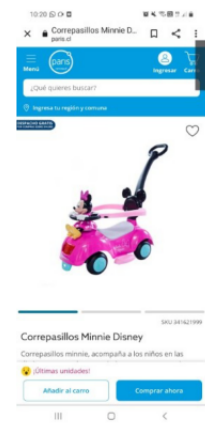
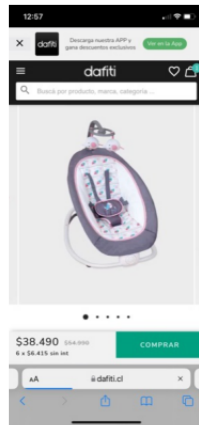


Figura 2: Muestra de imágenes de consultas en Tienda 2 durante Diciembre 2021

Anexo B

Visualización de búsquedas para clúster 'silla'

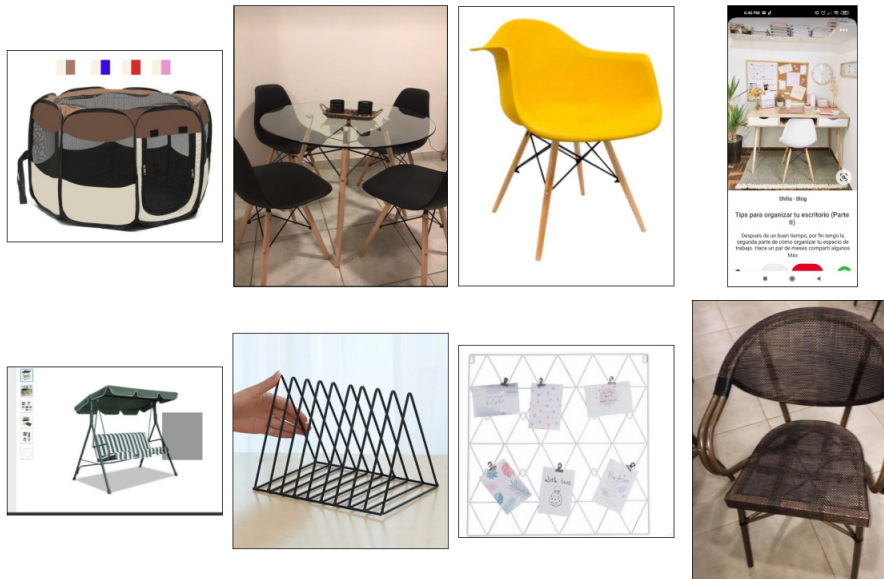


Figura 3: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021

Visualización de búsquedas para clúster 'mueble'



Figura 4: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021

Visualización de búsquedas para clúster 'silla'

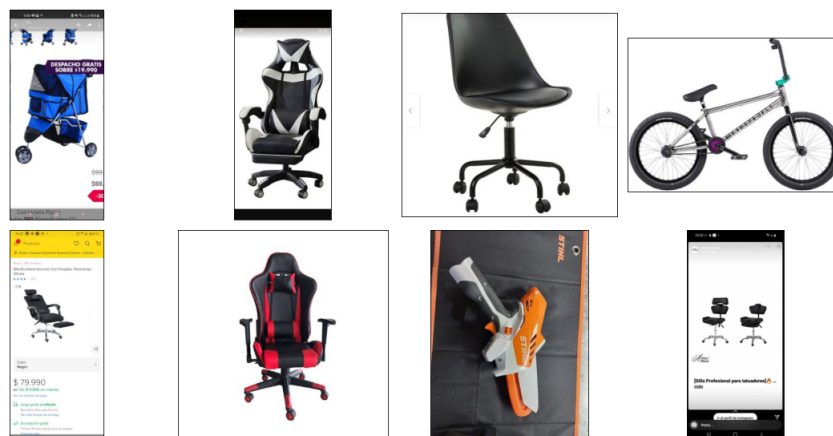


Figura 5: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021

Visualización de búsquedas para clúster 'mascaras mascara halloween'



Figura 6: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 1 - Agosto 2021

Visualización de búsquedas para clúster (cuna,playard) cuna

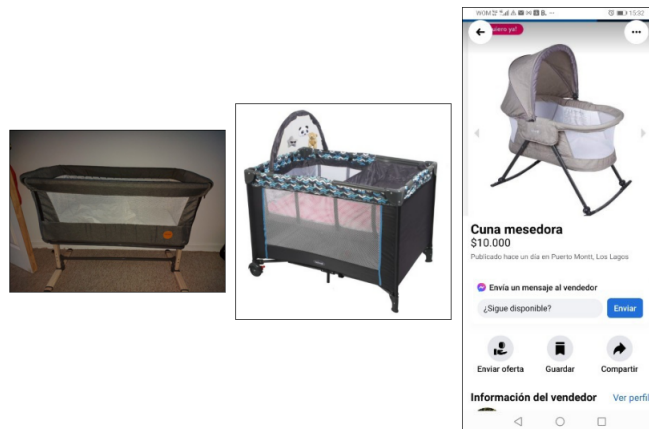


Figura 7: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021

Visualización de búsquedas para clúster mamaderas

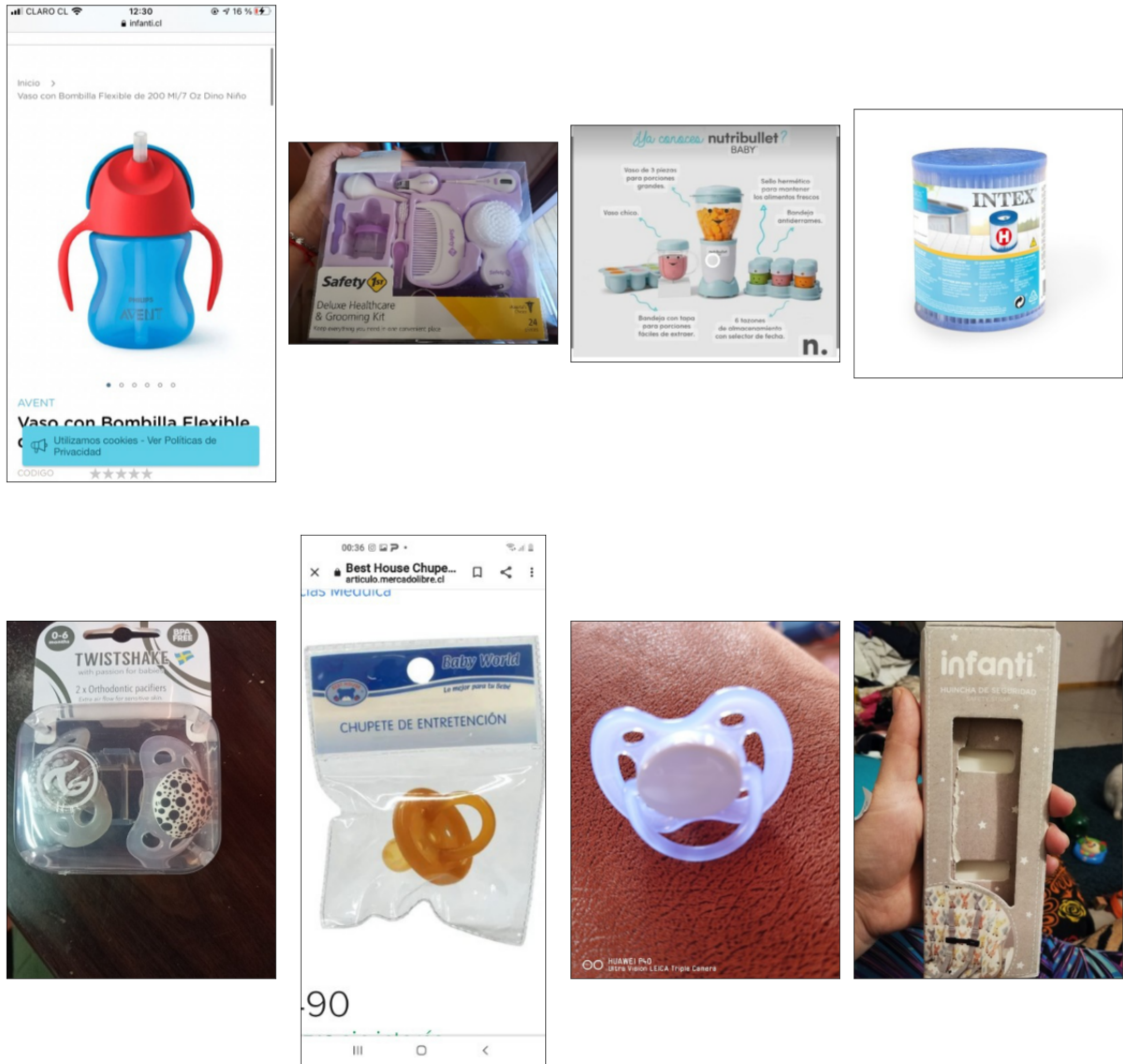


Figura 8: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021

Visualización de búsquedas para clúster sillas (sillas,auto) silla

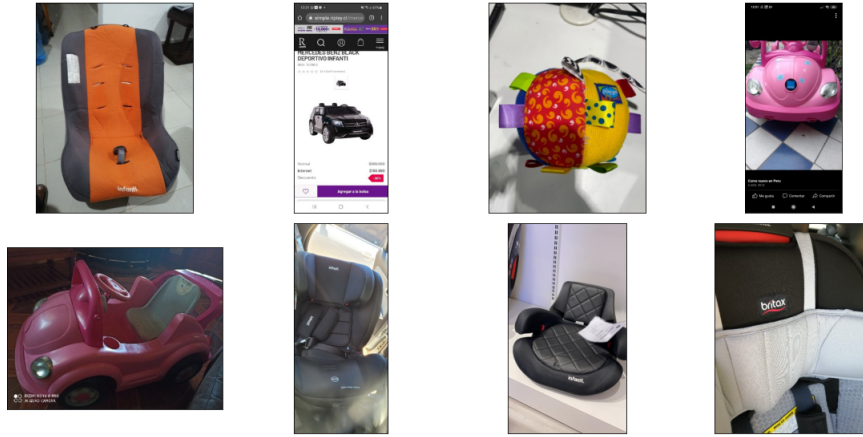


Figura 9: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021

Visualización de búsquedas para clúster coches coche

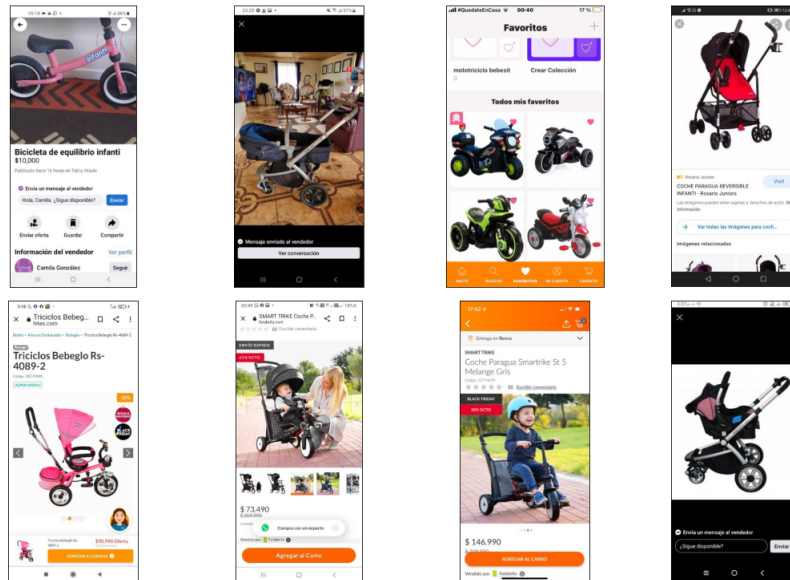


Figura 10: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Noviembre 2021

Visualización de búsquedas para clúster mamera mamaderas

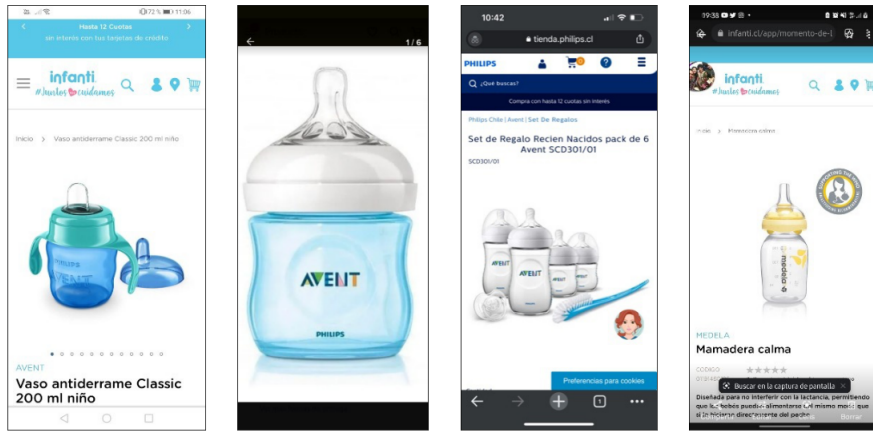


Figura 11: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021

Visualización de búsquedas para clúster sillas silla

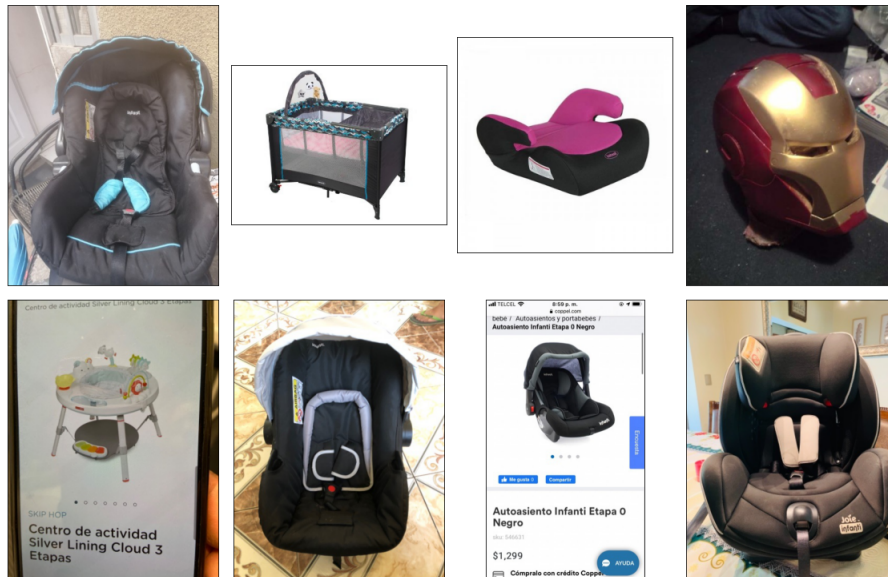


Figura 12: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021

Visualización de búsquedas para clúster coche system (travel,system)

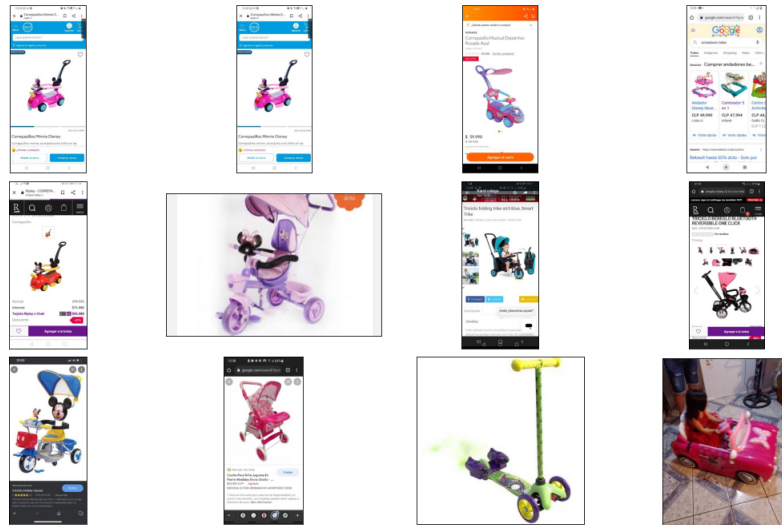


Figura 13: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021

Visualización de búsquedas para clúster gimnasio gimnasios

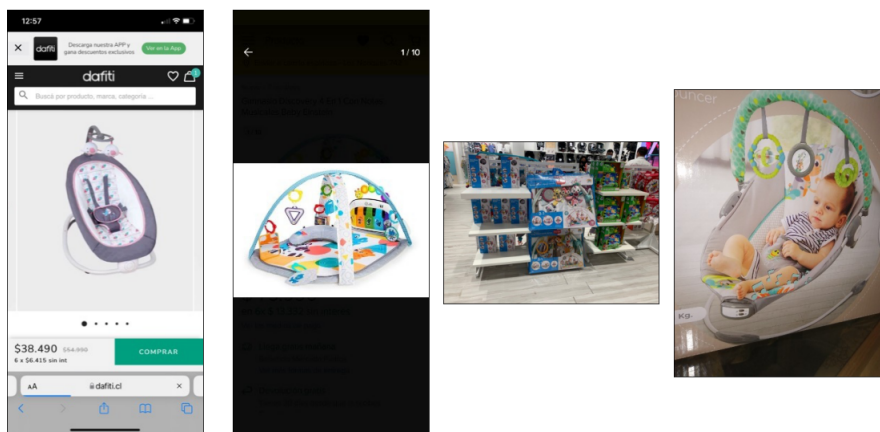


Figura 14: Clúster de búsqueda seleccionado para el conjunto de datos Tienda 2 - Diciembre 2021