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

ENTRENAMIENTO Y GENERACIÓN DE EMBEDDINGS PARA LA RECOMENDACIÓN DE PROPIEDADES EN UN MARKETPLACE INMOBILIARIO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

OMAR ALEJANDRO MIRANDA PAREDES

PROFESOR GUÍA:
JUAN BARRIOS NÚÑEZ

MIEMBROS DE LA COMISIÓN:
TOMÁS BARROS ARANCIBIA
ANDRÉS ABELIUK KIMELMAN

SANTIAGO DE CHILE
2024

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN
POR: OMAR ALEJANDRO MIRANDA PAREDES
FECHA: 2024
PROF. GUÍA: JUAN BARRIOS NUÑEZ

ENTRENAMIENTO Y GENERACIÓN DE EMBEDDINGS PARA LA RECOMENDACIÓN DE PROPIEDADES EN UN MARKETPLACE INMOBILIARIO

Los sistemas recomendadores son algoritmos que predicen o personalizan los ítems de un usuario en base a sus preferencias de un determinado producto. En el rubro inmobiliario, la implementación de estos sistemas debe cumplir los requerimientos del usuario para concretar la conversión del ítem pero considerando que la retroalimentación por parte de este es escasa, lo cual dificulta el entrenamiento y validación del sistema para entregar recomendaciones precisas. Este proyecto fue realizado en la empresa Goplaceit, una plataforma tecnológica con amplia gama de anuncios inmobiliarios para arriendo o venta de propiedades por personas naturales o profesionales del rubro.

El presente trabajo consiste en generar representaciones basadas en embeddings, calculados a partir de la actividad de los usuarios en el sitio, para ser utilizados en un contexto de recomendación de propiedades, denominado modelo de asistencia inmobiliaria, que recibe información de un usuario con un conjunto de propiedades y las ordena de forma decreciente según el interés de este.

Para la generación de los embeddings se requiere de sesiones de clicks a propiedades realizadas por los usuarios para luego entrenar una red neuronal que toma como entrada y salida distintos pares obtenidos de la actividad de los usuarios en el sitio. De esta red, se obtiene una capa de pesos para calcular los embeddings que representan a las propiedades, y por otro lado, para representar a los usuarios como un promedio de sus últimas visitas a propiedades. Finalmente, se utilizan estas representaciones en una API que permite ordenar un conjunto de propiedades candidatas para un cierto usuario.

A pesar de que actualmente el asistente inmobiliario no está disponible para uso comercial por disposición de la empresa involucrada, se cumplieron los objetivos propuestos en el plazo establecido.

We must try not to sink beneath our anguish,
but battle on.

Albus Dumbledore

Agradecimientos

Ha sido bastante largo y postergado el camino que he recorrido para llegar acá. Valoro y agradezco cada paso que di y las lecciones aprendidas. Conocerme, caerme y levantarme ha sido el mejor regalo que me dejado la tesis/memoria. Gracias a ti mismo Omar, por (sobre)vivir, darte otra oportunidad, atreverte y seguir adelante. En el camino me acompañaron muchas personas, quisiera dedicar unas palabras y agradecimientos a continuación.

A mis compañeros del DCC, quienes ya son compañeros de adultez. A Boris, Jazmine, Blasco, Pancho y Camilo, después de tantos años de ustedes titulados llega el momento de abrirme y agradecerles de todo corazón por tantas aventuras y apañe a lo largo de los años.

A mis amigas, Coni, Jesu y Vale, por ser un apoyo fundamental e incondicional dentro de toda la estadía universitaria y hasta el día de hoy. Por la comprensión y apoyo que han demostrado siempre y que sigue intacta. A la Javi, quién fue un apoyo anímico y motivacional en todo este proceso de crecimiento personal que ha sido la tesis.

A Felipe Vargas, por toda la confianza puesta en mí como profesional y confiar en mis capacidades dentro y fuera de lo laboral. Por cada desafío compartido en goplaceit y muchos que quedan por delante, junto al equipo de datos. Gracias también a Carlos, Gris, Roberto y Pablo. También a quienes nos entregaron su aporte y comentarios en este proyecto que transciende esta memoria: Denis, Manuel, Iván e Ignacio.

A mi profesor guía, Juan, quién aceptó apoyarme incluso en las circunstancias con las que llego a realizar la memoria. A todo profesor, profesora y funcionarios del DCC, gracias por su trabajo.

A mi pareja, la Fran, que desde que nos conocimos y estamos juntos ha sido testigo de cerca de lo que ha significado todo este proceso por el que he pasado. Gracias por amarme, acompañarme, contenerme, aconsejarme, motivarme, estar y no rendirte. Y a nuestra Blanca, por hacerme compañía las noches de turno o escritura. Las amo.

Finalmente a mi familia, que los amo mucho. Mi mamá y papá, gracias por hacer lo mejor que pudieron para en cada momento darme lo mejor. La educación y ejemplo que me dieron, tiene gran influencia en la persona que soy hoy. Gracias por todo esfuerzo y sacrificio para darnos un mejor vivir. Al David y Rafa, que me llena de orgullo verlos lograr sus metas y saber que muchas veces me veían a mi como referente; pero es al revés, ustedes son mis referentes. A mis tatas, hay mucho de ellos también en mi y en mi educación.

A todas y cada una de ellas, gracias.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

Buscar una propiedad para arrendar o comprar, es una decisión que no se toma a la ligera. Existen diversos factores que influyen en esta decisión, desde lo lógico o racional, como el poder adquisitivo o tamaño mínimo de la propiedad; hasta intereses más emocionales, como predilección por un barrio o estilo de construcción en particular.

Cualquiera sea el caso, gracias a las plataformas inmobiliarias de estilo *marketplace*, el proceso de búsqueda y publicación de propiedades se realiza de una manera simple y a gran escala; permitiendo al usuario utilizar filtros de búsqueda para encontrar según sus intereses.

Este trabajo de memoria fue desarrollado en la empresa Goplaceit, perteneciente al holding Capitalizarme. Goplaceit, es una plataforma tecnológica de anuncios con presencia en siete países de Latinoamérica y casi diez años de historia. Dicha empresa cuenta con una amplia gama de publicaciones para arriendo o venta de propiedades tales como casas, departamentos, terrenos u oficinas nuevas o usadas; además, permite que personas naturales o profesionales del rubro, como corredores de propiedades e inmobiliarias, puedan disponer de la plataforma realizando publicaciones que estarán visibles para los miles de usuarios con los que cuenta.

La plataforma dispone de una versión web y una aplicación móvil, donde el componente principal es el mapa, donde el usuario puede explorar visualmente las publicaciones, seleccionar filtros de características, dibujar zonas de interés, entre otras. Esto le permite al usuario explorar de una forma más dinámica y natural, permitiendo así *recorrer* la ciudad buscando nuevas oportunidades que se adapten a sus requerimientos. Aún así, debido a la gran cantidad de publicaciones con las que cuenta la plataforma y la gran cantidad de opciones que un usuario promedio debe mirar antes de encontrar la vivienda indicada, es relevante incorporar un elemento de personalización basada en los intereses personales del usuario.

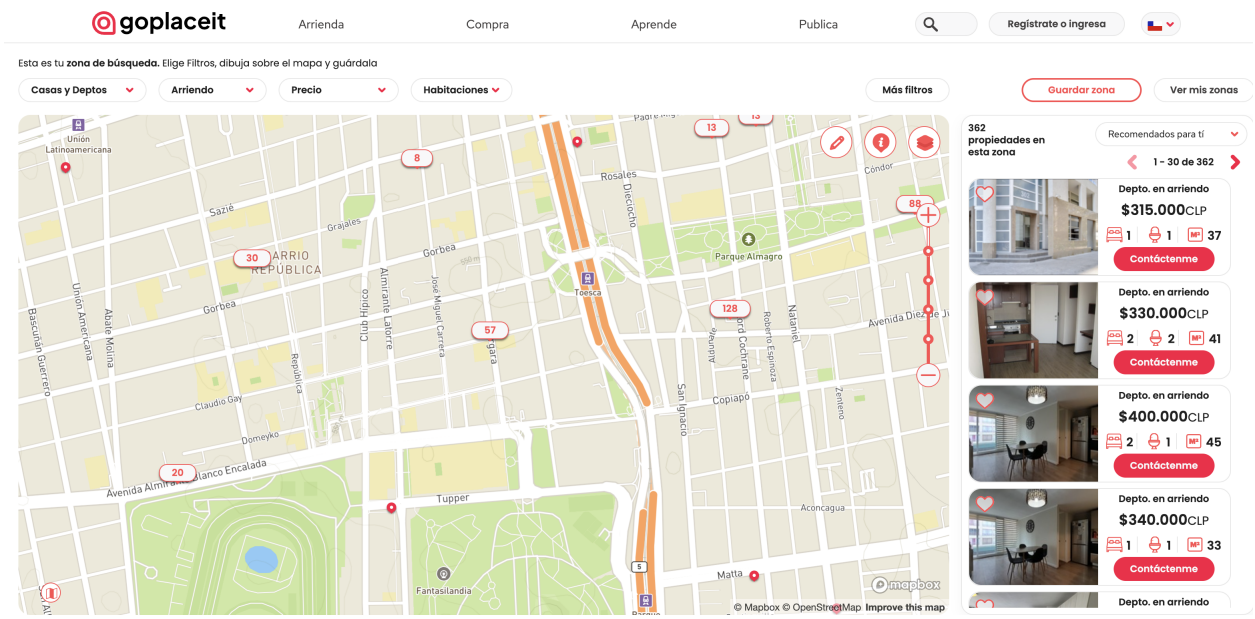

Figura 1.1: Vista principal del mapa de la plataforma

La empresa Goplaceit, al ser catalogada como una empresa tecnológica, adquieren una cultura donde el producto gira en torno a los datos, considerando desde el contenido que aportan los usuarios, como el almacenamiento, procesamiento y valor agregado que se pueda obtener de ello. Gracias a esto, se cuenta con una amplia cantidad de datos históricos guardados, siendo su principal activo: las propiedades publicadas. Esto permite que se puedan tomar decisiones o diseñar mejoras y soluciones orientadas a los datos.

En base a lo anteriormente expuesto, es que se genera la oportunidad de resolver el problema de mejorar la entrega de recomendaciones a los usuarios, utilizando técnicas basadas en datos, en particular técnicas que utilizan minería de datos o machine learning. Sin embargo, es importante mencionar que existen grandes desafíos para el correcto funcionamiento del sistema recomendador que pueden determinar el éxito o fracaso de este, como lo es la disponibilidad de datos de retroalimentación o también el *cold-start problem* dentro de los más frecuentes. Además se suma las dificultades dadas por el rubro donde se está desarrollando este sistema, que tiene una naturaleza y versatilidad que se diferencia de otros mercados.

Una de las grandes problemáticas existentes al trabajar con datos de una plataforma inmobiliaria, es que funciona solo como punto de encuentro para anunciantes y consumidores que no forman parte del acuerdo comercial que se puede concretar, es decir, en la venta o arriendo del inmueble. Esto dificulta el poder determinar una *conversión*, en comparación a otros rubros donde su conversación se basa en la compra de un producto, ver hasta el final una película o dar un *me gusta* en una publicación. Además, luego de que un usuario utiliza la plataforma hasta *convertir* (comprar, vender o arrendar), disminuye considerablemente las probabilidades de que este utilice la plataforma en el corto plazo, lo cual limita las recomendaciones que se puedan generar en la plataforma en base a sus intereses porque constantemente se están implementando en base a usuarios nuevos.

Al comienzo, los intentos de recomendación fueron con un enfoque hacia la subasta de anuncios (utilizando la técnica de BPR [1]). Estos intentos si bien fueron un punto de parti-

da, no prosperaron en el tiempo por cambios en el modelo de negocio, además de dificultades técnicas para el desarrollo relacionadas al elevado costo del entrenamiento y mantención de este. Ante esta situación y considerando las lecciones aprendidas se replanteó el proyecto hacia la representación de propiedades y/o usuarios como embeddings que pudiesen servir para distintas tareas. Este hallazgo fue altamente motivado por el creciente uso de las redes neuronales para tareas de recomendación, como lo es el caso particular de YouTube [2].

Adecuando al contexto del rubro inmobiliario y de la información que se dispone en Goplaceit, se decidió abordar el problema en dos partes:

Primero, generar una metodología para representar como embedding las propiedades de la plataforma, considerando por una parte los intereses de un usuario y por otro la similitud entre propiedades. Segundo, utilizar estas representaciones generadas, para recomendar a un usuario un conjunto de propiedades en las que potencialmente debería estar interesado.
De esta manera se plantean los siguientes objetivos para este trabajo.

1.1. Objetivos

1.1.1. Objetivo general

Generar un sistema recomendador de propiedades para un usuario, utilizando una representación vectorial del tipo embedding entrenada a partir de una red neuronal que capture intereses latentes de los usuarios mediante su historial de visitas.

1.1.2. Objetivos específicos

- Explorar y caracterizar los datos de usuarios y propiedades existentes dentro del contexto definido.
- Diseñar una estrategia de ordenamiento básica (baseline), tomando en consideración el análisis de datos del objetivo anterior.
- Generar y evaluar una representación vectorial (embedding) para las propiedades, utilizando una red neuronal que considere características asociadas a la propiedad, de su entorno y visitas de los usuarios.
- Desarrollar un sistema que ordene un conjunto de propiedades (utilizando sus embeddings) respecto a un usuario y su correspondiente representación como embedding.
- Evaluar mediante datos, la solución desarrollada contra un ordenamiento básico (baseline).

1.2. Organización del documento

En este documento se presenta el trabajo realizado, considerando las diversas partes tanto metodológicas como prácticas del proyecto, el cual es desempeñado como trabajador a tiempo completo de la empresa involucrada, y bajo ese contexto se presentan los principales hallazgos y aportes realizados.

En el segundo capítulo se presenta una descripción general de lo que son los sistemas recomendadores y trabajos aplicados en el rubro inmobiliario. También se presentan hallazgos relevantes encontrados en la revisión bibliográfica, incluyendo conceptos teóricos utilizados y formas de evaluación.

En el capítulo 3, se presenta el problema en detalle y los posibles casos de uso que se pudiesen dar.
En el capítulo 4 se presenta la solución al problema que consta de dos partes, abarcando distintas partes de manera general y particular, desde la naturaleza de los datos hasta el entrenamiento y uso de los embeddings generados.

Finalmente, en el capítulo 5, se presentan las principales conclusiones y aprendizajes de este trabajo, cerrando con una visión futura sobre las posibles mejoras al mismo.

Capítulo 2

Estado del Arte

La primera necesidad que permite la ejecución de este proyecto es desarrollar e incorporar en la plataforma de Goplaceit un sistema que permita entregar recomendaciones de propiedades a los usuarios basadas en los intereses que estos hayan determinado para la búsqueda de la propiedad. Para abordar esta problemática, se realizó una búsqueda sistemática de literatura sobre los sistemas recomendadores a modo general y posteriormente la información disponible en el rubro inmobiliario, preferentemente en aquellas plataformas del tipo marketplace o similares. Además, se presentarán conceptos técnicos básicos y relevantes para el entendimiento del trabajo.

2.1. Marco Teórico

Machine Learning

El aprendizaje automático (Machine Learning, ML) es una rama de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos matemáticos que permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas a través de la experiencia, sin necesidad de ser programadas explícitamente para ello. A diferencia de utilizar reglas específicas, en ML se utilizan datos para entrenar y ajustar modelos, que son luego utilizados para realizar predicciones, tomar decisiones o reconocer patrones de manera autónoma.

Normalmente, el proceso de ML generalmente implica:

- **Recopilación y procesamiento de datos:** adquirir y preparar un conjunto de datos que contenga idealmente ejemplos de resultados esperados
- **Entrenamiento del modelo:** Utilizar estos datos para entrenar un modelo matemático, que puede ser una red neuronal, un árbol de decisión, una regresión, entre otros, para aprender patrones y relaciones en los datos.
- **Evaluación y ajuste:** Evaluar el rendimiento del modelo en datos no utilizados durante el entrenamiento (conjunto de prueba) y ajustar sus parámetros para mejorar su precisión.

- **Predicción o Inferencia:** Utilizar el modelo entrenado para realizar predicciones o tomar decisiones en nuevas situaciones basadas en la información que ha aprendido del conjunto de datos de entrenamiento.

En el punto de evaluación y ajuste, el conjunto de datos se divide en dos subconjuntos, de entrenamiento y de prueba. El primero se utiliza para enseñar al modelo, mientras que el conjunto de prueba se utiliza para evaluar su rendimiento en datos no vistos previamente. Usualmente se divide el conjunto de datos bajo una relación 70/30 u 80/20.

Sin embargo, existe una técnica de entrenamiento y validación que genera modelos más robustos a la generalización. Se llama validación cruzada (cross-validation), donde el conjunto de datos iterativamente se participa en múltiples conjuntos (folds), utilizando diferentes combinaciones de ellos como conjuntos de prueba y entrenamiento. Esto implica que se entrena varias veces el modelo utilizando distintos folds, produciendo así un modelo más robusto frente a problemas de sobreajuste (overfitting) o subajuste (underfitting), generalizando mucho mejor a nuevos datos a los que no se ha expuesto.

Con el transcurso de los años han aparecido variantes en las técnicas y algoritmos que permiten resolver esta problemática, incorporando nuevos conocimientos, tecnología y posibilidades de abordaje, a modo de ejemplo, una de estas variantes son la incorporación de las redes neuronales y su versatilidad en aplicaciones como también en sistemas de recomendación.

Redes neuronales artificiales

Las redes neuronales son modelos matemáticos y computacionales que consisten en capas de unidades interconectadas llamadas neuronas artificiales. Estas neuronas procesan datos de entrada mediante cálculos ponderados y aplican funciones de activación para generar salidas. Las conexiones entre las neuronas tienen pesos ajustables que se adaptan durante el entrenamiento.

Las redes neuronales se utilizan para resolver problemas de aprendizaje automático, como clasificación, regresión y reconocimiento de patrones, al aprender patrones y representaciones de datos complejos y no lineales. Son ampliamente aplicables en campos como la visión por computadora, el procesamiento de lenguaje natural y más debido a su capacidad para modelar relaciones sofisticadas en datos.

Un ejemplo de la potencialidad de las redes neuronales, que además motivó a realizar este trabajo, fue el sistema recomendador que utiliza YouTube para recomendar videos [2]. En este trabajo, se presenta el valor que tiene el utilizar una representación vectorial del tipo embedding para caracterizar al usuario y a los videos, además de poder agregar información y características naturales sobre los mismos, ya sea la edad, el tiempo promedio de estadía por vídeo, entre otras.

Embeddings

Un *embedding* es una representación numérica de alta dimensión de un objeto, como una palabra, una frase, un documento o incluso una entidad en un espacio vectorial. Esta representación se utiliza comúnmente en procesamiento de lenguaje natural y otras disciplinas,

donde principalmente un *embedding* busca capturar las relaciones semánticas y estructurales entre objetos, de manera que objetos similares están cerca en el espacio vectorial creado donde *viven* estas representaciones.
Los embeddings ayudan a incorporar información que es difícil de representar numéricamente, como los datos discretos de alta dimensionalidad. De esta manera vectores de alta dimensión y comúnmente *sparse* (como vectores del tipo one-hot, donde cada dimensión representa un valor), son llevados a un espacio continuo y donde mantienen alguna relación del tipo semántica preferentemente.
Dentro de los trabajos mas notables de creación de embeddings, corresponde a Word2Vec, donde se trabaja con vectores de tipo one-hot encoding de palabras y se logra pasar de un espacio dimensional del tamaño de vocabulario, donde los vectores tienen solamente 0s y un valor 1 en la dimensión que representa esa palabra; a un espacio de menor dimensionalidad, con vectores continuos, que representan a una palabra.

En el caso del rubro inmobiliario, existen variables como la tipología o la modalidad, o la comuna donde está ubicada una propiedad, todas corresponden a valores categóricos, que si bien pueden ser representados mediante vectores del tipo one-hot, no es computacionalmente amigable trabajar con ellos, principalmente por la carencia de significado en de las dimensiones y la famosa "maldición de la dimensionalidad", que nos dice que a mayor dimensiones (a mayor datos únicos del tipo categórico que representar, mayor cantidad de datos se necesita para tener un buen resultado y el modelo resultando será altamente dimensional, lo que complejiza aún más el modelo mismo.

Métricas de Evaluación en Sistemas de Recomendación

En el ámbito de los sistemas de recomendación y sistemas de recuperación y búsqueda de información, la evaluación precisa y efectiva es fundamental para comprender la eficacia de los modelos propuestos. Entre las métricas más relevantes se encuentran Precision@K, mAP (Mean Average Precision) y Reciprocal Rank, cada una aportando perspectivas valiosas sobre la calidad y precisión de las recomendaciones realizadas.

Precision@K (Precisión en K)

Precision@K se refiere a la proporción de elementos relevantes presentes en los primeros K elementos de una lista recomendada. Matemáticamente, se define como:

$$\text{Precision@K} = \frac{\mid \text{Elementos relevantes en los primeros K} \mid}{K} \tag{2.1}$$

Esta métrica es crucial para comprender la efectividad de las recomendaciones realizadas en un conjunto limitado de resultados.

mAP (Mean Average Precision - Promedio de la Precisión Media)

mAP es una métrica que ofrece una visión general del rendimiento del sistema de recomendación al considerar la precisión promedio a lo largo de múltiples conjuntos de datos de prueba. Se calcula como el promedio de las precisiones promedio (AP) para cada usuario o consulta. Matemáticamente:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{2.2}$$

donde N es el número total de consultas o usuarios, y AP_i representa la precisión promedio para la consulta o usuario i.

Reciprocal Rank (Ranking Recíproco)
Reciprocal Rank evalúa la calidad del primer elemento relevante en la lista de recomendaciones al tomar su inversa. Matemáticamente, se define como:

$$\text{Reciprocal Rank} = \frac{1}{\text{Rango del primer elemento relevante}} \tag{2.3}$$

Esta métrica pone énfasis en la relevancia y posición del primer ítem recomendado, independiente de cuál item sea.
Estas métricas proveen una evaluación detallada y específica del rendimiento de los sistemas de recomendación, permitiendo una comprensión más profunda de su efectividad y precisión en la generación de recomendaciones relevantes para los usuarios.

Error cuadrático medio (MSE, Mean Squared Error)

Es una métrica que se utiliza para medir el error de predicción. Es la media de los cuadrados de las diferencias entre los valores reales y predichos.
Se define como:

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \tag{2.4}$$

Donde, y_i es el valor real para el dato i-ésimo, e $\hat{y}$ el valor predicho o estimado para el mismo ejemplo, y n el numero total de ejemplos.

2.2. Sistemas recomendadores

Los sistemas recomendadores son algoritmos diseñados en el área de recuperación de la información, que tiene como finalidad predecir o personalizar los ítems de un usuario en base a sus preferencias. Esta área de estudio es de específico interés en los sistemas que desean promover contenido a sus usuario y generar un efecto deseado, ya sea *ver la serie recomendada* o *comprar el producto recomendado.*

Existen varias técnicas y maneras de atacar distintos problemas de recomendación, dependiendo de la naturaleza de los datos a recomendar (ítems) y relación con sus usuarios. Por lo general, se identifican tres tipos de sistemas de recomendación:

Filtrado colaborativo:
Estos sistemas se basan en la idea de que las preferencias y comportamientos de un usuario pueden predecirse en función de las acciones y opiniones de otros usuarios con gustos y com-

portamientos similares. Utilizan la retroalimentación (valoraciones) y la interacción pasada de usuarios, para recomendar ítems o contenido que podría ser de interés para un usuario en función de lo que usuarios similares han valorado o consumido previamente. Por ejemplo, si una gran masa de usuarios comparte tres series en común, es probable que un usuario que haya visto solo dos de ellas, también le guste la tercera.

Filtrado basado en contenido:
Estos sistemas se centran en analizar las características o atributos de los elementos para identificar otros similares. Utilizan esta información para hacer recomendaciones personalizadas a los usuarios tomando como base las características de las preferencias pasadas. Por ejemplo, si un usuario ha mostrado interés en películas de cierto género o con actores específicos, el filtrado basado en contenido recomendará otros ítems que comparten esas características, anticipando que al usuario le gustarán por afinidad temática o de contenido.

Híbridos:
Estos sistemas combinan las ventajas de los sistemas de filtrado colaborativo y los sistemas de filtrado basado en contenido. La idea detrás de usar ambos, es mejorar la calidad de las recomendaciones mitigando las limitaciones de ambos enfoques de manera complementaria. Por ejemplo, un sistema híbrido podría recomendar películas a un usuario en función de sus preferencias personales, así como de las preferencias de otros usuarios que hayan valorado películas similares.

Mediante la revisión bibliográfica, se encontró un survey específico y reciente del área (2021), el cual permitió reafirmar ideas necesarias para abordar la comparativa de su utilización, como también fue la base para contraponer diversas soluciones y concluir en base a la realidad y restricción de los datos obtenidos para su implementación final.

El trabajo de Gharahigheni, Pliakos y Vens (2021) [3], se enfoca específicamente en la aplicación de sistemas recomendadores en el rubro inmobiliario. Estos trabajos utilizan diversas técnicas y metodologías para recomendar publicaciones de propiedades a sus usuarios. Para esto, consideran diversos tipos de información, algoritmos y formas de abordar las dificultades.
A modo general, este survey, agrupa y presenta los trabajos de acuerdo a la metodología de sistemas recomendadores utilizada, en el cual se encuentran seis enfoques diferentes (incluyendo los 3 anteriormente presentados):

- Filtrado colaborativo (Collaborative filtering).
- Filtrado basado en contenido (Content-based filtering).
- Basados en conocimiento (Knowledge-based).
- Análisis de desiciones multicriterio (Multi Criteria Decision Making).
- Aprendizaje reforzado (Reinforcement Learning).
- Modelo híbrido (Hybrid Approach)

Estos trabajos dependen directamente de los datos disponibles y del objetivo final de su aplicación, ya que no todos son aplicados en plataformas de comercio. Ante esta situación, la

solución propuesta se basa en principios asociados al filtrado colaborativo. Utilizar el historial de visitas de los usuarios para obtener una representación que capture su interés en ciertas propiedades, es una idea que aplican [4–7], con distintas metodologías o particularidades. Luego de obtener estas representaciones, utilizan modelos de cercanía (vecinos cercanos por ejemplo) para generar recomendaciones. Esos trabajos, son los que mayoritariamente se tomaron en cuenta para describir los puntos siguientes referentes a las características utilizadas y desafíos encontrados.

Para representar a las propiedades, los trabajos descritos en el survey coinciden en su gran mayoría en las características (features) que utilizan. Los autores entregan un listado que recopila los más utilizados y su respectiva aplicación, con los cuáles existe gran coincidencia y disponibilidad respecto a los datos a utilizar en este trabajo. A continuación se muestra la tabla modificada que muestra la disponibilidad para utilizar dicha característica en este trabajo.

Tabla 2.1: Features mas utilizados según el survey y su disponibilidad en goplaceit.com

Feature	Tipo	Disponibilidad
Precio	Numérico	Si
Tipo propiedad	Categórico	Si
Número de habitaciones	Numérico	Si
Número de estacionamientos	Numérico	Si
Amoblado	Binario	Si, poco confiable
Superficie útil (living area)	Numérico	Si
Superficie terreno (land area)	Numérico	Si
Ciudad	Categórico	Si
Locación (Latitud/Longitud)	Numérico	Si
Atributos del barrio	Categórico	No
Proximidad a puntos de interés	Numérico	Si, en su gran mayoría.
Comodidades	Categórico	Si, poco confiable
Prohibición de mascotas	Binario	No

Estas características están limitadas a la tabla anteriormente mencionada y solo es considerada como referencia para mostrar que gran parte de la información utilizada por los trabajos relacionados puede ser considerada en este. Adicional a estas features, existen otras que podrían ser incorporadas y que corresponden mayoritariamente a información del sector que se encuentra agregada geográficamente. Por ejemplo, a partir de los mismos datos, es posible realizar agrupaciones geográficas y calcular el precio promedio del sector y así una propiedad queda representada por información contextual, además de la propia.

2.3. Desafíos

Cold-Start Problem

Dentro de los desafíos identificados en la revisión bibliográfica, uno de los más mencionados corresponde al problema del cold-start, o el problema de inicio en frío. Este es uno de los desafíos clásicos en los sistemas recomendadores, que expone la dificultad de generar recomendaciones para un nuevo usuario o en base a un nuevo ítem, debido a que la información es escasa respecto a interacciones o a su perfil.

Este problema por lo general afecta a ambos lados de la recomendación, ya sea de cara al usuario (un nuevo usuario para el cuál no tenemos información) y de cara a un nuevo ítem (una propiedad nueva que no tiene visitas de usuarios, por lo que se hace difícil de representar). Dependiendo de la metodología que se utilice para realizar la recomendación, hay maneras para abordar estos problemas.

En particular para este trabajo, como se piensa realizar una representación vectorial de las propiedades considerando las interacciones (visitas) de los usuarios sobre estas, resaltamos la estrategia empleada por Zhang et al. [8]. Como los usuarios y las propiedades comparten el mismo espacio vectorial (primero se representan las propiedades en un espacio y luego se le representa a los usuarios dentro de ese mismo espacio), para representar a un usuario nuevo, se puede tomar una normalización de todas las propiedades del espacio. En esta normalización, existe la posibilidad de incorporar pesos a atributos que por conocimiento experto sean de mayor importancia. Además de eso, para relacionar a los usuarios y propiedades proponen utilizar la distancia coseno sobre este espacio vectorial donde conviven ambos elementos, por sobre correlación de pearson o distancia euclidiana.

Características muy específicas al dominio

Otro de los grandes desafíos de los sistemas recomendadores en el rubro inmobiliario, corresponde a los features para representar a las propiedades que están muy ligados a este dominio. Además de considerar características intrínsecas y evidentes de la propiedad (e.g., precio, habitaciones, área) que es lo principal que busca el usuario, también se encuentran otras características que algunos usuarios valoran por sobre otros. En particular, Yu et al. [4] explica que normalmente los usuarios están más interesados en propiedades con mayor proximidad geográfica, es decir, tienen en mente una zona en particular para comenzar la búsqueda y las recomendaciones que son cercanas geográficamente a esa zona capturan un alto interés.

En la misma dirección, Daly et al. [9] y Yuan et al [10] muestran que la distancia geográfica entre una propiedad en cuestión y algunos puntos de interés (POIs) para el usuario, como el lugar de trabajo, colegio u otros, tienen un rol decisor fuerte al momento de escoger una sobre otra. Estos puntos de interés varían y dependen netamente del contexto del usuario; por ejemplo, si se desea considerar la red de transporte como puntos de interés, no es lo mismo considerar la cercanía a una estación de metro, que a un paradero de micro o a una autopista, depende de lo que busca cada usuario en particular.

Escasez de datos

Otro desafío identificado en la revisión, corresponde a la escasez de datos en el contexto inmobiliario. Pero no respecto a información de usuarios o de las propiedades, sino más bien a los datos que unen a ambos.
La resolución clásica de tareas de recomendación se realizan a partir de la matriz de usuarios e ítems, donde se almacena un valor (normalmente una valoración que entrega el usuario respecto a ese ítem) cuando existe una interacción entre ambos. Esa matriz es altamente vacía (sparse), algo que es bastante común en los sistemas recomendadores y en este rubro no es la excepción a la regla, más aún cuando el proceso de búsqueda y elección de una propiedad no es una tarea frecuente y sistemática. Esta situación nos determina una perspectiva de la interacción efectiva entre usuario y propiedad considerando como más relevante el feedback implícito (como lo es realizar un click sobre una propiedad) por sobre uno explícito (guardar o contactar al corredor y/o vendedor a cargo de la propiedad).
Sobre la misma línea, Tonara y Widyawno [11] demostraron que la cantidad de interacciones por sobre la cantidad de propiedades es altamente limitada, por lo que proponen que al momento de analizar y agrupar interacciones por propiedad, se realice considerando criterios de búsqueda y categorización de las mismas, así es posible agregar las interacciones y obtener mejores números que si se consideran individualizadas a la propiedad.

Finalmente, existen otros dos desafíos a destacar, pero de menor relevancia para el desarrollo de este proyecto. Ambos corresponden a la parte más humana del problema y de cómo los usuarios toman las decisiones al momento de querer adquirir una propiedad. La primera corresponde al *complejo comportamiento de compra* [12, 13] y la segunda a *conflicto de criterios* [14]. Con respecto al comportamiento de compra, esto se determina por el activo que se va a adquirir; ya que al no ser una actividad que se realice frecuentemente, imposibilita por ejemplo, el tener a un mismo usuario con un historial de interacciones en la plataforma prolongada en el tiempo.

Por otra parte, los usuarios adquieren experiencia en la búsqueda mediante un sistema recomendador en la medida que interactúan y encuentran lo que buscan; en este caso, como regla de base, los usuarios generalmente son inexpertos cuando están en la búsqueda de una propiedad y buscan asesoría en corredores de propiedades o ejecutivos bancarios, por lo que muchas veces cuando interactúan con un sistema recomendador de este tipo, no tienen claridad de lo que realmente buscan.

El conflicto de criterios responde al efecto que se produce cuando existe una disparidad entre lo que el usuario está buscando como ideal y la oferta existente. Es normal que una persona siempre prefiera *una propiedad más amplia, pero que cueste menos* o *esta misma propiedad en un mejor vecindario*, esto lo explora [13, 14] y da pie a que los resultados que se entreguen al usuario, no respondan lo que está inicialmente buscando; pero no porque el sistema recomendador no funcione, sino más bien porque no existe oferta disponible que responda esos criterios.

Datos y evaluación

El último punto importante a la revisión bibliográfica, corresponde a los tipos de dataset utilizados y métricas de evaluación. Nuevamente, el trabajo base corresponde al survey en cuestión [3] con énfasis en los trabajos que son de nuestro interés.
Cada trabajo analizado recolectaba o generaba sus propios datos, ya que no existe una fuente pública de datos como referencia. Gran parte de los trabajos utilizan información de alguna plataforma de publicación de propiedades (como nuestro caso) y dependiendo de las limitaciones que esta tenga, generan fuentes adicionales de datos, como encuestas o recolección de meta-datos.

El punto relevante que se considera de distinta forma en los trabajos, es el feedback que realiza el usuario cuando interactúa con un ítem. Este feedback puede ser explícito, donde el usuario califica con una nota, un *me gusta* o mediante un cuestionario, explicitando la preferencia hacia una propiedad. Por el contrario, el feedback implícito, no representa necesariamente una preferencia, pero se acuerda por convención que esa acción, generalmente un click o visita a una propiedad en la plataforma, muestra una preferencia por sobre otras donde no se realizó esa misma acción.

Dada la realidad de la plataforma de goplaceit, la cantidad de propiedades que tienen un feedback explícito son considerablemente menor a lo que podría considerarse como feedback implícito, lo cuál muestra directa relación con el problema de la escasez de datos.

Por otro lado, la cantidad de usuarios y propiedades utilizadas en los trabajos relacionados son órdenes de magnitud menor de las que se utilizarán en este trabajo, incluso comparando con los trabajos que utilizan los clicks como feedback. [8] utiliza cerca de 90 mil usuarios y 50 mil propiedades, además de utilizar clicks como feedback. [15], un trabajo aplicado en una plataforma del rubro en Turquía, considera más de 300 mil propiedades y además combina un enfoque de filtrado colaborativo y basado en contenido.

Respecto a la evaluación, existe un consenso sobre las métricas utilizadas en los trabajos, que corresponden a las formas clásicas de evaluar un sistema recomendador. Primero, para evaluar que el sistema esté entregando recomendaciones que sean relevantes a la búsqueda del usuario, se utiliza mayoritariamente: precisión y recall, con sus variantes que consideran un top-K. Más aún, para evaluar que las recomendaciones relevantes se encuentren dentro de los primeros resultados, se utilizan métricas *basadas en ranking*, como: MRR (mean reciprocal rank), mAP (mean average precision), y NDCG (normalized discounted cumulative gain).

Capítulo 3

Problema

Debido a la gran cantidad de datos que posee la plataforma y la relevancia de mostrarle al usuario propiedades que sean de su interés y potencial adquisición (como comprador o arrendatario), es que goplaceit ha mostrado interés en aplicar técnicas basadas en datos para abordar este problema.

El interés de la empresa y problema propuesto a solucionar, consta de encontrar una representación del tipo embedding para las propiedades, esto con el propósito de utilizar estas representaciones principalmente para distintas tareas de recomendación a los usuarios.

Es importante que este problema se aborde desde la perspectiva de generar los embeddings primero, ya que estos puede ser utilizados de diversas maneras a posterior. Siendo un caso particular la recomendación de propiedades mediante un simple ejercicio de kNN. La dificultad está en encontrar una manera correcta de representar al usuario dentro de ese espacio vectorial.
Tal y como se mencionó en la introducción, goplaceit es una empresa motivada por los desafíos tecnológicos, por lo que desarrollar esta solución podría no solo darle la ventaja frente a otros competidores del mercado, si no que además es una oportunidad mediática para darse a conocer como una empresa que *aplica inteligencia artificial* para ofrecer un mejor servicio a sus usuarios.

Uno de los requisitos principales, es que la solución responda a una API que esté funcionando en un contenedor Docker y se comunique con las bases de datos de la plataforma. Además, es necesario que los resultados de los embeddings, queden almacenados en una base de datos relacional, utilizando el identificador único de las propiedades. Esto con el fin de realizar consultas SQL dentro del mismo entorno y extraer los embeddings de una manera directa en las consultas.

En detalle, se necesita una representación del tipo embedding para las propiedades, con motivación principal de ser utilizados para la recomendación de las mismas. Es importante que se capture una relación entre las propiedades que visitan usuarios similares, basados en las acciones similares que realizan. De esta manera, se puede realizar de manera mas directa una buena recomendación a los usuarios. Otro elemento importante, es la necesidad de representar no solo a los usuarios que han visitado hartas propiedades en la plataforma, si no también para los usuarios anónimos y los usuarios que han realizado menos de 2 o 3 visitas

a propiedades.

En un principio se planteó una necesidad particular de utilizar estos resultados en un producto que estuvo funcional un tiempo, sin embargo en este trabajo de memoria no se presenta de tal forma, ya que por una re-estructuración de la empresa, se dió de baja y no hubo como recuperar la parte de front y flujo del sistema para presentarlo de esa forma.
Los desafíos mas particulares sobre la implementación y desarrollo de la solución, han sido comentados en el capítulo anterior, gracias a la extensa revisión bibliográfica realizada.

Capítulo 4

Solución

4.1. Esquema

A continuación se presentará el esquema general del sistema general desarrollado, explicando cada uno de los componentes.

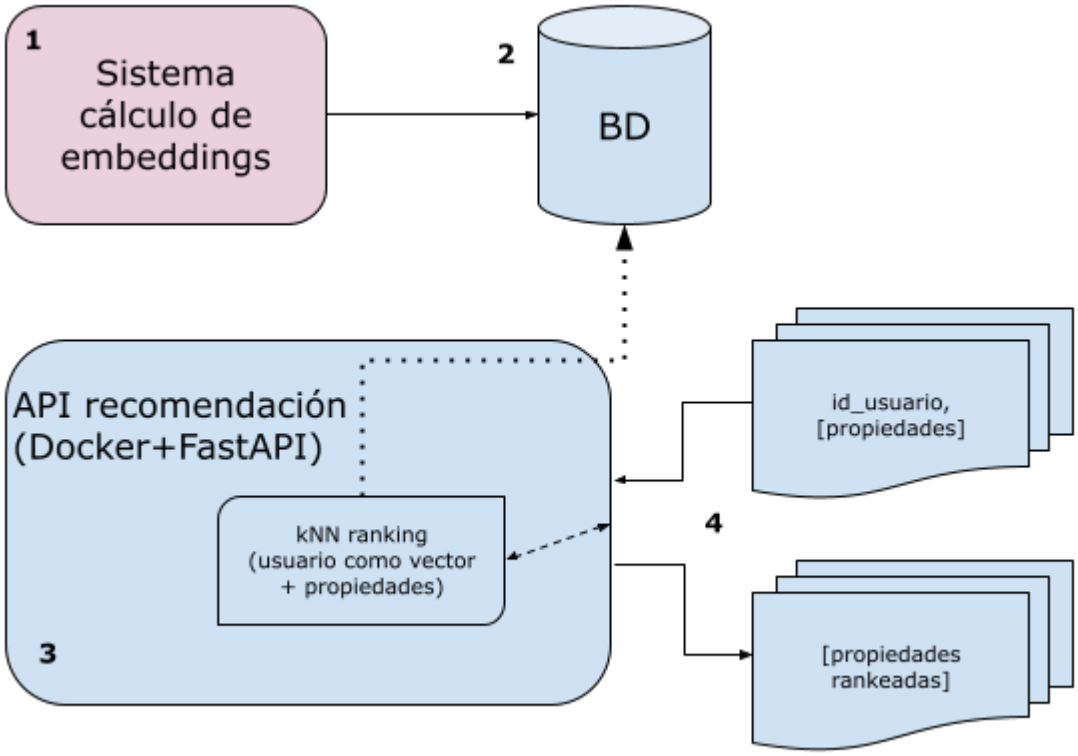

Figura 4.1: Esquema general de los componentes de la solución.

Como se ve en la figura, el sistema consta principalmente de cuatro partes.

1. **Sistema de cálculo de embeddings:** Este módulo contiene la lógica y es el motor principal del proyecto. Es el encargado de tomar los datos de la plataforma, procesarlos y mediante una red neuronal, crear un espacio vectorial para representar las propiedades.

2. **Base de datos:** Acá es donde se almacenan los embeddings anteriormente calculados, el objetivo es externalizar el guardado a una base de datos de un sistema que pueda consumirlos y utilizarlos, como sistemas de mailings, marketing digital o un sistema recomendador.

3. **Sistema recomendador:** Esta parte del sistema, es una aplicación disponibilizada mediante una API, que reside en un contenedor Docker y hace uso de la base de datos de embeddings para generar recomendaciones a usuarios según sus visitas históricas.

4. **Request y response:** Esta parte solo evidencia el interés inicial del proyecto, recibir como input a un usuario y un conjunto de propiedades y retornarlas de manera ordenada según potencial interés a ese usuario.

Esta solución desarrollada posee ventajas y desventajas, que fueron apareciendo a medida que se desarollaba el proyecto.

Dentro de las ventajas, es importante mencionar que:

- Independiza el cálculo de los embeddings del sistema recomendador u otra aplicación que haga uso de ellos. Esto permite que se realicen mejoras al algoritmo del cálculo, actualizaciones temporales o geográficas de datos u otros cambios, sin afectar a ninguna aplicación que esté utilizando los embeddings calculados.

- Debido a que, para la generación de embeddings se utilizan características reales de las propiedades (como un vector numérico de características), se puede utilizar una red ya entrenada para calcular un embedding de una propiedad nueva y que tome parte dentro del sistema recomendador de inmediato.

- La API está diseñada de tal manera que solo necesita un usuario y un conjunto de propiedades para funcionar. Eso hace que sea flexible su uso utilizando la misma metodología. Por ejemplo, un uso práctico podría ser una campaña de correo de marketing personalizada para cada usuario, utilizando un subconjunto de propiedades pertenecientes a una inmobiliaria en particular con quien se tiene acuerdo comercial.

Como desventajas, se pueden mencionar las siguientes:

- El modelo requiere de reentrenamientos cada ciertos periodos de tiempo que permitan incorporar las acciones que realizaron los usuarios en ese período de tiempo. Esto provoca un nuevo espacio de representación vectorial, invalidando el anterior. Es posible realizar un *fine-tuning* (ajustar el modelo solo con información nueva), pero esto de igual manera generaría nuevos embeddings para todas las propiedades, por lo que ahorra solo tiempo de procesamiento.

- Por el lado del sistema recomendador, la manera en cómo se soluciona ahora no es escalable para su utilización directa en el mapa de la plataforma, debido a que las propiedades candidatas pueden ser grandes cantidades y realizar un orden de estas agrega un costo adicional que impide que se pueda realizar con una latencia baja, ya que debe responder cada vez que el usuario se mueva en el mapa.

- La evaluación general de los embeddings resulta dificultosa dependiendo del uso que se les de. Se exploraron algunas formas, teniendo en mente el sistema recomendador que

se plantea como solución, sin embargo para generalizarlo a otras aplicaciones, podría necesitar otro tipo de evaluación. Por ejemplo, puede que capture muy bien los intereses de los usuarios, pero no sean de utilidad para alguien que cambió sus preferencias geográficas de búsqueda o de rango de precios drásticamente.

4.2. Descripción y procesamiento de los datos

Una vez que ya se tiene definido la necesidad y problema a tratar, se comienza por realizar una recolección y exploración de los datos con los que se trabajará. Principalmente se identifican las propiedades, los usuarios y la relación entre ambos, que en este caso sería la visita o click.

Propiedades

Las propiedades, son los ítemes o entidades principales de la plataforma, que contiene una publicación o aviso de arriendo o venta de una propiedad. Estos avisos son publicados por personas naturales (un propietario o dueño) o por corredores de propiedades o inmobiliarias (vía sincronización automática o publicación manual), quienes disponen la información de la propiedad en la publicación. Toda esta información es almacenada en bases de datos relacional, que además contiene soporte para realizar consultas espaciales (postgis).

Independientemente de si la publicación es realizada de manera manual o mediante sincronización proveniente de terceros, puede contener diversas fallas en su calidad, haciendo el proceso de limpieza y procesamiento bastante importante y fundamental.
Existen diversos tipos de propiedades que la plataforma alberga, como casa, departamento, oficina, terreno, bodega, entre otros. Sin embargo, para este trabajo sólo se considerarán las casas y departamentos; las cuáles representan el 99.2 % de lo publicado históricamente en Chile en la plataforma.
Dentro de las características principales asociada a una propiedad, se encuentran:

- Tipo de propiedad (departamento o casa)
- Modalidad (arriendo o venta)
- Precio (en UF o CLP)
- Habitaciones y baños
- Estacionamientos y bodegas
- Superficie útil y superficie total [1]
- Ubicación (latitud, longitud)
- Comuna, región, país
- Descripción escrita con más información asociada

[1] Supeficie útil hace referencia a la superficie interna en el caso de un departamento (no considera balcón) o superficie construida, en el caso de las casas. Superficie total, en las casas considera el terreno y en los departamentos la totalidad.

- Capas de información geográfica complementaria

Dado que las propiedades son la mayor fuente de información es necesario realizar tareas de preprocesamiento y limpieza exhaustiva, para mitigar diversas anomalías que puedan afectar los análisis posteriores. El conjunto de características mencionados en el listado anterior, corresponden a la información mas relevante de cada publicación, la cuál pasará a formar parte del espacio de características (feature space) de la propiedad, cuando sea representada como vector.

Lo anterior supone realizar un análisis exploratorio sobre la información de cada propiedad, para establecer filtros de limpieza, realizar transformaciones u obtener o generar características adicionales. Dentro de las características mas relevantes se encuentra el precio de la propiedad. Esta característica supone dos desafíos: primero, la utilización de una sola moneda para representarlo (en Chile se utiliza el peso chileno (CLP) o la unidad de fomento (UF)) y segundo, los valores para propiedades en arriendo y venta son órdenes de magnitud distinto.

Algo similar sucede con el tipo de propiedad, si es casa o departamento, por lo general tienen diferencias sustanciales en lo que respecta a cantidad de habitaciones, estacionamientos o superficie, en comparación a los departamentos. Por estas razones, los valores debieron ser analizados con cuidado y de manera separada para cada subconjunto de la combinación de tipo y modalidad.

La técnica más utilizada para realizar la limpieza, fue el análisis de percentiles (en particular 0.05 y 0.95) en la distribución de los datos y también una limpieza basada en el IQR (InterQuantile Range), que también se basa en el análisis de percentiles. Fuera de ello, se realizaron limpiezas que se basan en reglas de sanitización básicas del negocio, como limitar la cantidad de habitaciones o baños máximos que puede tener una vivienda normal (existen publicaciones con más de 10 habitaciones y baños y que si representan una propiedad real). También se eliminaron todas las propiedades que estuvieran etiquetadas a que pertenecían a la RM, pero geográficamente sus datos de latitud y longitud no corresponden a esta o bien, quedaban fuera de los límites de interés.

Afortunadamente la plataforma cuenta con muchas propiedades a nivel histórico, con harta diversidad en cuanto a calidad y tipo de la información de la que se dispone. Lo cuál es de vital importancia para explorar y entender los datos.

Se decidió limitar el análisis a propiedades que hayan estado activas en el 2021 y 2022. Esto quiere decir, que en alguna fecha de esos años, las propiedades estuvieron vigentes en el sitio, aunque haya sido por menos de un día. Esto es de vital importancia, ya que una propiedad solo puede ser visitada en su perfil si es que está disponible, lo cuál es el primer filtro contextual para el universo de posibilidades con que los usuarios pueden interacturar.

Como se mencionó anteriormente, gran parte de la limpieza se realizó analizando las distribuciones de las características (utilizando gráficos de boxplot, para ver los cuartiles) y eligiendo filtros adecuados para eliminar la mayor cantidad de outliers posible. Dentro de esta limpieza destaca el uso de la limpieza por IQR (InterQuartile range), un análisis basado en los cuartiles de la distribición de la característica [16].

Dentro de las características relevantes utilizadas para este trabajo, se encuentran: habitaciones, baños, estacionamientos, bodega, latitud, longitud, superficie procesada, modalidad, tipo de propiedad, comuna, precio en UF. También se utilizaron otras características propias del negocio, como el valor de la UFm2 (que corresponde al precio sobre la superficie), la cuál es un indicador bastante utilizado en el rubro e incluso se aplicaron filtros gracias a conocimiento experto del rubro.

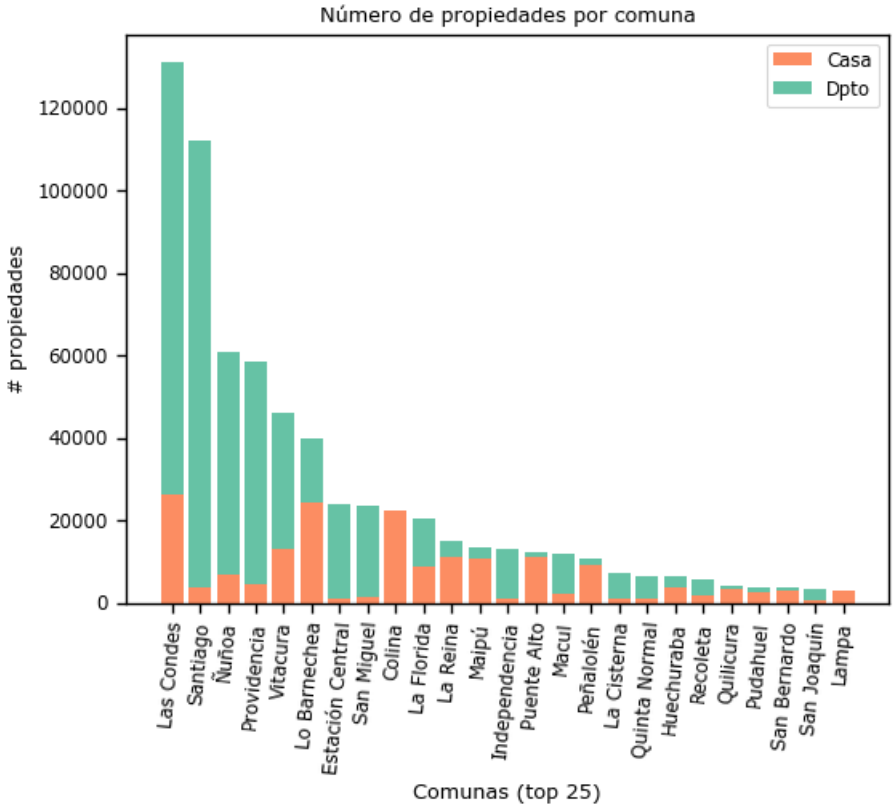

Figura 4.2: Distribución de propiedades en las 25 comunas con mayor presencia en el dataset

Un detalle sobre algunas reglas de limpieza, se encuentran en el Anexo A.

Usuarios

Los usuarios son los principales actores del sitio, quienes a su vez pueden publicar propiedades y/o buscar bajo su interés. Se decidió no utilizar información personal referente al usuario, por razones de privacidad y también por el bajo impacto que podría tener en el modelo (se contaba con datos como: nombre, correo, género, edad).

Es por esto que la caracterización de un usuario proviene principalmente de la actividad que realiza en el sitio, donde su identificador único (id_usuario) es el dato por el cuál se logra hacer un seguimiento. Históricamente la plataforma requería la creación y login del usuario para poder utilizarla, sin embargo, esto dejó de ser necesario, por lo que actualmente se cuenta también con *usuarios anónimos*. Esto plantea desafíos para el futuro, ya que el seguimiento de estos usuarios a través del tiempo sin una identificación certera dificulta su

caracterización. Se consideró el identificador único como única característica propia de los usuarios.

Un breve análisis de preferencia de las visitas realizadas por los usuarios, nos muestra que tanto la tipología (casa o departamento), como la modalidad (arriendo o venta), son características dicotómicas, es decir, los usuarios buscan un tipo de propiedad en específico, lo mismo sucede con la modalidad.

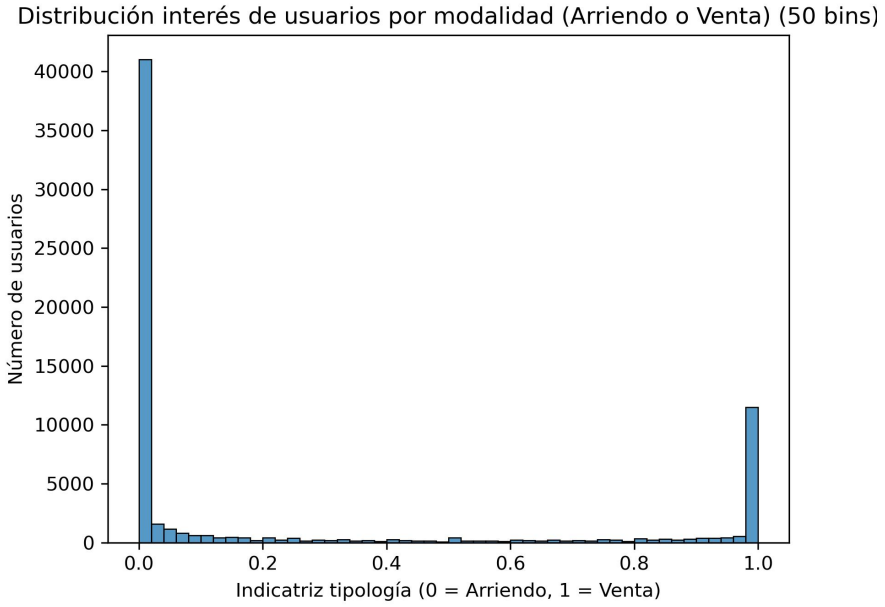

Figura 4.3: Distribución de preferencia por modalidad

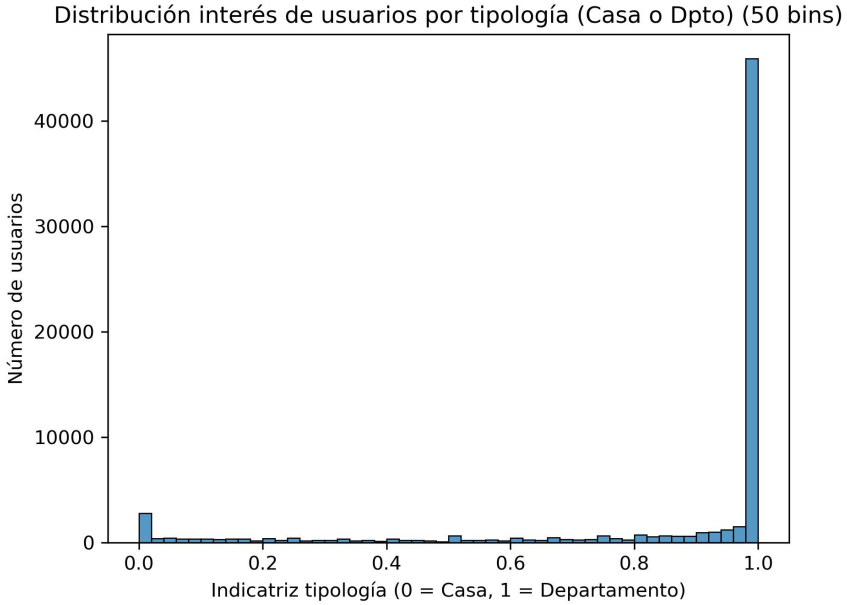

Figura 4.4: Distribución de preferencia por tipología

Visitas y sesiones

Dentro de los datos que almacena el sitio, se mantiene un registro de ciertas acciones que realiza un usuario, las cuáles pueden ir desde guardar en favoritos, enviar un mensaje, solicitar un contacto o simplemente ingresar al sitio. En particular, la acción que se consideró para este trabajo es la de *visitar una publicación de una propiedad* o *visitar el perfil de una propiedad*, independientemente de donde pueda provenir esa visita (desde distintas partes del sitio, como el mapa, landings u otros).

Por otro lado, se consideró que una visita denota un interés en la propiedad. Ya que por lo general un usuario ingresa al perfil de la propiedad para obtener información más detallada de esta misma, motivado por alguna característica que se muestra en el mapa o como resultado de un filtro o búsqueda de interés. Es natural que la propiedad pueda no ser de su interés luego de visitar su perfil, sin embargo, al igual que la mayoría de los modelos de recomendación con componente colaborativo, lo importante es que al considerar el agregado de muchos usuarios y propiedades debieran aparecer ciertos patrones (ya que todos los usuarios están sujetos al mismo efecto de *vitrinear* propiedades y que existan algunas que no sean de su interés).

Considerando el conjunto de usuarios y propiedades anteriormente filtrados (posterior al procesamiento de limpieza), se recolectaron todas las visitas realizadas por estos usuarios a propiedades de este conjunto. Para efectos de exploración, se consideraron las acciones realizadas en los años 2021 y 2022, que en su totalidad agrupan alrededor de 3.9 millones de visitas, realizadas por 95863 usuarios a 236031 propiedades.

Si bien, las propiedades se pueden considerar como una fuente confiable de datos (pues ya pasaron por un proceso de limpieza), los usuarios que realizan clicks a esas propiedades podrían mostrar un comportamiento anormal. Por esta razón es necesario explorar y eliminar potenciales usuarios que puedan inducir ruido en los datos.

Analizando los datos agrupados por usuario, obtenemos que en promedio un usuario realiza clicks a 53 propiedades, donde 4 y 44 clicks corresponden a los percentiles 25 y 75 respectivamente. Sin embargo, el usuario con mayor cantidad de clicks realizó mas de 400 mil clicks a propiedades, lo cuál corresponde a un comportamiento tipo de un proceso automatizado que realiza scraping del sitio.

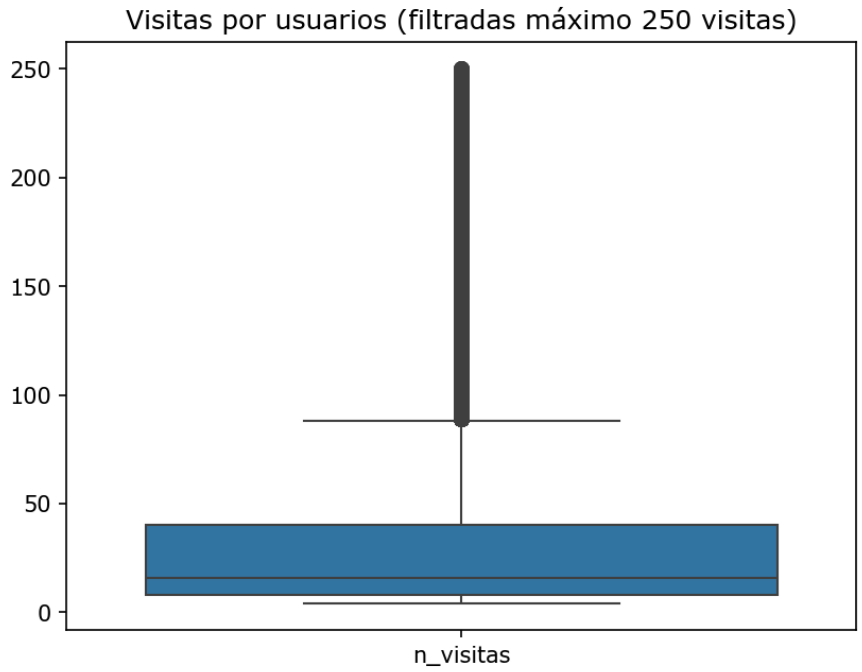

Figura 4.5: Distribucion de visitas a propiedades

Finalmente, se decide considerar a los usuarios que han realizado entre 4 e 150 visitas a propiedades. Para esta decisión se consideró por una parte que los usuarios que realizan pocas acciones (1 o 2) no nos proveen mucha información sobre sus preferencias y por otro lado, la distribución acumulada de clicks (junto con el análisis de percentil en el boxplot) indica que 150 es un buen número para filtrar usuarios que tengan un comportamiento fuera de la norma.

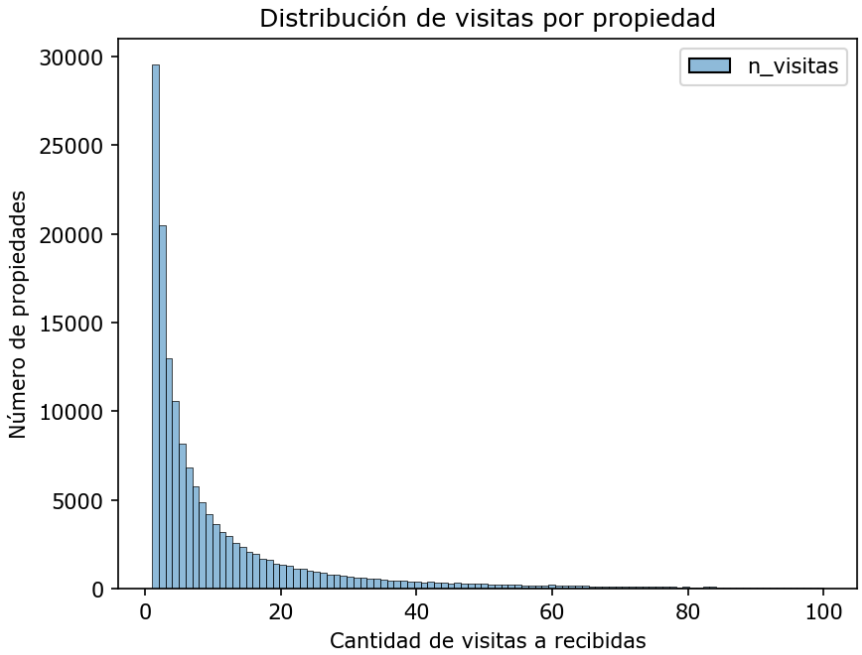

Figura 4.6: Distribución de visitas recibidas a propiedades

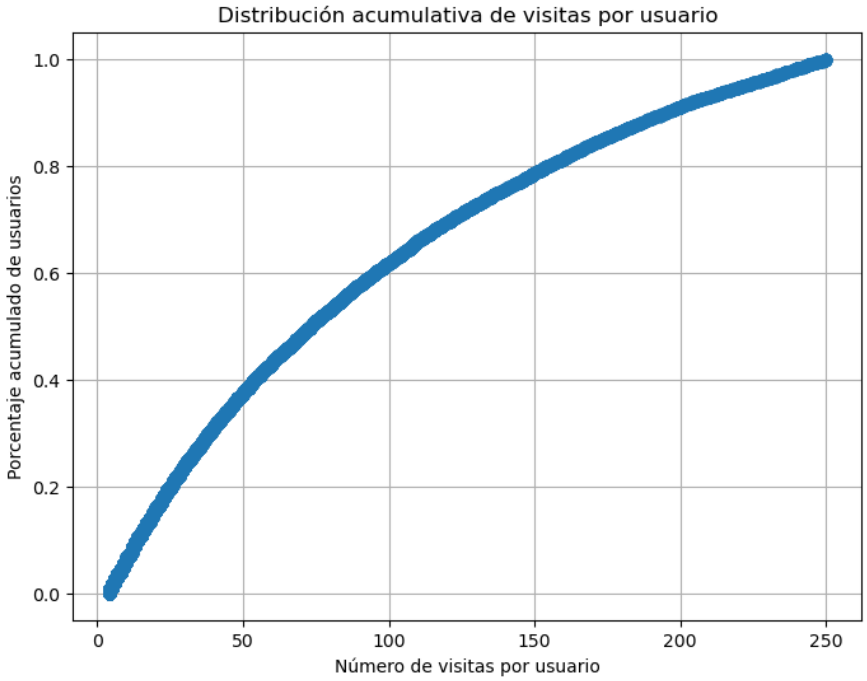

Figura 4.7: Visitas acumuladas por usuario

Considerando lo anterior, por cada usuario que realizó visitas a propiedades, se generó lo que denominamos una *sesión de visitas*. Informalmente se intenta representar una fracción de tiempo en que un usuario estuvo utilizando el sitio continuamente. Se decidió realizarlo de esta forma, asumiendo que una sesión de visitas tiene un propósito de búsqueda, que puede ser diferente en una sesión futura o pasada (los intereses de búsquedas pueden variar en el tiempo).

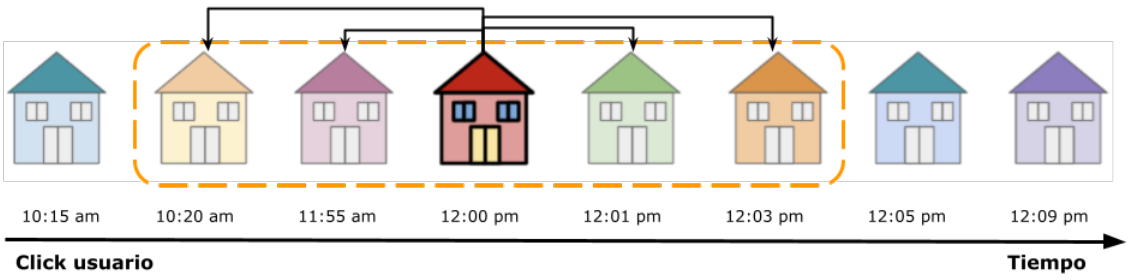

Figura 4.8: Forma gráfica de como se genera una sesión y sus pares de entrenamiento

Luego de que se generan las sesiones para cada usuario, segmentadas por períodos de tiempo de inactividad, se aplica un algoritmo de *sliding-window* con una ventana de contexto (de tamaño 5, es decir 2 para adelante y dos para atrás) y se comienzan a generar las combinaciones de pares, tomando como propiedad de entrada, la propiedad central. Esta ventana obviamente va avanzando y recorre todas las sesiones de todos los usuarios generando pares.

Esta técnica es la que da el sentido de elementos similares, ya que se asume que existen

varios pares generados por distintos usuarios, los vectores de embeddings de esas propiedades quedarán cerca.

4.3. Solución propuesta

4.3.1. Sistema de cálculo de embeddings

La idea general para el cálculo de los embeddings, es una combinación de algunos trabajos que utilizan redes neuronales para calcular una representación vectorial de ciertos elementos discretos o abstractos, guardando cierta relación entre ellos. Dentro de los trabajos fundacionales, se encuentra la representación de palabras como vectores continuos (Word2Vec [17]) que se basa en tomar una palabra como input y predecir cuales son las palabras que van antes y después.

Zillow, empresa referente del rubro inmobiliario de Estados Unidos, aplicó una idea similar utilizando sesiones de clicks de usuarios a propiedades para generar una representación vectorial de algunas características discretas, similar al Zip code [18]. Esto lo realizan mediante una red neuronal siamesa, donde se busca acercar espacialmente los vectores que representan a las propiedades que están cercanas en el tiempo de clicks (asumiendo que son similares) y luego se toman otras lejanas en el tiempo para alejar estos vectores; similar al caso de Word2Vec y la ventana de palabras dentro de una frase.
Tomando esta idea y algunos elementos de la revisión bibliográfica, decidimos realizar una red neuronal multicapa continua (feed-forward neural network) que tome como input un vector de características de una propiedad e intente aprender el vector de características de otra propiedad, minimizando una función de costo típica. Lo importante en este caso, son los valores que se almacenan en la última capa de la red neuronal (antes de la capa de vector de salida), donde los pesos de esa capa encapsulan la información aprendida por la red, de tal manera que al realizar el producto punto entre un vector de input x_i y esta capa de pesos W_e, se obtiene el embedding de la propiedad i.
Luego de algunas pruebas de concepto, se decidieron los siguientes parámetros para el entrenamiento de la red:

- 10 epochs
- Usar 10-fold cross-validation
- Dimensión del embedding de tamaño 64
- Función de activación: tanh (tangente hiperbólica)
- Dropout: 0.1 en cada capa
- Función de coste: MSE (Mean Squared Error)

Gran parte de estos hiper parámetros son recomendaciones de artículos académicos o son una buena decisión de base ya que funcionan correctamente para este tipo de problemas. Por ejemplo, la función de activación la escogimos mediante experimentos, ya que RELU llevaba a muchas componentes de los vectores a cero. Por otro lado, se agregó el componente de

Dropout para disminuir los posibles overfitting del modelo e introducir cierta variabilidad en cada epoch. (Gran parte de los parámetros fueron recomendados según el autor en este libro [19])

Para el tamaño de los embeddings en cuanto a dimensiones, hicimos un análisis comparativo utilizando la recall@K promedio en los conjuntos de validación, considerando al conjunto de usuarios del set de testing y realizando la comparativa entre las propiedades visitadas por los usuarios (propiedades relevantes) y respecto a donde las ordenaba respecto al K variable. Para cada K se calculó el recall@K para un usuario, utilizando los embeddings resultantes de dimensión 64 y 128; luego con el agregado de todos los usuarios se calcula el promedio y la desviación estándar. Esto se grafica mediante una línea para representar el promedio y una zona sombreada indicando el rango que alcanzó el promedio mas la desviación y el promedio menos la desviación.

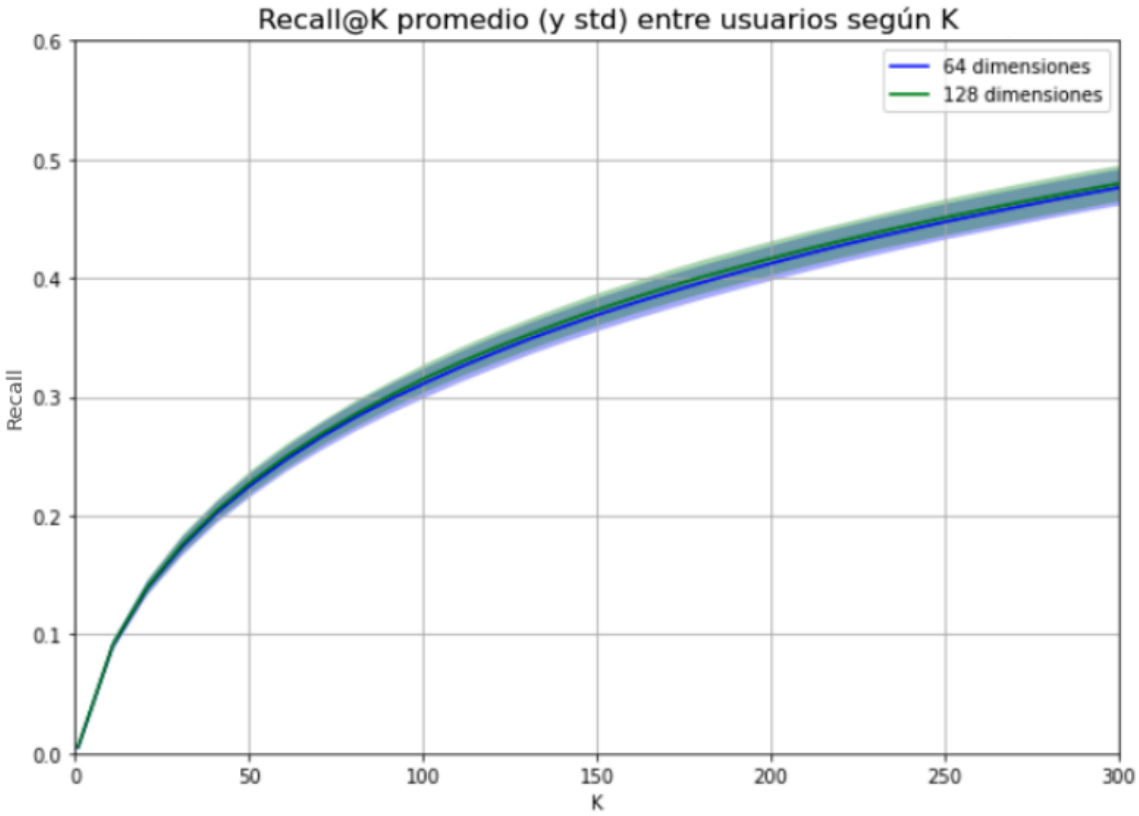

Figura 4.9: Recall@K comparativo para embeddings de dimensión 64 y 128

En la imagen 4.9 vemos que no existe una diferencia competitiva entre ambos; esto da cuenta que no se obtiene necesariamente un mejor resultado utilizando una dimensionalidad mayor (128 en este caso). Gracias a este experimento, además de la ganancia en cuanto a espacio de almacenamiento y tiempo de consulta de los embeddings, se decidió utilizar embeddings de tamaño 64. Por otra parte, también se decidió utilizar una menor dimensionalidad debido a la conocida *maldición de la dimensionalidad* y como afecta las medidas de distancia y agrupamiento de datos en altas dimensiones.

4.3.2. Base de datos

Se decidió almacenar en una base de datos relacional, como una tabla más dentro de las que utiliza la plataforma para su rápido acceso y no depender de terceras implementaciones para realizar cruces (joins) de información. El sistema anterior luego de calcular los embeddings, los guarda en la tabla (ver Figura 4.10), asociándolo a su respectivo id_propiedad, el campo embedding del tipo bytea, un id_modelo para registro de la versión del modelo usado y los campos internos por defecto que se utilizan en la plataforma (created_at y etl_sync).

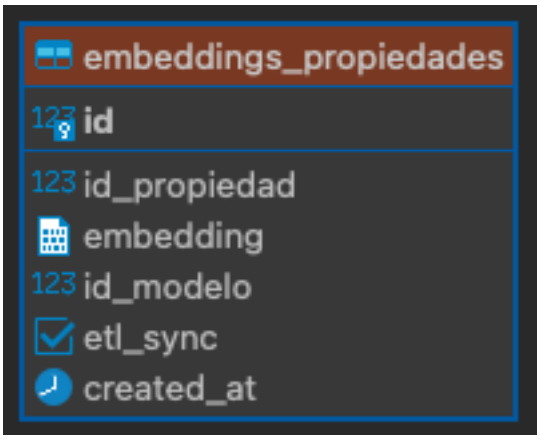

Figura 4.10: Esquema de la tabla de embeddings.

4.3.3. Sistema recomendador

Este sistema funciona dentro de un contenedor Docker, donde se habilita una API programada con el framework FastAPI de sencillo uso, con tan solo una ruta disponible, como se ve en la Figura 4.1.

La ruta es del tipo **POST: /recomendar** y recibe un json como dato asociado al request. Este tiene la siguiente estructura:

Código 4.1: Ejemplo de request

```
{
   "id_usuario": 1455037,
   "listado_propiedades": [
      1760025, 1593160, 2027436, 2023011, 2090588, 2023329, 1592809, 1764318, 1422618, 19
      ↪ 56174, 1962453, 2048100, 2002856, 2002834
   ]
}
```

Y el response en este caso sería un json con el siguiente contenido:

Código 4.2: Ejemplo de response

```
{
   "ids": [
      1962453, 2002834, 1592809, 2002856,
      2027436, 2023329, 1956174, 2048100,
      1760025, 1593160, 2023011, 1764318,
```

```
        1422618, 2090588
    ]
}
```

Los pasos lógicos de su funcionamiento son los siguientes a partir del request descrito:

1. Se recibe al usuario en cuestión y se le representa como un vector dentro del espacio de los embeddings de propiedades.
    a) Si es anónimo, se le asigna un vector fijo promedio de todos los embeddings, pues no se tiene información alguna del usuario.
    b) Si es un usuario conocido, se consulta sobre sus últimas 10 visitas a propiedades o lo máximo que haya realizado en 3 meses atrás.
    c) Se consulta en la base de datos los embeddings para estas propiedades y se calcula el promedio de ellas, asignando al usuario el vector resultante.
2. Se reciben los ids de las propiedades a ordenar, se verifica en la base de datos que tengan embedding calculado. Si no es así, entonces se asigna un embedding promedio entre todos los vectores de las propiedades (es un vector fijo calculado cuando se guarda el modelo en la base de datos).
3. Se crea un índice kNN con los embeddings (el vector que representa al usuario y los vectores de las propiedades candidatas). Se le consulta al índice que entregue los k-vectores más cercanos al usuario, con un k igual a la totalidad de las propiedades (si se desea ordenar todos los resultados).
4. Se retornan los resultados ordenados, indicando la distancia al vector consulta (el usuario).

Hay que mencionar que el uso del índice kNN se crea en el momento y no se mantiene en memoria, por el hecho de que se intenta ordenar la totalidad de las propiedades candidatas que se reciben. Si no fuese así, se generaría un índice en memoria con los vectores de todas las propiedades ya cargadas y luego se consultaría por los k más cercanos de interés. Esto ahorraría tiempo de consulta, pero no es algo requerido por la plataforma por el momento.

4.4. Validación

Si bien el modelo propuesto fue implementado y puesto en producción en el sitio, un punto importante sigue siendo el analizar el comportamiento y resultado de las soluciones propuestas; tanto para la generación de los embeddings, como su utilización en la tarea de recomendación propuesta.

4.4.1. Generación de embeddings

Para el primer caso, la generación de los embeddings mediante la red neuronal, una de las grandes validaciones es explorar como quedan los vectores de alta dimensionalidad proyectados en un espacio de 2 dimensiones (x e y), para ver gráficamente si se logró capturar y mantener alguna relación en este nuevo espacio.

Para lograr esto, el procedimiento es el siguiente: luego de obtener la representación de embedding para cada propiedad ($\mathbf{p} \in \mathbb{R}^{64}$), aplica un algoritmo de reducción de dimensionalidad, como lo es t-SNE (t-distributes Stochastic Neighbor Embedding [20]) para llevar los vectores de $\mathbb{R}^{64}$ a $\mathbb{R}^2$. t-SNE es bastante utilizado para este propósito, ya que su funcionamiento se basa en acercar los elementos que seas similares en el espacio original y mantener esas distancias entre los vectores.

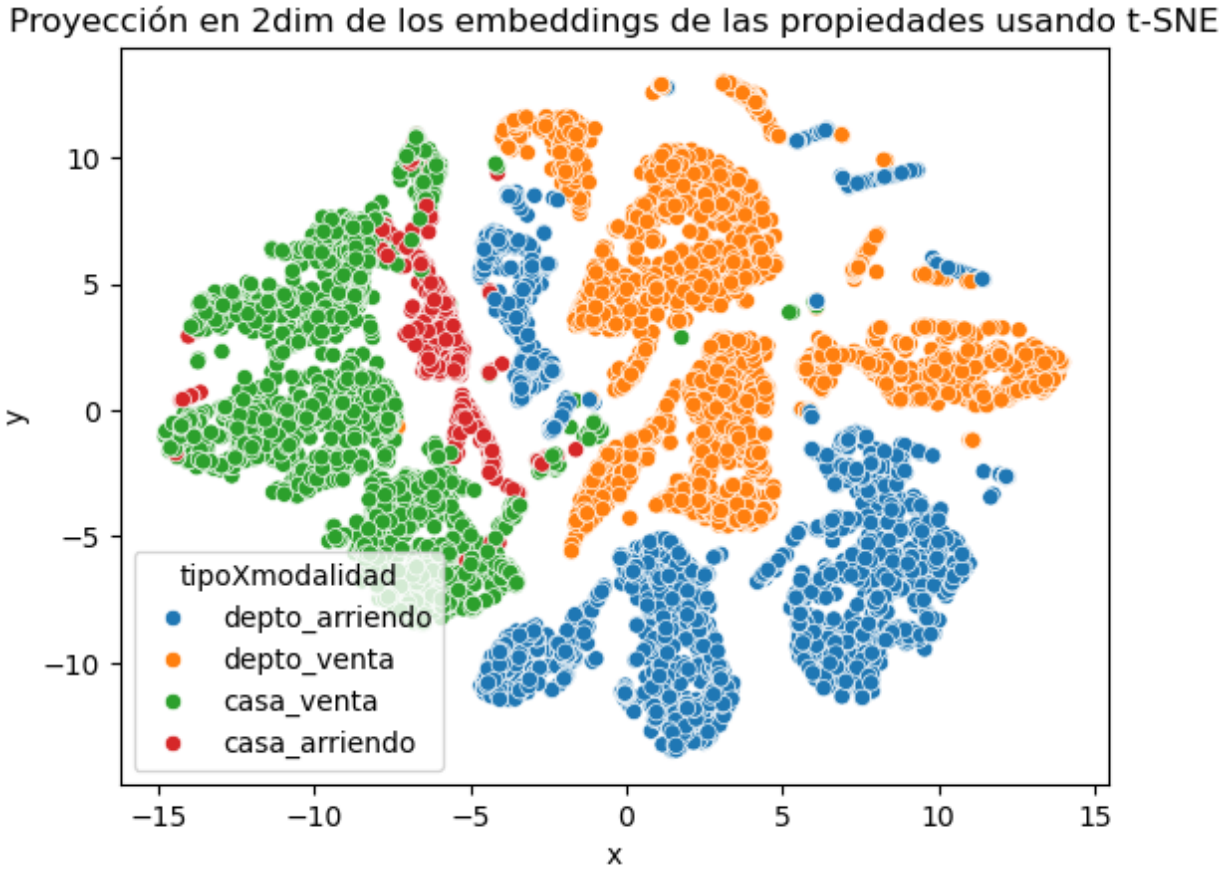

Figura 4.11: Vectores de propiedades con dimensionalidad reducida e indicando sus cruce de características de tipo y modalidad.

4.4.2. Sistema Recomendador

Para el segundo caso, una vez que se tienen los embeddings entrenados, es necesario evaluar su funcionalidad para recomendar propiedades. El contexto del sistema, descrito en la sección anterior, se decidió desarrollar un algoritmo baseline que realizara la misma tarea de recomendación y comparar el desempeño de ambos de manera comparativa.

La tarea de los algoritmos es la siguiente: dado un usuario y un conjunto de propiedades (que llamaremos propiedades candidatas), se deben ordenar según preferencia o interés posible del usuario a hacerles clicks, siendo las primeras propiedades del orden las más llamativas y luego descendiendo por posible interés.

Dado que se cuenta con información histórica para los usuarios, es posible evaluar el ordenamiento entregado por cada algoritmo, utilizando las visitas reales que realizó, evaluando en qué posiciones del ordenamiento quedaron esas propiedades. Uno de los problemas de esta evaluación, es el universo posible de propiedades candidatas que se entrega a cada algoritmo, ya que debe representar un comportamiento real del momento en que el usuario interactuó en la página. Dado que no se conoce en su totalidad las propiedades que el usuario observó sin interactuar, se debe asumir que el conjunto de propiedades candidatas en ese momento, corresponde al universo de propiedades que se encontraban activas al momento de la búsqueda.

Así, para un usuario u, que realizó una sesión de visitas a propiedades en el tiempo, se describe de la siguiente forma:

$$S_{u_i} : [(p_1, t_1), (p_2, t_2), \dots (p_f, t_f)]$$

Entonces, el conjunto de propiedades candidatas serán todas las propiedades p_i tal que su fecha de creación t_i, $t_1 \leq t_i \leq t_f$. Este conjunto es considerablemente grande (en promedio más de 20 mil propiedades), de las cuales el usuario llega a explorar no más de 50. Además considerando que al momento de la interacción con el mapa, la barra lateral del sitio está limitada para mostrar un máximo de 30 propiedades. Por estas razones, las métricas de evaluación utilizadas consideran las primeras 50 propiedades (max K=50). Ante esto, un buen resultado de la evaluación será que el algoritmo de ordenamiento posicione dentro de las primeras K propiedades aquellas a las que el usuario visitó anteriormente.

Ambos modelos funcionan de la misma forma, pero utilizan distintas maneras para representar a los usuarios y propiedades. A continuación se presenta el flujo:

Código 4.3: pseudo-codigo validacion

```
def recomendar_propiedades(
    listado_candidatas: List[int],
    id_usuario: int
):
    # Obtengo las últimas visitas del usuario
    ultimas_n_visitas_usuario: List[int] = get_last_visits(id_usuario)
    # Representan las ultimas visitas como vectores (embeddings o de otro tipo)
    # Se calcula su promedio para representar al usuario
    vector_usuario = representar_vector(ultimas_n_visitas_usuario).mean()
    # Se ordenan las propiedades candidatas respecto al vector usuario
    candidatas_ordenadas = KNN(vector_usuario, listado_candidatas)

    return candidatas_ordenadas
```

Ambas estrategias se centran en utilizar las últimas n visitas realizadas por el usuario, para transformarlo en un vector que comparte el mismo espacio vectorial que las propiedades. La diferencia radica en que el modelo baseline, utiliza un vector simple de características:

- Habitaciones (int)
- Baños (int)
- Superficie (float)
- Precio (float)
- modalidad_venta, modalidad_arriendo (one-hot)
- tipo_departamento, tipo_casa (one-hot)

Por otra parte, la estrategia de embeddings, utiliza el embedding calculado para cada propiedad como vector característico. Esto es un vector $p \in \mathbb{R}^N$, siendo N el tamaño del embedding (en este caso 64). Así, en ambos casos, el usuario es representado como el promedio de los vectores de las últimas n propiedades que visitó y luego los vectores de las propiedades candidatas se ordenan respecto al vector usuario, mediante la distancia euclidiana.

Según la metodología anteriormente explicada, se tomó una muestra aleatoria de sesiones de clicks realizadas por usuarios. Se consideró un largo fijo para estas (entre 15 y 25), y las primeras n propiedades de la sesión se utilizan para representar al usuario, dejándolas fuera del conjunto de propiedades candidatas. Las propiedades restantes de la sesión son las que se deben ordenar. La razón de lo anterior, es que se intenta replicar el comportamiento de un usuario en el sitio, como si estuviera navegando.

Una vez que se realiza este ordenamiento/recomendación de las propiedades de la sesión, se evalúa el rendimiento de cada algoritmo con las siguientes métricas: precision@k, recall@k, media de precision@k promedio (mean average precision o mAP) y se incluye además el reciprocal rank. Este valor se obtiene para cada una de las sesiones realizadas (4912), por lo que posteriormente se analizan las distribuciones y se comparan.

Además, a modo comparativo, se incluye un algoritmo de ordenamiento aleatorio con las propiedades candidatas. Si bien se espera que este algoritmo no tenga buenos resultados, sirve como referencia para los resultados de las otras estrategias.

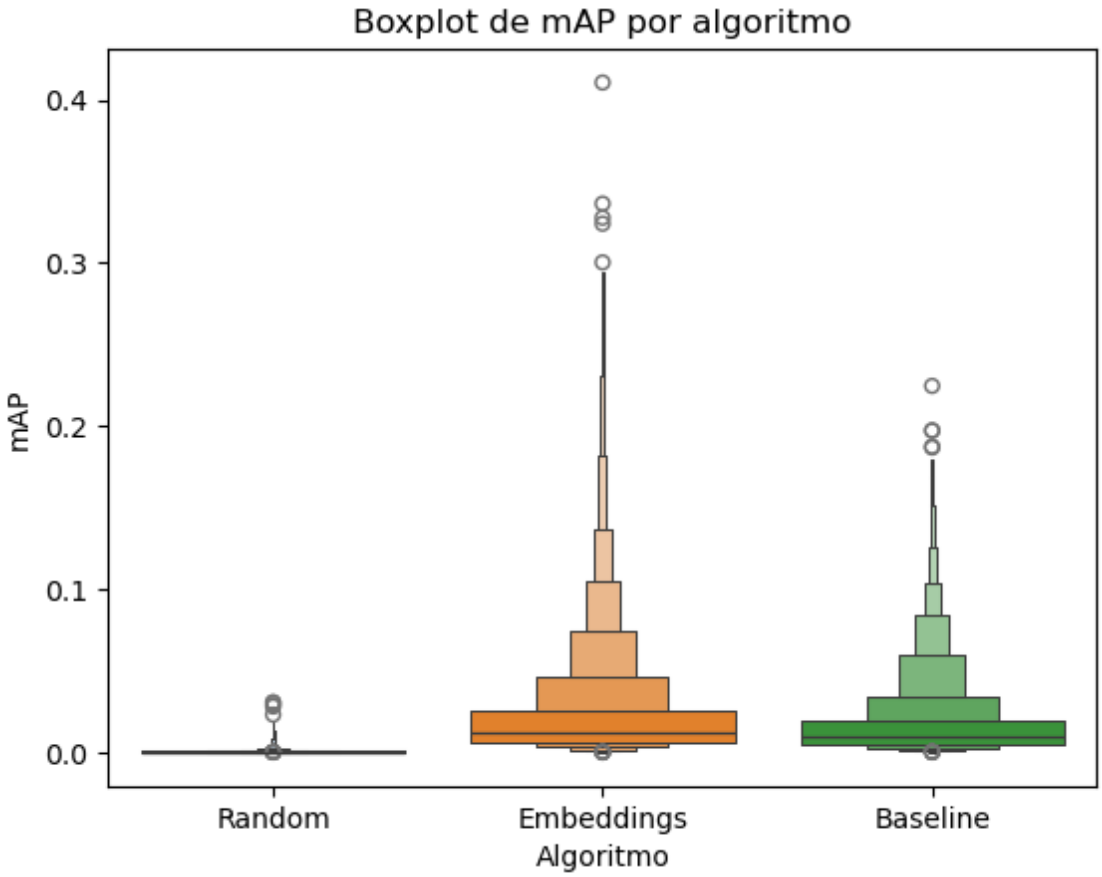

Figura 4.12: Distribución de la metrica mAP entre los distintos algoritmos

En las pruebas de validación, los resultados de la métrica mAP, para las 4912 sesiones de visitas escogidas (largo de clicks entre 15 y 25), el desempeño del algoritmo basado en embeddings, mostró una leve mejoría respecto al modelo baseline. Además, como era de esperar, el algoritmo random, no muestra ningun resultado positivo.

Para validar que esta diferencia de desempeño es significativa, se realizó un análisis estadístico de la prueba de t-student, asumiendo como hipótesis nula que ambas distribuciones tienen la misma media. Se utilizó la función ttest_ind de la librería scipy, sobre ambas distribuciones de mAP. Los resultados fueron los siguientes:

- **t-statistic: -9.227723703572929**
- **p-value: 3.335501593875762e-20**

Por lo que se rechaza la hipótesis nula y se concluye que existe una diferencia estadísticamente significativa entre ambas muestras, teniendo mejores resultados el algoritmo basado en embeddings.

Capítulo 5

Conclusiones

En este trabajo se presentaron dos aportes ante la necesidad propuesta por una empresa perteneciente al rubro inmobiliario. Como primer aporte, se presenta un sistema que genera una representación vectorial del tipo embedding para las propiedades, basada en la actividad histórica de los usuarios en el sitio web. Como segundo aporte, se le otorga utilidad a las representaciones vectoriales, con un sistema de recomendación que ordena un conjunto de propiedades para un usuario en particular, de acuerdo a su potencial interés en ellas.

Dentro del recuento de los objetivos alcanzados y contribuciones realizadas a Goplaceit y mencionadas en este trabajo se puede destacar:

- Definición de una metodología para la extracción y limpieza de datos de propiedades, usuarios y visitas (clicks).
- Desarrollo de un sistema de cálculo de embeddings de propiedades inmobiliarias mediante una red neuronal.
- Almacenamiento de los embeddings calculados en una base de datos, a disposición del uso de la empresa.
- Proceso de caracterización de usuario como embedding (basado en sus interacciones pasadas) y ordenamiento de propiedades bajo algoritmo de kNN en el espacio vectorial de los embeddings.
- Desarrollo de una API, dentro de un contenedor Docker, que responde para un usuario identificado o no, un ordenamiento para un conjunto de propiedades dadas, utilizando la búsqueda kNN mencionada.
- Funcionamiento del sistema completo en producción.

Por otra parte, el objetivo referente a la evaluación de la solución desarrollada frente a la solución del tipo baseline, se realizó utilizando datos históricos de la plataforma bajo ciertos supuestos. Lo ideal hubiese sido realizar una evaluación del sistema mediante una experimentación del tipo test A/B, pero por razones de implementación en la plataforma no se pudo realizar. Si bien la metodología utilizada para la evaluación utiliza información real generada en la plataforma, se diseñan escenarios ideales y suposiciones que pueden no representar de manera fidedigna el comportamiento real del usuario en la plataforma. Esto plantea la dificutad no solo de mejorar las recomendaciones entregadas por el ordenamiento de propiedades,

si no también de mejorar los embeddings generados. Por estas razones es que como trabajo futuro queda pendiente el desarollar estrategia de evaluación mas formal y cercana a la realidad.

Dentro de los resultados obtenidos por nuestro modelo mediante validación cruzada utilizando un set de testeo de sesiones de visitas de usuarios, se puso a prueba el algoritmo desarrollado versus otro modelo básico de tipo baseline, donde se obtuvieron resultados positivos que indican que la solución desarrollada es estadísticamente significativa.

En esa línea de mejorar la representación del tipo embeddings, existe la posibilidad de utilizar redes neuronales sobre grafos (Graph Neural Networks), lo que nos permite generar un grafo bipartito de nodos usuarios y nodos propiedades, donde una visita de un usuario a una propiedad genera un enlace entre ambos. Luego, se pueden realizar tareas de predicción de enlaces para recomendar propiedades "no vistas.ª un usuario, o también de clasificación de nodos, para aprender representaciones no tan solo de las propiedades, si no también del usuario; lo que potencialmente podría ayudar a realizar una caracterización más detallada de los usuarios del sitio.

En lo inmediato, la utilización más directa de este producto es como sistema recomendador sería la puesta en producción en la plataforma, para utilizar las recomendaciones directamente en el mapa de búsqueda. Si bien se exploró la posibilidad de hacerlo en el corto plazo, es un desafío tecnológico desde el punto de vista de tiempos de respuesta e integración en la interfaz del mapa, para mantener la experiencia del usuario. Sin esos desafíos, el sistema de recomendación diseñado podría ser utilizado sin problemas de la misma forma en que está planteado.

Es importante mencionar, que este sistema de recomendación fue utilizado en la empresa como un producto particular llamado *Asistente Inmobiliario* [2], que por motivos de reorientación de prioridades comerciales fue dado de baja. Sin embargo, la experiencia adquirida desarrollando dicho sistema recomendador fue enriquecedora, ya que se logró trabajar con datos reales en un rubro donde es particularmente difícil contar con información permanente de los usuarios. Además de trabajar con un equipo multidisciplinario y en una empresa con un alto componente científico y tecnológico, manteniendo alianzas estratégicas con CENIA (Centro Nacional de Inteligencia Artificial) y CORFO (Corporación de Fomento de la Producción).

Finalmente y a modo de sugerencia para explorar y desarrollar aún más esta solución al problema planteado, es ampliarlo a otras regiones del país como también externalizar a los demás países donde la empresa está presente. Con el fin de mejorar la experiencia de los usuarios y de paso también que a los clientes de Goplaceit, como corredores de propiedades e inmobiliarias, tengan una mayor llegada y más efectiva a la hora de publicar en el sitio.

Esto presenta desafíos no solo en la magnitud de los datos, contemplando que la plataforma cuenta con más datos en Argentina que en Chile por ejemplo, sino también a distintas dinámicas del mercado inmobiliario en otros lugares. Sin ir más allá, el trabajo remoto hizo que muchas personas ya no tuvieran la necesidad de vivir cerca del trabajo, aumentando la demanda por casas en sectores periféricos de Santiago; o el aumento de las tasas de interés

[2] Video promocional

de los créditos hipotecarios estos últimos años, hicieron que la velocidad de venta de departamentos de inversión disminuyera. Estos son ejemplos de la importancia y también dificultad que tiene ampliar este sistema no tan solo geográficamente, sino también de manera temporal.

Bibliografía

[1] Rendle, S., Freudenthaler, C., Gantner, Z., y Schmidt-Thieme, L., "Bpr: Bayesian personalized ranking from implicit feedback," 2012.

[2] Covington, P., Adams, J., y Sargin, E., "Deep neural networks for youtube recommendations," en Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, (New York, NY, USA), p. 191–198, Association for Computing Machinery, 2016, doi:10.1145/2959100.2959190.

[3] Gharahighehi, A., Pliakos, K., y Vens, C., "Recommender systems in the real estate market—a survey," Applied Sciences, vol. 11, no. 16, 2021, doi:10.3390/app11167502.

[4] Yu, Y., Wang, C., Zhang, L., Gao, R., y Wang, H., "Geographical proximity boosted recommendation algorithms for real estate," en International Conference on Web Information Systems Engineering, pp. 51–66, Springer, 2018.

[5] Jun, H. J., Kim, J. H., Rhee, D. Y., y Chang, S. W., ""seoulhouse2vec": An embedding-based collaborative filtering housing recommender system for analyzing housing preference," Sustainability, vol. 12, no. 17, p. 6964, 2020.

[6] Milkovich, K., Shirur, S., Desai, P. K., Manjunath, L., y Wu, W., "Zenden - a personalized house searching application," en 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 173–178, 2020, doi:10.1109/BigDataService49289.2020.00034.

[7] Rehman, F., Masood, H., Ul-Hasan, A., Nawaz, R., y Shafait, F., "An intelligent context aware recommender system for real-estate," en Pattern Recognition and Artificial Intelligence (Djeddi, C., Jamil, A., y Siddiqi, I., eds.), (Cham), pp. 177–191, Springer International Publishing, 2020.

[8] Zhang, Q., Zhang, D., Lu, J., Zhang, G., Qu, W., y Cohen, M., "A recommender system for cold-start items: A case study in the real estate industry," en 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1185–1192, 2019, doi:10.1109/ISKE47853.2019.9170411.

[9] Daly, E. M., Botea, A., Kishimoto, A., y Marinescu, R., "Multi-criteria journey aware housing recommender system," en Proceedings of the 8th ACM Conference on Recommender systems, pp. 325–328, 2014.

[10] Yuan, X., Lee, J.-H., Kim, S.-J., y Kim, Y.-H., "Toward a user-oriented recommendation system for real estate websites," Information Systems, vol. 38, no. 2, pp. 231–243, 2013.

[11] Tonara, D. B., Widyawono, A. A., y Ciputra, U., "Recommender system in property business a case study from surabaya, indonesia," SPECIAL ISSUE-International Journal of the Computer, the Internet and Management, vol. 23, pp. 30–31, 2013.

[12] Oh, D. y Tan, C. L., "Making better recommendations with online profiling agents," AI Magazine, vol. 26, no. 3, pp. 29–29, 2005.

[13] Burke, R. D., Hammond, K. J., y Young, B. C., "Knowledge-based navigation of complex information spaces," en Proceedings of the national conference on artificial intelligence, vol. 462, p. 468, 1996.

[14] Das, S., Ghosh, S., Mishra, B. S. P., y Mishra, M. K., "A novel recommendation system for housing search: An mcdm approach," en Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications, pp. 251–258, Springer, 2021.

[15] Tas, H., Sumnu, H. E., Gokoz, B., Aytekin, T., *et al.*, "Development of a hybrid real estate recommender system," Int. J. Technol. Eng. Stud, vol. 5, pp. 90–94, 2019.

[16] Tan, P., Steinbach, M., y Kumar, V., Introduction to Data Mining. Pearson International Edition, Pearson Addison Wesley, 2006, https://books.google.cl/books?id=_XdrQgAACAAJ.

[17] Mikolov, T., Chen, K., Corrado, G., y Dean, J., "Efficient estimation of word representations in vector space," 2013, doi:10.48550/ARXIV.1301.3781.

[18] Lin, S., 2018, https://www.zillow.com/tech/embedding-similar-home-recommendation/.

[19] Géron, A., Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Incorporated, 2019, https://books.google.cl/books?id=OCS1twEACAAJ.

[20] van der Maaten, L. y Hinton, G., "Visualizing data using t-sne," Journal of Machine Learning Research, vol. 9, no. 86, pp. 2579–2605, 2008, http://jmlr.org/papers/v9/vandermaaten08a.html.

Anexos

Anexo A. Reglas de limpieza de datos

Sobre algunas reglas definidas de limpieza y procesamientos de datos:

- Habitaciones y baños, sus valores pueden estar entre 0 y 10.
- Dimensiones (terreno y útil) no pueden superar los 5000 m2 y deben ser mayor a 25 m2.
- Para el precio, es recomendable transformar todos los valores a UF y filtrarlo utilizando la característica derivada siguiente,
- Precio por metro cuadrado, probablemente la barra de medir estándar que se utiliza en el rubro, existen valores de filtrado separados para venta y arriendo.
- Dato referente a la asignación de comuna o región, (es posible que hayan propiedades asignadas a Santiago, pero su ubicación geográfica indica que no lo están) se filtra y verifica que la latitud y longitud estén dentro de un bounding box que encierra Santiago.

Luego de haber establecido estos filtros de limpieza de información (que serán aplicados a todas las propiedades), la base inicial de propiedades corresponde a un total aproximado de 680 mil publicaciones. De las cuáles un 70 % corresponde a departamentos por sobre casas y un 60 % corresponden a arriendos por sobre ventas. La siguiente tabla A.1, muestra mas al detalle la cantidad de propiedades con las que se está trabajando.

Tabla A.1: Cantidad de propiedades por tipo y modalidad

Tipo	Modalidad	# propiedades
Casa	Arriendo	162471
	Venta	37636
Depto	Arriendo	244136
	Venta	240349

La plataforma tiene una alta concentración de propiedades en el sector oriente de Santiago, solo 4 comunas (Las Condes, Ñuñoa, Santiago y Providencia) agrupan el 50 % de las publicaciones, de un total de 52 comunas de la RM. En la siguiente Figura A.1 se muestra la cantidad por tipo de propiedad agrupado por comunas.

Una cosa importante que recalcar, es que hay comunas que tienen una oferta muy grande para uno de los dos tipos. Por ejemplo, Colina tiene casi un 100 % de casas en su oferta, contrario a Estación Central o San Miguel, comunas donde mayoritariamente se tienen

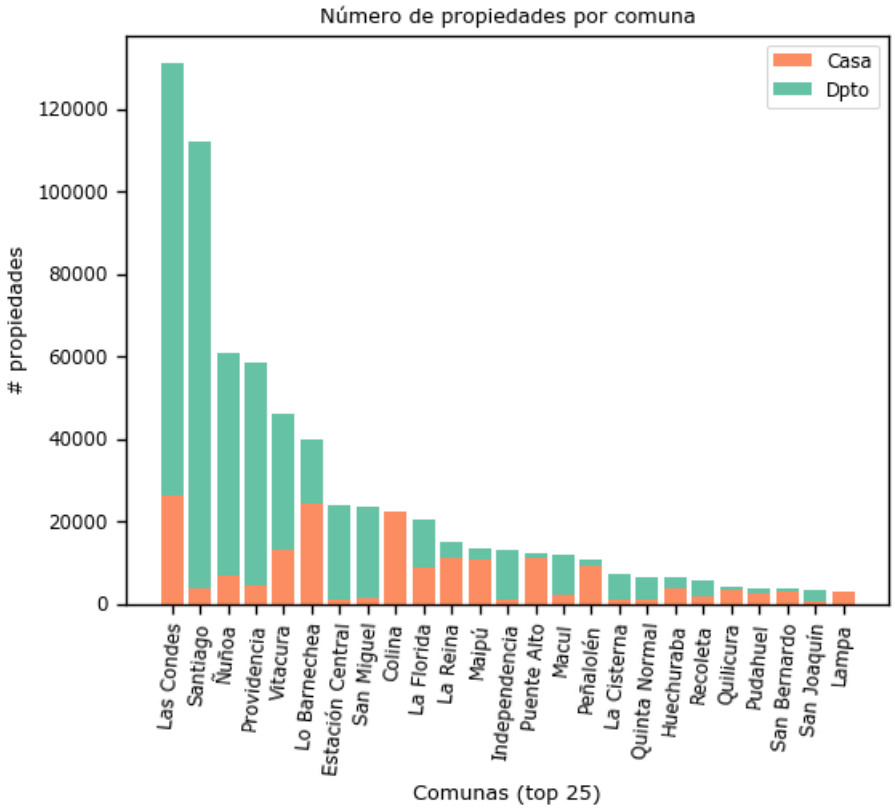

Figura A.1: Distribución de propiedades por comuna

departamentos. Este tipo de diferencias es importante descubrirlas, porque en la revisión bibliográfica un tema importante que se discutía, es que un usuario cuando busca una propiedad, dentro de los factores que están seguros, por lo general es si buscan arriendo o venta, y también casa, departamento o son indiferentes.

Anexo B. Datos utilizados para las sesiones

Notas y estadísticas sobre la generación de sesiones:

- Considera sesiones entre 2022-01-01 y 2023-01-01
- Se eliminan sesiones de menos de 4 visitas
- Se entrena un modelo de embedding usando pares de propiedades, con validación cruzada en 20 grupos de usuarios
- Se limita a los datos de entrada a aparecer cada (propiedad1, propiedad2) un máximo de 5 veces
- Por cada sesión de test se obtienen tres evaluaciones
- Considerando que el usuario vio una propiedad
- Considerando que el usuario vio dos propiedades
- Considerando que el usuario vio cuatro propiedad
- Cada usuario se caracteriza por el promedio de embeddings de propiedades vistas
- Por cada propiedad no vista, se obtiene su ranking considerando todos los embeddings
- Propiedades: 87842
- Usuarios: 9260
- Visitas: 1539597
- Visitas filtradas: 451545 out of 1539597
- Sesiones: 21156
- Visitas por usuario: 16.7
- Visitas por sesion: 7.3
- Sesiones de 4 visitas: 0.51
- Usuarios de 1 sesion: 0.56
- Largo sesion promedio para usuarios con 1 sesión: 5.97

B.1. Distribuciones de sesiones respecto a usuarios y propiedades