



**GRADO EN INGENIERÍA MATEMÁTICA E INTELIGENCIA
ARTIFICIAL**

**DESARROLLO DE ESTRATEGIAS DE IMPUTACIÓN DE
DATOS EN LA RECOMENDACIÓN DE PUNTOS DE
INTERÉS**

Autor: Antonio Mora Abos

Tutor: Pablo Sánchez Pérez

Madrid

Mayo 2025

Declaración de Autoría

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título **DESARROLLO DE ESTRATEGIAS DE IMPUTACIÓN DE DATOS EN LA RECOMENDACIÓN DE PUNTOS DE INTERÉS** en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/2025 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Firmado por
Mora Abós
Antonio -

Fdo.: Antonio Mora Abos
Fecha: 15 de junio de 2025

Firmado por
SANCHEZ PEREZ
PABLO -

Autorizada la entrega del proyecto:
EL DIRECTOR DEL PROYECTO

Fdo.: Pablo Sánchez Pérez
Fecha: 15 de junio de 2025

Agradecimientos

En primer lugar quiero dar las gracias a mi director de TFG Pablo Sánchez Pérez, ya que sin su profundo conocimiento sobre los distintos modelos de recomendación y su dedicación a este proyecto este TFG no hubiese sido posible.

Quiero expresar mi agradecimiento a todo el profesorado que me ha acompañado durante estos años de formación. En especial, gracias a David Alfaya Sánchez y a Manuel Villanueva Pesqueira, por haber sido mis guías desde el inicio de mi etapa universitaria y por su apoyo constante.

También quiero aprovechar esta ocasión para agradecer a nuestro jefe de estudios, David Contreras Bárcena, por su espíritu pionero al impulsar la creación de este grado tan innovador y por su incansable dedicación al desarrollo del mismo.

Por último, quiero dar las gracias de corazón a mis compañeros de promoción. Hemos compartido muchísimas horas de clase, trabajos, agobios antes de los exámenes... pero también muchas risas, cafés y momentos que siempre recordaré con cariño. Gracias por estar ahí en los días buenos y, sobre todo, en los no tan buenos. Me siento muy afortunado de haber vivido esta etapa con vosotros.

DESARROLLO DE ESTRATEGIAS DE IMPUTACIÓN DE DATOS EN LA RECOMENDACIÓN DE PUNTOS DE INTERÉS

Autor: Mora Abos, Antonio.

Director: Sánchez Pérez, Pablo.

Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

Resumen

Este trabajo analiza sistemas de recomendación de puntos de interés (POIs) en contextos con datos altamente dispersos utilizando un conjunto de datos de la plataforma Foursquare para Tokio, Nueva York y Londres. Se implementan y comparan diversos algoritmos, desde métodos básicos como Recomendación Aleatoria (Random), Popularidad y K-Nearest Neighbors (KNN), hasta modelos avanzados como Factorización Bayesiana por Ranking (Bayesian Personalized Ranking Matrix Factorization, BPRMF), BPR con Penalización Geográfica (BPR+GeoPenalty) y redes neuronales de grafos (GNN). Cada algoritmo es evaluado tanto en su desempeño original como tras aplicar técnicas de imputación para mitigar la dispersión de datos. Se cuantifica el impacto de estas estrategias en métricas de precisión, recall, F1, novedad, EPC y diversidad, identificando qué algoritmos se benefician más de la imputación y cuáles destacan en condiciones originales. Los resultados revelan claras ventajas del uso de imputación, así como comportamientos distintos entre algoritmos y ciudades. El código realizado para este proyecto puede encontrarse en el siguiente repositorio de GitHub https://github.com/antoniomora2003/TFG_RECOMENDATION_SYSTEMS/tree/main

1. Introducción

En la era digital, los sistemas de recomendación se han convertido en herramientas fundamentales en numerosos ámbitos, desde plataformas de *streaming* (Netflix) hasta redes sociales (Instagram) y tiendas *online* (Amazon), guiando nuestras elecciones y moldeando nuestras experiencias. Estos sistemas predicen qué elementos pueden resultar más atractivos para el usuario, basándose en su historial de interacciones. Dentro de esta categoría, este trabajo se enfoca en los sistemas de recomendación de puntos de interés (POIs), que sugieren lugares específicos como restaurantes, museos, monumentos o parques, conocidos como *venues*. En un contexto marcado por la digitalización y la movilidad global, y tras el auge del turismo post-pandemia, este tipo de recomendadores cobra una creciente relevancia al mejorar la experiencia turística y fomentar la cultura.

A diferencia de los sistemas clásicos, los recomendadores de POIs integran no solo las preferencias del usuario, sino también su ubicación y el contexto temporal. Un lugar puede ser apropiado en ciertas horas o bajo determinadas condiciones climáticas, pero no en otras. Además, deben abarcar una gran diversidad de categorías, lo que exige considerar factores del mundo real como el clima, el tráfico o eventos locales. Por tanto, una buena recomendación no solo debe ser relevante para el usuario, sino también coherente con su entorno y su momento.

Uno de los principales desafíos en este ámbito es la dispersión de datos: los usuarios interactúan con una fracción muy reducida del total de *venues* disponibles en una ciudad, lo que deja conjuntos de datos incompletos y altamente desbalanceados. Esta escasez dificulta que los algoritmos generen recomendaciones precisas y variadas. Para paliar este problema, se exploran estrategias como la imputación de datos, que permite enriquecer los datos de entrenamiento, mejorar el rendimiento y ampliar la diversidad de las recomendaciones.

El objetivo del proyecto es doble: por un lado, mejorar de forma práctica la relevancia y precisión de las recomendaciones; y por otro, contribuir académicamente mediante la comparación de enfoques bajo métricas como precisión, recall, novedad y diversidad, fomentando el desarrollo de entornos urbanos más

accesibles e inteligentes, en línea con la creación de ciudades y comunidades sostenibles, resumido por el Objetivo de Desarrollo Sostenible número 11, .

2. Definición del proyecto

El presente Trabajo de Fin de Grado tiene como objetivo principal estudiar y comparar distintas estrategias de imputación de datos aplicadas a sistemas de recomendación de puntos de interés (POIs), con el fin de mitigar el problema de la dispersión de datos que afecta a este tipo de sistemas. En contextos urbanos reales, los usuarios suelen interactuar con un número muy reducido de *venues* en relación con la oferta total de una ciudad, lo que da lugar a conjuntos de datos altamente dispersos y dificulta la generación de recomendaciones precisas y diversificadas.

Para abordar esta problemática, se han implementado distintos algoritmos de recomendación, desde técnicas tradicionales como el modelo aleatorio (Random), el basado en popularidad y los métodos K-Nearest Neighbors (KNN) por usuarios e ítems, hasta modelos más sofisticados de factorización de matrices como Bayesian Personalized Ranking Matrix Factorization (BPRMF), Bayesian Personalized Ranking con penalización geoespacial (BPR+GeoPenalty) y redes neuronales de grafos (GNN), capaces de modelar relaciones complejas mediante estructuras de grafos. Estos algoritmos han sido evaluados empleando un conjunto de datos real de Foursquare, una plataforma basada en geolocalización que registra *check-ins* en múltiples puntos de interés. Tras un proceso de filtrado y preprocesamiento, el estudio se ha centrado en tres ciudades clave: Londres, Nueva York y Tokio.

El estudio contempla tanto la evaluación de los algoritmos en su versión original como su rendimiento tras aplicar estrategias de imputación, como la incorporación artificial de interacciones recomendadas por un modelo base (BPRMF), generando así nuevas matrices de entrenamiento más densas. Las métricas empleadas para el análisis incluyen precisión, recall, F1, novedad, EPC y diversidad, lo que permite una evaluación exhaustiva del comportamiento de cada algoritmo. El trabajo no solo busca mejorar el rendimiento práctico de los modelos, sino también ofrecer una reflexión metodológica sobre el impacto de la imputación en la calidad y diversidad de las recomendaciones, con vistas a su aplicación en entornos reales.

3. Descripción del sistema

El sistema de recomendación desarrollado en este trabajo se compone de múltiples módulos interconectados que permiten llevar a cabo todo el flujo de trabajo necesario para evaluar el impacto de las estrategias de imputación. A continuación, se describen los componentes principales:

1. **Procesamiento del dataset:** Se parte del conjunto de datos de check-ins de Foursquare, enriquecido con datos adicionales sobre las ciudades y los puntos de interés. Se realiza un filtrado geográfico para centrarse exclusivamente en Londres, Nueva York y Tokio. Se eliminan categorías no relevantes (e.g., residenciales), se agrupan check-ins repetidos y se construye una columna de “score” que mide la intensidad de las interacciones por usuario y venue.
2. **División de datos:** Se aplica una estrategia de división temporal por usuario. Para cada ciudad, se reserva el 80 % de las interacciones más antiguas como conjunto de entrenamiento ($\mathcal{D}_{\text{train}}$) y el 20 % restante como conjunto de prueba ($\mathcal{D}_{\text{test}}$), simulando un entorno real de predicción de futuras visitas.
3. **Entrenamiento de algoritmos:** Se implementan varios modelos de recomendación (Random, Popularidad, KNN por usuarios e ítems, BPRMF, BPR+GeoPenalty, GNN), cada uno con su propio procedimiento de entrenamiento y cálculo de similitudes o representaciones latentes. Los modelos son ajustados mediante optimización bayesiana de hiperparámetros usando la librería **skopt**.
4. **Estrategia de imputación:** Se entrena primero un modelo de referencia (BPRMF) y se utiliza su salida para generar recomendaciones, de las cuales se selecciona un subconjunto (20 % más antiguo por usuario) para incluir en la matriz de entrenamiento como interacciones artificiales. Esto genera un nuevo conjunto $\mathcal{D}_{\text{train}}^{\text{imp}}$, que se utiliza para reentrenar todos los algoritmos, permitiendo medir el impacto de la imputación.

5. **Evaluación:** Se evalúa cada modelo con y sin imputación en el conjunto $\mathcal{D}_{\text{test}}$ mediante las métricas de Precisión, Recall, F1, Novedad, EPC y Diversidad. Estas métricas permiten analizar tanto la relevancia como la diversidad de las recomendaciones, capturando los efectos de las distintas estrategias sobre el sistema.
6. **Visualización y análisis de resultados:** Los resultados se presentan mediante tablas y gráficos comparativos, que muestran el impacto porcentual de la imputación por ciudad y por métrica. Esta visualización facilita la comparación entre algoritmos y escenarios.

4. Resultados

El análisis de los resultados experimentales (tablas Tabla 2 a Tabla 13 y figuras Figura 1, Figura 2) muestra que el parámetro *cutoff* impacta las métricas de recomendación. Aumentarlo de 5 a 10 mejora novedad y diversidad, pero reduce precisión y recall. Por ejemplo, en Londres, la GNN pasa de una precisión de 0,0187 a 0,0150 y un recall de 0,0584 a 0,0944, con novedad subiendo de 4,7209 a 4,9928 (Tabla 4). En Tokio, el algoritmo random alcanza una novedad de 12,1183 con $k = 10$, pero su precisión cae a 0,00007 (Tabla 12), evidenciando un trade-off entre relevancia y métricas de novedad. En relevancia, los modelos de popularidad y factorización de matrices (BPRMF, BPRMF+GeoPenalty) destacan: en Londres ($k = 5$), popularidad logra una precisión de 0,0203 y recall de 0,0649 (Tabla 2), mientras que en Nueva York, BPRMF+GeoPenalty obtiene 0,0330 y 0,1110 (Tabla 6). Sin embargo, el algoritmo random lidera en novedad y diversidad, alcanzando 12,1183 y 29.525 ítems en Tokio ($k = 10$) (Tabla 12), gracias a su naturaleza aleatoria. La GNN ofrece un desempeño intermedio, con recall de 0,0944 en Londres ($k = 10$) y precisión de 0,0327 en Nueva York ($k = 5$) tras imputación (Tabla 6), sin sobresalir en ninguna métrica. La imputación de datos afecta de forma variada: en KNN, mejora la relevancia (precisión sube 504,55 % en Londres, $k = 5$) pero reduce diversidad (Tabla 3); los modelos de factorización de matrices son más robustos, con cambios $< 3\%$; la GNN gana moderadamente (precisión $+21,95\%$ en Londres, $k = 10$); y random y popularidad apenas varían.

5. Conclusiones

El análisis de los resultados experimentales permite distinguir tres perfiles claros de algoritmos de recomendación. Los modelos basados en popularidad y factorización de matrices (BPRMF y BPRMF+GeoPenalty) sobresalen en métricas de relevancia inmediata como precisión, recall y F1, siendo especialmente eficaces en entornos como Londres y Nueva York. Por su parte, el algoritmo random muestra un comportamiento opuesto: aunque ofrece resultados muy bajos en relevancia, maximiza la diversidad y la novedad de las recomendaciones, convirtiéndose en una herramienta útil para fomentar la exploración del catálogo. En un punto intermedio se encuentra la GNN, que logra un equilibrio razonable entre relevancia y métricas de novedad, destacando en recall y manteniendo valores sólidos en novedad, aunque sin liderar ninguna métrica de forma contundente.

La imputación de interacciones tiene un impacto desigual según el tipo de algoritmo: mejora considerablemente la relevancia en modelos como KNN, a costa de una pérdida significativa de diversidad, los efectos de la imputación son más moderados y beneficiosos en la GNN, donde incrementa la precisión sin comprometer excesivamente otras métricas. En cambio, para algoritmos como el modelo de popularidad y el random, la imputación apenas aporta valor, ya que sus mecanismos no se ven afectados por la estructura de los datos faltantes.

En definitiva, la elección del algoritmo más adecuado depende del objetivo principal del sistema: si se prioriza la relevancia, los modelos de popularidad o factorización de matrices son los más adecuados; si se busca diversidad y novedad, el random es la opción preferente; y si se desea un compromiso entre ambos extremos, la GNN se presenta como una solución versátil. Una posible línea futura podría combinar modelos con buenas métricas de novedad con modelos con buenas métricas en efectividad, como random y BPRMF, para lograr un sistema más balanceado y eficaz.

Referencias

Se encuentran al final de la memoria.

Summary

This work analyzes point of interest (POI) recommendation systems in contexts with highly sparse data, using a dataset from the Foursquare platform for Tokyo, New York, and London. Various algorithms are implemented and compared, ranging from basic methods such as Random Recommendation, Popularity, and K-Nearest Neighbors (KNN), to advanced models like Bayesian Personalized Ranking Matrix Factorization (BPRMF), BPR with Geographic Penalty (BPR+GeoPenalty), and Graph Neural Networks (GNN). Each algorithm is evaluated both in its original performance and after applying imputation techniques to mitigate data sparsity. The impact of these strategies is quantified across metrics such as precision, recall, F1, novelty, EPC, and diversity, identifying which algorithms benefit most from imputation and which perform best in their original conditions. The results reveal clear advantages of using imputation, as well as distinct behaviors across algorithms and cities. The code developed for this project can be found in the following GitHub repository https://github.com/antoniomora2003/TFG_RECOMENDATION_SYSTEMS/tree/main

1. Introduction

In the digital era, recommendation systems have become essential tools in numerous domains, from streaming platforms (Netflix) to social networks (Instagram) and online stores (Amazon), guiding our choices and shaping our experiences. These systems predict which items may be most appealing to users based on their interaction history. Within this category, this work focuses on point of interest (POI) recommendation systems, which suggest specific locations such as restaurants, museums, monuments, or parks, known as *venues*. In a context marked by digitalization and global mobility, and following the post-pandemic tourism boom, this type of recommender is gaining increasing relevance by enhancing the tourist experience and promoting cultural engagement.

Unlike classical systems, POI recommenders integrate not only user preferences but also their location and temporal context. A place may be suitable at certain times or under specific weather conditions but not others. Additionally, they must cover a wide range of categories, requiring consideration of real-world factors such as weather, traffic, or local events. Therefore, a good recommendation must not only be relevant to the user but also consistent with their environment and timing.

One of the main challenges in this field is data sparsity: users interact with a very small fraction of the total available *venues* in a city, resulting in incomplete and highly imbalanced datasets. This scarcity hinders algorithms from generating accurate and diverse recommendations. To address this issue, strategies such as data imputation are explored, which enrich training data, improve performance, and expand the diversity of recommendations.

The project has a dual objective: on one hand, to practically improve the relevance and precision of recommendations; and on the other, to contribute academically by comparing approaches under metrics such as precision, recall, novelty, and diversity, fostering the development of more accessible and intelligent urban environments, in line with Sustainable Development Goal number 11, which focuses on creating sustainable cities and communities.

2. Project Definition

The primary objective of this Bachelor's Thesis is to study and compare different data imputation strategies applied to point of interest (POI) recommendation systems to mitigate the data sparsity problem affecting such systems. In real urban contexts, users typically interact with a very small number of *venues* relative to a city's total offerings, leading to highly sparse datasets that complicate the generation of accurate and diversified recommendations.

To tackle this issue, various recommendation algorithms have been implemented, from traditional techniques such as the random model (Random), popularity-based, and K-Nearest Neighbors (KNN) for users and items, to more sophisticated matrix factorization models like Bayesian Personalized Ranking Matrix Factorization (BPRMF), Bayesian Personalized Ranking with geospatial penalty (BPR+GeoPenalty), and Graph Neural Networks (GNN), capable of modeling complex relationships through graph structures. These algorithms were evaluated using a real dataset from Foursquare, a geolocation-based platform

that records *check-ins* at multiple points of interest. After a filtering and preprocessing process, the study focused on three key cities: London, New York, and Tokyo.

The study considers both the evaluation of algorithms in their original form and their performance after applying imputation strategies, such as artificially incorporating interactions recommended by a baseline model (BPRMF), thus generating denser training matrices. The metrics used for analysis include precision, recall, F1, novelty, EPC, and diversity, enabling a comprehensive evaluation of each algorithm's behavior. The work not only aims to improve the practical performance of the models but also to provide a methodological reflection on the impact of imputation on the quality and diversity of recommendations, with a view to their application in real-world settings.

3. System Description

The recommendation system developed in this work consists of multiple interconnected modules that enable the complete workflow required to evaluate the impact of imputation strategies. The main components are described below:

1. **Dataset Processing:** The starting point is the Foursquare check-in dataset, enriched with additional data about cities and points of interest. A geographic filter is applied to focus exclusively on London, New York, and Tokyo. Non-relevant categories (e.g., residential) are removed, repeated check-ins are aggregated, and a “score” column is constructed to measure the intensity of interactions per user and venue.
2. **Data Splitting:** A temporal splitting strategy is applied per user. For each city, the oldest 80 % of interactions are reserved as the training set ($\mathcal{D}_{\text{train}}$), and the remaining 20 % as the test set ($\mathcal{D}_{\text{test}}$), simulating a real-world environment for predicting future visits.
3. **Algorithm Training:** Several recommendation models (Random, Popularity, KNN for users and items, BPRMF, BPR+GeoPenalty, GNN) are implemented, each with its own training procedure and calculation of similarities or latent representations. The models are tuned using Bayesian hyperparameter optimization with the `skopt` library.
4. **Imputation Strategy:** A baseline model (BPRMF) is first trained and used to generate recommendations, from which a subset (the oldest 20 % per user) is selected to be included in the training matrix as artificial interactions. This creates a new set $\mathcal{D}_{\text{train}}^{\text{imp}}$, which is used to retrain all algorithms, allowing the measurement of imputation's impact.
5. **Evaluation:** Each model, with and without imputation, is evaluated on the $\mathcal{D}_{\text{test}}$ set using metrics such as Precision, Recall, F1, Novelty, EPC, and Diversity. These metrics enable analysis of both the relevance and diversity of recommendations, capturing the effects of different strategies on the system.
6. **Visualization and Result Analysis:** Results are presented through comparative tables and graphs, showing the percentage impact of imputation by city and metric. This visualization facilitates comparison across algorithms and scenarios.

4. Results

The analysis of the experimental results (tables Tabla 2 to Tabla 13 and figures Figura 1, Figura 2) shows that the *cutoff* parameter impacts recommendation metrics. Increasing it from 5 to 10 improves novelty and diversity but reduces precision and recall. For example, in London, the GNN moves from a precision of 0.0187 to 0.0150 and a recall of 0.0584 to 0.0944, with novelty rising from 4.7209 to 4.9928 (Tabla 4). In Tokyo, the random algorithm achieves a novelty of 12.1183 with $k = 10$, but its precision drops to 0.00007 (Tabla 12), highlighting a trade-off between relevance and novelty metrics. In terms of relevance, popularity and matrix factorization models (BPRMF, BPRMF+GeoPenalty) stand out: in London ($k = 5$), popularity achieves a precision of 0.0203 and recall of 0.0649 (Tabla 2), while in New York, BPRMF+GeoPenalty obtains 0.0330 and 0.1110 (Tabla 6). However, the random algorithm leads in novelty and diversity, reaching 12.1183 and 29.525 items in Tokyo ($k = 10$) (Tabla 12), due to its random nature. The GNN offers an intermediate performance, with a recall of 0.0944 in London ($k = 10$) and precision of 0.0327 in New York ($k = 5$) after imputation (Tabla 6), without excelling in

any metric. Data imputation has varied effects: in KNN, it improves relevance (precision increases by 504.55 % in London, $k = 5$) but reduces diversity (Tabla 3); matrix factorization models are more robust, with changes $< 3\%$; the GNN sees moderate gains (precision +21,95 % in London, $k = 10$); and random and popularity models show minimal variation.

5. Conclusions

The analysis of the experimental results distinguishes three clear profiles of recommendation algorithms. Popularity and matrix factorization models (BPRMF and BPRMF+GeoPenalty) excel in immediate relevance metrics such as precision, recall, and F1, being particularly effective in environments like London and New York. In contrast, the random algorithm exhibits opposite behavior: while it delivers very low relevance, it maximizes the diversity and novelty of recommendations, making it a useful tool for encouraging catalog exploration. The GNN occupies an intermediate position, achieving a reasonable balance between relevance and novelty metrics, standing out in recall and maintaining solid novelty values, though not leading decisively in any metric.

Imputing interactions has an uneven impact depending on the algorithm type: it significantly improves relevance in models like KNN, at the cost of a substantial loss in diversity; the effects of imputation are more moderate and beneficial in the GNN, where it boosts precision without overly compromising other metrics. In contrast, for algorithms like the popularity and random models, imputation adds little value, as their mechanisms are unaffected by the structure of missing data.

Ultimately, the choice of the most suitable algorithm depends on the system's primary goal: if relevance is prioritized, popularity or matrix factorization models are the most appropriate; if diversity and novelty are sought, the random model is the preferred option; and if a balance between both extremes is desired, the GNN presents a versatile solution. A potential future direction could involve combining exploratory and exploitative models—such as random and BPRMF—to achieve a more balanced and effective system.

References

Included at the end of the document.

Índice

1. Introducción	13
2. Revisión del Estado del Arte	14
2.1. Notación	14
2.2. Algoritmos de Recomendación	15
2.2.1. Algoritmo de Recomendación Aleatorio (Random)	15
2.2.2. Popularidad	15
2.2.3. K-Nearest Neighbors (KNN)	15
2.2.4. Bayesian Personalized Ranking Matrix Factorization (BPRMF)	16
2.2.5. BPR+GeoPenalty	16
2.2.6. Redes Neuronales de Grafos (GNN) con Embeddings BERT	17
2.2.7. Self-Attentive Sequential Recommendation (SASRec)	17
2.3. Evaluación de Sistemas de Recomendación	18
2.3.1. Métricas de Evaluación	18
2.4. Técnicas para Afrontar el Problema de la Dispersión	19
2.4.1. Imputación de Datos	20
2.5. Foursquare como fuente de datos	20
3. Metodología	21
3.1. Tecnologías Utilizadas	21
3.2. Herramientas	22
3.3. Objetivos	22
3.4. Conjunto de Datos	22
3.4.1. Preprocesamiento de Datos	22
3.4.2. Estadísticas del Conjunto de Datos por Ciudad	23
4. Resultados	24
4.1. Resultados de Londres	24
4.1.1. Resultados con cutoff de 5	24
4.1.2. Resultados con cutoff de 10	25
4.2. Resultados de Nueva York	27
4.2.1. Resultados con cutoff de 5	27
4.2.2. Resultados con cutoff de 10	28
4.3. Resultados de Tokio	29
4.3.1. Resultados con cutoff de 5	29
4.3.2. Resultados con cutoff de 10	30
5. Conclusiones	30
6. Trabajo Futuro	31
7. Referencias Bibliográficas	31

Índice de figuras

1.	Comparativa del número de usuarios, venues y check-ins en Londres, Nueva York y Tokio.	23
2.	Dispersión porcentual de check-ins en cada ciudad	24

Índice de Tablas

1.	Dispersión de check-ins por ciudad	24
2.	Resultados Londres con cutoff de 5	24
3.	Porcentaje de mejora al aplicar imputación para Londres con cutoff = 5	25
4.	Resultados Londres con cutoff de 10	25
5.	Porcentaje de mejora al aplicar imputación para Londres con cutoff = 10	26
6.	Resultados para Nueva York con cutoff de 5	27
7.	Porcentaje de mejora al aplicar imputación en Nueva York con cutoff = 5	27
8.	Resultados para Nueva York con cutoff = 10	28
9.	Porcentaje de mejora al aplicar imputación para Nueva York con cutoff = 10	28
10.	Resultados para Tokio con cutoff = 5	29
11.	Porcentaje de mejora al aplicar imputación para Tokyo con cutoff = 5	29
12.	Resultados para Tokio con cutoff = 10	30
13.	Porcentaje de mejora al aplicar imputación para Tokyo con cutoff = 10	30

Memoria

1. Introducción

En la era digital actual, los sistemas de recomendación se han convertido en herramientas clave en numerosos ámbitos, desde plataformas de streaming hasta redes sociales y tiendas online, guiando de manera sutil nuestras elecciones y moldeando nuestras experiencias. Estos sistemas buscan predecir artículos que puedan resultar interesantes a los usuarios en base al histórico de interacciones entre el usuario y su consumo, proporcionando recomendaciones personalizadas que mejoran de forma significativa la experiencia del usuario. Dentro de esta amplia categoría, este trabajo se centrará en los sistemas de recomendación de puntos de interés (POIs), consisten en recomendar a los usuarios lugares específicos tales como restaurantes, museos o parques, estos distintos lugares reciben el nombre de venues. La digitalización y la movilidad global son pilares fundamentales de la interacción humana, por ello los distintos sistemas de recomendación de puntos de interés desempeñan un papel crucial. Sumado al aumento exponencial del turismo desde la pandemia de COVID-19, esto hace que estos sistemas de recomendación sean cada vez más demandados por los distintos usuarios, mejorando la experiencia turística a la par que fomentando la cultura.

A diferencia de los sistemas de recomendación tradicionales, los sistemas de recomendación de POIs no se basan únicamente en las interacciones y preferencias pasadas del usuario. Estos sistemas incorporan un factor clave a la hora de recomendar: la influencia geográfica, ya que estos recomendadores emplean datos geoespaciales para priorizar venues cercanos a otros ya visitados por el usuario, un factor que no aplica a la hora de recomendar un libro o una película. La utilidad de un POI puede variar dependiendo de la hora del día o incluso del mes del año en que estemos: un parque puede ser perfecto para visitar por la mañana, pero mucho menos atractivo a medianoche. Además, los sistemas de recomendación de POIs deben hacer frente a una amplia gama de categorías (museos, restaurantes, tiendas, parques...), cada una con características únicas y muy distintas del resto, un aspecto que carece de importancia en sistemas que recomiendan ítems más uniformes. El contexto del mundo real juega también un papel crítico a la hora de recomendar, no solo en sistemas de recomendación de POIs sino en general en el ámbito de la recomendación. Los sistemas de recomendación deben a menudo incorporar información en tiempo real, como el clima, eventos locales o el tráfico, para hacer recomendaciones consistentes con el mundo real. En definitiva, una buena recomendación no solo debe alinearse con nuestros gustos, sino que debe tener en cuenta diversos factores relacionados con el entorno físico y el contexto del usuario.

Los sistemas de recomendación deben enfrentarse a un gran dilema que abordaremos en este trabajo: la dispersión de los datos, que se define como la escasez de datos, problema que se acentúa en los sistemas de recomendación de POIs. Este fenómeno sucede debido a que los usuarios suelen realizar check-ins en ubicaciones específicas de manera esporádica, por ello los conjuntos de datos suelen estar incompletos y desbalanceados, dificultando la capacidad de los algoritmos para generar recomendaciones precisas y generalizables. Para solucionar esta limitación, se proponen diversas técnicas como la generación de datos sintéticos mediante técnicas de imputación o redes neuronales generativas (GANs). Presentándose como una solución innovadora a este problema, permitiendo la creación de interacciones artificiales que enriquecen los datos de entrenamiento reduciendo dicha dispersión y mejoran la calidad de las recomendaciones. Abordar esta limitación no solo beneficiará la precisión y recall del sistema, sino que también permitirá obtener un mejor rendimiento en métricas de diversidad y novedad, ampliando la gama de recomendaciones disponibles para los usuarios.

Para la realización de este proyecto, vamos a trabajar con datos de la plataforma Foursquare, una red social basada en localización (LBSN). Este tipo de plataformas generan grandes volúmenes de datos geoespaciales al aprovechar la geolocalización de los usuarios, permitiéndonos emplear datos de manera gratuita. Otras LBSNs muy conocidas son Gowalla y Yelp. Para este trabajo, vamos a emplear un dataset de check-ins de Foursquare, dada la inabarcabilidad de estos datos, hemos optado por realizar un preprocesamiento de los datos para trabajar únicamente con los check-ins de tres ciudades: Tokio, Nueva York y Londres, como se describe en la sección 3.4. Utilizaremos diversas técnicas de recomendación, desde enfoques más tradicionales como KNN y modelos basados en popularidad hasta métodos más sofisticados como BPRMF (Bayesian Personalized Ranking Matrix Factorization), un algoritmo avanzado de factorización matricial que optimiza el ranking de POIs para cada usuario, ordenando los ítems de

manera que los venues visitados tengan mayor probabilidad de aparecer por encima de los no visitados. Este método es ideal para datos como los check-ins, ya que no requiere información explícita y puede manejar grandes volúmenes de interacciones. También implementaremos BPR+GeoPenalty, una extensión de BPRMF que incorpora información geográfica, ajustando las recomendaciones según las preferencias del usuario y la información geográfica de los venues visitados. El algoritmo más avanzado que implementaremos será una red neuronal basada en grafos (GNN), que permite modelar relaciones complejas entre usuarios y POIs mediante estructuras de grafos, donde los nodos son usuarios y venues, y las aristas representan los scores. Las GNNs capturan patrones no lineales y dependencias entre nodos, siendo ideales para mejorar los sistemas de recomendación al integrar datos contextuales. Para implementar este modelo, emplearemos la librería Torch-Geometric, que facilita la creación y entrenamiento de GNNs con herramientas optimizadas y flexibles.

Este proyecto busca también contribuir al campo académico, comparando mediante la evaluación de distintas métricas (precisión, recall, novedad y diversidad) que retornan algoritmos tradicionales, como KNN y Popularidad, con enfoques más modernos como BPRMF y BPR+GeoPenalty, y explorando el impacto de redes neuronales en sistemas de recomendación geográfica. Además, aplicaremos distintas técnicas de imputación de datos para mejorar las métricas mencionadas y compararlas con los datos sin imputar.

En definitiva, este proyecto se justifica no solo por su relevancia práctica en el ámbito de las aplicaciones de recomendación, sino también por su gran contribución al avance del conocimiento en métodos innovadores para el tratamiento de datos dispersos, aportando de manera significativa a la creación de ciudades y comunidades sostenibles (Objetivo de Desarrollo Sostenible 11). Al mejorar los sistemas de recomendación de puntos de interés, el proyecto facilita que los usuarios descubran y accedan a lugares y servicios dentro de las ciudades de manera más eficiente.

2. Revisión del Estado del Arte

Desde su aparición en los años 90, los sistemas de recomendación han experimentado una evolución constante. Esta evolución ha dado lugar a diferentes enfoques, como el filtrado colaborativo, los métodos basados en contenido y los modelos híbridos que combinan distintas técnicas (1). El filtrado colaborativo se ha consolidado como la técnica más influyente, ya que genera recomendaciones identificando patrones de comportamiento comunes entre distintos usuarios, estableciendo así similitudes en sus preferencias (2). Por otro lado, los métodos basados en contenido analizan las características de los ítems para encontrar relaciones entre ellos y mejorar la precisión de las sugerencias. En la actualidad, con los avances en Deep Learning y Reinforcement Learning, han surgido técnicas innovadoras que permiten capturar patrones de interacción más complejos, lo que ha llevado a una mejora significativa en las métricas de evaluación de estos sistemas (3). Por ejemplo, en el ámbito de las compras online, Amazon empezó utilizando un sistema de recomendación que combina filtrado colaborativo basado en ítems y métodos basados en contenido para sugerir productos a los usuarios (4). Este enfoque permite a Amazon personalizar las recomendaciones basadas en el historial de compras y visualizaciones de cada usuario, mejorando así la experiencia de compra, no obstante, hoy en día los modelos de recomendación de Amazon han incorporado técnicas de deep learning para mejorar estos filtros colaborativos. De manera similar, en las plataformas de streaming, Netflix emplea un sistema de recomendación basado en filtrado colaborativo que analiza las interacciones de los usuarios con el contenido para sugerir películas y series (5). Además, Netflix ha incorporado técnicas de aprendizaje profundo, como redes neuronales, para modelar las preferencias de los usuarios y mejorar la precisión de sus recomendaciones.

Sin embargo, el diseño de estos sistemas no es trivial. Deben combinar datos geográficos con las preferencias personales. A continuación, se presenta la notación básica y una serie de algoritmos de recomendación muy empleados.

2.1. Notación

En esta subsección, introduciremos la notación básica empleada en los sistemas de recomendación. En los problemas de recomendación, partimos de un conjunto de datos que registra interacciones entre usuarios e ítems. El conjunto de usuarios se denota como U , donde cada usuario individual se representa como u o v , y el número total de usuarios es $|U|$. El conjunto de ítems se denota como I , y un ítem

particular como i o j . Las interacciones entre usuarios e ítems forman nuestro conjunto de datos, que se divide en dos subconjuntos: D_{train} y D_{test} . El conjunto D_{train} se utiliza para entrenar un modelo de recomendación con parámetros θ . Una vez entrenado el modelo, se genera una lista de recomendaciones de longitud k para cada usuario. El conjunto de ítems recomendados se define como Rec . Para evaluar el modelo, utilizamos D_{test} , donde los ítems relevantes para cada usuario se denotan como Rel .

2.2. Algoritmos de Recomendación

En esta subsección, se describen distintos algoritmos de recomendación, incluyendo su origen y las fórmulas matemáticas asociadas para facilitar su comprensión.

2.2.1. Algoritmo de Recomendación Aleatorio (Random)

El algoritmo de recomendación aleatorio es un método simple que selecciona ítems al azar para recomendar a los usuarios, sin considerar preferencias previas ni características de los ítems. Su principal utilidad radica en servir como línea base para evaluar algoritmos más complejos y en destacar sus fortalezas en métricas como la *novedad*. Dado que no está sesgado hacia ítems populares, este algoritmo tiende a generar recomendaciones inesperadas y poco comunes, lo que lo hace realmente potente en términos de novedad. La recomendación se realiza, para cada usuario $u \in \mathcal{U}$ siguiendo la siguiente fórmula:

$$\mathcal{I}_u^{\text{rec}} \subset \mathcal{I} \setminus \mathcal{I}_u^{\text{train}}, \quad \text{con } |\mathcal{I}_u^{\text{rec}}| = k,$$

donde \mathcal{I} es el conjunto total de ítems del sistema, $\mathcal{I}_u^{\text{train}}$ es el subconjunto de ítems con los que el usuario u interactuó durante el entrenamiento. El espacio de muestreo, $\mathcal{I} \setminus \mathcal{I}_u^{\text{train}}$, garantiza que ninguna recomendación repita un ítem ya visto, y las recomendaciones se realizan mediante muestreo uniforme sin reemplazo, maximizando así la novedad y aleatoriedad de las recomendaciones.

2.2.2. Popularidad

El algoritmo de **Popularidad** es una técnica básica utilizada desde los inicios de los sistemas de recomendación en la década de 1990. Recomienda los ítems más populares, según la frecuencia de interacciones de cada artículo.

Fórmula asociada:

$$\text{pop}_i = \sum_{u \in \mathcal{U}} I_{u,i}$$

donde \mathcal{U} es el conjunto de usuarios, e $I_{u,i} = 1$ si el usuario u ha interactuado con el ítem i , y 0 en caso contrario. Recomendando para cada usuario una lista con los k ítems que tengan mayor pop_i que el usuario no haya visitado previamente.

2.2.3. K-Nearest Neighbors (KNN)

El algoritmo **K-Nearest Neighbors (KNN)** fue popularizado por el sistema GroupLens en 1994 (8). Existen dos enfoques principales para implementar este algoritmo: KNN basado en usuarios y KNN basado en ítems. Ambos métodos utilizan el concepto de similitud para generar recomendaciones, pero difieren en si esta similitud se calcula entre usuarios o entre ítems.

KNN basado en usuarios

En el KNN basado en usuarios, la similitud se calcula entre usuarios para identificar aquellos con gustos similares. Para un usuario objetivo u , el algoritmo busca los k usuarios más similares, conocidos como vecinos, y genera recomendaciones basándose en las interacciones de estos vecinos con ítems que el usuario objetivo no ha visitado.

La similitud entre usuarios puede calcularse de diversas formas, como la similitud del coseno, la correlación de Pearson o la similitud de Jaccard. En este trabajo, se emplea la similitud del coseno, definida como:

$$\text{sim}(u, v) = \frac{\sum_{i \in I_u \cap I_v} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I_v} r_{v,i}^2}}$$

donde I_u es el conjunto de ítems interactuados por el usuario u , y $r_{u,i}$ es el rating del usuario u para el ítem i .

Una vez calculada la similitud, el algoritmo identifica los k vecinos más cercanos al usuario objetivo, es decir, aquellos con mayor similitud. Luego, genera recomendaciones ponderando las interacciones de estos vecinos con ítems no visitados por el usuario objetivo, seleccionando aquellos con mayor score acumulado. La predicción de la puntuación de un ítem i para el usuario u se realiza con la siguiente fórmula:

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_k(u)} \text{sim}(u, v) \cdot r_{v,i}}{\sum_{v \in N_k(u)} |\text{sim}(u, v)|}$$

donde $N_k(u)$ representa el conjunto de los k vecinos más similares a u .

KNN basado en ítems

En el KNN basado en ítems, la similitud se calcula entre los ítems en lugar de entre los usuarios. Para un ítem dado i , el algoritmo identifica los k ítems más similares, basándose en cómo han sido interactuados por los usuarios. Las recomendaciones para un usuario u se generan sugiriendo ítems que son similares a aquellos con los que el usuario ha interactuado previamente.

La similitud entre ítems también puede calcularse mediante métodos como la similitud del coseno o la correlación de Pearson. Usando la similitud del coseno, esta se define como:

$$\text{sim}(i, j) = \frac{\sum_{u \in U_i \cap U_j} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U_i} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U_j} r_{u,j}^2}}$$

donde U_i es el conjunto de usuarios que han interactuado con el ítem i .

La predicción de la puntuación de un ítem i para el usuario u se basa en las puntuaciones que el usuario u ha asignado a los ítems similares a i . La fórmula empleada es:

$$\hat{r}_{u,i} = \frac{\sum_{j \in N_k(i)} \text{sim}(i, j) \cdot r_{u,j}}{\sum_{j \in N_k(i)} |\text{sim}(i, j)|}$$

donde $N_k(i)$ es el conjunto de los k ítems más similares a i que el usuario u ha interactuado.

Este enfoque basado en ítems puede ser más eficiente en sistemas con un gran número de usuarios, ya que las similitudes entre ítems pueden precomputarse y reutilizarse para generar recomendaciones.

2.2.4. Bayesian Personalized Ranking Matrix Factorization (BPRMF)

Bayesian Personalized Ranking Matrix Factorization (BPRMF) fue propuesto por Steffen Rendle et al. en 2009 (10). Este método optimiza el ranking de ítems en datos de retroalimentación implícita mediante factorización matricial, enfocándose en que los ítems visitados tengan mayor preferencia que los no visitados. La función de pérdida de BPRMF es:

$$\mathcal{L} = \sum_{(u,i,j) \in D} -\ln \sigma(\hat{r}_{u,i} - \hat{r}_{u,j}) + \lambda \|\theta\|^2$$

donde: D es el conjunto de tripletas que incluye un usuario u , un ítem visitado i , y un ítem no visitado j , σ es la función sigmoide que transforma la diferencia en una probabilidad, $\hat{r}_{u,i}$ y $\hat{r}_{u,j}$ son las predicciones de preferencia del usuario u por los ítems i y j , λ es un parámetro de regularización para evitar sobreajuste, θ representa los parámetros del modelo.

2.2.5. BPR+GeoPenalty

BPR+GeoPenalty es una extensión de BPRMF que incorpora información geográfica para mejorar las recomendaciones de POIs. En este modelo, se introduce una penalización geográfica basada en la distancia euclidiana entre el usuario y el ítem. La predicción de preferencia se calcula como:

$$\hat{r}_{u,i} = \text{latent_score} + \text{geo_penalty}$$

donde: $\text{latent_score} = \mathbf{u}_u \cdot \mathbf{i}_i + b_i$, $\text{geo_penalty} = -\lambda_{\text{geo}} \cdot \text{dist}(u, i)$, \mathbf{u}_u y \mathbf{i}_i son los vectores latentes del usuario u y del ítem i , b_i es el sesgo del ítem i , $\text{dist}(u, i)$ es la distancia euclidiana entre el usuario u , la media de las coordenadas de los ítems que visitó dicho usuario en train, y las coordenadas geoespaciales del ítem i , λ_{geo} es un parámetro de regularización geográfica. El modelo se entrena utilizando un enfoque de muestreo estocástico, actualizando los factores latentes y los sesgos para minimizar la función de pérdida de BPRMF con la penalización geográfica.

2.2.6. Redes Neuronales de Grafos (GNN) con Embeddings BERT

Redes Neuronales de Grafos (GNN) con Embeddings BERT se han utilizado en sistemas de recomendación para modelar relaciones complejas entre usuarios e ítems, integrando información contextual a través de embeddings de texto. En el trabajo de Javaji y Sarode (2023) (27), se propone un modelo híbrido que combina una GNN con embeddings BERT para recomendación de ítems. Este modelo utiliza embeddings de texto generados por BERT para capturar la información semántica, que se combinan con las características de los nodos en el grafo. La GNN, específicamente una implementación de GraphSAGE, se emplea para aprender representaciones de los nodos (usuarios e ítems) mediante la agregación de información de los vecinos. La fórmula de actualización de los nodos en GraphSAGE es:

$$\mathbf{h}_v^{(l)} = \sigma \left(\mathbf{W}^{(l)} \cdot \text{CONCAT} \left(\mathbf{h}_v^{(l-1)}, \text{AGGREGATE} \left(\{\mathbf{h}_u^{(l-1)} \mid u \in \mathcal{N}(v)\} \right) \right) \right)$$

donde: $\mathbf{h}_v^{(l)}$ es la representación del nodo v en la capa l , $\mathcal{N}(v)$ es el conjunto de vecinos del nodo v , AGGREGATE es una función de agregación (e.g., media, suma), $\mathbf{W}^{(l)}$ es una matriz de pesos aprendida, σ es una función de activación no lineal. Los embeddings BERT se utilizan para inicializar las características de los nodos de ítems, capturando la información semántica. Este enfoque permite al modelo generar recomendaciones más precisas al considerar tanto las interacciones usuario-ítem como el contenido textual de los ítems. En nuestro proyecto de recomendación de puntos de interés, vamos a emplear BERT para la creación de embeddings asociados a los venues, aprovechando la información de la categoría de cada venue y uniéndola con sus coordenadas geográficas.

2.2.7. Self-Attentive Sequential Recommendation (SASRec)

SASRec es un modelo de recomendación secuencial basado en la atención, propuesto por Kang y McAuley en 2018 (17). Este algoritmo modela las interacciones pasadas del usuario como secuencias y utiliza mecanismos de autoatención para capturar dependencias a largo plazo entre los ítems, superando a muchos modelos basados en RNNs y CNNs.

En SASRec, cada ítem visitado por un usuario se convierte en un vector de embedding. A partir de esa matriz de embeddings de tamaño $|I_u| \times d$ (donde $|I_u|$ es el número de ítems visitados por el usuario u y d la dimensión del embedding), se generan tres proyecciones lineales para construir las matrices:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V,$$

con $X \in R^{|I_u| \times d}$ la matriz de embeddings de los ítems, y $W^Q, W^K, W^V \in R^{d \times d_k}$ parámetros aprendibles. Así,

- Q (*queries*) captura “qué está preguntando” cada posición de la secuencia.
- K (*keys*) representa “la clave” de cada posición para comparar relevancia.
- V (*values*) contiene la información que se propagará si la clave resulta relevante.

La atención multi-head se calcula como

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V,$$

donde $\sqrt{d_k}$ escala el producto interno para estabilizar el gradiente. Este mecanismo asigna un peso $\alpha_{j,t}$ a cada par $(\mathbf{q}_j, \mathbf{k}_t)$, y la salida en la posición j es una combinación ponderada de los vectores \mathbf{v}_t .

En la práctica, SASRec emplea varias heads en paralelo, cada una con sus propias matrices W_h^Q, W_h^K, W_h^V . Sus resultados se concatenan y se proyectan finalmente:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H) W^O,$$

donde $\text{head}_h = \text{Attention}(QW_h^Q, KW_h^K, VW_h^V)$. Esto permite capturar distintos patrones de interacción simultáneamente.

Para recomendar el siguiente ítem, se toma la representación final de la última posición de la secuencia (es decir, $\mathbf{z}_{|I_u|}$) y se la compara con cada embedding de ítem candidato:

$$\text{score}(u, i) = \mathbf{z}_{|I_u|}^\top \mathbf{e}_i^{(\text{final})},$$

ordenando luego los ítems según ese score para obtener la lista de k recomendaciones. De este modo, el mecanismo de atención aprende a resaltar automáticamente aquellos ítems con los que ha interactuado el usuario que son más importantes para predecir el siguiente ítem.

2.3. Evaluación de Sistemas de Recomendación

Con el propósito de entender cómo funcionan los distintos sistemas de recomendación, es necesario evaluarlos. Para ello, como ocurre con cualquier modelo, primero debemos dividir el conjunto total de datos D en dos partes: un conjunto de entrenamiento (D_{train}) y otro de prueba (D_{test}). Esta división asigna datos al conjunto de entrenamiento y al conjunto de prueba, utilizando una semilla aleatoria para distribuir cada entrada de datos en uno de estos dos grupos. Con los datos de D_{train} , entrenamos el modelo, y luego evaluamos su desempeño en D_{test} usando métricas como la precisión (*accuracy*), el área bajo la curva ROC (AUC) o el error cuadrático medio (MSE), entre otras que se detallarán más adelante con sus fórmulas.

Sin embargo, la evaluación de los sistemas de recomendación se aparta del enfoque tradicional y en esto radica su forma de evaluación. La división de los datos no se hace de manera global siempre, sino que se puede hacer de forma específica para cada usuario. Comparando con otros modelos, como por ejemplo, un modelo de clasificación multiclase, sería como si todos los datos de una clase terminaran en el conjunto de prueba (18). La elección de la estrategia de división de datos es crucial, ya que puede influir significativamente en los resultados de la evaluación, como se ha demostrado en estudios recientes (19). Además, cabe destacar que muchas veces se emplea también un conjunto de validación, pero dada la naturaleza sparse del conjunto de datos, de momento lo obviaremos. A continuación, se describen brevemente las principales estrategias de división de datos identificadas en la literatura (19):

- **División temporal por usuario:** ordena las interacciones de cada usuario cronológicamente y reserva el 20 % más reciente de cada usuario para el conjunto de prueba. Es común, pero puede inducir filtración de información futura al modelo.
- **División temporal global:** utiliza un punto de corte temporal común para todos los usuarios. Es la estrategia más estricta y realista, ya que evita completamente la filtración de datos futuros, aunque reduce significativamente el número de datos disponibles.
- **Leave One Last Item:** selecciona la última interacción de cada usuario como prueba y la penúltima para validación. Es ampliamente usada y maximiza el conjunto de entrenamiento, aunque puede no reflejar bien el rendimiento general.
- **Leave One Last Basket/Session:** similar a la anterior, pero basada en sesiones o cestas completas de ítems. Es útil en escenarios como compras, pero su evaluación también es limitada temporalmente y sufre problemas similares de filtración.
- **División aleatoria:** elige aleatoriamente qué interacciones se usan para prueba. Es fácil de implementar, pero poco reproducible y ha sido sustituida en gran parte por estrategias temporales.
- **División por usuario:** separa a los usuarios en grupos de entrenamiento y prueba. Evalúa la capacidad del sistema frente al problema de *cold-start*, pero no todos los modelos pueden adaptarse a este escenario.

Para poder adaptarnos a la peculiaridad de la división de datos, es necesario aplicar métricas diferentes, que serán explicadas a continuación.

2.3.1. Métricas de Evaluación

A continuación, se describen las métricas utilizadas para evaluar el rendimiento de los sistemas de recomendación implementados, explicando cada término en las fórmulas. Cabe destacar que los resultados que arrojan las métricas están influenciados por el parámetro cutoff, o punto de corte, que define el número de recomendaciones consideradas en la evaluación. Se suelen emplear cutoffs de 5 o 10.

Novedad

La novedad mide la capacidad del sistema para recomendar ítems poco conocidos o inesperados. Se basa en la idea de que cuanto menos popular es un ítem, más novedoso será para el usuario. Se calcula acorde a esta formulación, que sigue el enfoque propuesto por Ziegler (9), permite evaluar cuánta información nueva aporta cada recomendación, priorizando ítems menos conocidos y, por tanto, más novedosos para los usuarios como la cantidad de *información* que aporta la recomendación, medida por

el logaritmo de la inversa de su popularidad:

$$\text{Novedad} = \frac{1}{|U|} \sum_{u=1}^{|U|} \left(\frac{1}{|R_u|} \sum_{i \in R_u} -\log_2 \left(\frac{\text{pop}_i}{\text{pop}_{\max}} \right) \right)$$

donde: R_u es el conjunto de ítems recomendados al usuario u , pop_i es la popularidad del ítem i , medida como el número de usuarios únicos que han interactuado con él en el conjunto de entrenamiento y pop_{\max} es la popularidad del ítem más popular (número máximo de usuarios únicos que han interactuado con un ítem).

Diversidad

La diversidad mide la amplitud del catálogo cubierto por las recomendaciones, mostrando la capacidad del sistema para evitar concentrarse en pocos ítems. Se calcula como:

$$\text{Diversidad} = \left| \bigcup_{u=1}^{|U|} R_u \right|,$$

donde U es el conjunto de usuarios y R_u la lista de ítems recomendados al usuario u . Este valor refleja cuántos ítems distintos se recomiendan en total; un número mayor indica mayor variedad en los resultados.

EPC(Expected Popularity Complement)

El EPC, propuesto por Vargas y Castells en 2011 (20), evalúa la impopularidad promedio de los ítems recomendados, penalizando la sobre-recomendación de ítems populares:

$$\text{EPC} = \frac{1}{|U|} \sum_{u=1}^{|U|} \left(\frac{1}{|R_u|} \sum_{i \in R_u} \left(1 - \frac{\text{pop}_i}{\text{pop}_{\max}} \right) \right)$$

Los términos son los mismos que en la métrica de Novedad. Sin embargo, a diferencia de la Novedad, el EPC aplica una penalización lineal a la popularidad en lugar de logarítmica.

Precisión

La precisión mide la proporción de ítems recomendados que son relevantes para el usuario:

$$\text{Precision} = \frac{|Rec_k \cap Rel|}{|Rec_k|}$$

donde: Rec_k es el conjunto de ítems recomendados para un cutoff de k . Rel es el conjunto de ítems relevantes (ítems visitados en el conjunto de prueba). Y $|Rec \cap Rel|$ es el número de ítems recomendados que son relevantes.

Recall

El recall mide la proporción de ítems relevantes que han sido recomendados:

$$\text{Recall} = \frac{|Rec_k \cap Rel|}{|Rel|}$$

F1

El F1-score es el promedio armónico entre precisión y recall, proporcionando un equilibrio entre ambas métricas:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.4. Técnicas para Afrontar el Problema de la Dispersión

La escasez de datos, conocida como *sparsity*, es un desafío crítico en los sistemas de recomendación de puntos de interés (POIs). En estos sistemas, los usuarios interactúan con un número muy reducido de POIs en comparación con el total disponible, lo que genera matrices de interacción usuario-ítem

extremadamente dispersas. Por ejemplo, en un dataset como el de Foursquare, un usuario promedio podría visitar menos del 0.1 % de los POIs disponibles en una ciudad, dejando la mayoría de las entradas en la matriz como valores faltantes. Esta dispersión dificulta que los algoritmos identifiquen patrones precisos y generalizables, reduciendo la calidad de las recomendaciones (7).

La sparsity se mide usualmente con la siguiente fórmula:

$$\text{Sparsity} = 1 - \frac{|\mathcal{D}|}{|\mathcal{U}| \cdot |\mathcal{I}|}$$

Siendo $|\mathcal{D}|$ el número total de check-ins, $|\mathcal{U}|$ el número total de usuarios $|\mathcal{I}|$ el número total de venues. Existen dos enfoques principales para abordar este problema:

1. **Técnicas de Imputación de Datos:** Estas estrategias buscan estimar o generar valores para las interacciones faltantes, enriqueciendo el conjunto de datos y permitiendo que los modelos aprendan patrones más robustos. Métodos como la imputación basada en promedios, factorización matricial o redes generativas son comunes en este enfoque (23).
2. **Modelos que Manejan Datos Dispersos de Forma Nativa:** Algunos algoritmos, especialmente los basados en aprendizaje profundo, como las redes neuronales de grafos (GNNs) o los modelos de autoatención, pueden capturar relaciones complejas incluso con datos escasos, sin necesidad de imputación explícita (3).

Entre los modelos de aprendizaje profundo que destacan en este contexto se encuentran:

- **Flashback:** Propuesto por Xuan Rao y Lisi Chen en 2018 (21), este modelo utiliza redes neuronales para recordar los movimientos pasados del usuario, integrando datos temporales y espaciales para mejorar las recomendaciones en escenarios dispersos.
- **STHGCN:** Desarrollado por Xiaodong Yan en 2020 (22), emplea hipergrafos convolucionales espacio-temporales para modelar interacciones, capturando dependencias complejas entre usuarios y POIs con datos limitados.

2.4.1. Imputación de Datos

La imputación de datos se ha consolidado como una técnica clave para mitigar la dispersión en sistemas de recomendación. Este enfoque consiste en estimar o generar interacciones faltantes en la matriz usuario-ítem, transformando un conjunto de datos disperso en uno más denso y útil para el entrenamiento de modelos. La imputación puede mejorar significativamente métricas como precisión, recall y novedad al proporcionar a los algoritmos una visión más completa de las preferencias de los usuarios (7).

Una técnica ampliamente utilizada es tomar las recomendaciones obtenidas sin emplear imputación como una forma de estimar los valores faltantes, ya que, al fin y al cabo, son los ítems que, según los algoritmos, es más probable que agraden al usuario. En este método, se entrena un modelo inicial (e.g., BPRMF o KNN) con los datos originales dispersos ($\mathcal{D}_{\text{ciudad, train}}$), se evalúa dicho algoritmo en ($\mathcal{D}_{\text{ciudad, test}}$), se obtienen las recomendaciones y se emplean para rellenar las entradas vacías de la matriz, creando de esta forma un nuevo conjunto de datos ($\mathcal{D}_{\text{ciudad, train}}^{\text{imp}}$). Por ejemplo, si un usuario no ha visitado un POI, pero el modelo predice una alta probabilidad de interés basada en patrones de usuarios similares, lo recomienda y ese valor predicho puede imputarse como una interacción implícita. Este enfoque es simple y efectivo, ya que aprovecha las capacidades predictivas del modelo sin requerir suposiciones externas complejas (24). Estudios como el de Koren et al. (2009) (25) han demostrado que combinar factorización matricial con imputación basada en predicciones mejora el rendimiento en datasets dispersos.

Otros métodos incluyen el uso de redes generativas adversariales (GANs), que generan interacciones sintéticas realistas (23), y técnicas estadísticas como la imputación por promedios o vecinos cercanos. Sin embargo, la imputación basada en recomendaciones sin imputación inicial destaca por su practicidad y por su capacidad para mantener la coherencia con las preferencias implícitas del usuario, siendo una solución adoptada en numerosos sistemas prácticos (24).

2.5. Foursquare como fuente de datos

Foursquare¹ es una plataforma de localización que proporciona servicios basados en la geolocalización de los usuarios, tales como recomendaciones personalizadas de lugares, registros de visitas (*check-ins*) y

¹<https://foursquare.com/>

análisis de movilidad. Originalmente lanzada como una red social, ha evolucionado hacia un proveedor de datos geoespaciales, siendo ampliamente utilizada en estudios de ciencia de datos, geografía urbana y sistemas de recomendación.

Uno de los mayores atractivos de Foursquare para la investigación académica es su conjunto de datos abiertos, como el “Foursquare Check-in Dataset”, que incluye millones de interacciones de usuarios con puntos de interés (POIs) en diversas ciudades del mundo. Estos datos contienen información contextual rica, incluyendo tipo de lugar, coordenadas geográficas, tiempo de visita y usuario anónimo, lo cual permite el estudio de patrones de comportamiento urbano y el entrenamiento de modelos de recomendación geoespacial. Algunos trabajos que emplean datos de esta plataforma son Yang et al. (26), Noulas et al. (14), Cheng et al. (15) y Bao et al. (16).

3. Metodología

Para llevar a cabo este estudio, se siguió la siguiente metodología. Se empleó un conjunto de datos de *check-ins*, provenientes de la plataforma Foursquare, representado como \mathcal{D} , que contiene interacciones entre usuarios y puntos de interés (POIs), descrito en la sección 2.5. Después del preprocesamiento descrito en la sección 3.4.1, se filtró el dataset \mathcal{D} , quedándonos únicamente con los *check-ins* de Londres, Nueva York y Tokio. A partir de este filtrado, se generaron tres subconjuntos de datos, uno por cada ciudad, denotados como $\mathcal{D}_{\text{ciudad}}$ (es decir, $\mathcal{D}_{\text{Londres}}$, $\mathcal{D}_{\text{Nueva York}}$ y $\mathcal{D}_{\text{Tokio}}$).

Para cada $\mathcal{D}_{\text{ciudad}}$, se empleó la técnica de **división temporal por usuario**, definida en la sección 2.3, para obtener los conjuntos de entrenamiento ($\mathcal{D}_{\text{ciudad, train}}$) y de prueba ($\mathcal{D}_{\text{ciudad, test}}$). Los algoritmos de recomendación se entrenaron utilizando $\mathcal{D}_{\text{ciudad, train}}$, y sus hiperparámetros fueron optimizados mediante la biblioteca **skopt**, aplicando un enfoque de **optimización bayesiana**. Este método permite explorar de forma eficiente el espacio de hiperparámetros, mejorando el rendimiento de los modelos al maximizar métricas clave como *precision*, *recall* y *F1*. Una vez obtenidas las métricas e hiperparámetros óptimos para cada uno de los algoritmos, se emplearon diversas técnicas de imputación aplicadas a $\mathcal{D}_{\text{ciudad, train}}$, generando un nuevo conjunto de interacciones $\mathcal{D}_{\text{ciudad, train}}^{\text{imp}}$. Estos nuevos archivos de entrenamiento imputado se generaron de la siguiente forma, tomamos las recomendaciones de un algoritmo anterior (en nuestro caso BPRMF), e incluimos un 20 % por usuario de estas recomendaciones, en caso de que el usuario no tenga más de 5 *check-ins* en train, añadiremos solamente un nuevo *check-in*, creando de esta forma un conjunto de datos imputados. Se entrenaron los distintos sistemas de recomendación con este nuevo conjunto de datos, se optimizaron de nuevo con **skopt** y se evaluaron en $\mathcal{D}_{\text{ciudad, test}}$, permitiendo comparar métricas e hiperparámetros en ambos enfoques: con y sin imputación.

3.1. Tecnologías Utilizadas

Para llevar a cabo este proyecto, se emplearon las siguientes librerías:

- **Python** (versión 3.10): Lenguaje de programación principal por su versatilidad y amplia comunidad.
- **NumPy** (versión 1.26.4) y **Pandas** (versión 2.2.3): Librerías esenciales para el manejo y preprocesamiento de los datos.
- **Scikit-learn** (versión 1.5.1): Utilizada en distintos algoritmos de recomendación para ajustar los datos a las características de cada uno de estos modelos.
- **PyTorch** (versión 2.4.0): *Framework* empleado para el desarrollo de redes neuronales avanzadas.
- **Json** (incluido en la biblioteca estándar de Python 3.10): Formato empleado para el almacenamiento de resultados e hiperparámetros de los distintos modelos.
- **Skopt** (versión 0.10.2): Utilizada para realizar optimización bayesiana de los hiperparámetros, permitiendo ajustar de manera eficiente los parámetros de los modelos y mejorar su rendimiento.
- **Torch** (versión 2.4.0): Librería empleada para construir redes neuronales optimizando el uso de distintos núcleos de GPU
- **Torch-geometric** (versión 2.6.1): Librería empleada para construir redes neuronales basadas en grafos (GNNs).

3.2. Herramientas

Las herramientas utilizadas incluyen:

- **Jupyter Notebook:** Para experimentación interactiva y análisis de resultados.
- **Git:** Para el control de versiones y colaboración en el código.
- **LaTeX:** Para la redacción y presentación formal del documento final.

3.3. Objetivos

Los objetivos principales del proyecto fueron:

- Implementar algoritmos de recomendación de POIs, desde métodos básicos hasta avanzados.
- Optimizar hiperparámetros utilizando métodos bayesianos para mejorar el rendimiento de cada uno de los algoritmos para cada una de las ciudades que vamos a estudiar.
- Abordar el desafío de la dispersión de datos mediante técnicas de imputación.
- Evaluar el impacto de las técnicas de imputación en las métricas de rendimiento.

3.4. Conjunto de Datos

En este proyecto, se empleó un extenso conjunto de datos de *check-ins* obtenido de la plataforma Foursquare, el cual registra interacciones entre usuarios y puntos de interés (POIs) en múltiples ciudades a nivel global, como se detalla en la sección 2.5. Este dataset principal se complementa con dos conjuntos adicionales: *Cities*, que proporciona información geográfica de las ciudades, y *Pois*, que detalla las características específicas de cada POI, como su ubicación y categoría.

- **check-ins.csv:** Contiene las interacciones entre usuarios y POIs, con columnas como *user_id* (int), identificador único del usuario; *venue_id* (str), identificador del punto de interés; *UTC_time* (str), *timestamp* en tiempo universal coordinado; y *time_offset* (int), desplazamiento horario respecto a UTC.
- **Cities:** Incluye datos geográficos de las ciudades, con columnas como *city_name* (nombre de la ciudad), *latitude* y *longitude* (coordenadas geográficas), *country_code* y *country_name* (código y nombre del país, respectivamente), y *city_type* (tipo de ciudad, e.g., capital, capital de provincia).
- **Pois:** Proporciona información detallada sobre cada punto de interés, incluyendo *venue_id* (identificador único del POI), *latitude* y *longitude* (coordenadas geográficas), *category_name* (categoría del POI, e.g., restaurante, hotel, museo) y *country_code* (código del país).

3.4.1. Preprocesamiento de Datos

Para preparar el conjunto de datos para el análisis, se realizaron varios pasos de preprocesamiento con el objetivo de optimizar los recursos computacionales y garantizar la utilidad de los datos. A continuación, se detallan los procesos llevados a cabo:

1. **Reducción y enriquecimiento del conjunto de datos:** Debido a limitaciones de hardware, se redujo el número de *check-ins* procesados, restringiendo el análisis a tres ciudades principales: Nueva York, Tokio y Londres. Para asignar cada *check-in* a su ciudad correspondiente, se empleó la distancia Haversine, identificando la ciudad más cercana según las coordenadas geográficas (*latitude* y *longitude*) de los puntos de interés (POIs) en el dataset *Cities*, y verificando que el *country_code* del POI coincidiera con el de la ciudad asignada. Posteriormente, se fusionó el dataset *Pois* con *check-ins.csv* usando la columna *venue_id*, enriqueciendo los *check-ins* con información adicional como coordenadas, categoría y código del país. Este proceso generó un dataset más completo y estructurado, optimizado para el análisis.
2. **Eliminación de *check-ins* residenciales:** Se excluyeron los *check-ins* cuyos *venues* pertenecen a la categoría "residencial", dado que estos puntos de interés son de carácter privado y no accesibles para todos los usuarios. Este paso garantiza que el análisis se centre en lugares públicos y pertinentes.

3. **Creación de la columna "score"**: Como el dataset original no cuenta con una columna de *rating* explícita, se introdujo la columna *score*, que indica el número de veces que un usuario ha visitado un mismo POI. Esta métrica refleja la preferencia implícita del usuario y permite evaluar la intensidad de sus interacciones.
4. **División de datos en *train* y *test***: Para cada ciudad, se crearon conjuntos de *train* y *test* siguiendo la lógica de división temporal por usuario. El 80 % de los *check-ins* más antiguos de cada usuario se asignaron a *train*, mientras que el 20 % más reciente se destinó a *test*, simulando una serie temporal para predecir futuras visitas. Los modelos de recomendación se entrenarán con esta estructura, aprovechando que se conocen las visitas reales para validar su desempeño. En caso de usuarios con un solo *check-in* por ciudad, este se incorporó directamente al conjunto de *train*, puesto que si lo enviásemos a *test*, no tendríamos información previa sobre ellos y sería equivalente a hacer una recomendación aleatoria.

3.4.2. Estadísticas del Conjunto de Datos por Ciudad

En esta subsección se presentan las estadísticas clave del conjunto de datos utilizado, centradas en las ciudades de Londres, Nueva York y Tokio. Estas estadísticas, que abarcan el número de *venues*, usuarios únicos y *check-ins* totales por ciudad, son fundamentales para entender la escala del problema de recomendación en cada contexto. La cantidad de datos y su dispersión (*sparsity*) influyen directamente en el rendimiento de los modelos de recomendación, donde un mayor volumen de datos y una menor *sparsity* podrían favorecer resultados más precisos, aunque otros factores también son relevantes. La siguiente figura resume estas estadísticas, destacando las diferencias entre las ciudades.

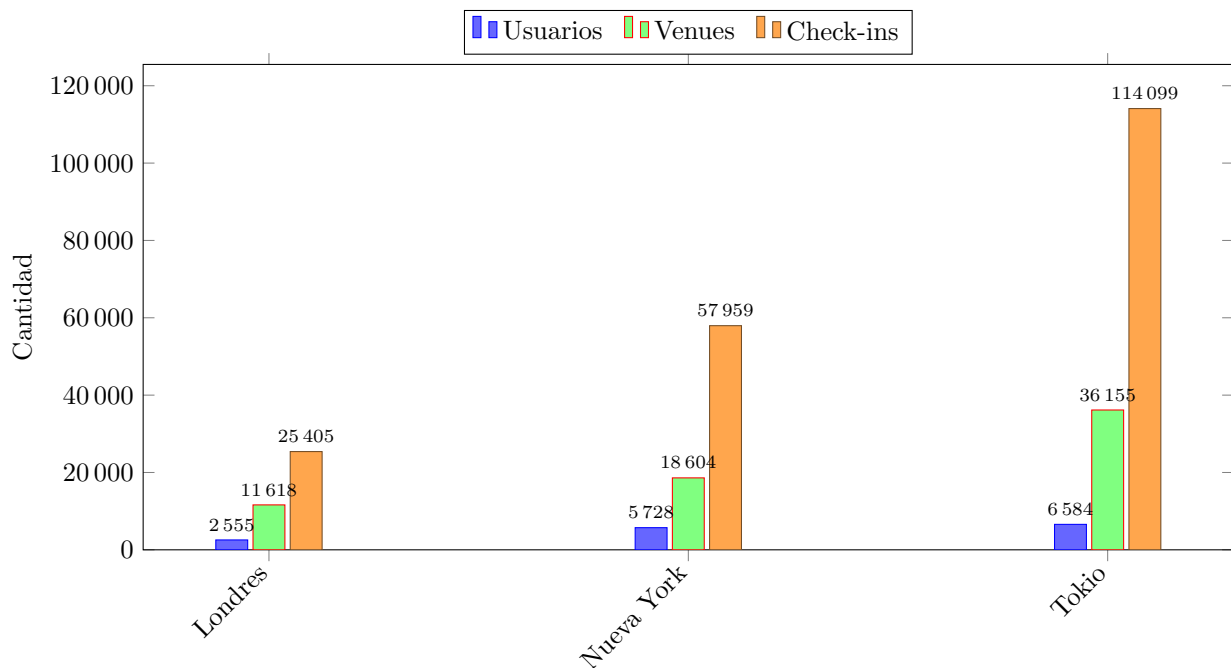


Figura 1: Comparativa del número de usuarios, venues y check-ins en Londres, Nueva York y Tokio.

Sparsity de los Datos

Como ya hemos mencionado previamente, la *sparsity* es el gran problema al que se enfrentan los recomendadores de POIs, afectando especialmente a los algoritmos basados en similitud y en aprendizaje profundo, los cuales requieren una cantidad considerable de datos por usuario e *item* para aprender representaciones robustas. Por ello, conocer el grado de dispersión del *dataset* resulta crucial para interpretar el rendimiento de los modelos.

A continuación se muestra una tabla y un gráfico que resumen el grado de dispersión, expresado como el porcentaje de interacciones existentes respecto al total posible, para las tres ciudades analizadas:

Ciudad	Dispersión de check-ins (%)
Londres	99,91
Nueva York	99,95
Tokio	99,95

Tabla 1: Dispersión de check-ins por ciudad

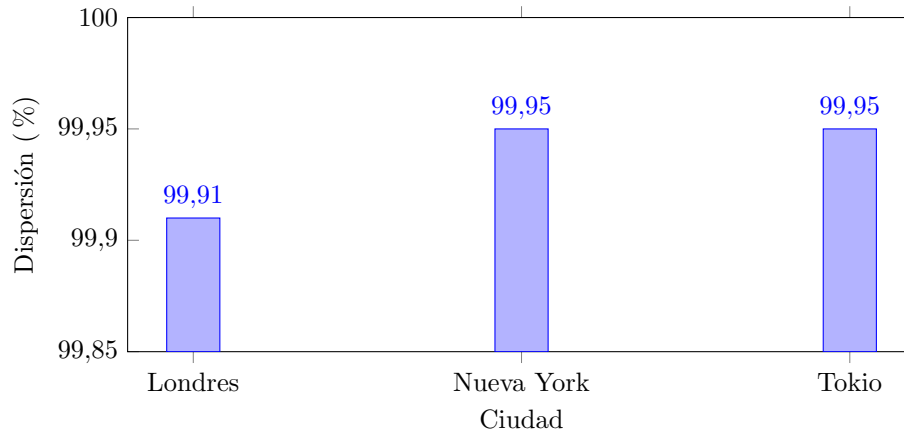


Figura 2: Dispersión porcentual de check-ins en cada ciudad

Estos datos reflejan que las tres ciudades presentan una *sparsity* extremadamente alta, lo que representa un desafío común para los modelos de recomendación implementados.

4. Resultados

En esta sección, se presentan los resultados experimentales organizados por ciudad. Para cada ciudad, se muestran dos tablas: una para un cutoff de 5 y otra para un cutoff de 10. En cada tabla, se incluyen los resultados tanto con imputación como sin imputación, indicados en la columna Imputación con Sí para los casos con imputación y No para los casos sin imputación. Los mejores valores para cada métrica dentro de cada tabla se resaltan en negrita.

4.1. Resultados de Londres

4.1.1. Resultados con cutoff de 5

Algoritmo	Imputación	Novedad	EPC	Precisión	Recall	F1	Diversidad
Random	No	10,8787	0,9993	0,0002	0,0003	0,0002	6.874
Random	Sí	10,8678	0,9993	0,0003	0,0040	0,0006	6.813
BPRMF	No	4,6895	0,9610	0,0175	0,0559	0,0240	26
BPRMF	Sí	4,6318	0,9595	0,0194	0,0627	0,0269	13
BPR+GeoPenalty	No	4,6930	0,9610	0,0172	0,0541	0,0236	25
BPR+GeoPenalty	Sí	4,6311	0,9595	0,0196	0,0620	0,0270	11
GNN	No	4,8368	0,9636	0,0171	0,0561	0,0239	10
GNN	Sí	4,7209	0,9616	0,0187	0,0584	0,0257	14
KNN_usuarios	No	8,7825	0,9929	0,0078	0,0226	0,0102	2.426
KNN_usuarios	Sí	7,5183	0,9806	0,0128	0,0353	0,0163	1.226
KNN_items	No	10,8555	0,9989	0,0022	0,0062	0,0029	4.828
KNN_items	Sí	7,7275	0,9790	0,0133	0,0313	0,0164	3.294
Popularidad	No	4,6372	0,9596	0,0203	0,0649	0,0281	9
Popularidad	Sí	4,6282	0,9594	0,0197	0,0624	0,0271	9

Tabla 2: Resultados Londres con cutoff de 5

El modelo de **Popularidad** destaca en precisión (**0,0203**), recall (**0,0649**) y F1, aunque con una diversidad muy limitada (**9 ítems**). Los modelos de **factorización de matrices** (BPRMF y BPR+GeoPenalty) logran también buenos resultados en relevancia, con precisión en torno al **0,017–0,019**; no obstante, las métricas de novedad, EPC y diversidad siguen siendo bastante bajas, aunque algo mejores que las de **Popularidad**. La **GNN** ofrece un rendimiento medianamente bueno en cuanto a relevancia, alcanzando un F1 de **0,0257**, pero las métricas relacionadas con cuán novedosas son sus predicciones son bastante bajas. Por otro lado, los modelos **KNN** muestran alta diversidad sin imputación (hasta **4.828 ítems**) pero baja precisión, mientras que el modelo **Random** alcanza la máxima novedad (**10,8787**) y diversidad (**6.874**), sacrificando completamente la relevancia.

Algoritmo	Precisión (%)	Recall (%)	EPC (%)	Novedad (%)
Random	+100,00	+371,30	-0,01	-0,10
BPRMF	+10,86	+12,18	-0,16	-1,23
BPR+GeoPenalty	+13,95	+14,61	-0,16	-1,32
GNN	+9,36	+4,10	-0,21	-2,39
KNN_usuarios	+64,10	+56,01	-1,24	-14,38
KNN_items	+504,55	+407,71	-1,99	-28,81
Popularidad	-2,96	-3,85	-0,02	-0,19

Tabla 3: Porcentaje de mejora al aplicar imputación para Londres con cutoff = 5

Los modelos **KNN_items** y **KNN_usuarios** son los que más se benefician en términos de relevancia, con incrementos del **+504,55 %** y **+64,10 %** en precisión, y **+407,71 %** y **+56,01 %** en recall, respectivamente. Sin embargo, esta mejora viene acompañada de una fuerte pérdida de capacidad exploratoria, con reducciones del **-28,81 %** en novedad y del **-1,99 %** en EPC para **KNN_items**. Los modelos de **factorización de matrices** (BPRMF y BPR+GeoPenalty) presentan mejoras moderadas en precisión (**+10,86 %** y **+13,95 %**) y recall, con caídas ligeras en novedad (alrededor del **-1,3 %**). La **GNN** también muestra una ganancia leve en precisión (**+9,36 %**), pero pierde un **-2,39 %** en novedad. Finalmente, el modelo de **Popularidad** sufre ligeras pérdidas en todas las métricas, incluyendo una reducción del **-2,96 %** en precisión y del **-3,85 %** en recall, lo que sugiere que no se beneficia de la imputación y que su desempeño depende más de las interacciones observadas que de las inferidas.

4.1.2. Resultados con cutoff de 10

Algoritmo	Imputación	Novedad	EPC	Precisión	Recall	F1	Diversidad
Random	No	10,8678	0,9993	0,0001	0,0009	0,0002	9.708
Random	Sí	10,8787	0,9993	0,0001	0,0010	0,0002	9.713
BPRMF	No	4,8502	0,9647	0,0134	0,0818	0,0211	42
BPRMF	Sí	4,8058	0,9639	0,0130	0,0813	0,0207	25
BPR+GeoPenalty	No	4,9116	0,9659	0,0129	0,0820	0,0206	54
BPR+GeoPenalty	Sí	4,8068	0,9639	0,0133	0,0822	0,0210	35
GNN	No	5,0697	0,9682	0,0123	0,0776	0,0197	10
GNN	Sí	4,9928	0,9670	0,0150	0,0944	0,0239	28
KNN_usuarios	No	8,9599	0,9933	0,0062	0,0364	0,0097	3.826
KNN_usuarios	Sí	8,3623	0,9869	0,0083	0,0461	0,0127	2.391
KNN_items	No	10,8059	0,9988	0,0015	0,0084	0,0023	6.370
KNN_items	Sí	8,4187	0,9839	0,0085	0,0407	0,0126	5.355
Popularidad	No	4,8516	0,9648	0,0143	0,0891	0,0228	17
Popularidad	Sí	4,8062	0,9639	0,0134	0,0837	0,0213	17

Tabla 4: Resultados Londres con cutoff de 10

Con un *cutoff* mayor, se observa un comportamiento claro: la **precisión** tiende a disminuir, mientras que el **recall** y la **diversidad** aumentan. Esto responde al hecho de que al recomendar más ítems por usuario, se incrementa la probabilidad de acierto (mayor recall), aunque disminuya la proporción de aciertos sobre el total de recomendaciones (menor precisión).

El modelo de **Popularidad** mantiene buenos resultados en relevancia, con una precisión de **0,0143** y un recall de **0,0891**, pero su diversidad sigue siendo limitada (**17 ítems**). Los modelos de **factorización de matrices** (**BPRMF** y **BPR+GeoPenalty**) alcanzan una precisión similar (**0,013–0,014**) y un recall algo inferior (**0,081–0,082**), con algo más de diversidad (hasta **54 ítems**). La **GNN**, especialmente tras imputación, obtiene uno de los mejores recalls (**0,0944**) con una diversidad aceptable (**28 ítems**). Los modelos **KNN** siguen destacando por su alta diversidad (hasta **6.370 ítems** sin imputar), y mejoran notablemente en precisión y recall con imputación. Por último, el modelo **Random** vuelve a ser líder en **novedad** (**10,8787**) y **diversidad** (**9.713**), aunque con una precisión de apenas **0,0001**, lo que lo hace poco útil para recomendaciones relevantes.

Algoritmo	Precisión (%)	Recall (%)	EPC (%)	Novedad (%)
Random	+0,00	+13,63	+0,00	+0,10
BPRMF	-2,99	-0,61	-0,83	-0,91
BPR+GeoPenalty	+3,10	+0,24	-0,21	-2,14
GNN	+21,95	+21,61	-0,12	-1,52
KNN_usuarios	+32,96	+26,46	-0,65	-6,67
KNN_items	+467,78	+384,30	-1,50	-22,06
Popularidad	-6,29	-6,06	-0,10	-0,93

Tabla 5: Porcentaje de mejora al aplicar imputación para Londres con cutoff = 10

La imputación continúa mostrando efectos dispares según el tipo de modelo, aunque su impacto global es algo más atenuado en comparación con *cutoff* = 5. En los modelos **KNN**, los efectos siguen siendo notables: **KNN_items** incrementa su precisión en un **467,78 %** y su **recall** en un **384,30 %**, aunque sufre una caída significativa de **22,06 %** en novedad y una pérdida de EPC de **1,50 %**. **KNN_usuarios** muestra un patrón similar, con mejoras del **32,96 %** en precisión y del **26,46 %** en recall, acompañadas de una pérdida de novedad del **6,67 %**. La **GNN** también se beneficia de la imputación, con un aumento en precisión del **21,95 %** y en recall del **21,61 %**, mientras que la pérdida de novedad es más contenida (**-1,52 %**), lo que refuerza su perfil de modelo equilibrado. Por su parte, los modelos de **factorización de matrices** presentan resultados más estables: **BPRMF** pierde ligeramente en todas las métricas, mientras que **BPR+GeoPenalty** mejora marginalmente en precisión y recall, con una caída modesta en novedad. El modelo de **Popularidad** presenta pérdidas moderadas en precisión (**-6,29 %**) y recall (**-6,06 %**), confirmando que la imputación tiene poco impacto positivo en modelos que ya dependen fuertemente de los ítems más frecuentes. Finalmente, el modelo **Random** se mantiene prácticamente inalterado, con variaciones mínimas, reflejando nuevamente su independencia frente a la estructura del conjunto de datos imputado.

4.2. Resultados de Nueva York

4.2.1. Resultados con cutoff de 5

Algoritmo	Imputación	Novedad	EPC	Precisión	Recall	F1	Diversidad
Random	No	11,8503	0,9996	0,00008	0,00008	0,00007	13.486
Random	Sí	11,8494	0,9996	0,00021	0,00060	0,00028	13.453
BPRMF	No	3,9150	0,9310	0,0330	0,1080	0,0460	34
BPRMF	Sí	3,8980	0,9300	0,0325	0,1092	0,0462	63
BPR+GeoPenalty	No	3,9050	0,9310	0,0330	0,1110	0,0470	19
BPR+GeoPenalty	Sí	3,8970	0,9300	0,0327	0,1093	0,0464	53
GNN	No	4,1890	0,9390	0,0320	0,1070	0,0450	10
GNN	Sí	3,9660	0,9330	0,0327	0,1107	0,0466	10
KNN_usuarios	No	9,9886	0,9924	0,0108	0,0306	0,0144	4.314
KNN_usuarios	Sí	8,1781	0,9677	0,0223	0,0611	0,0293	2.116
KNN_items	No	11,7568	0,9981	0,0050	0,0138	0,0065	8.457
KNN_items	Sí	6,6528	0,9556	0,0252	0,0697	0,0333	4.973
Popularidad	No	3,8840	0,9300	0,0330	0,1100	0,0460	10
Popularidad	Sí	3,8840	0,9300	0,0327	0,1093	0,0463	10

Tabla 6: Resultados para Nueva York con cutoff de 5

Con *cutoff* = 5, los resultados para Nueva York se mantienen en consonancia con lo observado para Londres. Los modelos de **Popularidad** y **BPR+GeoPenalty** lideran en precisión (**0,0330**) y recall (**0,1110**), aunque con baja diversidad. La **GNN** alcanza un **recall** competitivo (**0,1107**) y mejora levemente con imputación. Los modelos **KNN** destacan por su alta diversidad (hasta **8.457 ítems**) y muestran grandes mejoras en precisión tras imputación (hasta **0,0252**). El modelo **Random** conserva su perfil explorador, con la mayor novedad (**11,8503**) y diversidad (**13.486**), pero sin relevancia práctica.

Algoritmo	Precisión (%)	Recall (%)	EPC (%)	Novedad (%)
Random	+162,50	+650,00	+0,00	-0,01
BPRMF	-1,52	+1,11	-0,11	-0,43
BPR+GeoPenalty	-0,91	-1,53	-0,11	-0,21
GNN	+2,19	+3,46	-0,64	-5,32
KNN_usuarios	+106,48	+99,67	-2,48	-18,13
KNN_items	+404,00	+405,07	-4,26	-43,42
Popularidad	-0,91	-0,64	0,00	0,00

Tabla 7: Porcentaje de mejora al aplicar imputación en Nueva York con cutoff = 5

Los efectos de la imputación en Nueva York son similares a los observados en Londres, pero en este caso los algoritmos de **factorización de matrices** (**BPRMF** y **BPR+GeoPenalty**) apenas varían. En los modelos **KNN**, se produce una fuerte mejora en precisión y recall (hasta +404 %), a costa de caídas marcadas en diversidad y novedad (-43,4 %). La **GNN** mejora ligeramente en relevancia, pero pierde algo de capacidad exploratoria. Los modelos de **factorización de matrices** (**BPRMF** y **BPR+GeoPenalty**) experimentan cambios mínimos. En **Popularidad** y **Random**, el impacto es prácticamente nulo, confirmando su independencia estructural frente a la imputación.

4.2.2. Resultados con cutoff de 10

Algoritmo	Imputación	Novedad	EPC	Precisión	Recall	F1	Diversidad
Random	No	11,841	0,9996	0,0001	0,0003	0,0001	17.169
Random	Sí	11,845	0,9996	0,0001	0,0002	0,0001	17.135
BPRMF	No	4,420	0,949	0,0218	0,1421	0,0354	34
BPRMF	Sí	4,432	0,949	0,0219	0,1417	0,0354	46
BPR+GeoPenalty	No	4,514	0,9498	0,0216	0,1401	0,0349	232
BPR+GeoPenalty	Sí	4,406	0,9483	0,0216	0,1401	0,0349	21
GNN	No	4,725	0,9532	0,0212	0,1381	0,0344	15
GNN	Sí	4,783	0,9555	0,0202	0,1314	0,0328	18
KNN_usuario	No	10,256	0,9940	0,0077	0,0438	0,0121	6.678
KNN_usuario	Sí	9,291	0,9806	0,0135	0,0737	0,0209	4.180
KNN_item	No	11,803	0,9984	0,0034	0,0185	0,0052	11.020
KNN_item	Sí	8,211	0,9713	0,0158	0,0858	0,0245	8.993
Popularidad	No	4,423	0,949	0,0220	0,1420	0,0356	16
Popularidad	Sí	4,401	0,948	0,0220	0,1422	0,0356	16

Tabla 8: Resultados para Nueva York con cutoff = 10

Los resultados obtenidos para Nueva York con *cutoff* = 10 reflejan el mismo patrón observado en Londres: al aumentar el número de ítems recomendados, se incrementan el **recall** y la **diversidad**, a costa de una leve disminución en la **precisión**. El modelo de **Popularidad** es el que alcanza la mayor **precisión** (**0,0220**) y **recall** (**0,1422**) entre todos los métodos, mientras que los modelos **KNN_usuario** y **KNN_item** sobresalen en **diversidad** con valores muy elevados, aunque con menor relevancia. Los modelos de **factorización de matrices** (**BPRMF** y **BPR+GeoPenalty**) mantienen también una relevancia competitiva. El modelo **Random** se mantiene como el más destacado en **novedad** (**11,845**) y **diversidad** (**17.169**).

Algoritmo	Precisión (%)	Recall (%)	EPC (%)	Novedad (%)
Random	-20,00	-34,21	+0,00	+0,04
BPRMF	+0,46	-0,28	+0,00	+0,27
BPR+GeoPenalty	+0,00	+0,00	+0,00	+0,00
GNN	-4,72	-4,85	+0,21	+1,23
KNN_usuario	+74,94	+68,14	-1,35	-9,41
KNN_item	+361,36	+363,07	-2,71	-30,43
Popularidad	+0,00	+0,14	-0,11	-0,50

Tabla 9: Porcentaje de mejora al aplicar imputación para Nueva York con cutoff = 10

Los resultados se mantienen en la línea de lo previamente observado muestran que la **imputación** mejora notablemente la relevancia de los modelos **KNN**, con incrementos de hasta un **+361,36 %** en **precisión** y **+363,07 %** en **recall** para **KNN_item**, aunque a costa de fuertes caídas en **novedad** (**-30,43 %**) y **EPC** (**-2,71 %**). Entre lo más destacado encontramos que **GNN** sufre pérdidas en **precisión** y **recall** (**-4,72 %** y **-4,85 %**), contrario a lo observado en el resto de tablas. Por su parte, **Popularidad**, **BPR+GeoPenalty** y modelos como **BPRMF** apenas varían, con mejoras marginales (**+0,46 %** en precisión) y una ligera ganancia en **novedad** (**+0,27 %**) apenas se ven afectados por la imputación. En resumen, la imputación potencia especialmente la relevancia de los **KNN**, pero con un coste claro en métricas relacionadas con novedad.

4.3. Resultados de Tokio

4.3.1. Resultados con cutoff de 5

Algoritmo	Imputación	Novedad	EPC	Precisión	Recall	F1	Diversidad
Random	No	12,1183	0,9996	0,00013	0,00010	0,00011	20.558
Random	Sí	12,1161	0,9996	0,00007	0,00018	0,00007	20.727
Popularidad	No	3,1850	0,8850	0,0540	0,0870	0,0560	13
Popularidad	Sí	3,1741	0,8844	0,0529	0,0849	0,0549	13
BPRMF	No	3,1770	0,8850	0,0520	0,0840	0,0540	16
BPRMF	Sí	3,1919	0,8846	0,0516	0,0835	0,0538	93
BPR+GeoPenalty	No	3,1790	0,8850	0,0520	0,0830	0,0540	18
BPR+GeoPenalty	Sí	3,1920	0,8846	0,0517	0,0835	0,0538	93
GNN	No	3,4350	0,8940	0,0470	0,0810	0,0500	14
GNN	Sí	3,2676	0,8888	0,0523	0,0865	0,0551	13
KNN_usuarios	No	8,4563	0,9751	0,0213	0,0334	0,0219	4.584
KNN_usuarios	Sí	5,6134	0,9213	0,0462	0,0660	0,0461	1.987
KNN_items	No	11,2702	0,9868	0,0198	0,0298	0,0198	14.389
KNN_items	Sí	4,7121	0,9086	0,0489	0,0710	0,0492	11.358

Tabla 10: Resultados para Tokio con cutoff = 5

Para Tokio con *cutoff* = 5, el modelo de **Popularidad** sin imputación logra la mayor **precisión** (**0,0540**) y el mejor **F1** (**0,0560**). La **GNN** con imputación destaca en **recall** (**0,0865**). El modelo **Random** lidera en **novedad** (**12,1183**) y **diversidad** (**20.727 ítems** con imputación).

Algoritmo	Precisión (%)	Recall (%)	EPC (%)	Novedad (%)
Random	-44,99	+73,79	+0,002	-0,02
BPRMF	-0,75	-0,59	-0,05	+0,47
BPR+GeoPenalty	-0,58	+0,60	-0,05	+0,41
GNN	+11,29	+6,79	-0,58	-4,89
KNN_usuarios	+116,99	+97,84	-5,51	-33,63
KNN_items	+146,90	+137,91	-7,93	-58,20
Popularidad	-2,09	-2,41	-0,07	-0,34

Tabla 11: Porcentaje de mejora al aplicar imputación para Tokyo con cutoff = 5

La **imputación** mejora notablemente la relevancia en los modelos **KNN**, aumentando la **precisión** en un **+146,90%** y el **recall** en un **+137,91%** para **KNN_items**, pero reduce la **novedad** en un **-58,20%**. La **GNN** experimenta una ligera mejora en **precisión** (**+11,29%**) y **recall** (**+6,79%**), con un impacto menor en **novedad** (**-4,89%**). Los modelos de **Popularidad** y **factorización de matrices** apenas se ven afectados, mostrando cambios mínimos en sus métricas.

4.3.2. Resultados con cutoff de 10

Algoritmo	Imputación	Novedad	EPC	Precisión	Recall	F1	Diversidad
Random	No	12,1183	0,9996	0,00013	0,00010	0,00011	29.525
Random	Sí	12,1161	0,9996	0,00007	0,00018	0,00007	29.530
Popularidad	No	3,7230	0,9160	0,0330	0,1090	0,0440	20
Popularidad	Sí	3,7083	0,9151	0,0331	0,1093	0,0442	20
BPRMF	No	3,7170	0,9150	0,0330	0,1090	0,0440	41
BPRMF	Sí	3,7122	0,9152	0,0330	0,1090	0,0441	79
BPR+GeoPenalty	No	3,7160	0,9150	0,0330	0,1100	0,0440	43
BPR+GeoPenalty	Sí	3,7004	0,9150	0,0330	0,1088	0,0440	43
GNN	No	4,1630	0,9290	0,0300	0,1030	0,0400	20
GNN	Sí	4,1095	0,9265	0,0309	0,1026	0,0413	20
KNN_usuarios	No	8,8530	0,9806	0,0149	0,0475	0,0197	7.791
KNN_usuarios	Sí	6,8327	0,9497	0,0284	0,0814	0,0367	4.190
KNN_items	No	11,5788	0,9909	0,0124	0,0377	0,0161	19.981
KNN_items	Sí	6,1379	0,9363	0,0307	0,0919	0,0404	11.358

Tabla 12: Resultados para Tokio con cutoff = 10

Con $cutoff = 10$, el modelo de **Popularidad** sin imputación mantiene la mayor **precisión (0,0330)**, mientras que **BPR+GeoPenalty** sin imputación alcanza el mejor **recall (0,1100)**. En **F1**, **Popularidad** con imputación logra **0,0442**. El modelo **Random** sobresale en **novedad (12,1183)** y **diversidad (29.530 ítems con imputación)**.

Algoritmo	Precisión (%)	Recall (%)	EPC (%)	Novedad (%)
Random	-46,85	+80,45	+0,0003	-0,02
BPRMF	+0,00	+0,00	+0,02	-0,13
BPR+GeoPenalty	+0,00	-1,09	0,00	-0,42
GNN	+3,00	-0,37	-0,27	-1,28
KNN_usuarios	+90,63	+71,32	-3,15	-22,81
KNN_items	+147,26	+143,94	-5,51	-47,00
Popularidad	+0,30	+0,27	-0,10	-0,39

Tabla 13: Porcentaje de mejora al aplicar imputación para Tokyo con cutoff = 10

La imputación en Tokio con $cutoff = 10$ vuelve a beneficiar claramente a los modelos **KNN**, que logran incrementos notables en **precisión (hasta +147,26%)** y **recall (hasta +143,94%)**, pero a costa de reducciones marcadas en **novedad (-47,00%)** y **EPC (-5,51%)**. La **GNN** muestra un ligero aumento en precisión (+3,00%), con pequeñas pérdidas en novedad. Los modelos de **factorización de matrices** apenas se ven afectados, al igual que **Popularidad**, que presenta un perfil muy estable. El modelo **Random** mantiene nuevamente su independencia frente a los efectos de la imputación.

5. Conclusiones

Este proyecto evaluó algoritmos de recomendación de puntos de interés (POIs) en Londres, Nueva York y Tokio, utilizando datos de Foursquare con alta dispersión (*sparsity*), un desafío clave debido a la escasez de interacciones usuario-ítem. Se analizaron modelos como popularidad, factorización de matrices (BPRMF, BPR+GeoPenalty), GNN, KNN y un recomendador aleatorio (Random), con y sin imputación, para $cutoffs$ de 5 y 10, evaluando relevancia (precisión, recall, F1), exploración (novedad, EPC) y diversidad.

Los modelos de **popularidad** destacaron en relevancia (e.g., precisión de 0,0540 en Tokio, $cutoff=5$), pero con baja diversidad (9-20 ítems). La **factorización de matrices** fue robusta ante *sparsity* y imputación, con precisión sólida (e.g., 0,0330 en Nueva York). La **GNN** equilibró relevancia y exploración, alcanzando un *recall* de 0,0944 (Londres, $cutoff=10$). Los **KNN** lograron alta diversidad (hasta 19.981

ítems) y mejoraron en relevancia con imputación, sacrificando novedad. El **recomendador aleatorio** maximizó novedad (12,1183) y diversidad (29.530 ítems), pero con relevancia mínima.

La **imputación** mejoró notablemente la relevancia en KNN (e.g., +504,55 % en precisión, Londres, *cutoff*=5) y moderadamente en GNN (+21,95 % en precisión, Londres, *cutoff*=10), mientras que popularidad y factorización de matrices apenas variaron. Al aumentar el *cutoff* de 5 a 10, la precisión disminuyó, pero el *recall* y la diversidad crecieron, reflejando el *trade-off* entre cantidad y calidad.

La elección del algoritmo depende del objetivo: relevancia (popularidad, factorización), exploración (aleatorio) o balance (GNN). Futuras investigaciones podrían explorar **algoritmos híbridos** que combinen relevancia y novedad, o **imputación con GANs** para generar interacciones sintéticas de manera inteligente y mitigar la *sparsity*.

6. Trabajo Futuro

Una posible vía de investigación futura consiste en explorar el uso de redes neuronales generativas, conocidas como GANs, para mitigar el problema de dispersión de datos en nuestro sistema de recomendación de POIs. La arquitectura de las GANs se compone de dos componentes esenciales: el **generador** y el **discriminador**. El generador se encarga de producir nuevos datos sintéticos que imiten lo más fielmente posible las características de los datos reales del dataset, mientras que el discriminador evalúa y diferencia entre estos datos sintéticos y los reales. Integrar GANs en el sistema de recomendación podría enriquecer el conjunto de *check-ins* disponibles, reduciendo la escasez de datos (*sparsity*) y permitiendo que los algoritmos aprendan patrones más complejos y relaciones latentes, lo que se traduciría en recomendaciones más precisas y personalizadas.

7. Referencias Bibliográficas

Referencias

- [1] Adomavicius, G., & Tuzhilin, A. (2005). *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. <http://ids.csom.umn.edu/faculty/gedas/papers/recommender-systems-survey-2005.pdf>
- [2] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. https://www.researchgate.net/publication/2369002_Item-based_Collaborative_Filtering_Recommendation_Algorithms
- [3] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). *Deep learning based recommender system: A survey and new perspectives*. ACM Computing Surveys. <https://arxiv.org/pdf/1707.07435>
- [4] Linden, G., Smith, B., & York, J. (2003). *Amazon.com recommendations: Item-to-item collaborative filtering*. <https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf>
- [5] Gomez-Uribe, C. A., & Hunt, N. (2016). *The Netflix recommender system: Algorithms, business value, and innovation*. ACM Transactions on Management Information Systems (TMIS). https://ailab-ua.github.io/courses/resources/netflix_recommender_system_tmis_2015.pdf
- [6] Liu, B., Fu, Y., Yao, Z., & Xiong, H. (2016). *A Survey of Point-of-interest Recommendation in Location-based Social Networks*. arXiv. <https://arxiv.org/abs/1607.00647>
- [7] Islam, M. A., Mohammad, M. M., Das, S. S. S., & Ali, M. E. (2020). *A Survey on Deep Learning Based Point-Of-Interest (POI) Recommendations*. <https://arxiv.org/abs/2011.10187>
- [8] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. <http://ccs.mit.edu/papers/ccswp165.html>
- [9] Ziegler, C.-N. (2008). *The Definition of Novelty in Recommendation System*. ResearchGate. https://www.researchgate.net/publication/288787908_The_Definition_of_Novelty_in_Recommendation_System

- [10] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). *BPR: Bayesian Personalized Ranking from Implicit Feedback*. <https://arxiv.org/pdf/1205.2618>
- [11] Ye, M., Yin, P., Lee, W. C., & Lee, D. L. (2011). *Exploiting geographical influence for collaborative point-of-interest recommendation*. Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. <https://dl.acm.org/doi/10.1145/2009916.2009962>
- [12] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). *The graph neural network model*. <https://papers.baulab.info/papers/Scarselli-2009.pdf>
- [13] Yang, D., Zhang, D., Yu, Z., Wang, Z.-H., & Li, B. (2015). *Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs*. http://www-public.tem-tsp.eu/~zhang_da/pub/TSMC_YANG_2014.pdf
- [14] Noulas, A., Scellato, S., Mascolo, C., & Pontil, M. (2012). *Exploiting semantic annotations for clustering geographic areas and users in location-based social networks*. https://www.researchgate.net/publication/221297888_Exploiting_Semantic_Annotations_for_Clustering_Geographic_Areas_and_Users_in_Location-based_Social_Networks
- [15] Cheng, Z., Caverlee, J., Lee, K., & Sui, D. Z. (2011). *Exploring millions of footprints in location sharing services*. <https://people.engr.tamu.edu/caverlee/pubs/cheng11icwsm.pdf>
- [16] Bao, J., Zheng, Y., & Mokbel, M. F. (2012). *Location-based and preference-aware recommendation using sparse geo-social networking data*. <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/LocationRecommendation.pdf>
- [17] Kang, W. C., & McAuley, J. (2018). *Self-Attentive Sequential Recommendation*. <https://arxiv.org/pdf/1808.09781>
- [18] Shani, G., & Gunawardana, A. (2011). *Evaluating recommendation systems*. In *Recommender Systems Handbook* (pp. 257-297). Springer, Boston, MA.
- [19] Meng, Z., McCreddie, R., Macdonald, C., & Ounis, I. (2020). *Exploring Data Splitting Strategies for the Evaluation of Recommendation Models*. In *Proceedings of the 14th ACM Recommender Systems Conference (RecSys '20)*. ACM, New York, NY, USA, 8 pages. <https://arxiv.org/pdf/2007.13237>
- [20] Vargas, S., & Castells, P. (2011). *Rank and relevance in novelty and diversity metrics for recommender systems*. <https://castells.github.io/papers/recsys2011.pdf>
- [21] Xuan Rao, Lisi Chen.(2022) *Flashback: A Model for POI Recommendation*.https://www.atailab.cn/seminar2022Fall/pdf/2022_KDD_Graph-Flashback%20Network%20for%20Next%20Location%20Recommendation.pdf
- [22] Xiaodong Yan (2023). *STHGCN: A Spatial-Temporal Hypergraph Convolutional Network for Next Point-of-Interest Recommendation*. <https://dl.acm.org/doi/pdf/10.1145/3539618.3591770>
- [23] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). *Generative Adversarial Nets*. Advances in Neural Information Processing Systems. <https://arxiv.org/pdf/1406.2661>
- [24] Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems Handbook*. In *Recommender Systems Handbook* (pp. 1-35). Springer, Boston, MA.
- [25] Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix Factorization Techniques for Recommender Systems*. [https://datajobs.com/data-science-repo/Recommender-Systems-\[Netflix\].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf)
- [26] Yang, D., Zhang, D., Yu, Z., Wang, Z.-H., & Li, B. (2015). *Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs*. http://www-public.tem-tsp.eu/~zhang_da/pub/TSMC_YANG_2014.pdf
- [27] Javaji, S. R., & Sarode, K. (2023). *Hybrid Recommendation System using Graph Neural Network and BERT Embeddings*. arXiv:2310.04878. <https://arxiv.org/pdf/2310.04878>