

Escuela Politécnica Superior

20  
21

# Trabajo fin de máster

Estudio del impacto de aplicar ensembles y ventanas temporales en la recomendación de POIs.



Sergio Navarro Sánchez

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente nº 11



**UNIVERSIDAD AUTÓNOMA DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR**



**Máster en Investigación e Innovación en Inteligencia Computacional y Sistemas Interactivos (MU I2-ICSI).**

## **TRABAJO FIN DE MÁSTER**

**Estudio del impacto de aplicar ensembles y ventanas temporales en la recomendación de POIs.**

**Autor: Sergio Navarro Sánchez**

**Tutor: Alejandro Bellogín Kouki**

**septiembre 2020**

**Todos los derechos reservados.**

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

**DERECHOS RESERVADOS**

© Septiembre de 2020 por UNIVERSIDAD AUTÓNOMA DE MADRID  
Francisco Tomás y Valiente, nº 1  
Madrid, 28049  
Spain

**Sergio Navarro Sánchez**

*Estudio del impacto de aplicar ensembles y ventanas temporales en la recomendación de POIs.*

**Sergio Navarro Sánchez**

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

*A mis padres, mi novia y mi tutor*

*Si estás cambiando el mundo  
estás trabajando en cosas importantes.*

*Larry Page*



# AGRADECIMIENTOS

---

Me gustaría destacar todo el apoyo que me han dado mis padres y mi novia en los momentos difíciles a la hora de realizar este proyecto y a lo largo de la carrera y el máster.

A mi tutor de proyecto que me ha prestado la ayuda necesaria para poder cumplir todos los objetivos previstos.

Sin olvidarme de mis compañeros, que me han acompañado a lo largo de todos estos años y hemos podido compartir buenos y malos momentos.

También destacar a mis profesores que me han aportado los conocimientos necesarios para poder finalizar el proyecto y afianzar una base de los mismos que me servirán a la hora de incorporarme al mundo laboral.

Por último, al resto de mis familiares y amigos por preocuparse y apoyarme en mis estudios, en especial a Poncho y Bruja que siempre ha estado a mi lado en los momentos complicados.





# RESUMEN

---

En la actualidad, los Sistemas de Recomendación han adquirido gran peso en la vida cotidiana de las personas. Cada día aumenta el número de personas que interactúan a través de Internet realizando compras, consultas sobre sus necesidades e intereses, entre otras muchas tareas; debido a esto, se genera una gran cantidad de datos referentes a sus gustos y necesidades. Mediante el uso combinado de esta información y las técnicas de recomendación, se pueden generar recomendaciones personalizadas al usuario e incluso ofrecer recomendaciones basadas en usuarios similares a él.

En este Trabajo de Fin de Máster se implementan estrategias de recomendación basadas en conjuntos de algoritmos (conocidos como *ensembles*), es decir, mediante la combinación de algoritmos sencillos se busca obtener mejores resultados a la hora de generar las recomendaciones; y ventanas temporales para la recomendación de "Puntos de interés", comúnmente conocidos como "POIs", a partir de la serie de POIs visitados con anterioridad. Para estudiar esto, se han empleado algoritmos personalizados y no personalizados, varias metodologías de construcción de ensembles y un conjunto de datos basados en los datos recogidos en redes sociales basadas en check-ins, a partir de los cuales se han generado ventanas temporales para poder simular los principales estados por los que pasa un recomendador a lo largo de su ciclo de vida.

Para lograr el objetivo de nuestro trabajo, se han empleado un conjunto de algoritmos basados en diferentes estrategias de recomendación, así como diversas metodologías de ensemble. Por otro lado, se ha implementado una librería para la combinación de rankings, empleada para generar la recomendación final del ensemble, y se han usado algunas de las métricas más conocidas para poder determinar la calidad de los ensembles y poder discutir los resultados de forma fundamentada.

Como conclusión, se ha podido observar que, dependiendo de la metodología de ensemble empleada, la combinación de algoritmos seleccionados y la longitud de ensembles, los resultados obtenidos son superiores a los obtenidos de manera individual, tal y como se podrá observar a lo largo de esta memoria. Otro punto importante ha sido la selección del tamaño de la ventana temporal ya que, como más adelante podremos comprobar, cuanta más información posea el ensemble del usuario, mejores serán las recomendaciones.

# PALABRAS CLAVE

---

Sistema de recomendación, Punto de Interés, POI, ensemble, check-in, ventana temporal, evaluación temporal



# ABSTRACT

---

Nowadays, Recommender Systems have acquired a great weight in people's daily lives. Every day the number of people who interact through the Internet by making purchases or inquiries about their needs and interests, among many other tasks, increase steadily. Because to this, a large amount of data is generated regarding the tastes and needs of users. Through the combined use of this information and recommendation techniques, personalized recommendations can be generated for the user, even offering recommendations based on similar users to the current user.

In this Master's Thesis, recommendation strategies based on sets of algorithms (known as *ensembles*) are implemented, that is, by combining simple algorithms we seek to obtain better results when generating recommendations. We study this, together with temporal windows for the recommendation of "Points of interest"(commonly known as "POIs") from the sequence of POIs previously visited. For this, personalized and not personalized algorithms have been used, various ensemble construction methodologies, and datasets collected from social networks based on check-ins, from which temporary windows have been generated to be able to simulate the main states that a recommender goes through throughout its life cycle.

To achieve the objective of our work, a set of algorithms based on different recommendation strategies, as well as various ensemble methodologies, have been used. On the other hand, a library for the combination of rankings has been implemented, and used to generate the final recommendations of each ensemble; moreover, some of the best-known evaluation metrics have been used to determine the quality of the ensembles and to be able to discuss the results in an informed manner.

In conclusion, it has been observed that, depending on the ensemble methodology used, the combination of selected algorithms, and the length of ensembles, the results obtained are superior to those obtained individually, as can be observed throughout this report. Another important point has been the selection of the size of the time window, since, as we will later see, the more information the user ensemble has, the better the recommendations will be.

# KEYWORDS

---

Recommendation system, Point of Interest, POI, ensemble, check-in, temporal window, temporal evaluation



# ÍNDICE

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación .....	1
1.2	Objetivos .....	2
1.3	Organización de la memoria .....	2
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Sistemas de recomendación .....	5
2.1.1	Algoritmos no personalizados .....	5
2.1.2	Algoritmos de filtrado colaborativo .....	6
2.1.3	Algoritmos geográficos .....	10
2.1.4	Algoritmos híbridos .....	10
2.2	Ensembles .....	11
2.2.1	Votación .....	11
2.2.2	Bagging .....	12
2.2.3	Boosting .....	12
2.2.4	Stacking .....	12
2.3	Trabajos actuales .....	14
2.3.1	Diseño de un recomendador musical híbrido mediante el uso de ensembles .....	14
2.3.2	El método de recomendación personalizada con combinación de conjuntos .....	15
2.3.3	Predicción de la influencia social basada en el dominio consciente del tiempo ...	15
2.3.4	Discusión .....	16
<b>3</b>	<b>Diseño y desarrollo</b>	<b>17</b>
3.1	Descripción del entorno de trabajo .....	17
3.2	Descripción de los conjuntos de datos .....	18
3.2.1	Proceso de generación de los datos .....	18
3.2.2	Descripción de los ficheros .....	19
3.2.3	Descripción de las ventanas temporales .....	20
3.3	Proceso de generación del ensemble .....	20
3.3.1	Diagrama de flujo del ensemble .....	20
3.3.2	Parametrización de los elementos del ensemble .....	20
3.4	Estrategias para combinar rankings .....	23
3.5	Estrategias de evaluación .....	24
3.5.1	Precision@k .....	24

3.5.2 Recall@k .....	24
3.5.3 Mean average precision mAP@k .....	24
3.5.4 Normalized discounted cumulative gain nDCG@k .....	25
<b>4 Pruebas y resultados</b>	<b>27</b>
4.1 Estudio de las diferencias en los conjuntos de datos .....	27
4.2 Resultados obtenidos con los baselines .....	28
4.3 Resultados obtenidos con ensembles basados en votación .....	33
4.4 Resultados obtenidos con ensembles basados en bagging .....	38
4.5 Discusión de todos los resultados .....	42
<b>5 Conclusiones y trabajo futuro</b>	<b>45</b>
5.1 Conclusiones .....	45
5.2 Trabajo futuro .....	46
<b>Bibliografía</b>	<b>49</b>
<b>Acrónimos</b>	<b>51</b>
<b>Apéndices</b>	<b>53</b>
<b>A Representación gráfica de los datasets basada en las ventanas temporales.</b>	<b>55</b>
<b>B Representación gráfica de los baselines.</b>	<b>57</b>
B.1 Foursquare .....	57
B.2 Gowalla .....	61
B.3 Yelp .....	65

# LISTAS

---

## Lista de ecuaciones

2.1	Cálculo del estimador $\hat{u}$ , algoritmo random	6
2.2	Cálculo del estimador $\sigma^2$ , algoritmo random	6
2.3	Ley de potencias	6
2.4	Probabilidad de que el usuario visite el POI (UCF)	7
2.5	Similitud coseno.	8
2.6	Probabilidad de que el usuario visite el POI (FCF)	8
2.7	Similitud de usuario basados en amigos (FCF)	8
2.8	SVD cálculo del rating.	9
2.9	SVD cálculo del error mínimo.	9
2.10	MGMPFM cálculo de PFM.	9
2.11	MGMPFM cálculo de MGM.	10
2.12	Probabilidad de check-in empleando (USG)	10
2.13	MGMPFM cálculo de MGMPFM.	11
3.1	Precision@k.	24
3.2	Recall@k.	24
3.3	mAP@k.	24
3.4	CG@k.	25
3.5	DCG@k.	25
3.6	iDCG@k.	25
3.7	nDCG@k.	26

## Lista de figuras

2.1	Esquema básico de un sistema de recomendación	5
2.2	Matrices filtrado colaborativo basado en usuario	7
2.3	Esquema básico de un ensemble	12
2.4	Ensembles, diferencias entre votación y bagging.	13
2.5	Ensembles, ejemplo de boosting.	13
2.6	Ensembles, ejemplo de stacking.	14
3.1	Generación de ensembles, diagrama de flujo.	21

4.1	Resultados de los baselines de Foursquare empleando nDCG y longitud de ranking 5 .	30
4.2	Resultados de los baselines de Gowalla empleando nDCG y longitud de ranking 5 . . .	31
4.3	Resultados de los baselines de Yelp empleando nDCG y longitud de ranking 5 . . . . .	33
A.1	Datos de entrenamiento simulado un estado nuevo . . . . .	55
A.2	Datos de entrenamiento simulado un estado avanzado . . . . .	55
A.3	Datos de entrenamiento simulado un estado final . . . . .	56
A.4	Datos de test . . . . .	56
B.1	Baselines de Foursquare en estado nuevo evaluados con Precision . . . . .	57
B.2	Baselines de Foursquare en estado nuevos evaluados con Recall . . . . .	57
B.3	Baselines de Foursquare en estado nuevo evaluados con mAP . . . . .	58
B.4	Baselines de Foursquare en estado nuevo evaluados con nDCG . . . . .	58
B.5	Baselines de Foursquare en estado avanzado evaluados con Precision . . . . .	58
B.6	Baselines de Foursquare en estado avanzado evaluados con Recall . . . . .	59
B.7	Baselines de Foursquare en estado avanzado evaluados con mAP . . . . .	59
B.8	Baselines de Foursquare en estado avanzado evaluados con nDCG . . . . .	59
B.9	Baselines de Foursquare en estado final evaluados con Precision . . . . .	60
B.10	Baselines de Foursquare en estado final evaluados con Recall . . . . .	60
B.11	Baselines de Foursquare en estado final evaluados con mAP . . . . .	60
B.12	Baselines de Foursquare en estado final evaluados con nDCG . . . . .	61
B.13	Baselines de Gowalla en estado nuevo evaluados con Precision . . . . .	61
B.14	Baselines de Gowalla en estado nuevos evaluados con Recall . . . . .	61
B.15	Baselines de Gowalla en estado nuevo evaluados con mAP . . . . .	62
B.16	Baselines de Gowalla en estado nuevo evaluados con nDCG . . . . .	62
B.17	Baselines de Gowalla en estado avanzado evaluados con Precision . . . . .	62
B.18	Baselines de Gowalla en estado avanzados evaluados con Recall . . . . .	63
B.19	Baselines de Gowalla en estado avanzado evaluados con mAP . . . . .	63
B.20	Baselines de Gowalla en estado avanzado evaluados con nDCG . . . . .	63
B.21	Baselines de Gowalla en estado final evaluados con Precision . . . . .	64
B.22	Baselines de Gowalla en estado final evaluados con Recall . . . . .	64
B.23	Baselines de Gowalla en estado final evaluados con mAP . . . . .	64
B.24	Baselines de Gowalla en estado final evaluados con nDCG . . . . .	65
B.25	Baselines de Yelp en estado nuevo evaluados con Precision . . . . .	65
B.26	Baselines de Yelp en estado nuevos evaluados con Recall . . . . .	65
B.27	Baselines de Yelp en estado nuevo evaluados con mAP . . . . .	66
B.28	Baselines de Yelp en estado nuevo evaluados con nDCG . . . . .	66
B.29	Baselines de Yelp en estado avanzado evaluados con Precision . . . . .	66
B.30	Baselines de Yelp en estado avanzados evaluados con Recall . . . . .	67



B.31	Baselines de Yelp en estado avanzado evaluados con mAP	67
B.32	Baselines de Yelp en estado avanzado evaluados con nDCG	67
B.33	Baselines de Yelp en estado final evaluados con Precision	68
B.34	Baselines de Yelp en estado final evaluados con Recall	68
B.35	Baselines de Yelp en estado final evaluados con mAP	68
B.36	Baselines de Yelp en estado final evaluados con nDCG	69

## Lista de tablas

3.1	Tabla entorno de trabajo	17
3.2	Tabla descripción de los conjuntos de datos antes de ser procesados	18
3.3	Tabla descripción de los conjuntos de datos antes de ser procesados	19
3.4	Tabla descripción individual de los parámetros de los modelos	23
4.1	Tabla descripción de los datasets por estado del ensemble	27
4.2	Tabla descripción de las abreviaturas de los algoritmos	28
4.3	Descripción de los mejores resultados con el dataset de Foursquare	29
4.4	Descripción de los mejores resultados con el dataset de Gowalla	30
4.5	Descripción de los mejores resultados con el dataset de Yelp	32
4.6	Resultados ensemble votación de Foursquare en ventana nueva	34
4.7	Resultados ensemble votación de Foursquare en ventana avanzada	34
4.8	Resultados ensemble votación de Foursquare en ventana final	35
4.9	Resultados ensemble votación de Gowalla en ventana nueva	35
4.10	Resultados ensemble votación de Gowalla en ventana avanzada	36
4.11	Resultados ensemble votación de Gowalla en ventana final	36
4.12	Resultados ensemble votación de Yelp en ventana nueva	36
4.13	Resultados ensemble votación de Yelp en ventana avanzada	37
4.14	Resultados ensemble votación de Yelp en ventana final	37
4.15	Resultados ensemble bagging de Foursquare en ventana nueva	39
4.16	Resultados ensemble bagging de Foursquare en ventana avanzada	39
4.17	Resultados ensemble bagging de Foursquare en ventana final	39
4.18	Resultados ensemble bagging de Gowalla en ventana nueva	40
4.19	Resultados ensemble bagging de Gowalla en ventana avanzada	40
4.20	Resultados ensemble bagging de Gowalla en ventana final	41
4.21	Resultados ensemble bagging de Yelp en ventana nueva	41
4.22	Resultados ensemble bagging de Yelp en ventana avanzada	41
4.23	Resultados ensemble bagging de Yelp en ventana final	42



# INTRODUCCIÓN

---

## 1.1. Motivación

La interacción entre los usuarios, redes sociales y plataformas de servicios ha ido aumentando cada vez más, esto genera gran cantidad de datos referentes a estas interacciones. Gracias a esa recopilación de datos, podemos conocer en detalle los gustos y preferencias de los usuarios. Es en este contexto donde cobran gran relevancia los sistemas de recomendación, ya que a partir de estos datos podemos ofrecer servicios o productos de primera necesidad, e incluso novedosos para el usuario.

Ya que a la mayoría de personas les gusta viajar y conocer lugares nuevos, la temática de este trabajo va a estar centrada en el sector turístico. Por este motivo, se ha decidido investigar en el campo de los sistemas de recomendación enfocados a turismo, en concreto, para ofrecer un enfoque nuevo y explorar sus posibilidades en este contexto, vamos a estudiar los recomendadores basados en ensembles y ventanas temporales.

El planteamiento que ofrece este TFM es ofrecer un enfoque actual y original sobre el empleo de forma conjunta de redes sociales, ensembles y puntos de interés, analizando las interacciones de los usuarios con las redes sociales Yelp, Gowalla y Foursquare. Mediante el estudio de los datos procedentes de estas redes sociales se busca poder generar recomendaciones satisfactorias en función de los datos conocidos sobre el propio usuario y similares a este, a la par de intentar introducir técnicas novedosas y provechosas en el campo de la recomendación basada en el turismo.

Para finalizar, destacar que la elección del tema ha sido muy gratificante para mí, ya que el trabajo con sistemas de recomendación, y en concreto sobre turismo, y que el trabajo tenga una posible utilidad en el mercado, ha hecho que su desarrollo sea muy interesante. Desde el punto de vista académico, el desarrollo de este TFM me ha permitido adquirir y profundizar en los estudios realizados en el Máster y el Grado además de mejorar mi nivel a la hora de afrontar nuevos proyectos y dificultades.

## 1.2. Objetivos

El objetivo principal del proyecto es recomendar Puntos de interés (*POIs*), que son ubicaciones que los usuarios pueden encontrar útiles o interesantes. Para ello se combinan distintos algoritmos de recomendación mediante la implementación de *ensembles* y ventanas temporales.

Los hitos que se deben cumplir para alcanzar el objetivo final son los siguientes:

### -Datos:

Obtener datos de diferentes fuentes para obtener resultados más globales.

Adaptar los datos a las especificaciones de la literatura.

Emplear distintos tamaños de ventanas temporales.

### -Ensembles:

Emplear diferentes metodologías.

Utilizar diferentes tamaños de conjuntos de modelos de recomendación.

Usar distintos tamaños de ventanas temporales.

Emplear distintos números de algoritmos de recomendación.

### -Algoritmos de recomendación:

Implementar un número considerable de modelos.

Emplear modelos basados en diferentes técnicas de recomendación.

Parametrización de los modelos en función a los datos.

Diseñar diferentes métodos de combinación (*merge*) para obtener la recomendación final, a partir de las generadas de manera individual por cada algoritmo del conjunto.

### -Pruebas y resultados:

Implementar diversas métricas de calidad.

Realizar medidas de eficiencia.

Estudio de los datos obtenidos mediante tablas y gráficas.

Extraer conclusiones de los experimentos realizados.

## 1.3. Organización de la memoria

La memoria se compone de las siguientes secciones, exponiéndose los pasos que se han realizado hasta llegar al desarrollo final del proyecto.

**Estado del arte:** en la primera sección se documenta el trabajo de investigación realizado para la toma de decisiones: se han tenido en cuenta librerías y frameworks de recomendación ya existentes para disponer de un amplio abanico de algoritmos y así poder obtener una idea inicial sobre cómo abordar el diseño y la implementación de los ensembles; el lenguaje de programación y módulos que mejor se adapten a la idea y objetivos que se quieren implementar.

**Diseño y desarrollo:** esta sección contiene la documentación básica sobre el proyecto. En ésta se describen los pilares en los que se basa el proyecto: software adaptado al hardware disponible; objetivos a cumplir; conjuntos de datos a emplear; decisiones tomadas para la implementación de los ensembles y determinar el procedimiento de medida de calidad de los ensembles.

**Pruebas y resultados:** en este bloque se detallan los experimentos realizados y se analiza si los resultados generados son mejores que los datos obtenidos de los baselines y si la implementación ha alcanzado los objetivos establecidos.

**Conclusiones y trabajo futuro:** en esta parte del trabajo se detallan posibles mejoras del proyecto, optimizando su funcionamiento o aportando nuevas funcionalidades para asegurar la vida del mismo.

**Bibliografía:** para finalizar se enumera la literatura empleada en el desarrollo del proyecto.



## ESTADO DEL ARTE

---

En este apartado se abordan algunos algoritmos relevantes en la recomendación de puntos de interés, además de las principales técnicas empleadas para la construcción de los ensembles.

### 2.1. Sistemas de recomendación

Este subapartado engloba la explicación de los algoritmos de recomendación empleados en el proyecto para la recomendación de los POIs. Un sistema de recomendación es un conjunto de algoritmos encargados de filtrar y analizar gran cantidad de información con el fin de realizar recomendaciones personalizadas para cada usuario. Para generar las recomendaciones, los sistemas se basan en el análisis y filtrado de los datos conocidos de los usuarios e items, para, posteriormente, procesarlos y generar recomendaciones que satisfagan las necesidades del usuario [1].

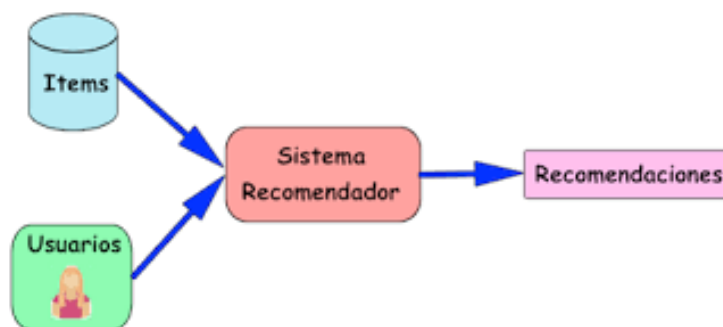


Figura 2.1: Esquema básico de un sistema de recomendación [1]

#### 2.1.1. Algoritmos no personalizados

Estos algoritmos son los más sencillos de implementar, su principal característica es que no tienen en cuenta las preferencias de los usuarios, por lo que a menudo se emplean cuando no se dispone de información del usuario a recomendar [2].

## Aleatorios

Algoritmo que predice un ranking aleatorio basado en la distribución de los datos de entrenamiento. La recomendación  $\hat{r}_{ui}$  se generará a partir de la  $\mathcal{N}(\mu, \sigma^2)$  y donde  $\hat{u}$  y  $\hat{\sigma}$  están estimados mediante MLE (Maximum Likelihood Estimation) <sup>1</sup>:

$$\hat{u} = \frac{1}{R_{Train}} * \sum_{r_{ui} \in R_{Train}} r_{ui} \quad (2.1)$$

$$\sigma^2 = \sqrt{\sum_{r_{ui} \in R_{Train}} \frac{(r_{ui} - \hat{u})^2}{|R_{Train}|}} \quad (2.2)$$

## Basado en popularidad

Es uno de los modelos más simples, independientemente del usuario al que se recomienda siempre se devuelven los N items más populares pertenecientes al conjunto de entrenamiento, donde N es la longitud del ranking de recomendaciones.

## Basado en la Ley de potencias

Dicho algoritmo de basa en la influencia geográfica de los POIs visitados mediante el uso de la Ley de potencias. La importancia de esta influencia se debe a que se ha demostrado que la probabilidad de que un usuario haga check-in en otro POI aumenta notablemente cuanto más cerca esté del último check-in realizado. Dicha ley nos permite relacionar el número de visitas de cada POI con la cercanía de unos a otros, como se puede apreciar en la siguiente fórmula:

$$M = Cm^p \quad (2.3)$$

donde M es la afluencia de visitas de los POIs, m determina la cercanía, mientras que C y p son dos números reales que se emplean para normalizar las dos magnitudes [4].

### 2.1.2. Algoritmos de filtrado colaborativo

El filtrado colaborativo (CF) es una técnica que emplean los sistemas de recomendación para realizar el proceso de filtrado y evaluación de ítems para un determinado usuario mediante las opiniones

<sup>1</sup> Método habitual para ajustar un modelo y estimar sus parámetros [3].

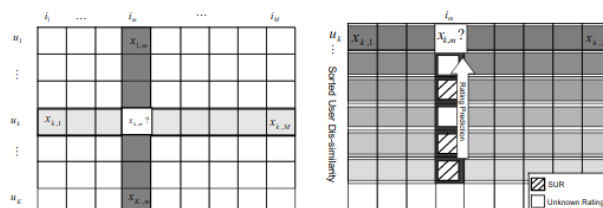


ofrecidas por otros usuarios. Su funcionamiento se basa en emplear la información de valoración de los usuarios para generar las predicciones en el momento que se produce una interacción usuario-ítem, permitiendo reducir el número de ítems a recomendar a cada usuario [5].

### Basado en usuario

También conocidos como algoritmos de vecinos cercanos. Su funcionamiento se basa en emplear de manera conjunta toda la base de datos y técnicas de probabilidad para estudiar las similitudes de todos los usuarios entre sí (cálculo de los vecinos más próximos) y, por otro lado, para cada usuario se combinan las preferencias de sus  $N$  vecinos más próximos, para predecir las preferencias sobre los ítems que no se posee información referente a dicho usuario [6]. En el área se han propuesto versiones donde los vecinos son otros usuarios u otros ítems, en esta sección nos centraremos en los primeros.

La representación de la información relacionada con los usuarios similares a uno dado se realiza mediante una matriz de similitud, donde cada fila representa la información de los usuarios y las columnas la información de los ítems [7, 8].



**Figura 2.2:** Matrices filtrado colaborativo basado en usuario [9].

Enfocándonos en el ámbito del turismo, se puede denotar a  $U$  como el conjunto de usuarios y  $L$  como el conjunto de POIs en un sistema basado en check-ins de usuarios que contiene el registro de las actividades del sistema. Dada la actividad de registro de un usuario  $u_i \in U$  tiene en un POI  $l_j \in L$ , se denota como  $c_{i,j}$  donde  $c_{i,j} = 1$  si  $u_i$  ha realizado un check-in en  $l_j$  previamente, en caso contrario  $c_{i,j} = 0$ . Por lo tanto, estos registros de usuarios se emplean para determinar la preferencia implícita de un usuario, representada por una probabilidad de que el usuario desee visitar ese POI, esta predicción se denota como [9]:

$$\hat{c}_{i,j} = \frac{\sum_{u_k} w_{i,k} \cdot c_{k,j}}{\sum_{u_k} w_{i,k}} \quad (2.4)$$

donde  $w_{i,k}$  denota la similitud entre los usuarios  $u_i$  y  $u_k$ . Para calcular dicho valor se ha empleado la similitud coseno <sup>2</sup> [9].

$$w_{i,k} = \frac{\sum_{l_j \in L} c_{i,j} \cdot c_{k,j}}{\sqrt{\sum_{l_j \in L} c_{i,j}^2} \cdot \sqrt{\sum_{l_j \in L} c_{k,j}^2}} \quad (2.5)$$

## Basado en amistad

Diversos estudios han demostrado que dos personas que son amigas tienden a poseer un comportamiento similar debido a que comparten un gran número de intereses entre sí, lo que produce el efecto de comportamientos correlacionados [11]. Tomando el supuesto que dos amigos pasan un día juntos, estos realizarán las mismas actividades, comerán en el mismo lugar e incluso pueden ir a alguna localización recomendada por un tercer amigo, que puede ser amigo de ambos o solo de uno de ellos. Por lo que es un argumento con suficiente peso para poder ofrecer recomendaciones en base los amigos de un determinado usuario, es decir, tomamos la formulación del apartado anterior pero en lugar de calcular los usuarios más cercanos, usamos sus amigos:

$$\widehat{c}_{i,j} = \frac{\sum_{u_k \in F_i} SI_{k,i} \cdot c_{k,j}}{\sum_{u_k \in F_i} SI_{k,i}} \quad (2.6)$$

dónde  $\widehat{c}_{i,j}$  es la probabilidad de que el usuario  $u_i$  visite el POI  $l_j$  y  $F_i$  es un amigo perteneciente al conjunto de amigos de  $u_i$ , y  $SI_{k,i}$  es la influencia sobre  $u_i$  [9].

A la hora de determinar cuáles son los mejores amigos en los que basarnos, se tiene en cuenta la cercanía dentro de la red social y que los comportamientos sean lo más similares posibles, como se puede apreciar en la siguiente formulación:

$$SI_{k,i} = \eta \cdot \frac{|F_k \cap F_i|}{|F_k \cup F_i|} + (1 - \eta) \cdot \frac{|L_k \cap L_i|}{|L_k \cup L_i|} \quad (2.7)$$

dónde  $\eta$  es un parámetro de modelización con un rango entre  $[0-1]$  y  $F_k$  and  $L_k$  denota el conjunto de amigos y POIs de usuario  $u_k$ . Teniendo en cuenta el desarrollo del algoritmo, cada vez que nos refiramos a éste nos referiremos a él como FCF.

<sup>2</sup>Medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar. Cualquier ángulo existente entre los vectores, el coseno arrojaría un valor inferior a uno [10].

### Basado en la descomposición de valores singulares

Algoritmo que emplea la técnica de factorización de matrices para reducir el número de atributos del dataset, transformando un espacio de tamaño  $N$  a otro con tamaño  $K$ , donde  $K \ll N$ . Al emplear SVD en el campo de la recomendación, lo interesante no es tanto reducir la dimensionalidad, si no la búsqueda de dos matrices cuyo producto nos devuelva la matriz inicial de los datos, o una aproximación muy cercana. Cada ítem se puede representar como un vector  $q_i$ , por lo que cada usuario se puede representar como un vector denominado  $p_u$  tal que el producto escalar de esos dos vectores sea el valor de la celda correspondiente en la matriz original.

$$\hat{r}_{ui} = q_i^T p_u \quad (2.8)$$

donde  $q_i$  y  $p_u$  denotan el punto que se aproxima al rating ideal  $\hat{r}_{ui}$ , cuando la diferencia del error cuadrático de su producto escalar y el rating conocido es mínimo, por lo que se obtiene:

$$\min(p, q, b_i, b_u) = \sum_{(u,i) \in K} (r_{u,i} - q_i^T \cdot p_u - \mu - b_i - b_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2) \quad (2.9)$$

donde  $\lambda(\|q_i\|^2 + \|p_u\|^2)$  evita el sobreaprendizaje mediante el uso de una constante de penalización  $\lambda$  y  $b_i^2 + b_u^2$  hacen las veces de sesgos, calculados como el promedio de visitas del ítem  $i$ -ésimo y el número promedio de ítems visitados por el usuario respectivamente [12].

### Basado en un modelo probabilístico de factorización de matrices

El modelo probabilístico de factorización de matrices es uno de los modelos más empleados en filtrado colaborativo, asume que existe una distribución gaussiana en el ruido generado por los datos observados y emplea valores de esta distribución a la hora de generar las matrices de factores latentes <sup>3</sup>  $U$  y  $V$ . Su desarrollo matemático es el siguiente:

$$\min_{U, L} \sum_{i=1}^{|U|} \sum_{j=1}^{|L|} I_{i,j} (g(F_{i,j}) - g(U_i^T L_j))^2 + \lambda_1 \|U\|_F^2 + \lambda_2 \|L\|_F^2 \quad (2.10)$$

donde  $\|\cdot\|_F$  denota la norma Frobenius,  $I_{i,j}$  es el indicador de check-in por parte del usuario  $u$  y su valor es 0 si no realiza el check-in o 1 en caso contrario y  $g(x)$  es la función logística.

<sup>3</sup>Es la representación matricial de las características de los usuarios y los POIs [13].

### 2.1.3. Algoritmos geográficos

Son aquellos algoritmos que se rigen por la localización de los POIs, al igual que en el algoritmo de Ley de potencias descrito en la Sección 2.1.1, se basan en la afirmación de que la probabilidad de que un POI sea visitado aumenta cuanto más cerca esté del último POI que se hizo check-in.

#### Recomendación basada en modelos Gaussianos multicéntricos

Dicho algoritmo de basa en el uso de una distribución gaussiana, esta se basa en dos principios mediante el uso de la localización de los POIs, el primero dice que los check-ins tienden a agruparse en varios centros de actividad. La segunda afirma que la probabilidad de check-in es inversamente proporcional a la distancia del siguiente POI. Por lo que la probabilidad que el usuario  $u$  visite el POI  $l$  dado el conjunto  $C_u$  se define como:

$$P(l|C_u) = \sum_{C_u=1}^{|C_u|} P(l \in C_u) \frac{f_{C_u}^\alpha}{\sum_{i \in C_u} f_{C_i}^\alpha} \frac{N(l|\mu_{C_u}, \Sigma_{C_u})}{\sum_{i \in C_u} N(l|\mu_{C_i}, \Sigma_{C_i})} \quad (2.11)$$

donde  $l$  define la latitud y longitud del POI,  $C_u$  es un conjunto de centros para el usuario,  $P(l \in C_u)$  es la probabilidad de que la ubicación pertenezca al centro  $C_u$ ,  $f_{C_u}$  denota la frecuencia de de registro en la ubicación y  $N(l|\mu_{C_i}, \Sigma_{C_i})$  es la probabilidad normalizada de una ubicación perteneciente al centro.

### 2.1.4. Algoritmos híbridos

Es una técnica que consiste en combinar algoritmos de recomendación sencillos para generar otro que implemente las fortalezas y elimine las debilidades de cada uno de ellos. Gracias a esta técnica podemos generar recomendaciones con un conjunto de información más amplio, permitiendo conocer mejor a los usuarios e ítems [14].

#### Recomendación basada en preferencias de usuario e influencias sociales y geográficas

Para la implementación de este algoritmo se han empleado de forma conjunta los algoritmos de la Ley de potencias (ver sección 2.1.1), filtrado colaborativo basado en usuario (ver sección 2.1.2) y filtrado colaborativo basado en amistad (ver sección 2.1.2) desarrollados anteriormente.

Para generar la fusión de los tres algoritmos,  $S_{i,j}$  indica la probabilidad de que el usuario  $u_i$  situado en el POI  $l_j$  haga check-in en dicho POI. Disponiendo de  $S_{i,j}^u$ ,  $S_{i,j}^s$  y  $S_{i,j}^g$ , que denotan las probabilidades de manera independiente de que el usuario haga el check-in del POI, podemos desarrollar la siguiente fórmula:

$$S_{i,j} = (1 - \alpha - \beta)S_{i,j}^u + \alpha S_{i,j}^s + \beta S_{i,j}^g \quad (2.12)$$

donde  $\alpha$  y  $\beta$  son dos parámetros que regulan el peso del aporte de cada algoritmo y su valor oscila entre  $(0 \leq \alpha + \beta \leq 1)$  [9].

### Recomendación basada en modelos Gaussianos y probabilísticos.

Algoritmo que sigue la filosofía de los anteriores, este está formado por un modelo de filtrado colaborativo basado en un modelo probabilístico con factorización de matrices (ver sección 2.1.2) y un modelo gaussiano multicéntrico (ver sección 2.1.3). Dicho modelo está fuertemente sesgado por el principio de cercanía de los POIs, ya que sus dos principios son que los POIs tienden a acumularse en conjuntos circulares y que la probabilidad de que un POI sea visitado aumenta cuanto más cerca este el último visitado [15]. Para poder obtener la probabilidad conjunta se realiza el siguiente cálculo:

$$P_{u,l} = P(F_{u,l}) \cdot P(l|C_u) \quad (2.13)$$

donde  $P(F_{u,l})$  es la probabilidad correspondiente al modelo probabilístico (ver fórmula 2.10) y  $P(l|C_u)$  es la probabilidad calculada mediante el modelo gaussiano (ver fórmula 2.11).

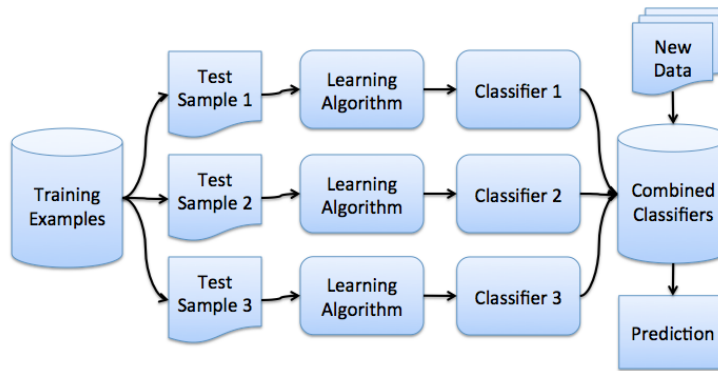
## 2.2. Ensembles

Un *ensemble* se puede definir como un conjunto de modelos de machine learning donde cada modelo genera su propia predicción y cada una de éstas se combina para obtener una única predicción final. La principal ventaja que ofrece esta técnica es que la mayoría de las veces los errores individuales de los modelos tienden a compensarse, disminuyendo el error de generalización [16, 17].

### 2.2.1. Votación

Consiste en realizar el entrenamiento de N modelos con los mismos datos de entrenamiento y de test. Cuando se obtengan nuevos datos, cada uno de los modelos generará su predicción. En nuestro caso, el siguiente POI a visitar. En caso de empate, se pueden aplicar las siguientes técnicas para deshacerlo: seleccionar un POI de manera aleatoria; predecir el más popular; retornar el más cercano; entre otras [19]. Al aplicar esta técnica hay que tener en cuenta una serie de factores para generar su correcto funcionamiento:

- Si se usa el mismo modelo, entrenar cada uno de ellos con diferentes datos, si no todos generarán la misma predicción.



**Figura 2.3:** Esquema básico de un ensemble [18]

- Si son modelos diferentes, emplear los mismos datos de entrenamiento.
- Se puede utilizar el soft-voting <sup>4</sup>.

## 2.2.2. Bagging

Técnica que consiste en la combinación de algoritmos del mismo tipo con el fin de reducir la varianza y sobreaprendizaje del modelo. Para ello, a partir del conjunto de datos de entrenamiento se generan  $N$  subconjuntos de idéntico tamaño seleccionados aleatoriamente para entrenar  $M$  modelos, con  $N = M$ . Para generar las predicciones se toman las predicciones individuales de cada modelo y se combinan para generar la predicción conjunta mediante los criterios citados en el apartado anterior [19].

## 2.2.3. Boosting

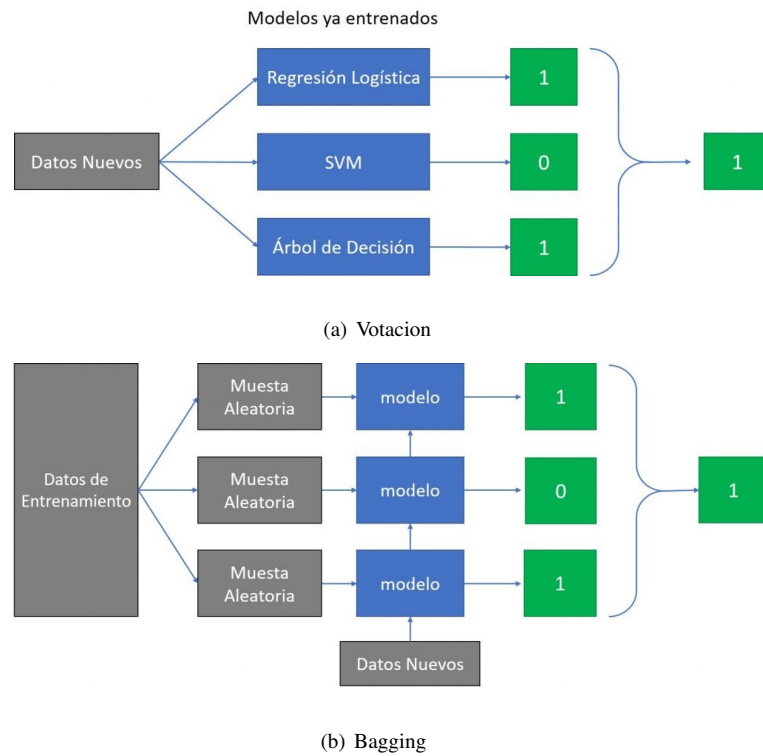
Su principal característica se centra en que cada modelo intenta corregir el error de su antecesor. El primer modelo realizará las predicciones en función de los datos de entrada generando una serie de errores, el siguiente modelo dará mayor peso a las predicciones erróneas y menor peso a las acertadas, para intentar reducir la tasa de error [19].

## 2.2.4. Stacking

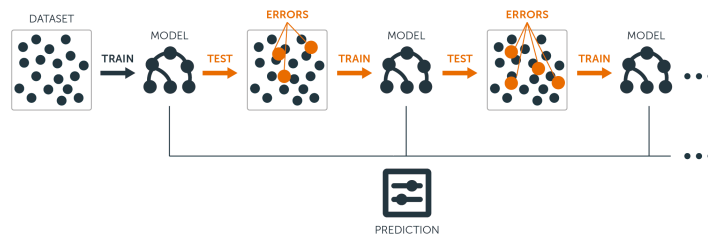
Técnica que consiste en generar una pila de modelos, donde la salida de un modelo es la entrada de uno o más modelos [19]. Las principales características a tener en cuenta de esta técnica son:

- Los modelos empleados al inicio se denominan modelos débiles.

<sup>4</sup>Técnica que introduce pesos dentro de la votación. El peso de cada modelo se genera en función de sus resultados [20].



**Figura 2.4:** Ensembles, diferencias entre votación y bagging [21].



**Figura 2.5:** Ensembles, ejemplo de boosting [22].

- Los modelos empleados al final se denominan modelos fuertes.
- El conjunto de todos los modelos se llama meta-modelo.
- Si el meta-modelo posee  $N$  modelos débiles es recomendable realizar  $N$  splits sobre el conjunto de entrenamiento para evitar el sobreaprendizaje de los datos.

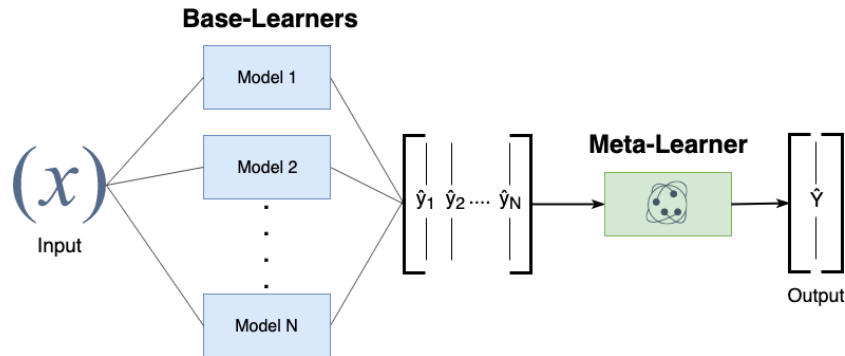


Figura 2.6: Ensembles, ejemplo de stacking [23].

## 2.3. Trabajos actuales

Tras filtrar la literatura para la realización de este trabajo, los proyectos que describimos a continuación son algunos de los que nos han parecido interesantes. Lo destacable de este apartado es que nos ha ofrecido la información necesaria para dotar a nuestro trabajo de un carácter innovador.

### 2.3.1. Diseño de un recomendador musical híbrido mediante el uso de ensembles

En este proyecto los autores desarrollan un recomendador centrado en el ámbito musical. Su trabajo consiste en predecir la puntuación con un rango de  $[0 - 10]$  de todas las canciones pertenecientes a todos los usuarios que aún no han sido valoradas.

Para dicho propósito emplean la técnica de generación de ensembles bagging (ver sección 2.2.2) junto con el algoritmo base filtrado colaborativo basado en ítem. Este algoritmo es similar al filtrado colaborativo basado en usuario (ver sección 2.1.2), con la diferencia que las recomendaciones se realizan sobre las valoraciones realizadas por otros usuarios y a la hora de calcular la similitud se emplean los ítems no los usuarios.

Como conclusión los autores determinaron que *existe un grado de diversidad entre las recomendaciones sociales y las basada en contenido*, ya que los resultados de su ensemble fueron un 10% peor que las que generaba el algoritmo base [24].



### 2.3.2. El método de recomendación personalizada con combinación de conjuntos

Mediante este trabajo, los autores han implementado un recomendador de ítems basado en la venta online de productos. Cada producto posee una valoración de  $[0 - 5]$ , repartida en tres categorías:  $[0 - 2]$  referencia a malo,  $[3 - 4]$  implica producto aceptable y  $[5]$  determina que el producto es inmejorable.

La técnica de ensemble que han utilizado es la de votación (ver sección 2.2.1) junto con los algoritmos de filtrado colaborativo:

**Euclídeo:** a la hora de calcular las similitudes se emplea la distancia euclídea.

**Coseno:** para calcular la distancia se emplea la similitud coseno.

**Relaciones sociales:** en el momento de calcular la distancia se fundamenta en la valoración de confianza, reputación y amistad de cada usuario.

Para finalizar, la conclusión que extraen los autores es que el uso de ensembles ha aumentado la precisión de las recomendaciones, aunque de manera leve (en torno a un 0,02). Por otro lado determinaron que para emplear algoritmos basados en relaciones sociales es necesario adquirir gran cantidad de datos referentes a la relación social de cada usuario y no tanto en la información referente a los ítems [25].

### 2.3.3. Predicción de la influencia social basada en el dominio consciente del tiempo

En dicho proyecto los autores emplean ventanas temporales y redes neuronales para determinar la fiabilidad de los usuarios de la red de Twitter<sup>5</sup> y de manera general el grado de confiabilidad de la red social. Para desarrollar las ventanas temporales los autores han ordenado los tweets por orden cronológico y por franjas temporales, por ejemplo, semanas, meses, años, entre otras.

Aunque de este proyecto solo se ha tenido en cuenta la parte de generación de ventanas temporales, como conclusión referente a este tema se puede extraer que los intereses de los usuarios a lo largo de un espacio de tiempo varían. Por lo que este factor actúa negativamente a la hora de determinar la confianza de un usuario en su caso y en nuestro ámbito a la hora de recomendar un POI. Por lo que es recomendable dar mayor peso a las ventanas que se aproximan a la fecha actual y que su rango de fechas no sea excesivamente dilatado en el tiempo [26].

<sup>5</sup>Red social de microblogueo para que los creadores de contenido expresen y compartan sus pensamientos, creencias y dominios de interés a través de mensajes de texto cortos (tweets)

### 2.3.4. Discusión

El desarrollo de esta sección nos ha permitido conocer el estado actual de las metodologías en el campo de los sistemas de recomendación enfocados al ámbito turístico. Como hemos podido comprobar, no se ha encontrado ningún trabajo enfocado donde se aplique el uso de ensembles y ventanas temporales a POIs, de ahí que nos pareciese interesante cubrir este hueco.

Desde una perspectiva más global, los trabajos de recomendación musical y recomendación personalizada nos han ofrecido un posible enfoque de implementación de los ensembles, como hemos podido ver los algoritmos de filtrado colaborativo son los que mejores resultados generan a la hora de emplearlos en la implementación de un ensemble, pero para ello es necesario que el conjunto de datos sea de un gran tamaño. Por otro lado, en caso de disponer de gran cantidad de información referente a las relaciones sociales entre los usuarios, la importancia de los datos referentes a las interacciones usuario-ítem se reduce siempre que empleemos algoritmos basados en relaciones sociales. En cuanto al uso de ventanas temporales, hemos podido ver que para realizar un uso correcto de las mismas, es necesario ordenar la interacciones usuario-ítem de manera cronológica y realizar las particiones por franjas temporales, donde a medida que nos alejamos de la fecha actual perderá relevancia ya que los gustos y necesidades de los usuario en general varían con el tiempo.

Por último, refiriéndonos a los resultados, hemos podido conocer que el uso de ensembles y ventanas temporales en general mejoran de manera sensible el rendimiento de los sistemas de recomendación.

## DISEÑO Y DESARROLLO

---

Esta sección detalla el procedimiento seguido para lograr los objetivos propuestos al inicio del documento: comenzando con la descripción del hardware empleado; seguimos con el proceso de tratamiento sobre los datos; indicando las estrategias y metodologías a la hora de implementar los ensembles; describimos la funcionalidad de la librería que se ha implementado encargada de realizar el merge de los modelos que forman un ensemble; y por último se indican las medidas de calidad empleadas para medir los resultados de cada experimento.

### 3.1. Descripción del entorno de trabajo

Para el desarrollo de los experimentos se ha empleado el siguiente equipo:

Hardware	Requisito
Memoria RAM	16GB
CPU	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
SO	Ubuntu 18.04.5 LTS
Python	3.6

**Tabla 3.1:** Descripción de las características del entorno de trabajo.

De manera paralela se han empleado los servicios que ofrece Google Colaboratory <sup>1</sup> permitiendo reducir el tiempo que se ha invertido en ejecución, además de poder emplear la prueba gratuita, donde te permiten utilizar una GPU y gran capacidad de memoria por un tiempo limitado.

Es importante tener en cuenta esta información ya que en la mayoría de algoritmos empleados se ha tenido que reducir el tamaño de los conjuntos de datos debido a la cantidad de memoria que consume su ejecución tal y como se detalla en la siguiente sección.

---

<sup>1</sup> Entorno que permite ejecutar código escrito en Python idéntico a Jupyter Notebook mediante un navegador.

## 3.2. Descripción de los conjuntos de datos

Para la realización de los experimentos del proyecto se han empleado diferentes fuentes de datos permitiéndonos observar y comparar los resultados obtenidos. Los conjuntos de datos usados proceden de las redes sociales Gowalla <sup>2</sup> que dejó de estar activa en 2012, Foursquare <sup>3</sup> y Yelp <sup>4</sup>, estas dos aún están en activo. Las características de cada conjunto de datos antes de ser procesados son las siguientes:

Plataforma	Usuarios	POIs	Check-ins	Relaciones sociales
Foursquare	266,909	3,680,126	33,278,683	50,429
Gowalla	196,591	950,327	6,442,890	118,717
Yelp	1,968,703	2,018,995	8,021,122	78,634

**Tabla 3.2:** Tabla descripción de los conjuntos de datos antes de ser procesados.

En todos los conjuntos de datos se ha usado una partición de 80 % para entrenamiento y el 20 % restante para test. En la siguiente subsección se detalla el proceso de tratamiento de los datos y las características de los conjuntos de datos empleados para los experimentos.

### 3.2.1. Proceso de generación de los datos

A continuación se van a exponer todos los procesos necesarios para adecuar los datos a las necesidades de implementación de los algoritmos empleados en el proyecto. Para la realización de esta tarea se han empleado las librerías Pandas <sup>5</sup> y Matplotlib <sup>6</sup>, que permiten realizar el procesamiento de los datos de una manera simple y eficiente:

**Eliminar tuplas erróneas:** este proceso permite eliminar los valores nulos o sin lógica para evitar que la ejecución del algoritmo genere alguna excepción.

**Reindexación de los identificadores:** tanto los IDs de los usuarios como los de los POIs no eran correlativos ni comenzaban en 0, mediante esta transformación lo que se consigue es manejar números más pequeños reduciendo el espacio en memoria y el tiempo de ejecución de los algoritmos.

**Transformación de los datos:** mediante la librería Pgeocode <sup>7</sup> se ha transformando la lati-

<sup>2</sup>Fuente: <http://snap.stanford.edu/data/loc-gowalla.html>

<sup>3</sup>Fuente: <https://sites.google.com/site/yangdingqi/home/foursquare-dataset>

<sup>4</sup>Fuente: [https://www.yelp.com/dataset\\_challenge](https://www.yelp.com/dataset_challenge)

<sup>5</sup>Biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python [27].

<sup>6</sup>Biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy [28].

<sup>7</sup>Biblioteca de Python para consultas fuera de línea de alto rendimiento de coordenadas GPS, nombre de región y nombre de municipio a partir de códigos postales. También se admiten distancias entre códigos postales, así como consultas generales de distancia [29].

tud y longitud de cada POI a su código ZIP, permitiendo reducir la dimensionalidad de los datos y trabajar con números más pequeños. Por otro lado, se ha transformado la fecha de check-in de UTF a formato UNIX timestamp.

**Generación de la dimensión de rating:** todos los algoritmos empleados están basados en la puntuación que el usuario ha dado sobre cada item explorado, cuando se trabaja con predicción del siguiente POI esta dimensión no existe. Para solucionarlo se definen los ratings como 0 (no check-in) y 1 (check-in).

**Definición del tipo de dato:** permite asignar a cada dimensión de los datos el tipo que mejor se ajusta a este, permitiendo reducir el espacio ocupado en memoria y tiempo de ejecución.

**Reducción del tamaño de conjunto de datos:** debido a la limitación del hardware empleado y la cantidad de experimentos realizados, es necesario realizar este proceso para reducir los tiempos de ejecución.

**Split de los datos:** es importante tener en cuenta que de todos los POIs que queden en el conjunto de test, al menos exista un check-in de estos en el conjunto de entrenamiento, si no el modelo desconocerá ese POI y nunca podrá recomendarlo (este efecto solo ocurre en los algoritmos que no emplean matriz de utilidad).

Tras finalizar el procesado de generación de datos, los conjuntos de datos han quedado con las siguiente características:

Plataforma	Usuarios	POIs	Check-ins	Relaciones sociales
Foursquare	24,941	28,593	1,196,248	50,429
Gowalla	25,379	32,623	1,395,856	118,717
Yelp	30,887	18,995	860,888	78,634

**Tabla 3.3:** Tabla descripción de los conjuntos de datos después de ser procesados.

### 3.2.2. Descripción de los ficheros

Tras realizar el procesamiento de datos descrito en la sección anterior, los ficheros generados por cada dataset de los descritos en la sección 3.2.1 son los siguientes:

**check\_ins.txt:** Contiene todos los registros validados de todos los usuarios en los POIs.

**coords.txt:** Guarda la latitud y longitud de todos los POIs.

**data\_size.txt:** Almacena el número de usuarios, número de POIs.

**data\_size\_categories.txt:** Alberga el número de usuarios, número de POIs y número de categorías de los POIs.

**friends.txt** Contiene las relaciones de amistad en forma de tupla entre los usuarios.

**poi\_categories.txt:** Guarda la categoría de cada POI.

**test.txt:** Almacena el 20 % de los registros para emplearlos en la fase de test.

**train.txt:** Alberga el 80 % de los registros para utilizarlos en la fase de entrenamiento.

**tune.txt:** Contiene el número de registros en el fichero de entrenamiento y de test.

### 3.2.3. Descripción de las ventanas temporales

Mediante el uso de las ventanas temporales simulamos los estados de Nuevo, Avanzado y Final que se pretenden emplear en los experimentos. Para ello, se ha ordenado los conjuntos de datos de manera cronológica en función de la fecha y la hora del check-in y se han generado subconjuntos en base a una fecha de inicio y otra de fin. En la sección de parametrización del ensemble, se detalla en profundidad el tamaño destinado a cada uno de los estados definidos.

## 3.3. Proceso de generación del ensemble

### 3.3.1. Diagrama de flujo del ensemble

#### Descripción del flujo (según figura 3.1)

- 1- Se genera la ventana temporal a partir de los datos de entrenamiento en función de la etapa que se quiera simular.
- 2- Selección del tipo de ensemble a implementar.
- 3- Determinar el tamaño del ensemble y los modelos involucrados.
- 4- Decidir el tamaño de las recomendaciones.
- 5- Realizar el entrenamiento de los modelos.
- 6- Simulamos que un usuario realiza una solicitud.
- 7- Se generan los rankings individuales en función de la localización actual del usuario.
- 8- Selección del método de merge.
- 9- Generación del ranking final.
- 10- Evaluación del ranking, simulando que es un usuario.

### 3.3.2. Parametrización de los elementos del ensemble

Para poder generar las mejores recomendaciones de los ensembles y algoritmos implementados, ha sido necesario determinar una serie de parámetros que afectan drásticamente al tiempo de ejecu-

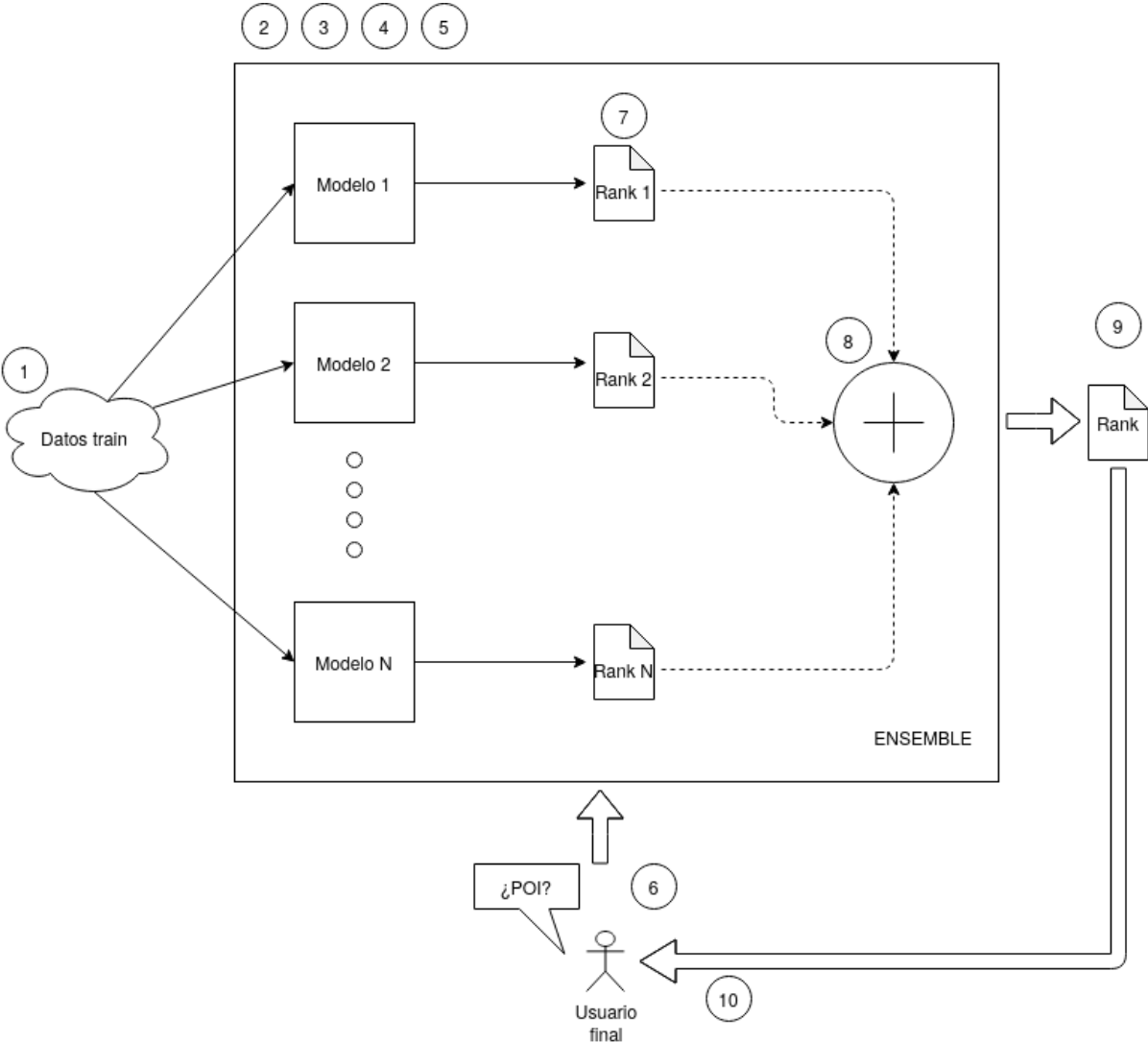


Figura 3.1: Generación de ensembles, diagrama de flujo.

ción, espacio ocupado en memoria y a los resultados obtenidos al obtener la recomendación final. A continuación se describen detalladamente los parámetros empleados comenzando por los más globales hasta llegar a los más específicos:

**Tamaño de la ventana temporal:** Este parámetro permite simular el estado del sistema, es decir, la cantidad de patrones empleados en la fase de entrenamiento. Pudiéndose diferenciar tres estados:

**Nuevo:** los modelos se entrenan con el 30 % de los patrones de entrenamiento más actuales.

**Avanzado:** los modelos se entrenan con el 60 % de los patrones de entrenamiento más actuales.

**Final:** los modelos se entrenan con el 100 % de los patrones de entrenamiento.

Estas ventanas permitirán determinar si el efecto de añadir datos en el entrenamiento, mejora los resultados de las recomendaciones generadas.

**Tipo del ensemble:** como se pudo apreciar en la sección de introducción a los ensembles (ver sección 2.2), en función del tipo seleccionado, se definirán los pasos y métodos a seguir para generar las recomendaciones finales.

**Tamaño del ensemble:** Determina el número de modelos a emplear a la hora de generar el ensemble. Para realiza los experimentos se ha determinado emplear un barrido de  $[3 - 6]$ , ya que estos valores permiten combinar varios modelos sin que afecte de forma negativa al tiempo de ejecución de la fase de entrenamiento.

**Modelos a emplear en el ensemble:** Para poder detallar el mejor conjunto de modelos, en la fase de experimentación, se ha realizado un barrido con todos los modelos contemplados en la sección del estado del arte (ver sección 2.1), teniendo en cuenta que no se ha contemplado la repetición de modelos en la implementación de los ensembles, exceptuando en la técnica de bagging (ver sección 2.2.2).

**Tamaño de la recomendación:** Este factor es uno de los más determinantes del funcionamiento del ensemble ya que determina el número de POIs que se van a retornar por recomendación. Para la realización de los experimentos, se han empleado los valores de  $[5, 10, 15, 20, 50]$ , siempre teniendo en cuenta que si simulásemos una experiencia de usuario lo recomendable es ofrecer 5 o 10 POIs.

**Estrategia de combinación:** Otro factor a tener en cuenta es el método que se empleará para generar el ranking final, para ello en las pruebas se han empleado todas las técnicas explicadas en la sección 3.4 para todas las implementaciones de ensembles realizadas.

**Parametrización de los modelos:** Por último, otro punto muy importante es la parametrización de cada modelo de forma individual. Para ello se han empleado los valores recomendados en la literatura y un rango de valores próximo a los mismos para determinar el mejor



valor dado los datos. Los parámetros a tener en cuenta de manera individual son los siguientes:

Modelo	Parámetro 1	Parámetro 2	Parámetro 3	Parámetro 4	Parámetro 5
Random	-	-	-	-	-
POP	-	-	-	-	-
PL	-	-	-	-	-
UCF	-	-	-	-	-
FCF	etta	-	-	-	-
SVD	-	-	-	-	-
MGM	alpha	theta	dmax	-	-
PFM	K	alpha	beta	iters	IRate
USG	etha	-	-	-	-
GeoSoCa	alpha	-	-	-	-

**Tabla 3.4:** Tabla descripción individual de los parámetros de los modelos.

### 3.4. Estrategias para combinar rankings

A la hora de emplear los modelos dentro del meta-modelo del ensemble, cada uno de ellos generará un ranking de tamaño N, por lo que se ha tenido que implementar una librería que permita combinar todos los rankings para ofrecer la recomendación final para el usuario. Las técnicas de combinación implementadas son las siguientes:

**Aleatorio:** Se genera un ranking aleatorio a partir de los rankings de entrada.

**Equilibrado:** Se retorna un ranking con el mismo número de POIs mejor posicionados de los rankings de entrada.

**Por equipos:** Se genera un patrón aleatorio con el id de los ranking de entrada y se van extrayendo POIs hasta que se alcance el tamaño del ranking de salida.

**Personalizado por secuencia:** Se genera el ranking de salida en función al patrón de los ID de los rankings definido.

**Personalizado por pesos** El patrón se decide en base a los pesos de los rankings individuales. Puede darse el caso de que algún ranking no se seleccione.

**Items comunes:** El ranking se genera en base a los items comunes de los rankings de entrada, en caso de empate se seleccionan de manera aleatoria.

**Popularidad:** El ranking se genera en base a los items más populares de los rankings de entrada, en caso de empate se seleccionan de manera aleatoria.

## 3.5. Estrategias de evaluación

Para determinar cómo de bueno es el funcionamiento de los algoritmos, es necesario emplear una o varias métricas que permitan clasificarlos mediante los resultados. Se han empleado las siguientes métricas, que se han extraído a partir de la literatura usada para desarrollar este trabajo.

### 3.5.1. Precision@k

Se puede definir como la cantidad de items recomendados hasta la posición k que son relevantes.

$$Precision@k = \frac{1}{|U|} \sum_{u \in U} \frac{|Rel_u@k|}{k} \quad (3.1)$$

donde  $|U|$  es el número total de usuarios y  $|Rel_u@k|$  es el total de items relevantes de los primeros k items del ranking [30].

### 3.5.2. Recall@k

Medida inversamente proporcional a la precision@k que indica la cantidad de items relevantes en los primeros k items del ranking de recomendación en relación al total de items relevantes.

$$Recall@k = \frac{1}{|U|} \sum_{u \in U} \frac{|Rel_u@k|}{|Rel_u|} \quad (3.2)$$

donde  $|U|$  es el número total de usuarios y  $|Rel_u@k|$  es el total de items relevantes de los primeros k items del ranking [31].

### 3.5.3. Mean average precision mAP@k

Es la precisión promedio calculada en cada posición del ranking donde haya un ítem relevante:

$$mAP@k = \frac{1}{|U|} \sum_u \frac{1}{|Rel_u|} \sum_{i \in Rel_u} P@rank(u, i) \quad (3.3)$$

donde  $|U|$  es el número total de usuarios,  $|Rel_u@k|$  es el total de items relevantes del usuario  $u$  en los primeros k items del ranking y  $P@rank(u, i)$  es la precisión del ranking actual para el usuario  $u$  pero calculada en la posición donde aparece el ítem  $i$  en dicho ranking, es decir,  $rank(u, i)$  [32].

### 3.5.4. Normalized discounted cumulative gain nDCG@k

Es una medida que permite evaluar la calidad del ranking. Se basa en la suposición de que los items con mayor relevancia (puntuación o frecuencia de visitas en el ámbito del turismo) y que aparecen antes en el ranking son más útiles que los documentos con menor relevancia, y estos a su vez son más útiles que los irrelevantes. Para calcular esta métrica es necesario realizar algunos cálculos intermedios.

#### Cumulative gain CG@k

Es la ganancia acumulada de todas las frecuencias de los POIs relevantes en un conjunto de recomendación.

$$CG@k = \sum_{i \in k}^n freq_i \quad (3.4)$$

donde  $i$  corresponde a los items del ranking de tamaño  $k$ .

#### Discounted cumulative gain DCG@k

En este paso del cálculo, se realiza la ganancia acumulativa teniendo en cuenta la posición que ocupa el item en el ranking, como se puede apreciar en la siguiente fórmula:

$$DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (3.5)$$

donde  $i$  corresponde a la posición de los items dentro del ranking de tamaño  $k$ , y  $rel_i$  devuelve la relevancia asociada al ítem que ocupa la posición  $i$ -ésima (teniendo en cuenta tanto su posición  $i$  en el ranking como el nivel de preferencia por el usuario según el test).

#### Ideal discounted cumulative gain iDCG@k

Se realiza el mismo cálculo que en DCG (ver sección 3.5.4) con la salvedad de que previamente hay que generar el ranking ideal, que consiste en situar los items dentro del ranking ordenados de mayor a menor relevancia o frecuencia.

$$iDCG@k = \sum_{i=1}^{|Rel_k|} \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (3.6)$$

usando la misma notación explicada en la fórmula de DCG@k, considerando también que  $Rel_k$

devuelve el número de elementos relevantes hasta la posición  $k$ .

### Normalized discounted cumulative gain $nDCG@k$

Finalmente, para determinar la calidad del ranking basta con comparar el ranking obtenido con el ranking ideal de la siguiente manera:

$$nDCG@k = \frac{DCG@k}{iDCG@k} \quad (3.7)$$

donde la calidad se mide en el rango de  $[0 - 1]$ .

## PRUEBAS Y RESULTADOS

El desarrollo de esta sección nos ha permitido valorar los aspectos del funcionamiento de los ensembles, permitiéndonos evaluar: si se han cumplido los hitos propuestos en la sección de objetivos (ver sección 1.2); cerciorarnos que cada uno de los algoritmos y librerías funcionan de manera correcta durante el proceso de generación del ensemble; y determinar si los resultados obtenidos por parte de los ensembles mejoran a los generados por los baselines.

### 4.1. Estudio de las diferencias en los conjuntos de datos

A continuación se van a detallar las diferencias existentes en los datos obtenidos de cada red social empleada en el desarrollo del proyecto. En todos los datasets se han realizado las mismas particiones para entrenamiento y test, con una proporción de **80 %-20 %**. Para generar las ventanas temporales se ha empleado un **30 %** de los datos de entrenamiento para el estado **Nuevo**, un **60 %** para el estado **Avanzado** y un **100 %** para el estado **Final**.

Dataset	Estado	Usuarios	POIs	Check-ins	Promedio de check-ins/usuario
Foursquare	Nuevo	24.445	27.386	281.767	12
Gowalla	Nuevo	18.637	31.109	294.900	16
Yelp	Nuevo	29.900	18.255	660.219	23
Foursquare	Avanzado	24.933	28.471	556.916	23
Gowalla	Avanzado	18.737	32.414	591.592	32
Yelp	Avanzado	30.868	18.850	1.320.300	23
Foursquare	Final	24.941	28.551	933.074	38
Gowalla	Final	18.737	32.495	984.386	53
Yelp	Final	30.887	18.942	2.201.404	72

**Tabla 4.1:** Tabla descripción de los datasets por estado del ensemble.

Como se puede apreciar en la tabla 4.1 cuando el ensemble se genera en un estado Nuevo el sistema posee prácticamente toda la información referente a los usuario y los POIs. El efecto de au-

mentar la ventana temporal únicamente va a afectar a la cantidad de información que se posee sobre los check-ins, tal y como refleja la columna de promedio de check-ins. Por otro lado, si observamos las gráficas del anexo A se puede apreciar que las representaciones de cada dataset en cada estado del ensemble son similares en forma, pero no en escala.

## 4.2. Resultados obtenidos con los baselines

Antes de comenzar con la descripción de los resultados, hemos definido las siguientes abreviaturas para los nombres de los algoritmos.

Nombre	Abreviatura
Aleatorio	Random
Popularidad	POP
Ley de potencias	PL
Filtrado colaborativo basado en usuario	UCF
Filtrado colaborativo basado en amistad	FCF
Filtrado colaborativo basado en descomposición de valores singulares	SVD
Filtrado colaborativo basado en un modelo probabilístico	PFM
Recomendación basada en las preferencias del usuario e influencias sociales y geográficas	USG
Modelo conjunto basado en modelos Gaussianos y probabilísticos	MGMPMF
Recomendación geográfico-social basada en las categorías de los POI's	GeoSoCa

**Tabla 4.2:** Descripción de las abreviaturas de los algoritmos.

A continuación se muestran los resultados que han generado los baselines con los tres conjuntos de datos evaluados con las métricas de calidad a partir de la literatura.

El procedimiento seguido para la selección de los mejores resultados ha sido: realizar el análisis por ventanas temporales; crear un orden de prioridad de métricas de calidad: nDCG será la más relevante, ya que tiene en cuenta la posición del POI dentro del ranking, dejando como complementarias las métricas mAP, precisión y recall que se emplearán en caso de empate en nDCG; escoger los tres algoritmos que han obtenido el mayor valor en la métrica; y, como último paso, seleccionar los que posean un tamaño de ranking razonable, en nuestro caso [5, 10, 15].

### Baselines generados del dataset de Foursquare

A partir de la tabla 4.3, hemos determinado cuáles son los algoritmos que mejor se adaptan a los datos obtenidos de la red social Foursquare. Según lo observado en los datos de esta tabla, los algoritmos con mayor acierto y que cumplen con las especificaciones en cada ventana temporal son:

Algoritmo	Ventana	Tam. ranking	Precision	Recall	mAP	nDCG
POP	Nueva	5	0.0426	0.0265	0.0232	0.0450
PFM	Nueva	5	0.0434	0.0272	0.0207	0.0411
POP	Nueva	10	0.0339	0.0426	0.0182	0.0384
PFM	Avanzada	5	0.0463	0.0297	0.0231	0.0456
SVD	Avanzada	5	0.0431	0.0295	0.0256	0.0449
POP	Avanzada	5	0.0419	0.0261	0.0238	0.0436
SVD	Final	5	0.0573	0.0398	0.0352	0.0596
PFM	Final	5	0.0469	0.0300	0.0232	0.0453
POP	Final	5	0.0426	0.0265	0.0240	0.0441

**Tabla 4.3:** Descripción de los mejores resultados con el dataset de Foursquare.

**Ventana nueva:** para este subconjunto de datos los mejores algoritmos han sido POP empleando dos longitudes diferentes de ranking de recomendación y PFM, fijándonos en la métrica nDCG se puede apreciar que todos los algoritmos están aproximadamente en un **0.04**. Otro resultado destacable es que los mejores valores se han alcanzado cuando el tamaño de ranking es **5**.

**Ventana avanzada:** en este caso los algoritmos que han generado mejores resultados han sido PFM, SVD Y POP. En comparación con los datos de la ventana nueva, PFM ha superado el rendimiento de POP; POP\_5 ha reducido su rendimiento y POP\_10 ya no está entre los mejores; y por último, SVD ha entrado en el grupo de los 3 de mejores algoritmos quedando muy próximo del rendimiento generado por PFM.

**Ventana avanzada:** para esta última ventana los algoritmos que han producido mejor rendimiento vuelven a ser SVD, PFM y POP. Lo más destacable es que en comparación con la ventana avanzada los tres algoritmos han mejorado su rendimiento, destacando a SVD por encima de todo ya que ha mejorado aproximadamente un **0.015** hasta alcanzar un **0.6** en la métrica de nDCG.

Haciendo alusión a la gráfica 4.1 se observa que los datos descritos en la tabla concuerdan con los resultados representados con **k=5** y empleando la métrica nDCG. Lo más llamativo en la gráfica es que independientemente de la ventana temporal en la que nos encontremos, siempre son los mismos algoritmos los que mejores resultados generan y que la diferencia con el resto es aproximadamente de un **0.03**.

Como conclusión, se puede afirmar que los mejores candidatos para los datos de Foursquare son SVD, PFM, POP. El tamaño del ranking recomendable es **k=5** y el efecto de aumentar el tamaño de la ventana temporal solo afecta a SVD, que a medida que aumenta obtiene mejor rendimiento.

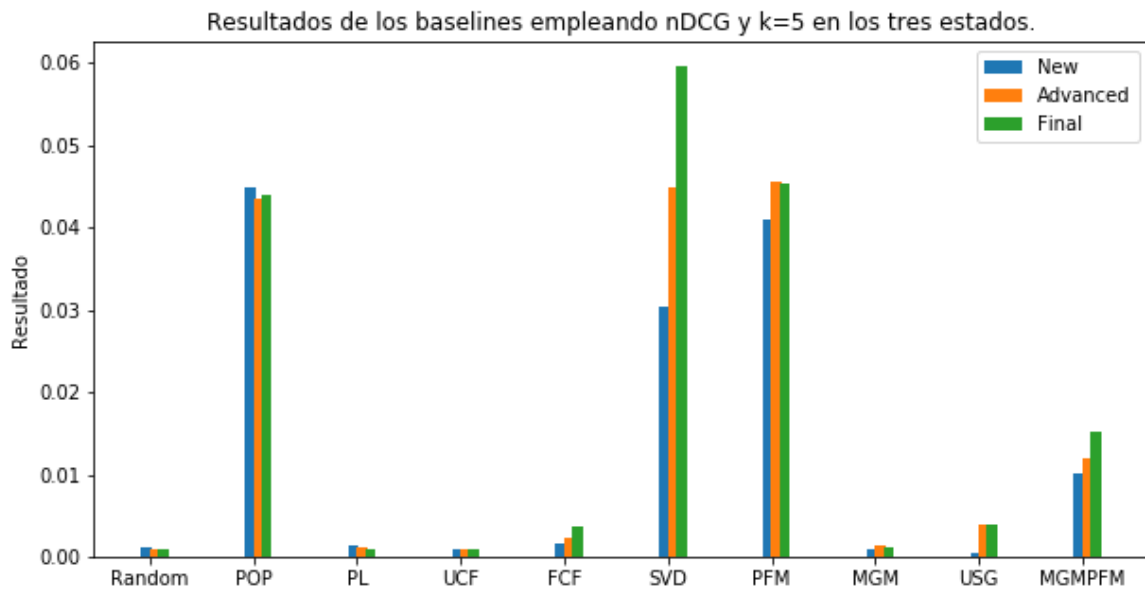


Figura 4.1: Resultados de los baselines de Foursquare empleando nDCG y longitud de ranking 5.

### Baselines generados del dataset de Gowalla

A partir de la tabla 4.4, se realizará el análisis de los algoritmos que mejor resultado han obtenido al emplear el dataset de Gowalla.

Algoritmo	Ventana	Tam. ranking	Precision	Recall	mAP	nDCG
SVD	Nueva	5	0.0573	0.0398	0.0352	0.0596
PFM	Nueva	5	0.0434	0.0272	0.0207	0.0451
POP	Nueva	5	0.0426	0.0265	0.0232	0.0410
SVD	Avanzada	5	0.0551	0.0279	0.0315	0.0563
PFM	Avanzada	5	0.0486	0.0199	0.0262	0.0508
POP	Avanzada	5	0.0425	0.0173	0.0257	0.0441
SVD	Final	5	0.0749	0.0376	0.0453	0.0770
PFM	Final	5	0.0485	0.0202	0.0291	0.0503
POP	Final	5	0.0425	0.0173	0.0257	0.0441

Tabla 4.4: Descripción de los mejores resultados con el dataset de Gowalla.

Fijándonos en los datos reflejados en dicha tabla, los algoritmos que mejores resultados han obtenido y se adaptan a las condiciones reflejadas en la introducción (ver sección 4.2) han sido:

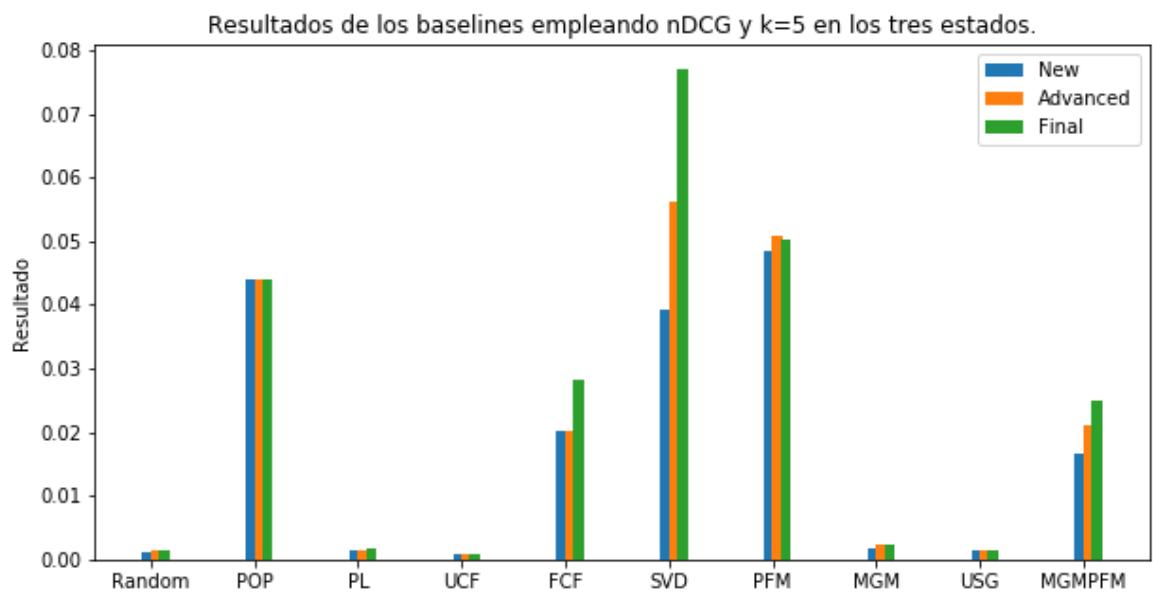
**Ventana nueva:** en esta etapa los mejores algoritmos han sido SVD, PFM y POP; todos ellos con un ranking de tamaño **5**. Todos los algoritmos están en acierto de **0.04** excepto SVD,



que ha obtenido un rendimiento de **0.6**.

**Ventana avanzada:** en esta fase el orden de los algoritmos es el mismo en comparación con la fase nueva, también han obtenido los mejores valores con un tamaño de ranking **5**. Observando el rendimiento de manera individual, PFM y POP han mejorado sus resultados en un **0.05** y **0.03** respectivamente; por contra, SVD ha reducido su rendimiento en un **0.03**, aunque sigue siendo el que mejores resultados obtiene.

**Ventana avanzada:** para esta última ventana los algoritmos siguen siendo los mismos y en el mismo orden y tamaño de ranking **5**. Lo más destacable es que en esta ventana PFM y POP mantiene sus resultados y SVD ha mejorado de manera significativa hasta alcanzar un **0.076** de rendimiento.



**Figura 4.2:** Resultados de los baselines de Gowalla empleando nDCG y longitud de ranking 5.

Por otro lado, centrándonos en la gráfica 4.2 se puede decir que SVD es el algoritmo que mejor rendimiento ofrece, ya que casi ha obtenido el doble de acierto que PFM que es el segundo mejor algoritmo. Además habría que tener en cuenta los algoritmos MGMPFM y FCF, ya que aunque no aparezcan en la tabla 4.4 han obtenido resultados considerables respecto al resto de algoritmos que ninguno se aproxima al **0.01**.

Como conclusión podemos decir que los algoritmos SVD, PFM, POP, MGMPFM y FCF son los candidatos a tener mejor rendimiento a la hora de utilizarlos con los datos procedentes de Gowalla. Por otro lado, los algoritmos han alcanzado sus máximos con la ventana final y con un tamaño de ranking de longitud 5.

## Baselines generados del dataset de Yelp

En este bloque debemos determinar cuáles son los algoritmos que mejor se adaptan a los datos obtenidos de la red social Yelp. Para ello nos fijaremos en la tabla 4.5 que se presenta a continuación.

Algoritmo	Ventana	Tam. ranking	Precision	Recall	mAP	nDCG
SVD	Nueva	5	0.0247	0.0108	0.0122	0.0251
POP	Nueva	5	0.0124	0.0054	0.0058	0.0133
MGMPFM	Nueva	5	0.0126	0.0069	0.0065	0.0123
SVD	Avanzada	5	0.0345	0.0155	0.0174	0.0351
MGMPFM	Avanzada	5	0.0169	0.0089	0.0086	0.0171
FCF	Avanzada	5	0.0150	0.0060	0.0079	0.0154
SVD	Final	5	0.0482	0.0210	0.0260	0.0492
FCF	Final	5	0.0213	0.0092	0.0121	0.0215
MGMPFM	Final	5	0.0194	0.0096	0.0101	0.0196

**Tabla 4.5:** Tabla descripción de los mejores resultados con el dataset de Yelp.

Como se puede apreciar en dicha tabla, los modelos de recomendación que han tenido mayor acierto han sido:

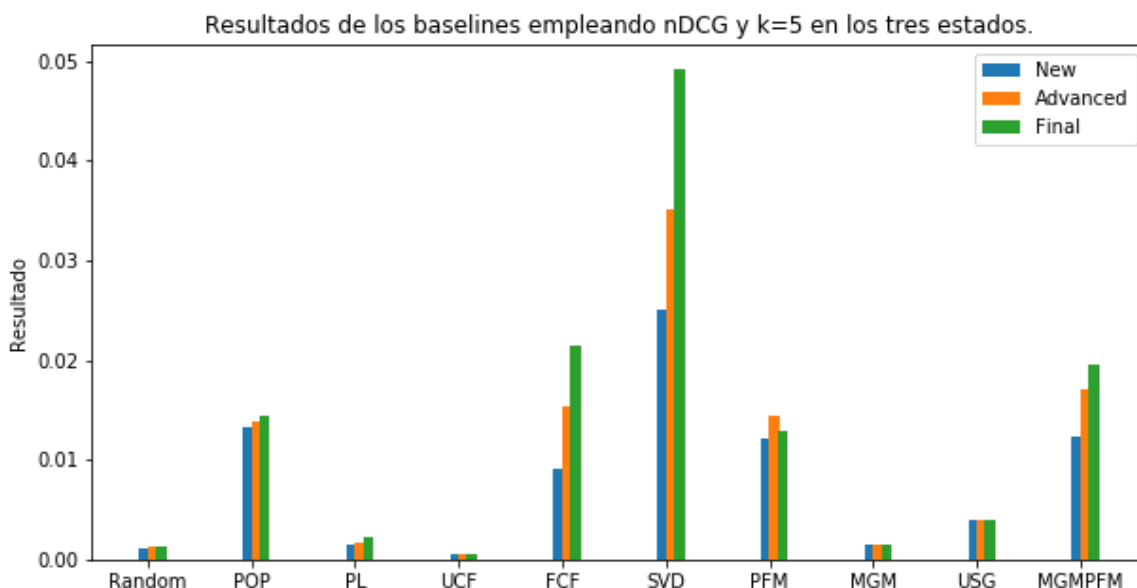
**Ventana nueva:** en esta fase los mejores algoritmos han sido SVD, POP y MGMPFM todos ellos con un ranking de tamaño **5** y un rango de acierto de  $[0,01 - 0,02]$ .

**Ventana avanzada:** en esta ventana el algoritmo de POP ha sido reemplazado por el algoritmo FCF. Los resultados ofrecidos por los algoritmos han aumentado, en el caso de MGMPFM es mínimo; en cambio, SVD ha mejorado un **0.01**. Los mejores resultados se siguen obteniendo con un ranking de tamaño **5**.

**Ventana avanzada:** para esta última ventana los algoritmos y el tamaño de ranking se mantienen; en cambio, el orden de los mismos ha cambiado entre MGMPFM y FCF. Observando los resultados MGMPFM y FCF ha aumentado su rendimiento en torno a un **0.03**, en cuanto al rendimiento de SVD ha generado una mejoría del doble.

Si lo representamos mediante una gráfica como en la figura 4.3, se puede ver que los algoritmos con mejores resultados son SVD, FCF Y MGMPFM en los tres tamaños de ventana. Por otro lado, se deberían tener en cuenta los algoritmos POP y PFM ya que su rendimiento es muy próximo al del algoritmo MGMPFM.

Para finalizar, destacar que los algoritmos seleccionados para generar el ensemble para el dataset de Yelp son SVD, FCF, MGMPFM, PFM y POP.



**Figura 4.3:** Resultados de los baselines de Yelp empleando nDCG y longitud de ranking 5.

### 4.3. Resultados obtenidos con ensembles basados en votación

A partir de la siguiente sección se estudiará el comportamiento de los baselines. Los emplearemos de forma conjunta en ensembles implementados con la estrategia de votación. A pesar del gran número de pruebas realizadas en este apartado, para abreviar nos centraremos por cada uno de los conjuntos de datos y dentro de éstos, por ventanas temporales. Para ello, nos plantearemos si van a cumplir o no los siguientes requisitos:

- 1- Determinar si el uso conjunto de los baselines seleccionados en el apartado anterior mejoran el rendimiento al implementarlos como ensemble.
- 2- Estudiar el efecto de combinar los mejores baselines con los peores, ¿afecta positivamente o negativamente al rendimiento?.
- 3- Establecer el tamaño de conjunto recomendable para implementar un ensemble.
- 4- Indagar cuál ha sido la estrategia de merge que mejores resultados ha ofrecido.
- 5- Comprobar si se ha mejorado el rendimiento frente a los baselines que se han seleccionado de referencia.

Además, antes de mostrar y analizar los resultados, se darán una serie de pautas para entender el contenido de las tablas: en primer lugar los tres baselines de cada tamaño de ventana aparecerán

en las tres últimas filas de las tablas; los mejores resultados en cada métrica serán resaltados en negrita, en caso de empate, serán resaltados todos los valores iguales; los resultados obtenidos en cada experimento han sido ordenados mediante el valor de la métrica nDCG, ya que es la que se tomará de referencia para comparar los rendimientos de los ensembles.

## Resultados del conjunto de datos de Foursquare

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(POP, FCF')	5	Ítems comunes	0.0426	0.0265	<b>0.0232</b>	<b>0.0450</b>
(POP, MGMPFM)	5	Ítems comunes	0.0426	0.0265	<b>0.0232</b>	<b>0.0450</b>
(POP, MGMPFM, FCF)	5	Ítems comunes	0.0426	0.0265	<b>0.0232</b>	<b>0.0450</b>
POP (Baseline)	5	-	0.0426	0.0265	<b>0.0232</b>	<b>0.0450</b>
PFM (Baseline)	5	-	<b>0.0434</b>	0.0272	0.0207	0.0410
POP (Baseline)	10	-	0.0339	<b>0.0426</b>	0.0182	0.0384

**Tabla 4.6:** Resultados ensemble votación de Foursquare en ventana nueva.

Fijándonos en la tabla de la ventana nueva, se puede observar mediante la métrica nDCG, que las mejores combinaciones han obtenido los mismos resultados que el algoritmo POP. Esto se debe a que que todas ellas emplean dicho algoritmo; y en conjunto con el resto de algoritmos del ensemble han generado el mismo ranking. También se puede apreciar que la mejor estrategia de merge ha sido la de ítems comunes. Referente al resto de métricas en mAP ha sucedido el mismo efecto que en nDCG y en precision y recall ningún ensemble ha superado a los baselines.

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(SVD, POP)	5	Equilibrado	<b>0.0509</b>	<b>0.0331</b>	<b>0.0281</b>	<b>0.0528</b>
(SVD, POP)	5	Por equipos	0.0497	0.0326	0.0273	0.0522
(SVD, PFM, POP)	5	Por equipos	0.0484	0.0321	0.0253	0.0509
(SVD, PFM, POP)	5	Equilibrado	0.0480	0.0319	0.0259	0.0505
(SVD, PFM)	5	Equilibrado	0.0451	0.0306	0.0246	0.0485
PFM (Baseline)	5	-	0.0463	0.0297	0.0231	0.0456
SVD (Baseline)	5	-	0.0431	0.0295	0.0256	0.0449
POP (Baseline)	5	-	0.0419	0.0261	0.0238	0.0436

**Tabla 4.7:** Resultados ensemble votación de Foursquare en ventana avanzada.

De los resultados contenidos en la tabla 4.7 se puede destacar que la mejor combinación de ensemble en todas las métricas ha sido SVD y POP empleando los métodos de ranking equilibrado y por equipos (ha superado al mejor baseline en **0.07**). También se puede observar que el algoritmo SVD aparece en todas las combinaciones; y POP en todas excepto una. Otro factor destacable es que todos los conjuntos únicamente contienen los algoritmos que se emplean como baseline. Por último, destacar que todos los ensembles han alcanzado su máximo con un ranking de longitud **5**.

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(SVD, POP)	5	Equilibrado	0.0581	0.0382	0.0330	<b>0.0615</b>
(SVD, POP)	5	Por equipos	0.0578	0.0381	0.0330	0.0612
(SVD, PFM, POP, MGMPFM)	5	Ítems comunes	<b>0.0587</b>	0.0392	<b>0.0360</b>	0.0611
(SVD, PFM, POP, FCF)	5	Ítems comunes	0.0573	<b>0.0398</b>	0.0352	0.0606
SVD (Baseline)	5	-	0.0573	<b>0.0398</b>	0.0352	0.0596
PFM (Baseline)	5	-	0.0469	0.0300	0.0232	0.0453
POP (Baseline)	5	-	0.0426	0.0265	0.0240	0.0441

**Tabla 4.8:** Resultados ensemble votación de Foursquare en ventana final.

A partir de la tabla de la ventana final se puede apreciar que las combinaciones POP y SVD han superado al mejor baseline en **0.02**. En todas las combinaciones se emplean estos dos algoritmos. La combinación SVD, PFM, POP y MGMPFM ha obtenido los mejores resultados en las métricas precision y mAP, y SVD, PFM, POP y FCF; y el baseline SVD son los que poseen el mejor rendimiento mediante la métrica recall. Otro dato a tener en cuenta es que todos los rankings poseen una longitud de **5**. Los mejores métodos de merge han sido equilibrado, por equipo e ítems comunes.

### Resultados del conjunto de datos de Gowalla

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(PFM, POP)	5	Equilibrado	0.0355	0.0147	0.0217	0.0518
(PFM, POP)	5	Por equipos	0.0357	0.0147	0.0222	0.0505
(SVD, PFM, POP)	5	Por equipos	0.0416	0.0184	0.0240	0.0499
(SVD, POP)	5	Equilibrado	0.0457	0.0203	0.0254	0.0488
SVD (Baseline)	5	-	<b>0.0573</b>	<b>0.0398</b>	<b>0.0352</b>	<b>0.0596</b>
PFM (Baseline)	5	-	0.0434	0.0272	0.0207	0.0451
POP (Baseline)	5	-	0.0426	0.0265	0.0232	0.0410

**Tabla 4.9:** Resultados ensemble votación de Gowalla en ventana nueva.

Lo más destacable de la tabla 4.9 es que ninguna combinación de ensemble ha sido capaz de mejorar al baseline de SVD. Sin embargo, sí han sido capaces de mejorar los resultados de los otros dos. También se puede apreciar que como mínimo, todas las combinaciones contienen dos de los tres baselines; y que la combinación (PFM, POP) es la que genera mejores resultados empleando las técnicas de merge equilibrado y por equipos. Otro factor a tener en cuenta es que todas las combinaciones han obtenido su mejor resultado cuando su ranking es de longitud **5**.

Mediante la tabla 4.10 se puede observar que los ensembles solo superan a los baselines en la métrica nDCG entorno a un  $[0,02 - 0,04]$ . En el resto, el baseline SVD es el que ha generado mejores resultados. También se puede ver que las mejores técnicas de merge han sido por equipos y equilibrado, además ambas se emplean en la misma combinación de algoritmos. Por último resaltar que el algoritmo PFM únicamente aparece en los ensembles que mejor resultado han generado.

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(SVD, PFM, POP)	5	Por equipos	0.0433	0.0203	0.0260	<b>0.0599</b>
(SVD, PFM, POP)	5	Equilibrado	0.0428	0.0199	0.0250	0.0592
(SVD, PFM)	5	Por equipos	0.0546	0.0258	0.0293	0.0589
(SVD, PFM)	5	Equilibrado	0.0545	0.0255	0.0291	0.0589
(SVD, POP)	5	Equilibrado	0.0526	0.0247	0.0299	0.0586
SVD (Baseline)	5	-	<b>0.0551</b>	<b>0.0279</b>	<b>0.0315</b>	0.0563
PFM (Baseline)	5	-	0.0486	0.0199	0.0262	0.0508
POP (Baseline)	5	-	0.0425	0.0173	0.0257	0.0441

**Tabla 4.10:** Resultados ensemble votación de Gowalla en ventana avanzada.

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(PFM, POP)	5	Equilibrado	0.0355	0.0147	0.0217	0.0518
(SVD, PFM, POP)	5	Por equipos	0.0416	0.0184	0.0240	0.0499
(SVD, POP)	5	Equilibrado	0.0457	0.0203	0.0254	0.0488
SVD (Baseline)	5	-	<b>0.0749</b>	<b>0.0376</b>	<b>0.0453</b>	<b>0.0770</b>
PFM (Baseline)	5	-	0.0485	0.0202	0.0291	0.0503
POP (Baseline)	5	-	0.0425	0.0173	0.0257	0.0441

**Tabla 4.11:** Resultados ensemble votación de Gowalla en ventana final.

En cuanto a la tabla que describe la ventana final (tabla 4.11), lo más relevante es que ningún conjunto ha generado mejores resultados que el baseline SVD. Además la diferencia entre ellos es bastante significativa (en torno a un 0.025). Sin embargo, sí han generado mejores resultados que el resto de baselines. Por otro lado, todas las combinaciones generadas contienen como mínimo dos baselines.

## Resultados del conjunto de datos de Yelp

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(SVD, PFM, POP, FCF)	5	Ítems comunes	<b>0.0289</b>	<b>0.0129</b>	<b>0.0210</b>	<b>0.0295</b>
(SVD, MGMPFM, FCF)	5	Ítems comunes	0.0275	0.0119	0.0142	0.0284
(SVD, PFM, POP)	5	Ítems comunes	0.0269	0.0112	0.0139	0.0269
SVD (Baseline)	5	-	0.0247	0.0108	0.0122	0.0251
POP (Baseline)	5	-	0.0124	0.0054	0.0058	0.0133
MGMPFM (Baseline)	5	-	0.0126	0.0069	0.0065	0.0123

**Tabla 4.12:** Resultados ensemble votación de Yelp en ventana nueva.

A partir de los datos de la tabla 4.12 podemos afirmar que todos los conjuntos generados han superado a los baselines en todas las métricas, en especial el formado por (SVD, PFM, POP, FCF), que ha obtenido los mejores resultados. También se puede observar que todos los conjuntos están formados por los baselines reflejados en la tabla; a excepción de PFM, que no forma parte de los tres mejores algoritmos estudiados de forma individual, pero sí pertenece a los candidatos recomendados

para implementar un ensemble (ver sección B.3). La mejor implementación de merge ha sido la de ítems comunes que aparece en todos los conjuntos reflejados en la tabla. Por último destacar que todos los ensembles han generado sus mejores resultados con un ranking de longitud **5**.

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(SVD, PFM, MGMPFM)	5	Ítems comunes	<b>0.0395</b>	<b>0.0198</b>	<b>0.0202</b>	<b>0.0401</b>
(SVD, PFM, POP)	5	Ítems comunes	0.0377	0.0176	0.0190	0.0380
(SVD, FCF)	5	Ítems comunes	0.0357	0.0171	0.0196	0.0363
SVD (Baseline)	5	-	0.0345	0.0155	0.0174	0.0351
MGMPFM (Baseline)	5	-	0.0169	0.0089	0.0086	0.0171
FCF (Baseline)	5	-	0.0150	0.0060	0.0079	0.0154

**Tabla 4.13:** Resultados ensemble votación de Yelp en ventana avanzada.

A través de la tabla de ventana avanzada se puede apreciar que los resultados obtenidos son similares a la tabla de ventana nueva. Todas las combinaciones de ensemble han superado en todas las métricas a los baselines. De nuevo el método de merge ítems comunes es el que mejor resultados ha obtenido; los rankings con mejor acierto son los de longitud **5**. Por último resaltar que la mejor combinación ha sido (SVD, PFM, MGMPFM) con una mejora en comparación con el baseline SVD de **0.05**.

Combinación	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
(SVD, PFM, MGMPFM)	5	Ítems comunes	0.0497	<b>0.0258</b>	0.0262	<b>0.0526</b>
(SVD, POP, FCF)	5	Ítems comunes	<b>0.0510</b>	0.0235	<b>0.0271</b>	0.0513
(SVD, FCF)	5	Ítems comunes	0.0482	0.0210	0.0260	0.0492
SVD (Baseline)	5	-	0.0482	0.0210	0.0260	0.0492
FCF (Baseline)	5	-	0.0213	0.0092	0.0121	0.0215
MGMPFM (Baseline)	5	-	0.0194	0.0096	0.0101	0.0196

**Tabla 4.14:** Resultados ensemble votación de Yelp en ventana final.

Como refleja la tabla 4.14, podemos ver que el mejor ensemble ha sido el compuesto por (SVD, PFM, MGMPFM), aunque su mejora no ha sido muy significativa. Por otro lado, el ensemble (SVD, POP, FCF) ha generado un rendimiento muy similar al otro ensemble, de hecho le ha superado en las métricas de recall y mAP. Al igual que en las ventanas nueva y avanzada, la mejor técnica de merge que se adapta a estos datos ha sido la de ítems comunes y la mejor longitud de ranking ha sido **5**.

### Discusión sobre ensembles basados en votación

En esta sección se va dar respuesta a los puntos enumerados en la introducción de los ensembles de tipo votación (ver sección 4.3).

En cuanto al rendimiento general del ensemble de votación ha sido bastante bueno para los conjuntos de datos de Foursquare y Yelp, en todas las ventanas temporales los resultados generados por los ensembles han superado a los baselines. En cambio, cuando se ha empleado los datos procedentes

de Gowalla en ninguna ventana se ha podido superar, haciendo alusión a las características de los conjuntos de datos (ver sección 4.1) la diferencia más significativa es que en todas las ventanas el número de POIs siempre supera al número de usuario.

Haciendo alusión a los algoritmos podemos ver que los que mejor se han adaptado a esta metodología son SVD, PFM y POP, aunque en algunas ocasiones también han aparecido FCF y MGMPFM. Lo más destacable es que a excepción de POP, todos los algoritmos pertenecen a la categoría de filtrado colaborativo (ver sección 2.1.2) o están formados por algún algoritmo de esa categoría. Por lo que podemos afirmar que el efecto de combinar los baselines con el resto de algoritmos en general no producen tan buenos resultados como los compuestos por ellos mismos.

Refiriéndonos al tamaño óptimo de los ensemble, los mejores resultados se han obtenido con una longitud de 2 para todas las ventanas temporales de los conjuntos de datos pertenecientes a Foursquare y Gowalla, y longitud 3 para Yelp.

Por último, las estrategias de merge que han generado mejores resultados han sido equilibrado, por equipos e ítems comunes. En cuanto a las ventanas temporales o los conjuntos de datos no se ha apreciado un patrón de aparición, lo único que se puede decir es que el método equilibrado es el que se repite con más frecuencia.

## 4.4. Resultados obtenidos con ensembles basados en bagging

En esta sección expondremos los resultados obtenidos con los ensembles de tipo bagging (ver sección 2.2.2). Para el desarrollo de los experimentos se dividirán por conjuntos de datos y a su vez por ventanas temporales. Debido al gran número de pruebas realizadas únicamente se mostrarán las tres mejores combinaciones. Durante la realización de los experimentos, nos planteamos las siguientes cuestiones:

- 1- Comprobar si se ha mejorado el rendimiento frente a los baselines que se han seleccionado de referencia.
- 2- Estudiar el efecto de combinar los mejores baselines con los peores, ¿afecta positivamente o negativamente al rendimiento?.
- 3- Establecer el tamaño de conjunto recomendable para implementar un ensemble.
- 4- Indagar cuál ha sido la estrategia de merge que mejores resultados ha ofrecido.
- 5- Determinar el número óptimo de particiones.

La notación seguida para mostrar las tablas de resultados es la misma que en la sección anterior.



## Resultados del conjunto de datos de Foursquare

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Por equipos	0.0116	0.0052	0.0027	0.0123
FCF	4	5	Por equipos	0.0132	0.0102	0.0012	0.0107
POP	4	5	Ítems comunes	0.0072	0.0084	0.0007	0.0073
POP (Baseline)	-	5	-	0.0426	0.0265	<b>0.0232</b>	<b>0.0450</b>
PFM (Baseline)	-	5	-	<b>0.0434</b>	0.0272	0.0207	0.0411
POP (Baseline)	-	10	-	0.0339	<b>0.0426</b>	0.0182	0.0384

**Tabla 4.15:** Resultados ensemble bagging de Foursquare en ventana nueva.

Observando la tabla 4.15 podemos ver que los mejores algoritmos son los mismos que los baselines. En referencia a los resultados podemos ver que éstos han sido muy pobres en comparación con los baselines, incluso los resultados generados por SVD se aproximan al peor baseline POP\_10. Otro factor a tener en cuenta es que se han logrado los mejores resultados con **4** particiones y un tamaño de ranking de **5**. En cuanto a la técnica de merge, la que mejores resultados ha generado es la de por equipos.

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Por equipos	0.0123	0.0065	0.0030	0.0156
PFM	4	5	Por equipos	0.0238	0.0116	0.0026	0.0125
POP	4	5	Ítems comunes	0.0087	0.0097	0.0009	0.0074
PFM (Baseline)	-	5	-	<b>0.0462</b>	<b>0.0297</b>	0.0231	<b>0.0456</b>
SVD (Baseline)	-	5	-	0.0430	0.0294	<b>0.0256</b>	0.0449
POP (Baseline)	-	5	-	0.0418	0.0261	0.0238	0.0436

**Tabla 4.16:** Resultados ensemble bagging de Foursquare en ventana avanzada.

A partir de los datos de la tabla de ventana avanzada (tabla 4.16), debemos destacar que, aunque aumente el tamaño de la ventana temporal, los resultados siguen siendo igual de malos: entre el mejor ensemble y el peor baseline hay una diferencia de **0.03**. También se puede apreciar que el algoritmo FCF ha sido sustituido por PFM. Además se observa que los mejores resultados se han alcanzado con un tamaño de partición **4** y una longitud de ranking **5**.

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Por equipos	0.0281	0.0182	0.0129	0.0378
PFM	4	5	Por equipos	0.0237	0.0115	0.0125	0.0319
POP	4	5	Ítems comunes	0.0215	0.0097	0.0087	0.0307
SVD (Baseline)	-	5	-	<b>0.0573</b>	<b>0.0397</b>	<b>0.0352</b>	<b>0.0596</b>
PFM (Baseline)	-	5	-	0.0468	0.0300	0.0232	0.0453
POP (Baseline)	-	5	-	0.0425	0.0264	0.0240	0.0440

**Tabla 4.17:** Resultados ensemble bagging de Foursquare en ventana final.

Fijándonos en la tabla 4.17 podemos observar que no se ha superado a los baselines en ninguna

combinación. Al igual que en las dos ventanas anteriores, los resultados máximos se alcanzan con **4** particiones y ranking de longitud **5** y existe una diferencia entre el mejor ensemble y el peor baseline de **0.022**. En cuanto al tipo de merge, el mejor resultado se produce cuando se aplica el basado en equipos.

### Resultados del conjunto de datos de Gowalla

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
PFM	4	5	Equilibrado	0.0138	0.0119	0.0101	0.0130
POP	4	5	Por equipos	0.0104	0.0115	0.0094	0.0104
SVD	4	5	Equilibrado	0.0078	0.0049	0.0057	0.0085
SVD (Baseline)	-	5	-	<b>0.0573</b>	<b>0.0397</b>	<b>0.0352</b>	<b>0.0596</b>
PFM (Baseline)	-	5	-	0.0434	0.0272	0.0207	0.0450
POP (Baseline)	-	5	-	0.0425	0.0264	0.0232	0.0410

**Tabla 4.18:** Resultados ensemble bagging de Gowalla en ventana nueva.

Lo más destacable de la tabla 4.18 es que ningún ensemble ha superado a los baselines, entre el peor baseline y el mejor ensemble existe una diferencia de **0.03**. Otro factor a tener en cuenta es que los algoritmos empleados para los ensembles son los mismos que los baselines, y han alcanzado su mejor resultado con un tamaño de partición de **4** y una longitud de ranking de **5**. Por último destacar que la mejor implementación de merge ha sido la equilibrada, que aparece en dos de los merges.

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Equilibrado	0.0157	0.0148	0.0112	0.0179
PFM	4	5	Por equipos	0.0148	0.0138	0.0109	0.0166
POP	4	5	Equilibrado	0.0105	0.0087	0.0098	0.0130
SVD (Baseline)	-	5	-	<b>0.0551</b>	<b>0.0279</b>	<b>0.0314</b>	0.0563
PFM (Baseline)	-	5	-	0.0486	0.0198	0.0262	0.0508
POP (Baseline)	-	5	-	0.0425	0.0173	0.0257	0.0441

**Tabla 4.19:** Resultados ensemble bagging de Gowalla en ventana avanzada.

A partir de la tabla 4.19 se puede apreciar que los resultados son similares a los de la ventana nueva con la diferencia de que el orden de las implementaciones ha variado por SVD, PFM y POP, aunque la diferencia entre el peor baseline y el mejor ensemble ha aumentado a **0.04**. Por otro lado, de nuevo la implementación de merge equilibrada sigue siendo la que mejores resultados genera, con un tamaño de ranking **5** y un número de particiones **4** se alcanzan los mejores resultados.

Como refleja la tabla 4.20 y al igual que las ventanas nueva y avanzada, la diferencia entre el peor baseline y el mejor merge ha ido aumentando hasta alcanzar una diferencia de **0.05**. De nuevo los algoritmos SVD, PFM y POP con un número de particiones **4** y un tamaño de ranking **5** han generado los mejores resultados. Además la mejor técnica de merge ha sido la equilibrada.

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Equilibrado	0.0265	0.0183	0.0133	0.0279
PFM	4	5	Por equipos	0.0239	0.0163	0.0124	0.0226
POP	4	5	Equilibrado	0.0163	0.0126	0.0106	0.0178
SVD (Baseline)	-	5	-	<b>0.0749</b>	<b>0.0375</b>	<b>0.0452</b>	<b>0.0769</b>
PFM (Baseline)	-	5	-	0.0484	0.0201	0.0291	0.0502
POP (Baseline)	-	5	-	0.0425	0.0173	0.0257	0.0441

Tabla 4.20: Resultados ensemble bagging de Gowalla en ventana final.

### Resultados del conjunto de datos de Yelp

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
POP	4	5	Ítems comunes	0.0088	0.0070	0.0064	0.0096
FCF	4	5	Ítems comunes	0.0075	0.0065	0.0062	0.0086
SVD	4	5	Ítems comunes	0.0072	0.0062	0.0059	0.0083
SVD (Baseline)	-	5	-	<b>0.0247</b>	<b>0.0108</b>	<b>0.0121</b>	<b>0.0251</b>
POP (Baseline)	-	5	-	0.0124	0.0054	0.0057	0.0132
MGMPFM (Baseline)	-	5	-	0.0126	0.0068	0.0064	0.0123

Tabla 4.21: Resultados ensemble bagging de Yelp en ventana nueva.

En cuanto a la tabla 4.21, podemos observar que los resultados generados por los ensembles no ha superado a los baselines. Existe una diferencia entre el ensemble POP y el baseline MGMPFM de **0.003** aproximadamente. En cuanto a los algoritmos, podemos observar que son diferentes los escogidos como ensemble y como baseline, excepto SVD que es común a los dos. Por otro lado, podemos ver que la implementación de merge que ha generado mejores resultados ha sido la de ítems comunes, y que el número de particiones y tamaño de ranking óptimos han sido **4** y **5** respectivamente.

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Ítems comunes	0.0175	0.0128	0.0114	0.0187
FCF	4	5	Ítems comunes	0.0128	0.0106	0.0099	0.0139
POP	4	5	Ítems comunes	0.0100	0.0106	0.0086	0.0117
SVD (Baseline)	-	5	-	<b>0.0345</b>	<b>0.0154</b>	<b>0.0174</b>	<b>0.0351</b>
MGMPFM (Baseline)	-	5	-	0.0169	0.0088	0.0086	0.0170
FCF (Baseline)	-	5	-	0.0150	0.0060	0.0078	0.0153

Tabla 4.22: Resultados ensemble bagging de Yelp en ventana avanzada.

Gracias a la tabla referente a la ventana avanzada, podemos observar que se han seleccionado los mismos algoritmos que en la ventana nueva, pero el orden ha cambiado por SVD, FCF, POP. Además se puede ver que en este caso el ensemble SVD ha superado a los baselines FCF y MGMPFM, y la diferencia entre el baseline SVD es de **0.017**. Por otro lado, de nuevo la mejor implementación de merge son los ítems comunes, la longitud de ranking óptima **5** y el mejor número de particiones es **4**.

De los resultados de la tabla 4.23, se puede observar que los resultados en comparación con las

Combinación	Particiones	Tam. ranking	Tipo de merge	Precision	Recall	mAP	nDCG
SVD	4	5	Ítems comunes	0.0289	0.0139	0.0140	0.0295
FCF	4	5	Ítems comunes	0.0169	0.0081	0.0089	0.0179
POP	4	5	Ítems comunes	0.0149	0.0073	0.0073	0.0146
SVD (Baseline)	-	5	-	<b>0.0482</b>	<b>0.0210</b>	<b>0.0259</b>	<b>0.0491</b>
FCF (Baseline)	-	5	-	0.0213	0.0092	0.0121	0.0214
MGMPFM (Baseline)	-	5	-	0.0194	0.0095	0.0101	0.0196

**Tabla 4.23:** Resultados ensemble bagging de Yelp en ventana final.

otras ventanas se mantienen: de nuevo el ensemble SVD ha conseguido superar a los dos últimos baselines, en cambio, la diferencia con el baseline SVD ha aumentado a **0.02**. Además el tamaño de ranking y el número de particiones siguen siendo **5** y **4** respectivamente; y como mejor técnica de merge ha resultado ítems comunes de nuevo.

## Discusión sobre ensembles basados en bagging

En esta sección se va dar respuesta a los puntos enumerados en la introducción de los ensembles de tipo votación (ver sección 4.4).

Refiriéndonos a los resultados generados por esta técnica de ensemble podemos decir que los resultados han sido lo contrario a los ensembles de tipo votación (ver sección 4.3), ya que en ninguno de los datasets el rendimiento ha superado al mejor baseline, aunque haciendo alusión a las pruebas realizadas sobre el dataset de Yelp, sí se ha podido superar al resto de baselines en las ventanas avanzada y final.

En cuanto a la variedad de algoritmos, observamos que los que mejores resultados han generado han sido SVD, PFM, POP y FCF en dicho orden. Referente a estos algoritmos podemos decir que pertenecen a la categoría de filtrado colaborativo.

En cuanto a la parametrización de particiones y longitudes de rankings, vemos que tanto para los baselines como para los ensembles siempre ha sido **4** particiones y **5** de longitud.

Por último, la implementación de merge que mejor rendimiento ha generado ha sido por equipos para los conjuntos de datos de Foursquare y Gowalla, e ítems comunes en todas las implementaciones para Yelp.

## 4.5. Discusión de todos los resultados

En esta sección realizaremos un estudio de los resultados de los ensembles de manera global. Al comparar los resultados generados por las técnicas implementadas en las secciones anteriores, podemos decir que solo la técnica de votación (ver sección 2.2.1) ha mejorado los resultados de los

baselines. Esto es debido a que los algoritmos que mejores rendimientos generan pertenecen a la familia del filtrado colaborativo (ver sección 2.1.2) y una de sus principales características es que se emplean con grandes volúmenes de datos. De ahí que al realizar las particiones de los datos, provoca que los algoritmos entrenen con menor cantidad de datos y en consecuencia, se reduce su acierto a la hora de recomendar los POIs.

De cara a los algoritmos empleados por las dos técnicas, en general hemos visto que existen muchas similitudes. Los más empleados y que mejores resultados han generado han sido SVD (con una diferencia notable), PFM y POP. Además los algoritmos que en la fase de selección de baselines (ver apéndice B) generaban malos resultados, no han sido capaces de mejorar cuando se empleaban de forma conjunta en los ensembles de tipo votación.

Refiriéndonos a los conjuntos de datos, los que mejor se han adaptado para votación han sido Foursquare y Yelp, y para bagging únicamente Yelp. Los datos procedentes de Gowalla han generado los peores resultados en ambas implementaciones con una diferencia considerable en el rendimiento; creemos que se puede deber a una mala calidad en los datos temporales de este conjunto de datos.



# CONCLUSIONES Y TRABAJO FUTURO

---

En este capítulo se exponen las principales ideas obtenidas al realizar el proyecto y su posterior análisis de los resultados; además de las posibles mejoras y empleabilidad que podrá obtener el proyecto en un futuro próximo.

## 5.1. Conclusiones

A lo largo de este trabajo hemos puesto en práctica técnicas de generación de ensembles votación y bagging en un ámbito turístico para generar recomendaciones de POIs. Comparando las dos técnicas de ensemble hemos podido determinar que la de votación es la que mejores resultados ha producido. Este efecto se debe a que la mayoría de algoritmos empleados pertenecen a la familia de filtrado colaborativo y, en consecuencia, tienen un rendimiento mayor con una mayor cantidad de datos.

Por otro lado, al haber empleado diferentes conjuntos de datos, hemos podido observar que los que poseen mayor cantidad de usuarios que de ítems han generado mejores resultados en las recomendaciones, esto es debido a que al tener menos ítems aumenta el número de check-ins en el mismo POI.

Además hemos generado tres tipos diferentes de ventanas temporales para simular los estados por los que pasa un sistema de recomendación a lo largo de su ciclo de vida (nuevo, avanzado y final). De esta parte hemos podido observar que por la categoría de los mejores baselines la ventana temporal que mejores resultados ha generado ha sido la de final.

En cuanto a la parametrización de los ensembles, hemos podido determinar que a la hora de seleccionar una técnica de merge es independiente a la estrategia de ensemble utilizada. Referente al tamaño de ranking, los resultados obtenidos sí poseen gran peso ya que los mejores resultados se han obtenido con el menor ranking posible y los experimentos se han evaluado con una métrica que tiene en cuenta la posición del POI dentro del ranking.

Por último, respecto al coste computacional de la generación de ensembles, hemos podido ver que es necesario disponer de un hardware avanzado ya que a la hora de realizar los experimentos más

exigentes fue necesario el uso de una plataforma externa para la ejecución de las pruebas.

## 5.2. Trabajo futuro

Como trabajo futuro se podrían seleccionar otros algoritmos basados en redes neuronales que se adapten a las metodologías de ensembles de Boosting y Stacking. De hecho, durante la lectura de literatura para el desarrollo del proyecto se han podido extraer los siguientes algoritmos como punto de partida:

- STF-RNN**: este algoritmo emplea redes neuronales recurrentes [33].
- NCF**: en esta implementación se emplea el perceptrón multicapa como base.
- AT-LSTM**: es otra implementación basada en redes neuronales [34].
- RecQ**: es una librería basada en el estado del arte de las últimas aportaciones de los expertos de investigación en sistemas de recomendación mediante el uso de la inteligencia artificial [35].
- Recommenders**: es una librería que ofrece Microsoft generada a partir de la competición "Microsoft MIND Competition" [36].

Otra posible mejora sería modificar la apreciación de las ventanas temporales: en vez de usar los estados Nuevo, Avanzado y Final, se podrían realizar las particiones por semanas, meses, trimestres, años, entre otros; y estudiar el comportamiento de los ensembles con esta nueva implementación. Como complemento, se podría emplear la técnica de aprendizaje activo, que consiste en añadir al entrenamiento del recomendador los patrones de test permitiendo que éste vaya aprendiendo a lo largo de de su vida útil.

Otro enfoque sería formatear los datos como secuencias temporales. Esto consiste en ordenar los datos por usuario y orden de visita de los POIs, de esta manera se tendría en cuenta el factor de temporalidad de visita. Por ejemplo, si disponemos de una secuencia de check-ins de tamaño 20, se podría emplear como primer patrón los cuatro primeros check-ins como entrada y el quinto como objetivo, después del segundo al quinto como entrada y el sexto como objetivo y así sucesivamente hasta que completar toda la secuencia.

Por último, se podrían emplear conjuntos de datos procedentes de otros ámbitos y estudiar en cuáles de éstos los ensembles generan mejores resultados y bajo qué condiciones.



# BIBLIOGRAFÍA

---

- [1] M. A. Ghazanfar, “Robust, scalable, and practical algorithms for recommender systems.” [https://eprints.soton.ac.uk/343761/1/MusiThesisFinal\\_Mirrored.pdf](https://eprints.soton.ac.uk/343761/1/MusiThesisFinal_Mirrored.pdf), 2012. [Estado: Activo].
- [2] S. Khatwani and M. Chandak, “Building Personalized and Non Personalized recommendation systems.” <https://ieeexplore.ieee.org/document/7877661>, 2016. [Estado: Activo].
- [3] Rossi and R. J, *Mathematical Statistics : An Introduction to Likelihood Based Inference*, pp. 227–228. John Wiley Sons, first ed., 2018.
- [4] Y. Zheng, L. Zhang, X. Xie, and W. Y. Ma, “Mining interesting locations and travel sequences from gpstrajectories.” <https://www.microsoft.com/en-us/research/publication/mining-interesting-locations-and-travel-sequences-from-gps-trajectories/>, 2009. [Estado: Activo].
- [5] B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems.” [https://www.researchgate.net/publication/200121027\\_Collaborative\\_Filtering\\_Recommender\\_Systems](https://www.researchgate.net/publication/200121027_Collaborative_Filtering_Recommender_Systems). [Estado: Activo].
- [6] F. L. Vreixo, “Técnicas eficientes para la recomendación de productos basadas en filtrado colaborativo.” <https://ruc.udc.es/dspace/handle/2183/10122>. [Estado: Activo].
- [7] J. S. Breese, D. Heckerman, and C. Kadie, “ Empirical analysis of predictive algorithms for collaborative filtering.” <https://arxiv.org/pdf/1301.7363.pdf>, 1998. [Estado: Activo].
- [8] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “ An algorithmic framework for performing collaborative filtering.” <http://files.grouplens.org/papers/algs.pdf>, 1999. [Estado: Activo].
- [9] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee”, “Exploiting geographical influence for collaborative point-of-interest recommendation.” [https://www.cse.cuhk.edu.hk/irwin.king.new/\\_media/presentations/p325.pdf](https://www.cse.cuhk.edu.hk/irwin.king.new/_media/presentations/p325.pdf), 2011. [Estado: Activo].
- [10] S. Amit, “ Modern Information Retrieval: A Brief Overview.” <http://www1.cs.columbia.edu/~gravano/cs6111/Readings/singhal.pdf>, 2001. [Estado: Activo].
- [11] H. Ma, H. Yang, M. R. Lyu, and I. King, “SoRec: social recommendation using probabilistic matrix factorization.” [http://web.cs.ucla.edu/~yzsun/classes/2014Spring\\_CS7280/Papers/Recommendation/paper\\_cikm08\\_sorec\\_hao.pdf](http://web.cs.ucla.edu/~yzsun/classes/2014Spring_CS7280/Papers/Recommendation/paper_cikm08_sorec_hao.pdf), 2008. [Estado: Activo].
- [12] Y. Koren, R. Bell, and C. Volinsky, “MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS.” <https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf>, 2009. [Estado: Activo].

- [13] P. Dayan, "Non-negative Matrix Factorization with Sparseness Constraints." <https://www.jmlr.org/papers/volume5/hoyer04a/hoyer04a.pdf>, 2004. [Estado: Activo].
- [14] B. R., "Hybrid Recommender Systems: Survey and Experiments." <https://link.springer.com/article/10.1023/A:1021240730564>, 2002. [Estado: Activo].
- [15] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks." [https://www.researchgate.net/publication/285809856\\_Fused\\_matrix\\_factorization\\_with\\_geographical\\_and\\_social\\_influence\\_in\\_location-based\\_social\\_networks](https://www.researchgate.net/publication/285809856_Fused_matrix_factorization_with_geographical_and_social_influence_in_location-based_social_networks), 2012. [Estado: Activo].
- [16] H. D. Navone, P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, "A Learning Algorithm For Neural Network Ensembles." <https://www.redalyc.org/pdf/925/92551209.pdf>, 2000. [Estado: Activo].
- [17] M. Mohammadi, H. Alizadeh, and B. Minaei-Bidgoli, "Neural Network Ensembles Using Clustering Ensemble and Genetic Algorithm." <https://www.redalyc.org/pdf/925/92551209.pdf>, 2000. [Estado: Activo].
- [18] F. S. Caparrini, "Métodos combinados de aprendizaje." <http://www.cs.us.es/~fsancho/?e=106>, 2018. [Estado: Activo].
- [19] A. C. Tan and D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification." <https://bura.brunel.ac.uk/bitstream/2438/3013/1/TanGilbertNZ2003.pdf>, 2003. [Estado: Activo].
- [20] J. Cao, S. Kwong, R. Wang, X. Li, K. Li, and X. Kong, "Neurocomputing." <https://www.sciencedirect.com/science/article/abs/pii/S0925231214011345>, 2015. [Estado: Activo].
- [21] J. M. Heras, "Ensembles: voting, bagging, boosting, stacking." <https://iartificial.net/ensembles-voting-bagging-boosting-stacking/>, 2018. [Estado: Activo].
- [22] mariajesusbigml, "Introduction to Boosted Trees." <https://blog.bigml.com/2017/03/14/introduction-to-boosted-trees/>, 2017. [Estado: Activo].
- [23] J. Borges, "The Power of Ensembles in Deep Learning." <https://towardsdatascience.com/the-power-of-ensembles-in-deep-learning-a8900ff42be9>, 2019. [Estado: Activo].
- [24] A. Schclar, A. Tsikinovsky, L. Rokach, A. Meisels, and L. Antwarg, "Ensemble methods for improving the performance of neighborhood-based collaborative filtering." [https://www.researchgate.net/publication/221141016\\_Ensemble\\_methods\\_for\\_improving\\_the\\_performance\\_of\\_neighborhood-based\\_collaborative\\_filtering](https://www.researchgate.net/publication/221141016_Ensemble_methods_for_improving_the_performance_of_neighborhood-based_collaborative_filtering), 2020. [Estado: Activo].
- [25] J. Seo, S. Choi, M. Kim, and S. H. Chung-Ang, "The method of personalized recommendation with ensemble combination." <http://isyu.info/jowua/papers/jowua-v4n4-8.pdf>, 2020. [Estado: Activo].

- 
- [26] K. Y. Chan, O. Al-Kadi, M. Al-Tawil, P. Wongthongtham, T. Issa, H. Saadeh, M. Al-Hassan, B. Bremie, and A. Albahlal, "Time-aware domain-based social influence prediction." <https://link.springer.com/article/10.1186/s40537-020-0283-3#Abs1>, 2020. [Estado: Activo].
- [27] V. Autores, "Pandas documentation for version 1.1.0." <https://pandas.pydata.org/docs/>, 2020. [Estado: Activo].
- [28] V. Autores, "Matplotlib documentation for version 3.3.0." <https://matplotlib.org/>, 2020. [Estado: Activo].
- [29] V. Autores, "Pgeocode documentation for version 0.2.1." <https://pypi.org/project/pgeocode/>, 2020. [Estado: Activo].
- [30] A. Bellogín, P. Castells, and I. Cantador, "Precision-oriented evaluation of recommender systems: An algorithmic comparison." <https://repositorio.uam.es>, 2011. [Estado: Activo].
- [31] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms." <http://www.tnt3.ir/wp-content/uploads/recommender-system-results-using-ga.pdf>, 2011. [Estado: Activo].
- [32] C. D. Manning, P. Raghavan, and H. Schütze, "Evaluation in information retrieval." <https://nlp.stanford.edu/IR-book/pdf/08eval.pdf>, 2009. [Estado: Activo].
- [33] A. Al-Molegi, M. Jabreel, and B. Ghaleb, "Stf-rnn: Space time features-based recurrent neural network for predicting people next location." <https://ieeexplore.ieee.org/document/7849919>, 2016. [Estado: Activo].
- [34] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal lstm network for next poi recommendation." <https://ieeexplore.ieee.org/document/8723186>, 2020. [Estado: Activo].
- [35] Yu, Junliang, G. Min, Y. Hongzhi, L. J. Gao, Chongming, and W. Qinyong, "Generating reliable friends via adversarial training to improve social recommendation," *arXiv preprint arXiv:1909.03529*, 2019.
- [36] Microsoft, "Recommenders by microsoft." <https://github.com/microsoft/recommenders>, 2020. [Estado: Activo].



# ACRÓNIMOS

---

**CPU** Control Processing Unit.

**FCF** Friend Collaborative Filtering.

**GPU** Graphics Processing Unit.

**ID** Identification.

**PFM** Probabilistic Matrix Factorization.

**POI** Point of interest.

**POP** Most Popularity.

**SO** System Operating.

**SVD** Singular Value Decomposition.

**TFM** Trabajo Fin de Máster.

**USG** User collaborative filtering based on Social and Geographical.

**UTF** Unicode Transformation Format.

**ZIP** Código Postal.



# APÉNDICES





# REPRESENTACIÓN GRÁFICA DE LOS DATASETS BASADA EN LAS VENTANAS TEMPORALES.

---

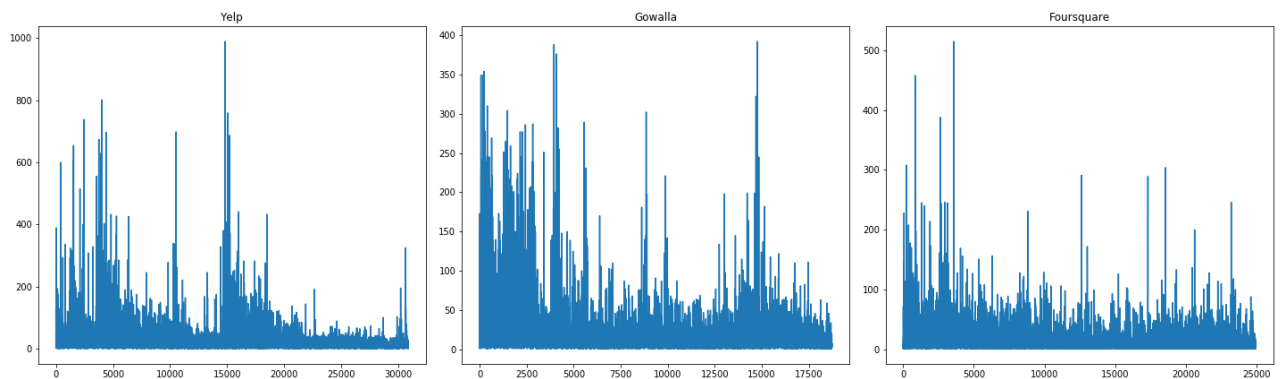


Figura A.1: Datos de entrenamiento simulado un estado nuevo.

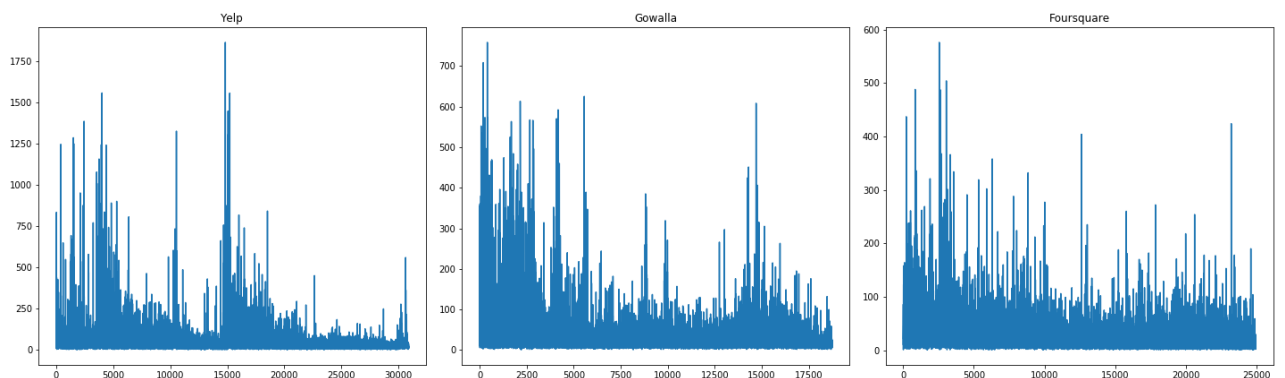
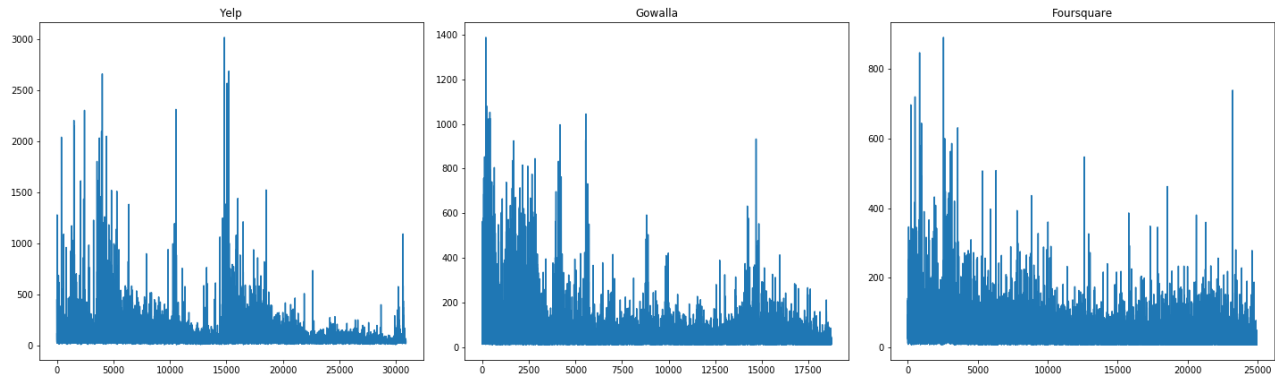
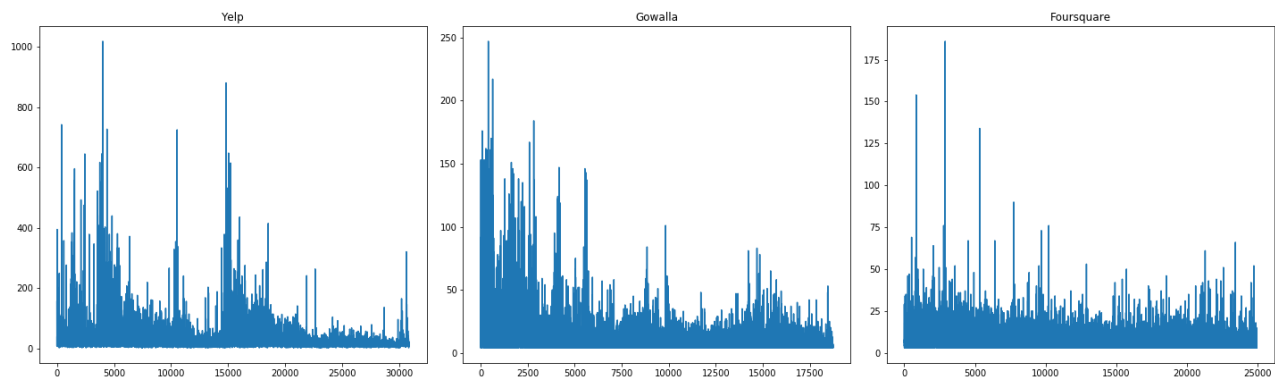


Figura A.2: Datos de entrenamiento simulado un estado avanzado.



**Figura A.3:** Datos de entrenamiento simulado un estado final.



**Figura A.4:** Datos de test.

## REPRESENTACIÓN GRÁFICA DE LOS BASELINES.

En este apartado del apéndice se muestra de forma gráfica los resultados obtenidos en los diferentes datasets y métricas de calidad empleadas.

### B.1. Foursquare

#### Estado nuevo

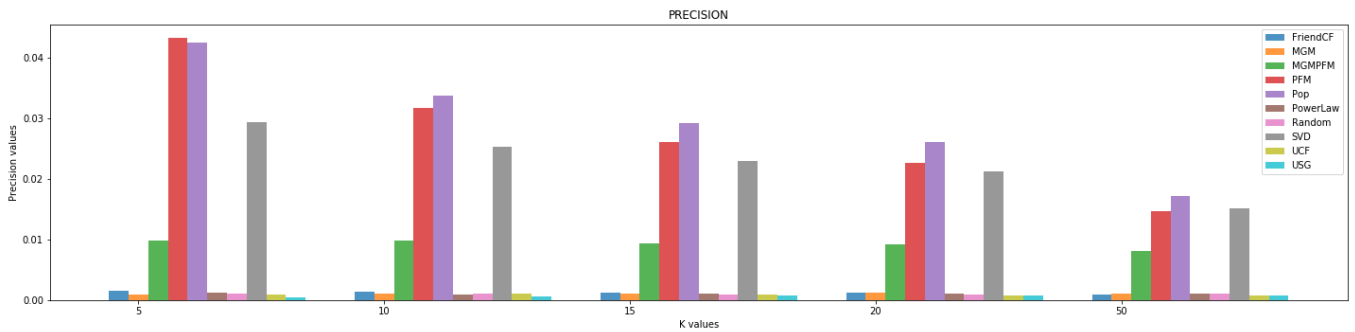


Figura B.1: Baselines de Foursquare en estado nuevos evaluados con Precision.

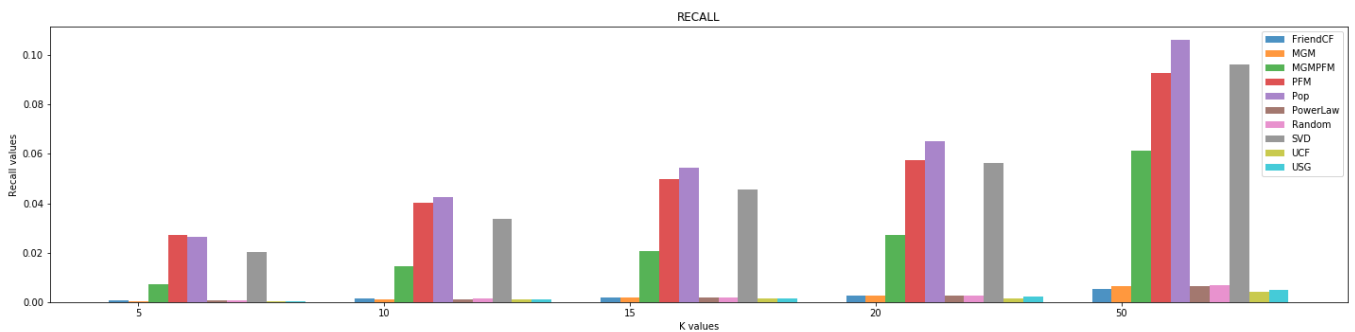


Figura B.2: Baselines de Foursquare en estado nuevos evaluados con Recall.

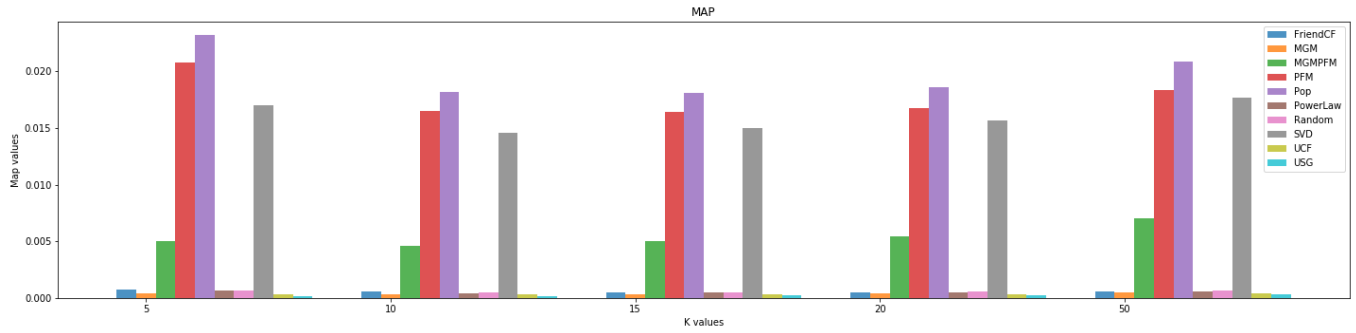


Figura B.3: Baselines de Foursquare en estado nuevos evaluados con mAP.

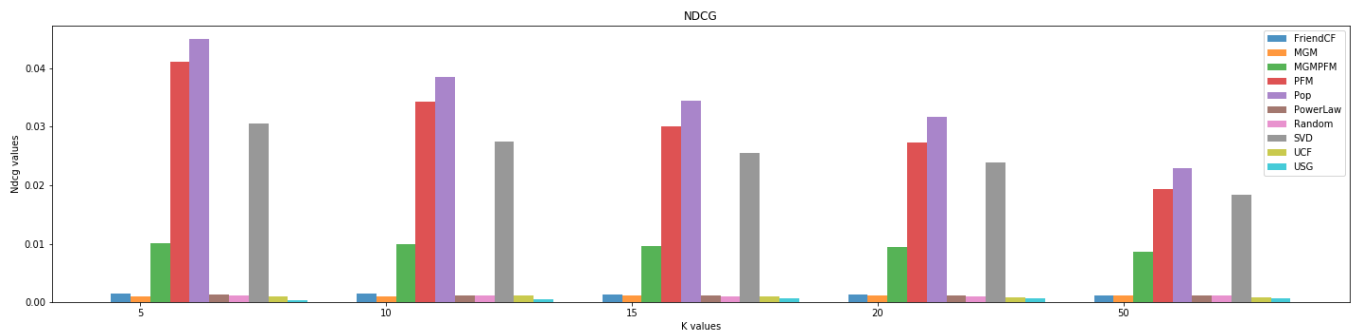


Figura B.4: Baselines de Foursquare en estado nuevos evaluados con nDCG.

## Estado avanzado

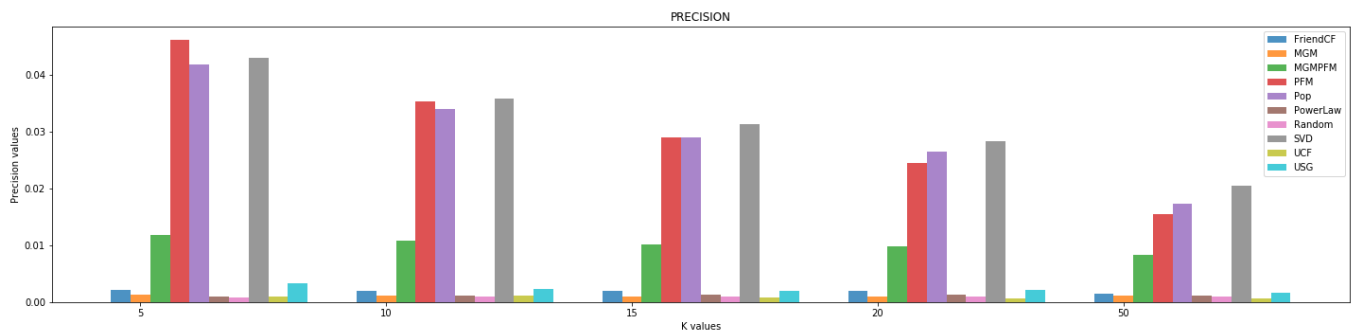


Figura B.5: Baselines de Foursquare en estado avanzado evaluados con Precision.

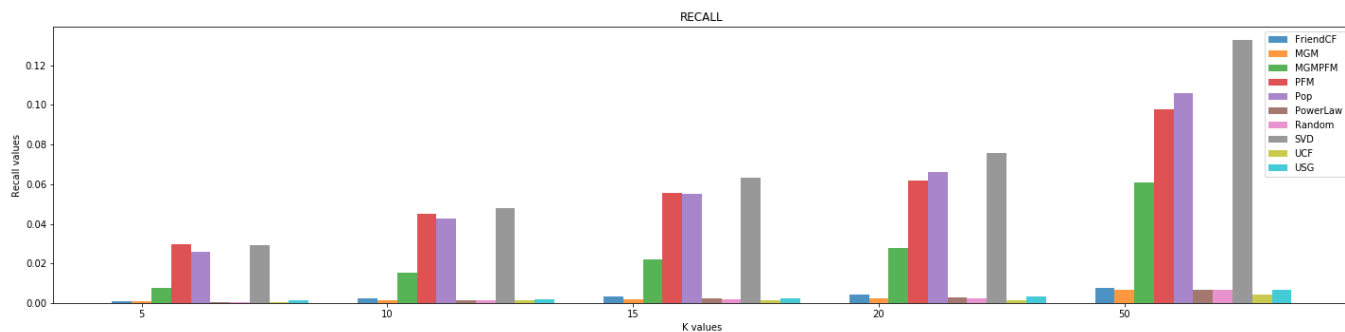


Figura B.6: Baselines de Foursquare en estado avanzado evaluados con Recall.

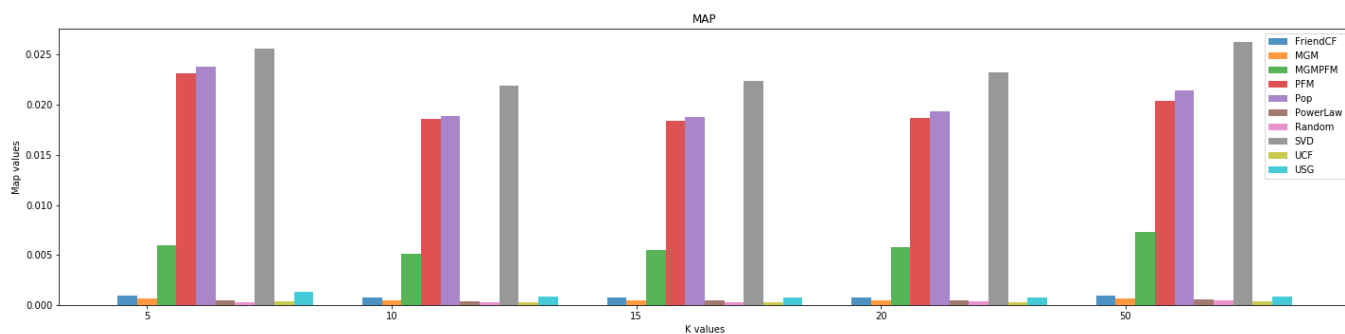


Figura B.7: Baselines de Foursquare en estado avanzado evaluados con mAP.

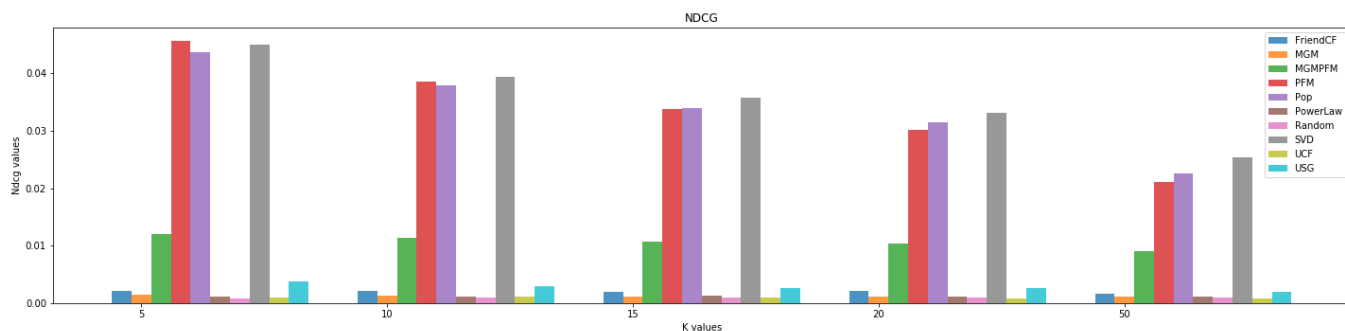


Figura B.8: Baselines de Foursquare en estado avanzado evaluados con nDCG.

Estado final

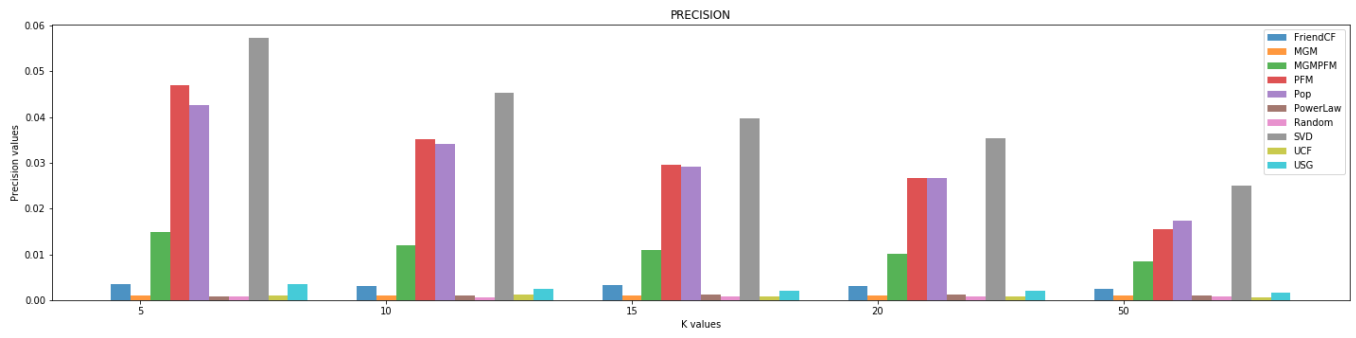


Figura B.9: Baselines de Foursquare en estado final evaluados con Precision.

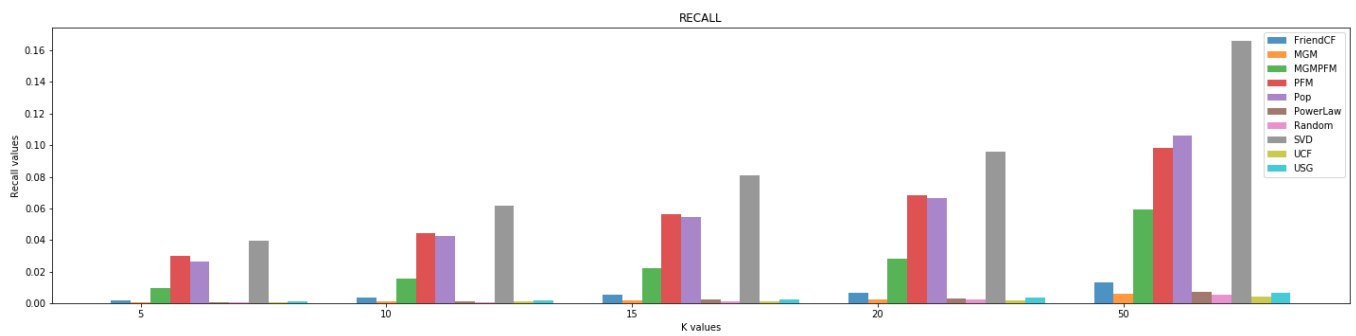


Figura B.10: Baselines de Foursquare en estado final evaluados con Recall.

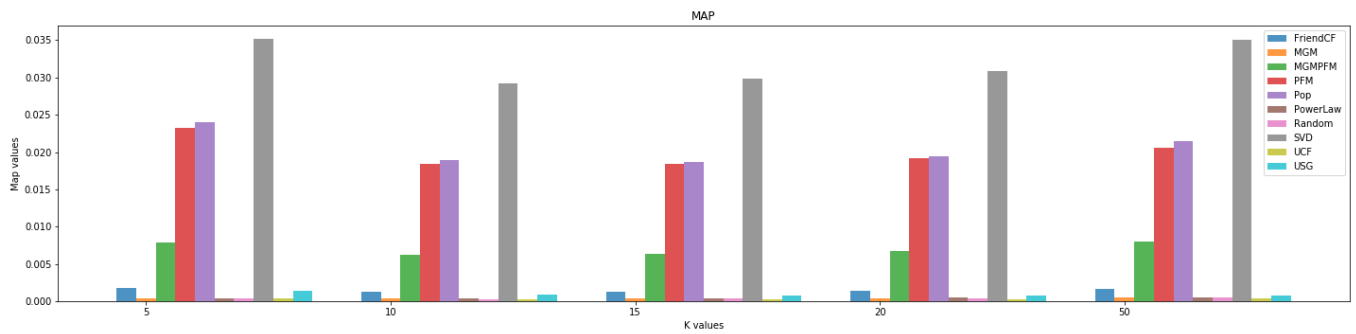


Figura B.11: Baselines de Foursquare en estado final evaluados con mAP.

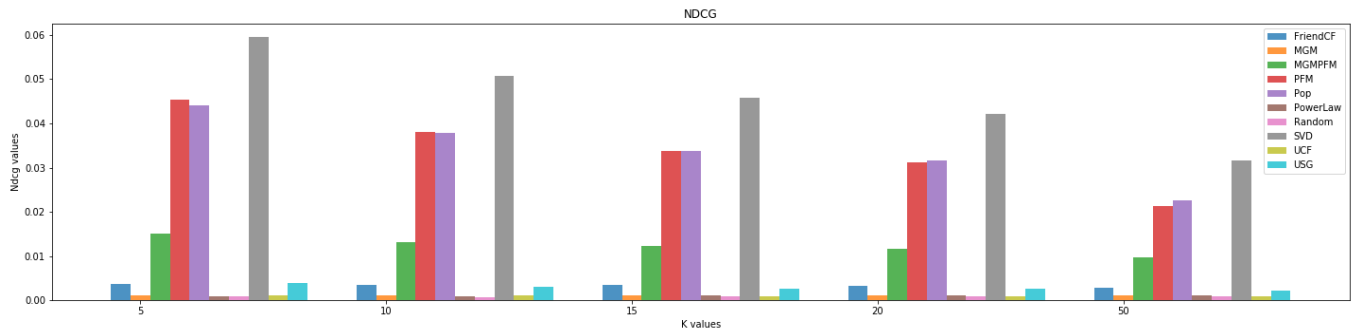


Figura B.12: Baselines de Foursquare en estado final evaluados con nDCG.

## B.2. Gowalla

### Estado nuevo

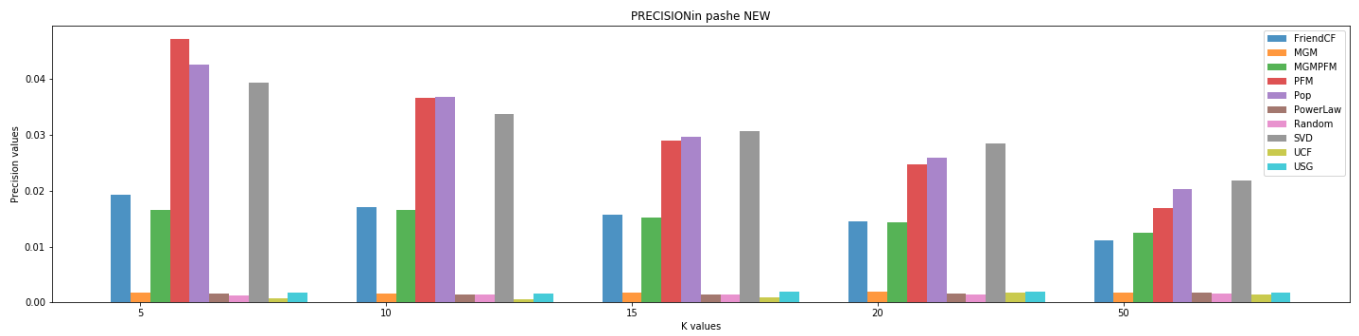


Figura B.13: Baselines de Gowalla en estado nuevos evaluados con Precision.

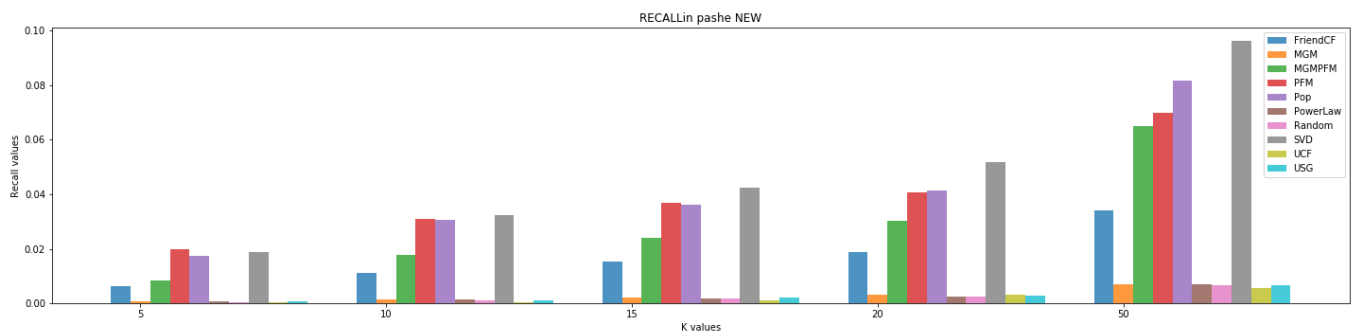


Figura B.14: Baselines de Gowalla en estado nuevos evaluados con Recall.

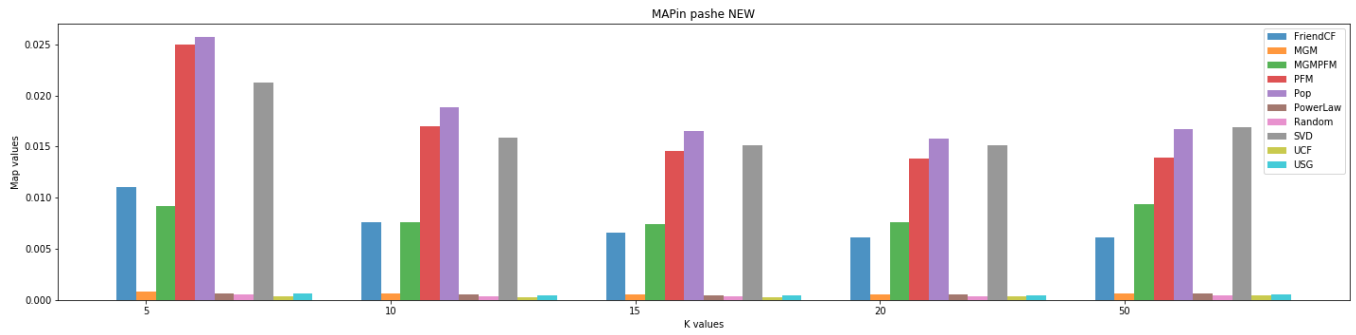


Figura B.15: Baselines de Gowalla en estado nuevos evaluados con mAP.

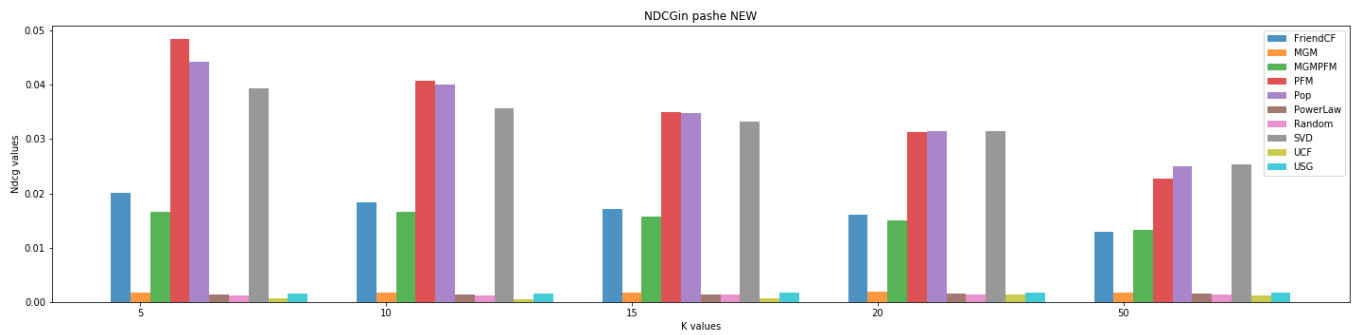


Figura B.16: Baselines de Gowalla en estado nuevos evaluados con nDCG.

## Estado avanzado

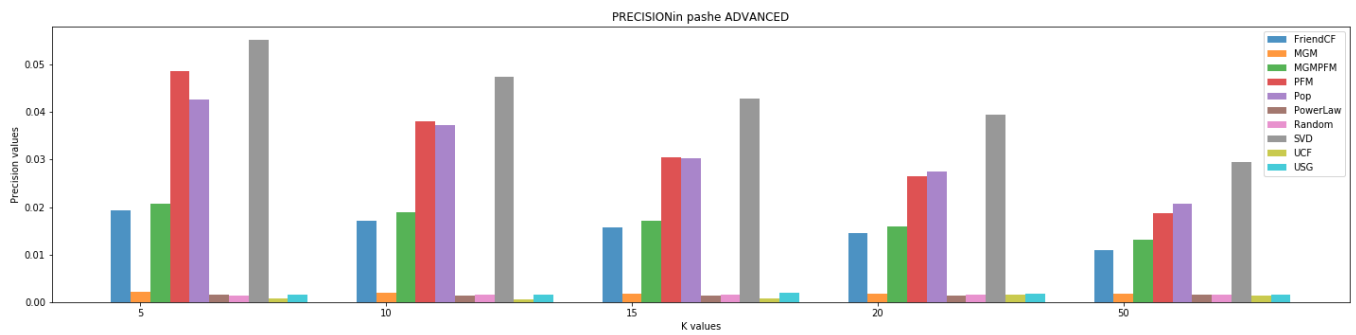


Figura B.17: Baselines de Gowalla en estado avanzados evaluados con Precision.



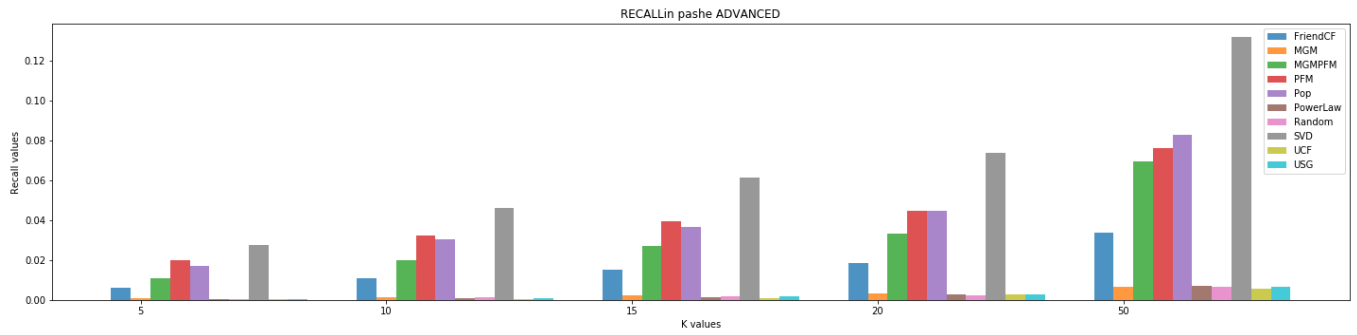


Figura B.18: Baselines de Gowalla en estado avanzados evaluados con Recall.

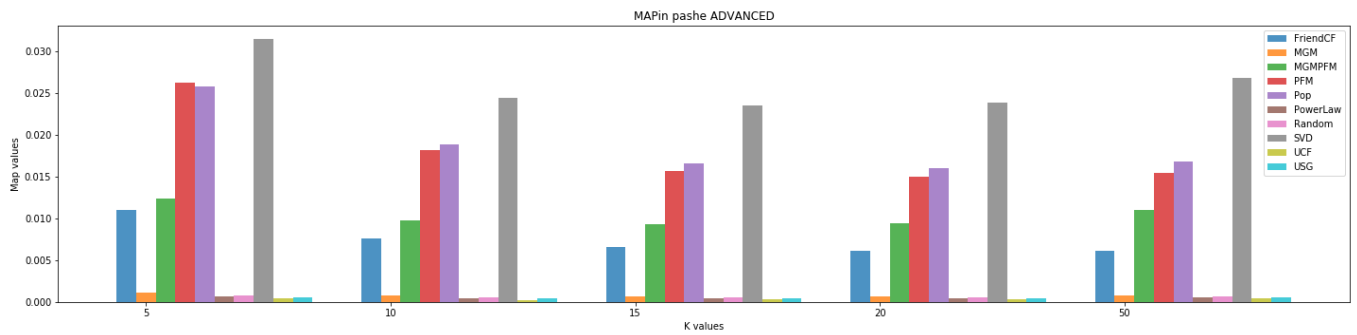


Figura B.19: Baselines de Gowalla en estado avanzados evaluados con mAP.

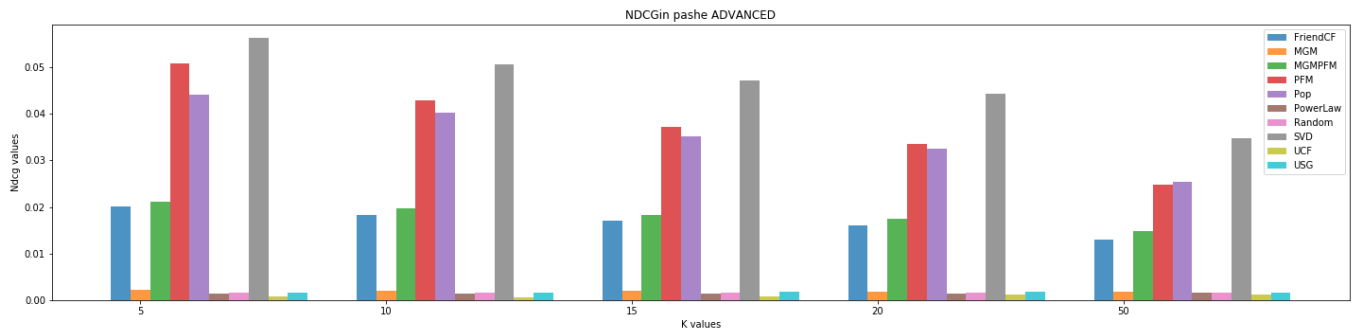


Figura B.20: Baselines de Gowalla en estado avanzados evaluados con nDCG.

Estado final

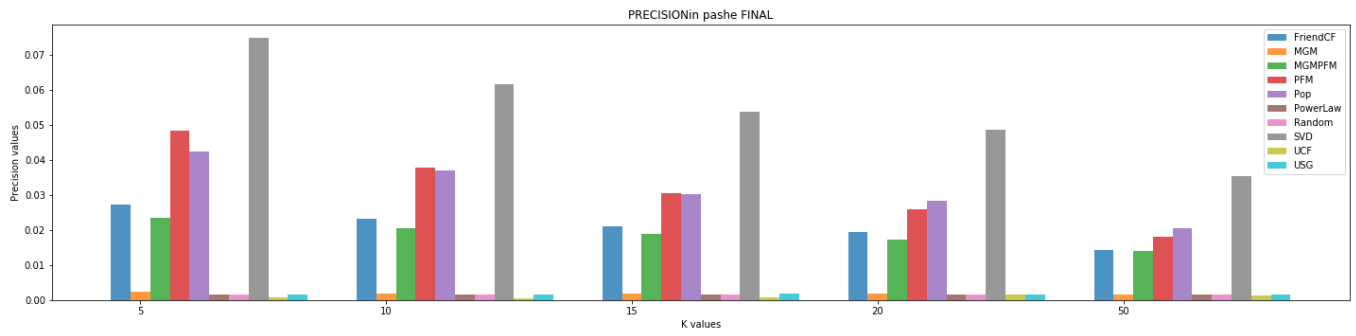


Figura B.21: Baselines de Gowalla en estado final evaluados con Precision.

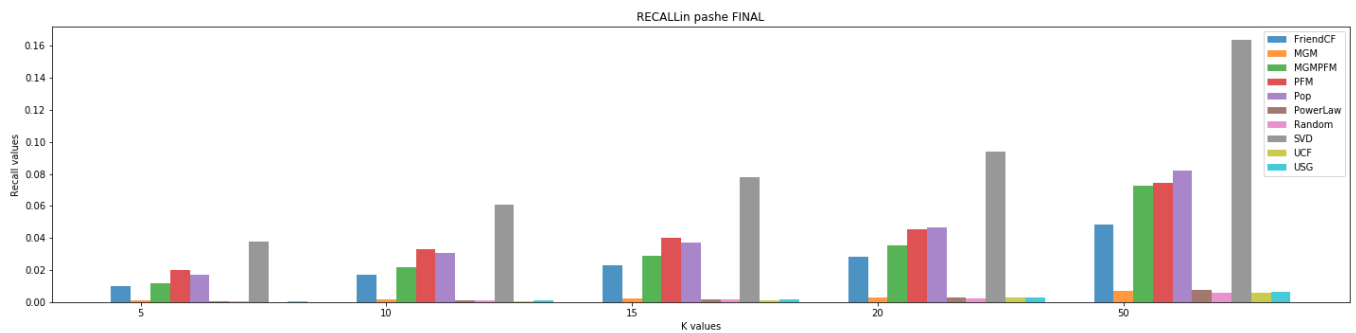


Figura B.22: Baselines de Gowalla en estado final evaluados con Recall.

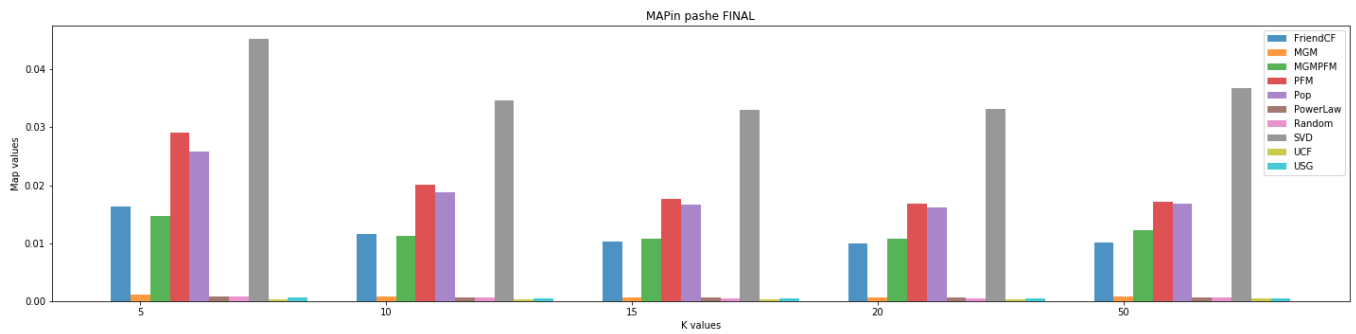


Figura B.23: Baselines de Gowalla en estado final evaluados con mAP.

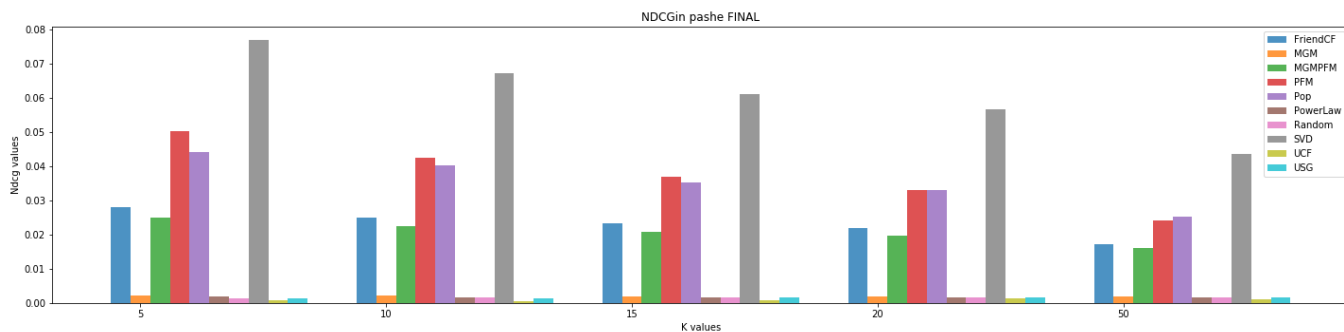


Figura B.24: Baselines de Gowalla en estado final evaluados con nDCG.

## B.3. Yelp

### Estado nuevo

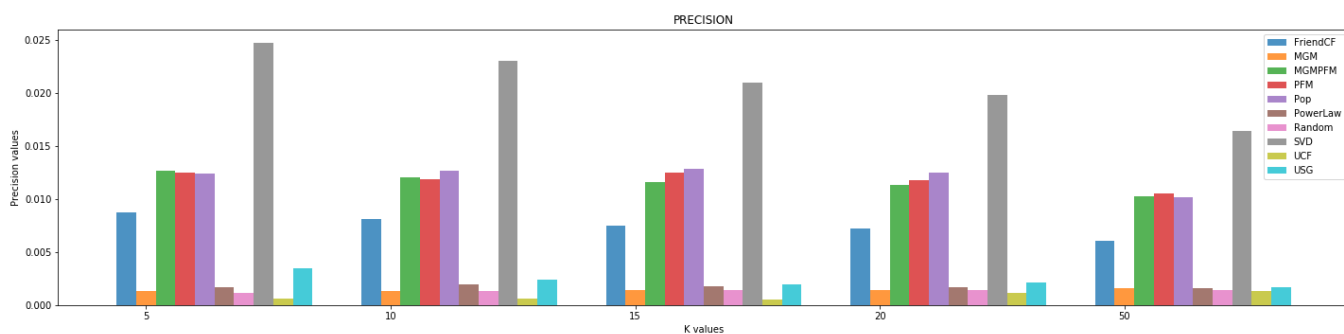


Figura B.25: Baselines de Yelp en estado nuevos evaluados con Precision.

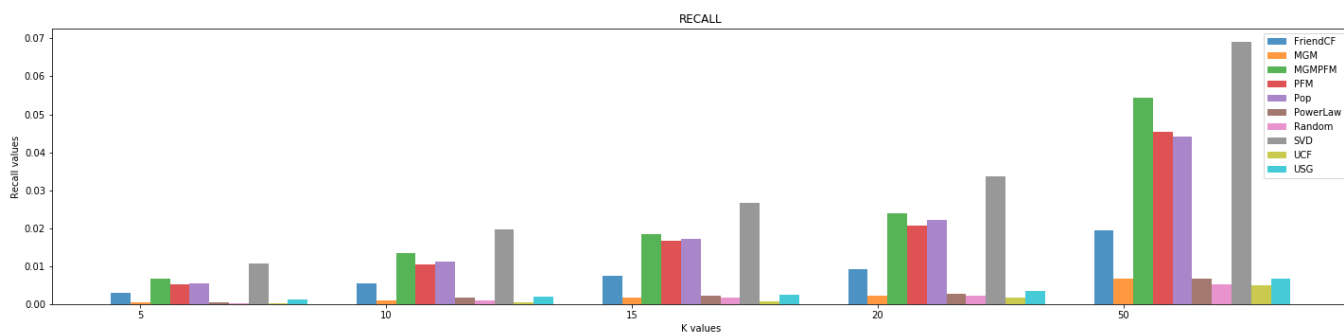


Figura B.26: Baselines de Yelp en estado nuevos evaluados con Recall.

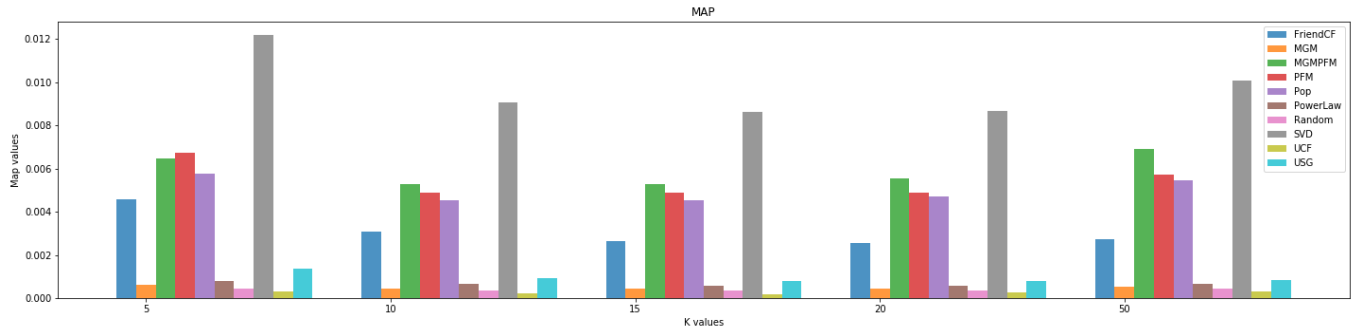


Figura B.27: Baselines de Yelp en estado nuevos evaluados con mAP.

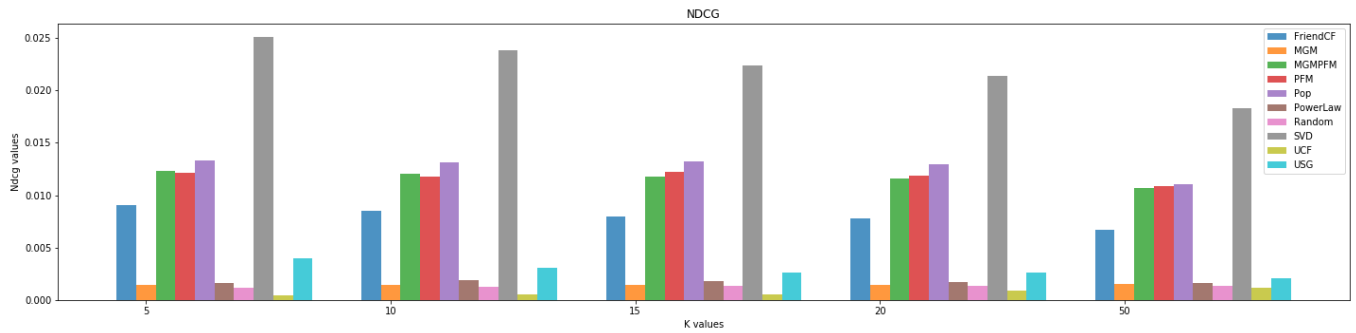


Figura B.28: Baselines de Yelp en estado nuevos evaluados con nDCG.

## Estado avanzado

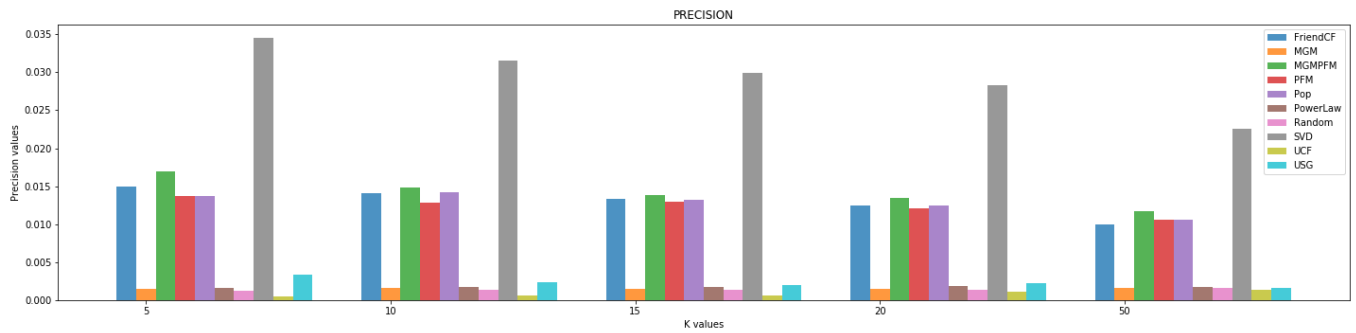


Figura B.29: Baselines de Yelp en estado avanzados evaluados con Precision.

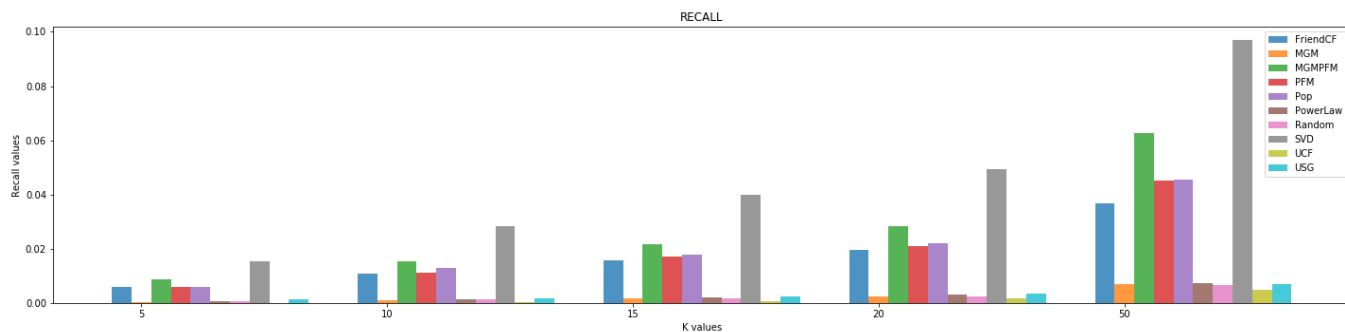


Figura B.30: Baselines de Yelp en estado avanzados evaluados con Recall.

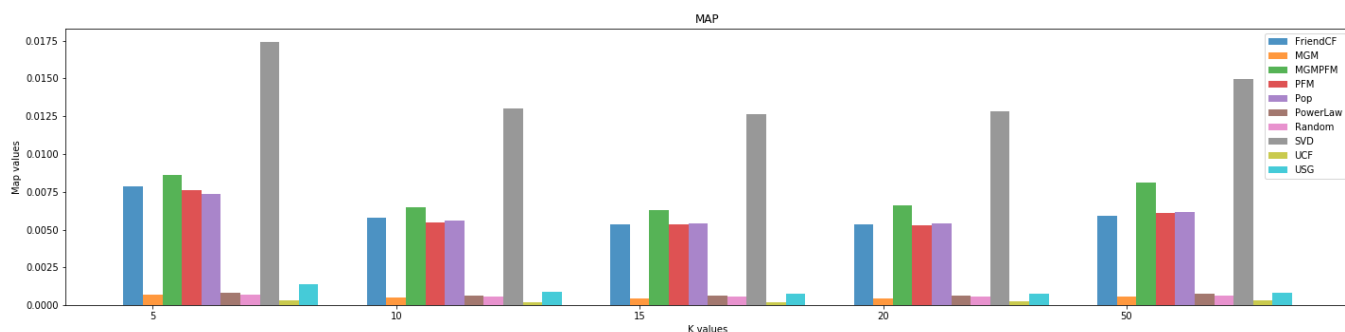


Figura B.31: Baselines de Yelp en estado avanzados evaluados con mAP.

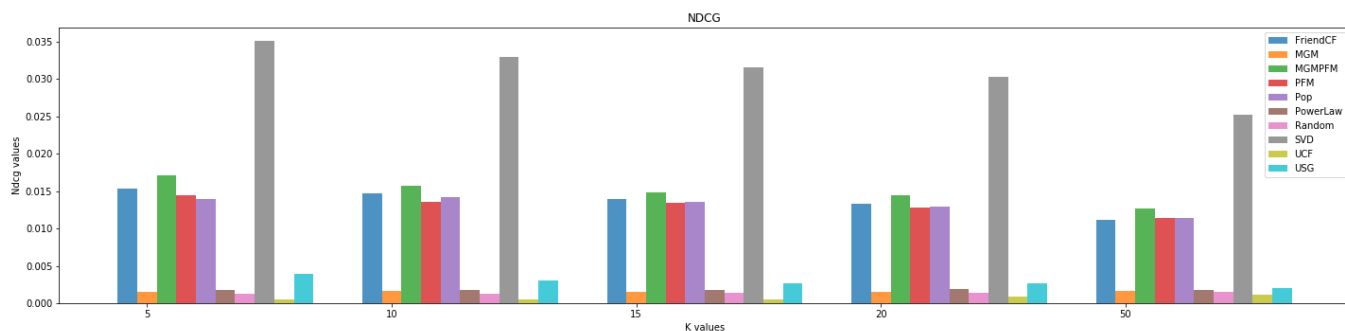


Figura B.32: Baselines de Yelp en estado avanzados evaluados con nDCG.

Estado final

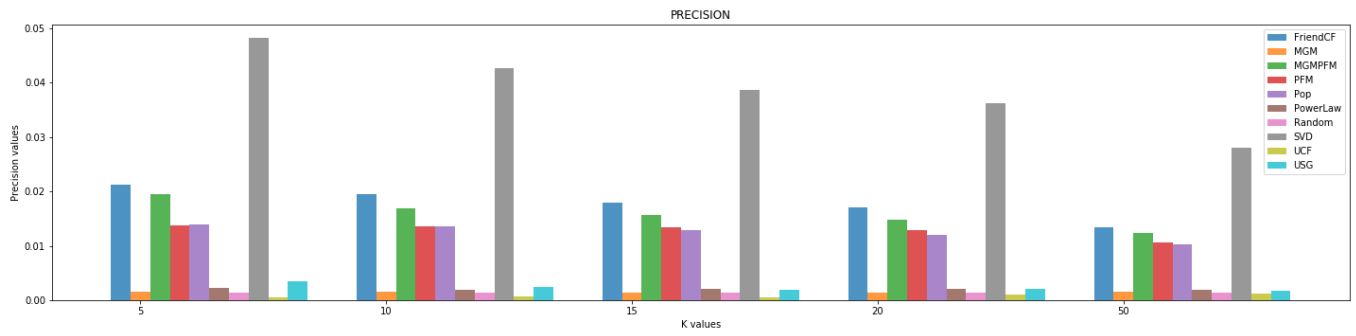


Figura B.33: Baselines de Yelp en estado finals evaluados con Precision.

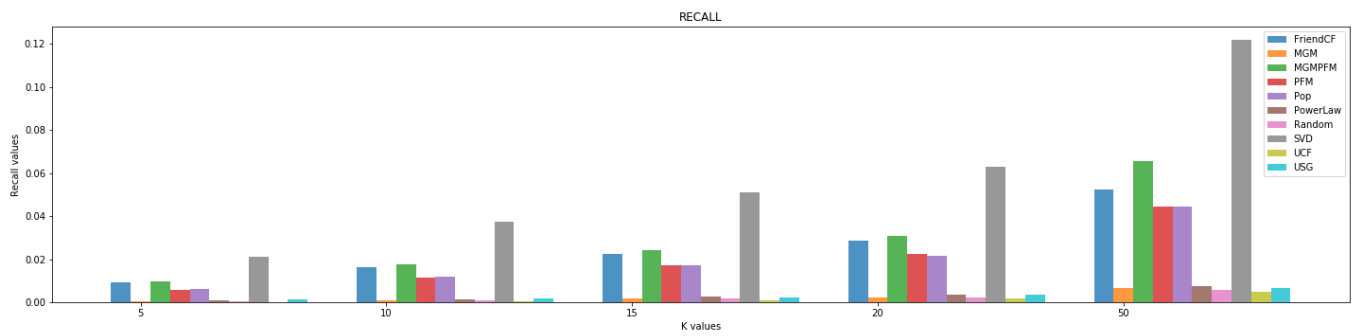


Figura B.34: Baselines de Yelp en estado finals evaluados con Recall.

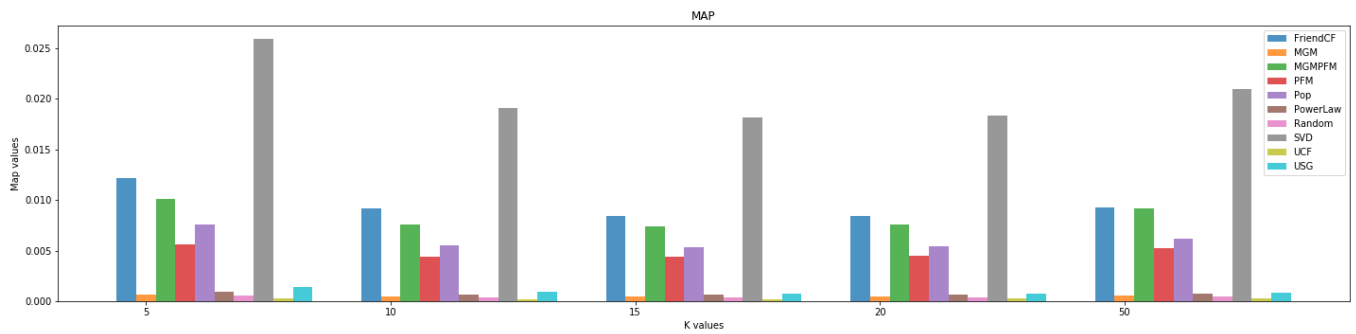


Figura B.35: Baselines de Yelp en estado finals evaluados con mAP.

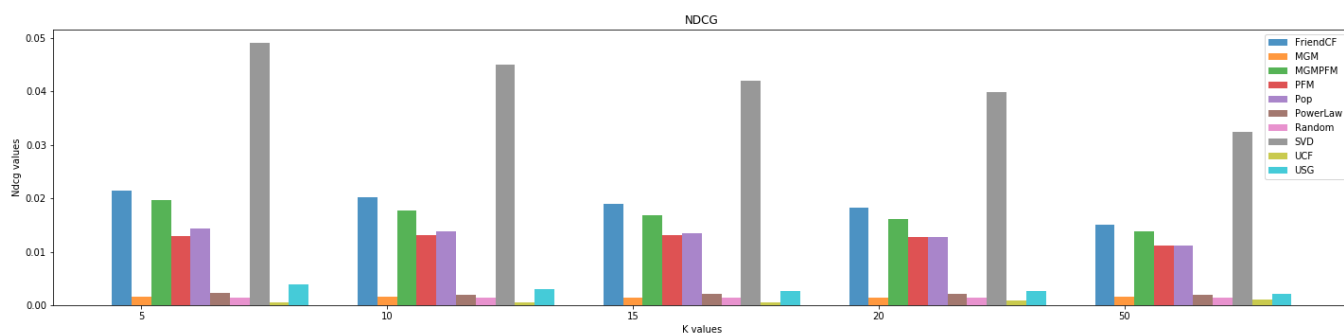


Figura B.36: Baselines de Yelp en estado finals evaluados con nDCG.







UAM

UNIVERSIDAD AUTONOMA  
DE MADRID