



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3633 Sistemas Recomendadores (2025-1)

## Tarea 1

---

### Indicaciones

- Fecha de entrega: **Viernes 11 de abril del 2025 , 20:00 horas.**
- La tarea debe realizarse **en grupos de dos personas**. Para inscribir sus grupos, deberán editar la hoja de cálculo en este [enlace](#) a más tardar el día **viernes 21 de marzo del 2025 a las 20:00** . Quiénes no tengan grupo para esa fecha, serán asignados de manera aleatoria.
- La copia será sancionada con una nota 1,1 en la tarea, además de las sanciones disciplinarias correspondientes.
- Entrega a través de CANVAS, en el buzón de la tarea correspondiente. Solamente deberá entregar uno de los estudiantes del grupo, con los nombres de ambos estudiantes en todos los archivos entregados.
- Cada hora o fracción de atraso descuenta 0,5 puntos de la nota obtenida, llegando a 1,0 en 12 horas. Se considera como entrega el último archivo subido por alguno de los miembros del grupo. No se revisarán tareas que hayan sido subidas con anterioridad a la última.
- Se debe hacer la tarea en Google Colab o en Jupyter Notebooks para facilitar la revisión. Deberán entregar estos notebooks ejecutados como parte de su código.
- Los datos entregados contienen más información de la estrictamente necesaria para el desarrollo de las actividades. Está permitido utilizar esta información extra si se desean mejorar los métodos descritos en las actividades. Sin embargo, debe existir una justificación para el uso de los mismos y además, los métodos de las actividades no pueden perder su estructura.
- **Uso de IA:** Se permite el uso de modelos de lenguaje como ayuda para el desarrollo de la tarea siempre y cuando se cumplan las siguientes condiciones:
  - Se debe compartir la conversación completa de cada una de las sesiones usadas (link de ChatGPT, Claude, Gemini, etc)

- El código y el informe debe ser redactado por ustedes. Se puede usar la IA para realizar consultas de contenido, ortografía y dudas de código, pero no se puede pedir a la IA que realice la tarea (o una parte de esta) por ustedes.
- 

## OBJETIVO

En esta tarea tendrán la oportunidad de poner en práctica sus conocimientos sobre Sistemas Recomendadores. En particular, experimentarán con recomendación no personalizada, basada en feedback implícito y basada en contenido.

## DESCRIPCIÓN DEL CONJUNTO DE DATOS

En esta tarea utilizarán un fragmento del dataset de preferencias de anime de la página [myanimelist.net](#). Puede descargarlo en este [enlace](#). Este set de datos se compone de los siguientes archivos:

- Dataset de entrenamiento (*train.csv*): **27,400** registros que contienen la clasificación entregada por un usuario a un anime de la plataforma. Cada fila contiene el id del usuario, el id del anime y el rating entregado. Si un usuario vio un anime y no lo clasificó, aparece con clasificación -1.
- Dataset de validación (*validation.csv*): **2,312** registros con el mismo formato del set de entrenamiento para la obtención de métricas.
- Dataset de animes (*anime.csv*): **12,294** animes distintos. Cada fila tiene el id del anime, el código, nombre, lista de géneros, tipo de anime, número de episodios, su rating promedio y el número de usuarios unidos al grupo virtual de dicho anime.
- Dataset de géneros de anime (*anime\_genre.csv*): **12,294** anime distintos con su id y sus respectivos géneros registrados en columnas distintas de manera binaria. Si el género de la columna se encuentra en el listado de géneros del anime se marca 1 y en caso contrario se marca 0.
- Dataset de testeo de ratings (*test\_bonus\_rating.csv*): **1,542** registros de usuarios que visualizaron animes. Estos registros tienen la columna de rating entregado vacía y es la que hay que completar para participar por el bonus de la tarea.
- Dataset de testeo de rankings (*test\_bonus\_ranking.csv*): **884** IDs de usuarios. Estos usuarios son los que hay que utilizar para generar las listas de recomendación para el bonus de la tarea.

## ALGORITMOS A UTILIZAR

Para la predicción de los ratings entregados por los usuarios se deben utilizar los siguientes algoritmos:

- User-based collaborative filtering
- Item-based collaborative filtering
- FunkSVD
- SVD++ (opcional)

Para la generación de las listas de recomendación se deben utilizar los siguientes algoritmos:

- Random
- Item-based collaborative filtering
- FunkSVD
- ALS
- BPR
- Factorization Machines

## LIBRERÍAS

Pueden utilizar cualquier librería en python implementadas para recomendación. Las más utilizadas son **PyReclab**, **Surprise**, **Implicit** y **LigthFM**, pero esto queda a su criterio.

## MÉTRICAS DE EVALUACIÓN DE MODELOS

Para la primera parte se deberá utilizar **RMSE** para medir performance, mientras que en la segunda parte se deberá utilizar **Recall@K**, **nDCG** y **MAP** para medir la calidad de las recomendaciones, además de diversidad y novedad de las listas de recomendación. Recordar que se está trabajando con listas de 10 recomendaciones para cada usuario (Top-N recommendation).

## ENTREGABLES

La tarea deberá ser entregada a través de la plataforma CANVAS, se les solicita enviar los siguientes archivos:

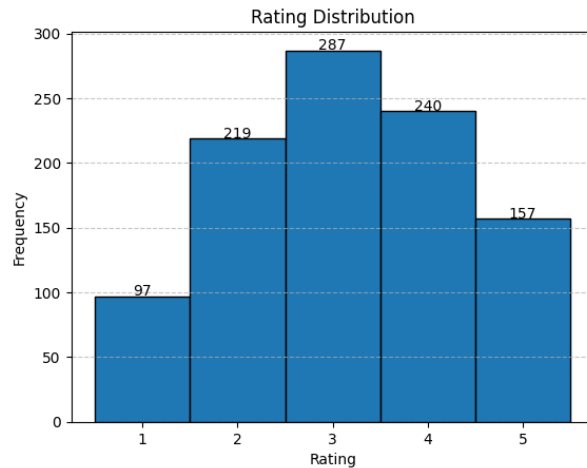


Figura 1: Ejemplo de gráfico mostrando la distribución de ratings en un dataset ficticio.

1. Informe de análisis de los datos entregados y de los métodos utilizados
2. Código de los algoritmos implementados
3. Resultados de la evaluación de las predicciones

A continuación encontrarán una descripción detallada de cada uno de estos.

## 1. Informe (4.2 pts.)

El informe debe estar en formato pdf y deberá ser parte de la entrega en canvas. El informe tiene dos partes.

### 1.1. Análisis de datos (0.8 pts.)

La primera consiste en un análisis de los datos (tablas y gráficos) que incluya, al menos, estadísticas de los usuarios, de los ítems, densidad del dataset (ítems por usuario y usuarios por ítem), como por ejemplo la tabla Cuadro 1. Así mismo debes reportar gráficos de distribución, como las figuras Figura 1 y Figura 2.

### 1.2. Métodos RecSys (3.4 pts.)

En la segunda parte deben comparar y analizar los métodos utilizados respecto a, al menos, su implementación (dificultades y otros), tiempo de ejecución y memoria requeridos, además de las métricas:

- RMSE (Predicción de ratings)
- Recall@10, MAP@10 y nDCG@10 (Generación de rankings)
- Diversidad (cantidad de estilos distintos recomendados en promedio)

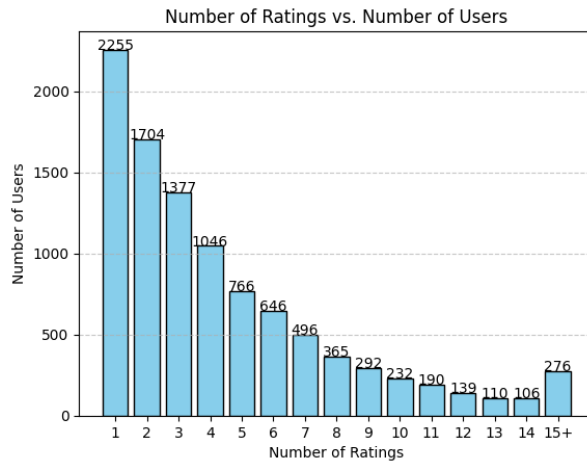


Figura 2: Ejemplo de gráfico mostrando una distribución long-tail del número de ratings por usuario.

Statistic	Training	Validation
Number of Users	80	20
Number of Items	400	100
Total Ratings	960	240
Average Number of Ratings per User	12.0	12.0
Average Number of Ratings per Item	2.4	2.4
Average Rating	3.7	3.8
Rating Standard Deviation	0.5	0.6
Highest Number of Ratings by a User	45	20
Highest Number of Ratings for an Item	14	5
Density (%)	3.0 %	12.0 %

Cuadro 1: Ejemplo de tabla con estadísticas ficticias de un dataset.

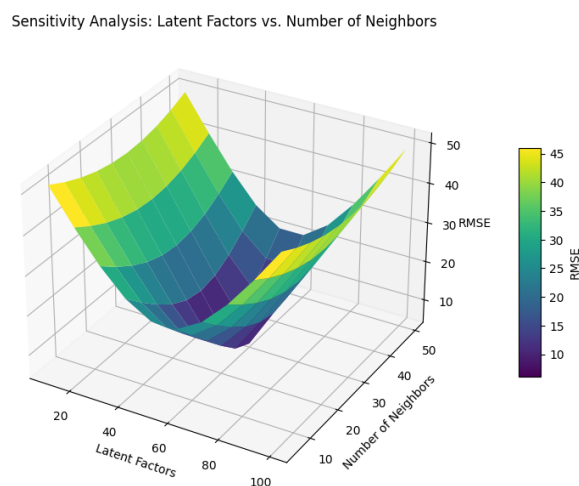
- Novedad (self-information):  $\frac{1}{|U|} \sum_{u \in U} \sum_{i \in L} \log(1/pop_i)$

Para reportar estas métricas debes entrenar con el dataset de entrenamiento y **reportar los resultados sobre el dataset de validación**. Debes reportar con una tabla similar a la tabla Cuadro 2.

**ANÁLISIS DE SENSIBILIDAD.** Se espera que el informe contenga gráficos sobre el impacto de usar diferente número de vecinos (mínimo dos K en métodos KNN) y de factores latentes (mínimo tres números de factores en métodos como funkSVD, ALS y BPR). Considerar también informar sobre impacto de hiperparámetros como learning rate y regularización ( $\lambda$ ). Para esto es ideal gráficos como la Figura 3.

Method	RMSE	Recall@10	NDCG@10	MAP@10	Diversity	Novelty
Method A	0.845	0.175	0.890	0.340	0.620	0.740
Method B	0.812	0.185	0.905	0.350	0.640	0.755
Method C	0.830	0.180	0.895	0.345	0.635	0.750
Method D	0.870	0.170	0.885	0.335	0.610	0.730
Method E	0.910	0.160	0.870	0.320	0.590	0.710

**Cuadro 2:** Ejemplo de tabla comparando métodos de recomendación en varias métricas.



**Figura 3:** Ejemplo de gráfico 3D mostrando el efecto de dos variables sobre RMSE.

## 2. Código (1.8 pts.)

Por cada uno de los métodos solicitados debe entregar el código que permita replicar los resultados obtenidos. Se solicita entregar uno o varios jupyter notebooks que permitan replicar experimentos.

Es obligatorio agregar un archivo README.md que permita entender la estructura de archivos y detalles necesarios para replicar los experimentos realizados.

## 3. Bono por innovación (hasta 0,5pts)

Se entregará un bono de hasta 0,5 pto en la nota final (5 décimas) si se innova en alguno de los algoritmos existentes incorporando el uso de las calificaciones (*ratings*) iguales a  $-1$  que implique una mejora en el rendimiento de alguna de las métricas de ranking o error.

## 4. Competición (Bono para top5)

Para optar a una bonificación, los grupos deberán enviar sus predicciones y recomendaciones completando los archivos `test_bonus.rating.csv` y `test_bonus.rating.json` con los resultados obtenidos por su mejor modelo.

Las bonificaciones se otorgarán a los cinco mejores trabajos según la siguiente distribución:

- 1 puntos para el primer lugar
- 0.75 puntos para el segundo lugar
- 0.5 punto para el tercer lugar
- 0.25 puntos para el cuarto lugar
- 0.1 puntos para el quinto lugar

La evaluación de las predicciones de rating se realizará utilizando el error RMSE, mientras que la evaluación de ranking se basará en el promedio de 3 métricas (NDCG, MAP y recall).

Para esta parte pueden ir más allá de los algoritmos analizados en las partes previas y experimentar con ensambles (ver **ensemble learning**) y utilizar otras técnicas no vistas en clases, como SVD++. A continuación se detallan los archivos:

- `test_bonus_rating.csv`: Este archivo tiene el mismo formato que los sets de entrenamiento y validación, pero con los valores de ratings vacíos. La columna **rating** deberá ser llenada con sus predicciones de ratings para los correspondientes usuarios y ánimes.

user_id	anime_id	rating
16049	20787	
68087	2904	
3815	2403	
32702	20	
51882	770	
67551	2966	
17698	2847	
71897	19815	
18182	16417	
51919	24833	
47109	29998	
1959	8769	
20587	14527	

Figura 4: Inicio del archivo `test_bonus_rating.csv`

- `test_bonus_ranking.csv`: Este archivo contiene una lista de IDs de usuarios y columnas vacías con enumeración del 1 al 10. Se deben completar las columnas vacías con el top 10 de recomendaciones para el usuario del ID correspondiente, siendo la recomendación de la columna 1 la más deseada a recomendar y la de la columna 10 la menos deseada.

user_id	1	2	3	4	5	6	7	8	9	10
7345										
66816										
38660										
8820										
2632										
49094										
19593										
11867										
2951										
21926										

**Figura 5:** Inicio del archivo test\_bonus\_raking.csv