



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

IIC3633 Sistemas Recomendadores (2024-2)

Tarea 1: Recomendacion de Cervezas

Indicaciones

- Fecha de entrega: **Viernes 13 de septiembre de 2024, 23:59 horas.**
 - La tarea debe realizarse **individualmente o en grupos de máximo dos personas**. En Canvas se encuentran disponibles los grupos para inscribirse en la tarea en la sección Personas → Tarea 1.
 - La copia será sancionada con una nota 1,1 en la tarea, además de las sanciones disciplinarias correspondientes.
 - Entrega a través de pestaña grupos en la plataforma CANVAS.
 - Cada hora o fracción de atraso descuenta 0,5 puntos de la nota obtenida, llegando a nota 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.
 - El código de la tarea debe ser realizada en uno o varios archivos .ipynb, el cual debe ser entregado con todas sus celdas corridas. Se recomienda realizar la tarea en Google Colab para evitar posibles problemas de incompatibilidad de versiones.
 - Además del código, se debe entregar un informe en formato PDF. Detalles de este informe en la sección correspondiente de este enunciado.
 - 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.
-

OBJETIVO

En esta tarea tendrán la oportunidad de poner en práctica sus conocimientos sobre Sistemas Recomendadores, utilizando distintas técnicas para poder predecir la evaluación de los usuarios y elaborar listas de recomendaciones para estos. Adicionalmente, se espera que sean capaces de evaluar distintas métricas para las recomendaciones y que realicen una comparación entre dichos métodos.

INSTRUCCIONES

La tarea tiene dos partes. En ambas partes debes utilizar y comparar los algoritmos que se mencionan en la sección de Algoritmos a utilizar. Los datos necesarios para cada una de estas partes están descritos en la sección Datos.

1. La primera parte consiste en predecir ratings (completar archivo .csv)
2. La segunda parte tiene por objetivo generar una lista con 10 (diez) recomendaciones para cada uno de los usuarios presentes en la lista (completar archivo .json)

Descripción del conjunto de datos

En esta tarea utilizarán un dataset de calificaciones de cervezas entregados por distintos usuarios, el cual pueden descargar [aquí](#).

El dataset con el que trabajarán consiste en:

- Dataset de train (*training_set.csv*): **35,534** registros con **8,320** usuarios distintos. Cada fila contiene el id del usuario, el id del artículo, el tipo de cerveza y la calificación entregada.
- Dataset de validación (*validation_set.csv*): **8,845** registros con **3,410** usuarios distintos. Cada fila contiene el id del usuario, el id del artículo, el tipo de cerveza y la calificación entregada.
- Dataset de test de rating (*rating_template_fill.csv*): **2,945** registros distintos a completar. Estas son las calificaciones a predecir para los items por parte de los usuarios de la parte 1, donde hay que completar el rating que entregará el usuario.
- Dataset de test de ranking (*ranking_template_fill.json*): **1,953** llaves distintas a completar. Estas son las listas de recomendación a completar para los usuarios de la parte 2.

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:

- 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** e **implicit**, 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:

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.

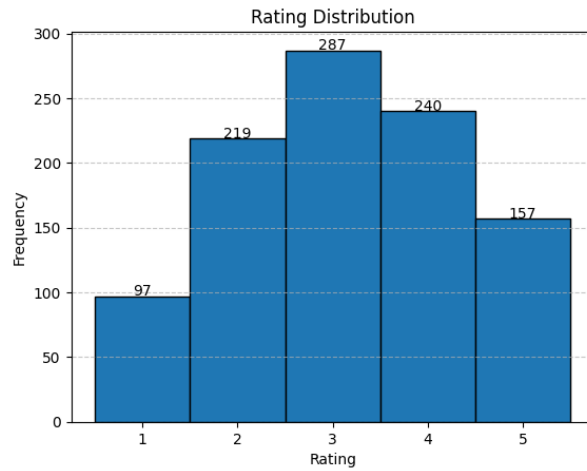


Figura 1: Ejemplo de gráfico mostrando la distribución de ratings en un dataset ficticio.

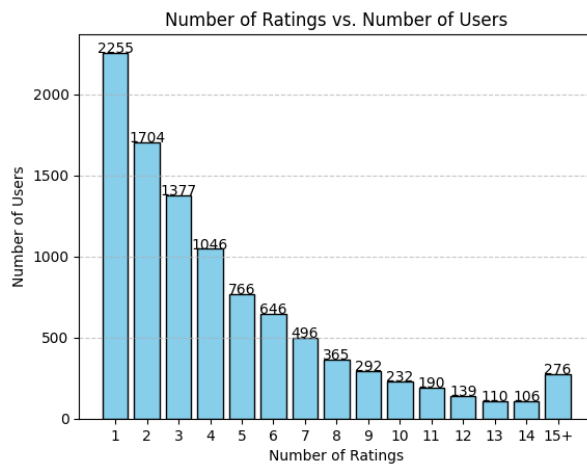


Figura 2: Ejemplo de gráfico mostrando una distribución long-tail del número de ratings por usuario.

Informe

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

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.

En la segunda parte deben comparar y analizar los métodos utilizados respecto a, al menos, su implementación (dificultades y otros), tiempo de ejecución, procesamiento 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)

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.

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.

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

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

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

Código

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.

Sensitivity Analysis: Latent Factors vs. Number of Neighbors

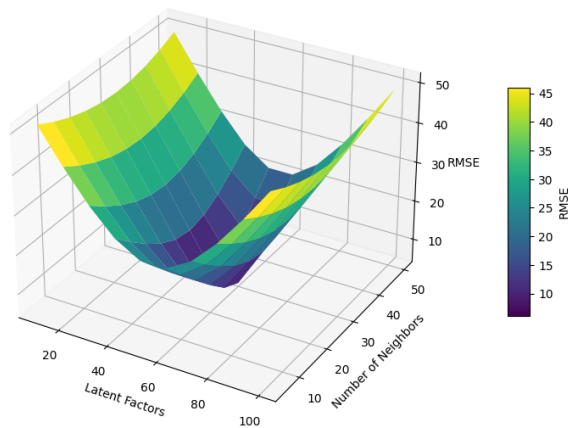


Figura 3: Ejemplo de gráfico 3D mostrando el efecto de dos variables sobre RMSE.

Competición (Bono)

Para ganar una bonificación al puntaje deberán enviar sus predicciones y recomendaciones rellenando los archivos `rating_template_fill.csv` y `ranking_template_fill.json` con los resultados de su mejor modelo. 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:

- `rating_template_fill.csv`: Este archivo contiene 2945 tuplas de `itemID`, `styleID`, `Rating`, `userID`.

```
itemID;styleID;Rating;userID
39289;14756;;715
3939;1417;;4944
47695;14879;;4890
7348;1199;;5324
37403;1199;;5544
47658;1199;;2244
77006;22288;;3462
1622;394;;453
19960;1199;;4101
20478;1199;;7347
74870;27079;;1960
11757;1199;;7639
727;263;;2149
34146;1199;;2916
4904;1417;;7735
```

Figura 4: Inicio del archivo `rating_template_fill.csv`

La columna **Rating** deberá ser llenada con sus predicciones de ratings para los correspondientes usuarios e items.

- `ranking_template_fill.json`: Este archivo contiene un diccionario de 1953 llaves cuyo valor es una lista de strings.

```
{
  "3022": [],
  "4049": [],
  "1122": [],
  "1156": [],
  "3386": [],
  "1818": [],
  "3510": [],
  "5224": [],
  "6323": [],
  "8176": [],
  "1260": [],
  "2725": [],
  "4971": [],
  "2049": [],
  "1406": []
}
```

Figura 5: Inicio del archivo ranking_template.fill.json

Las claves corresponden a los IDs de los usuarios para los cuales deberán crear una recomendación de 10 cervezas en formato de ranking. La lista correspondiente a cada usuario deberá ser reemplazada por una lista de los IDs de las 10 cervezas recomendadas en el orden en que fueron recomendadas (la recomendación principal debe ir en la primera posición).

```
"2049": ["39289", "3939", "47695", "7348", "37403", "47658", "77006", "22288", "1622", "19960"]
```

Figura 6: Ejemplo de ranking para un usuario