Informe de Taller 1

Cristian C. Moreno Mojica, Juan J. Ovalle Villamil, Maria C. Rodríguez Niño
Curso MINE4201 – Sistemas de recomendación
Universidad de los Andes, Bogotá, Colombia
c.morenom@uniandes.edu.co, jj.ovalle@uniandes.edu.co,
mc.rodriguezn12@uniandes.edu.co
Fecha de presentación: 17 de marzo del 2021

Tabla de contenido

1) Introducción

En el siguiente documento se describe el proceso realizado para construir y evaluar un modelo colaborativo de recomendación de información basado en datos reales, se establecieron las fases de reprocesamiento de datos, construcción de modelos usuario-usuario como ítem-ítem, así mismo para la visualización y operacionalización se estableció la aplicación Web mediante la cual se valida el desempeño de los modelos.

Para el presente taller se complementaron las temáticas vistas en el curso reconociendo en la práctica algunas de ventajas y desventajas de cada uno de los modelos, así como la interpretación de las métricas que indican la calidad de las predicciones

2) Desarrollo

a) Pre-procesamiento de datos

i) Transforme los datos correspondientes a la interacción entre usuarios e ítems, implementando una estrategia para convertir estos datos en unos que sean compatibles con los modelos vistos en clase. Justifique en el informe sus decisiones en este paso.

Inicialmente se realiza un reconocimiento de la estructura de los dataset que se usaran en el taller, el primero denominado “userid-profile.tsv” corresponde a los daros de los perfiles de los usuarios (#id; gender; age; country; registered), el segundo denominado “userid-timestamp-artid-artname-traid-traname.tsv” indica las interacciones realizadas por cada uno de los usuarios respecto a los ítems o canciones

Out[2]:

	#id	gender	age	country	registered
0	user_000001	m	NaN	Japan	Aug 13, 2006
1	user_000002	f	NaN	Peru	Feb 24, 2006
2	user_000003	m	22.0	United States	Oct 30, 2005
3	user_000004	f	NaN	NaN	Apr 26, 2006
4	user_000005	m	NaN	Bulgaria	Jun 29, 2006

Estructura dataset ““userid-profile.tsv”

Out[3]:

	UserId	TimeStamp	ArtId	ArtName	TraId	TraName
0	user_000001	2009-05-04T23:08:57Z	f1b1cf71-bd35-4e99-8624-24a6e15f133a	Deep Dish	NaN	Fuck Me Im Famous (Pacha Ibiza)-09-28-2007
1	user_000001	2009-05-04T13:54:10Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Composition 0919 (Live_2009_4_15)
2	user_000001	2009-05-04T13:52:04Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Mc2 (Live_2009_4_15)
3	user_000001	2009-05-04T13:42:52Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Hibari (Live_2009_4_15)
4	user_000001	2009-05-04T13:42:11Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Mc1 (Live_2009_4_15)

Estructura data set ““userid-timestamp-artid-artname-traid-traname.tsv”

A partir de la data de itentarcciones se crea el objeto que denominaremos “user_interact” el cual refleja las interacciones de los usuarios y los ítems. Dicha relación se puede ver desde los dos tipos de ítems disponibles, es decir, canciones y artistas. Así, **cada registro de la tabla representa una canción de un artista x escuchada por un usuario y**. Con esto en mente, se puede llegar a insights relevantes para entender el conjunto de datos. Por ejemplo, si se desea ver las canciones o artistas más escuchados, basta con realizar un conteo por el id del artista o el id de la canción respectivamente, para la implementación de los modelos diseñados se seleccionaron las columnas de UserId, ArtId inicialmente

El conjunto de datos posee 992 usuarios, 107296 artistas únicos y 960403 canciones únicas en el conjunto de datos. Por limitaciones de las maquinas utilizadas, se trabajarán los artistas como ítems.

Con base en Jawaheer, Szomszor y Kostkova (2010) (https://core.ac.uk/download/pdf/207051652.pdf) decidimos utilizar 3 alternativas para medir los ratings de los usuarios:

- La frecuencia total de los ítems por usuario.
- El logaritmo de la frecuencia total de los ítems por usuario.
- Una normalización de la frecuencia total de los ítems por usuario la cual está definida como la frecuencia sobre el número total de artistas que un usuario ha escuchado.

	UserId	ArtId	frecuencia	log_frecuencia	normalizada
0	user_000001	00c73a38-a449-4990-86ca-5088dde1b8df	2	0.301030	0.00346
1	user_000001	012a77c9-c897-494f-87d0-0a730996494d	1	0.000000	0.00173
2	user_000001	014ba96b-b8da-49e3-8a2b-b720ae42e84c	3	0.477121	0.00519
3	user_000001	01ce7548-dab4-4ca6-8dfc-8e2e4b50d461	4	0.602060	0.00692
4	user_000001	03282c56-8a24-42f4-8bfc-96188933aefa	4	0.602060	0.00692

Tabla que presenta los usuarios con las frecuencias de Artistas escuchados

De acuerdo a los valores presentados se observa que la frecuencia que más se acerca a 0 es la frecuencia Normalizada por tal razón se selecciona para ser usada en los modelos como el rating.

ii) Tome los datos compatibles con modelos colaborativos y pártalos en dos conjuntos: un grupo de datos le sirve para construir el modelo y el resto para medir sus predicciones. Sepárelos en archivos distintos.

Con el fin de comparar las diferentes frecuencias se genera una función para mostrar cual métrica representa mejor los ratings de cada usuario, luego se realiza la separación de train y test para cada una de las alternativas con un porcentaje de 0.8 para train y 0.2 para test.

```
---------  Alternativa: frecuencia ---------
La escala para la alternativa frecuenciava de 1 hasta 26496
<class 'surprise.dataset.DatasetAutoFolds'>
<class 'surprise.trainset.Trainset'>
<class 'list'>
realizado.
---------  Alternativa: log_frecuencia ---------
La escala para la alternativa log_frecuenciava de 0.0 hasta 4.42
<class 'surprise.dataset.DatasetAutoFolds'>
<class 'surprise.trainset.Trainset'>
<class 'list'>
realizado.
---------  Alternativa: normalizada ---------
La escala para la alternativa normalizadava de 0.0 hasta 236.57
<class 'surprise.dataset.DatasetAutoFolds'>
<class 'surprise.trainset.Trainset'>
<class 'list'>
realizado.
```

3) Construcción de modelos colaborativos usuario-usuario

a) Construya un modelo colaborativo basado en perfiles de usuario con la primera parte de los datos de ratings.

   Inicialmente se construye el modelo usuario-usuario con las métricas de similitud de Coseno y de Person, ya que el uso de la métrica de Jaccard requirió un tratamiento especial.

b) Realice las predicciones de relevancia para los usuarios e ítems que encuentra en la segunda parte de los datos.

Se construye los modelos implementando las librerías de KNNBasic y se toma como K = 20 vecinos cercanos, este parámetro es común para todos los modelos obteniendo los siguientes valores RMSE:

Alternativa	Métrica	RMSE
Frecuencia	Pearson	146.4401
Frecuencia	Coseno	109.9802
Log Frecuencia	Pearson	0.6379
Log Frecuencia	Coseno	0.6450
Normalizada	Pearson	0.3488
Normalizada	Coseno	0.3957

c) Compare su predicción de rating con el efectivamente encontrado en el dataset. Establezca una forma de evaluar globalmente sus distancias en las predicciones que refleje la calidad de las mismas

Para evaluar las predicciones de cada uno de los modelos se utiliza el RMSE. Así, se quiere encontrar el modelo con el menor RMSE posible. Desde esta primera aproximación del modelo usuario-usuario, se puede evidenciar que la mejor alternativa es la normalizada, luego la log_frecuencia y por último la frecuencia. Estos resultados tienen lógica con lo encontrado por Jawaheer, Szomszor y Kostkova (2010). Para el caso la mejor alternativa fue Normalizada con Person.

d) Varíe la estrategia de selección de vecinos por umbral de similitud y por número de vecinos. Revise cuál es el impacto al variar estos parámetros

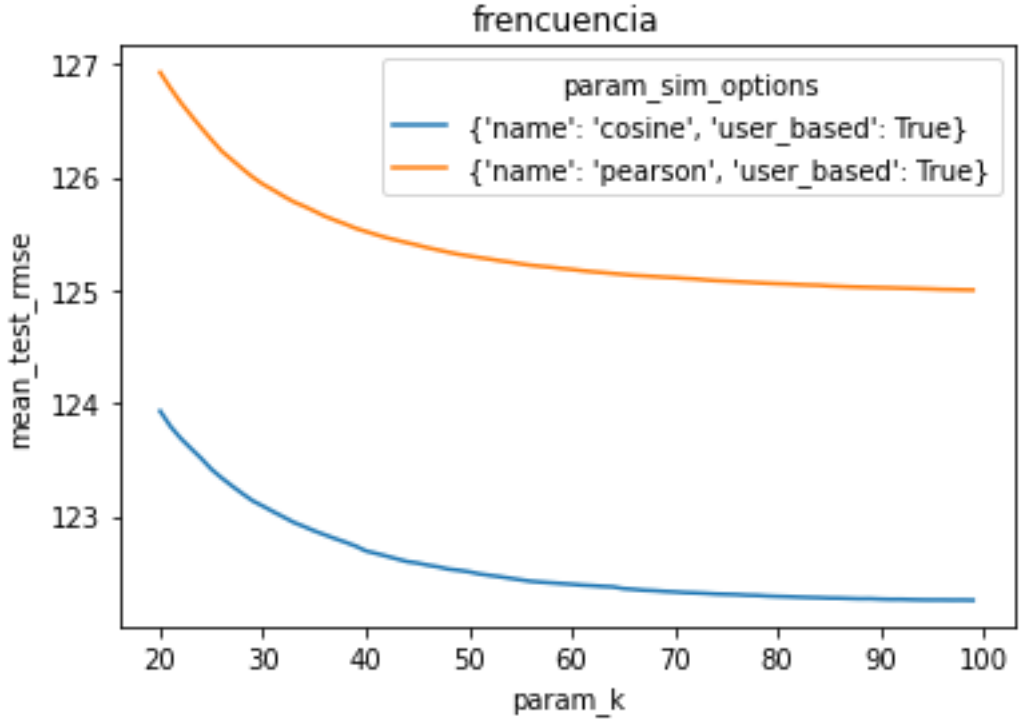
Se define una búsqueda de grilla con 5 folds de validación cruzada que va a recorrer las siguientes opciones:

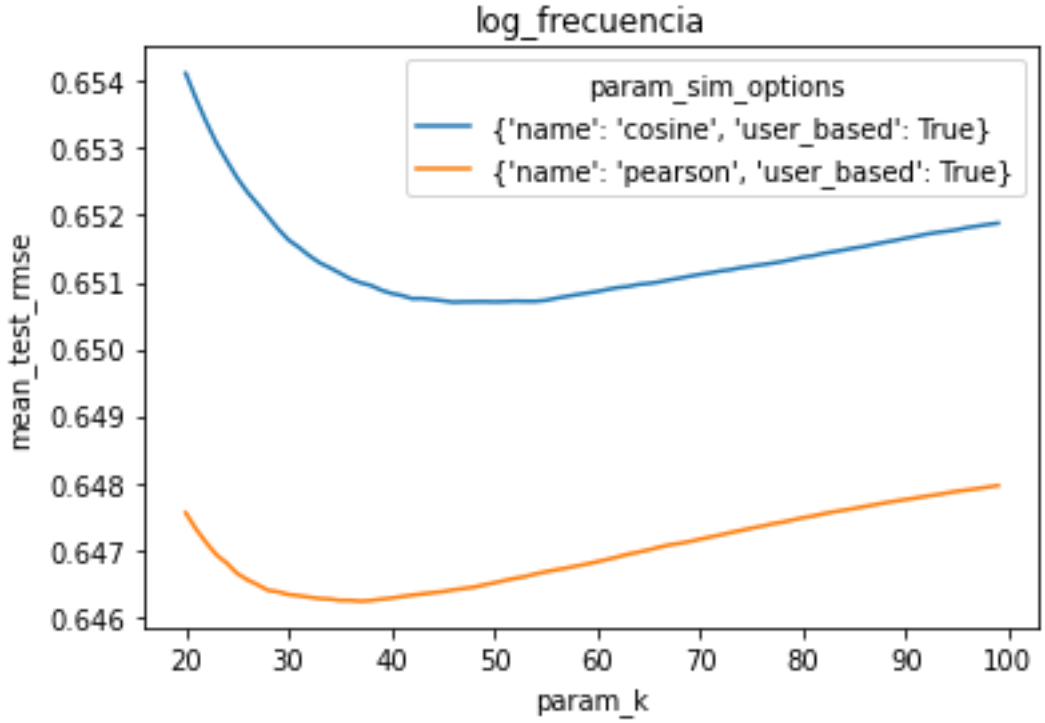
- Todos los k's dentro de [20,99]
- Métrica de similitud coseno y Pearson.

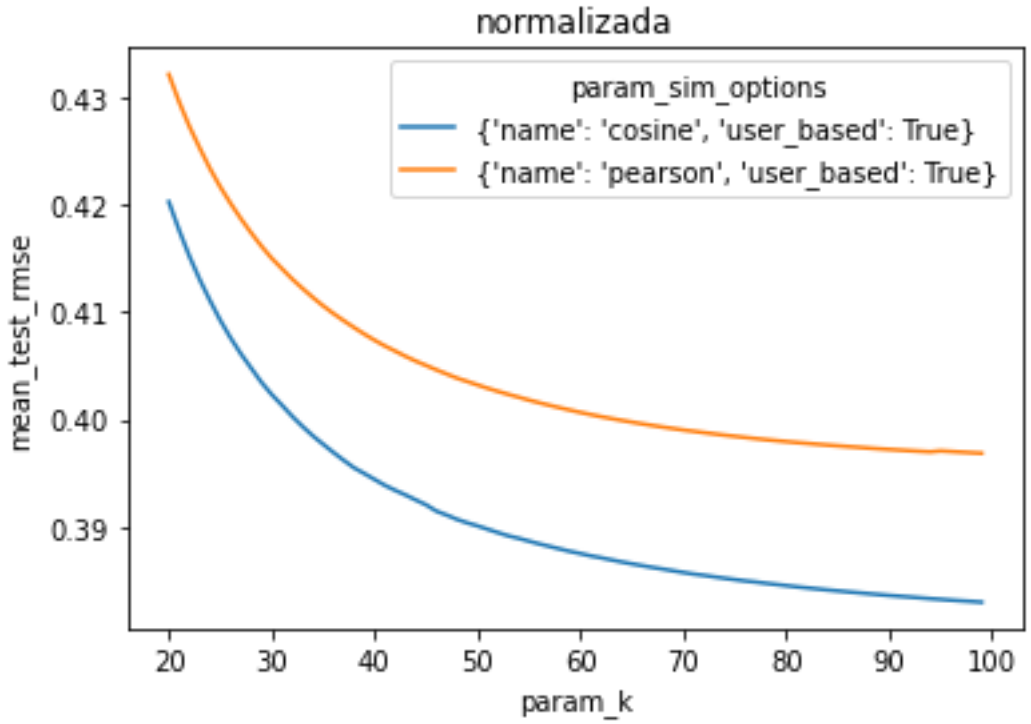
Los parámetros del mejor modelo son guardados mediante la librería joblib dado que cada búsqueda de grilla demora 6 horas aproximadamente. Además, se pueden guardar los resultados en un conjunto de datos.

Con estas iteraciones se revisa cuál de los K entre 20 y 99 presentan un mejor RMSE con las diferentes medidas de similitud, lo que indicaría que el modelo sea más preciso para su predicción
Como resultados se observa las siguientes tabla y gráficas, la cuales muestra el mejor K para cada medida de similitud con cada alternativa de frecuencia, y su respectiva distribución.

	alternativa	param_k	param_sim_options	mean_test_rmse
318	normalizada	99	{'name': 'cosine', 'user_based': True}	0.383043
355	log_frecuencia	37	{'name': 'pearson', 'user_based': True}	0.646242
158	frencuencia	99	{'name': 'cosine', 'user_based': True}	122.261087







e) Medida de similitud Jaccard

i) Modelo Usuario- Usuario –Medida Normalizada

Dado que se evaluó con las medidas de Coseno y Person que la mejor alternativa es la Frecuencia Normalizada, Jaccar se aplica usando dicha medida

Para la medida de Jaccard se realizó la construcción de una función que incluyo los siguientes pasos:

En la preparación de datos fue necesario validar cuales usuario presentaban solo calificación en 1 ítem, dado que generaba problema para el proceso de estratificación, encontrando solo 1 usuario con esta característica el cual fue eliminado.

Se estableció el proceso de división de train y test, pero aplicando el proceso de estratificación

Se pivotea la data obteniendo en las filas los usuarios y en las columnas los ítems.

A continuación, se establece un loop a tareas la cual recorre la data y ejecuta la función que calcula el nivel de similitud y su estimación por usuario obteniendo los siguientes datos

:

	UserId	ArtId	normalizada	Jaccard
0	user_000253	ba550d0e-adac-4864-b88b-407cab5e76af	0.004515	0.001402
1	user_000860	e77b7ecf-c417-4124-b255-a3a927bb604e	0.003077	0.000601
2	user_000116	d7fbbe08-e3fc-43d1-ab47-082ba7a4202f	0.000976	0.000000
3	user_000783	d4d17620-fd97-4574-92a8-a2cb7e72ce42	0.006993	0.004970
4	user_000096	451f9db1-f75f-44f9-b218-f8bdf22035a1	0.001353	0.000000
5	user_000034	1c89bbcc-a253-452a-9181-546eb536e06f	0.037954	0.000125
6	user_000155	789e5db3-502b-4f13-8039-88c70e053fa5	0.001691	0.000919
7	user_000702	94460cce-560f-48dd-a7b9-41d90fb85a21	0.000149	0.000000
8	user_000943	5508631d-697f-4839-a669-06637e5bcb90	0.001597	0.000000
9	user_000074	393ce5ee-4550-48e2-97f9-50a47a74bdc1	0.000424	0.000000

Alternativa	K	RMSE
Normalizada	20	0.712071
	40	07120725
	60	0.71207256
	80	0.712072564
	100	0.712072564

ii) Modelo ítem-ítem –Medida Normalizada

Para la construcción del modelo ítem-ítem se utilizó de la misma forma la frecuencia normalizada y se aplicó la misma estrategia del Usuario –Usuario solo que al pivotear la data se ubicó en las filas los ítems y en las columnas los usuarios, el resto de la dinámica se mantuvo igual y se obtuvieron los siguientes datos

Alternativa	K	RMSE
Normalizada	20	0.0936
	40	0.0936
	60	0.0936
	80	0.0936
	100	0.0936

La variación de numero de K no afecto significativamente las predicciones

4) Construcción de modelos colaborativos ítem – ítem

Antes de realizar el mismo proceso, es importante tener en cuenta que:

- Por limitaciones de las maquinas solo se podrá utilizar una parte de los ítems para realizar los modelos.
- Esta selección se hará a partir del conteo de ratings de cada ítem

Como se puede ver, para usuarios si era posible utilizar toda la base ya que eran únicamente 992 usuarios, por ende, solo tenía que calcular 992 similitudes. Sin embargo, para ítems este proceso se tendría que hacer 107295 veces.

Para subsanar este punto se realiza una agrupación por ArtID ordenando por frecuencia de forma descendente y se toma de dicha lista el 10 % de ítems con mayor calificación

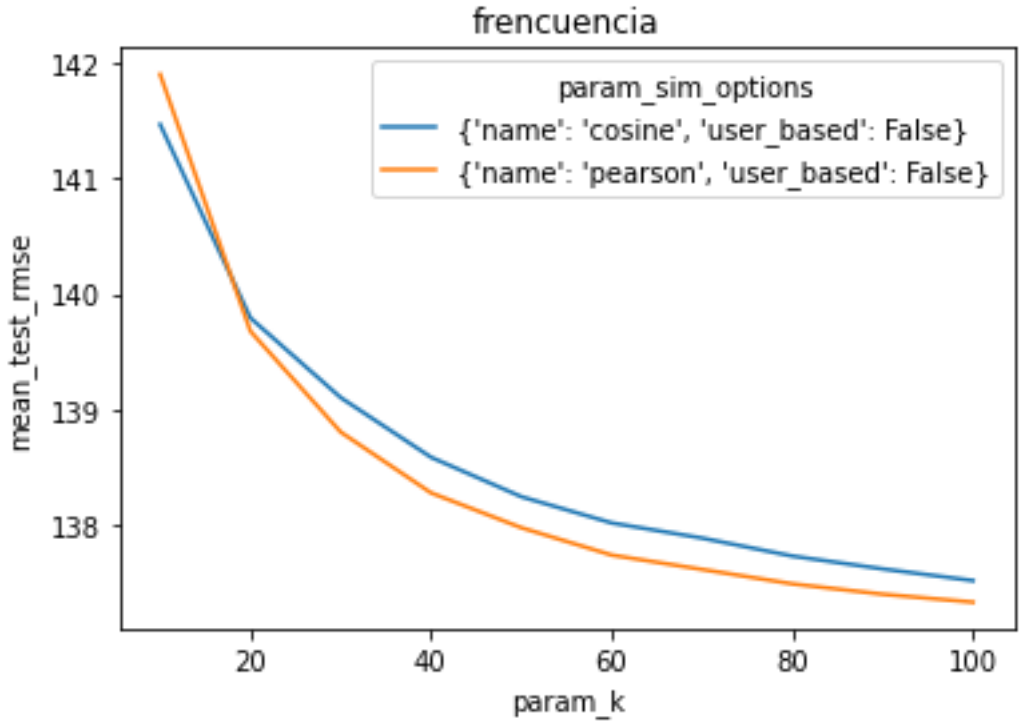
A continuación, se implementan los modelos ítem-ítem con la librera KNNbasic con un K =20 obteniendo los siguientes resultados

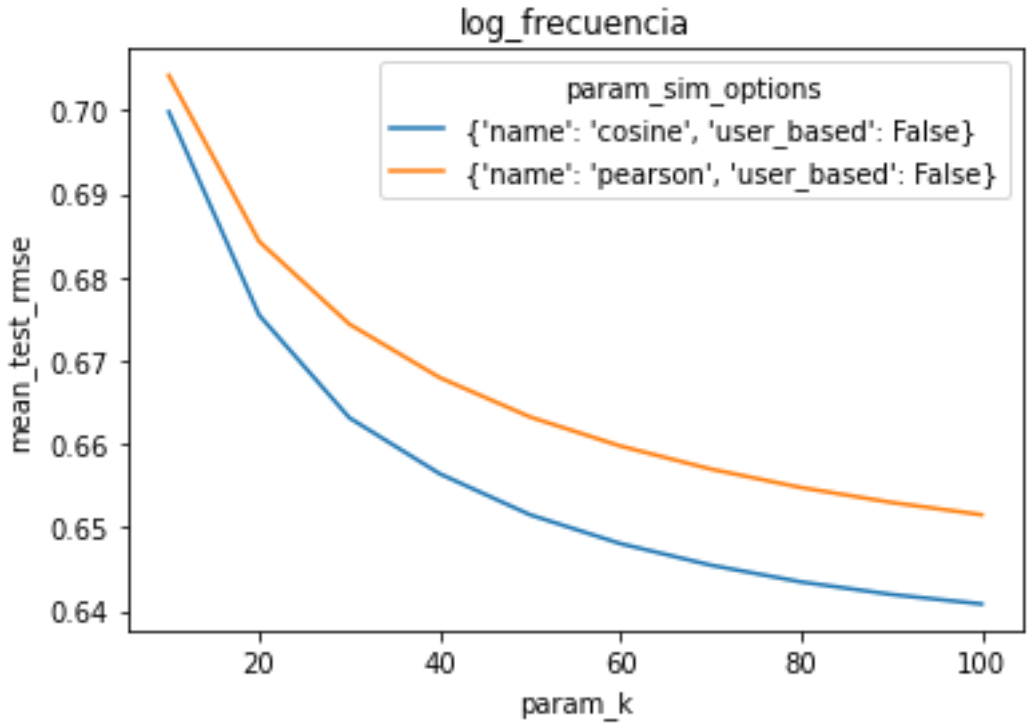
	Alternativa	Metrica Similitud	RMSE
5	normalizada	cosine	0.252194
4	normalizada	pearson	0.297666
3	log_frecuencia	cosine	0.676680
2	log_frecuencia	pearson	0.685043
0	frecuencia	pearson	129.969449
1	frecuencia	cosine	140.544081

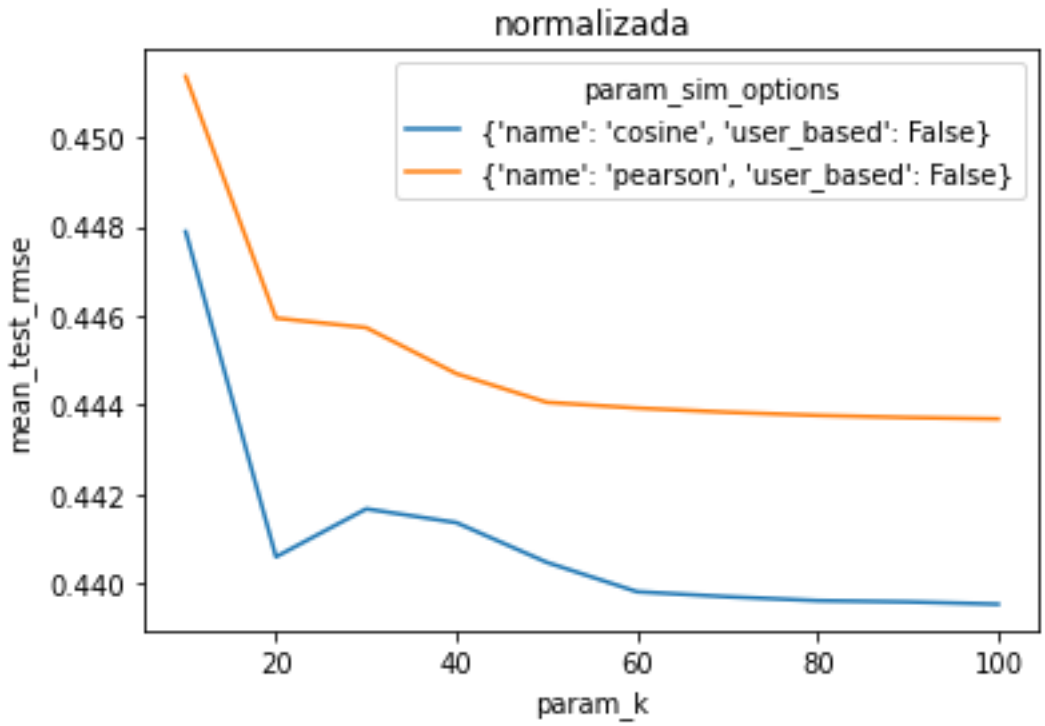
Se realiza la variación de los K en múltiplos de 10 hasta 100 y se aplican las alternativas y las diferentes medidas de similitud obteniendo los siguientes resultados con sus distribuciones

```
{'rmse': {'k': 100, 'sim_options': {'name': 'pearson', 'user_based': False}},
 'mae': {'k': 40, 'sim_options': {'name': 'cosine', 'user_based': False}}}

{'rmse': {'k': 100, 'sim_options': {'name': 'cosine', 'user_based': False}},
 'mae': {'k': 100, 'sim_options': {'name': 'cosine', 'user_based': False}}}

{'rmse': {'k': 100, 'sim_options': {'name': 'cosine', 'user_based': False}},
 'mae': {'k': 20, 'sim_options': {'name': 'cosine', 'user_based': False}}}
```

	alternativa	param_k	param_sim_options	mean_test_rmse
38	normalizada	100	{'name': 'cosine', 'user_based': False}	0.439530
58	log_frecuencia	100	{'name': 'cosine', 'user_based': False}	0.640830
19	frencuencia	100	{'name': 'pearson', 'user_based': False}	137.335848





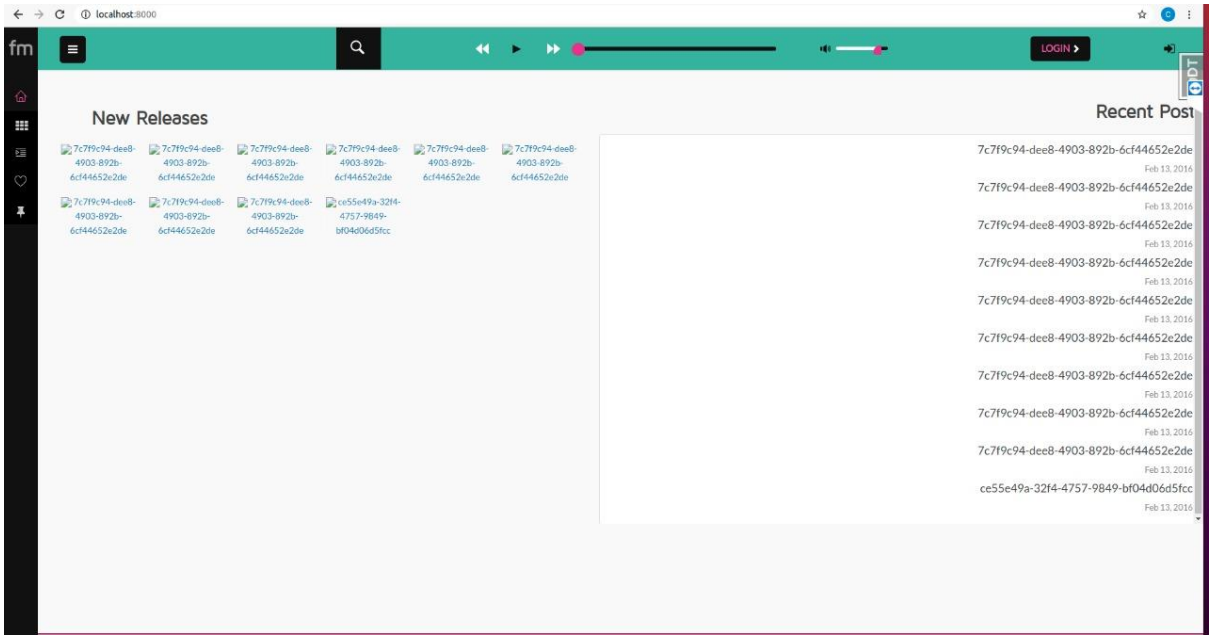


5) Construya una aplicación Web interactiva sencilla que permita interactuar con sus experimentos.

La aplicación consta de

Respecto a la arquitectura general de la aplicación fue desarrollada en Django con el Front desarrollado en HTML y JavaScrip, el motor de base de datos es RDS de Amazon

La aplicación consta de un botón de Login y salida, diferencial en el proceso de un usuario nuevo o antiguo. Posee un panel en el cual se muestran las recomendaciones de lo más escuchado por el usuario activo. De la misma forma presenta a una tabla donde se muestran los ítems y los ratings para cada ítem y cada usuario.



6) Análisis de resultados

a) Explique las alternativas contempladas para el procesamiento de datos y por qué se eligió la alternativa implementada.

El proceso de procesamiento de datos realmente es una de la fase fundamental en la construcción de modelos de recomendación dado que a partir de como están estructurado los datos se deben identificar, daros inconsistentes, faltantes o tipos de datos no validos entre algunos. Para el caso específico se realizaron varias acciones

Se validó del listado de usuario cuáles de ellos no contaba sino con la calificación de un artista, dado que esto generaba dificultades en la aplicación de la estratificación de los conjuntos de datos de train y test

Así mismo, después de la separación de datos de train y set se validó que el conjunto de set realmente contara con calificaciones en train para evitar inconsistencias en la comparación de la predicción.

De la misma forma se revisaron las alternativas para la generación del rating relacionadas con la Frecuencia la frecuencia log y la normalizada de los con las 3 métricas de similitud, encontrado que la mejor alternativa e la que arroja el RMSE más cercano a 0

Modelo	Alternativa	Métrica	RMSE
Usuario-Usuario	Frecuencia	Person	125.032565

Usuario-Usuario	Frecuencia	Coseno	138.853218
Usuario-Usuario	Frecuencia Log	Person	0.636512
Usuario-Usuario	Frecuencia Log	Coseno	0.645505
Usuario-Usuario	Normalizada	Person	0.412365
Usuario-Usuario	Normalizada	Coseno	0.381981
Usuario-Usuario	Normalizada	Jaccard	0.71207256
ítem-ítem	Frecuencia	Person	129.969449
ítem-ítem	Frecuencia	Coseno	140.544081
ítem-ítem	Frecuencia Log	Person	0.685043
ítem-ítem	Frecuencia Log	Coseno	0.676680
ítem-ítem	Normalizada	Person	0.297666
ítem-ítem	Normalizada	Coseno	0.252194
ítem-ítem	Normalizada	Jaccard	0.0936

Concluyendo que el mejor modelo para implementar es el modelo que presenta el menor RMSE

ítem-ítem	Normalizada	Coseno	0.252194

Respecto al usuario nuevos, como no se puede correr el modelo de recomendació dado que el modelo corre diario, en la aplicación se creó un nuevo modelo que valida la fecha de registro del sistema y si coincide con la fecha del sistema, genera recomendación a partir de la interacción que realice con los ítems nuevos por las similitudes entre los ítems los cuales se obtuvieron de matriz de similitudes del mismo.

Explique la arquitectura general del sistema implementado (arquitectura, tecnologías utilizadas, modelo de datos)

Respecto a la arquitectura general de la aplicación fue desarrollada en Django con el Front desarrollado en HTML y JavaScrip, el motor de base de datos es RDS de Amazon

La aplicación consta de un botón de Login y salida, diferencial en el proceso de un usuario nuevo o antiguo. Posee un panel en el cual se muestran las recomendaciones de lo más escuchado por el usuario activo. De la misma forma presenta a una tabla donde se muestran los ítems y los ratings para cada ítem y cada usuario.

El modelo de datos está inicialmente con Django y a partir de las migraciones se llevó RDS, está conformado por los siguientes elementos:

Tabla Artista: ArtID, Name_Art
Tabla Recuequery: UserId, Artid, Create
Tabla Rating: UserId, ArtId, rating
Similary: ArtID, baseArtID, rating
Clase picture: ArtId, Path (donde está alojada la imagen del artista)

El despliegue de la aplicación ha sido un poco complejo por el nivel de afinación de las herramientas utilizadas, pues se realizaba mediante prueba y error

7) Conclusiones

Lecciones aprendidas

Desde lo técnico alguno de los aprendizajes se reconoce en el uso de la librería train_test_split1 para hacer las particiones de conjunto de train y set con el parámetro de stratify. El aprendizaje se centró en reconocer que existe la función con el mismo nombre desde dos librerías (Surprise y Sklearn)

Desde lo conceptual se evidencia un alto impacto del proceso de estratificación en las predicciones del modelo, dado que esto afecta en el nivel de precisión para el modelo, desde cómo se distribuyen los datos.

Quedo totalmente clara la dinámica de un modelo usuario-usuario e item-item y cómo identificar a partir de las métricas cual puede ser el modelo más recomendado.