

# Tema 7: Aplicaciones relacionadas con KDD

## Minería de Datos

# Sistemas de recomendación

## ¿Qué es un sistema de recomendación?

- Un “**recomendador**” es un sistema que selecciona un *producto* que, si se compra, maximiza el valor tanto para el comprador como para el vendedor en un determinado momento del tiempo.
- Para hacer las recomendaciones, el sistema analiza y procesa información histórica de los usuarios (edad, compras previas, calificaciones...), de los productos o de los contenidos (marcas, modelos, precios, contenidos similares...) y la transforma en conocimiento accionable, es decir, **predice qué producto puede ser interesante para el usuario y para la empresa**.
- Los recomendadores, además, tienen cierto nivel de autonomía a la hora de presentar las recomendaciones al usuario final.

### > ¿Por qué se usa?

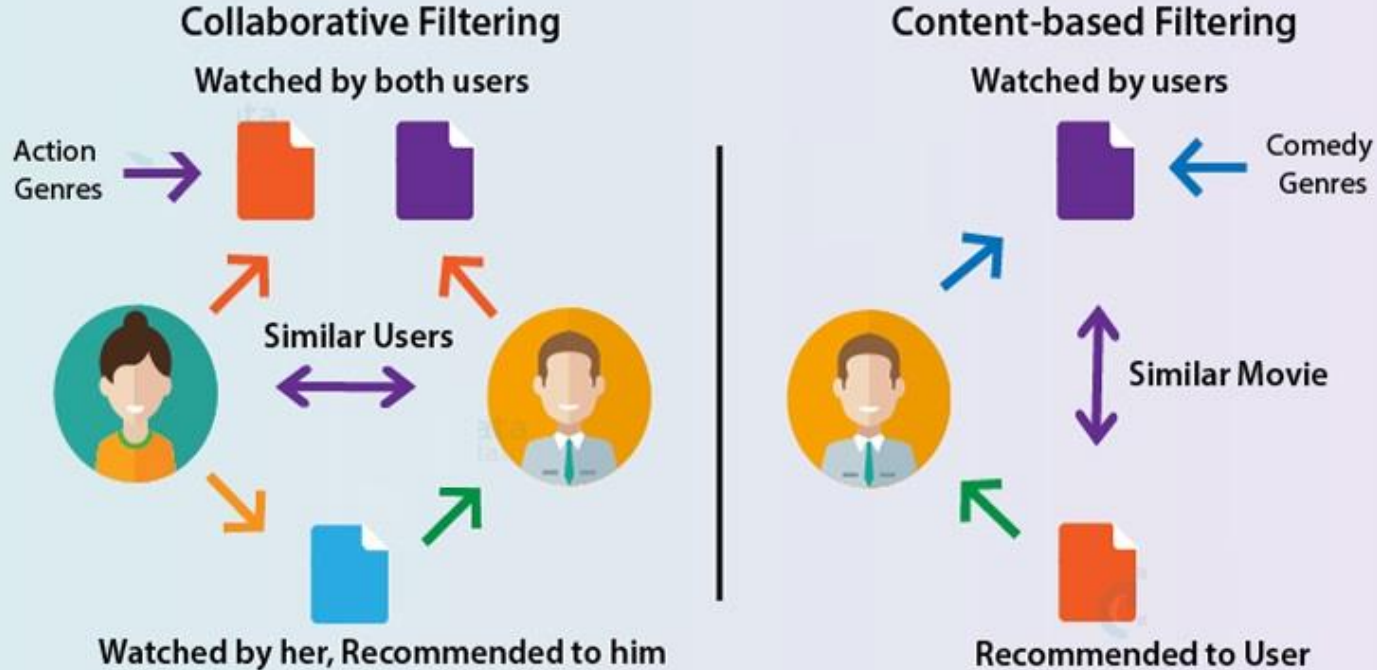
- Los **sistemas de recomendación de contenidos** se han hecho muy populares en los últimos años.
- Entre los ámbitos más extendidos se encuentran las recomendaciones de productos en tiendas online, películas, vídeos, música, libros, productos o recomendaciones de perfiles a los que seguir en redes sociales.



### > La importancia del proceso

- Desde un punto de vista más técnico, los recomendadores habitualmente son de dos tipos:
  - **filtros colaborativos**
  - **filtros basados en contenido**
- En este contexto, un filtro es el algoritmo matemático que “decide” cuál es la recomendación óptima basado en los datos que le entreguemos.

### Data Science Project - Movie Recommendation



### > Filtros colaborativos (collaborative filtering)

- Los filtros colaborativos generalmente basan su lógica en las **características** del usuario.
- Los datos que se tienen del usuario se convierten en el **centro** de un filtro colaborativo.
- El **sistema** analiza las compras anteriores, las preferencias, las calificaciones que ha dado de otros productos, el importe medio de las compras, etc.
- Busca otros usuarios que se parecen a él y que han tomado decisiones parecidas.
- Los productos que han tenido éxito con usuarios similares, seguramente también le interesarán al nuevo usuario.





### > Filtros basados en contenido (content-based filtering)

- Los filtros basados en contenido tienen el producto como base de la predicción, en lugar de tener al usuario. Es decir, utiliza las **características** del artículo (marca, precio, calificaciones, tamaño, categoría, etc.) para hacer las recomendaciones.
- Ejemplo: Pensemos en un sistema de recomendaciones de un servicio de música en streaming. El “producto” en este caso serían las canciones. Los datos de los que disponemos para cada canción son por ejemplo el grupo, el cantante, la discográfica y el género (pop, rock, clásica, banda sonora...). Para enriquecer más al sistema, también vamos a valorar las calificaciones que el usuario ha hecho sobre los temas – calificaciones explícitas, como las puntuaciones con estrellas, o implícitas, como las veces que ha escuchado el tema –, así como las características propias del usuario (edad, sexo y país).

### > Filtros basados en contenido (content-based filtering)

- Estos datos, centrados en el producto y en consonancia con datos del usuario, serán la materia prima de este sistema de recomendación.
- Veamos cómo se hace la predicción.
  - La pregunta que debe responder en este caso es: este usuario al que tengo que hacer una recomendación y que tiene estas características, este comportamiento y que ha calificado previamente estas canciones, ¿qué calificación le daría a esta canción, que es del grupo X, de la discográfica Y y de género Rock?
  - La respuesta del nos daría un número entre 0 y 10, basado en las calificaciones que les han dado otros usuarios que se parecen a él. Esta pregunta se debe repetir con todas las canciones que se incluyan en el catálogo.

### > Puntos a tener en cuenta

- La experiencia del usuario es lo más importante: no saturar a los usuarios.
- Lo imprescindible es que se muestren en el flujo de navegación cuando al usuario le sea más útil recibirlas.
- Es diferente una buena recomendación que una recomendación útil, es decir, que las recomendaciones no sean obvias para el cliente.

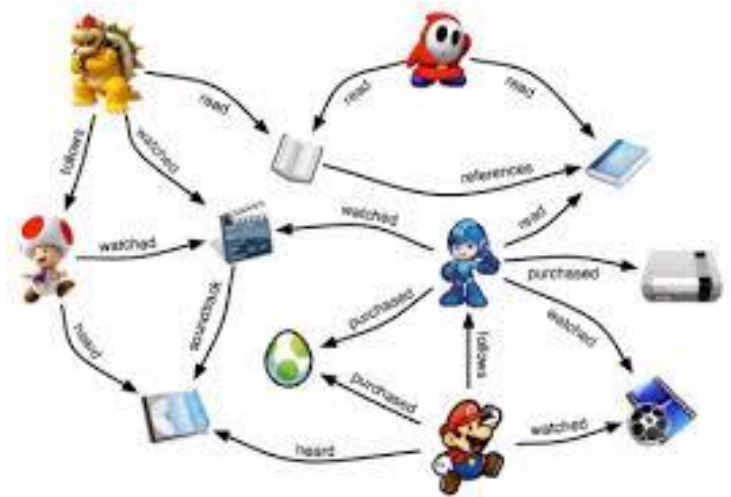


### > Puntos a tener en cuenta

- Balancear algoritmos y datos: la calidad y cantidad de los datos es más importante que los algoritmos.

# Recordar

- El proceso de selección de características, limpieza de datos es muy importante (lo hemos visto en el tema 3).



### > Conclusiones

- En la creación de un recomendador hay que tener en cuenta:
- Base de conocimiento 20%
- Procesamiento de la base de conocimientos – 10%
- Analítica y control de negocio – 20%
- Interfaz con el usuario – 50%

### > 4.4 Ejemplos de sistemas de recomendación: Fab

En Fab, Balavanovic y Shoham [1997] crean un sistema híbrido, basado en contenido y filtrado colaborativo, que recomienda al usuario páginas de Web. El sistema modela el perfil del usuario basado en el contenido de los análisis (cuando un usuario califica una página) y compara estos perfiles para determinar similitudes entre usuarios para una recomendación colaborativa. De esta manera el usuario recibirá páginas, tanto las que ha calificado relevantes (con respecto a su perfil) como las que han recibido calificaciones altas por usuarios con un perfil similar al suyo (vecinos cercanos).

La arquitectura de Fab se encuentra formada por tres componentes principales: agentes de colección, que seleccionan páginas de un tema específico; agentes de selección, los cuales encuentran páginas para un usuario específico, y un ruteador central que los conecta. Cada agente mantiene un perfil del usuario basado en las palabras que contienen las páginas de Web que el usuario ha calificado.

### > 4.6 Referral Web

Es un sistema interactivo para la construcción, visualización y búsqueda de redes sociales en el World Wide Web [Kautz *et al.* 1997]. Este sistema modela una red social mediante un grafo en cuyos nodos son representados los individuos y las conexiones entre nodos indican una relación directa descubierta entre ellos. Para identificar la relación directa entre individuos, se utilizan métodos tales como referencias encontradas en las páginas personales, referencias de co-autores en publicaciones técnicas, citas en las publicaciones y organigramas.

Referral Web no intenta crear nuevas comunidades sino más bien ayudar a los usuarios a hacer un uso más eficiente de sus redes existentes de colegas profesionales. Perteneciendo a una comunidad, el usuario puede descubrir contactos a gente o a información que de otra manera le estaría oculta.

```
> import graphlab
import pandas as pd

# Load up the data with pandas
r_cols = ['user_id', 'food_item', 'rating']
train_data_df = pd.read_csv('train_data.csv', sep='\t', names=r_cols)
test_data_df = pd.read_csv('test_data.csv', sep='\t', names=r_cols)

# Convert the pandas dataframes to graph lab SFrames
train_data = graphlab.SFrame(train_data_df)
test_data = graphlab.SFrame(test_data_df)

# Train the model
collab_filter_model = graphlab.item_similarity_recommender.create(train_data, user_id='user_id',
item_id='food_item', target='rating', similarity_type='cosine')

# Make recommendations
which_user_ids = [1, 2, 3, 4]
how_many_recommendations = 5
item_recomendation = collab_filter_model.recommend(users=which_user_ids, k=how_many_recommendations)
```



```
> import graphlab
import pandas as pd

# Load up the data with pandas
r_cols = ['user_id', 'food_item', 'rating']
train_data_df = pd.read_csv('train_data.csv', sep='\t', names=r_cols)
test_data_df = pd.read_csv('test_data.csv', sep='\t', names=r_cols)

# Convert the pandas dataframes to graph lab SFrames
train_data = graphlab.SFrame(train_data_df)
test_data = graphlab.SFrame(test_data_df)

# Train the model
cotent_filter_model = graphlab.item_content_recommender.create(train_data, user_id='user_id',
item_id='food_item', target='rating')

# Make recommendations
which_user_ids = [1, 2, 3, 4]
how_many_recommendations = 5
item_recomendation = cotent_filter_model.recommend(users=which_user_ids, k=how_many_recommendations)
```

### Bibliografía

- <https://apple.github.io/turicreate/docs/userguide/>
- <https://www.datacamp.com/community/tutorials/recommender-systems-Python+>
- <https://medium.com/ai-society/a-concise-recommender-systems-tutorial-fa40d5a9c0fa>

# Gracias