

Trabajo Final de Master

Recomendador musical para eHealth



Autor: Adrián Quijada Gomariz

Tutor: Raúl Parada Medina
Profesor: Jordi Casas Roma



Índice

01 Motivación

Contexto

Parkinsons: eHealth app

Estado del arte: sistemas de recomendación

02 Diseño de la solución

Recomendador musical para eHealth

Conjunto de datos

Definición de la solución

03 Resultados

Análisis comparativo

Estudio sobre las recomendaciones realizadas por los distintos algoritmos

04 Conclusiones

Conclusiones

Líneas de trabajo futuras



01

Motivación

¿Sabías que?



ParkinSons: eHealth app

4

- Ganadora del OpenEhealth 2017. Colaboran
- Aplicación única en el mercado. Permite **mejorar el estado físico y mental** del paciente al mismo tiempo
- **Experiencia personalizada** para cada usuario
- El usuario **mejora sus procesos cognitivos y motores** de forma **autónoma**



- **Enfoque *content-based***

- Utilizan medidas de similitud para realizar las recomendaciones
- No utilizan información comportamental del usuario
- *Gap* semántico

- **Enfoque *collaborative filtering***

- Utilizan información comportamental del usuario y permiten la introducción de la información contextual
- Técnicas con buenos resultados: Matrix Factorization, Factorization Machine. Nuevas tendencias
- *Cold-start*

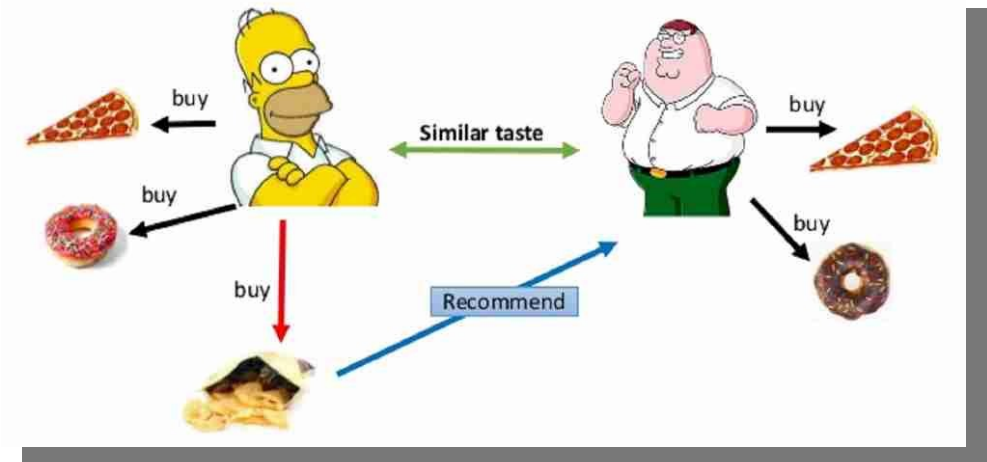
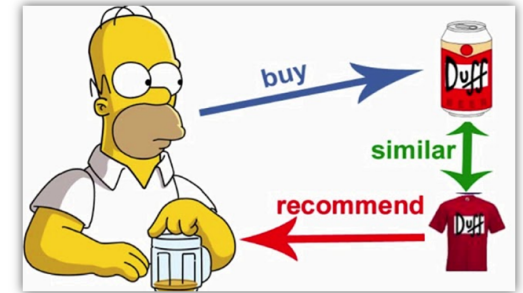
- **Enfoque híbrido**

- Combinan las características de los dos enfoques previos
- Más complicados de implementar

En el ámbito de la salud o *eHealth*:



- Problema muy amplio y difícil de abordar
- Sistemas de recomendación basados en reglas
- Conocimiento experto



02

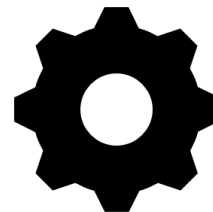
Diseño de la solución

Recomendador musical para eHealth

7

Usuario

- Configura su perfil
- Personaliza sus gustos musicales
- Selecciona su estado de ánimo



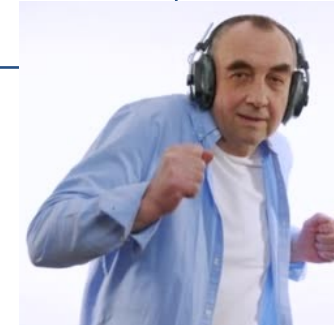
Sistema de recomendación

Usuario

- Obtiene un listado de hasta 10 canciones
- No escuchadas previamente
- Se solicita *feedback*

Objetivo

Desarrollar un sistema de recomendación musical óptimo dentro del entorno eHealth proporcionado por la aplicación Parkinsons





- 1.000.000 de canciones
- 53 características



- ~48M de interacciones usuario-canción
- 1.019.318 usuarios únicos
- 384.546 canciones únicas



- 676.342 canciones con una serie de géneros y su peso. Sólo consideramos el género con mayor peso

Los requerimientos son:

- Datos de canciones, indispensable el género y el tempo*
- Datos comportamentales de usuarios, con información explícita en forma de *feedback*



*Antoni Crespo Martnez. Consideracions sobre la implementacio musical de l'app parkinsons, 2019.

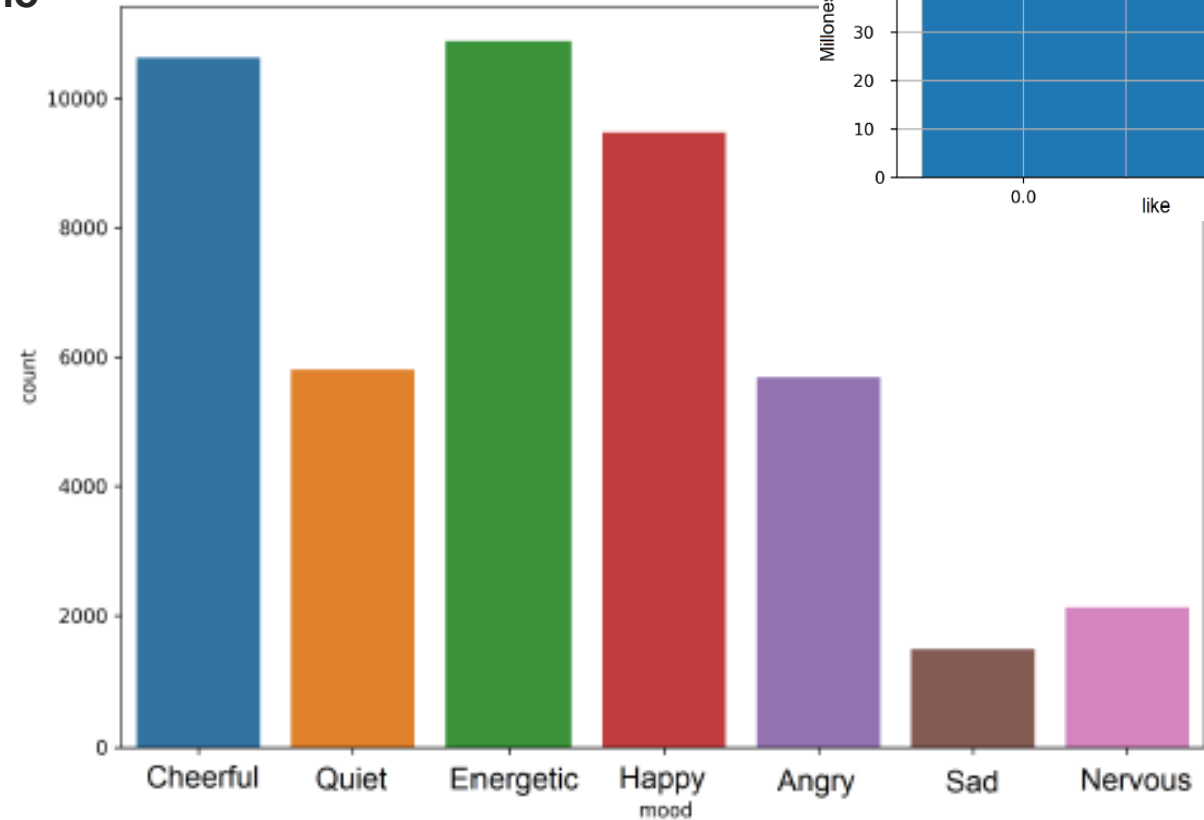
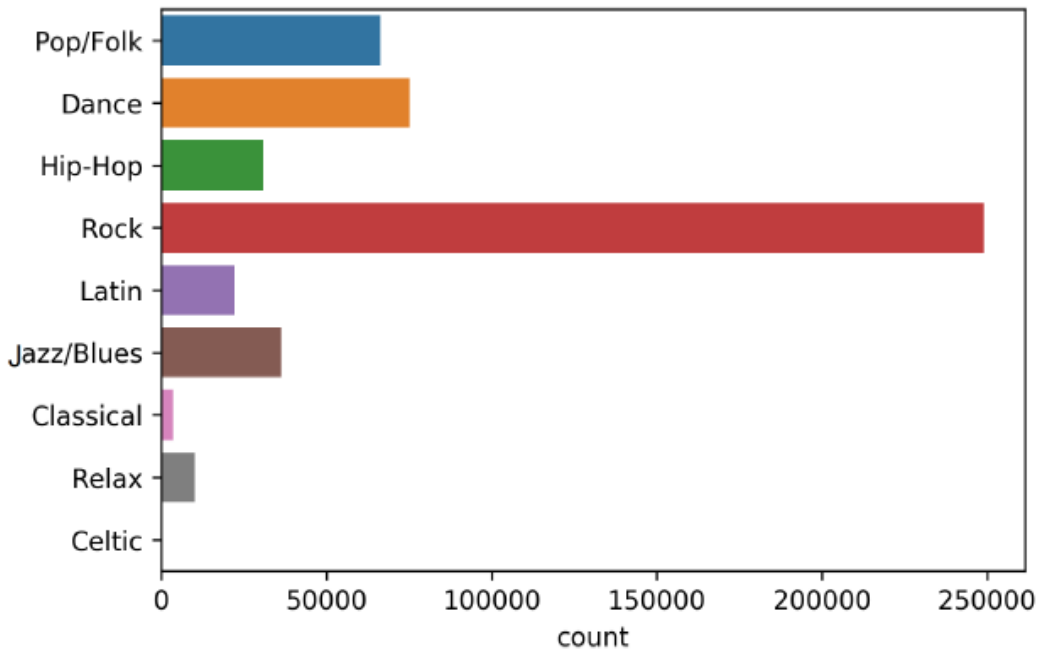
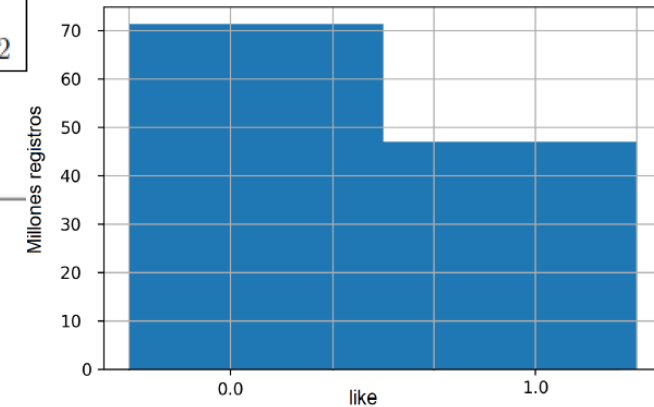


Conjunto de datos

Algunas acciones del **Preprocesado**:

- **Limpieza y filtrado** de datos
- **Transformación** de variables
- Generar nueva característica **estado de ánimo**
- Obtenemos df_{music} y df_{user}

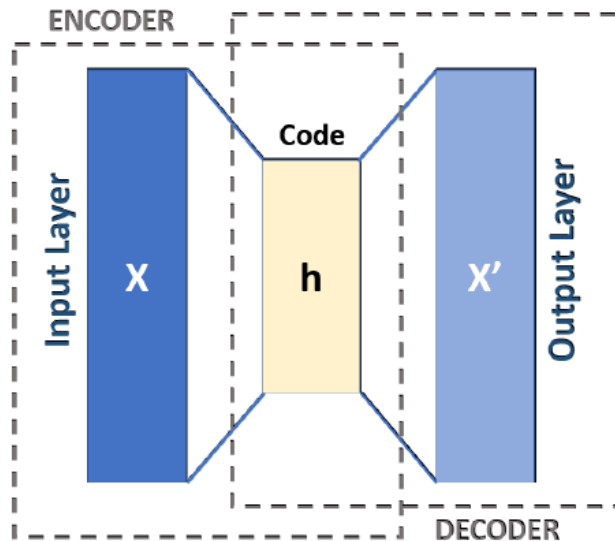
Índice	count	\bar{x}	σ	min	25 %	50 %	75 %	max
Usuarios	129.507	91,39	50,56	51	60	75	103	942
Canciones	20.815	568,63	987,82	29	193	299	555	25.172



Definición de la solución

Enfoque *content-based*

- Implementación de un *autoencoder* con **redes neuronales**
- Aprenderá las **características latentes** de cada canción
- Recomendaciones en función de la **similitud** entre ellas



Enfoque híbrido

- Implementación *Matrix Factorization* + características latentes del *autoencoder*

Enfoque *collaborative filtering*

- Técnica *Matrix Factorization*
 - Descomponen las **interacciones entre el usuario y la canción** mediante matrices
 - Capturan las **características latentes** entre ambas entidades
- Técnica *Factorization Machine*
 - **Evolución** de la técnica anterior
 - Se factorizan múltiples entidades. Introducimos el **contexto**
- Implementación con **redes neuronales** mediante *embeddings*



Entrenamiento

- *Binary Cross Entropy*

$$L = \sum_{i=1}^N y_i \log(\sigma(p_i)) + (1 - y_i) \log(1 - \sigma(p_i))$$

A destacar:

- *Autoencoder*: 5 capas. Tamaño 128x64x50x64x128



Mayor tamaño en el espacio latente proporciona mejores resultados

- *Deep Matrix Factorization*: 3 capas. Tamaño 128x64x32 con *dropout*



Redes profundas mejor que con gran tamaño

- *Deep Factorization Machine*: 5 capas. Tamaño 64x64x64x32x16 con *dropout*



Concatenamos el estado de ánimo a la canción

- *Deep Matrix Factorization Hybrid*: 3 capas. Tamaño 64x128x64 con *dropout*



Aplicar una capa *Fully Connected* a las entidades de entrada mejora los resultados

Evaluación

- *Cosine proximity*

$$\text{similarity} = \cos(\Theta) = \frac{\sum_{i=1}^N A_i B_i}{\sqrt{\sum_{i=1}^N A_i^2} \sqrt{\sum_{i=1}^N B_i^2}}$$

- *Accuracy*

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2$$

- *Mean Squared Error*

- ~~Precisión-en-k~~

$$P_k(u, y) = \frac{1}{k} \sum_{j=1}^k M_{u,y(j)}$$



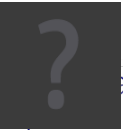
03

Resultados



Modelo	Contexto	<i>Loss</i>	<i>Accuracy</i>	P_{10}
DMF	✗	0,5869	0,6866	0,6159
DeepFM	✓	0,5786	0,6945	0,6875
DMFHybrid	✗	0,5867	0,6877	0,6178

- Consideramos Deep Matrix Factorization (DMF) como *baseline* en el enfoque *collaborative-filtering*
- Deep Factorization Machine incluye el contexto: estado de ánimo del usuario; género, artista, publicación, año de la canción
- DeepFM > DMFHybrid > DMF



Estudio sobre las recomendaciones

14

- Estudio sobre el top-10 de recomendaciones → Métrica P_{10}
- Muestra reducida → 500 usuarios del conjunto de datos total
- Resultados obtenidos filtrando por **género**

Género	DMF	DeepFM	DMFHybrid	Autoencoder
Celtic	0,3878	0,3878	0,3878	0,2295
Classical	0,4597	0,4606	0,4566	0,4451
Dance	0,4972	0,5623	0,5330	0,3726
Hip-Hop	0,4873	0,5296	0,5038	0,3610
Jazz/Blues	0,4866	0,5149	0,4974	0,3969
Latin	0,5030	0,5334	0,5157	0,4422
Pop/Folk	0,4958	0,5491	0,5190	0,4118
Relax	0,4702	0,4799	0,4734	0,4290
Rock	0,5122	0,5978	0,5559	0,3870

CONCLUSIONES:

- Los peores resultados se observan para el género Celtic
- Los mejores resultados se observan para el género Rock
- DeepFM siempre proporciona los mejores resultados
- Autoencoder siempre proporciona los peores resultados

La información comportamental es determinante para ofrecer mejores recomendaciones



Estudio sobre las recomendaciones

15

- Estudio sobre el top-10 de recomendaciones → Métrica P_{10}
- Muestra reducida → 500 usuarios del conjunto de datos total
- Resultados obtenidos filtrando por **estado de ánimo**

Estado de ánimo	DMF	DeepFM	DMFHybrid	Autoencoder
Angry	0,4785	0,5164	0,4962	0,4070
Cheerful	0,5062	0,5459	0,5236	0,4163
Energetic	0,4771	0,5184	0,4958	0,3648
Happy	0,4935	0,5350	0,5119	0,4095
Nervous	0,4477	0,4765	0,4605	0,3830
Quiet	0,4808	0,5175	0,4978	0,3636
Sad	0,4605	0,4800	0,4694	0,3586

CONCLUSIONES:

- Los peores resultados se observan para los estados de ánimo *Nervous* y *Sad*
- Los mejores resultados se observan para los estados de ánimo *Cheerful* y *Happy*

El recomendador tiende a funcionar mejor para estados de ánimo más “neutros”, sin penalizar en exceso aquellos más “extremos”



04 Conclusiones



- El modelo DeepFM es la mejor solución encontrada
- La información contextual añadida a las interacciones usuario-canción favorece un mejor rendimiento
- La arquitectura ideada se ha de determinar particularmente para cada aplicación o conjunto de datos
- La solución basada en *content-filter* (autoencoder) se puede utilizar en la aplicación mientras no haya un volumen significativo de datos para solucionar el problema del *cold-start*
- El autoencoder ofrece peor rendimiento que el resto de modelos, pero no está condicionado al volumen de datos disponible sino a las características propias de las canciones

- Implementar el sistema de recomendación en la aplicación Parkinsons
- Analizar nuevas técnicas vanguardistas usando teoría de grafos para capturar las señales de muy alto orden entre los usuarios y los objetos. Como *Graph Autoencoders*, *Graph Convolutional Networks* y evoluciones de *Factorization Machines* basadas en grafos

¡Muchas gracias!

