
Recomendación de videojuegos basada en emociones

Benito Alonso Oliva Tesla^{* 1} Justo Matías Solís Zlatar^{* 1} Daniel Ignacio Vera Ortiz^{* 1}

Abstract

Este trabajo propone un sistema de recomendación de videojuegos que incorpora el estado emocional que el usuario desea experimentar. Para lograrlo, se combinan dos enfoques: un modelo tradicional DeepFM que utiliza datos explícitos e implícitos del usuario, y un componente basado en NLP que interpreta un prompt emocional mediante un modelo BERT **fine-tuneado**. Ambos modelos se integran mediante un ensamble ponderado que permite recomendaciones personalizadas según emociones buscadas en los videojuegos. Los resultados muestran que el NLP no es competitivo por su cuenta, pero el ensamble logra incorporar la emoción deseada sin sacrificar demasiado rendimiento.

1. Introducción

Actualmente, los sistemas de recomendación de videojuegos en plataformas como *Steam* o *Epic Games* se basan principalmente en información implícita y explícita de los usuarios, pero estos sistemas no consideran el estado emocional o la motivación que desean experimentar. Debido a lo anterior, en este proyecto se busca proponer un modelo de ensamble que integre información implícita y explícita de los usuarios, junto con la experiencia emocional que quieren experimentar, con el objetivo de generar recomendaciones con un mayor grado de personalización. Cabe destacar que el trabajo realizado se encuentra disponible en GitHub¹.

2. Objetivos

El objetivo del proyecto es desarrollar un modelo de ensamble entre un recomendador tradicional (*DeepFM*) que considere el comportamiento del usuario, y un componente

basado en NLP capaz de interpretar un *prompt*, donde el usuario pueda expresar la emoción que desea experimentar. El sistema buscará mapear esa intención emocional con juegos compatibles, y se evaluará su desempeño según diferentes métricas.

2.1. Objetivos específicos

- Explorar y preparar los datos provenientes de distintos archivos para construir un conjunto coherente de interacciones usuario-juego enriquecidas con metadatos.
- Utilizar un modelo de aprendizaje profundo (como *DeepFM*) que utilice la información explícita e implícita de los usuarios y juegos.
- Desarrollar un modelo basado en NLP que mapee las emociones del usuario a los juegos que evocan emociones similares, según sus títulos, descripciones y *tags*.
- Realizar un ensamble entre ambos modelos mediante un *score* ponderado para realizar recomendaciones en base a las emociones.
- Evaluar los modelos utilizando las métricas *Recall@10*, *MAP@10*, *nDCG@10*, *Diversity@10* y *Novelty@10*.

3. Trabajos relacionados

De los trabajos recientes basados en emociones, algunos de estos utilizan *Computer Vision* para obtener la emoción del usuario a través de imágenes faciales, particularmente planteado en [5] y [7] que son recomendadores de música. Estos tienen una limitación clara y es la necesidad de obtener información visual facial del usuario, la cual no siempre estará disponible, por lo que es un recomendador situacional que no logra funcionar en la práctica, debido a que es probable que muchos de los usuarios no permitan que se use su rostro para una tarea tan simple como la recomendación de música.

Por otro lado, hay otros trabajos que han abordado las recomendaciones basadas en emociones, infiriendo la emoción de los comentarios, *reviews* y historial de los usuarios, así como lo son los trabajos [1], [2], [4], [6], [8] y [9], donde a partir de los comentarios del *item* y del usuario, así

^{*}Equal contribution ¹Pontificia Universidad Católica de Chile, Chile. Correspondence to: Benito Alonso Oliva Tesla <benito.oliva@uc.cl>, Justo Matías Solís Zlatar <jsolis@estudiante.uc.cl>, Daniel Ignacio Vera Ortiz <di.vera@uc.cl>.

Asignatura Sistemas Recomendadores, profesor Denis Parra Santander

¹<https://github.com/BenitoOl/Proyecto-Recomendadores>

como su historial, se calcula una tendencia promedio emocional que busca ayudar en futuras recomendaciones. Estos trabajos, por su parte, a pesar de trabajar con emociones, son distintos a lo que se plantea en este documento, debido a que buscan determinar las emociones de la información que ya se tiene de los *items* o usuarios, por ello, en lugar de buscar la emoción cambiante del momento, busca la emoción que más tiende a mostrar el usuario.

Por último, existe un trabajo que utiliza tanto imágenes (CNN) como NLP para determinar el estado de ánimo del usuario, a partir del cual recomienda ciertos géneros que se alinean con la emoción que experimenta [3]. Por su lado, este trabajo muestra un método similar al que se busca plantear en este documento. Sin embargo, en ese caso se busca mapear las emociones que siente el usuario a algún o algunos géneros. El problema en este caso, es que esto requiere un análisis profundo, debido a que no es trivial cuáles géneros debieran ser recomendados a partir de un estado emocional en particular. Esto conduce a dudas respecto a por qué un género se recomienda a una cierta emoción, tema que no es muy tratado en el trabajo mencionado.

De los trabajos mencionados, vemos que hay múltiples enfoques para tomar en cuenta las emociones a la hora de recomendar, así como que hay limitaciones para las formas actualmente probadas. Por ello, este trabajo busca utilizar NLP para poder entender que emoción quiere experimentar el usuario en lugar del estado emocional del mismo, dado que en el segundo caso, se requiere un análisis más profundo que contemple **que** juegos recomendar a un usuario en base a cómo se siente, lo cual no es trivial, dado que a una persona triste no es tan fácil como recomendarle juegos felices o tristes, sino que requiere de un análisis que se base en estudios para determinar que juegos puede disfrutar más una persona triste.

4. Dataset

Para el desarrollo de este proyecto, se utilizaron 3 *datasets* de *Steam* y un archivo con metadatos. Originalmente, la base de recomendaciones tenía 41.154.794 datos ², lo cual ralentizaba la ejecución y generaba problemas en Google Colab. Debido a esto, se decidió considerar solamente los videojuegos que presentaran más de 100 *reviews*.

4.1. Archivos

1. **games.csv** (Archivo con 35.032 registros con la información de los juegos. Sus principales columnas son: ID del juego, título, fecha de lanzamiento, cantidad de *reviews*, categoría de *reviews* y precio).

²<https://www.kaggle.com/datasets/antonkozyriev/game-recommendations-on-steam/data>

2. **recommendations.csv** (Archivo con 1.539.390 registros, con las recomendaciones de los usuarios. Sus principales columnas son: ID del juego, ID del usuario, fecha y horas jugadas).
3. **users.csv** (Archivo con 8.364 registros, con información de los usuarios. Sus columnas son: ID del usuario, cantidad de productos comprados y cantidad de *reviews* realizadas).
4. **game-metadata.json** (Archivo con 50.872 registros, que entrega las descripciones y *tags* de los juegos).

5. Modelos usados

5.1. Baseline

Los modelos que conformarán el *baseline* para este proyecto son *Random*, *Most Popular* e *Item-based Collaborative Filtering* (IBCF). El primer modelo se encarga de seleccionar *items* al azar, el segundo realiza recomendaciones de videojuegos en base a aquellos que son considerados como más populares, mientras que IBCF se encarga de realizar recomendaciones según similitudes entre ítems, con respecto a las preferencias de los usuarios.

5.2. DeepFM

DeepFM es un modelo que combina FM (Factorization machine) y DNN (Deep Neurologic Network). Complementando lo simple de FM con las relaciones complejas que puede detectar DNN.

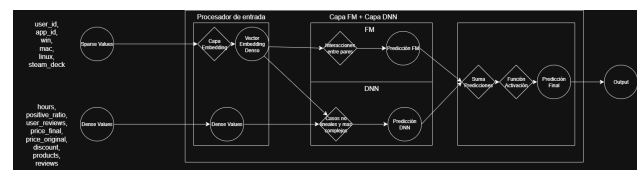


Figure 1. DeepFM

Se **utilizo** la librería DeepCTR-Torch ya que permite implementar de manera fácil DeepFM y entrenarlo.

Para entrenar el modelo se dividieron las columnas entre categoricas (sparse) y **numéricas** continuas (Dense). Para mayor detalle revisar Anexo A. Este modelo de aprendizaje automático se encarga de combinar *Factorization Machines* con *Deep Neural Networks*, permitiendo un mejor aprendizaje de relaciones simples y capturar relaciones complejas y/o no evidentes. Se utilizó la librería *deepctr-torch* y se dividió las columnas según eran categóricas o numéricas continuas, lo cual entregó como numéricas a *user_id*, *app_id*, *win*, *mac*, *linux*, *steam_deck* y las columnas numéricas continuas son *hours*, *positive_ratio*, *user_reviews*, *price_final*,

price_original, discount, products, reviews. A continuación, se muestra un diagrama que muestra el funcionamiento de este modelo.

5.3. NLP

En este trabajo, el componente NLP utilizado es un modelo preentrenado, el cual es **BERT base uncased**, fine-tuneado por Bhadresh-savani en el dataset **GoEmotions**. Este modelo se encarga de recibir texto y entrega un diccionario con puntuaciones entre 0 y 1 para las siguientes emociones: *love*, *fear*, *joy*, *sadness*, *surprise* y *anger*. En este proyecto, se concatenaron el título, la descripción y las etiquetas (o *tags*) de cada juego como *input* del modelo, y se almacenó la puntuación de cada emoción en columnas del *Dataframe*. El *score* para cada usuario se obtiene a través de **similitud coseno** entre los vectores de emociones del *prompt* con los de los videojuegos. Todo lo anterior se puede ver en la figura a continuación:

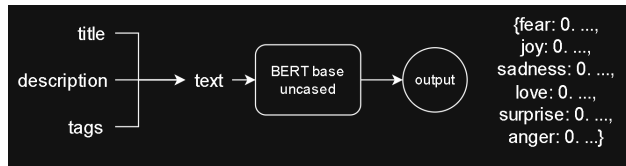


Figure 2. NLP

Un ejemplo concreto del modelo es con el *prompt* "i want to play a videogame with a sad history", para el cual los juegos más parecidos por similitud de coseno son los siguientes:

	title	emotion_match_score	surprise	fear	love	sadness	anger	joy
35097	THE LAST OF US	1.00000	0.000135	0.000197	0.000665	0.997395	0.000720	0.000039
33781	鬼泣4特别版-不属于任何人的灵魂故事-MOBIUS & ELECHOUS TO N.	1.00000	0.000230	0.000375	0.000460	0.997632	0.000491	0.000044
34194	High School DxD	1.00000	0.000136	0.000551	0.000489	0.997300	0.000602	0.000042
34989	Wilderling	1.00000	0.000202	0.000410	0.000391	0.997697	0.000469	0.000023
19677	海狗 Sea of ans	1.00000	0.000224	0.000461	0.000810	0.996933	0.000449	0.001167
1143	Metaphren	0.949199	0.000615	0.030666	0.001106	0.696936	0.000468	0.225429
8098	HRZC VC	0.948900	0.004767	0.247027	0.001525	0.742672	0.001668	0.002362
10231	May May Prince - The Vampire Star	0.948427	0.000996	0.004308	0.011573	0.717061	0.027611	0.238491
11369	Tierista	0.948160	0.004973	0.165027	0.004649	0.649110	0.136602	0.048159
25432	Cursor Engine	0.948125	0.001447	0.000549	0.003649	0.721605	0.022728	0.243812

Figure 3. Juegos similares para el *prompt* relacionado a tristeza

De estos juegos, el primero tiene una descripción: "My daily life brutally collapsed the day Marie, my wife, took her own life...", por lo cual, si cumple con ser un videojuego triste. Otro ejemplo es el tercer título, que tiene como descripción: "... troubled relationship with her grandmother, who is her only family guardian after her parents died ... devastating news that her Grandmother has cancer." Nuevamente damos cuenta que el NLP está asociando bien las emociones a los juegos. Aun así, cabe destacar que el modelo depende de qué tan buena sea la descripción del juego.

Por otro lado, es importante destacar que la emoción más

predicha por el modelo es *joy*, lo cual se debe a que las descripciones están orientadas a dar el mensaje de que el juego es divertido dentro de su categoría, ya sea terror, tristeza o demás. Es importante destacar, que este conteo hace referencia a si la emoción está presente sobre un valor límite, no necesariamente debe ser la emoción predominante. A continuación se muestra la distribución total de emociones en los juegos:

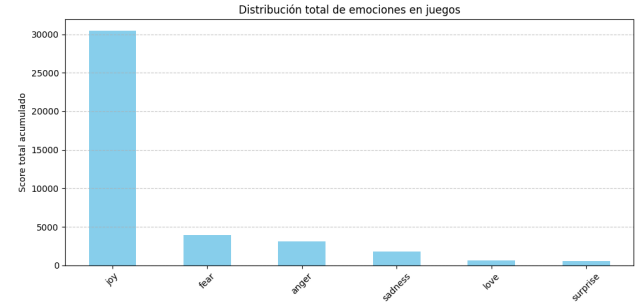


Figure 4. Distribución total de emociones en juegos

Finalmente, es importante destacar que para obtener un *score* se crearon seis *prompts* genéricos que evocaran una de las emociones del modelo escogido. Los *prompts* son:

- "i want to play a sad game, i want to cry". Para que la emoción más evocada sea *sadness*.
- "i want to play a horror game, i wanna feel fear". Para que la emoción más evocada sea *fear*.
- "idk, i just wanna play a game about a relationship". Para que la emoción más evocada sea *love*.
- "i just wanna play a normal and fun game". Para que la emoción más evocada sea *joy*.
- "i want to play a game about wars that make me feel anger, or something like that". Para que la emoción más evocada sea *anger*.
- "i want to feel surprise". Para que la emoción más evocada sea *surprise*.

5.4. Ensamble (DeepFM + NLP)

Este modelo combina los *scores* de recomendación entregados por los modelos *DeepFM* y NLP, para que el modelo considere las emociones del usuario al realizar la recomendación. Matemáticamente, se puede expresar de la siguiente forma:

$$S_{ens.}(u, i) = \alpha * S_{DFM}(u, i) + (1 - \alpha) * S_{NLP}(u, i) \quad (1)$$

6. Experimentos y resultados

Para trabajar en los modelos de recomendación, se dividió la información en datasets de entrenamiento y validación, de los que se obtuvieron diversas estadísticas, las que se pueden ver en la siguiente tabla:

Estadísticas	Entrenamiento	Validación
Número de usuarios	8.364	8.364
Número de juegos	34.388	26.382
Reviews totales	1.228.168	310.423
Promedio de reviews por usuario	146,84	37,21
Promedio de reviews por juego	35,71	11,52
Tasa de reviews positivas	77,97%	78,06%
Máxima cantidad de reviews de un usuario	4.836	1.209
Máximo número de reviews en un juego	1.972	541
Densidad (%)	0,43%	0,14%

Table 1. Estadísticas de entrenamiento y validación

Cabe mencionar que, a pesar de la baja densidad presente en ambas bases, se tiene un buen número promedio de interacciones tanto para los usuarios como para los juegos. También, es importante destacar la mayor cantidad de reviews positivas que negativas, que se debe tomar en cuenta a la hora de desarrollar los modelos.

A continuación, se muestran algunos gráficos con estadísticas importantes para analizar los datasets.

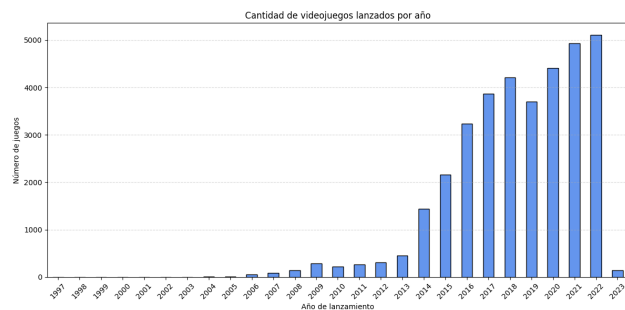


Figure 5. Cantidad de videojuegos lanzados por año

La Figure 5 da cuenta que el dataset tiene más juegos de los años recientes.

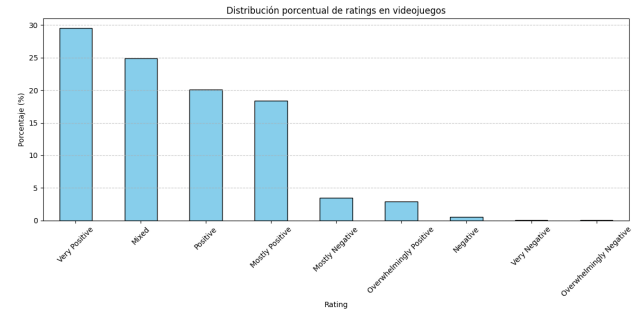


Figure 6. Distribución porcentual de ratings en videojuegos

La Figure 6 cuenta la cantidad de juegos que tienen una calificación promedio en una de las categorías de la imagen. De la figura podemos ver que en general los juegos tienen un promedio positivo de reseñas, lo cual puede ser algo característico de la industria de los videojuegos, donde priman las reviews positivas en los juegos conocidos.

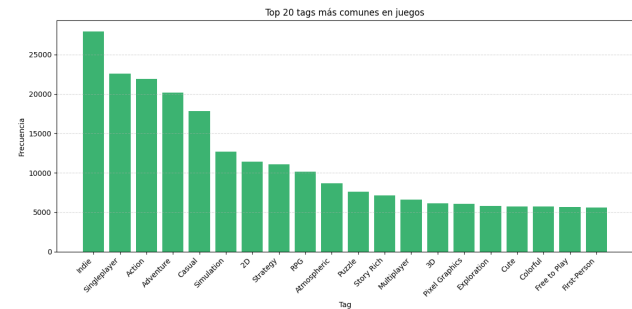


Figure 7. Top 20 tags más comunes en juegos

La figura anterior permite conocer la variedad de categorías existentes de videojuegos en el dataset, así como los que más se repiten.

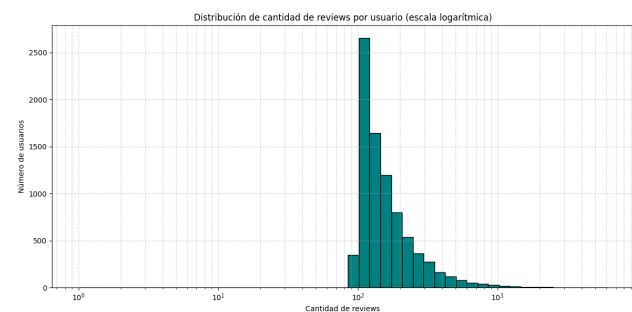


Figure 8. Cantidad de reviews por usuario

Tal como se mencionó, debido a la enorme cantidad de datos, se trabaja con los usuarios que tienen al menos 100 reviews, lo cual se demuestra en la figura anterior.

Finalmente, los resultados obtenidos, en el caso del NLP el promedio de los seis *prompts*, se presenta a continuación:

Indicadores	Random	Most popular	Item kNN
Recall@10	0,0003	0,0200	0,0042
MAP@10	0,0003	0,0183	0,0007
nDCG@10	0,0041	0,1866	0,0015
Novelty@10	16,9452	9,4787	...
Diversity@10	0,9905	0,6597	...
Indicadores	DeepFM	NLP	Ensamble
Recall@10	0,4560	0,0001	0,4508
MAP@10	0,7164	0,0001	0,7059
nDCG@10	0,9100	0,0018	0,9064
Novelty@10	12,7133	18,8589	12,8310
Diversity@10	0,9506	0,9939	0,9470

Table 2. Resultados de métricas por modelo

Cabe destacar que para **Item kNN**, se tuvo que utilizar un conjunto de datos más pequeño, debido a que la RAM de Colab no alcanzaba para entrenarlo. Por esta misma razón, no se obtuvo Novelty y Diversity para el modelo, debido a que no era comparable a los demás. Por el lado del NLP, vemos que al probarlo solo, este obtiene resultados incluso peores que *random* en las tres primeras métricas. Sin embargo, logra mejor Novelty y Diversity, lo cual se debe a que solo se preocupa de que la emoción que evoca el juego sea la que el usuario busca.

Luego, vemos un rendimiento levemente peor en el ensamble (comparado a *DeepFM*), que implicaría que agregar las emociones a la tarea de recomendación está ayudando a priorizar mejor los videojuegos, en base a la emoción que se busca experimentar, sin sacrificar mucho rendimiento. Esto se puede deber a que casi todo lo que reordena son *items* relevantes para el usuario.

De hecho, a continuación se puede apreciar cómo el ensamble cambia la recomendación completa del *DeepFM*:

Ranking por DeepFM:		
app_id		title
0	30476	CrocoMans
1	19698	Spooky Cats
2	12929	SEARCH ALL - CRABS
3	6818	Schustye
4	31373	NALOGI 2
5	31422	WIN THE GAME: DO IT!
6	9900	Puzzle Light
7	9417	100 hidden mushrooms
8	6601	Hidden Lands - Spot the differences
9	4316	Chaos
Ranking por Ensamble:		
app_id		title
0	30347	Just Bearly
1	30790	This Child Of Mine
2	33171	La Rana
3	2316	My name is You and it's the only unusual thing...
4	27496	Space of Darkness
5	6601	Hidden Lands - Spot the differences
6	18889	chip's Challenge 1
7	24196	Bloody Walls
8	30019	Mouse (Sneaking)
9	21196	CUPID - A free to play Visual Novel

Figure 9. Recomendación de *DeepFM* y el ensamble

Respecto al ensamble, se realizó un análisis de sensibilidad

para tres valores del ponderador, lo cual se muestra en la tabla a continuación:

Indicadores	$\alpha = 0,5$	$\alpha = 0,7$	$\alpha = 0,9$
Recall@10	0,4506	0,4478	0,4508
MAP@10	0,7055	0,7020	0,7059
nDCG@10	0,9062	0,9059	0,9064
Novelty@10	12,8369	12,8390	12,8310
Diversity@10	0,9467	0,9425	0,9470

Table 3. Resultados promedio por valor de α

De estos valores vemos que la mejor combinación es utilizar α 0,9. Sin embargo, a la vez son diferencias tan pequeñas, que para tener un mayor impacto de la emoción se debería colocar α igual a 0,5. En la *table 2*, se utilizaron los datos del α igual a 0,9 para comparar.

7. Discusión y Conclusiones

De los resultados obtenidos, es posible inferir que el hecho de añadir las emociones de los usuarios en el modelo de recomendación no afecta significativamente las métricas de *ranking* en este caso. Por lo tanto, se podría obtener una mejor recomendación personalizada según las emociones que busque experimentar el usuario.

A su vez, es necesario destacar las limitaciones que se presentaron durante la ejecución del trabajo. La primera limitación fue el hecho de trabajar con un modelo preentrenado, dado que este hecho implica trabajar con las emociones que define el modelo, cuando debería adaptarse a este contexto particular mediante *fine-tuning*, con un conjunto de emociones acorde al problema.

La segunda limitación es el conjunto de datos, el cual no contenía descripciones o *tags* para todos los juegos, por lo cual el NLP no era siempre aplicable. También, no contaba con imágenes, comentarios o demás información relevante para este tipo de problemas.

Finalmente, se sugiere como trabajo futuro explorar otros conjuntos de datos relacionados al mundo de los videojuegos o de otros dominios (música, libros, películas, entre otros), junto con enfoques similares al propuesto, donde un modelo alimente directamente a otro como parte de un sistema de recomendación integrado.

8. Bibliografía

1. Y. Gajula. *Sentiment-Aware recommendation systems in E-Commerce: A review from a Natural Language Processing Perspective*, 2025.
2. L. Yuan, M. Ding, F. Meng y Y. Tian. *Sentiment Analysis and Rating Video Game Dimensions via NLP*.
3. M. Sanadi, S. More, V. Shinde y H. Raghuvanshi.

Game Recommendation System Based on User Mood, 2024.

4. J. Leung, I. Griva y W. Kennedy. *Text-Based Emotion Aware Recommender*, 2020.
5. T. Babua, R. Naira y G. Aa. *Emotion-Aware Music Recommendation System: Enhancing User Experience Through Real-Time Emotional Context*, 2023.
6. N. Vashist, R. Kumar, M. Bisht y S. Singh. *Collaborative and Hybrid Recommender System for Enhanced Movie Recommendations*.
7. S. Katkuri, M. Chegoor, K. C. Sreedhar y M. Sathyanarayana. *Emotion Based Music Recommendation System*, 2023.
8. A. Abdul, J. Chen, H. Liao y S. Chang. *An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach*, 2018.
9. W. Xiong y Y. Zhang. *An intelligent film recommender system based on emotional analysis*, 2023.

9. Anexos

Anexo A

- Categóricas (Sparse): user_id, app_id, win, mac, linux, steam_deck.
- Numéricas (Dense): hours, positive_ratio, user_reviews, price_final, price_original, discount, products, reviews.