Pontificia Universidad Católica de Chile
Escuela de Ingeniera
Departamento de Computación

Informe intermedio: Recomendación de videojuegos basada en emociones

Integrantes Grupo 06

Benito Alonso Oliva Testa

Justo Matías Solís Zlatar

Daniel Ignacio Vera Ortiz

Tabla de contenidos

1. Progreso en el desarrollo de la solución propuesta 3
2. Experimentación realizada y evaluación intermedia 3
3. Análisis preliminar de los resultados obtenidos 4
4. Problemas identificados durante el proceso 5
5. Revisión del plan propuesto en etapa anterior y justificación de ajustes 6
6. Bibliografía Relevante 7
7. Anexos 9

Pontificia Universidad Católica de Chile
Escuela de Ingeniera
Departamento de Computación

1. Progreso en el desarrollo de la solución propuesta

La solución propuesta respecto a la primera entrega tuvo un cambio de enfoque, debido a la retroalimentación por parte de los docentes. Actualmente, se busca la recomendación de videojuegos basado en las emociones del usuario, abordando el contenido explícito e implícito mediante DeepFM y las emociones a través de un mapeo entre prompts a emociones y emociones a juegos. Con esto se busca crear un ensamble ponderado entre ambos modelos y analizar el impacto de este en las métricas. Más detalle al respecto se encuentra en el anexo A1, que son las partes corregidas del informe 1.

Hasta el momento, se ha desarrollado DeepFM y el NLP con un subconjunto de los datasets originales[1]. Así también como se ha hecho el ensamble de estos. Sin embargo, no está desarrollado por completo el mapeo entre prompts a emociones, debido a que falta definir la lógica bajo la cual se evaluará. Por ejemplo, cuando estén los prompts de por medio, estos harán falta para cada usuario, o al menos, uno general para todos para ver cómo se comportan las métricas, lo cual es un análisis exhaustivo.

En el caso de DeepFM se usó la librería de DeepCTR-Torch, ya que DeepCTR tenía un problema con los paquetes y el uso de DeepCTR-Torch es parecido a DeepCTR. Un punto importante fue el uso de la columna “hours” para entrenar DeepFM, usando de esta manera información implícita a la hora de recomendar un videojuego

En el caso de NLP, se utiliza el transformer preentrenado DistilRobERTa-base, que es una versión más ligera que RoBERTa y fue ajustado por Jan Hartmann para la tarea de análisis emocional. Utilizar estos modelos implican grandes ventajas y desventajas, lo cual se discute en el anexo A2.

Por último, es necesario destacar que los avances del proyecto se están trabajando en Github en el siguiente link: https://github.com/BenitoOl/Proyecto-Recomendadores.

2. Experimentación realizada y evaluación intermedia

Antes de los modelos, como se recomendo en la retroalimentación, a continuación, se colocan ejemplos de perfiles de usuarios de los datasets:

```
# Mostrar 3 perfiles representativos
print(user_profiles.head(10).to_markdown(index=False))
```

user_id	total_reviews	total_hours	avg_rating_given	percent_recommended	top_games	products
1239	311	7404.1	4.49839	98.7138	['WhiteLily 2: 梦醒少女', 'Celeste', 'Cyberpunk 2077']	1301
2821	321	9647.9	4.46262	97.5078	['The Deed', 'Darkwood', 'Cozy Grove']	1100
2881	171	25134.7	4.53216	99.4152	['Left 4 Dead 2', 'ARK: Survival Evolved', 'Command & Conquer: Red Alert 3 - Uprising']	464
3629	132	3189.6	4.54924	84.8485	['Coffee Talk', 'Inside the Backrooms', 'Detroit: Become Human']	523
4040	101	1266.8	4.5198	92.0792	['Wildermyth', 'Little Nightmares', 'Titanfall® 2']	584
8142	150	10291.4	4.38	56	['GTFO', 'Space Engineers', 'Green Hell']	232
8144	105	4743.7	4.01905	76.1905	['Tribes of Midgard', 'Gotham Knights', 'Gunfire Reborn']	458
10999	106	5007.6	4.17925	83.9623	['PUBG: BATTLEGROUNDS', 'Beat Saber', '7 Days to Die']	538
12596	114	2496.5	4.37719	95.614	['Resident Evil Village', 'Middle-earth™: Shadow of War™', 'Ghostrunner']	1108
13426	138	4990.7	4.49638	69.5652	['Super Meat Boy', 'Borderlands: The Pre-Sequel', 'Papers Please']	1218

Figura 1: Perfiles de usuario

[1] https://www.kaggle.com/datasets/antonkozyriev/game-recommendations-on-steam?select=games.csv

La figura 1 muestra perfiles de usuarios aleatorios de los datasets, principalmente mostrando la cantidad de reviews realizadas, las horas dedicadas a todos los juegos a los cuales les ha hecho una reseña, el porcentaje de reseñas en que recomienda el juego y los juegos más jugados por el usuario.

Los modelos ejecutados y probados para la recomendación de videojuegos fue random, most popular, item KNN, DeepFM y DeepFM + NLP. Este último permitirá cubrir las emociones del momento del usuario, que ayudará a personalizar las recomendaciones según el estado de ánimo de este.

Los resultados obtenidos al procesar los modelos se muestran en la siguiente tabla:

Indicadores	**Random**	**Most popular**	**Item KNN**	**DeepFM**	**Ensamble (DeepFM y NLP)**
Recall@10	0,0003	0,0177	0,0042	**0.4047**	0.4035
MAP@10	0,0003	0,0203	0,0007	**0.7425**	0.7396
nDCG@10	0,0053	0,2027	0,0015	**0.9228**	0.9217

Tabla 1: Resultados de los modelos

Tal como podemos ver, el mejor modelo es DeepFM, y le sigue un poco por debajo los valores del ensamble, que aún no está completamente desarrollado para mapear los prompts y evaluar los resultados.

3. Análisis preliminar de los resultados obtenidos

De los resultados obtenidos en la tabla 1, es posible especular que el añadir las emociones en la recomendación no afecta demasiado las métricas de ranking. Por lo tanto, se podría mejorar la personalización según los estados de ánimo o emociones. Sin embargo, el hecho de que los resultados sean tan buenos provoca una mayor necesidad de validarlos. De hecho, DeepFM luego del entrenamiento termino con un AUC de 0.56, por lo que se estima que se deberá revisar con cautela el código para comprobar si en realidad los resultados corresponden a la evaluación obtenida.

Por otro lado, los buenos resultados no son la única alerta, al recomendar a los usuarios, en general no producen recomendaciones diferentes el ensamble y DeepFM. Por ejemplo, para el usuario 2285, se obtiene las siguientes recomendaciones:

```
Ranking por DeepFM:
   app_id                                                title
0   12270  Batman: Arkham City - Game of the Year Edition
1   21062                                     NieR:Automata™
2   17838                                          Undertale
3   14884                          Assassin's Creed® Unity
4    2963                            Red Dead Redemption 2
5    7835                                 Sword and Fairy 7
6   20041                                       Mad Father
7    7525                                           Mitoza
8    2738                                    Paradise Killer
9   14059                      The Binding of Isaac: Rebirth

 Ranking por Ensemble:
   app_id                                                title
0   12270  Batman: Arkham City - Game of the Year Edition
1   21062                                     NieR:Automata™
2   17838                                          Undertale
3   14884                          Assassin's Creed® Unity
4    2963                            Red Dead Redemption 2
5    7835                                 Sword and Fairy 7
6   20041                                       Mad Father
7    7525                                           Mitoza
8    2738                                    Paradise Killer
9   14059                      The Binding of Isaac: Rebirth
```

Figura 2: Recomendaciones para el usuario 2285

De la figura 2 podemos ver qué es exactamente la misma recomendación, por lo que no hay mucho aporte del ensamblaje. Al igual, cabe destacar que también los scores dados por DeepFM y por el NLP cambian poco, esto se puede ver en el anexo A3, que a pesar de que parecen iguales, tienen diferencias en los demás decimales. Por último, otro ejemplo similar al de la figura 2 se encuentra en el anexo A4.

Respecto al NLP, no hay muchas maneras de evaluarlos de forma directa, debido a que no hay un ground truth como los demás casos. Si se buscará evaluar con este el ranking sería similar a un most popular de emociones. Por ello, se descarta evaluarlo y se busca analizar cómo cambia los resultados del DeepFM. También, es importante destacar que este método tiene cobertura para todo juego con descripción o tags, lo cual, a pesar de parecer obvio, es un problema del dataset con el cual se trabaja, debido a que algunos juegos no presentan esta información.

4. Problemas identificados durante el proceso

Durante el proceso de trabajo se han enfrentado diferentes problemas. Los principales problemas externos al desarrollo han sido el cambio de enfoque del proyecto, de un recomendador de videojuegos para probar las herramientas del curso a un recomendador basado en emociones. Luego, otro problema ha sido que el equipo no ha contado con la disponibilidad deseada, debido a carga académica, cambio de enfoque y retrabajo realizado.

Ahora, referente al desarrollo, los problemas que han surgido son principalmente tres. En primer lugar, los tiempos de ejecución al ser tan altos, se tuvo que trabajar con un subconjunto de los datos totales, lo cual se debió probar a mano, debido a que eran millones de datos.

El segundo problema es la definición del NLP y la limitación de nuestro trabajo, debido a que se debió buscar como realizarlo, optando en este caso por modelos preentrenados que

a pesar de que ofrecen rendimiento, limitan el alcance. Como se mencionó en el anexo A2, estos te obligan a trabajar con emociones del transformer, lo cual, para cambiarlo y usar emociones acordes al problema, implica realizar fine tunning, que se puede escapar de los plazos del curso.

Por último, como se mencionaba en apartado anterior, los resultados no parecen del todo consistentes con el desempeñó práctico de los modelos, lo que requiere revisarlos para comprobar la veracidad de las métricas obtenidas. Dicho proceso, en parte ya realizado, ha evitado que avancemos en tareas como definir las métricas de diversidad, novedad y demás.

5. Revisión del plan propuesto en etapa anterior y justificación de ajustes

Semana	Metodo	Criterio
26-30 mayo	Desarrollo DeepFM	nDCG, MAP y Recall
02-06 junio	Desarrollar NLP y ensamble	nDCG, MAP y Recall
09 - 13 junio	Definición de métricas definitivas para evaluar y ajustes en modelos	Evaluar métricas nuevas, además de diversity y novelty
16 - 20 junio	Entrenamiento del ensamble con más datos	Evaluar y comparar con baselines
23 – 27 junio	Desarrollar paper	-
30 – 03 junio a julio	Desarrollar poster	-

Tabla 2: Planificación

Debido al cambio de enfoque, la nueva planificación se expone en la tabla 2. Se estima que a partir de ahora el equipo de desarrollo tenga más constancia, debido a tener el nuevo enfoque definido y mayor disponibilidad en las siguientes semanas.

De esta planificación, debido a resultados obtenidos, la semana clave será la del 09 –13 de junio, debido a que se deberán revisar resultados, ajustar ensamble y conectar con prompts del usuario. Sin embargo, también se cuenta con ocupar el principio de la semana del 16 – 20 de junio, debido a que en esta solo se plantea el entrenamiento del modelo final. Finalmente, en las últimas dos semanas se plantea desarrollar o terminar, en caso de que se trabaje en paralelo, tanto el paper como poster para la evaluación final.

6. Bibliografía Relevante

Se ha revisado bibliografía reciente relacionada con sistemas de recomendación basados en emociones. En general, gran parte de los trabajos se enfocan en obtener el estado emocional del usuario a partir de comentarios escritos, imágenes faciales o cuestionarios breves. Sin embargo, no hay muchos trabajos que lo realicen mediante modelos LLMs u otras técnicas de NLP que mapeen prompt a emociones y emociones a juegos, al igual que no está definido el conjunto de emociones con las cuales se debe trabajar, debido a que no hay mucha bibliografía de juegos y a la vez en esta no se específica cuales usan.

Dentro de la literatura revisada se destaca *SENTIMENT-AWARE RECOMMENDATION SYSTEMS IN E-COMMERCE: A REVIEW FROM A NATURAL LANGUAGE PROCESSING PERSPECTIVE,* que da cuenta del estado del arte actual, y *Integrated sentiment analysis with BERT for enhanced hybrid recommendation systems,* que menciona las contribuciones de basar las recomendaciones en sentimientos/emociones y se mencionan por ejemplo BERT y DeepFM como el mejor para el contexto del paper.

A continuación, se deja la literatura revisada:

Tina Babua , Rekha R Naira and Geetha Aa (2023). *Emotion-Aware Music Recommendation System: Enhancing User Experience Through Real-Time Emotional Context*. https://arxiv.org/pdf/2311.10796

Neelesh Vashist, Rohit Kumar, Mukul Bisht, Saurabh Singh (s.f.). *Collaborative and Hybrid Recommender System for Enhanced Movie Recommendations*. https://github.com/NeeleshVashist/Emotion-Enhanced-Movie-Recommender-System/blob/main/PPT%20and%20Report/Social%20Media%20Analytics%20-%20Report.pdf

Sriraj Katkuri, Mahitha Chegoor, Dr.K.C. Sreedhar,M. Sathyanarayana (2024). *Emotion Based Music Recommendation System*. https://www.ijert.org/research/emotion-based-music-recommendation-system-IJERTV12IS050143.pdf

Mr. Mehatab Mahibub Sanadi, Dr. Sujeet More, Prof. Vishal Shinde, Prof. Harshada Raghuvanshi (2024). *Game Recommendation System Based on User Mood*. https://www.ijaresm.com/uploaded_files/document_file/Mr._Mehatab_Mahibub_Sanadi_VnAf.pdf

Yogesh Gajula (2025). *SENTIMENT-AWARE RECOMMENDATION SYSTEMS IN E-COMMERCE: A REVIEW FROM A NATURAL LANGUAGE PROCESSING PERSPECTIVE*. https://arxiv.org/pdf/2505.03828#page=12&zoom=100,92,165

Jinglong Gao, Xiao Ding, Bing Qin, Ting Liu (2023). *Is ChatGPT a Good Causal Reasoner? A Comprehensive Evaluation*. https://arxiv.org/pdf/2305.07375

John Kalung Leung, Igor Griva, William G. Kennedy (2020). *TEXT-BASED EMOTION AWARE RECOMMENDER*. https://arxiv.org/pdf/2007.01455

Ashu Abdul, Jenhui *Chen, Hua-Yuan Liao, Shun-Hao Chang (2018). An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach*. https://www.mdpi.com/2076-3417/8/7/1103

Wenzuixiong Xiong, Yichao Zhang (2023). *An intelligent film recommender system based on emotional analysis*. https://peerj.com/articles/cs-1243/

Liwei Yuan, Maoxiang Ding, Feiyang Meng, Yumin Tian (s.f.). *Sentiment Analysis and Rating Video Game Dimensions via NLP*. https://www.ewadirect.com/proceedings/ace/article/view/20540/pdf

Sanne Vrijenhoek, Gabriel Bénédict, Mateo Gutierrez Granada, Daan Odijk, Maarten de Rijke. *RADio – Rank-Aware Divergence Metrics to Measure Normative Diversity in News Recommendations*. *https://arxiv.org/pdf/2209.13520*

7. Anexos

Anexo A1: Corrección del informe anterior

1. Descripción del problema

Actualmente, los sistemas de recomendación de videojuegos de plataformas como Steam o Epic Games se basan principalmente en los patrones de comportamiento e información explícita de los usuarios. Sin embargo, estos sistemas no consideran el estado emocional o la motivación cambiante de los usuarios. Por ello, se busca desarrollar un ensamble que aproveche tanto la información implícita y explícita, como el estado emocional del usuario.

Para esto se cuenta con 3 dataset de Steam actualizado hace 9 meses[2] que contiene: games.csv (entrega el id del juego, el título, la fecha de lanzamiento, cantidad de reviews, categoría de reviews, precio y demás), recommendations.csv (entrega la recomendación de un usuario[3], el id del usuario, el id del juego, las horas jugadas del usuario, fecha y demás), users.csv (entrega id del usuario, cantidad de productos comprados y cantidad de reviews hechas) y game_metadata.json (entrega el género del juego, descripción y demás).

2. Objetivos del proyecto

- 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 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 descripciones.
- Realizar un ensamble entre ambos modelos mediante un score ponderado para recomendar en base a las emociones también.
- Evaluar los modelos utilizando métricas relevantes como recall@10, MAP@10 y nDCG@10. Además de otras definidas en el estado del arte actual.

3. Implementación de modelos baselines

Los modelos que fueron implementados para mostrar los resultados del análisis fueron Random, Most Popular y Item KNN.

Para comparar estos modelos, se calcularon 3 indicadores: Recall@10, MAP@10 y nDCG@10.

Finalmente, los resultados obtenidos se muestran en la siguiente tabla:

Indicadores	**Random**	**Most Popular**	**Item KNN**
Recall@10	0,0003	**0,0177**	0,0042
MAP@10	0,0003	**0,0203**	0,0007
nDCG@10	0,0053	**0,2027**	0,0015

Tabla 1: Resultados modelos

[2] https://www.kaggle.com/datasets/antonkozyriev/game-recommendations-on-steam?select=games.csv
[3] Si lo recomienda es True y si no es False

Aquí, como podemos ver, los mejores resultados se obtuvieron con Most Popular, pero el modelo IBCF, a diferencia de los otros modelos, se implementó con sólo una muestra de los datos, por lo que se podrían obtener mejores resultados si se hubiera usado la muestra original descrita en la sección 3.

Anexo A2: Ventajas y desventajas de usar un modelo preentrenado

Utilizar el transformer preentrenado DistilRobERTa-base tiene grandes ventajas, debido a que este será mucho más potente y eficiente que cualquier modelo realizable por nosotros. Por ejemplo, al principio como este demoraba bastante en ejecutarse (más de 2 horas en CPU de colab (anexo A5)), se optó por utilizar un enfoque más rápido con una librería llamada NRCLex, que mapeaba textos a emociones, pero lo hacía al encontrar ciertas palabras en los textos, con lo cual no entendía realmente los textos y por ello condujo a resultados mediocres. Con el modelo preentrenado pasa lo contrario, a pesar de demorarse más, entrega asociaciones más precisas basadas en el contexto completo del texto, lo que ayuda a que se realicen buenas recomendaciones.

Por otro lado, respecto a las desventajas, usar este tipo de modelos implica usar las emociones que estos ofrecen, lo cual puede limitar los resultados, dado que podría ser mejor definir categorías o emociones en base al contexto en el cual se trabaja. Esto puede ser llevado a cabo mediante un fine tunning que se realiza con un dataset con etiquetas de las emociones que se buscan utilizar, lo que implica un trabajo exhaustivo que se escapa de los plazos del curso. Por lo tanto, esto involucra una limitación en el proyecto, que se mencionará en evalauciones posteriores.

```
val
```

review_id	numeric_rating_x	title	...	price_original	discount	steam_deck	numeric_rating_y	products	reviews	numeric_rating	deepfm_score	emotion_score	ensemble_score
37649459.0	4.0	NeonCode	...	0.99	0.0	1	4.0	1301.0	346.0	1	0.730812	0.519384	0.667384
15632065.0	5.0	Batman: Arkham City - Game of the Year Edition	...	19.99	0.0	1	5.0	1301.0	346.0	1	0.731059	0.519384	0.667556
23611506.0	4.5	NieR:Automata™	...	0.00	0.0	1	4.5	1301.0	346.0	1	0.731059	0.519384	0.667556
23754347.0	5.0	Undertale	...	0.00	0.0	1	5.0	1301.0	346.0	1	0.731059	0.519384	0.667556
21129658.0	4.5	Red Dead Redemption 2	...	0.00	0.0	1	4.5	1301.0	346.0	1	0.731059	0.519384	0.667556
...	...	...	...	...	...	...	...	...	...	...	...	...	...
13087627.0	4.5	Tom Clancy's Ghost Recon® Wildlands	...	0.00	0.0	1	4.5	458.0	193.0	1	0.731059	0.519384	0.667556
30141370.0	5.0	Gorogoa	...	14.99	0.0	1	5.0	458.0	193.0	1	0.731059	0.519384	0.667556
16649277.0	4.5	Far Cry® Primal	...	29.99	0.0	1	4.5	458.0	193.0	1	0.731059	0.519384	0.667556
11486799.0	4.5	Far Cry® 4	...	29.99	0.0	1	4.5	458.0	193.0	1	0.731059	0.519384	0.667556
18593325.0	4.5	The Witcher: Enhanced Edition Director's Cut	...	9.99	0.0	1	4.5	458.0	193.0	1	0.731059	0.519384	0.667556

Anexo A3: Scores casi iguales por parte de los modelos

```
Ranking por DeepFM:
   app_id                                     title
0   13648              Brothers - A Tale of Two Sons
1   20572                                 911 Operator
2   28532                                      Valheim
3    1303                                       Arma 3
4   18553                   Resident Evil 7 Biohazard
5   25822                                    Cloudpunk
6   21727                                   Far Cry® 5
7    8517                                   God of War
8    4939                              Cult of the Lamb
9   13904  Injustice: Gods Among Us Ultimate Edition

 Ranking por Ensemble:
   app_id                                     title
0   13648              Brothers - A Tale of Two Sons
1   20572                                 911 Operator
2   28532                                      Valheim
3    1303                                       Arma 3
4   18553                   Resident Evil 7 Biohazard
5   25822                                    Cloudpunk
6   21727                                   Far Cry® 5
7    8517                                   God of War
8    4939                              Cult of the Lamb
9   13904  Injustice: Gods Among Us Ultimate Edition
```

Anexo A4: Recomendación para el usuario 3277

```
from transformers import pipeline

emotion_classifier = pipeline("text-classification", model="bhadresh-savani/bert-base-uncased-emotion")

def get_emotions_transformers(text):
    if not text or text.strip() == '':
        return []
    results = emotion_classifier(text)
    # results: lista de dicts con label y score
    # Podemos devolver solo labels con score altos, por ejemplo:
    return [(r['label'], r['score']) for r in results]

games_md['emotion_labels'] = games_md['description'].apply(get_emotions_transformers)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%  935/935 [00:00<00:00, 16.8kB/s]
model.safetensors: 100%  438M/438M [00:06<00:00, 149MB/s]
tokenizer_config.json: 100%  285/285 [00:00<00:00, 20.9kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 10.2MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 2.78MB/s]
special_tokens_map.json: 100%  112/112 [00:00<00:00, 10.5kB/s]
Device set to use cpu
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-54-593a11f6b9be> in <cell_line: 0>()
```

Anexo A5: Primer intento al usar transformer preentrenado