

Sistema Recomendador de Items para League of Legends

Vladimir Araujo

Pontificia Universidad Católica de Chile
Santiago, Chile
vgaraujo@uc.cl

Felipe Rios

Pontificia Universidad Católica de Chile
Santiago, Chile
farios@uc.cl

ABSTRACT

League of Legends es un video juego perteneciente al género *arena de combate multijugador en línea* (MOBA), que durante los últimos años se ha popularizado mundialmente. La idea central es controlar un personaje y equiparlo con items durante la partida, que mejoran sus estadísticas base, con el fin de obtener la victoria de su equipo. Es por esta razón que la recomendación de items o predicción de compras es un tema de interés en este género, debido a que diferentes factores de una partida puede afectar la decisión de compra, dando como resultado un mayor o menor impacto del personaje para obtener la victoria. En este trabajo se propone dos sistemas recomendadores de items, dado el personaje seleccionado, y sus enemigos. El primero esta basado en un sistema de asociación de reglas para la generación de set de items frecuentes, el cual obtuvo un 58 % de media de precisión. El segundo esta basado en redes neuronales para predicciones de probabilidad, para obtener un set de items, obteniendo un 75 % de media de precisión.

KEYWORDS

League of legends, sistema recomendador, asociación de reglas, redes neuronales

1. INTRODUCCIÓN

Los videojuegos multijugadores en línea vieron varios avances la década pasada en jugabilidad y tecnología. Algunos los consideran una actividad con un cierto grado de interacción superficial [8]. Sin embargo, según [9] que cita a la Comisión Olímpica Asiática “son los mayores juegos multideportivos después de los Juegos Olímpicos”. El mercado es tan grande, que según [13] para 2021 éste alcanzará hasta los 43.000 millones de dólares.

League of Legends (LoL) es un juego muy popular perteneciente al género MOBA (Multiplayer Online Battle Arena). El juego consiste en dos equipos (rojo y azul) de cinco jugadores cada uno, los cuales compiten por ser el primero en destruir la base enemiga. Cada jugador tiene control de un *campeón* con el que puede interactuar con otros jugadores. Estas interacciones adoptan la forma de combates entre los avatares, las cuales se llevan a cabo en un mapa de tres carriles. Cada carril esta defendido por torretas las cuales deben ser destruidas para poder alcanzar la base del enemigo.

Cada personaje tiene capacidades y habilidades únicas, que evolucionan durante el transcurso de una partida. Los jugadores reciben oro por matar a sus oponentes, capturar posiciones y eliminar pequeños bots llamados súbditos. Con esos recursos obtenidos, cada jugador puede comprar items (armas, armaduras, pociones, etc) especializados con los que puede potenciar tanto ofensivamente como defensivamente su personaje [19].

League of Legends tiene dos problemas interesantes de selección que pueden ser abordados por sistemas recomendadores: “¿Que



Figura 1: El problema de recomendación consiste en establecer relaciones entre el campeón seleccionado y su equipo enemigo. De esta manera es posible agregar contexto a la recomendación de ítems.

personaje llevar a la batalla?” y “¿Qué items comprar durante una partida?”. Con una selección de 136 campeones para equipos de a 5 ($\approx 3,6 \cdot 10^9$ combinaciones) y 93 items con máximo espacio permisible de 6 ($\approx 7,6 \cdot 10^9$ combinaciones), el problema combinatorial es grande.

La comunidad que practica estos juegos publica continuamente guías en Internet. Éstas son recomendaciones de expertos, para el uso adecuado de un cierto personaje. Se trata de guías empíricas de las tácticas y estrategias que les dan buenos resultados, en las cuales uno de sus aspectos más importantes es la recomendación de items, lo que se conoce por *builds*.

Estas recomendaciones pueden resultar o muy generales o muy específicas, según los problemas que vieron estos expertos. No se espera que en cada caso estén bien adecuadas al contexto de una nueva instancia. En otras palabras, las recomendaciones no suelen tomar en cuenta los contrincantes, siendo esta información muy importante a la hora de seleccionar una *build* que permita al personaje tener impacto durante la partida (Figura 1).

Dada esta problemática, en este trabajo se proponen dos sistemas de recomendación de items para campeones de *League of Legends*, tomando en cuenta su contexto (enemigos). Para lograr esto se utilizó la experiencia de la comunidad, de un dataset con partidas de clasificatoria. Como hipótesis se propuso que, la información de contexto se encuentra implícitamente en el dataset, según la estrategia de compra del usuario. Debido al tipo de recomendación requerida, se exploró técnicas de minería de datos y machine learning aplicada a sistemas recomendadores.

Este artículo se estructura de la siguiente forma: La sección 2 presenta un breve resumen de los trabajos relacionados con sistemas recomendadores y su aplicación en el género MOBA. La sección

3 presenta la metodología, incluyendo el dataset y los sistemas propuestos. La sección 4 presenta la evaluación y resultados de los sistemas recomendadores. Finalmente, la sección 5 presenta las conclusiones del trabajo.

2. TRABAJOS RELACIONADOS

En los últimos años los sistemas recomendadores han tomado gran fuerza. En un principio el *Netflix Price* funcionó como motor impulsor de estas tecnologías. Ahora, con un *ecommerce* más y más exigente la necesidad de estos sistemas es cada vez más apremiante. Esta técnica ha encontrado más de un nicho, con usuarios siempre en la búsqueda de nuevas experiencias muchas son las áreas en las que estos sistemas son aplicados.

Las técnicas para minería de datos han sido también usadas para desarrollar sistemas recomendadores [20, 22], su enfoque es encontrar los artículos que coincidan mejor con la preferencia de un usuario. Las tareas de minería se clasifican en descriptiva y predictiva, en las cuales podemos encontrar por un lado técnicas de asociación de reglas y por el otro técnicas de clasificación [2].

Apriori es una de los algoritmos más usados dentro de las técnicas de asociación de reglas. En [16] presentan un modelo basado para recomendación de amigos en microblogs. Mediante la modificación de apriori, pudieron optimizar el escaneo de los datos, logrando menores tiempos de generación. Además, alcanzaron un rendimiento superior a los recomendadores no personalizados. El algoritmo apriori también ha sido usado para lograr sistemas híbridos tomando características de sistemas basados en contenidos [1] y basados en filtrado colaborativo [12].

Las redes neuronales artificiales (ANN) son un campo de estudio con mucho éxito en varias áreas. En los últimos años varias aplicaciones han sido implementadas para sistemas recomendadores. Las podemos encontrar desde filtrado colaborativo [5, 7, 11] hasta basado en contenido [18] llegando hasta aproximaciones híbridas que suponen modelos embebidos de ANN entre filtrado colaborativo y el uso de contenido [24]. Por supuesto, algunos de estos enfoques parten resolviendo el problema con retroalimentación explícita [5, 7] mientras que otros persiguen el enfoque actual, más desafiante, de incluir retroalimentación implícita [11, 15].

2.1. Sistemas de Recomendación Aplicados a los MOBA

Tal como lo señala [19] existen muchos trabajos que tienen como foco de interés los juegos MOBA. De entre ellos, el problema relacionado a nuestro interés específico ha sido tratado con dos enfoques particulares. Existen los sistemas que intentan hacer predicciones para asistir con la selección experta (o para revisión de terceros) y aquellos que buscan la recomendación directa.

La literatura es diversa sobre el tema de la selección de campeones. [23] intenta predecir la selección de campeones para *Dota 2* utilizando redes bayesianas y redes neuronales recurrentes. Por otro lado [25] observa que una regresión logística tiene un mejor desempeño que un perceptrón multicapa. En su enfoque [10] construyen un recomendador de héroes basado en reglas de asociación para ayudar a formar el equipo. Al mismo tiempo utiliza un predictor basado en redes neuronales para intentar diagnosticar el desenlace

de la partida, mejorando por mucho los trabajos anteriores. [4] propone un sistema de recomendación basado en árboles de búsqueda con algoritmos de Montecarlo. En este artículo, el agente genera un árbol combinatorial con las posibilidades de equipos. Se centra en buscar en él aquellas ramas que maximizan el radio de victorias.

La recomendación de items también es examinada como foco de interés por algunos autores. Por un lado, en el género MMORPG (siglas en inglés para massively multiplayer online role-playing game), se utiliza sistemas de combates y especificaciones de items similares a los MOBA. El trabajo de [14] utiliza coeficientes de similitud de preferencias para filtrado colaborativo para abordar el problema. Por otro lado, está el trabajo de [17], en el cual utilizan métodos de asociación de reglas, regresión para generar recomendaciones, y *clúster* para asociar los items con campeones. Esta última metodología, concluye su autor, no necesariamente mejora las recomendaciones, pero permite asociar mejor los roles de los personajes entre sí y vincular items adecuados a estos roles.

De la revisión del estado del arte se pudo observar varios trabajos relacionados con sistemas recomendadores aplicados a videojuegos pertenecientes al género MOBA. Sin embargo un problema menos trabajado por la comunidad científica, es el relacionado a la recomendación de items para un personaje tomando en cuenta el contexto de la partida.

3. METODOLOGÍA

En este trabajo se proponen dos aproximaciones que permiten resolver la tarea de recomendación de items para un personaje de LoL. La Figura 2 muestra el marco de trabajo propuesto, este se basa en la metodología clásica de minería de datos [22], pero con ciertos cambios que permiten introducir este proceso al trabajo de recomendación.

El primer paso es el preprocesamiento del dataset que comprende el filtrado, reordenamiento y codificación. Luego el proceso de análisis, que en conjunto con el proceso de interpretación comprenden el sistema recomendador. El motor de recomendación ejecuta los modelos basados en asociación de reglas y redes neuronales para finalmente generar un set de items como recomendación.

3.1. Preprocesamiento del Dataset

En este estudio, un dataset con 184070 partidas de clasificación de LoL fue usado. Cada una de estas partidas comprenden la participación de 10 usuarios de alto nivel y experiencia por lo cual es más probable que sus compras de items para sus personajes sean basadas en el análisis de la partida.

El dataset original consta de varios archivos, los cuales contienen información relacionada con los campeones, items, y estadísticas obtenidas durante una partida. Este dataset no incluye información de la cuenta del usuario, por esta razón nos hemos enfocado en trabajar con la información de los campeones y los items.

Debido a que se desea recomendar items para un campeón seleccionado dado su contexto (enemigos), se decidió generar un nuevo archivo el cual incluye solo la información útil para este trabajo. Adicionalmente, se modificaron dos archivos originales de items y de campeones. Los detalles del dataset final se describen a continuación:

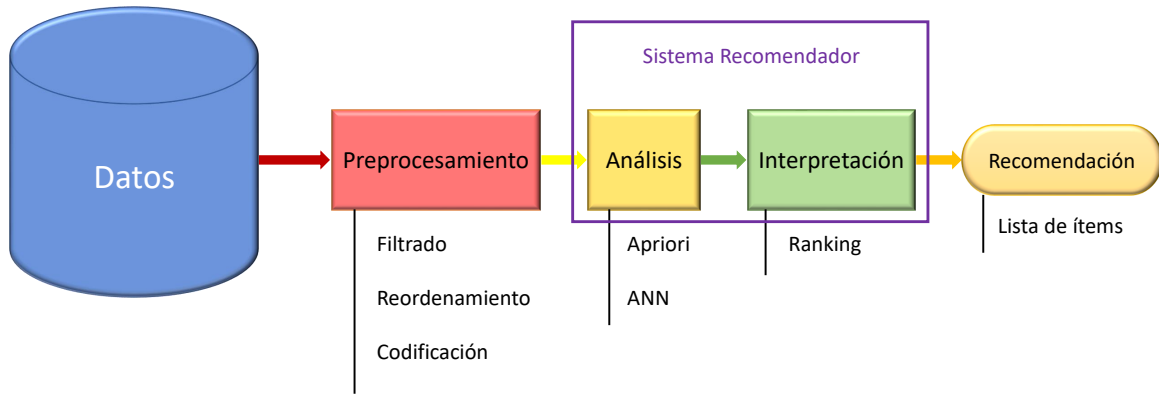


Figura 2: Diagrama general del marco de trabajo.

- Archivo de partidas: Este fue el archivo generado, el cual incluye información relevante de varios archivos del dataset original. Este archivo comprende la participación de los campones durante una partida de la temporada 7, y los valores que contienen son el identificador del campeón usado, los identificadores de los items usados, un identificador de partida ganada o perdida, y los identificadores de los aliados y enemigos. Para su uso en los modelos propuestos, solo las partidas con victoria fueron usadas y además fue dividido en un archivo de entrenamiento con 631590 instancias y uno de prueba con 70405 instancias.
- Archivo de campeones: Este archivo incluye los nombres de los campones y su identificador. Todos los campeones lanzados después de la temporada 7 fueron removidos, dando un total de 136 finalmente.
- Archivo de items: Este archivo contiene los nombres de los items y su identificador. Solo los items finales se mantuvieron, además, los items consumibles también fueron removidos. Finalmente, 93 items fueron los obtenidos.

Los dos archivos mencionados al final, fueron usados únicamente para identificar los nombres de los items y campeones del archivo de partidas.

3.2. Sistemas Recomendadores

3.2.1. *Recomendación Basada en Asociación de Reglas.* Se propone el uso de técnicas de asociación de reglas para la generación de set de items. El algoritmo apriori ya fue usado para Dota2 con buen resultado en [17], pero en esta implementación, proponemos una aproximación un poco diferente.

La Figura 3 muestra la arquitectura del sistema propuesto. Por un lado, existe un bloque encargado de extraer los set de items frecuentes haciendo uso de reglas de asociación, generadas por el algoritmo apriori. Dado que las estrategias de compra tienden a ser específicas para cada campeón, se realiza el proceso por cada personaje. Por otro lado, el bloque *buscar* se encarga de hallar los set de items asociados a la consulta para su recomendación. La consulta se realiza indicando el campeón seleccionado para la partida y sus cinco enemigos.

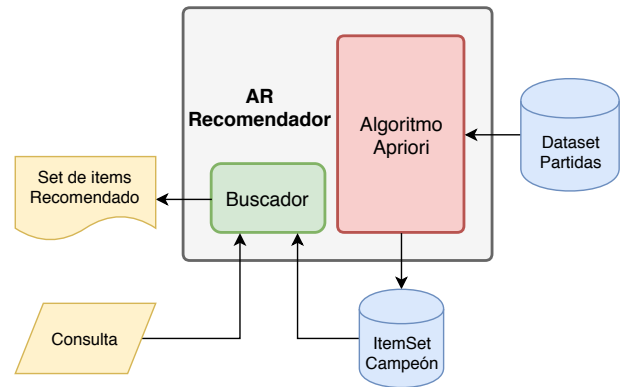


Figura 3: Arquitectura de Sistema de Recomendación por Asociación de Reglas.

Apriori mediante $X, Y = \phi_A(D | supp_{min}, conf_{min})$, genera reglas de asociación de la forma $X \rightarrow Y$, donde Y un campeón enemigo y X es un subset de items.

D es el dataset de transacciones compuesta por los campeones enemigos y los ítems usados por el personaje. El soporte se define como la frecuencia porcentual de observación de un mismo item en todas las transacciones. El valor de confianza se podrá definir como la probabilidad $P(Y | X)$. Tanto para la confianza y soporte se debe establecer un valor mínimo $supp_{min}$ y $conf_{min}$.

De esta manera es posible encontrar los set de ítems más frecuentes $I_f = X$. Pero, dado que un set de items I_f puede contener un sub set de items frecuentes J_f , se decidió usar los item set máximos, definidos como $I_m = I_f | I_f \wedge \nexists J_f \supset I_f$. Estos sets de items son los usados para recomendar al usuario.

Para esta implementación se usó el paquete PyFIM [3], el cual incluye varias técnicas para el minado de reglas de asociación. Como se mencionó anteriormente, se usó el algoritmo apriori, con soporte entero mínimo 4 y confianza mínima del 80%.

3.2.2. *Recomendador Basado en Redes Neuronales.* La arquitectura de nuestro recomendador consiste en un perceptrón multicapa, con

una capa oculta. El perceptrón tiene 136 neuronas en la primera capa, 150 en la oculta y 93 en la de salida. La red establece sus propios patrones considerando funciones de activación *relu* en la capa de entrada y la oculta, y *softmax* en la capa de salida.

Los 136 campeones fueron codificados utilizando one hot. Mientras que el campeón que recibe la recomendación ha sido codificado con un 1, cada uno de los campeones del equipo enemigo entra en la misma codificación con un -1. En las partidas de clasificación los campeones no pueden repetirse por lo que esta codificación no entra en conflicto con la información en la base de datos.

Los 93 items finales fueron codificados de similar manera. Considerando las metodologías vistas en la bibliografía [4], solamente fueron etiquetados los items de este tipo con codificación one hot que formaron parte de los campeones del equipo victorioso en la partida. Esta codificación equivale a una retroalimentación implícita de la observación de la transacción de un item.

Para el entrenamiento de la red se consideró una suposición probabilística en la observación de los items en las partidas ganadoras. La etiqueta binaria ayudó con este propósito. De esta manera resultó adecuado utilizar el optimizador *Adam* con función de pérdida de *entropía binaria cruzada*. El código fue escrito usando la biblioteca *Keras* de *Tensorflow*.

Finalmente la recomendación se toma ranqueando los items con más probabilidad de transacción según sugiere la ANN.

4. EVALUACIÓN Y RESULTADOS

Por un lado, el sistema recomendador basado en apriori generó desde 4 hasta 114706 reglas por cada personaje. Estas reglas tienen un promedio de confianza de 99.97%. Por otro lado, el sistema basado en redes neuronales logró una precisión de 95.80% con el conjunto de entrenamiento y un error del 12%.

Para la evaluación de estos recomendadores con el conjunto de datos de prueba, se propone la generación de set de items @N de diferentes tamaños y su evaluación con métricas de exactitud de clasificación que son ampliamente usadas para evaluar, si un conjunto de items recomendados son o no de interés [6].

La Tabla 1 muestra la comparación entre los dos sistemas propuestos. *Precision* presenta la medida de eficiencia de items relevantes seleccionados. En contraste, *Recall* mide la amplitud del resultado. Se puede apreciar el comportamiento clásico de un sistema recomendador, *Precision* decae cuando la lista de recomendación es más grande, mientras *Recall* incrementa. Para dar una visión general del recomendador tomando en cuenta las dos métricas mencionadas, se presenta también *F1 score*, la cual presenta un comportamiento estable en ambos recomendadores. Pero es claro

Cuadro 1: Resultados para @N-itemsets Recomendados

Sistema	Precision	Recall	F1	MAP	MRR
AR@3	0.41	0.68	0.50	0.57	0.63
ANN@3	0.60	0.69	0.53	0.75	0.71
AR@6	0.40	0.62	0.48	0.58	0.63
ANN@6	0.44	0.69	0.53	0.75	0.81
AR@9	0.41	0.75	0.51	0.56	0.63
ANN@9	0.34	0.80	0.47	0.72	0.81

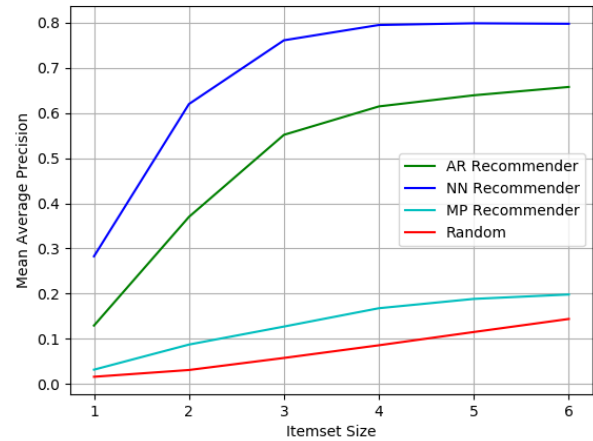


Figura 4: Association Rules Recommendation System Architecture.

que el sistema basado en redes neuronales supera en sus resultados al sistema basado en apriori.

Además, basado en la experiencia de campo, se puede afirmar que el contexto situacional tiene cierta importancia en el juego. Existen items que son recomendados por la comunidad, como compras en primera instancia, por lo que la posición en la recomendación es un foco de interés. Para ello fue incluido en este análisis las métricas MAP y MRR. Aquí también se puede observar que la red neuronal da mejores resultados.

Dado que no existe una línea base para este problema, se consideró comparar las recomendaciones de los sistemas contra una recomendación aleatoria y una recomendación de items más populares. La recomendación *Most Popular*, es hallada por la mayor frecuencia de uso de ítems en partidas victoriosas de toda la temporada 7. La Figura 4 muestra las cuatro tipos de recomendaciones comparadas con diferentes tamaños de listas de recomendación con métrica MAP.

Se puede apreciar que ambos sistemas recomendadores propuestos superan las recomendaciones *Most Popular* y *Random*, las cuales llegaron a un máximo de 20% y 14% respectivamente. También es posible observar una vez más como el sistema basado en redes neuronales supera a todos los demás llegando a casi 80%.

5. DISCUSIÓN DE LAS SOLUCIONES

Como ya ha sido mencionado, una de las razones para que la comunidad científica se haya centrado más en el problema de recomendación de campeones, es que la comunidad lúdica publica *builds* en foros. Éstas son recomendaciones expertas de ítems que los jugadores entregan basándose en su experiencia. Pero la estrategia de compra juega un papel muy importante, debido a que muchas veces las transacciones de los items son situacionales a las necesidades particulares de la instancia de juego. A continuación se presenta un análisis cualitativo de un ejemplo concreto de recomendación.

Para el caso ejemplo de la Figura 1, se buscó una recomendación experta en [21], y se generó recomendaciones con los sistemas propuestos y *Most Popular*. La Figura 5 muestra los resultados obtenidos de manera gráfica. Donde:

- A: recomendación experta
- B: recomendación *Most Popular*
- C: recomendación basada en reglas
- D: recomendación basada en red neuronal

El contexto de este caso de análisis comprende cuatro enemigos que realizan daño mágico y uno de ellos además realiza gran daño físico. Se puede observar que la recomendación A contiene solo un ítem que incrementa la resistencia mágica base, mientras que las recomendaciones B, C y D, contienen dos. En este caso se prefiere mayor resistencia mágica debido a los enemigos.

Al fijarnos en la estrategia de recomendación de los ítems más populares nos encontramos con una violación de las reglas del juego. El sistema recomienda los ítems B3 y B4 que no pueden ser comprados en conjunto. Los recomendadores C y D no cometen este error. La probabilidad calculada de observar ambos ítems al mismo tiempo es mínima, por lo que tal contradicción no es observada en ambos casos.

Por otro lado, cuatro de los campeones enemigos tienen habilidades de control de movimiento, en otras palabras pueden paralizar a sus adversarios por cierto tiempo. En este caso se busca usar un ítem que reduzca ese tiempo de aturdimiento, la recomendación B3, C1 y D2 es permite hacer eso.

Fianlmente, dos de los enemigo tienen mucha resistencia física. Por tal razón, el recomendador de D considera el ítem D5 que permite reducir armadura al adversario. Mientras que el recomendador C, propone el uso de el ítem C2 el cual permitiría resistir más en peleas grupales mientras hace más daño.

La calidad de las recomendaciones C y D, permiten comprender que la información de contexto está siendo utilizada. Para este ejemplo en concreto, una set de ítems con mejor en la resistencia mágica es altamente deseable, por lo cual la recomendación experta no es adecuada. Pero también se necesita reducir la armadura del adversario y en este caso, existe la coincidencia de que estos fueron ítems populares.

6. INTERFAZ GRÁFICA DE USUARIO

Como trabajo adicional, se desarrolló una interfaz gráfica (GUI), la cual permite observar de manera visual las recomendaciones. Esto facilitó la realización el análisis cualitativo del sistema mostrado en la sección anterior. Esta GUI fue diseñada para formar parte del marco de trabajo propuesto, de esta manera nuevas aproximaciones para este recomendador pueden ser añadidas con facilidad.

Esta GUI permitió probar cada sistema con una recomendación a la vez, como se muestra en la Figura 6. El resultado fue desplegado en el mismo formato gráfico que las *builds*: con el ítem más relevante a la izquierda. La Figura 1 presenta la recomendación para el ejemplo de la Figura 5.

7. CONCLUSIONES

Los resultados evidencian que es posible realizar una recomendación de un set de ítems para un campeón, considerando un equipo

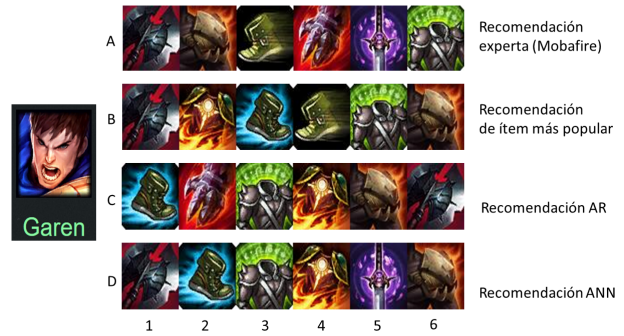


Figura 5: Recomendación entregada por el sistema recomendador para cada técnica aplicada (fila C AR y fila D ANN). Esta recomendación es comparada gráficamente contra una *build* popular (fila A) y contra una recomendación de los ítems más populares utilizados con ese campeón (fila B).



Figura 6: Interfaz de usuario gráfica para recomendaciones de los sistemas propuestos.

enemigo como su contexto. Esto se logra dotando a los sistemas del contexto durante el procesamiento de la información.

Considerando el efecto general de las métricas sobre el modelo, se puede concluir que ambos sistemas son adecuados para hacer una recomendación confiable para el juego. Sin embargo, la red neuronal presenta un mejor desempeño que el algoritmo basado en reglas. Esto puede ser debido a que la red logra capturar relaciones de más alto nivel.

El sistema recomendador basado en asociación de reglas mostró un rendimiento no tan destacable, pero las reglas generadas tienen una alta confianza. Lo que permitiría generar un motor de explicación en el cual, el usuario pueda observar el porque de la recomendación haciendo alusión a esos porcentajes de confianza.

Como trabajo futuro se propone agregar más técnicas de recomendación en el marco de trabajo. Se propone además, agregar la información de los equipos perdedores observada para reducir la introducción de sesgos.

Además se desea usar estos sistemas directamente con un grupo de la comunidad con el fin de realizar una evaluación cualitativa. En este tipo de trabajos, el punto de vista del usuario y su experiencia de uso es importante para obtener más conclusiones relevantes.

REFERENCIAS

- [1] Ahmed Alsalama. 2015. A Hybrid Recommendation System Based on Association Rules. *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 9, 1 (2015), 367–370. <https://doi.org/10.1145/2792838.2796545>
- [2] Xavier Amatriain and Josep M. Pujol. 2015. Data mining methods for recommender systems. In *Recommender Systems Handbook, Second Edition*. Springer US, Boston, MA, 227–262. https://doi.org/10.1007/978-1-4899-7637-6_7 arXiv:arXiv:1011.1669v3
- [3] Christian Borgelt. 2018. PyFIM - Frequent Item Set Mining for Python. <http://www.borgelt.net/pyfim.html>.
- [4] Zhengxing Chen, Truong-Huy D Nguyen, Yuyu Xu, Chris Amato, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 2018. The Art of Drafting: A Team-Oriented Hero Recommendation System for Multiplayer Online Battle Arena Games. (jun 2018). <https://doi.org/10.1145/nnnnnnn.nnnnnnn> arXiv:1806.10130
- [5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.
- [6] Joost de Wit. 2008. Evaluating recommender systems : an evaluation framework to predict user satisfaction for recommender systems in an electronic programme guide context. <http://essay.utwente.nl/59711/>
- [7] Robin Devooght and Hugues Bersini. 2016. Collaborative filtering with recurrent neural networks. *arXiv preprint arXiv:1608.07400* (2016).
- [8] Nicolas Ducheneaut, Nicholas Yee, Eric Nickell, and Robert J Moore. 2006. Alone together?: exploring the social dynamics of massively multiplayer online games. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 407–416.
- [9] Kirstin Hallmann and Thomas Giel. 2018. eSports–Competitive sports or recreational activity? *Sport management review* 21, 1 (2018), 14–20.
- [10] Lucas Hanke and Luiz Chaimowicz. 2015. A Recommender System for Hero Line-Ups in MOBA Games. *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference* (2015), 43–49.
- [11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.
- [12] JinHyun Jooa, SangWon Bangb, and GeunDuk Parka. 2016. Implementation of a Recommendation System Using Association Rules and Collaborative Filtering. In *Procedia Computer Science*, Vol. 91. Elsevier, 944–952. <https://doi.org/10.1016/j.procs.2016.07.115>
- [13] JuniperResearch. 2017. WORLD OF WARCRAFT OTHER MMO/MOBA GAMES TO APPROACH 25 % OF THE \$100BN DIGITAL GAMES MARKET IN 2017. <https://www.juniperresearch.com/press/press-releases/world-of-warcraft-other-mmo-moba-games-to-approa>.
- [14] SG Li and L Shi. 2014. The recommender system for virtual items in MMORPGs based on a novel collaborative filtering approach. *International Journal of Systems Science* 45, 10 (2014), 2100–2115.
- [15] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. *arXiv preprint arXiv:1803.05170* (2018).
- [16] Li Liu, Shuo Yu, Xiang Wei, and Zhaolong Ning. 2018. An improved Apriori-based algorithm for friends recommendation in microblog. *International Journal of Communication Systems* 31, 2 (jan 2018), e3453. <https://doi.org/10.1002/dac.3453>
- [17] Wenli Looi, Manmeet Dhaliwal, Reda Alhaji, and Jon Rokne. 2018. Recommender System for Items in Dota 2. *IEEE Transactions on Computational Intelligence and AI in Games* (2018), 1–1. <https://doi.org/10.1109/TG.2018.2844121>
- [18] Damian Franken Manning and Omar Emad Shams. 2017. Content filtering with convolutional neural networks. US Patent App. 15/354,377.
- [19] Marçal Mora-Cantallops and Miguel-Ángel Sicilia. 2018. MOBA games: A literature review. *Entertainment computing* (2018).
- [20] Maryam Khanian Najafabadi, Azlinah Hj Mohamed, and Mohd Naz’ri Mahrin. 2017. A survey on data mining techniques in recommender systems. , 28 pages. <https://doi.org/10.1007/s00500-017-2918-7>
- [21] MOBAFIRE NETWORK. 2018. Mobafire. <https://www.mobafire.com/>.
- [22] Mehrbakhsh Nilashi. 2016. An Overview of Data Mining Techniques in Recommender Systems. *Journal of Soft Computing and Decision Support Systems* 3, 6 (oct 2016), 16–44. <http://jscdss.com/index.php/files/article/view/116>
- [23] Adam Summerville, Michael Cook, and Ben Steenhuisen. 2016. Draft-Analysis of the Ancients: Predicting Draft Picks in DotA 2 Using Machine Learning. *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference Godec* (2016), 100–106. www.aaai.org
- [24] Hao Wang and Wu-Jun Li. 2015. Relational collaborative topic regression for recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2015), 1343–1355.
- [25] Weiqi Wang. 2016. Predicting Multiplayer Online Battle Arena (MOBA) Game Outcome Based on Hero Draft Data. (dec 2016). <http://trap.ncirl.ie/2523/>