Exploración y minería de datos Historia de Hearthstone

Rodrigo Esteban Valenzuela Cardenas
Departamento de Computación e Informática
Universidad de la frontera
Temuco, Chile
r.valenzuela07@ufromail.cl

Pablo Nahuelpán
Departamento de Computación e Informática
Universidad de la frontera
Temuco, Chile
p.nahuelpan@ufromail.cl

Nicolás Hidalgo
Departamento de Computación e Informática
Universidad de la frontera
Temuco, Chile
n.hidalgo02@ufromail.cl

Armin Patricio Rodríguez Garrido
Departamento de Computación e informática
Universidad de la frontera
Temuco, Chile
a.rodriguez09@ufromail.cl

***Abstract*— Se detalla el proceso de exploración y minería sobre los datos que se atribuyen a la historia de Hearthstone y sus diversos mazos a través de las distintas expansiones. Se exploran y analizan los resultados obtenidos.**

***Index Terms*—*EDA, minería de datos, mazos, expansiones, kmodes, árboles, clustering, clasificación*.**

I. INTRODUCCION

Hearthstone es un juego de cartas coleccionables creado por la empresa Blizzard [1]. Este juego ha visto hasta 6.5 millones de jugadores concurrentes y debido a su popularidad sitios como Hearthpwn.com [2] fueron desarrollados para permitirle a los jugadores una herramienta para crear y compartir mazos de prueba.

Hearthpwn nos ha encargado con la búsqueda de factores que permitan clasificar mazos de cartas en categorías que no estaban definidas dentro de las diferentes plataformas que serán futuramente absorbidas por el sitio. Esto con el objetivo de clasificar los datos no cubiertos por ellas como los nuevos formatos de juegos implementados que son estándar y salvaje.

En este reporte se analizarán resultados relacionados a la clasificación de mazos a través de las categorías provistas por parte de Hearthpwn para asistir en la introducción de datos de mazos externos a su plataforma. El objetivo final será el desarrollo de uno o más modelos experimentales que puedan asistir a esta tarea.

En el conjunto de datos se encuentran registrados distintos mazos subidos a la plataforma de Hearthpwn desde el año 2013 hasta 2017 y fueron proporcionados por el sitio Kaggle [3].

II. METODOLOGÍA

En primer lugar y con el objetivo de familiarizarnos con los datos y entender mejor su relevancia se lleva a cabo el denominado Análisis exploratorio de datos (EDA por sus siglas en inglés). Este análisis se basa en técnicas estadísticas y gráficas, a través de tablas o gráficos, que permiten identificar y resumir los datos.

A continuación, se describen los métodos utilizados para la obtención de los datos establecidos.

Basados también en el EDA, se identifican preguntas o problemas que consideramos de interés, y que puedan ser resueltos a través de técnicas de minería de datos.

A. RESULTADOS EDA

Para comenzar con el análisis de datos, se realizó una limpieza de los datos, en donde se eliminan variables sin relevancia de acorde al contexto planteado, dejando únicamente aquellas que ofrecían información relevante, las variables antes mencionadas son:

- craft_cost
- title
- user
- rating
- deck_id

También se tomaron en cuenta algunas variables que poseen datos faltantes, erróneos o nulos, los cuales son:

- deck_archetype

Para comenzar con la exploración del dataset, se tomó en cuenta que el año en donde se lanzó oficialmente el jiuego fue el 21 de marzo del 2014, se filtraron los decks creados antes de esta fecha, con lo cual el dataset quedó con un total de 346232 filas y 36 columnas.

Una vez finalizada la fase de filtrado, se da comienzo a la fase de exploración de datos.

En primer lugar, se debe tener en cuenta que a partir del 02 de febrero del 2016, se anuncian dos tipos de formatos principales para el juego, salvaje (wild en inglés) y el formato estándar (standard en inglés), comenzando por el lanzamiento de la expansión “Old Gods”, es decir que dada una cierta cantidad de tiempo, las cartas del formato estándar dejarán de pertenecer a este, para posteriormente ser partes del formato salvaje. Al existir dos estilos de juegos principales, es necesario analizarlos de maneras individuales, ya que ambos presentan diferentes razas y mazos fuertes.

Una vez filtrado en base a la expansión “Old Gods” al igual que eliminar variables como “Unknown” y “Edit”, el dataset presenta 30087 filas y 36 columnas, estos datos se separan en dos subconjuntos “deck_w” (para el formato salvaje) y “deck_s” (para el formato estándar),

posteriormente se analiza qué raza presenta una mayor predominancia en cada formato.

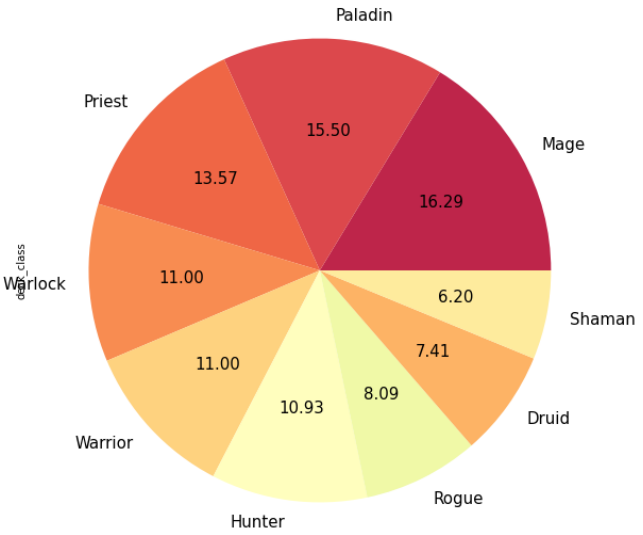

Fig. 1.- Razas populares en formato Salvaje.

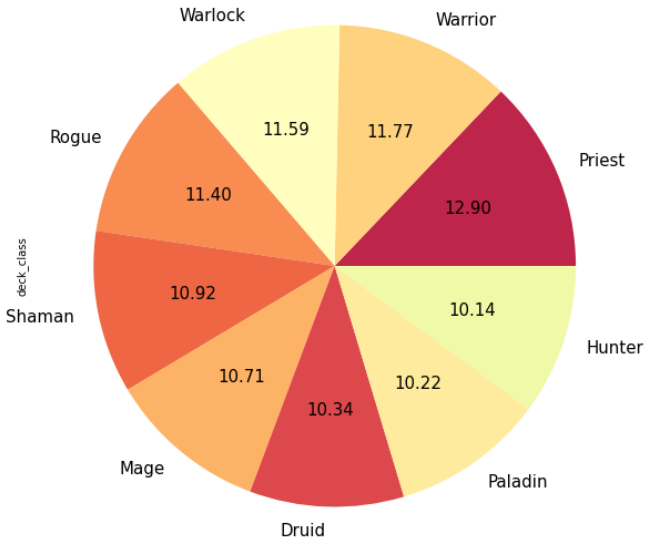

Fig. 2.- Razas populares en formato Estándar.

Según la figura 1, en el formato salvaje se aprecia que la raza más utilizada es el mago, mientras que la figura 2 muestra que, en el formato estándar, la raza más utilizada es el sacerdote.

Como se puede apreciar, ambos formatos presentan razas y estilos de juegos diferentes, esto podría ocurrir ya que, en el formato estándar, las cartas rotan según a las expansiones que se encuentren en dicho formato, en otras palabras, que hay varias cartas que pasan del formato estándar al salvaje, en periodos de tiempos no tan prolongados.

Ahora que se tiene un conocimiento extendido de cómo se componen los datos del dataset, se puede dar comienzo al análisis solicitado por la página hearthpwn, la cual no ha encomendado resolver un problema de clasificación, esta se realizará con un concepto que en la expansión "old gods" no estaba incorporado y este es clasificar un mazo al formato que pertenece, y también luego del análisis con los arquetipos de mazos más utilizados no se obtuvo grandes conclusiones, pero se puede incorporar un problema para resolver, por lo tanto en vez de ver cuáles son los arquetipos más utilizados por raza hearthpwn quiere incorporar cuáles son las cartas que más se utilizan en un mazo con un arquetipo en especial.

B. PREGUNTAS Y PROBLEMAS

Según los resultados obtenidos en el EDA se ha decidido abordar distintas preguntas que podrían ser relevantes para unas pruebas experimentales. Por lo que se abordarán las siguientes preguntas:

- Problema de clasificación binaria basado en la variable "deck_format": consiste en una variable que es categórica que se comporta de manera binaria correspondiente a un atributo que hace referencia al formato de un mazo, y en este problema se debe encontrar la forma de poder clasificar el formato de los nuevos mazos que serán registrados en la plataforma.

- Problema basado en encontrar asociaciones entre variables categóricas de un mazo. Dado que se registraron mazos con sus respectivos atributos, en este problema se debe encontrar cuáles son los atributos de un mazo que se asocian entre sí para obtener información relevante, ya que dentro de la plataforma se quiere implementar una sección de recomendación a través de la información que se obtenga de las asociaciones de atributos.

C. PROPUESTAS EXPERIMENTALES

Luego de haber planteado los problemas en el punto anterior, se formularon las siguientes propuestas para resolver las preguntas.

- “Para enfrentar el problema 1, vamos a clasificar con 2 modelos de predicción de tipo categórico, estos son árbol de decisión y bosques aleatorios con el fin de identificar qué modelo es mejor para poder clasificar el formato de un nuevo mazo, y los resultados que nos entreguen los dos modelos se evaluaran respecto a la métrica de "Recall", debido a que el valor para este problema proviene de ser capaces de detectar decks Wild, los cuales son considerados como el TP de esta situación.”

- “Para enfrentar el problema 2, vamos a utilizar reglas de asociación de los atributos de un mazo para encontrar reglas de asociación utilizando un modelo con el que será entrenado con el algoritmo Apriori y esto será medidos por la métrica de "support" y la razón de elegir esta medida es porque nos entrega un valor de la cual podemos saber si se puede confiar en las transacciones para obtener la información que necesitamos dependiendo del valor que nos entregue.”

D. EXPERIMENTOS

- Experimento 1

Para comenzar con el experimento, se utilizó el subconjunto de "filter_deck", esto debido a que contiene todos los datos que fueron necesarios para trabajar los modelos.

Luego de utilizar el modelo de árbol de decisión como el primer modelo, en este se apreció un rendimiento bastante alto en todas las métricas, lo cual podría significar la presencia de Overfitting, sin embargo, en la matriz de confusión se pudo ver que la distribución TP y TN reflejo este comportamiento casi perfecto.

	precision	recall	f1-score	support
S	0.99	0.99	0.99	6876
W	0.93	0.93	0.93	646
accuracy			0.99	7522
macro avg	0.96	0.96	0.96	7522
weighted avg	0.99	0.99	0.99	7522

```
Matriz de confusión:
 [[6834   42]
 [  45  601]]
```

Fig 3: Resultados Arbol de decisión

	precision	recall	f1-score	support
S	0.99	1.00	1.00	6881
W	1.00	0.92	0.96	641
accuracy			0.99	7522
macro avg	1.00	0.96	0.98	7522
weighted avg	0.99	0.99	0.99	7522

```
Matriz de confusión:
 [[6881    0]
 [  51  590]]
```

Fig 4: Resultados Random Forest

Al investigar más a fondo se descubrió que, aunque las variables utilizadas si son categóricas en comportamiento, debido a que poseen un alto número de valores únicos, factor que afecta en gran medida a los algoritmos de árboles que utilizamos, nuestra elección de algoritmos no fue la correcta.

Debido a esto fue necesario buscar una nueva propuesta experimental.

Propuesta Experimental 1.2:

- " En consecuencia del experimento anterior, se decide abordar el problema 1 como un problema de agrupación, basado en el algoritmo Kmode [3], variante de K Means pensada para clustering usando variables categóricas, y se mantendrá el foco en mantener el Recall de los decks Wild lo más alto posible."

- Experimento 1.2

El primer paso de este experimento fue el filtrado de datos, Kmodes está diseñado para variables categóricas, por lo que "date" fue removido, además sumamos a esto "deck_class" y "deck archetype", que no traen información relevante al formato y por lo mismo pueden afectar negativamente los resultados.

A continuación modificamos el atributo "deck_format" para convertir sus valores "S" y"W" a valores binarios numéricos, debido a que al estar basado en K Means, Kmodes requiere el uso de una clase binaria numérica para funcionar con clases binarias.

Un tema para considerar es que Kmodes realiza su agrupación con la utilización de "Líderes", existen 3 métodos disponibles para la obtención de estos:

- Random: Elige de forma aleatoria sus líderes, y compara en múltiples ciclos que líderes tuvieron mejores resultados.
- Huang: Utiliza un criterio más especializado para la elección de sus líderes, también a través de múltiples ciclos.
- Cao: Otro método más especializado, pero solo a través de un ciclo único.

Todos estos métodos son no deterministas, por lo que un resultado negativo no significa que no exista la chance de un mejor modelo, en este caso escogimos el método "Huang", que es más especializado, pero tiende a tomar más tiempo por cada ciclo de ejecución.

Nuestro resultado inicial fue el siguiente:

	precision	recall	f1-score	support
0	0.91	0.86	0.89	13758
1	0.09	0.14	0.11	1286
accuracy			0.80	15044
macro avg	0.50	0.50	0.50	15044
weighted avg	0.84	0.80	0.82	15044

```
Matriz de confusión:
 [[11838  1920]
 [ 1104   182]]
Costo de prediccion:  417107.0
```

Fig 5: Resultados Kmodes Huang

Contrario a lo que buscamos nuestros resultados sobre mazos Wild (clase 1) tuvieron un rendimiento deficiente, por otro lado, podemos observar que no existe el mismo grado de Overfitting observado en el experimento anterior.

Otra observación que decidimos seguir fue el hecho que la presencia de mazos Standard (clase 0) es 10 veces más alta que la de nuestra clase objetivo, por lo que el modelo podría verse beneficiado por el uso de oversampling y undersampling [4].

Realizamos undersampling a nuestros mazos Standard, y Oversampling a nuestros mazos Wild, para dejar a ambos en un punto medio de los datos, decidimos en contra de simplemente duplicar mazos Wild hasta alcanzar a su contraparte debido a que puede generar un alto grado de sesgo dentro de los datos.

Luego de varias ejecuciones del algoritmo, se llegó a un modelo final.

```
              precision    recall  f1-score   support

           0       0.60      0.78      0.68      6932
           1       0.67      0.47      0.55      6792

    accuracy                           0.62     13724
   macro avg       0.64      0.62      0.61     13724
weighted avg       0.64      0.62      0.61     13724

Matriz de confusión:
 [[5399 1533]
 [3629 3163]]
Costo de prediccion:  384617.0
```

Fig 6: Resultados experimento 1.2

- Experimento 2

En el segundo experimento se cambió el lenguaje con el cual se estaba trabajando, siendo este cambio de python a R, ya que este último posee las librerías necesarias para poder utilizar la regla de asociación.

Una vez importado todo lo necesario para comenzar con el experimento, se realizó un análisis previo de los atributos en el cual se decidió no utilizar los atributos enumerados "card_0", "card_1" ... "card_29", ya que estos no entregan información relevante a la hora de realizar asociaciones.

Luego, cuando se tuvo el dataset con el cual se podían encontrar las asociaciones de los atributos de los mazos, se procedió a observar los itemsets más frecuentes.

```
   items                                          support   count
[1] {deck_type=Ranked Deck}                       0.9156114 27548
[2] {deck_format=S}                               0.9121880 27445
[3] {deck_format=S, deck_type=Ranked Deck} 0.8411274 25307
```

Fig 7: Itemsets más frecuentes.

Tras haber analizado los itemsets, se ha encontrado que "deck_archetype" no aparece entre los más frecuentes, lo que claramente indica un mal indicio para que exista una asociación. Sin embargo para comprobar estos resultados, se ejecutó el algoritmo "apriori" el cual se utiliza para encontrar las reglas.

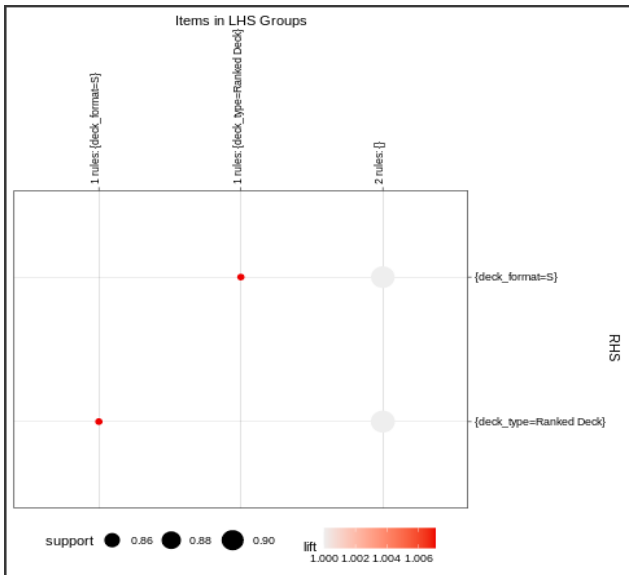

Fig 8: Gráfica resultante del algoritmo Apriori.

Este resultado indica que si existen relaciones entre las variables "deck_format" y "deck_type", no obstante, para fines del análisis deseado, no existe relación con la variable "deck_archetype", lo cual indica que este problema no es abordable, ya que la inexistencia de relación con esta variable puede deberse a que en los datos utilizados en las columnas, son, en muchos casos, cualitativos, lo que implica que se deberían de parsear los datos de cada mazo, lo que causaría que se tomara un análisis de lo que se debe parsear y podría llegar a presentar errores luego de este proceso. También se debe a la precaria cantidad de atributos restantes luego de haber realizado la limpieza de las columnas que se encontraban irrelevantes para este análisis.

III. RESULTADOS

Luego de haber desarrollado los experimentos, se obtuvieron los siguientes resultados:

- En el problema uno, se puede apreciar que la tasa del recall en tipo Wild es del 47% y también se puede observar que, en la categoría de estándar la precisión, fue solo del 60%, esto indica que no es despreciable la posibilidad de que se puedan introducir falsos positivos.

- En el segundo problema, se lograron observar dos asociaciones de las cuales sí se podría realizar un análisis, esta relación se aprecia con las variables "deck_type" y "Ranked Deck", a lo que podríamos decir que aquellos usuarios que creaban un mazo para jugar dentro del formato estándar, lo hacen principalmente para jugar de manera competitiva, en este caso en específico, el problema sería abordable, sin embargo entrega muy poca información, lo que implicaría realizar un parseo a los datos dentro de las columnas que hacen referencia a las cartas, esto debido a que las columnas "card_0", "card_1"..."card_29", solo contiene la Id de la carta, lo cual significa que no proporciona información.

IV. ANÁLISIS DE LOS RESULTADOS

En el experimento 1 y 1.2 observamos que en nuestro mejor caso, sólo alcanzamos un 47% de recall, lo cual no es una cifra satisfactoria, esto no es tan grave por sí solo, pero debido a que la precisión de clasificación de decks standard es de un 60%, en casos como este experimento donde existe una brecha grande de cantidades entre ambos formatos, es muy posible que la gran mayoría de los decks que sean clasificados como nuestra clase objetivo, sean solo falsos positivos.

Un factor importante que se debe tener en cuenta es que la predicción pudo verse afectada por la disparidad en la cantidad de soporte que hay por clase, es decir que si hubiese un mayor número de mazos en Wild naturales (que no hayan sido generados por oversampling), la clasificación de este sería probablemente diferente, dado que habría una menor incertidumbre sobre la existencia de sesgo en la clasificación.

Dentro de los resultados del experimento 2, como se mencionó anteriormente se logra obtener una regla de

asociación {deck_format=S} => {deck_type=Ranked Deck} de la cual se puede inferir que los usuarios que registraron un mazo con el formato Estándar, generalmente lo creaban para juego competitivo/Ranked, y sólo se obtiene una regla de la cual se puede obtener información, y para abordar un problema de regla de asociación se deben utilizar más atributos y por esta razón el problema es abordable pero entrega muy poca información relevante como para tener una sección de recomendaciones en la plataforma.

V. Conclusiones

Al finalizar el desarrollo de este proyecto, podemos decir que los resultados obtenidos no fueron satisfactorios para la resolución de los problemas planteados, no consideramos sin embargo que el trabajo realizado haya sido en vano.

Los resultados del problema 1 por ejemplo no tuvieron el rendimiento deseado, pero no se descarta la posibilidad de que con una mayor cantidad de datos el resultado pudiese haber sido mejor, o teniendo a disposición más tiempo para realizar iteraciones del algoritmo Kmodes exista un modelo con mayor rendimiento que no se pudo encontrar a tiempo.

Adicionalmente en el problema 2 debido a las limitaciones de tiempo y el hecho que el contenido de reglas de asociación fue lo último cubierto por la asignatura, no se descarta la existencia de una mejor forma de utilizar este algoritmo.

Finalmente, como grupo hemos podido distinguir que, al finalizar el trabajo, los procesos para realizar minería de datos y sus variantes son difíciles de comprender en primera instancia, pero si se tiene tiempo para estudiar a profundidad estos temas entonces se pueden abrir muchas puertas, por ejemplo, con todo lo que tiene que ver con Machine Learning.

VI. Contribuciones

- Pablo Nahuelpan: Redacción de preguntas y propuestas, desarrollo de experimentos del problema 2.
- Nicolás Hidalgo: Redacción y Exploración de datos, observaciones en los gráficos de barra sobre formato Estándar y Salvaje.
- Armin Rodriguez: Limpieza de datos y apoyo en exploración de datos en la sección de los gráficos de pie.
- Rodrigo Valenzuela: Redacción reporte y presentación, desarrollo de preguntas, retroalimentación de cambios, desarrollo de experimentos del problema 1.

References

[1] HeartStone. (2014, 11 marzo). [Software]. En Blizzard Entertainment (24.2). Peter McConnell. https://hearthstone.blizzard.com/es-es

[2] HeathPwn. (2022). HeathPwn. https://www.hearthpwn.com/

[3] H. Bonthu, "KModes Clustering Algorithm for Categorical data," *Analytics Vidhya*, Jun. 13, 2021. https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/

[4] K. Pykes, "Oversampling and Undersampling," *Medium*, Sep. 10, 2020. https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf

Anexos

Repositorio: ID-2022-2-Hearthstone

Video Presentación: https://youtu.be/ObWtALxCChc

Set de datos: History of Hearthstone. https://www.kaggle.com/datasets/romainvincent/history-of-hearthstone

TP: True positive (Positivo verdadero)
TN: True negative (Positivo negativo)
FP: False negative (Falso negativo
FN: False negative (Falso negativo)