De los datos a la imitación: estudio y modelado del comportamiento de jugadores en Wordle
From Data to Imitation: Study and modeling Player Behavior in Wordle

Trabajo de Fin de Grado
Curso 2022–2023

Autor
Óscar Calvet Sisó

Director
Antonio Alejandro Sánchez Ruíz-Granados

Doble grado en Matemáticas e Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

De los datos a la imitación: estudio y modelado del comportamiento de jugadores en Wordle
From Data to Imitation: Study and modeling Player Behavior in Wordle

Trabajo de Fin de Grado en Ingeniería Informática

Autor
Óscar Calvet Sisó

Director
Antonio Alejandro Sánchez Ruíz-Granados

Convocatoria: *Junio 2023*

Doble grado en Matemáticas e Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

29 de MAYO de 2023

Resumen

Este proyecto se centra en el estudio del comportamiento de jugadores humanos y en la imitación de sus estilos de juego. Para ello utilizaremos Wordle un popular rompecabezas de palabras del cuál se dispone un gran número partidas publicadas en redes sociales.

El proyecto comienza con la recopilación de un conjunto de datos de partidas y su preprocesamiento. A continuación, se aplican técnicas de análisis de datos para explorar el conjunto. Esto incluye estadísticas descriptivas, visualización de datos y algoritmos de aprendizaje automático. Después de la fase de análisis, se desarrolla un marco de modelado para simular el comportamiento del jugador. Los modelos están diseñados para imitar las estrategias y los procesos de toma de decisiones observados. Se emplean técnicas de aprendizaje automático, como algoritmos genéticos, para entrenar estos modelos utilizando los datos procesados. El rendimiento de los modelos desarrollados se evalúa a través de métricas rigurosas con el objetivo de evaluar su capacidad de imitar a los jugadores humanos y si pueden lograr tasas de éxito similares.

Los resultados obtenidos ayudan a comprender el comportamiento de los jugadores de Wordle y mejorar potencialmente el rendimiento de estos. Además, los modelos desarrollados en este proyecto podrían aplicarse a varios otros rompecabezas basados en palabras, proporcionando información sobre estrategias y procesos de toma de decisiones que pueden mejorar las experiencias de juego de los jugadores humanos.

Palabras clave

Wordle, análisis de datos, modelado de jugadores, algoritmos genéticos, juegos.

Abstract

This project focuses on the study of the behavior of human players and the imitation of their playing styles. For this we will use Wordle, a popular word puzzle of which a large number of games are published on social networks.

The project starts with collecting a game data set and pre-processing it. Data analysis techniques are then applied to explore the set. This includes descriptive statistics, data visualization, and machine learning algorithms. After the analysis phase, a modeling framework is developed to simulate player behavior. The models are designed to mimic observed strategies and decision-making processes. Machine learning techniques such as genetic algorithms are used to train these models using the processed data. The performance of the developed models is evaluated through rigorous metrics with the aim of evaluating the ability of the models to imitate human gamers and whether they can achieve similar success rates.

The results obtained help to understand the behavior of Wordle players and potentially improve their performance. Additionally, the models developed in this project could be applied to various other word-based puzzles, providing insights into strategies and decision-making processes that can improve gaming experiences for human players.

Keywords

Wordle, data analysis, player imitation, genetic algorithms, games.

Índice

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

"Even though I play it every day, I still feel a sense of accomplishment when I do it: it makes me feel smart, and people like that"
— Josh Wardle, creador de Wordle

1.1. Motivación

Wordle es un juego de palabras que revolucionó la red social *Twitter* a principios del año 2022, fue creado por Josh Wardle y actualmente pertenece al periódico *The New York Times* formando parte del repertorio de pasatiempos de las ediciones digitales. El juego consiste en adivinar una palabra de cinco letras que cambia cada día, con el objetivo de adivinarla en seis o menos intentos. El jugador recibe información acerca de cuán similar es la palabra introducida a la palabra secreta después de cada intento en forma de cuadros de colores que indican si una letra en la palabra adivinada es correcta, incorrecta o está en el lugar equivocado. El juego ha ganado una inmensa popularidad, con jugadores compartiendo sus logros en las redes sociales y compitiendo con sus amigos para obtener la mayor racha de aciertos en la menor cantidad de intentos. En cada intento el jugador introduce una palabra en una cuadrícula y el juego colorea las letras de esta de color verde, amarillo o gris en función de las siguientes reglas:

- La letra introducida se colorea en verde si se encuentra en la palabra objetivo y ocupa la misma posición.
- La letra introducida se colora en amarillo si aparece en la palabra objetivo pero no se encuentra en la misma posición.
- La letra introducida se colorea en gris si no aparece en la palabra objetivo.
- Multiples apariciones de la misma letra en la palabra introducida serán coloreadas de verde o amarillo siguiendo las reglas anteriores solamente si estas aparecen multiples veces en el objetivo. En cualquier otro caso, las letras repetidas sobrantes se colorearán de gris.

Figura 1.1: Partida de ejemplo

Para ejemplificar mejor las normas descritas, en la figura 1.1 se pueden observar las combinaciones de colores devueltas por las palabras introducidas en una partida cuyo objetivo es *abbey*.

El éxito de Wordle se puede atribuir a los principios de ludificación que incorpora. Una de las razones principales por las que a la gente le gusta jugar es la sensación de logro y superación asociada al éxito. Wordle está diseñado para proporcionar *feedback* instantáneo y recompensas a los jugadores, lo que lo hace más satisfactorio y atractivo que un juego de palabras tradicional. La simplicidad y facilidad de uso del juego también lo hacen accesible a una amplia gama de jugadores, desde jugadores casuales hasta entusiastas más serios [29]. Otra forma en que Wordle utiliza la ludificación para motivar a los jugadores es a través de la competencia. El juego anima a los jugadores a competir contra sus amigos y otros jugadores, fomentando un sentido de comunidad y camaradería. La naturaleza competitiva del juego crea una sensación de urgencia y emoción que mantiene a los jugadores comprometidos y motivados a seguir jugando.

Esto ha provocado una gran afición por Wordle y consecuentemente su comunidad de jugadores ha aumentado y evolucionado, diversificándose en el proceso y generando jugadores con estrategias variadas [46]. Pese a que no existen datos acerca de los jugadores previos a la adquisición por parte del *New York Times* y que este periódico mantiene en privado las estadísticas que posee del juego, sí que existen varios artículos de la propia empresa donde se detallan varios tipos de jugadores y sus estrategias [2], [16]. Tras recopilarlos hemos obtenido las siguientes categorías:

- **El jugador ocasional**: El jugador casual se relaciona con Wordle como una actividad de ocio, en busca de entretenimiento y estimulación mental. Disfrutan de la simplicidad del juego y, a menudo, juegan de forma intermitente. Los jugadores casuales generalmente confían en la intuición, la asociación de palabras y el conocimiento del vocabulario medio para realizar sus intentos. Suelen utilizar únicamente palabras candidatas a ser solución aunque no las eligen de manera idónea para disminuir el tamaño del conjunto de palabras restantes.

- **El cazador de patrones**: Los cazadores de patrones prestan una gran atención al feedback proporcionado por el juego. Observan cómo cambia la palabra con cada respuesta y usan esta información eficientemente para disminuir notablemente el conjunto de palabras restantes. Pese a esto, siguen utilizando casi exclusivamente palabras candidatas a ser solución en cada intento.

- **El estratega**: Los estrategas juegan a Wordle con una estrategia premeditada de prueba y error. Sus intentos iniciales representan una amplia gama de posibilidades buscando reducir al máximo el espacio de soluciones y luego ajustan sus intentos posteriores en función del *feedback* recibido. Este proceso claramente diferenciado en dos partes cuyo cambio se da entre el segundo y tercer intento permite a este grupo adivinar la palabra con una media de intentos inferior a los otros dos.

Estas caracterizaciones nos permiten obtener una primera aproximación de los tipos de jugadores, no obstante al no poseer los datos de los que se ha inferido la información no nos resultan particularmente útiles. A pesar de esto, Wordle cuenta con una funcionalidad que nos va a permitir obtener millones de partidas.

En la primera semana del año 2022, Wordle tenía alrededor de trescientos mil usuarios diarios activos [46], en la segunda semana este número aumentó hasta los tres millones [21]. Este incremento coincidió con la implementación de la funcionalidad de compartir las partidas en diversas redes sociales que fue lanzada el primer día del año, esta mejora permitió a los usuarios generar publicaciones que contenían caracteres UTF-8 correspondientes a cuadrados con los colores obtenidos en sus intentos. Este formato permitió que los jugadores pudieran publicar y compartir sus partidas sin desvelabar la palabra objetivo del día.

La popularidad de estos *posts* sacudió *Twitter* por completo, en las dos primeras semanas del año 2022 se publicaron más de un millón doscientas mil partidas en formato *tweet* [8]. La red social no tardó en ser inundada por partidas de cientos de miles de usuarios y Wordle se convirtió en un fenómeno de masas.

Durante el transcurso del año *Twitter* se ha convertido en un repositorio de partidas de Wordle gigantesco. Pese a que las publicaciones no contienen las palabras literales introducidas por los usuarios, la información proporcionada por los colores y los metadatos de los *posts* nos permiten aventurar la dificultad de las palabra objetivo del día y construir una idea general de la evolución de la habilidad de los jugadores.

Este inusual fenómeno, acompañado de la masiva cantidad de datos reales generados y de mi afición por los puzzles de palabras me ha motivado a centrar el tema de este trabajo en la obtención, análisis y estudio de estas partidas. Estamos ante una gran cantidad de datos que todavía no se han analizado en profundidad. Por ello en este trabajo pretendemos acceder a estos, procesarlos, generar un conjunto de partidas, hacerlo público, analizar el comportamiento de los jugadores, clasificarlos e intentar imitarlos mediante algoritmos de aprendizaje automático.

El uso de algoritmos de aprendizaje automático y de la inteligencia artificial (IA) no es arbitrario, la IA ha tenido un impacto significativo en muchas industrias, incluyendo la de los juegos. Si bien los videojuegos han sido uno de los focos principales de la investigación en esta, los juegos más casuales también han visto un impacto importante de este desarrollo. La relación entre la investigación en IA y los juegos se puede ver en muchas áreas, como el diseño, la jugabilidad y la experiencia del jugador [50] y, además, se ha utilizado en el diseño de juegos para crear oponentes inteligentes y desafiantes para estos [41]. Analizando los datos de la jugabilidad y las estrategias humanas, se pueden desarrollar algoritmos eficaces e inteligentes que simulan el comportamiento de los jugadores humanos [37]. Esto ha resultado en una jugabilidad más desafiante y estratégica para los jugadores, así como en una experiencia más agradable para aquellos que prefieren batirse contra oponentes no humanos [7].

En términos de jugabilidad, la IA se ha utilizado para crear experiencias más realistas e inmersivas. La IA puede crear oponentes inteligentes que ajustan su juego en función de los movimientos realizados por el jugador humano. Esto crea una experiencia más realista e inmersiva para los jugadores, así como una jugabilidad más desafiante y dinámica.

El uso de la IA en el que nos vamos a centrar en este proyecto es el de imitar el comportamiento humano, existen múltiples aproximaciones entre las cuales destacamos las siguientes:

- **Aprendizaje por imitación**: también conocido como clonación conductual. Implica entrenar a agentes de IA para imitar a jugadores humanos aprendiendo de demostraciones de expertos. Esta técnica se ha aplicado con éxito en varios dominios de juego. Por ejemplo, en los juegos de Atari [35] donde se entrenó una red neuronal convolucional mediante una variante del algoritmo de *Q-learning* [48], también se pueden utilizar estas técnicas para acelerar el aprendizaje de algunos algoritmos como el 'DQN' de *DeepMind* que aprende a jugar de manera autónoma a partir de un conjunto de dimensionalidad muy alta [36], con un notable coste computacional, no obstante, el uso de partidas humanas en el comienzo del entrenamiento, permite reducir notablemente el tiempo de aprendizaje [14]. Este enfoque resulta muy útil ya que permite a los agentes imitar a jugadores humanos de cualquier nivel siempre que se disponga de suficientes datos y capacidad de cómputo.

- **IA adaptativa**: Imitar jugadores no solo se limita a replicar sus acciones, sino que también implica comprender sus procesos de toma de decisiones y su comportamiento. El modelado de jugadores tiene como objetivo capturar y predecir las acciones, preferencias y estrategias de los jugadores humanos. A diferencia de los sistemas de aprendizaje por imitación, los sistemas de modelado son flexibles y pueden adaptarse a nuevos datos, son particularmente útiles en la creación de adversarios que mejoran sus capacidades a lo largo de la partida gracias a los *inputs* proporcionados por los jugadores [51]. Los algoritmos utilizados en estos casos son mucho más variados, por ejemplo se utilizan redes de destilación experta que se entrenan mediante variables adaptativas del algoritmo *Q-learn* [47] y también existen aproximaciones mediante redes neuronales generativas adversarias (GAN) [17, 20].
- **Retroalimentación humana**: Otro enfoque para imitar a los jugadores implica sesgar el aprendizaje directamente usando el *feedback* proporcionado por los jugadores. Es decir, los modelos se entrenan con partidas reales y tras observar las acciones realizadas, se evalúa su comportamiento. Posteriormente los jugadores pueden realizar indicaciones o cambios que modifican el comportamiento del agente en una situación determinada sesgando así el aprendizaje futuro. Este método, también conocido como aprendizaje interactivo a partir de demostraciones, permite que los sistemas de IA imiten y se adapten a las preferencias y matices de los jugadores humanos aunque requiere una supervisión constante [10, 33, 32].

1.2. Objetivos

Con este trabajo se pretende:

O1. Recopilar, limpiar y publicar un conjunto de datos de partidas.

O2. Analizar los datos de partidas y del vocabulario usado.

O3. Encontrar patrones en los datos analizados, especialmente aquellos correspondientes a distintos tipos de jugadores.

O4. Crear programas capaces de imitar a los diferentes tipos de jugador encontrados.

O5. Comparar los resultados de los modelos con partidas reales.

O6. Evaluar los diversos modelos generados y obtener datos sobre su aprendizaje y los usuarios que intentan modelar.

O7. Comprender la base de jugadores, en particular la relación del rendimiento de estos y su vocabulario, así como, las estrategias empleadas.

O8. Caracterizar a los jugadores según los patrones encontrados y su vocabulario inferido.

En general, este proyecto busca proporcionar información valiosa sobre el comportamiento del jugador y caracterizarlo.

1.3. Plan de trabajo

Para lograr los objetivos descritos en el apartado anterior, se ha acordado realizar reuniones cada dos semanas entre el autor y el director del trabajo. Considerando lo anterior, se han establecido los siguientes hitos para organizar el trabajo a realizar en este proyecto:

- Obtención de los datos. (Del 5 al 19 de octubre de 2022)
- Limpieza de los datos. (Del 19 de octubre al 16 de noviembre de 2022)
- Realización del análisis exploratorio de los datos. (Del 16 de noviembre al 8 de diciembre de 2022)
- Procesamiento y creación de clústers sobre los datos. (Del 8 de diciembre de 2022 al 22 de febrero de 2023)
- Creación de modelos de inteligencia artificial basados en los resultados obtenidos en la etapa anterior. (Del 22 de febrero al 29 de marzo de 2023)
- Evaluación de los modelos y selección de los modelos. (Del 29 de marzo al 12 de abril de 2023)
- Obtención de información a partir de los modelos y el procesamiento de datos. (Del 12 al 26 de abril de 2023)

Dentro del tiempo de cada objetivo también se contabiliza el tiempo dedicado a la redacción de la memoria del proyecto. Hemos decidido mantener la última quincena antes de la entrega sin asignar para terminar la memoria y también como tiempo extra en caso de que surja algún contratiempo.

1.4. Código fuente

El desarrollo de los distintos apartados de este proyecto se ha realizado en su integridad en el lenguaje `Python`, haciendo uso de la herramienta `Jupyter Notebook` que permite almacenar los gráficos generados. El proyecto en su totalidad se encuentra en el repositorio de *Github* público siguiente `https://github.com/OscarCal/ImitatingWordlePlayers` con licencia `MIT` organizado por capítulos.

Además el conjunto de partidas generado se encuentra también en un repositorio público de la página *Kaggle* bajo la url: `https://www.kaggle.com/datasets/scarcalvetsis/wordle-games` con una licencia `CC BY-SA 4.0`.

Capítulo 2

Introduction

"Even though I play it every day, I still feel a sense of accomplishment when I do it: it makes me feel smart, and people like that"
— Josh Wardle, creador de Wordle

2.1. Motivation

Wordle is a word game that revolutionized the social network Twitter at the beginning of the year 2022, was created by Josh Wardle and currently belongs to the newspaper *The New York Times*, being part of the repertoire of pastimes of digital editions. The game involves guessing a five-letter word that changes every day, with the goal of guessing it in six or fewer tries. The player receives feedback on how similar the word entered is to the secret word after each attempt in the form of colored squares that indicate whether a letter in the guessed word is correct, incorrect or in the wrong place. The game has gained immense popularity, with players sharing their achievements on social networks and competing with their friends to get the longest streak of correct guesses in the fewest number of tries. On each attempt the player enters a word into a grid and the game colors the letters in the grid green, yellow or gray based on the following rules:

- The entered letter is colored green if it is in the target word and occupies the same position.
- The entered letter is colored yellow if it appears in the target word but is not in the same position.
- The entered letter is colored gray if it does not appear in the target word.
- Multiple occurrences of the same letter in the entered word will be colored green or yellow following the above rules only if they appear multiple times in the target. In any other case, the remaining repeated letters will be colored gray.

Figura 2.1: Example of a game

To better exemplify the rules described above, in figure 2.1 you can see the color combinations returned by the words entered in a game whose target word is *abbey*.

Wordle's success can be attributed to the principles of gamification it embodies. One of the main reasons people like to play is the sense of accomplishment and achievement associated with success. Wordle is designed to provide instant feedback and rewards to players, making it more satisfying and engaging than a traditional word game. The game's simplicity and ease of use also make it accessible to a wide range of players, from casual gamers to more serious enthusiasts [29]. Another way Wordle uses gamification to motivate players is through competition. The game encourages players to compete against their friends and other players, fostering a sense of community and camaraderie. The competitive nature of the game creates a sense of urgency and excitement that keeps players engaged and motivated to keep playing.

This has led to a great fondness for Wordle and consequently its community of players has grown and evolved, diversifying in the process and generating players with varied strategies [46]. Although there is no data on players prior to the acquisition by the New York Times and the newspaper keeps its statistics on the game private, there are several articles by the company itself detailing various types of players and their strategies [2, 16]. After compiling them we have obtained the following categories:

- **The casual gamer**: The casual gamer relates to Wordle as a leisure activity, seeking entertainment and mental stimulation. They enjoy the simplicity of the game and often play intermittently. Casual players generally rely on intuition,

word association and knowledge of average vocabulary to make their attempts. They tend to use only candidate solution words although they do not choose them ideally to decrease the size of the remaining word set.

- **Pattern hunter**: Pattern hunters pay close attention to the feedback provided by the game. They observe how the word changes with each response and use this information efficiently to natably discriminate the remaining set of words. Despite this, they still almost exclusively use candidate solution words on each attempt.

- **The strategist**: Strategists approach Wordle with a premeditated trial-and-error strategy. Their initial attempts represent a wide range of possibilities seeking to reduce the solution space as much as possible and then adjust their subsequent attempts based on the feedback received. This clearly differentiated two-part process whose change occurs between the second and third attempt allows this group to guess the word with a lower average number of attempts than the other two.

These characterizations allow us to obtain a first approximation of the types of players, but since we do not have the data from which the information has been inferred, they are not particularly useful. However, Wordle has a functionality that will allow us to obtain millions of games.

In the first week of the year 2022, Wordle had around three hundred thousand active daily users [46], in the second week this number increased to three million [21]. This increase coincided with the implementation of the game sharing functionality on various social networks that was launched on the first day of the year, this enhancement allowed users to generate posts containing UTF-8 characters corresponding to squares with the colors obtained in their attempts. This format allowed players to post and share their games without revealing the target word of the day.

The popularity of these posts shook Twitter completely, in the first two weeks of the year 2022 more than one million two hundred thousand games were published in the tweet format [8]. The social network was soon flooded with games from hundreds of thousands of users and Wordle became a mass phenomenon.

Over the course of the year *Twitter* has become a gigantic repository of Wordle games. Although the posts do not contain the literal words entered by users, the information provided by the colors and metadata of the posts allow us to guess the difficulty of the day's target words and to build a general idea of the evolution of the players' skill.

This unusual phenomenon, together with the massive amount of real data generated and my fondness for word puzzles, has motivated me to focus the subject of this paper on the collection, analysis and study of these games. We are faced with a large amount of data that have not yet been analyzed in depth. Therefore, in this work we intend to access these data, process them, generate a set of games, make

them public, analyze the behavior of the players, classify them and try to imitate them by means of machine learning algorithms.

The use of machine learning algorithms and artificial intelligence (AI) is not arbitrary; AI has had a significant impact on many industries, including gaming. While video games have been a major focus of AI research, more casual games have also seen a significant impact from this development. The relationship between AI research and games can be seen in many areas, such as design, gameplay and player experience [50], and has also been used in game design to create intelligent and challenging opponents for games [41]. By analyzing gameplay data and human strategies, efficient and intelligent algorithms can be developed that simulate the behavior of human players [37]. This has resulted in more challenging and strategic gameplay for players, as well as a more enjoyable experience for those who prefer to battle against non-human opponents [7].

In terms of gameplay, AI has been used to create more realistic and immersive experiences. The AI can create intelligent opponents that adjust their gameplay based on the moves made by the human player. This creates a more realistic and immersive experience for players, as well as more challenging and dynamic gameplay.

The use of AI that we are going to focus on in this project is to imitate human behavior, there are multiple approaches among which we highlight the following:

- **Imitation learning**: also known as behavioral cloning. It involves training AI agents to mimic human players by learning from expert demonstrations. This technique has been successfully applied in several gaming domains. For example, in Atari [35] games where a convolutional neural network was trained using a variant of the *Q-learning* algorithm. [48], it is also possible to use these techniques to accelerate the learning of some algorithms such as the 'DQN' of *DeepMind* which learns to play autonomously from a very high dimensionality set [36], with a remarkable computational cost, however, the use of human games at the beginning of the training, allows to reduce significantly the learning time [14]. This approach is very useful as it allows agents to imitate human players of any level as long as sufficient data and computational power are available.
- **Adaptative AI**: Mimicking players is not only limited to replicating their actions, but also involves understanding their decision-making processes and behavior. Player modeling aims to capture and predict the actions, preferences and strategies of human players, unlike imitation learning systems, modeling systems are flexible and can adapt to new data, they are particularly useful in the creation of adversaries that improve their capabilities throughout the game thanks to the inputs provided by the players [51]. The algorithms used in these cases are much more varied, for example expert distillation networks are used that are trained by adaptive variables of the algorithm *Q-learn* [47] and there are also approaches using generative adversarial neural networks (GAN) [17, 20].

- **Human feedback**: Another approach to imitate players involves biasing learning directly using the feedback provided by the players. That is, models are trained with real games and after observing the actions performed, their behavior is evaluated. Subsequently, the players can make indications or changes that modify the agent's behavior in a given situation, thus biasing future learning. This method, also known as interactive learning from demonstrations, allows AI systems to mimic and adapt to the preferences and nuances of human players although it requires constant monitoring [10, 33, 32].

2.2. Goals

With this project we aim to:

O1 Collect, clean and publish a dataset of Wordle games.

O2 Analyze the games and the vocabulary used in them.

O3 Find patterns in the analyzed data, especially those corresponding to different types of players.

O4 Create programs capable of imitating the different types of players found

O5 Compare the results of the models with real games.

O6 Evaluate the various models generated and obtain data on their learning and the users they attempt to model.

O7 Understand the player base, in particular the relationship of player performance and vocabulary, as well as the strategies employed.

O8 Characterize players according to the patterns found and their inferred vocabulary.

Overall, this project seeks to provide valuable information on player behavior and characterize the player.

2.3. Work plan

In order to achieve the objectives described in the previous section, it has been agreed to hold meetings every two weeks between the author and the director of the work. Considering the above, the following milestones have been established to organize the work to be done in this project:

- Data collection (From October 5 to October 19, 2022)
- Data cleaning (From October 19 to November 16, 2022)
- Exploratory data analysis (From November 16 to December 8, 2022)

- Data processing and clustering (December 8, 2022 to February 22, 2023).
- Creation of artificial intelligence models based on the results obtained in the previous stage (From February 22 to March 29, 2023).
- Evaluation of the models and selection of the models (From March 29 to April 12, 2023).
- Obtaining information from the models and data processing (From April 12 to April 26, 2023).

Within the time of each objective we also count the time dedicated to the writing of the project report. We have decided to keep the last fortnight before delivery unallocated to finish the report and also as extra time in case of any setbacks.

2.4. Source code

The development of the different sections of this project has been done entirely in the `Python` language, making use of the tool `Jupyter notebook` that allows to store the generated graphics, the project in its entirety is in the following public *Github* repository `https://github.com/OscarCal/ImitatingWordlePlayers` with license `MIT`.

In addition the generated set of items is also in a public repository on the page *Kaggle* under the url: `https://www.kaggle.com/datasets/scarcalvetsis/wordle-games` with a `CC BY-SA 4.0` license.

Capítulo 3

Trabajos relacionados

Debido a la corta vida que Wordle posee actualmente, no existe gran catálogo de estudios sobre la temática. Pese a esto, su similitud con otros juegos de puzzles y su gran popularidad han hecho posible la existencia de varias fuentes que podremos utilizar como base de nuestro proyecto.

3.1. Mastermind

Uno de los puzzles que inspiraron la creación de Wordle es Mastermind, este juego se basa en intentar averiguar un código oculto compuesto de una secuencia de cuatro fichas de seis posibles colores donde el orden importa y se permiten repetir los colores. Este código es propuesto por el "codificador" (uno de los dos jugadores) y la labor del "decodificador" es intentar averiguar este código mediante propuestas que se van refinando gracias a las pistas que el codificador ofrece tras cada intento. Las pistas que se facilitan al "decodificador" se rigen por las siguientes normas.

- Se devuelve una ficha de color negro por cada ficha de la combinación introducida que coincida en posición y color con la combinación oculta.
- Se devuelve una ficha blanca por cada ficha de la combinación introducida cuyo color aparezca en la combinación oculta pero no se encuentre en la posición correcta.
- Si la combinación introducida contiene fichas del mismo color solo se otorgarán tantas fichas como fichas de ese color aparezcan en la combinación oculta. Las fichas sobrantes en la combinación introducida no pueden provocar la entrega de ninguna ficha.

El "decodificador" tiene hasta doce intentos para averiguar la combinación, el juego termina cuando el "codificador" entrega cuatro fichas negras o cuando se alcanza el límite de intentos. Presentamos una partida de muestra en la figura 3.1. Podemos observar claras similitudes entre Wordle y Mastermind, especialmente a nivel de jugabilidad ya que ambos juegos se basan en adivinar una combinación de caracteres ocultos y en cada intento se reciben pistas. Además, el formato de las pistas es

Figura 3.1: Partida completa de Mastermind en un tablero oficial

prácticamente idéntico salvo por el formato de los colores y que la ausencia de ficha en Mastermind se traduce en devolver un cuadrado gris en Wordle. Pese a todo esto, creemos que es importante presentar las diferencias más relevantes entre cada juego:

- En Mastermind solo se indica el número de fichas en la posición correcta, en Wordle se indica además cuales son.
- En Mastermind las códigos continen seis colores distintos y poseen 4 posiciones, en Wordle los códigos contienen veintiséis letras distintas y poseen 5 posiciones.
- En Mastermind todas las combinaciones de colores son validas tanto como código objetivo como código introducido, en Wordle solo se permite utilizar palabras del idioma inglés pertenecientes a una lista fijada por el desarrollador.
- Mastermind permite hasta doce intentos, Wordle permite únicamente seis.

Es claro que las diferencias entre los juegos son menores y, por tanto, estudiar las diversas implementaciones que se han creado para resolver Mastermind nos puede ayudar a modelar a los jugadores de Wordle. Por ello en las siguientes secciones vamos a introducir los métodos más relevantes que se conocen para resolver estos juegos.

3.2. Algoritmos de reducción de la entropía

La reducción de entropía es una técnica esencial en el campo de la inteligencia artificial, y se utiliza en una amplia variedad de aplicaciones, desde el procesamiento del lenguaje natural hasta la visión por computadora y la toma de decisiones. [42]

En general, el objetivo de la reducción de entropía es reducir la incertidumbre en un sistema, lo que a su vez mejora la capacidad de este para tomar decisiones precisas y eficientes. Cuanto mayor sea la entropía, más difícil será predecir los resultados del sistema. Por lo tanto, reducirla es un objetivo clave. [23]. Definimos la entropía de un sistema aletorio X con n posibles resultados mediante la formula 3.1 donde

p_i es la probabilidad de se seleccione la variable i.

$$E(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{3.1}$$

Una de las aplicaciones más importantes de la reducción de entropía es en la toma de decisiones. En este campo, se puede utilizar para evaluar las posibles soluciones y seleccionar la más óptima [49].

Esta técnica se ha empleado para resolver Mastermind y también Wordle, por ello procedemos a presentar cada una de estas soluciones.

3.2.1. Algoritmo de los cinco intentos de Knuth

El algoritmo de los cinco intentos para resolver Mastermind fue presentado por Donand Knuth en 1977. Este se basa en la selección de códigos que reduzcan al máximo la entropía del conjunto de posibles códigos objetivo, es decir, en cada intento, el conjunto de códigos posibles se reduce en base a la respuesta otorgada por el "codificador" y, en la siguiente ronda, el "decodificador" introduce la secuencia que en base a todas las posibles combinaciones de fichas que pueda obtener al introducirla, reduzca más la cardinalidad del conjunto de posibles combinaciones restantes. Knuth demostró que este algoritmo necesitaba a lo sumo cinco intentos para adivinar cualquier posible código objetivo [27].

Este funcionamiento es fácilmente implementable si se codifican los seis colores numericamente del uno al seis y los códigos posibles se expresan como el conjunto de enteros de cuatro cifras que se pueden construir con esos seis dígitos. A continuación detallamos los pasos:

1. Inicializar el conjunto S como el conjunto de los 1296 posibles códigos objetivo, es decir, $S = \{1111, 1112, \ldots, 4565, 4566, \ldots, 6665, 6666\}$.

2. Realizar el primer intento con el código 1122 (Knuth demuestra en su paper que es uno de los que reduce al máximo el conjunto inicial S)

3. Si se obtienen cuatro fichas negras se ha ganado la partida y termina el algoritmo. En caso contrario se continúa al paso 4.

4. Se actualiza el conjunto S eliminando de él cualquier código que no podría haber generado el conjunto de fichas obtenido en el paso 2.

5. Se introduce un nuevo código elegido mediante árboles minimax, este debe ser el código que cumple que su mínimo número de elementos que puede llegar a eliminar del conjunto S es el máximo entre los del resto de códigos. Nótese que el código elegido no tiene por qué pertenecer a S. En caso de que existiera más de un código que cumpliera los requisitos, se elige como condidato el menor de todos ellos.

6. Se continua en el paso 3.

Este algoritmo pese a no generar un jugador perfecto, es capaz de ganar cualquier partida y proporciona una cota para el número de intentos.

3.2.2. Algoritmo para Wordle

A continuación pasamos a explicar cómo usar la reducción de entropía para resolver Wordle [45]:

1. Realizamos un primer intento con una palabra prefijada. Guardamos el número de letras que coinciden con la palabra oculta tanto en posición como en valor (el número de letras "verdes") y el número de letras que coinciden con la palabra oculta en valor pero no en posición (el número de letras "amarillas").

2. Con esta información, eliminamos de la lista de posibles palabras cualquier palabra que no tenga el mismo número de letras coincidentes que la suposición que hicimos. Por ejemplo, si la suposición obtuvo tres letras verdes y una amarilla, eliminamos cualquier palabra de la lista que no tenga exactamente tres de las mismas letras en la misma posición y una de la misma letra en una posición diferente.

3. Eliminamos de la lista de posibles palabras cualquiera que contenga letras que no coincidan con las posiciones de las letras verdes y amarillas que obtuvimos en el paso 2.

4. Para la nueva suposición elegimos la palabra que tenga más probabilidades de eliminar la mayor cantidad de soluciones posibles. Para ellos se selecciona la palabra que reduzca en una mayor cantidad la entropía de la lista. Esto suele requerir un equilibrio entre elegir una palabra que probablemente coincida con muchas letras de la palabra oculta y una que probablemente no coincida con muchas letras de la palabra oculta.

5. Se repiten los pasos del 2 al 4 hasta que se haya adivinado con éxito la palabra oculta o se exceda el número de intentos.

Cabe destacar que este procedimiento asume que todas las palabras tienen la misma probabilidad de ser elegidas como palabra del día, no obstante, la lista de palabras que realmente pueden ser solución es mucho más reducida. Además, este algoritmo usa todas las palabras aceptadas por el juego, lo cual no es realista si pretendemos imitar a un jugador humano. Para mitigar estos dos problemas existe una variable de este procedimiento que asigna una probabilidad a cada palabra posible basada en su frecuencia de aparición en el lenguaje inglés y en si forma parte del conjunto de palabras posibles. Esta probabilidad junto a la reducción de entropía constituyen una heurística que se utiliza en una versión modificada del algoritmo similar a una búsqueda A* que permite refinar el comportamiento del programa para ajustarse más al comportamiento humano.

3.3. Algoritmos genéticos

Otra de las varias alternativas para resolver el problema han sido los algoritmos genéticos. Un algoritmo genético es una técnica heurística de búsqueda y optimización inspirada en el proceso de selección natural y la genética. Se utiliza para encontrar soluciones aproximadas a problemas de optimización y búsqueda. El algoritmo imita el proceso de evolución al modificar iterativamente una población de posibles soluciones al problema en cuestión [22, 19].

A continuación presentamos el esquema general de este tipo de algoritmos [34, 13]:

1. **Inicialización**: Se genera de manera aleatoria una población de posibles soluciones (también llamadas individuos o cromosomas). Cada individuo representa una posible solución al problema.

2. **Evaluación de aptitud o idoneidad**: Se evalúa cada individuo de la población y se le asigna un valor de aptitud en función de cuán bien resuelve el problema. El valor de aptitud se determina mediante una función objetivo o una función de *fitness* que cuantifica la calidad o idoneidad del individuo.

3. **Selección**: Los individuos con valores de aptitud más altos tienen una mayor probabilidad de ser seleccionados para la reproducción. El proceso de selección se basa típicamente en un mecanismo probabilístico, como la selección de la ruleta o la selección por torneo.

4. **Reproducción**: Los individuos seleccionados se utilizan para crear descendencia para la siguiente generación. Esto se logra a través de operadores genéticos como el cruce y la mutación. El cruce implica combinar información genética de dos o más padres para crear nueva descendencia. La mutación introduce pequeños cambios aleatorios en el material genético de un individuo.

5. **Reemplazo**: La nueva descendencia, junto con una parte de la población existente, reemplaza a la población antigua. Esto asegura la diversidad de la población y permite explorar soluciones potencialmente mejores en generaciones posteriores.

6. **Iteración**: Los pasos 2-5 se repiten durante un cierto número de iteraciones o hasta que se cumpla una condición de terminación. Esto puede ser un umbral de aptitud específico, un número máximo de iteraciones o un límite de tiempo predefinido.

7. **Terminación**: El algoritmo finaliza cuando se cumple la condición de terminación. El mejor individuo de la población final se considera la solución o solución aproximada al problema.

3.3.1. Algoritmo genético para Mastermind

Los algoritmos genéticos han ofrecido una visión nueva al problema de resolver Mastermind y esto ha llevado a la creación de modelos que tratan minimizar el número intentos por debajo del límite establecido por Knuth. A continuación se presentan los diversos parámetros empleados por Berghman, Goossens y Leus para generar un decodificador con una media de intentos inferior al algoritmo de Knuth [4], la estructura de este se define en el pseudocódigo 1 donde el conjunto de valores posibles en cada intento se representa como $\hat{E}_i$ y donde se añaden dos fases adicionales al proceso de mutación: la permutación y la inversión.

Algoritmo 1: Algoritmo genético para Mastermind

```
i ← 1;
Fijar y evaluar selección inicial para g_1;
Obtener respuesta X_1 e Y_1;
while X_i ≠ P do
    i ← i + 1;
    Ê_i ← {};
    h ← 1;
    Inicializar población;
    while h ≤ maxgen AND |Ê_i| ≤ maxsize do
        Generar nueva población mediante el cruce de individuos;
        Aplicar las funciones de mutación, permutación e inversión a los individuos;
        Calcular el valor de fitness;
        Añadir combinaciones posibles a Ê_i;
        h ← h + 1;
    end while
    Evaluar g_i ∈ Ê_i;
    Obtener respuesta X_i e Y_i;
end while
```

- **Representación cromosómica**: Los individuos de cada generación se representan como una lista de listas de cuatro números naturales del uno al seis que representan cada uno de los colores del código, cada lista del individuo se corresponde con un intento. La primera lista de cada individuo se inicializa con una combinación aleatoria.

- **Número de individuos por generación**: Cada generación posee ciento cincuenta individuos.

- **Función de fitness**: Para calcular el valor de fitness del código c del último intento de un individuo se deben considerar los valores de fitness de sus intentos anteriores g_i y determinar el número de fichas negras $X_i'(c)$ y fichas blancas $Y_i'(c)$ que se hubieran obtenido si el código secreto hubiera sido g_i con los valores realmente obtenidos $X_i(C)$ y $Y_i(c)$. A priori podría parecer que se debería dar un mayor peso a las fichas negras pues el objetivo del juego es

obtener cuatro de ellas, no obstante como solo se comprueban combinaciones válidas (según las fichas obtenidas en los intentos anteriores) las fichas de colores fuera de lugar aportan información relevante en los conjuntos reducidos. En pos de generalizar la función al máximo, se incluye un factor $a > 0$ que multiplica a las diferencias de fichas negras. Para terminar, se añade un término P proporcional al número de intentos y un peso b que determina la importancia relativa de este término con respecto a las diferencias de las fichas obtenidas. Resultantemente, la función f en función de la combinación c y el número de intento i se puede representar mediante la fórmula 3.2.

$$f(c;i) = a\left(\sum_{q=1}^{i}\left|X'_q(c) - X_q(c)\right|\right) + \sum_{q=1}^{i}\left|Y'_q(c) - Y_q(c)\right| + bP(i-1) \qquad (3.2)$$

- **Cruce**: Tras la generación del valor de fitness se realiza un cruce de uno o dos puntos (cada opción con una probabilidad de 0,5) sobre los individuos, la probabilidad de selección de un individuo es proporcional a su valor de fitness.
- **Mutación**: Con una probabilidad del 0,03, después de terminar el cruce se realiza una mutación que modifica aleatoriamente el color de una posición.
- **Permutación**: Con una probabilidad del 0,03, después de intentar realizar una mutación se pueden permutar la posición de dos colores de la representación del individuo.
- **Inversión**: Con una probabilidad de 0,02, después de intentar realizar una inversión se pueden seleccionar aleatoriamente dos posiciones y la secuencia comprendida entre estas se invierte.
- **Eliminación de inviduos**: Si tras el proceso de cruce, mutación, permutación e inversión el código resultante ya figura en la nueva población, este es descartado.

3.4. Modelado de jugadores

Las aplicaciones de la IA que más nos interesan en este proyecto son las de imitación de jugadores. La capacidad de simular y replicar el comportamiento humano no es fácil, y en la mayoría de casos resulta incluso más complicado que tratar de resolver óptimamente el juego [38]. A pesar de esto, existen múltiples aproximaciones a este problema que detallamos a continuación:

3.4.1. DQN

Las *Deep Q-learning networks* son un tipo de algoritmo que presenta un enfoque innovador que combina redes neuronales profundas y aprendizaje por refuerzo para lograr un comportamiento igual o superior al humano en un entorno complejo determinado [35].

La idea principal detrás del algoritmo es el aprendizaje directo a partir de una entrada sensorial *en crudo*, como imágenes o vídeos, sin ningún conocimiento previo y que los agentes tomen decisiones y realicen acciones en entornos complejos. Para aproximar la función Q, que representa las recompensas futuras esperadas para cada acción en un estado dado se utiliza una red neuronal profunda. Al estimar la función Q, un agente puede determinar la mejor acción a tomar en un estado particular para maximizar su recompensa acumulada a lo largo del tiempo.

A continuación describimos los mecanismos y elementos empleados por el algoritmo:

- **Experience replay**: El agente almacena sus experiencias (que consisten en estado, acción, recompensa y siguiente estado) en un búfer de repetición. Luego, este búfer se muestrea aleatoriamente durante el proceso de aprendizaje para romper las correlaciones temporales y mejorar la eficiencia del aprendizaje.
- **Red neuronal profunda**: DQN utiliza una red neuronal profunda, normalmente una red neuronal convolucional (CNN), como un aproximador de funciones. La red toma el estado actual como entrada y produce un valor-Q para cada posible acción. La red se entrena para minimizar la diferencia entre los valores-Q predichos y los valores-Q objetivo.
- **Red objetivo**: Para estabilizar el proceso de aprendizaje, DQN introduce una red objetivo separada, que es una copia de la red principal. La red objetivo se actualiza periódicamente con los pesos de la red principal pero no con la misma frecuencia, proporcionando un objetivo consistente en el tiempo para la estimación de los valores-Q manteniendo la atención del modelo en los objetivos principales.
- **Exploración vs. Explotación**: DQN equilibra la exploración (probar nuevas acciones) y la explotación (utilizar el conocimiento actual para maximizar las recompensas). Esto se logra típicamente mediante una política ϵ-greedy, donde el agente elige una acción aleatoria con una cierta probabilidad ϵ y selecciona la acción con el valor-Q predicho más alto en caso contrario.

Al mejorar iterativamente la estimación de la función Q a través del proceso de entrenamiento, DQN permite que el agente imite el comportamiento de su entrada sensorial y también es capaz de desarrollar estrategias óptimas o casi óptimas para un entorno dado. DQN ha logrado éxitos destacados, como dominar varios juegos de Atari 2600 y lograr un rendimiento humano en dominios complejos [35, 36]. Si bien en un comienzo se ha utilizado para crear jugadores óptimos, la imitación de jugadores aparece cada vez más en este tipo de algoritmos, ya sea como el objetivo final del trabajo [52] o como utilidad para acelerar el aprendizaje en las primeras etapas de un modelo de jugador óptimo [39].

3.4.2. Aprendizaje por refuerzo profundo a partir de preferencias humanas

El aprendizaje por refuerzo profundo a partir de las preferencias humanas es proceso que permite a un agente aprender a partir de la retroalimentación o preferencias

humanas en lugar de las señales de recompensa tradicionales en el aprendizaje por refuerzo [10].

En muchos escenarios del mundo real, puede resultar desafiante o consumir mucho tiempo diseñar una función de recompensa que capture de manera precisa el comportamiento deseado. En este método, se utilizan evaluadores humanos que proporcionan comparaciones binarias o clasificaciones de diferentes trayectorias o acciones, indicando sus preferencias al agente que se desea entrenar.

A continuación presentamos los mecanismos y componentes principales de estos algoritmos:

- **Retroalimentación basada en comparaciones**: En lugar de proporcionar recompensas explícitas, los evaluadores humanos brindan preferencias entre pares de trayectorias o acciones. Por ejemplo, para comparar dos estrategias de conducción diferentes o elegir la mejor acción en una situación determinada. Esta retroalimentación basada en comparaciones se utiliza para guiar el proceso de aprendizaje.

- **Agregación de la retroalimentación humana**: La retroalimentación humana se debe adaptar como una única señal de recompensa o política. Un enfoque posible consiste en entrenar un modelo de recompensa utilizando aprendizaje supervisado para predecir los juicios de preferencia de los evaluadores humanos. Este modelo de recompensa se utiliza junto con el aprendizaje por refuerzo para optimizar el comportamiento del agente.

- **Aprendizaje de recompensa adversarial**: Se utiliza un enfoque adversarial en el que se entrena un modelo de recompensa que compite contra un segundo modelo llamado política de comportamiento. La política de comportamiento tiene como objetivo generar trayectorias que sean difíciles de clasificar correctamente por el modelo de recompensa. Mediante actualizaciones iterativas de ambos modelos, el agente puede aprender una política optimizada basada en las preferencias humanas.

- **Evaluación del rendimiento**: Para evaluar el rendimiento de la política aprendida se debe realizar una comparación con la política de comportamiento y, además, utilizar retroalimentación adicional de los evaluadores humanos.

Este enfoque tiene aplicaciones en diversos campos donde resulta desafiante definir explícitamente una función de recompensa, como la conducción autónoma, la robótica y los sistemas de recomendación personalizados. Al involucrar a los evaluadores humanos en el proceso de aprendizaje, se ayuda a cerrar la brecha entre los algoritmos de aprendizaje automático y las preferencias humanas, lo que conduce a comportamientos más intuitivos, deseables y humanos.

3.4.3. Aprendizaje por imitación a través de la estimación de políticas de apoyo expertas

El aprendizaje por imitación a través de la estimación de políticas de apoyo expertas es un enfoque novedoso para el aprendizaje por imitación, una rama del aprendizaje automático en la que un agente aprende a imitar el comportamiento de un experto observando sus demostraciones [47].

En el aprendizaje por imitación tradicional, el agente aprende directamente de las demostraciones del experto minimizando la discrepancia entre sus propias acciones y las acciones del experto. Sin embargo, este enfoque tiene limitaciones, como el problema del cambio covariante, donde la distribución de estados encontrados por el agente durante el entrenamiento puede diferir de las demostraciones del experto. Esto puede llevar a un rendimiento subóptimo.

Para abordar este problema, se propone un método que estima la política del experto, que representa la probabilidad de que el experto tome acciones específicas en diferentes estados. Al estimar esta política, el agente puede adaptarse mejor a diferentes estados e imitar de manera más precisa el comportamiento del experto.

Detallamos los mecanismos y componentes más importantes de este método:

- **Estimación del soporte de la política del experto**: El método se centra en estimar la política del experto, que representa la distribución de probabilidad de las acciones en función de los diferentes estados. Esta estimación se realiza utilizando un modelo de recompensa, que predice la recompensa acumulada esperada al tomar una acción específica en un estado dado, y un modelo de razón de densidad, que estima la densidad relativa de las acciones del experto en comparación con las acciones del agente.
- **Clonación de comportamiento con soporte**: Se combina la clonación de comportamiento, una técnica en la que el agente imita directamente las acciones del experto, con el soporte de la política estimado. La clonación de comportamiento se utiliza para entrenar la política del agente para imitar al experto. Al incorporar el soporte de la política estimado, el agente puede tener en cuenta las diferencias en el comportamiento entre el experto y el agente, lo que mejora el rendimiento general de la imitación.
- **Aprendizaje de imitación adversarial**: Se utiliza una variante adversarial que implica entrenar una red discriminatoria que tiene como objetivo distinguir entre las acciones del experto y las acciones del agente. La política se entrena para maximizar el error del discriminador, engañándolo efectivamente e imitando el comportamiento del experto.

Este método propuesto supera a los enfoques tradicionales de clonación de comportamiento, especialmente en escenarios con cambio covariante. El soporte de la política estimado ayuda al agente a adaptarse a diferentes estados, lo que resulta en una mejora en el rendimiento de la imitación.

Capítulo 4

Recopilación y limpieza de las partidas

Como primer paso de este proyecto, tal y como se detalló en la planificación, se procede a recopilar partidas publicadas en Twitter. La recopilación de tweets puede ser una excelente manera de obtener datos para un proyecto de ciencia de datos, ya que permite a los investigadores acceder a una gran cantidad de contenido generado por los usuarios. La API de Twitter brinda a los desarrolladores acceso y permisos especiales a la plataforma, incluidos tweets, perfiles de usuario y métricas de participación. Sin embargo, recopilar tweets para un proyecto de ciencia de datos requiere una cuidadosa consideración de cuestiones éticas y legales, como la privacidad del usuario, la propiedad de los datos y los términos de servicio de Twitter. Además, la calidad y la relevancia de los datos recopilados deben evaluarse cuidadosamente para garantizar que se alineen con los objetivos y la metodología de la investigación. Por ello consideramos relevante explicitar el proceso de obtención así como la manipulación realizada posteriormente y así facilitar la comprensión del conjunto con el que se trabajará en los capítulos posteriores.

4.1. Recopilación de los tweets

Para poder realizar una recopilación automatizada de los tweets, en primer lugar se deben fijar los prerequisitos que los estos debían cumplir. En este proyecto nos centraremos únicamente en las partidas en inglés publicadas entre los días 1 de enero y el 15 de noviembre de 2022. La imagen 4.1 muestra el formato de los tweets válidos, como podemos comprobar, estos degen contener la tabla con los cuadrados correspondientes al *feedback* obtenido, en número de intentos y el identificador de partida.

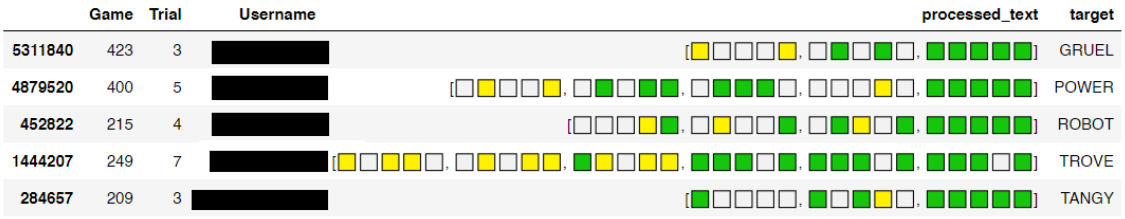
Para concretar el criterio para recolectar un tweet utilizamos el formato definido por Wordle para compartir una partida que especificamos a continuación: la primera frase sigue la siguiente expresión regular: `Wordle [0-9]{3} [1-6,X]/6` donde el primer bloque de numeros después de la palabra Wordle hace referencia al número de juego (que aumenta en uno cada día) y el segundo bloque hace referencia al número de intentos en los que se ha finalizado la partida (en caso de perder la partida se coloca una equis mayúscula para en lugar de un dígito). Esta primera expresión es

Figura 4.1: Ejemplo de tweets recopilados

precedida por tantas líneas como intentos realizados que contienen cinco cuadrados de tres colores distintos: verde, amarillo o negro. Al depender del número de intentos, no podemos generar una expresión regular para ellas interpretable por el buscador de Twitter [44] y por lo tanto recopilaremos los tweets que contengan solamente la primera expresión regular.

El proceso de automatización se ha realizado mediante la librería `snscrape` de Python que implementa el módulo `snscrape.modules.twitter` que permite realizar búsquedas y recopilaciones de tweets de manera relativamente simple. Estas búsquedas se han configurado de manera que de cada día en nuestra franja temporal de exploración extraiga hasta cinco mil tweets de cada uno de los intentos, es decir, que de cada día extraiga a lo sumo treinta y cinco mil tweets. Para acelerar este proceso de recopilación, se ha paralelizado el trabajo mediante la librería `multiprocessing` y también se han dividido los datos en paquetes correspondientes a días. De cada tweet se guarda la siguiente información:

- **Fecha y hora** de publicación original.
- **TweetId**, un ID propia de la plataforma que es único para cada tweet publicado por un usuario.
- **Texto del tweet**, que debe contener la expresión regular definida anteriormente.
- **Nombre de usuario**: indica el nombre del autor.
- **URL** con el enlace al tweet público, es posible que algunos ya no estén disponibles ya que en cualquier momento una cuenta puede decidir borrar u ocultar sus tweets.



	Game	Trial	Username	processed_text	target
5311840	423	3	[redacted]	[🟨⬜⬜⬜🟨, ⬜🟩⬜🟩⬜, 🟩🟩🟩🟩🟩]	GRUEL
4879520	400	5	[redacted]	[⬜🟨⬜⬜🟨, ⬜🟩⬜🟩🟩, ⬜🟩🟩🟩⬜, ⬜⬜⬜🟨⬜, 🟩🟩🟩🟩🟩]	POWER
452822	215	4	[redacted]	[⬜⬜⬜🟨🟩, ⬜🟨⬜⬜🟩, ⬜🟩🟨⬜🟩, 🟩🟩🟩🟩🟩]	ROBOT
1444207	249	7	[redacted]	[🟨⬜🟨🟨⬜, ⬜🟨⬜🟨🟨, 🟩🟨⬜🟨🟨, 🟩🟩🟩⬜🟩, 🟩🟩🟩⬜🟩, 🟩🟩🟩⬜🟩]	TROVE
284657	209	3	[redacted]	[🟩⬜⬜⬜⬜, 🟩⬜🟩🟨⬜, 🟩🟩🟩🟩🟩]	TANGY

Figura 4.2: Formato definitivo de los datos con el nombre de usuario oculto

4.1.1. Extracción de las variables

Tras obtener las variables en crudo hemos procedido a procesarlas para extraer la información necesaria de cada una. Salvo por el texto del tweet, el resto de variables proporcionadas por `snscrape` no requieren ningún tratamiento previo. Para extraer los datos de las partidas hemos utilizado la librería `re` de Python que nos permite analizar y manipular strings mediante expresiones regulares, en nuestro caso hemos utilizado una expersión que encontraba todas las cadenas de cinco cuadrados de colores dentro de un tweet y hemos colocado cada aparición en un *array* que es el valor que hemos utilizado en tabla de datos. Además hemos guardado el texto original por redundancia, una vez extraídos los intentos hemos generado una variable extra que contiene el número de intentos de la partida, que se corresponde con la longitud del *array* obtenido. En la figura 4.2 podemos ver el formato definitivo.

4.2. Limpieza de los datos

Realizar la limpieza de datos es un paso crucial en cualquier proyecto de análisis de datos, incluido el análisis de tweets sobre juegos de Wordle. Twitter es una plataforma con un control de contenido relativamente pequeño, lo que puede resultar en datos ruidosos, irrelevantes o engañosos. La limpieza de datos es el proceso de identificar y corregir o eliminar errores, inconsistencias e imprecisiones en los datos para mejorar su calidad y confiabilidad. En el contexto del análisis de tweets sobre juegos de Wordle, la limpieza de datos puede ayudar a eliminar *spam*, *bots*, tweets irrelevantes y partidas engañosas. Al realizar esta limpieza nos aseguraremos de que los datos que utilizamos para el análisis son precisos, consistentes y representativos, lo que puede generar conocimientos más confiables y procesables.

4.2.1. Eliminación de partidas con formato incorrecto

A partir de las variables generadas en el apartado anterior, hemos procedido a eliminar las partidas que presentaban las siguientes discrepancias:

- Se han eliminado todas las entradas cuyo número de intentos no coincidía con el número de cadenas encontradas,
- Se han eliminado aquellas partidas tales que supuestamente se ha ganado la partida pero no terminan con cinco cuadrados verdes.

Número de tweets inicial	7.102.548	**Partidas perdidas con discrepancias en valor final**	998
Número de partidas con discrepancias entre los intentos	147.085	**Partidas con una victoria en una posición inválida**	116
Partidas ganadas con discrepancias en el valor final	278	**Tweets restantes tras la limpieza preliminar**	6.954.106

Tabla 4.1: Número de partidas eliminadas por el proceso de limpieza preliminar

- Se han eliminado las partidas que supuestamente se habían perdido pero sí terminaban con cinco cuadrados verdes.
- Se han eliminado las partidas que contenían cinco cuadros verdes en una posición distinta a la final, ya que una vez se ha ganado no se puede seguir jugando.

Finalmente, se han convertido todos los cuadrados de color negro en cuadrados de color blanco para estandarizar la entrada. La presencia de la diferencia de colores es debida a la posibilidad de usar el modo noche al jugar, este hace modo únicamente modifica el color de los cuadrados sin completar en el texto a compartir. En la tabla 4.1 analizamos el número de entradas que se han eliminado en cada parte del proceso de limpieza.

4.2.2. Eliminación de partidas falseadas

Con el trabajo de la sección 4.2.1 hemos podido eliminar de nuestro conjunto de datos aquellos tweets que claramente no se ajustaban al formato de las partidas de Wordle usuales, no obstante, esta metodología no permite encontrar aquellas partidas falseadas, es decir, aquellas que contienen códigos de colores imposibles de generar con la palabra objetivo del día. Este tipo de partidas no son representativas de los datos que queremos analizar y, por ello, una limpieza más minuciosa es pertinente.

Para poder realizar esto hemos generado un algoritmo que imita el funcionamiento del juego y que además es capaz de generar todas las posibles combinaciones de cuadrados de colores que se pueden obtener a partir de un conjunto de palabras válidas y una palabra objetivo.

Seguidamente se ha generado un conjunto que contenga todas las palabras que el juego admite como entrada a partir del documento que se encuentra en [43]. También se ha generado una tabla con la correspondencia entre el id de partida y la palabra objetivo correspondiente a este, esta información se ha obtenido a partir de la siguiente fuente [1]. Cabe destacar que desde la adquisición del juego por parte de periódico *The New York Times* no existen registros oficiales de la palabra del día ni tampoco de las palabras de entrada válidas, por tanto la fiabilidad de estos conjuntos, especialmente del segundo puede ser cuestionada, pese a esto hemos utilizado como fuente páginas web populares y que consideramos fiables.

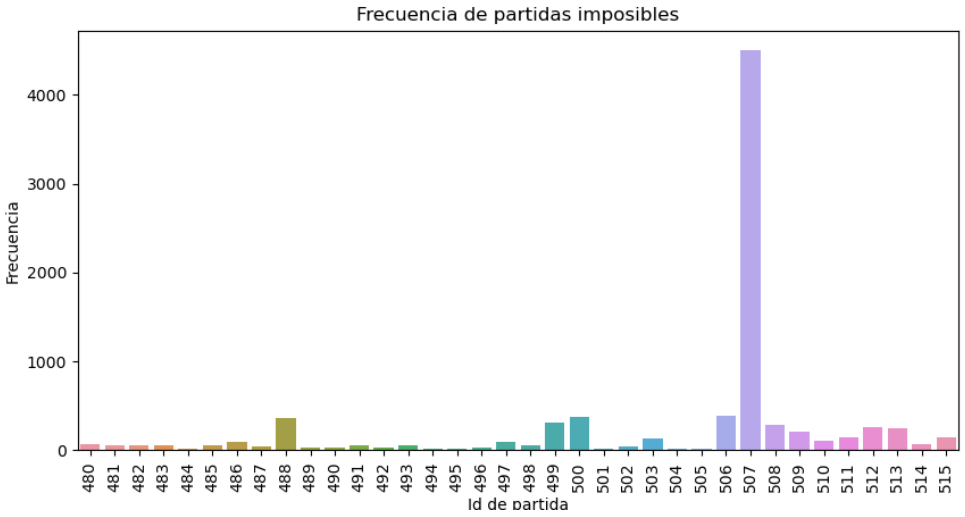


Figura 4.3: Número de partidas imposibles encontradas entre los id 480 y 515

Pese al uso de páginas fiables hemos preferido analizar el número de partidas falseadas para cada día para tratar de encontrar algún en nuestros conjuntos de palabras objetivo. Para ello se ha pasado a generar el conjunto de todas las posibles respuestas para cada palabra objetivo según el id de las partidas presentes en nuestro conjunto de datos. Finalmente, para cada tweet se ha comprobado si todas las respuestas de su partida correspondiente pertenecían al conjunto de respuestas posibles para esa palabra del día. Una vez se han identificado las partidas problemáticas se ha procedido a analizar la distribución de estas.

Al hacerlo encontramos un único un valor atípico muy destacado en el id 507 ya que en este se encuentran aproximadamente el 25 % de todas las partidas erróneas. El gráfico resultante es demasiado amplio como para presentarlo y, por ello, en la figura 4.3 solo mostramos la sección donde se encuentra el valor atípico.

Al indagar en esta partida podemos observar de que la palabra objetivo es *snarl*, que parece ser complicada de adivinar en vista de su alto contenido de consonantes, no obstante, esto no nos parece sufiente para explicar este notable aumento en el número de partidas falseadas. Al seguir investigando acerca sobre esta problemática nos percatamos de que estas partidas datan del 8 de noviembre de 2022. Al buscar noticias relacionadas con Wordle sobre estas fechas encontramos que el día anterior se habían anunciado cambios en el conjunto de palabras sujetas a ser palabra del día por parte de la nueva compañía propietaria del juego [28]. Esto nos motivó a evaluar la posibilidad de que la palabra que figuraba como palabra del día en nuestra lista no fuera la correcta. Tras consultar varias páginas que recolectaban las partidas diarias y encontrar un gran número de discrepancias con la palabra objetivo con id 507 decidimos que era muy probable que la que habíamos tomado no fuera la que verdaderamente se eligió ese día.

Tras probar varias combinaciones, decidimos usar la palabra *spell* que figuraba en

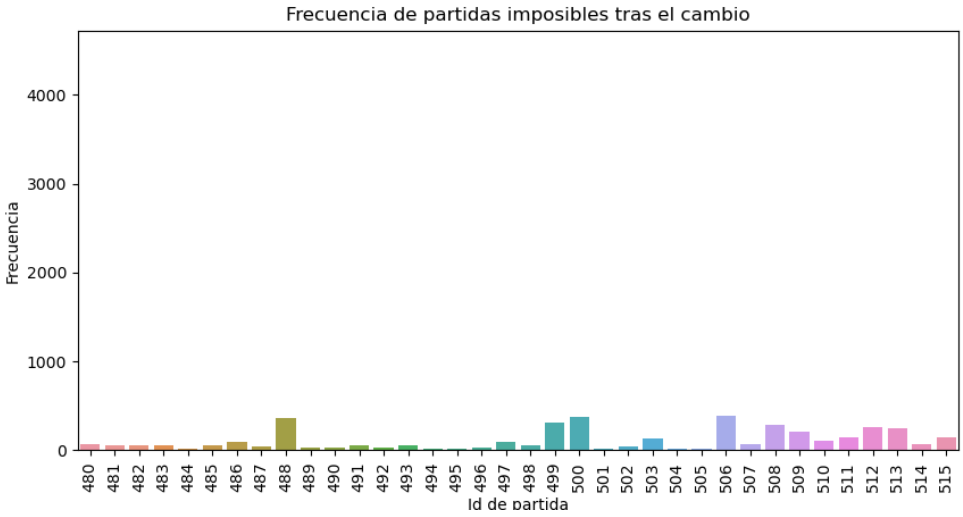


Figura 4.4: Número de partidas imposibles tras realizar la corrección

Número de tweets al empezar la segunda limpieza	6.954.106
Número de partidas imposibles utilizando la lista lista de palabras objetivo original	35.415
Número de partidas imposibles utilizando la lista de palabras objetivo modificada	30.978
Número de partidas duplicadas	46.082
Número de tweets final	6.877.046

Tabla 4.2: Número de partidas eliminadas por el proceso de limpieza secundaria

varias publicaciones. Consecuentemente realizamos el procedimiento anterior implementando este pequeño cambio y observamos una notable reducción de las partidas imposibles que puede apreciarse en el gráfico 4.4 donde el número se ha reducido drásticamente. Gracias a este análisis concluimos que la explicación más probable del valor atípico fue debida al error en la palabra clave del juego con id 507 y, por lo tanto, trabajaremos suponiendo que la palabra correcta era *spell*.

Finalmente eliminamos todas las entradas duplicadas, estas corresponden a las entradas que posean el mismo identificador de partida y el mismo identificador de usuario, estas entradas corresponden a jugadores que han compartido su partida más de una vez o que han publicado varias partidas, nos quedamos con la partida publicada más antigua.

Para concluir esta sección analizamos el número de entradas que se han eliminado con cada procedimiento y el número de tweets del el que finalmente dispondremos para realizar el análisis, esto lo podemos apreciar en la tabla 4.2.

4.3. Creación del conjunto de datos público

Como primer resultado de este trabajo, hemos publicado el conjunto de datos final y lo hemos enviado a un repositorio público para que lo añadan a su repertorio y conseguir así conseguir un mayor alcance. Para ello hemos anonimizado los datos obtenidos, en particular, modificaremos los datos asociados a los nombres de usuario asignándoles a cada uno un número aleatorio entre uno y el número total de jugadores diferentes presentes en nuestros datos. Así, hemos creado el conjunto final y lo hemos subido a la plataforma *Kaggle* donde se puede acceder de manera pública y con una licencia `CC BY-SA 4.0`. Se puede acceder mediante el siguiente `https://www.kaggle.com/datasets/scarcalvetsis/wordle-games`.

Capítulo 5

Análisis de las partidas

El siguiente paso en nuestro proyecto es el análisis de los datos que hemos recolectado, este puede ayudar a comprender con qué frecuencia aparecen ciertas letras en el juego y cómo están posicionadas dentro de la palabra. Esta información se puede utilizar para desarrollar estrategias y mejorar las posibilidades de ganar el juego. Además, analizar un conjunto de datos de partidas de Wordle puede ayudar a identificar las palabras más comúnmente utilizadas. Esta información puede ser útil para los jugadores que buscan expandir su vocabulario y mejorar sus habilidades ya que pueden utilizar estos datos para aprender nuevas palabras e incrementar sus posibilidades de adivinar la respuesta correcta, así como identificar errores comunes que cometen y mejorar sus habilidades en el futuro. Nuestros objetivos son describir los datos que poseemos, identificar patrones y tendencias y presentar nuestros hallazgos de manera visual.

5.1. Distribución de las variables de juego

En primer lugar procederemos a analizar la distribución de las distintas variables que caracterizan las partidas de Wordle y utilizarlas para evaluar el rendimiento de los jugadores.

5.1.1. Número de intentos

La medida más habitual en la evaluación de la calidad de un juego es la longitud de este, es decir, el número de intentos que ha necesitado el jugador para adivinar la palabra secreta. En la tabla 5.1 podemos ver la cantidad de partidas presentes en el conjunto de datos según el número de intentos y también la cantidad de partidas que resultaron en una derrota. En la tabla 5.2 podemos ver los distintos estimadores de la distribución tomando las derrotas como un séptimo intento. Vemos que la media se encuentra en $4,19$ y la desviación típica es cercana al $1,5$ por tanto estimamos que la mayoría de partidas requieren entre tres y cinco intentos. Finalmente, podemos observar la distribución en el gráfico 5.1.

Nº de intentos	Cantidad
1	139.077
2	789.447
3	1.459.465
4	1.552.646
5	1.501.858
6	1.106.965
Derrotas	373.670
Total	6.923.128

Tabla 5.1: Número de partidas según el número de intentos

Media	4,19
Desviación típica	1,46
25 %	3
50 %	4
75 %	5

Tabla 5.2: Estimadores de la longitud de las partidas

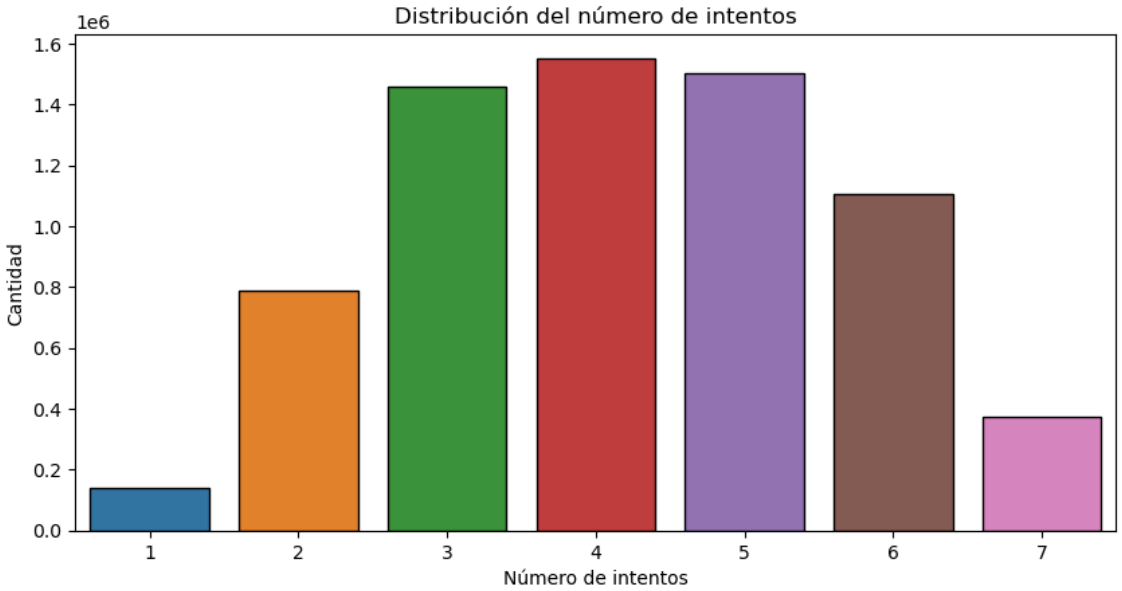


Figura 5.1: Distribución del número de intentos

5.1.2. Tasa de aciertos por intento

Pasamos ahora a analizar la distribución de las partidas, comenzamos analizando el número de aciertos (número de cuadrados verdes) obtenidos en cada uno de los intentos. Esta información nos puede ayudar a distinguir a los tipos de jugadores más adelante cuando intentemos encontrar patrones en las partidas. En las figuras 5.2 y 5.3 podemos observar como tanto la cantidad como la proporción de aciertos aumenta a medida que aumenta el número de intentos, lo cuál concuerda con la suposición de que a medida que se van probando palabras y obteniendo pistas, los jugadores mejoran su estrategia. Destacamos la falta de una clara tendencia dominante en el número de aciertos en los intentos 3 y 4 lo cuál podría ser indicativo de la variedad de jugadores pues la media de intentos es ligeramente superior a 4, es decir, los intentos 3 y 4 presentan las distribuciones más equilibradas y posiblemente sean los más decisivos a la hora de determinar la longitud de una partida.

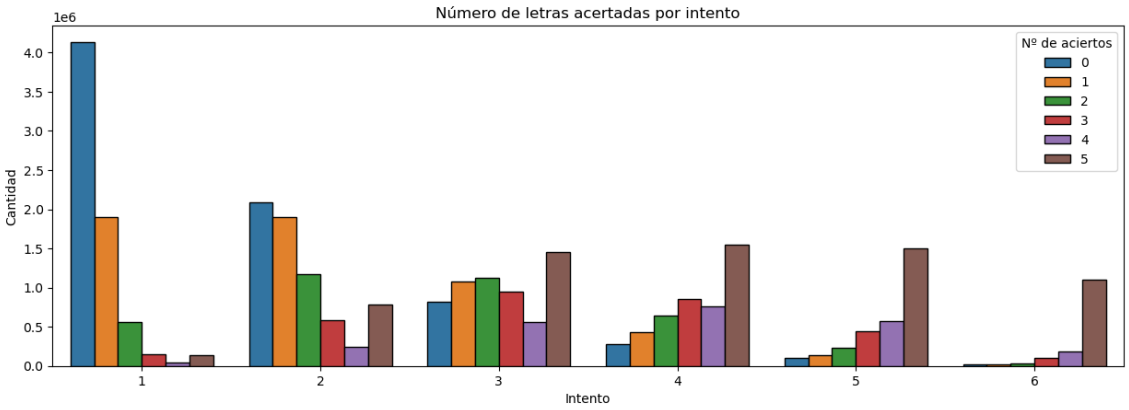


Figura 5.2: Distribución del número de aciertos por intento

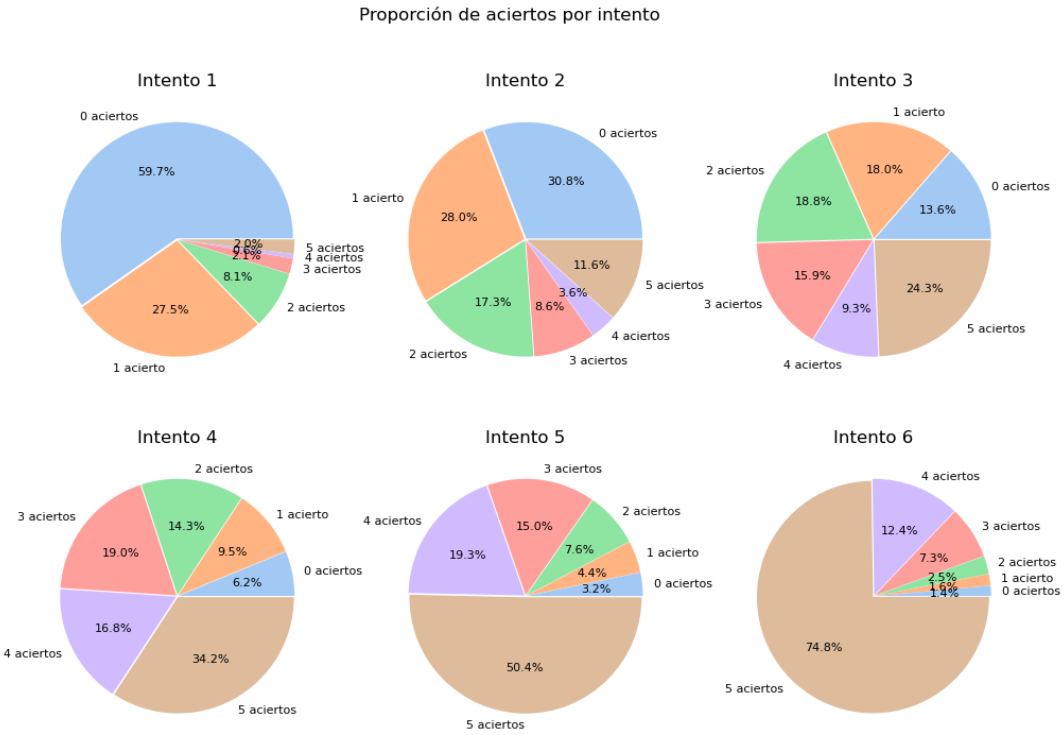


Figura 5.3: Proporción de aciertos por intento

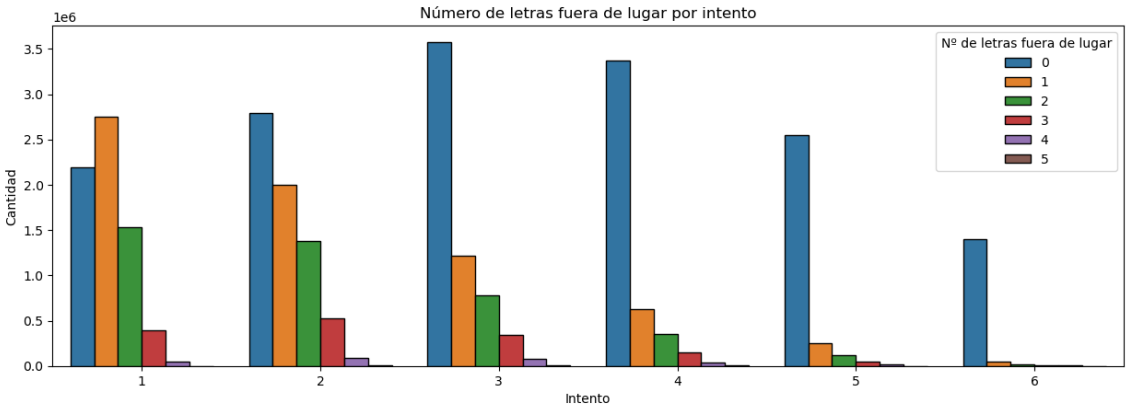


Figura 5.4: Distribución del número de letras fuera de lugar por intento

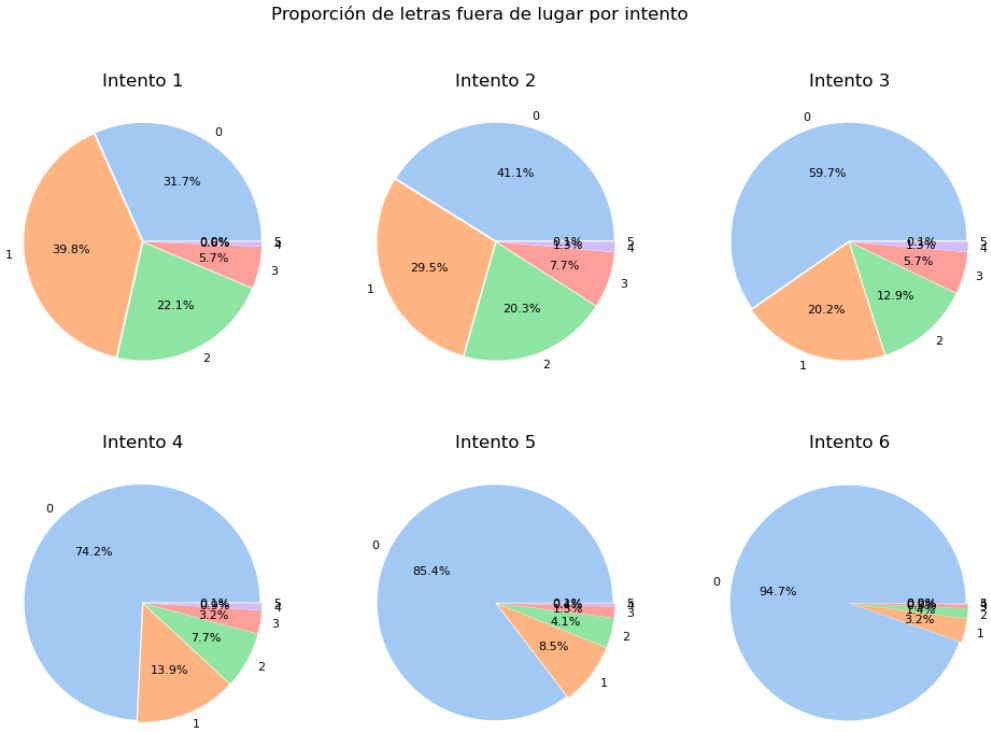


Figura 5.5: Proporción de letras fuera de lugar por intento

5.1.3. Tasa de letras en una posición equivocada

De manera similar a la que hemos procedido en la sección 5.1 procedemos a analizar la distribución del número de letras fuera de lugar (cuadrados amarillos). En los gráficos 5.4 y 5.5 podemos ver que el la distribución tiende a ser dominada por la cantidad cero a medida que aumenta el número de intentos lo cuál concuerda con el la noción de que a una vez se conocen todas o la mayoría de las letras de la palabra, los jugadores tratan de realizar combinaciones con estas hasta ganar, lo cual explica la disminución tan acelerada que se da, especialmente a partir del

intento número 3. Cabe descatacar también que, a diferencia de la distribución de los aciertos, en este caso todas las distribuciones tienen una tendencia dominante clara y las mayores distinciones se dan en los intentos 1 y 2 y estos presentan claras similitudes entre ellos.

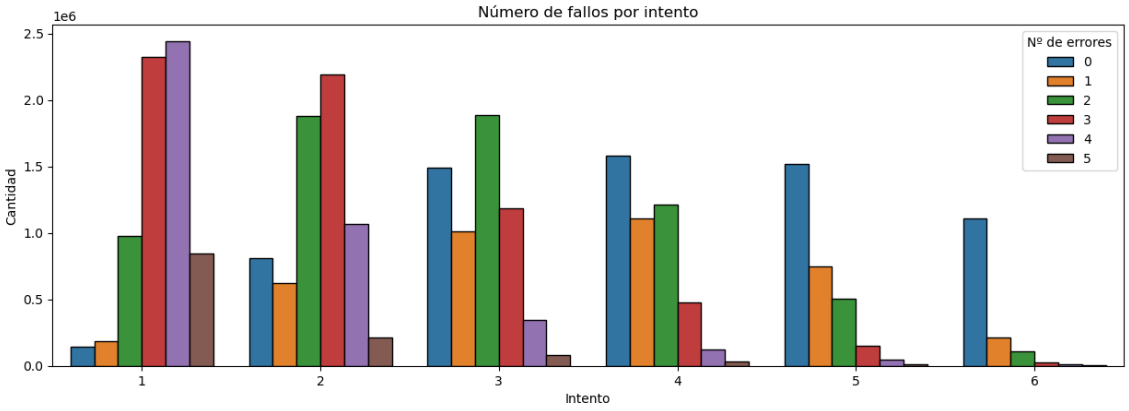


Figura 5.6: Distribución del número de fallos por intento

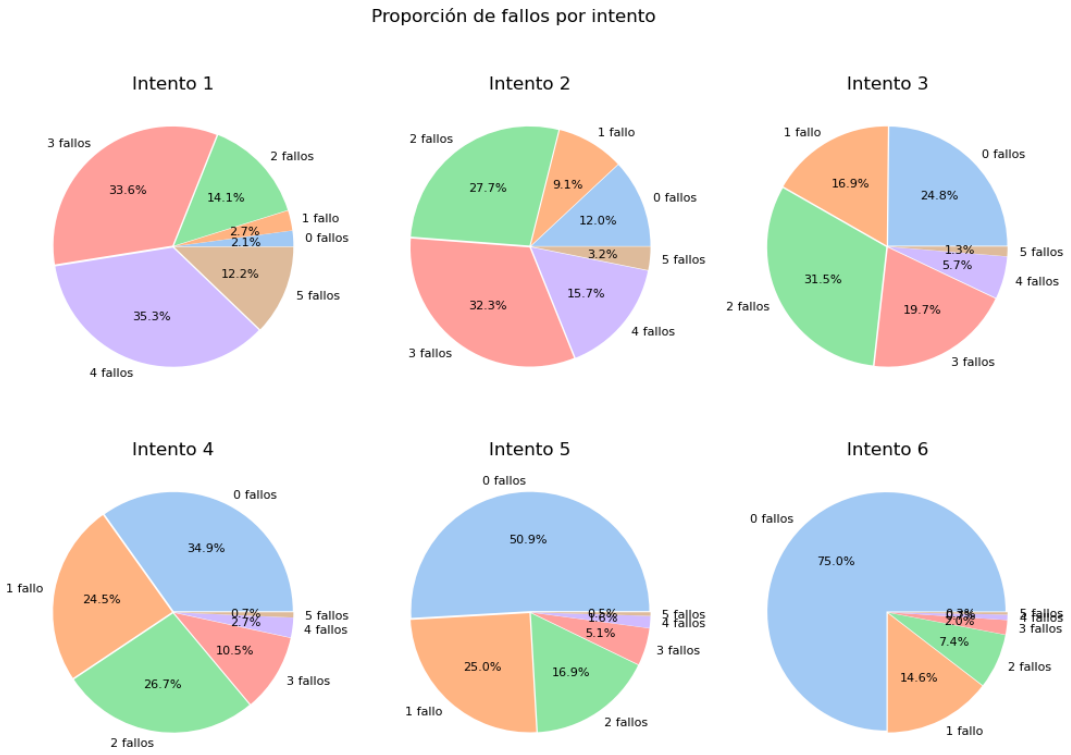


Figura 5.7: Proporción de fallos por intento

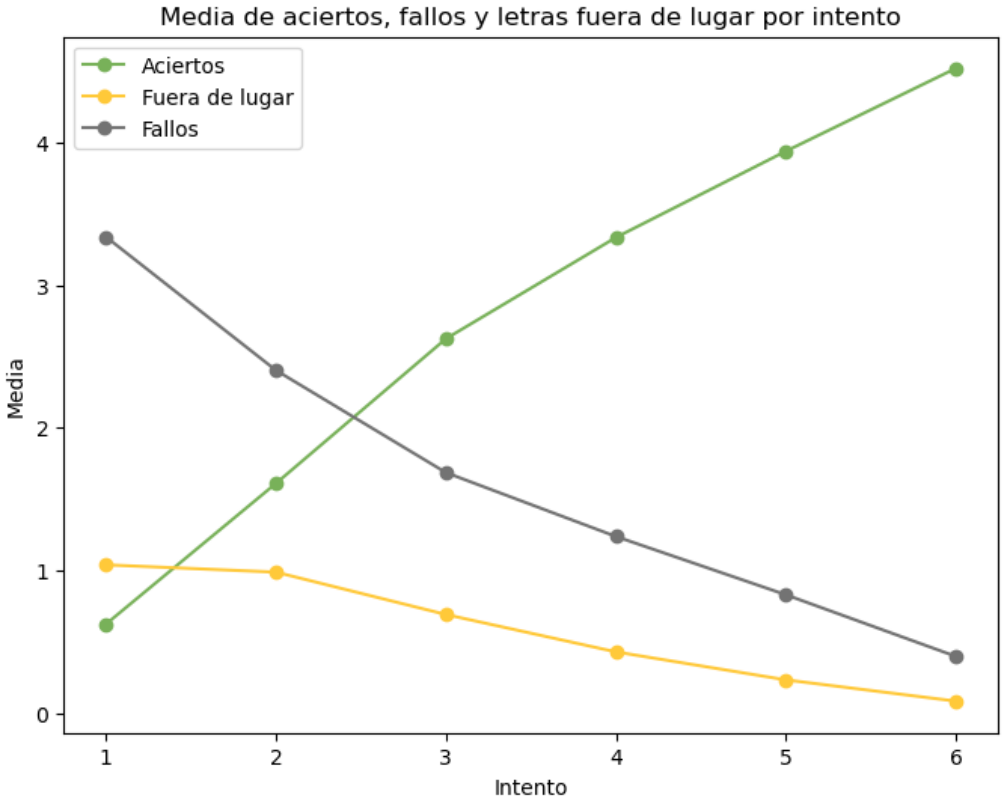


Figura 5.8: Media de aciertos, fallos y letras fuera de lugar

5.1.4. Tasa de fallos

Seguimos ahora con el análisis de la distribución del número de fallos (cuadrados negros) de las partidas que estamos estudiando. En los gráficos 5.6 y 5.7 podemos ver que el número de fallos disminuye a medida que aumenta el número de intentos, como podíamos imaginar ya que la distribución de los fallos debe ser prácticamente opuesta a la de los aciertos. Destaca la heterogeneidad de las distribuciones salvo las de los intentos 5 y 6 lo cúal contrasta con el número de letras fuera de lugar pero resulta similar a las del número de aciertos.

5.1.5. Evolución de las medias

Finalmente analizamos la distribución de las medias de cada una de las tasas presentadas en las secciones anteriores, pretendemos analizar el crecimiento y decrecimiento de estos valores ya que podrían ser usados como variables para caracterizar a los jugadores. En la figura 5.8 podemos ver claramente que el número de aciertos tiene un crecimiento prácticamente lineal que disminuye su pendiente en el intento 3, algo que parece reforzar la hipótesis de que es un conjunto de interés para nuestro trabajo. La media de fallos y letras fuera de lugar es claramente decreciente, cabe destacar que la distribución de los fallos disminuye su pendiente también en el intento 3, lo cuál es de esperar daba la correlación entre esta y la distribución de aciertos. Resulta sorprendente el bajo valor que toma la distribución de las letras fuera de

lugar, pues está acotada prácticamente por 1 y tiene un decrecimiento mucho más lento que el resto, esto indica que la mayoría de jugadores obtiene una letra en una posición incorrecta y a lo largo de sus intentos la coloca en una posición correcta y obtiene otra nueva letra en una posición errónea, dado el aumento constante de los aciertos.

5.2. Análisis según la palabra objetivo

Continuamos nuestro análisis ahora tratando de encontrar relaciones entre los datos de las partidas y su palabra objetivo. Esto nos permitirá generar nuevas variables que posiblemente contengan información relevante a la hora de encontrar diferencias entre jugadores.

5.2.1. Distribución de las partidas según su duración

Procedemos a analizar el número de partidas según su palabra objetivo y considerando su duración y comparándola con la duración media. Este análisis es necesario para asegurar la uniformidad de los datos disponibles tanto en representatividad de cada palabra objetivo así como de variabilidad de partidas. Los resultados han sido los esperados: la distribución no presenta ningún valor anómalo, sí apreciamos una tendencia hacia la disminución del número total de partidas a media que incrementa la fecha de las partidas pero no es estadísticamente significante y se asocia a la pérdida gradual de popularidad del juego.

En la figura 5.9 podemos ver una muestra de esta distribución donde las barras rojas representan las partidas con un número de intentos por encima de la media y las azules las que se encuentran por debajo. Observamos la uniformidad de los datos y la ligera predominancia de partidas más cortas.

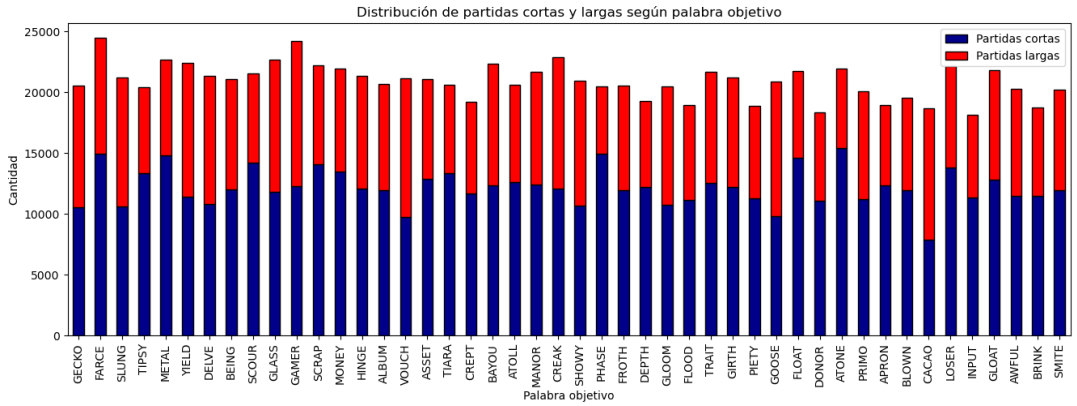


Figura 5.9: Muestra de 45 elementos del número de partidas por palabra objetivo y según su duración

5.2.2. Dificultad de las palabras

Procedemos a evaluar la complejidad de cada palabra basándonos en la longitud media de las partidas que la tienen como objetivo. Calculando la media de la longitud de todas las partidas y comparándola con la de cada palabra, podremos establecer un criterio simple para distinguir si una palabra es fácil o difícil, si la media de sus partidas es inferior a la media global la clasificamos como fácil y en caso contrario como dificil. En la figura 5.10 podemos ver esta caracterización para un conjunto reducido de palabras objetivo, donde las barras rojas representan las difíciles y las azules las fáciles. Descatamos que la media global de intentos se encuentra en 4,2, que la tendencia general de la distribución de la dificultad no presenta ningún patrón claro y que la aparición de palabras fáciles y difíciles no parece tener relación con ninguna variable.

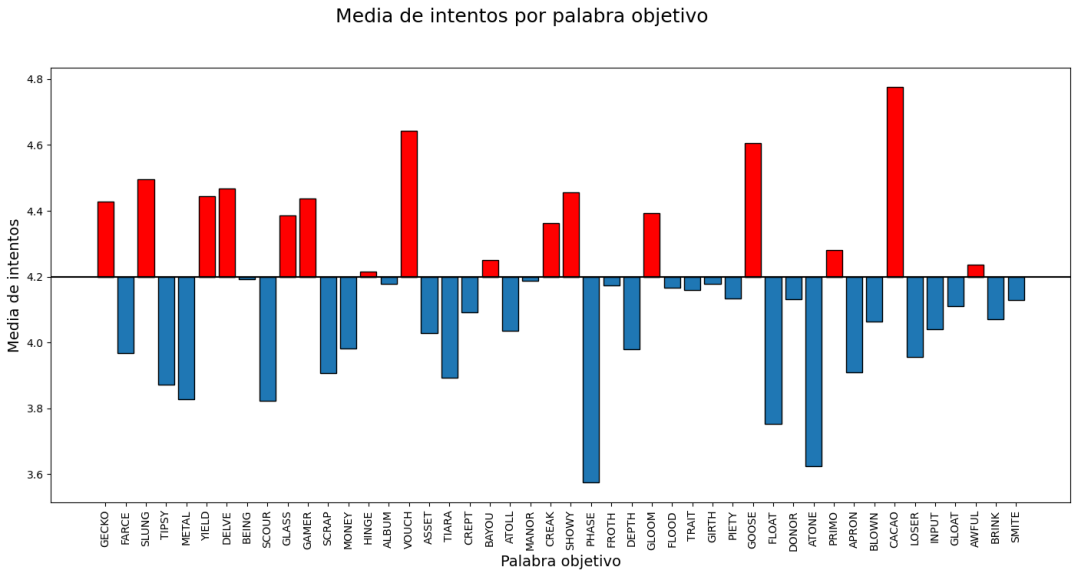


Figura 5.10: Muestra de 45 elementos de la media de intentos por palabra objetivo

En la tabla 5.3 mostramos las cinco palabras más difíciles y las 5 más fáciles junto a la media de intentos de sus partidas correspondientes, destacamos que todas las palabras fáciles contienen al menos dos vocales y no presentan letras repetidas salvo en el caso de *TREAT*, además son frecuentes en el lenguaje, por otra parte las difíciles presentan dos o más letras repetidas en todos los casos salvo en la palabra *GAWKY* y contienen únicamente una vocal salvo en el caso de *PARER*, además frecuencia de todas ellas en el lenguaje coloquial es reducida.

Palabrás más fáciles	Media	Palabras más difíciles	Media
TRAIN	3,283	PARER	5,514
RAINY	3,449	MUMMY	5,389
DREAM	3,471	COYLY	5,241
ALIEN	3,477	GAWKY	5,028
TREAT	3,525	DANDY	4,920

Tabla 5.3: Palabras con menor y mayor media de intentos

Esta información nos resultará útil a para caracterizar los grupos que encontremos en la fase de clústering y también permitirán evaluar a los agentes que crearemos en apartados posteriores.

Capítulo 6

Tipos de jugadores

En este apartado trataremos de clasificar a los diversos jugadores presentes en el conjunto de datos recolectado. Para ello los agruparemos según el identificador de usuario, que es un valor único y permite distinguir inequívocamente a cada jugador y estudiaremos las partidas que haya realizado. A continuación generaremos variables a partir de estos datos que utilizaremos para caracterizarlos, aplicaremos algoritmos de clústering para obtener clases de jugadores, evaluaremos los resultados utilizando distintas métricas. Finalmente trataremos de caracterizar a los grupos obtenidos mediante un análisis de las variables que resulten más relevantes.

6.1. Distribución del número de partidas de cada jugador

El primer paso para obtener la clasificación de jugadores es el de agrupar partidas. Necesitamos conocer el número de usuarios distintos que poseemos así como la distribución del número de partidas del que disponemos de cada uno, esta información queda resumida en la tabla 6.1, en ella podemos intuir que nos encontramos ante una distribución muy sesgada hacia la derecha puesto que los dos primeros cuartiles continen únicamente jugadores con una o dos partidas pero alcanzando el máximo en 295. Además la desviación típica es de un orden de magnitud superior a los tres cuartiles y eso indica que existirá una gran varianza de los valores más alla del tercer cuartil. Esto queda explicitado en la figura 6.1 donde podemos ver el extremo sesgo de la distribución.

En aras de realizar un estudio realista y estadísticamente relevante de los datos hemos decidido establecer un umbral en el número mínimo de partidas de los usuarios para tener suficientes datos para analizar el estilo de juego del jugador. Por esto hemos tomado el umbral más cercano a la media de la distribución y, consecuentemente, lo hemos situado en siete partidas. El número de jugadores se ve reducido de manera importante en el subconjunto, pasamos de casi un millón a menos de 154,000. Por otra parte la media de partidas pasa a elevarse enormemente y se sitúa en las $33,72$ partidas y la desviación típica pasa a estar en el mismo orden de mag-

nitud que la media, tal y como podemos apreciar en la tabla 6.2. A pesar de esta mejoría, la distribución seguirá estando fuertemente sesgada a la derecha aunque en la figura 6.2 podemos apreciar la reducción de este fenómeno notablemente.

En base a las similitud de las distribuciones y de la cantidad de datos presentes en el subconjunto lo consideramos suficientemente relevante y procederemos a utilizarlo en los siguientes apartados como conjunto base para realizar *clústering*. Para terminar presentamos la distribución en escala logarítmica en la figura 6.3 donde podemos apreciar mejor como se reparten los valores. Observamos que el sesgo a la derecha es imperante.

Estimadores de la distribución del número de partidas	
Nº de usuarios	977.510
Media	7,035
Desviación típica	19,436
Mínimo	1
Máximo	295
25 %	1
50 %	2
75 %	4

Tabla 6.1: Estimadores de la distribución del número de partidas por jugador

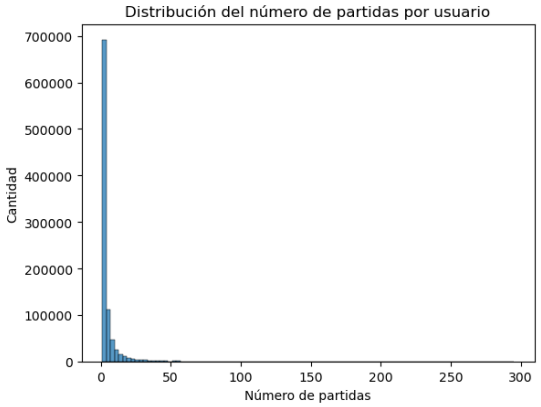


Figura 6.1: Distribución del número de partidas por jugador

Estimadores de la distribución del subconjunto de jugadores	
Nº de usuarios	153.893
Media	33,721
Desviación típica	39,262
Mínimo	7
Máximo	295
25 %	10
50 %	17
75 %	37

Tabla 6.2: Estimadores de la distribución del número de partidas para jugadores con siete o más partidas

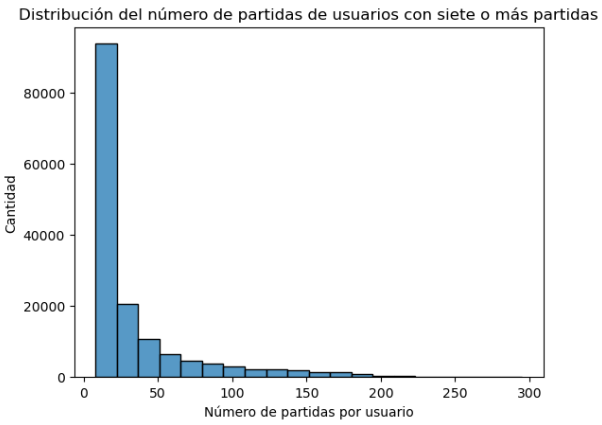


Figura 6.2: Distribución para jugadores con siete o más partidas

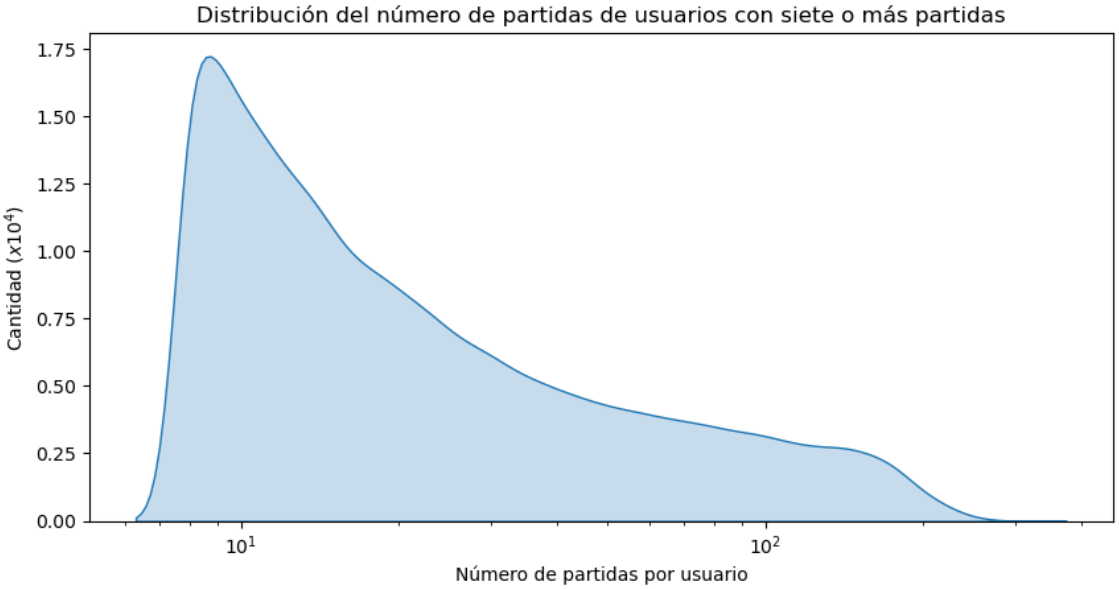


Figura 6.3: Distribución del subconjunto de jugadores con 7 o más partidas en escala logarítmica

6.2. Generación de las variables para el agrupamiento

Dada la naturaleza de los datos presentes en nuestro conjunto, resulta complicado aplicarles algún algoritmo de clústering. Debido a esto, debemos generar variables interpretables por los distintos métodos basadas en los conocimientos que hemos obtenido al realizar el análisis previo. Además, dado el objetivo de clasificar jugadores, denotamos también la relevancia de generar variables relacionadas con las métricas empleadas para evaluarlos, tomaremos como referencia las que utiliza el periódico *The New York Times* en su bot oficial [25] y, consecuentemente, haremos incapié en la media de intentos de las partidas de cada jugador e intentaremos aproximar el porcentaje de disminución del espacio de soluciones dado por cada palabra que introduce.

Comenzaremos generando las variables que podemos obtener más fácilmente a partir de los datos que poseemos.

- **Longitud media de partida**: Generamos una variable que contiene la longitud media de las partidas de cada jugador presente en el conjunto de datos. Esta variable permite evaluar fácilmente a un jugador ya que dentro de las métricas de evaluación del rendimiento, esta es la más utilizada y fácilmente reconocible. Nos referiremos a ella como `mean_length`.
- **Media de partidas perdidas**: Para usuario calculamos el porcentaje de partidas que ha perdido. Añadimos esta métrica para complementar la longitud media ya que esta no tiene en cuenta si un juego de longitud seis ha resultado en victoria o derrota. Además también puede ayudar a caracterizar jugadores

ya que, en general, un mayor número de victorias es indicativo de un mejor jugador. Nos referiremos a ella como `mean_loss`.

- **Número medio de aciertos, fallos y letras fuera de lugar por partida**: Generamos tres variables que contabilizan la media de aciertos, fallos y letras fuera de lugar. Con ellas se busca ver si, a parte de las diferencias provinientes de las diferencias de longitud entre jugadores, existen subgrupos de usuarios que utilicen distintas tácticas como utilizar paralabras con pocas letras en común o similares a las anteriormente introducidas. Nos referiremos a ellas como `mean_green`, `mean_yellow` y `mean_white` respectivamente.

- **Media de la diferencia entre acietos, fallos y letras fuera de lugar entre intentos consecutivos**: Creamos otras tres variables que contienen la media de las diferencias entre intentos consecutivos de los aciertos, fallos y las letras fuera de lugar, tratando de modelar así la evolución de la partida y por tanto de la habilidad del jugador ya que, por ejemplo, el rápido aumento de aciertos estará relacionado con partidas más cortas y por tanto mejores jugadores. Nos referiremos a ellas como `mean_green_diff`, `mean_yellow_diff` y `mean_white_diff` respectivamente.

- **Media de aciertos, fallos y letras fuera de lugar en el tercer intento intento**: Generamos tres variables que contienen la media de aciertos, fallos y letras fuera de lugar en el tercer intento de cada partida para cada jugador. Esta variable está motivada por los resultados obtenidos en 5.3, 5.5 y 5.7 ya que en este intento es donde podemos observar la mayor hetereogeneidad entre el número total de estos valores y, por tanto, donde posiblemente se diferencien más claramente los tipos de jugadores. Nos referiremos a ellas como `mean_green_third`, `mean_yellow_third` y `mean_white_third` respectivamente.

El cálculo de la métrica de reducción del espacio de soluciones por palabra introducida no puede ser realizado de manera simple ya que no poseemos las palabras introducidas por los usuarios, solamente tenemos los códigos producidos al introducirlas y la palabra objetivo, esto claramente no genera una relación biyecciva pues dos palabras distintas pueden producir el mismo código por ejemplo si introducimos como palabra *ABBOT* y la palabera objetivo es *ABBEY* obtendremos como resultado el mismo código que si hubiéramos introducido la palabra *ABBAS*.

Para aproximar el valor de la métrica hemos optado por aproximar la primera palabra introducida por el usuario en cada partida usando un analizador de frecuencias de aparición de palabras en el idioma inglés, para ello hemos generado el conjunto de posibles palabras iniciales empleadas para cada partida y hemos seleccionado la palabra más frecuente de entre todas las posibilidades usando la librería `wordfreq`. A continuación, hemos precalculado la disminución del espacio de soluciones para cada código obtenido en el primer intento, cada palabra introducida y cada posible palabra objetivo. Finalmente hemos calculado la media de estos valores para cada jugador y generado una variable a la que hemos llamado `mean_reduc`.

6.2.1. Correlación entre las variables

Antes de poder aplicar los algoritmos de clústering debemos analizar las relaciones entre las variables que hemos generado para evitar sesgar nuestro algoritmo en caso de que existan valores muy correlacionados, para ello utilizamos la matriz de correlación que indica el coeficiente de correlación de Pearson entre dos variables. Un coeficiente cercado a uno indica correlación directa, uno cercano a cero indica ausencia de correlación y uno próxima a menos uno indica correlación inversa[26]. En la figura 6.4 mostramos estos valores acompañados de una escala de color para facilar la indentifiación de las variables muy correladas. En particular vemos que las variables `mean_length` y `mean_green_third` están altamente correlacionadas con un gran número de variables y que la variable con menor relación con las demás es `mean_reduc`.

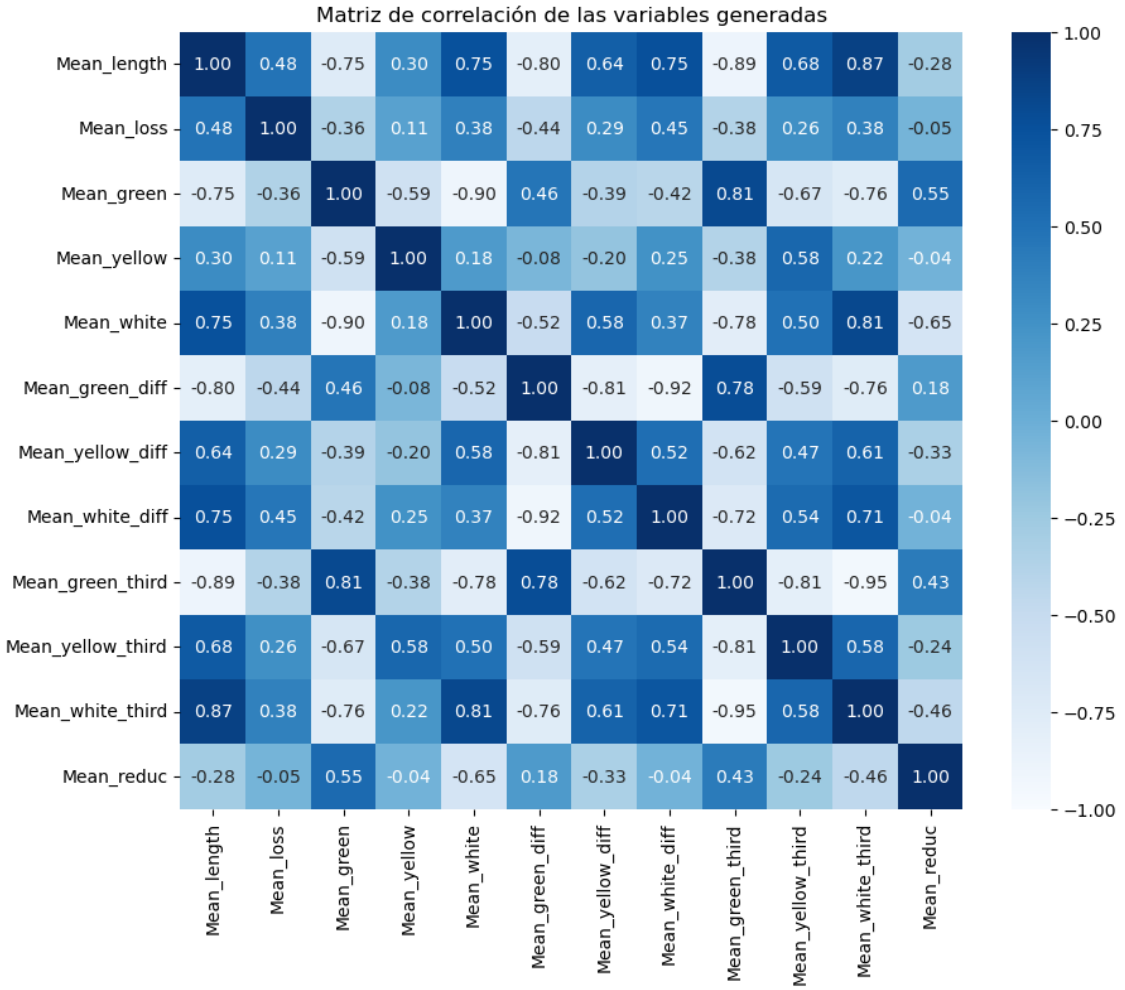


Figura 6.4: Mapa de calor de la matriz de correlación

Esta gran cantidad de correlaciones causaría cierto sesgo en la mayoría de algoritmos de clústering. A medida que aumenta el número de dimensiones, los puntos de datos se dispersan más en el espacio de variables, esto es debido al aumento volumen del espacio de datos ya que este incremento causa que el número de puntos de datos

requeridos para mantener un cierto nivel de densidad en el espacio crezca exponencialmente con el número de dimensiones. Como resultado, los datos se vuelven cada vez más dispersos y las distancias relativas entre los puntos se vuelven menos significativas.

Otro problema que podmeos encontrar es que la distancia entre los puntos de datos tienden a volverse más uniformes, lo que dificulta la distinción entre grupos significativos y no significativos. En espacios de alta dimensionalidad, los puntos de datos pueden parecer equidistantes entre sí, lo que dificulta todavía más la distinción de los grupos [30, 9].

Para solucionar esto hemos decido realizar un análisis de componentes principales ya que pese a los altos coeficientes de correlación, creemos que el conjunto de variables contiene suficiente información relevante que perderíamos si simplemente seleccionásemos las variables más importantes.

6.2.2. Análisis de componentes principales

El análisis de componentes principales (ACP) [24] es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos transformando las variables originales en un conjunto más pequeño de variables linealmente no correlacionadas llamadas componentes principales. El objetivo del ACP es identificar los patrones más importantes en un conjunto de datos y resumirlos en un conjunto más pequeño de variables que expliquen la mayoría de la varianza. [18]

ACP funciona encontrando las combinaciones lineales de las variables originales que explican la mayor cantidad de varianza en los datos. La primera componente principal es la combinación lineal de las variables que explican la mayor cantidad de varianza en los datos, y cada componente principal siguiente explica la mayor cantidad de varianza restante manteniendo la ortogonalidad a las componentes anteriores.

El ACP se puede utilizar para una variedad de propósitos, como compresión de datos, visualización de datos y preprocesamiento de datos para algoritmos de aprendizaje automático. Al reducir el número de dimensiones en un conjunto de datos, ACP puede simplificar el análisis de datos y facilitar la identificación de patrones y relaciones en los datos. ACP también se puede utilizar para eliminar el ruido y mejorar el rendimiento de los algoritmos de aprendizaje automático.

En aplicaciones prácticas, ACP a menudo se utiliza en conjunto con otras técnicas estadísticas, como análisis de agrupamiento o análisis de regresión, para ayudar a comprender conjuntos de datos complejos. Si bien puede ser una herramienta útil para nuestro análisis, es importante interpretar cuidadosamente los resultados y considerar las limitaciones de la técnica. ACP asume que los datos están relacionados de manera lineal y distribuidos normalmente, por lo tanto debemos normalizar nuestras variables antes de aplicar esta técnica [15, 40].

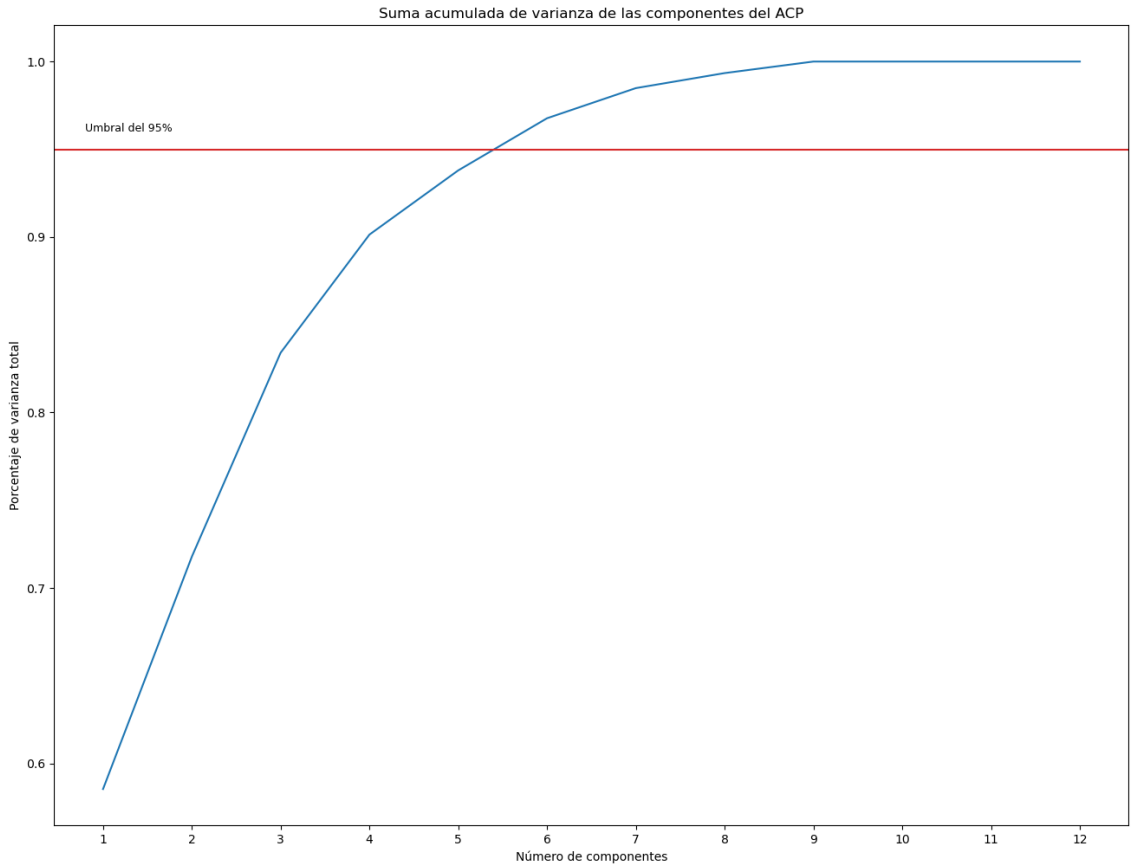


Figura 6.5: Suma acumulada del porcentaje de la varianza total de cada componente del ACP

Para nuestro proyecto hemos utilizado el método `PCA` de la librería `sklearn` de Python que nos permite transformar los datos de manera sencilla. En la figura 6.5 podemos ver el porcentaje acumulado de la varianza de cada una de las componentes obtenidas, podemos ver que las primeras seis componentes contienen más del 95 % de la varianza total, tomaremos este valor como umbral y por tanto utilizaremos hasta seis variables en nuestros algoritmos de clústering.

6.3. Obtención de los grupos

A partir de las variables generadas en el apartado anterior hemos obtenido diversos grupos utilizando el algoritmo k-medias [31, 12], para evaluar la ideonead del número de conjuntos hemos utilizado el coeficiente de Silhouette y también el diagrama de codo haciendo uso de la inercia, una medida basada en la suma del cuadrado de las distancias de cada punto a su centroide [12]. El uso de ACP nos permite también implementar fácilmente el método de selección de variables ya que para aplicarlo simplemente tenemos que eliminar gradualmente las variables con menor porcentaje de varianza. Por ello hemos ejecutado el algoritmo con cinco, tres y dos variables y, además, también hemos utilizado las variables originales pero solamente manteniendo aquellas cuya correlación estuviera por debajo de 0,8 y por

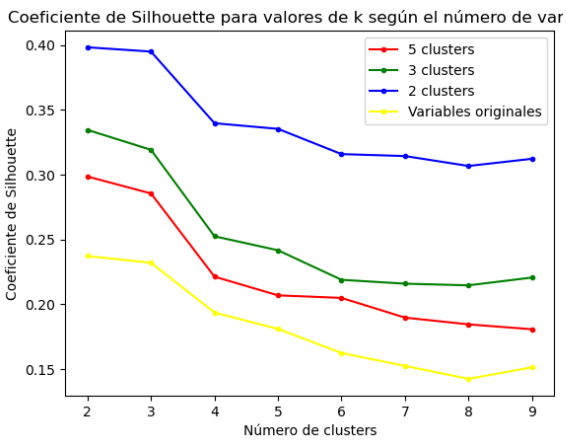


Figura 6.6: Coeficiente de Silhouette

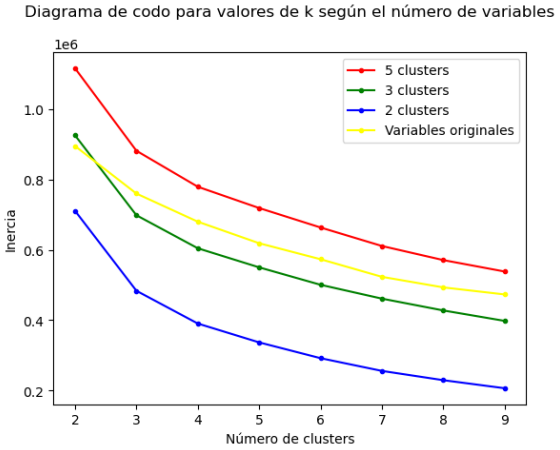


Figura 6.7: Diagrama de codo

encima de $-0{,}8$. Podemos observar los resultados obtenidos en las figuras 6.6 y 6.7.

En base a las observaciones hemos determinado que, pese a que $k = 3$ tiene un coeficiente de Silhouette ligeramente inferior a $k = 2$, este primero es el punto de inflexión de todos los diagramas de codo y por tanto consideramos que es el número adecuado de clústers. Por lo anterior determinamos que los mejores resultados se obtienen al utilizar únicamente las dos variables más significativas de ACP, formando 3 clústers. Los valores obtenidos para estos parámetros para el coeficiente de Silhouette es de $0,395$ y la inercia de $483668,243$.

Dado que nuestro agrupamiento ha utilizado únicamente dos componentes de ACP, podemos observar gráficamente la clasificación realizada por el algoritmo en la figura 6.8. Claramente la clasificación se basa práctimente en la componente cero del análisis, lo cuál es razonable debido a que esta contiene más del 80 % del la varianza. Tras consultar el perso de la variable `mean_length` dentro de la primera componente del ACP concluimos que esta es la más relevante de todas. No obstante, realizaremos un estudio detallo de la distribución de las variables de cada jugador según su clasificación.

6.4. Análisis de los grupos

Una vez obtenidos los grupos pasamos a analizar la distribución de las variables según su clasificación. Tras observar las distribuciones de las variables originales presentamos en las figuras 6.9 y 6.10 aquellas con las diferencias más significativas correspondientes a la longitud media de las partidas y a la media de aciertos en el intento 3. Los valores relacionados con la representatividad de cada grupo en el conjunto total se resumen en la tabla 6.3.

A partir de estas distribuciones podemos caracterizar los grupos de la siguiente ma-

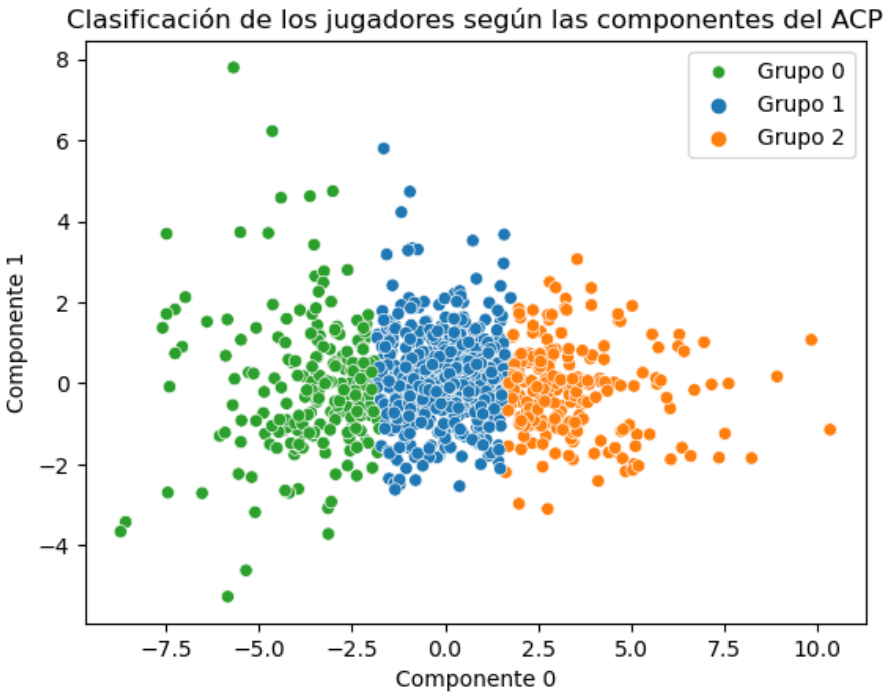


Figura 6.8: Clasificación de los jugadores según las componentes generadas por el ACP

Grupo	**0**	**1**	**2**	**Total**
Nº de jugadores	86.095	35.868	31.930	153.893
Proporción	55,94	23,30	20,76	100

Tabla 6.3: Representatividad de cada uno de los grupos de jugadores

nera: el grupo 0 corresponde a los jugadores cuyas partidas tienen una duración media de 4,23 intentos, son los jugadores más comunes y los que obtienen un rendimiento medio. El grupo 1 corresponde a los jugadores cuyas partidas tienen una media de 4,99 intentos, son un grupo más reducido que el anterior y representan a los jugadores con el peor rendimiento. Finalmente el grupo 2 corresponde a los jugadores con una media de intentos de 3,31, son el grupo minoritario y representan a los mejores jugadores. Teniendo en cuenta también las distribuciones de la figura 6.10, concluimos también que los jugadores del grupo 0 en el tercer intento han acertado una media de 2,91 letras, es decir conocen un 58,2 % del contenido de la solución, los del grupo 1 una media de $1,77$ es decir un $35,4\,\%$ de la solución y finalmente el grupo 2 una media de $4,02$ y por tanto un $80,4\,\%$ de las letras de palabra objetivo. Esto es esperable puesto que cuanto menos dure una partida más aciertos se esperan obtener un el tercer intento.

Dados los resultados obtenidos en esta fase sería fácil concluir que el único factor relevante que hemos encontrado para diferenciar a los juagadores es la longitud

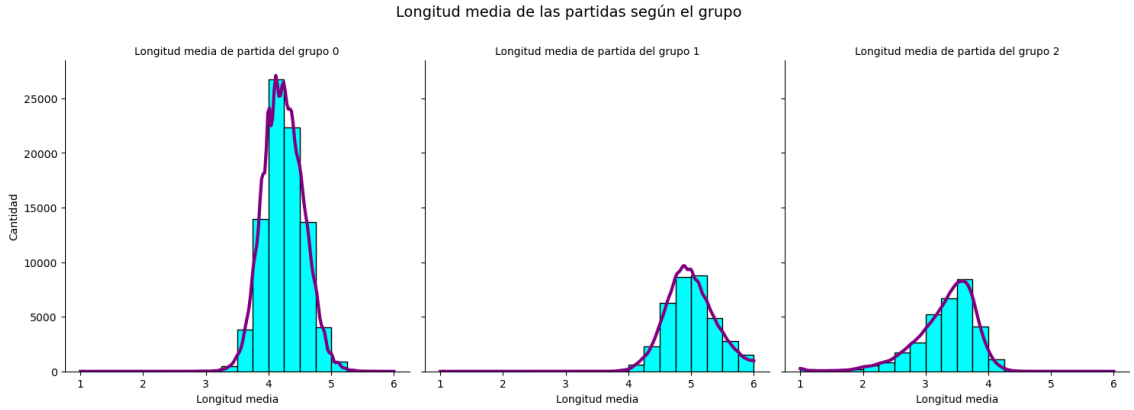


Figura 6.9: Longitud media de las partidas según la agrupación por clústers

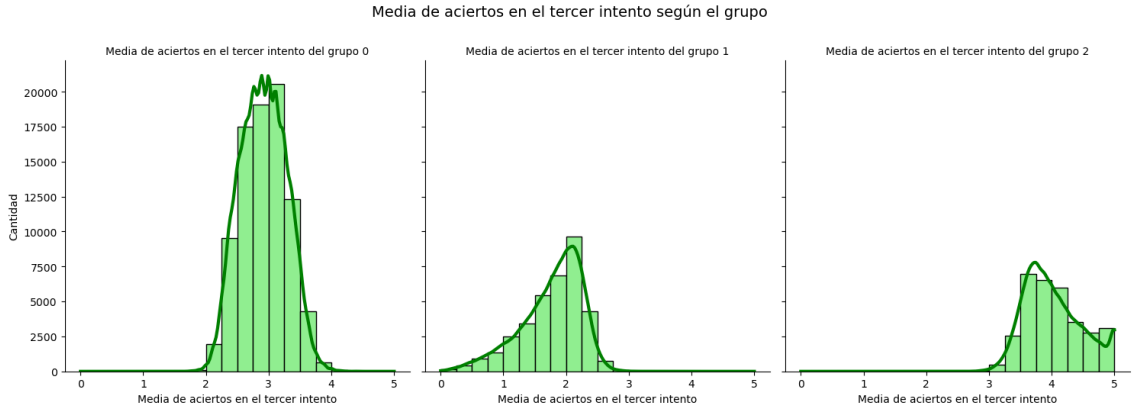


Figura 6.10: Número de aciertos en el tercer intento según la agrupación por clústers

media de sus partidas, no obstante si estudiamos la distribución de las medias de aciertos, letras fuera de lugar y fallos de los tres grupos, presentes en los gráficos de la figura 6.11 podemos ver una clara diferenciación entre los tres, que se ve acentúa tercer intento. Creemos que estas gráficas permiten apreciar el desarrollo de las partidas y el estilo de juego de los grupos encontrados así como las diferencias entre estos, por ejemplo, el grupo 1 suele reducir su número de errores en una unidad por cada dos intentos y obteniendo una media de una letra en posición incorrecta en los tres primeros intentos; el grupo 2 en cambio reduce su número de fallos en una unidad por intento y, a su vez, obtiene reduce muy rápidamente la presencia letras en una posición errónea. Esto sugiere que los jugadores del grupo 1 en los primeros intentos utilizan palabras muy similares entre sí y obtienen resultados malos en los primeros intentos mientras que, el grupo 2, utiliza palabras que mantienen las letras correctas en su posición pero varían el resto de posiciones erróneas. Ejemplos como este pueden indicar una relación entre el vocabulario de los jugadores y su rendimiento, por ello, hemos considerado relevante estudiar el posible vocabulario de cada grupo.

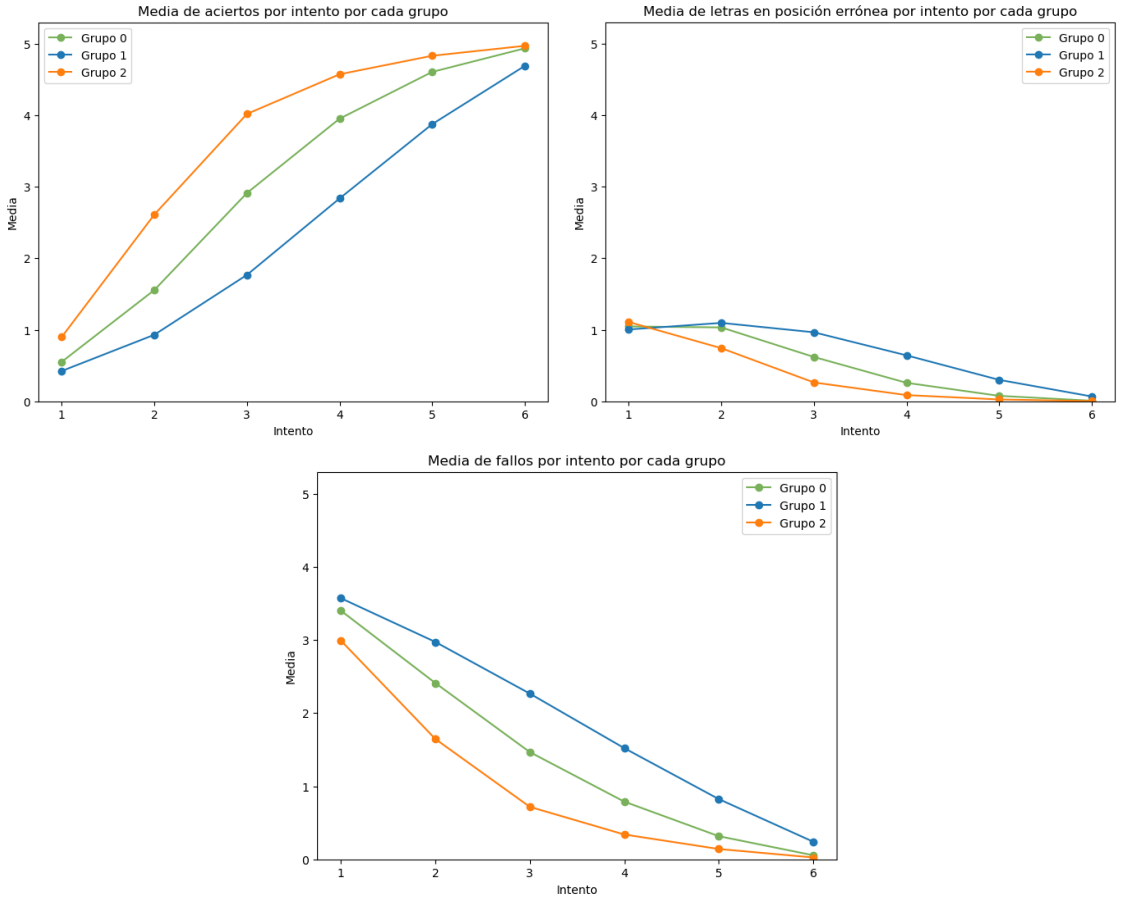


Figura 6.11: Distribución de la media de aciertos, letras fuera de lugar y fallos de cada grupo

6.4.1. Estudio de las palabras iniciales

La elección de la palabra del primer intento es diferente al resto puesto que no se tiene ninguna información acerca de la palabra objetivo, consideramos importante estudiar con detenimiento las posibles palabras que cada grupo podría utilizar. Cada posible palabra inicial disminuye un porcentaje medio diferente del espacio de soluciones, en particular, las mejores palabras son aquellas que no contienen letras repetidas e incluyen las consonantes más frecuentes en la lengua inglesa [3]. Así pues, palabras como *TARES*, *SLATE* o *CRATE* suelen ser las más utilizadas para inicializar los algoritmos, no obstante, en este proyecto no buscamos minimizar la longitud de las partidas, sino imitar en la mejor medida de lo posible a los jugadores.

Para tratar de encontrar las palabras iniciales más comunes en cada grupo hemos analizado las combinaciones iniciales de cada partida de cada usuario y creado un conjunto de posibles palabras introducidas dado el código obtenido y la palabra objetivo y, a continuación, hemos contabilizado la aparición de cada palabra en cada grupo. Este método nos ha proporcionado una lista ordenada de mayor a menor frecuencia de aparición de cada palabra para cada grupo, no obstante, los resultados están lejos de ser realistas puesto que aparecen palabras que probablemente no sean

conocidas por la mayoría de jugadores como *PZAZZ*, esto se debe a que este tipo de palabras con una gran cantidad de letras repetidas y pocas vocales son las que producen en la mayoría de casos combinaciones con cinco fallos o cuatro fallos y una letra fuera de lugar, que son las frecuentes entre las partidas que disponemos.

Para aproximar mejor el vocabulario de nuestros jugadores hemos pensado en tener en cuenta la frecuencia de aparición en la lengua inglesa de las palabras y también el número de letras no repetidas de cada una. Consideramos que el primer criterio está justificado debido a que las partidas se han obtenido de cuentas de habla inglesa y el segundo se basa en la hipótesis de que la mayoría de jugadores utilizan como estrategia inicial el reducir el espacio de posibles soluciones eliminando la mayor cantidad posible de letras. Para incorporar estos dos criterios adicionales hemos recurrido a la técnica de análisis de decisión multicriterio más simple, la media ponderada. La frecuencia de aparición de cada palabra se ha calculado gracias a la librería `wordfrecs` y el porcentaje de letras repetidas en cada palabra lo hemos calculado con una función propia. Una vez obtenidas todas las distribuciones las hemos normalizado y hemos calculado el nuevo valor en base a la fórmula 6.1 donde $value(i)$ corresponde al valor asignado a la palabra i, $frec\_ini(i)$ es el valor normalizado de la frecuencia de aparición de la palabra en la distribución calculada al inicio del subapartado, $frec\_nat$ es el valor normalizado de frecuencia de aparición de la palabra i en la lengua inglesa y $repeat\_char(i)$ es el porcentaje normalizado de letras que aparecen más de una ocasión en la palabra i.

$$value(i) = frec\_ini(i) + 0{,}1 \cdot frec\_nat(i) + 0,2 \cdot repeat\_char(i) \tag{6.1}$$

Este enfoque ha propocionado resultados mucho más plausibles. eEn las tablas 6.4, podemos observar las tres palabras con mayor puntuación para cada grupo. Descatamos la similitud entre las del grupo 0 y las del grupo 2, todas terminan en E y contienen la letra A en la segunda posición, además, la mayoría contienen las letras T y R que son las consonantes más comunes en la lengua inglesa. Las diferencuas más notables se hallan en las palabras del grupo 1, en este parece que las palabras con varias vocales distintas tienen una mayor frecuencia de aparición y en la palabra con mayor valor encontramos una única consonante, a diferencia de las otras mejores palabras de cada grupo que incluyen tres.

Palabras iniciales para el grupo 0	
Palabra	Valor
STATE	4,91
BRAZE	4,78
CRAZE	4,46

Palabras iniciales para el grupo 1	
Palabra	Valor
ADIEU	4,43
BRAZE	4,28
ABOUT	4,07

Palabras iniciales para el grupo 2	
Palabra	Valor
STARE	6,42
STATE	6,18
SPARE	5,31

Tabla 6.4: Palabras iniciales para cada grupo con la media agregada más alta

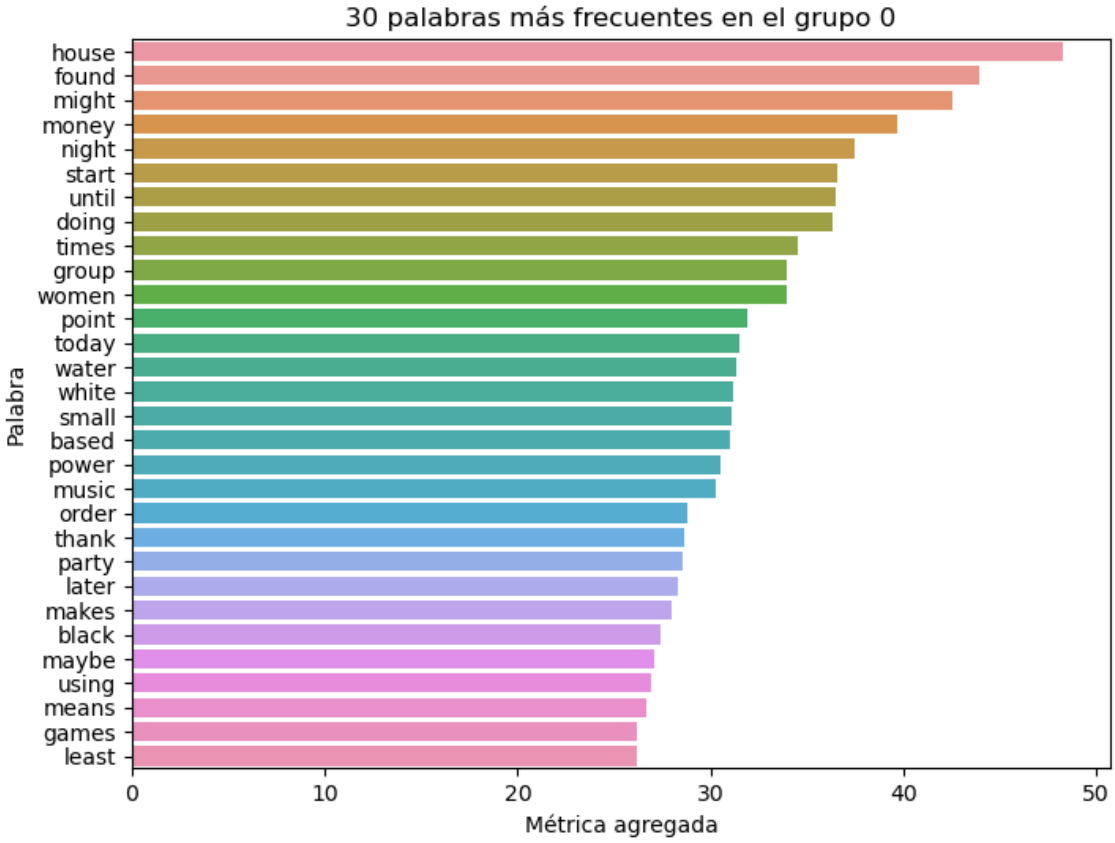
6.4.2. Estudio del vocabulario de cada grupo

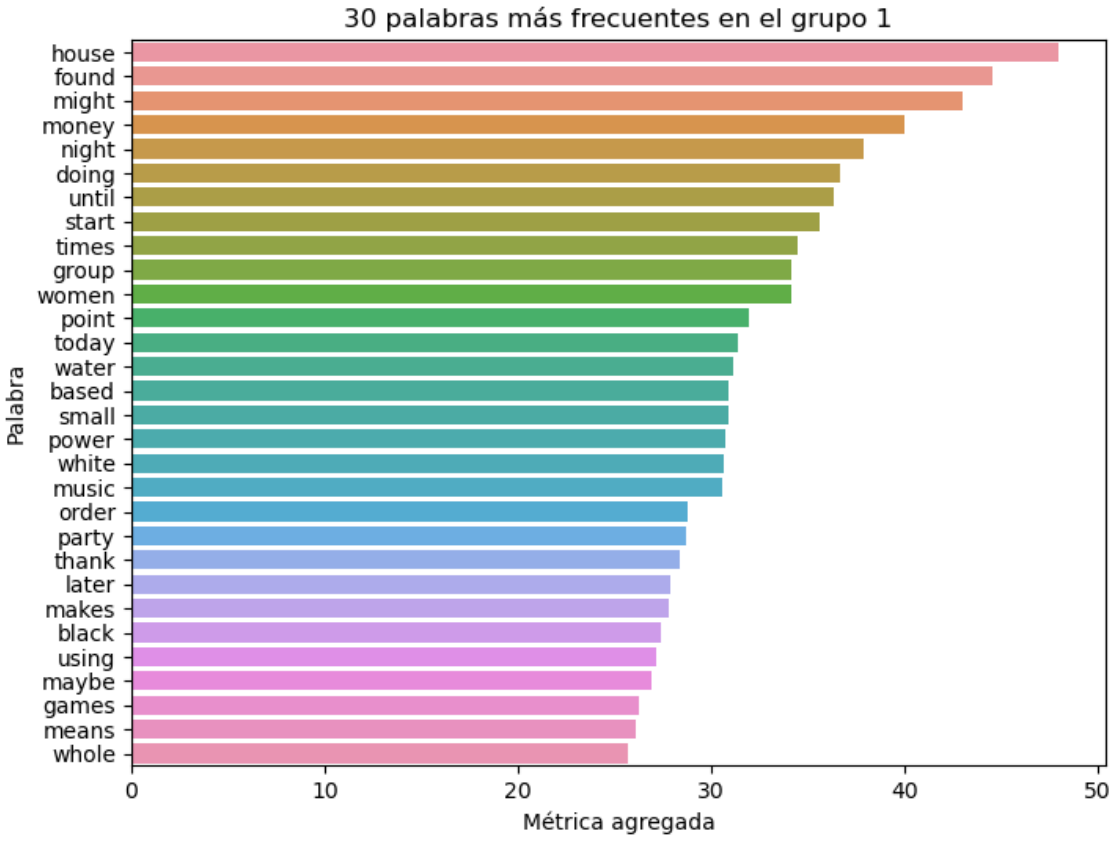
Terminamos el análisis de los grupos generados con el estudio del vocabulario de cada uno de ellos, en concreto, pretendemos generar un valor numérico para cada palabra que represente cuán habitual es en el vocabulario de cada tipo de jugador. Para ello hemos analizado cada patrón obtenido en cada intento de cada partida generando una nueva lista de frecuencias de aparición, también hemos recuperado las frecuencias en el lenguaje natural pero no hemos tenido en cuenta la repetición de las letras puesto que las palabras utilizadas por cada jugador están fuertemente influenciadas por los resultados obtenidos en intentos previos. Para combinar estas dos métricas hemos vuelto a usar una media ponderada pero esta vez reajustando los pesos de cada distribución normalizada, en particular hemos usado la fórmula 6.2 donde podemos ver que el coeficiente de la frecuencia en el lenguaje natural es seis y en el obtenido mediante el análisis de los patrones es uno, este claro sesgo hacia el lenguaje natural lo justificamos para tratar de compensar la gran cantidad de datos erróneos que se obtienen al realizar el análisis de los patrones puesto que solamente podemos recuperar un conjunto de posibilidades no la palabra introducida por el usuario.

$$value\_vocab(i) = frec\_ini(i) + 6 \cdot frec\_nat(i) \tag{6.2}$$

Tras generar los valores para cada grupo hemos obtenido tres ránkings de palabras. Todos ellos presentan coincidencias entre sí pero sus distribuciones están suficientemente diferenciadas, podemos ver en las gráficas presentes en la figura 6.12 las treinta palabras con mayor media agregada para cada tipo de jugador. A diferencia de lo observado en la subsección 6.4.1, no observamos cambios drásticos en ninguna de las tres distribuciones, aunque sí existen variaciones entre las posiciones de cada palabra.

En el próximo capítulo tilizaremos este conjunto de datos para modelar el comportamiento de cada grupo de jugadores e intentar aproximar su media de intentos así como las distribuciones de los aciertos, fallos y letras fuera de lugar según cada intento.


30 palabras más frecuentes en el grupo 0
house
found
might
money
night
start
until
doing
times
group
women
point
today
water
white
small
based
power
music
order
thank
party
later
makes
black
maybe
using
means
games
least
Palabra
0
10
20
30
40
50
Métrica agregada


30 palabras más frecuentes en el grupo 1
house
found
might
money
night
doing
until
start
times
group
women
point
today
water
based
small
power
white
music
order
party
thank
later
makes
black
using
maybe
games
means
whole
Palabra
0
10
20
30
40
50
Métrica agregada

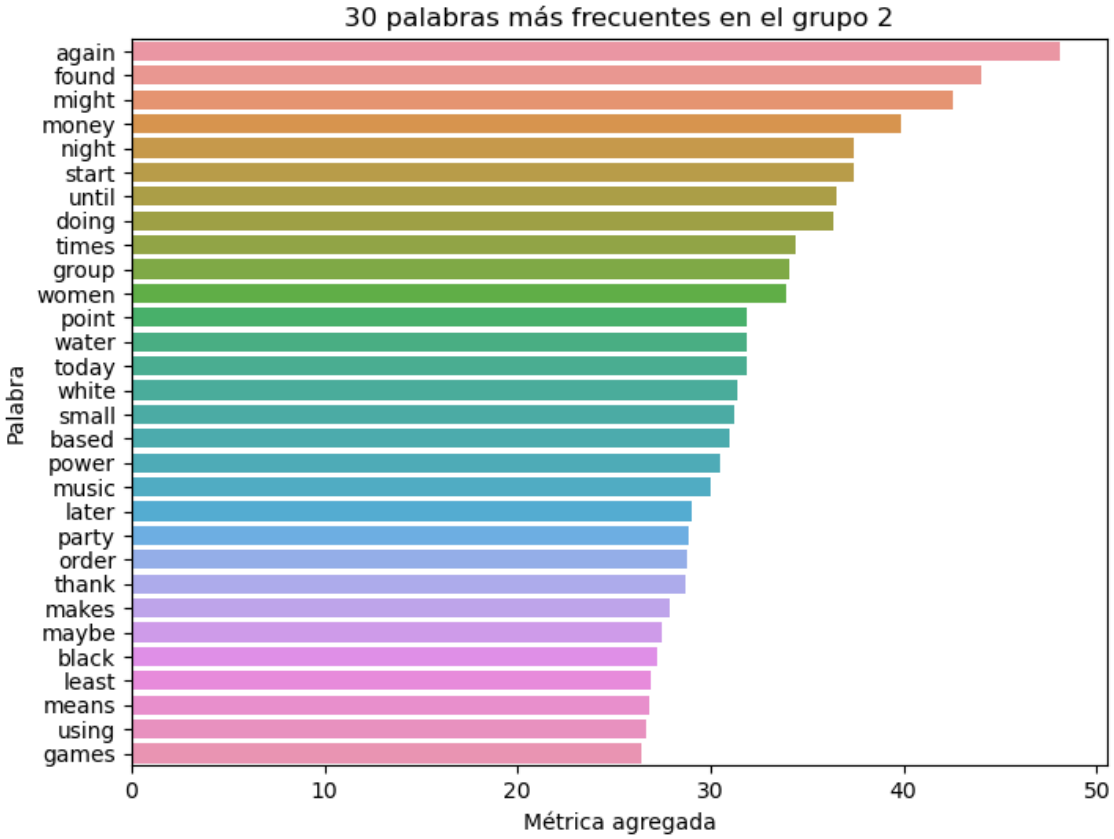


Figura 6.12: 30 palabras más frecuentes en el vocabulario de cada grupo

Capítulo 7

Imitando jugadores

En esta sección implementaremos un algoritmo genético para imitar a cada grupo de jugadores teniendo en cuenta las tablas de frecuencias generadas en el apartado anterior. Definiremos los diversos componentes del algoritmo inspirándonos en la implementación de Mastermind explicada en la sección 3.3. Una vez implementados realizaremos un ajuste de hiperparámetros para encontrar aquellos que modelicen mejor a muestros usuarios y finalmente presentaremos los resultados obtenidos.

7.1. Descripción del modelo

El modelo que vamos a implementar se basa en la ejecución de un algoritmo genético para cada intento. En cada intento se actualizará la función de fitness para tener en cuenta los valores introducidos anteriormente y los códigos de colores obtenidos, esta se basará en un coeficiente de similitud entre las palabras elegidas y las candidatas a serlo. La población inicial de cada intento se irá reduciendo puesto que se eliminarán todas las palabras que no encajen con los patrones obtenidos. La primera iteración se realiza de manera diferente, la elección de la palabra inicial se basará en las tablas de frecuencias de palabras iniciales calculadas en el capítulo anterior y se hará uso del vocabulario inferido a la hora de realizar las funciones de selección, cruce, mutación, permutación e inversión. El modelo finaliza su ejecución cuando se acierta la palabra o cuando se supera el límite de los seis intentos. El procedimiento utilizado queda plasmado en el pseudocódigo 2.

7.2. Descripción de las funciones

En las subsecciones posteriores definimos cada una de las partes del algoritmo implementado:

7.2.1. Representación cromosómica

Cada individuo de la población será una palabra de cinco letras perteneciente al conjunto de palabras admitidas. La selección de la primera palabra introducida

Algoritmo 2: Algoritmo genético para imitar jugadores de Wordle

$attempt \leftarrow 1$;
$X_1 \leftarrow$ Elección por ruleta de la palabra inicial a partir de la tabla de frecuencias;
Evaluar X_1 y obtener el código Y_1;
while $Y_{attempt} \neq$ código verde AND $attempt \leq 6$ **do**
    Generamos $E_{attempt}$ como el conjunto de palabras posibles restantes dado $X_{attempt}$ e $Y_{attempt}$;
    $attempt \leftarrow attempt + 1$;
    $h \leftarrow 1$; Inicializamos el número de generaciones a 1;
    $\hat{E}_h \leftarrow \{\}$; Inicializamos la población vacía;
    Seleccionamos por ruleta individuos pertenecientes a $E_{attempt-1}$ usando la tabla de frecuencias;
    while $h \leq maxgen$ **do**
        Calcular el valor de fitness de los individuos de la población;
        Realizamos el cruce de los individuos con probabilidad p_{cruce};
        Realizamos la mutación con probabilidad $p_{mutacion}$;
        Realizamos la permutación con probabilidad $p_{permutacion}$;
        Realizamos la inversión con probabilidad $p_{inversion}$;
        Guardar los descendientes resultantes en $\hat{E}_h$;
        $h \leftarrow h + 1$;
    end while
    Encontramos el individuo $X_{attempt}$ con mejor fitness, en caso de que existan varios seleccionamos aquel con mayor valor en la tabla de frecuencias;
    Evaluar $X_{attempt}$ y obtener un nuevo $Y_{attempt}$;
end while

se realizará mediante ruleta utilizando los valores obtenidos para cada palabra en el apartado 6.4.1. La inicialización de la primera generación de individuos de cada intento se realizará por ruleta a partir de las palabras posibles restantes teniendo en cuenta los resultados obtenidos en los intentos anteriores, las probabilidades se obtendrán utilizando las tablas de frecuencias generadas en el apartado 6.4.2.

7.2.2. Función de fitness

La función de fitness se actualizará tras cada intento para tener en cuenta los resultados obtenidos. Se basará en la fórmula 7.1, donde w es la palabra sobre la que se evaluará el fitness, g_i corresponde a la palabra introducida en el intento i, c_i indica el número de letras correctas obtenidas en el intento i, m_i indica el número de letras fuera de lugar obtenidas en el intento i, $correct(g_i, w)$ es una función que devuelve el número de letras correctas que se hubieran obtenido introduciendo g_i y teniendo como objetivo la palabra w y $misplaced(g_i, w)$ es una función que devuelve el número de letras fuera de lugar que se hubieran obtenido introduciendo g_i y teniendo como objetivo la palabra w.

$$fitness(w) = \sum_{i=1}^{intentos} (|correct(g_i, w) - c_i|) + |misplaced(g_i, w) - m_i|) \qquad (7.1)$$

Con esta función pretendemos evaluar cuan similar es una palabra a la posible solución de la partida, el mejor valor de fitness se obtienen en el cero pero es posible que exista más de una palabra para que se alcalce en mínimo.

7.2.3. Función de cruce

El cruce de los individuos es una función indispensable en los algoritmos genéticos. Debido a que, en nuestro caso, el cruce de uno o dos puntos de dos palabras no tienen porqué generar dos individuos válidos ya que solamente se admiten palabras existentes y el intercambio de segmentos de letras entre dos individuos no suele generar palabras reales, hemos decidido implementar una función más compleja que garantice la generación de individuos válidos y a su vez que contengan la información de sus progenitores.

Para ello hemos realizado lo siguiente: realizamos un cruce de un punto entre los dos progenitores, en caso de que alguna de las palabras resultantes pertenezcan al conjunto de palabras posibles las mantenemos, en caso contrario, buscamos palabras que contengan las mismas letras que el descendiente y en caso de que haya más de un candidato elegimos aquel con mejor fitness. En caso de que no exista ninguna combinación válida, sustituimos al descendiente por una palabra válida aleatoria. Para acelerar el cálculo de esta función hemos precalculado el conjunto de las palabras válidas que se pueden formar con cada posible combinación de cinco letras.

7.2.4. Función de mutación

Al implementar la función de mutación hemos encontrado con un problema similar al de la función de cruce, cambiar una letra por otra dentro de una palabra no siempre produce un individuo válido, por ello hemos modificado la función para que solamente devuelva individuos válidos. Para realizar fácilmente este proceso hemos generado una tabla que, para cada palabra, contiene el conjunto de palabras que difieren de ella únicamente en una letra. En caso de que al mutar exista más de una opción seleccionamos aquella con el mejor valor de fitness y, en caso de que no haya ninguna, no se realiza la mutación.

7.2.5. Función de permutación

La permutación también ha estado condicionada por la necesidad de generar palabras válidas, por ello la función de permutación únicamente permuta dos letras diferentes siempre y cuando el resultado sea otra palabra. En caso de que existan varias posibilidades siempres se escoge la que tenga un menor valor de fitness. Para

acelerar este procedimiento hemos precalculado para cada palabra el conjunto de individuos que pueden resultar tras permutar dos de sus letras. En caso de que no existiera la posibilidad de permutar, no se aplicaría la transfomación al individuo.

7.2.6. Función de inversión

La inversión de una subcadena de nuestros individuos consite en tomar un segmento de la palabra e invertir el orden de las letras que aparecen. Claramente esto no siempre produce un resultado válido y por tanto nos hemos visto obligados a generar una tabla que contenga todas las posibles inversiones de cada tabla. Por lo tanto, nuestra función dado un individuo selecciona su inversión posible con mayor valor de fitness y, en caso de que no exista ninguna válida, no realiza ningún cambio.

7.3. Ajuste de hiperparámetros

El ajuste de hiperparámetros consiste en encontrar el mejor conjunto de parámetros prefijados de un algoritmo de aprendizaje automático dado. En el contexto de los algoritmos genéticos, los hiperparámetros se refieren a los parámetros que no se aprenden durante el proceso de optimización, sino que se establecen antes de que se ejecute el algoritmo. Estos parámetros tienen un impacto significativo en el rendimiento del algoritmo y, por lo tanto, ajustarlos puede ser fundamental para obtener resultados óptimos [11].

Para aplicarlo a nuestro problema vamos a realizar una búsqueda en cuadrícula, para ello el primer paso es definir el espacio de búsqueda para cada hiperparámetro. Este espacio incluye todos los valores posibles que puede tomar el hiperparámetro. Por ejemplo, el espacio de búsqueda para el tamaño de la población podría definirse como un rango de valores de 50 a 500 individuos. De manera similar, el espacio de búsqueda para la tasa de mutación podría definirse como un conjunto de valores discretos, como 0,01, 0,05 y 0,1.

Una vez que se define el espacio de búsqueda, el algoritmo de búsqueda en cuadrícula evalúa cada combinación de hiperparámetros en el espacio. Este proceso implica ejecutar el modelo para cada combinación de hiperparámetros y evaluar la optimalidad de las soluciones resultantes. A continuación, se selecciona la mejor combinación de hiperparámetros en función del valor de aptitud. [5]

Si bien la búsqueda en cuadrícula puede ser una técnica eficaz para encontrar el conjunto óptimo de hiperparámetros, también puede ser computacionalmente costosa. A medida que aumenta la cantidad de hiperparámetros y el tamaño del espacio de búsqueda, la cantidad de evaluaciones requeridas crece exponencialmente. A pesar de esto, hemos decidido implementarla con los siguientes rangos de valores para cada parámetro:

- **Probabilidad de cruce**: con valores en $[0{,}5, 1]$, con paso de 0,1.

- **Probabilidad de mutación**: con valores en [0,01, 1], con paso de 0,1.
- **Probabilidad de permutación**: con valores en [0,01, 1], con paso de 0,1.
- **Probabilidad de inversión**: con valores en [0,01, 1], con paso de 0,1.
- **Tamaño de la población**: con valores en $\{50, 100, 150, 200\}$

Debido a limitaciones computacionales, las evaluaciones de cada modelo se han realizado sobre un subconjunto representativo de las palabras objetivo del conjunto de datos. Para evaluar el rendimiento de cada modelo se ha utilizado la fórmula 7.2, donde $correct_j^i$, $misplaced_j^i$ y $wrong_i^j$ es la media de aciertos, letras fuera de lugar y fallos respectivamente en el intento i del grupo j y G_i, Y_i y W_i son la media de acietos, letras fuera de lugar y fallos del modelo a evaluar en el intento i. Es decir, se calcula la diferencia en valor absoluto de la media de aciertos, letras fuera de lugar y fallos para cada intento con sus respectivos valores para grupo.

$$performance = \sum_{i=1}^{6}(|correct_i - A_i|) + \sum_{i=1}^{6}(|misplaced_i - Y_i|) + \sum_{i=1}^{6}(|wrong_i - W_i|) \tag{7.2}$$

Tras realizar tres búsquedas en cuadrícula, una para cada grupo, hemos obtenido los valores presentes en las tabla 7.1,

Hiperparámetros con mejor rendimiento para el grupo 0	
Hiperparámetro	Valor
Prob. cruce	0.5
Prob. mutación	0.01
Prob. permutación	0.03
Prob. inversión	0.05
Tam. población	50

Hiperparámetros con mejor rendimiento para el grupo 1	
Hiperparámetro	Valor
Prob. cruce	0.5
Prob. mutación	0.06
Prob. permutación	0.01
Prob. inversión	0.08
Tam. población	100

Hiperparámetros con mejor rendimiento para el grupo 2	
Hiperparámetro	Valor
Prob. cruce	0.5
Prob. mutación	0.06
Prob. permutación	0.01
Prob. inversión	0.05
Tam. población	150

Tabla 7.1: Resultados de la búsqueda en cuadrícula para cada grupo

7.4. Evaluación del rendimiento de cada modelo

Para terminar, hemos evaluado a cada modelo resultante en todas las palabras presentes en el conjunto de datos tres veces y hemos comparado los resultados con los valores obtenidos en la sección 6.4, los resultados los podemos observar en las figuras 7.1, 7.2 y 7.3. Como podemos ver a simple vista, los resultados para los jugadores del grupo 0, son bastante similares con $performance(0) = 2,01$ y con una longitud media de partide de $4,36$. Para el grupo 1 en cambio hemos obtenido $performance(1) = 8,54$ y una longitud de partida de $4,76$, por lo que concluimos que el modelo 1 no es suficientemente similar a los jugadores que trata de imitar, en particular, es un mejor jugador. Para el grupo 2, el modelo ha obtenido $performance(2) = 10,88$ con una longitud media de $3,99$, por lo que, a diferencia del modelo anterior, este es notablemente peor que los jugadores que trata de imitar. Los resultados obtenidos para los grupos 1 y 2 difieren demasiado con los jugadores que tratan de imitar, por ello hemos decidido realizar cambios para mejorar su valor de $performance$.

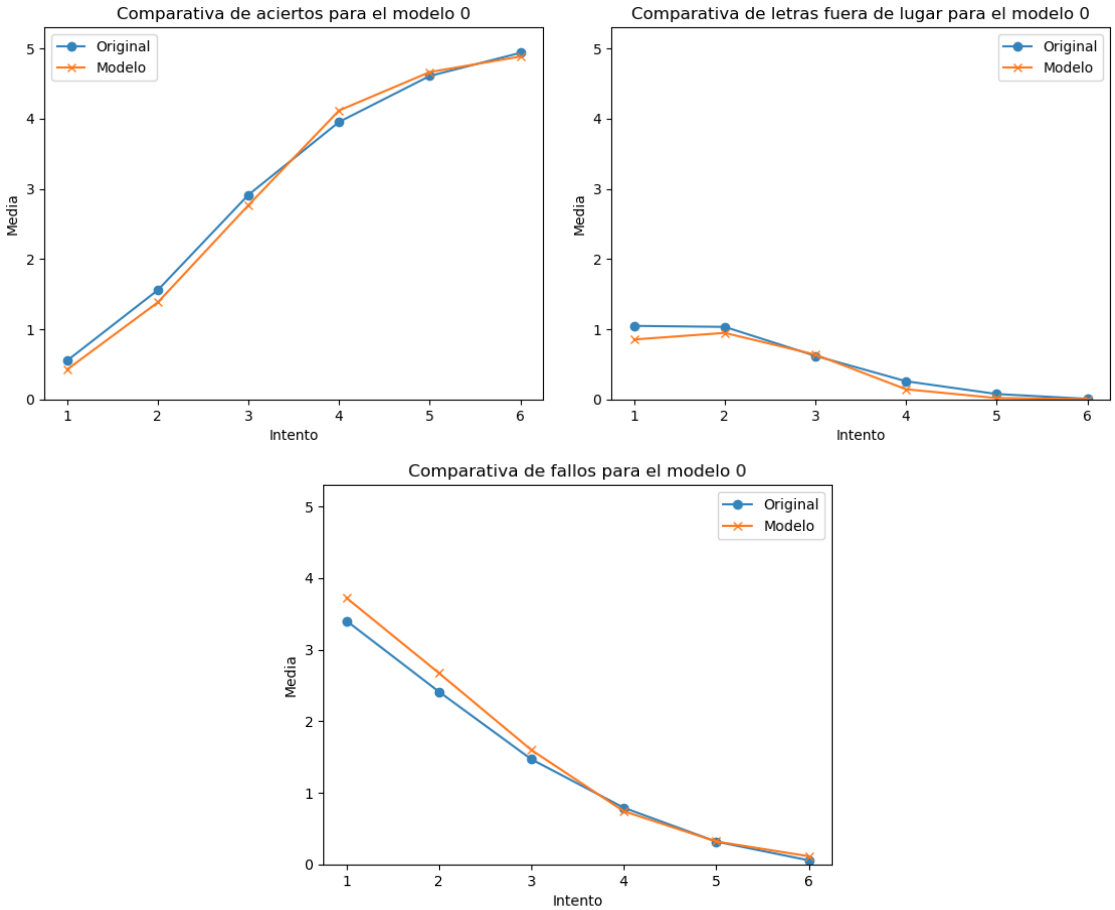


Figura 7.1: Comparación de aciertos, letras fuera de lugar y fallos entre el grupo 0 y el algoritmo que lo modela

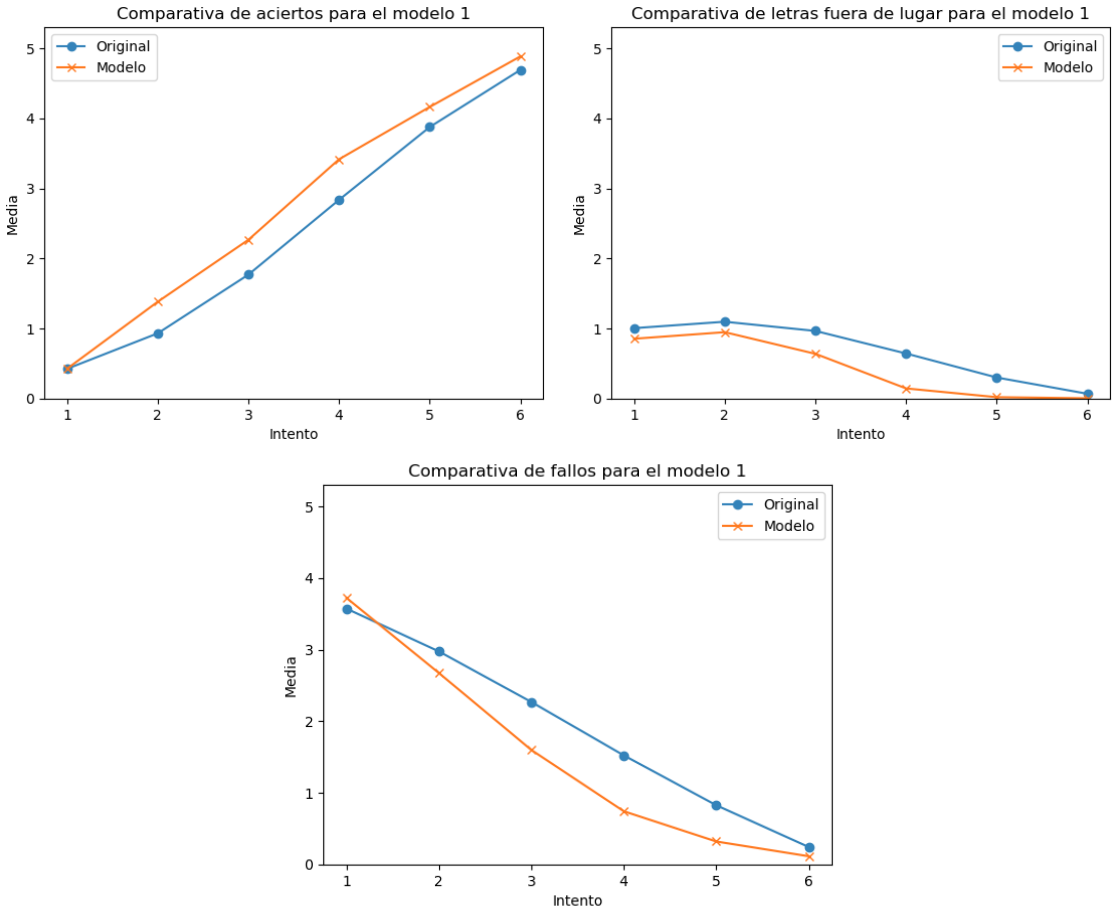


Figura 7.2: Comparación de aciertos, letras fuera de lugar y fallos entre el grupo 1 y el algoritmo que lo modela

7.4.1. Mejora para los grupos 1 y 2

Para mejorar el rendimiento de los modelos de los grupos 1 y 2 hemos procedido a modificar los conjuntos de vocabulario y sus frecuencias, en particular hemos realizado un nuevo ajuste de hiperparámetros teniendo en cuenta un nuevo valor a, cuya función se puede ver reflejada en la nueva defición de la función $value\_vocab(i)$ 7.3 con valores en el conjunto $\{0, 1, 2, 3, 4, 5\}$. En las figuras 7.4 y 7.5 podemos observar la comparativa entre las medias de aciertos, fallos y letras fuera de lugar para cada intento de los modelos con mejor valor de $performance$.

$$value\_vocab(i) = frec\_ini(i) + a \cdot frec\_nat(i) \tag{7.3}$$

En el caso del modelo para el grupo 1 se ha alcanzado un valor de $performance = 4,53$ con una media de intentos de 4,91, este resultado se ha obtenido con el mismo valor para los hiperparámetros que en la sección 7.4 y con un valor $a = 1$. Podemos considerar estos resultados como buenos, aunque no nos debería sorprender la mejora de $performance$ puesto que buscábamos empeorar el rendimiento del modelo respecto del número de intentos.

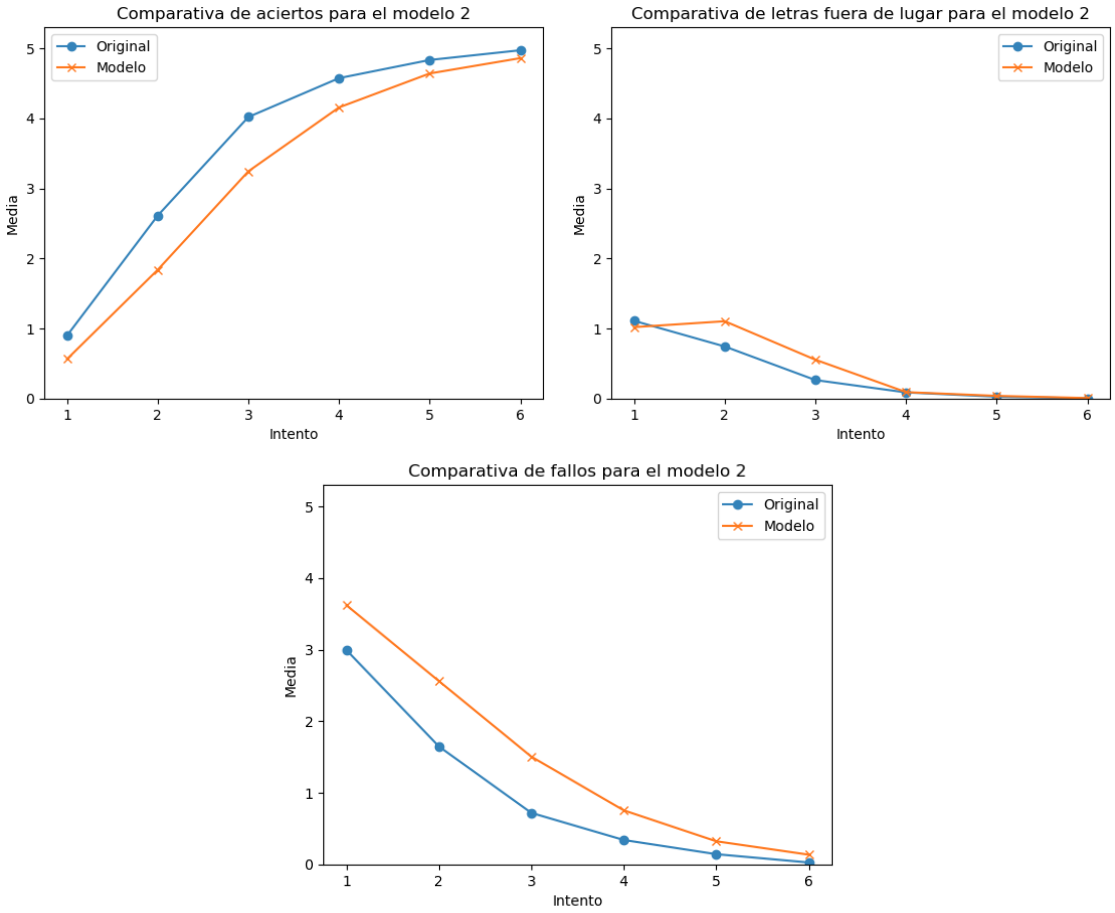


Figura 7.3: Comparación de aciertos, letras fuera de lugar y fallos entre el grupo 2 y el algoritmo que lo modela

En cambio, en el caso del modelo para el grupo 2, no hemos sido capaces de imitar a los jugadores de manera similar a los modelos anteriores, se ha obtenido un valor $performance = 9,21$, el más alto hasta hasta el momento con los mismos hiperparámetros que en la sección 7.4 y con $a = 2$. La longitud de partida media es de $3,75$, más $0,4$ décimas por encima de la media de los jugadores a los que imita. Es posible que para mejorar el rendimiento debamos recurrir a técnicas más sofisticadas como la reducción de la entropía para poder reducir la media de intentos y consecuentemente aproximar mejor las distribuciones de aciertos, fallos y letras fuera de lugar medias por intento.

7.5. Resultados

Para terminar el apartado, hemos recopilado en la tabla 7.2 los valores exactos de cada métrica aplicada a cada uno de los modelos finales generados y su longitud media. Al compararlos con los valores de los jugadores que modelan comprobamos que para los grupos 0 y 1 los resultados son satisfactorios e imitan correctamente

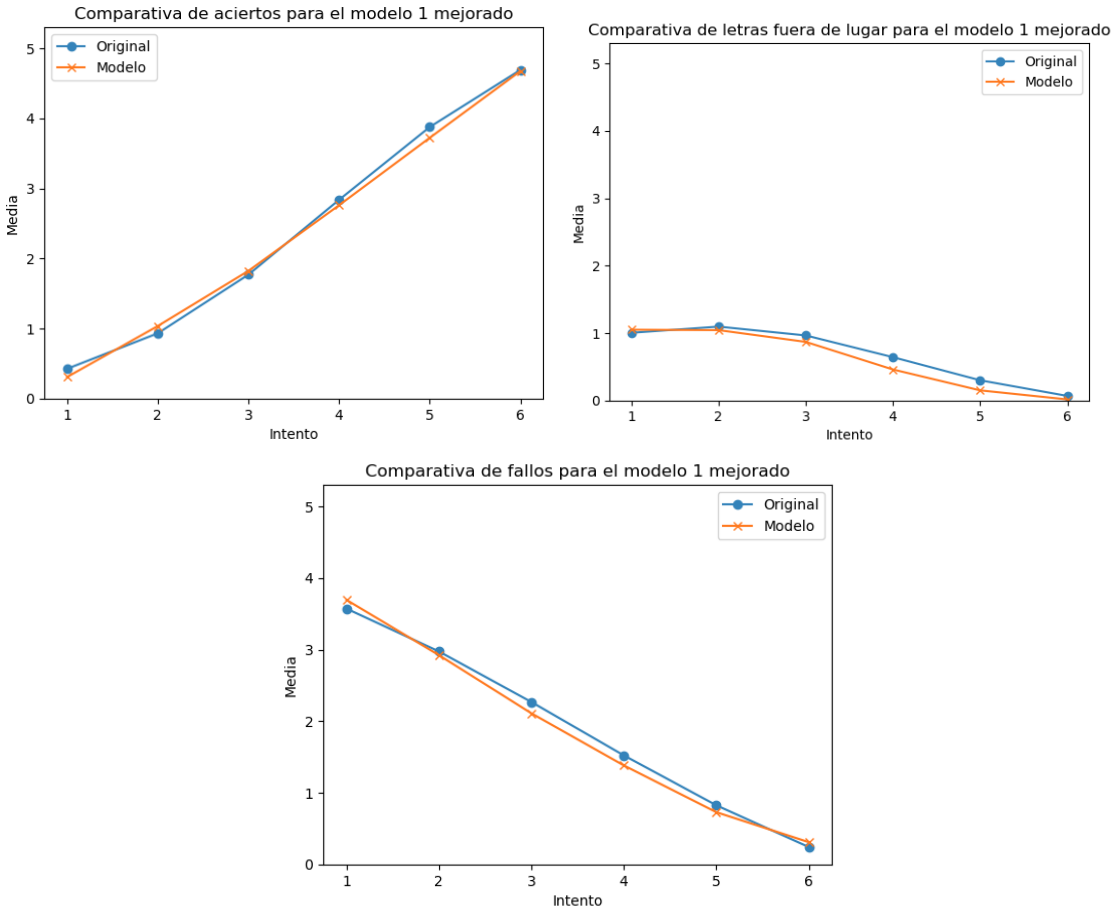


Figura 7.4: Comparación de aciertos, letras fuera de lugar y fallos entre el grupo 1 y el algoritmo que lo modela mejorado

a sus respectivos grupos. Concluimos entonces que el vocabulario de los jugadores de estos dos clústers es un factor determinante en su rendimiento, siendo capaces de imitar el comportamiento humano usando únicamente una aproximación de las frecuencias de aparición de cada palabra en su vocabulario junto con otra aproximación de su frecuencia en el lenguaje natural. No obstante, la importancia de estas aproximaciones en cada modelo es distinta, en particular para el grupo 0 se obtiene un mejor rendimiento cuando el lenguaje natural tiene un peso seis veces mayor que el del vocabulario inferido, en el grupo 1, en cambio, el mejor rendimiento se obtiene cuando ambas aproximaciones tienen el mismo peso.

Para el grupo 2, en cambio, se consigue imitar la evolución de las medias pero siempre varias décimas por debajo del valor obtenido por los jugadores. A pesar de esto, consideramos que el rendimiento obtenido sigue siendo notable ya que es capaz de equiparar en media de intentos a los programas especializados en la reducción de esta que alcanzan una media de 3,42117 intentos [6], que superior a la media del grupo 2. El vocabulario parece seguir siendo un factor altamente relevante puesto que la tabla de frecuencias aproximadas de este grupo era la más diferenciada del resto y disminuyendo la importancia de la frecuencia de las palabras en el lenguaje

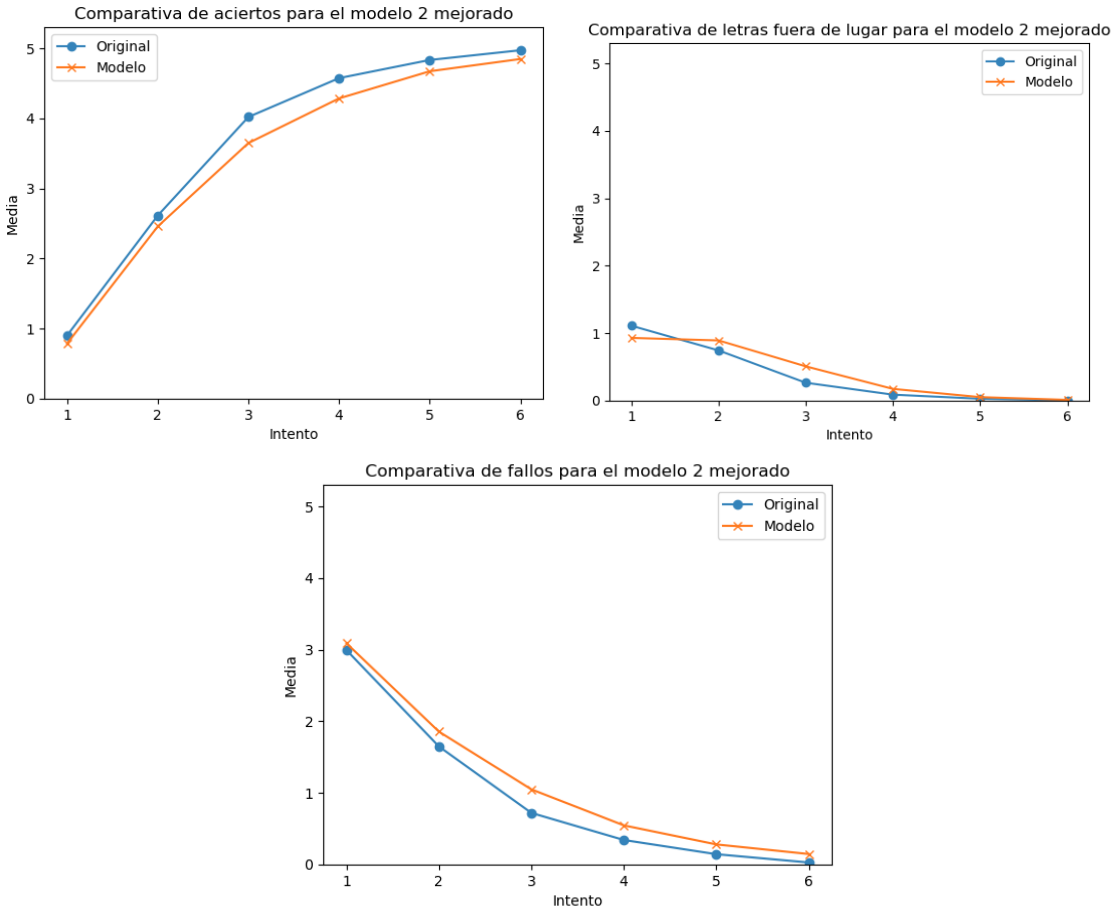


Figura 7.5: Comparación de aciertos, letras fuera de lugar y fallos entre el grupo 2 y el algoritmo que lo modela mejorado

natural en el modelo hemos obtenido la mejora más significativa en el rendimiento de todos los algoritmos revisados en la sección 7.4.1.

Concluimos pues que los mejores jugadores tienen patrones de juego más complejos y más difíciles de predecir que la frecuencia de palabras que utilizan para adivinar la solución consecuentemente, se necesita más investigación para identificar los factores que contribuyen al rendimiento de los mejores jugadores y desarrollar modelos que puedan imitarlos con mayor precisión.

Métricas de los modelos finales						
	Grupo 0	Modelo 0	Grupo 1	Modelo 1	Grupo 2	Modelo 2
Aciertos intento 1	0.55	0.43	0.42	0.31	0.90	0.78
Aciertos intento 2	1.56	1.38	0.93	1.03	2.61	2.46
Aciertos intento 3	2.91	2.77	1.77	1.82	4.02	3.67
Aciertos intento 4	3.95	4.11	2.84	2.76	4.57	4.28
Aciertos intento 5	4.60	4.66	3.86	3.72	4.83	4.67
Aciertos intento 6	4.93	4.88	4.69	4.65	4.97	4.85
Fuera lugar 1	1.05	0.85	1.00	1.05	1.11	0.93
Fuera lugar 2	1.03	0.95	1.10	1.04	0.74	0.89
Fuera lugar 3	0.62	0.64	0.96	0.87	0.26	0.51
Fuera lugar 4	0.26	0.14	0.64	0.46	0.09	0.17
Fuera lugar 5	0.08	0.02	0.30	0.15	0.03	0.05
Fuera lugar 6	0.01	0.00	0.07	0.02	0.00	0.01
Fallos intento 1	3.40	3.72	3.57	3.69	2.99	3.09
Fallos intento 2	2.41	2.67	2.97	2.92	1.65	1.85
Fallos intento 3	1.47	1.60	2.28	2.11	0.72	1.05
Fallos intento 4	0.79	0.74	1.52	1.38	0.34	0.54
Fallos intento 5	0.32	0.32	0.82	0.73	0.14	0.28
Fallos intento 6	0.06	0.11	0.24	0.31	0.03	0.14
Media de intentos	4.23	4.36	5.00	4.92	3.31	3.75

Tabla 7.2: Valores de las métricas de cada modelo y del grupo de jugadores al que imita

Capítulo 8

Conclusiones y trabajo futuro

En este proyecto hemos reunido más de seis millones de partidas de Wordle extraídas de Twitter. Tras realizar una limpieza exhaustiva hemos obtenido un conjunto de datos que hemos hecho público en la plataforma *Kaggle* donde se puede acceder y descargar libremente desde el siguiente `https://www.kaggle.com/datasets/scarcalvetsis/wordle-games`, cumpliendo así el objetivo O1.

Al analizar este conjunto hemos encontrado varios datos relevantes como que la media de intentos se encuentra en 4,19, que el $6,66\,\%$ de las partidas se pierden y que el número de intentos por partida obedece una distribución normal con media 4,19 y desviación típica 1,46. En relación a esto hemos podido comprobar que el tercer intento es el más decisivo de todos puesto que el número de aciertos, fallos y letras fuera de este lugar de un jugador en él es uno de los factores más correlacionados con la longitud de su partida. Esto nos ha llevado a considerarlo relevante en el estudio de la clasificación de jugadores. Finalmente hemos realizado un breve estudio de la dificultad de cada partida basándonos en su longitud y clasificado numéricamente cada una de estas. Hemos podido concluir que las palabras más fáciles son aquellas que contienen las letras más frecuentes en el idioma inglés, es decir que contienen las vocales 'a' y 'e' y consonantes como 'n', 'r' y 't' así como combinaciones consonánticas de estas como 'tr'. Las más difíciles, en cambio, no se caracterizan tan fácilmente, una baja frecuencia en el lenguaje usual parece ser el factor más decisivo lo cual es razonable ya que el desconocimiento de la palabra por parte del jugador elimina cualquier posibilidad de ganar la partida, sin embargo, también encontramos factores como la presencia de dos o más letras repetidas así como la falta de las vocales 'a' y 'e' a favor de la presencia de 'y' en la posición final. Este análisis nos ha permitido cumplir el objetivo O2.

Seguidamente hemos utilizado el análisis de los datos para generar variables para caracterizar los jugadores. Mediante el uso de clústering hemos encontrado 3 grupos diferenciados caracterizados por lo siguiente:

- **Grupo 0**: Se caracteriza por el grupo más numeroso compuesto por más de la mitad de los jugadores, su longtitud media de partida es de 4,23 y representan a los jugadores medios. Analizando las distribuciones de las medias de aciertos,

fallos y letras fuera de lugar podemos comprobar que estos jugadores suelen acertar una media de 0,5 letras en su intento inicial y suelen obtener también una letra en la posición equivocada. A partir del segundo intento sus aciertos aumentan en una tasa del 1,4 de media y suelen conseguir tres aciertos en el tercer intento. Su estrategia se basa en reducir el número de palabras posibles en los dos primeros intentos, en el tercero colocar en la posición correcta las letras fuera de lugar obtenidas en los intentos anteriores y en los intentos posteriores introducen palabras que se adapten al patrón de aciertos.

- **Grupo 1**: Se trata del segundo grupo más numeroso de los tres encontrados, su longitud media de partida es de 4,99 y representan a los jugadores con peor rendimiento y posiblemente con menos experiencia. Tras analizar sus distribuciones concluimos que en su primer intento no suelen obtener ningún acierto pero sí una única letra fuera de lugar, además su segundo intento se caracteriza por presentar resultados muy similares al primero lo cual es indicador de que usan palabras parecidas o con letras en común. Esto hace que en su tercer intento tengan una media inferior a 2 aciertos lo cual dificulta sus posibilidades de ganar la partida. En particular estos jugadores no siempre obtienen un acierto nuevo en cada intento pero sí letras fuera de lugar. Concluimos que utilizan palabras muy similares entre sí, posiblemente por limitaciones en el vocabulario y que consecuentemente son el grupo con más partidas perdidas.

- **Grupo 2**: Se trata del grupo minoritario con una longitud media de partida de $3,31$ intentos. Represenan a los jugadores con mejor rendimiento y posiblemente expertos. Se caracterizan por obtener un acierto y una letra fuera de lugar en el primer intento y en una gran capacidad de aumentar los aciertos muy rápidamente entre los intento 1 y 3. En el tercer intento suelen haber ganado o han conseguido 4 aciertos de media lo cual reduce sustancialmente el conjunto de palabras posibles. Su estrategia se basa en utilizar los dos primeros intentos para reducir al máximo el conjunto de palabras posibles, lo consiguen utilizando palabras con las mismas letras en la posición correcta que en su primer y segundo intento pero variando las restantes denotando un gran dominio del vocabulario. A continuación, si no han ganado en el tercer intento, realizan intentos adaptados al patrón obtenido que es altamente restrictivo y por ello requieren de menos intentos que el grupo 0.

Los resultados obtenidos nos han hecho considerar el estudio del vocabulario de los jugadores, así como su palabra preferente en el primer intento. Al no poseer las palabras introducidas por cada jugador si no el patrón devuelto, hemos recurrido a herramientas de inferencia para generar los diferentes conjuntos de frecuencias para cada tipo de jugador y cumplir así con los objetivos O3 y O8.

Usando estos datos junto a los patrones extraídos de cada jugador, desarrollamos varios modelos que trataban de imitar el comportamiento de cada grupo utilizando algoritmos genéticos con una función de fitness basada en la similitud entre patrones obtenidos en cada intento. Tras realizar los ajustes necesarios a los hiperparámetros de los modelos y de realizar un estudio del rendimiento dependiendo del peso dado

a la frecuencia de las palabras en el lenguaje, obtuvimos tres modelos finales consiguiendo cumplir los objetivos O4 y O5. Cada uno de ellos nos aportó información adicional sobre los grupos, en particular el modelo obtenido para el grupo 0 se ajusta notablemente bien al patrón de juego observado, este modelo da una importancia seis veces superior a la frecuencia de las palabras en el lenguaje usual que a la obtenida mediante la inferencia de los patrones. Esto indica una fuerte relación entre el comportamiento de este grupo y el lenguaje habitual, lo cual no resulta sorprendente ya que se trata de los jugadores medios. Para el modelo del grupo 1 también se obtuvieron resultados notablemente certeros aunque en este caso las frecuencias en el lenguaje pasaron a tener la misma importancia que las frecuencias inferidas, esto sugiere que los jugadores del grupo 2 tienen un vocabulario más reducido que el resto de grupos y que esto es un factor decisivo en su rendimiento en el juego. Finalmente en modelo del grupo 2 es el que presenta el peor rendimiento respecto a las métricas utilizadas para compararlo con los jugadores reales. Pese a haber conseguido imitar la distribución de aciertos, fallos y letras fuera de lugar, el modelo no consigue alcanzar el rendimiento de los jugadores reales. Fue entrenado dando el doble de peso al lenguaje natural que a las frecuencias inferidas y, pese a esto, los resultados sugieren que existen otros aspectos comportamiento de los jugadores del grupo 2 que aún no se comprenden por completo y que no están relacionados con el vocabulario de estos. Este análisis nos permite dar respuesta a los objetivos O6 y O7 de nuestro proyecto.

En general, este trabajo nos ha brindado información valiosa sobre el comportamiento de los jugadores de Wordle y nos ha ayudado a crear modelos más precisos que pueden imitar su juego. Si bien aún queda mucho por aprender sobre las complejidades del comportamiento de los jugadores, este proyecto sentó las bases para futuras investigaciones en esta área y destacó el potencial del uso de modelos computacionales para comprender mejor el comportamiento humano.

8.1. Trabajo futuro

En vistas a un proyecto futuro, es evidente que se requiere un análisis más profundo de las estrategias del grupo 2 y cómo estas afectan al rendimiento de los jugadores y su capacidad para reducir tan drásticamente el espacio de soluciones, posiblemente utilizando técnicas más sofisticadas como la reducción de la entropía o el uso de palabras iniciales concretas. También sería conveniente tratar de mejorar las aproximaciones del vocabulario de cada grupo, ya sea con técnicas de inferencia más sofisticadas o mediante la recopilación directa de los intentos de los usuarios. Otro factor a tener en consideración es la mejoría de los jugadores con el tiempo y tratar de observar fluctuaciones de usuarios entre grupos. También sería interesante investigar cómo los modelos obtenidos pueden ayudar a mejorar a los jugadores actuales, ya sea utilizándolos para predecir la palabra que deberían introducir, sugerírsela y posiblemente ampliar su vocabulario, explicarles la estrategia que están siguiendo e intentar mejorarla o incluso valorar la idoneidad de las palabras introducidas en cada intento.

Como se podrá apreciar, existe una amplia variedad de posibles continuaciones del trabajo comenzado en este proyecto y esperamos que tanto este documento como el conjunto de datos que lo acompaña sean útiles a todas las personas que quieran realizar aportaciones.

Conclusions and Future Work

In this project we have collected more than six million Wordle matches extracted from Twitter. After an exhaustive cleaning we have obtained a dataset that we have made public on the platform *Kaggle* where it can be freely accessed and downloaded from the following `https://www.kaggle.com/datasets/scarcalvetsis/wordle-games`, thus fulfilling our first objective.

When analyzing this set we have found several relevant data such as that the mean number of attempts is 4.19, that 6.66% of the games are lost and that the number of attempts per game obeys a normal distribution with mean 4.19 and standard deviation 1.46. In relation to this we have been able to verify that the third try is the most decisive of all since the number of hits, misses and letters out of this place of a player in it is one of the factors most correlated with the length of his game. This has led us to consider it relevant in the study of player ranking. Finally, we have made a brief study of the difficulty of each game based on its length and numerically ranked each of them. We have been able to conclude that the easiest words are those that contain the most frequent letters in the English language, i.e. that contain the vowels 'a' and 'e' and consonants such as 'n', 'r' and 't' as well as consonant combinations of these such as 'tr'. The more difficult ones, on the other hand, are not so easily characterized, a low frequency in the usual language seems to be the most decisive factor which is reasonable since the player's ignorance of the word eliminates any chance of winning the game, however, we also found factors such as the presence of two or more repeated letters as well as the lack of the vowels 'a' and 'e' in favor of the presence of 'y' in the final position. This analysis has allowed us to fulfill our second objective.

We then used data analysis to generate variables to characterize the players. Through the use of clustering we have found 3 differentiated groups characterized by the following:

- **Grup 0**: It is characterized by the largest group composed of more than half of the players, their average starting latitude is 4.23 and they represent the average players. Analyzing the distributions of the averages of hits, misses and misplaced letters we can see that these players usually hit an average of 0.5 letters in their initial attempt and usually also get a letter in the wrong position. From the second try onwards their hits increase at a rate of 1.4 on

average and they usually get three hits on the third try. Their strategy is based on reducing the number of possible words in the first two attempts, in the third attempt they place in the correct position the misplaced letters obtained in the previous attempts and in the subsequent attempts they introduce words that fit the pattern of hits.

- **Grup 1**: This is the second largest group of the three found, their average game length is 4.99 and they represent the players with the worst performance and possibly the least experience. After analyzing their distributions we conclude that in their first attempt they usually do not get any hits but only one letter out of place, and their second attempt is characterized by very similar results to the first one, which is an indicator that they use similar words or words with letters in common. This means that in their third attempt they have an average of less than 2 hits, which hinders their chances of winning the game. In particular, these players do not always get a new hit on each try but they do get misplaced letters. We conclude that they use words very similar to each other, possibly because of vocabulary limitations, and that consequently they are the group with more lost games.

- **Grup 2**: This is the minority group with an average starting length of 3.31 attempts. They represent the best performing and possibly expert players. They are characterized by getting one hit and one letter out of place on the first try and a great ability to increase hits very quickly between tries 1 and 3. By the third try they usually have won or have gotten 4 hits on average which substantially reduces the set of possible words. Their strategy is based on using the first two attempts to reduce the set of possible words as much as possible, they achieve this by using words with the same letters in the correct position as in their first and second attempts but varying the remaining letters, denoting a great mastery of vocabulary. Then, if they have not won on the third attempt, they make attempts adapted to the pattern obtained, which is highly restrictive and therefore requires fewer attempts than group 0.

The results obtained made us consider the study of the vocabulary of the players, as well as their preferred word in the first attempt. Since we do not possess the words entered by each player but the returned pattern, we have resorted to inference tools to generate the different sets of frequencies for each type of player and thus fulfill our third and eighth objectives.

Using these data along with the patterns extracted from each player, we developed several models that attempted to mimic the behavior of each group using genetic algorithms with a fitness function based on the similarity between patterns obtained at each attempt. After making the necessary adjustments to the hyperparameters of the models and performing a study of the performance depending on the weight given to the frequency of words in the language, we obtained three final models achieving our fourth and fifth objectives. Each of them provided us with additional information about the groups, in particular the model obtained for group 0 fits remarkably well to the observed pattern of play, this model gives six

times more importance to the frequency of words in the formal language than that obtained by pattern inference. This indicates a strong relationship between the behavior of this group and the usual language, which is not surprising since these are average players. For the group 1 model we also obtained remarkably accurate results, although in this case the language frequencies became as important as the inferred frequencies, suggesting that group 2 players have a smaller vocabulary than the other groups and that this is a decisive factor in their performance in the game. Finally, the model of group 2 is the one that presents the worst performance with respect to the metrics used to compare it with the real players. Despite having managed to mimic the distribution of hits, misses and misplaced letters, the model fails to achieve the performance of real players. It was trained by giving twice as much weight to natural language as to inferred frequencies and, despite this, the results suggest that there are other aspects of the behavior of group 2 players that are not yet fully understood and are not related to their vocabulary. This analysis allows us to answer objectives 6 and 7 of our project.

Overall, this work has provided us with valuable information about the behavior of Wordle players and has helped us to create more accurate models that can mimic their play. While there is still much to learn about the complexities of player behavior, this project laid the groundwork for future research in this area and highlighted the potential of using computational models to better understand human behavior.

8.2. Future work

In view of a future project, it is clear that further analysis of group 2 strategies and how these affect player performance and their ability to reduce the solution space so drastically is required, possibly using more sophisticated techniques such as entropy reduction or the use of specific initial words. It would also be worthwhile to try to improve the vocabulary approximations of each group, either with more sophisticated inference techniques or by direct collection of user attempts. Another factor to take into consideration is the improvement of players over time and to try to observe fluctuations of users between groups. It would also be interesting to investigate how the models obtained can help improve current players, either by using them to predict the word they should enter, suggest it to them and possibly expand their vocabulary, explain the strategy they are following and try to improve it, or even assess the appropriateness of the words entered in each attempt.

As will be seen, there is a wide variety of possible continuations of the work begun in this project and we hope that both this document and the accompanying data set will be useful to all who wish to contribute.

Bibliografía

[1] Today's wordle answer: All words for 2023 (updated daily). `https://screenrant.com/wordle-answers-updated-word-puzzle-guide/`. Accessed: 2023-02-28.

[2] Alexis Benveniste and Jackie Frere. A collection of the best wordle tips and tricks, February 2022.

[3] Angel Benítez and Enrique Cavanillas. Wordle solving algorithms using information theory. September 2022.

[4] Lotte Berghman, Dries Goossens, and Roel Leus. Efficient solutions for mastermind using genetic algorithms. *Computers Operations Research*, 36(6):1880–1885, 2009.

[5] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[6] D. Bertsimas and A. Paskov. An Exact and Interpretable Solution to Wordle. *MIT Sloan School journal*, pages 59–70, 09 2022.

[7] Cédric Buche, Cindy Even, and Julien Soler. Autonomous virtual player in a video game imitating human players: The orion framework. pages 108–113, 10 2018.

[8] Steffi Cao and Ikran Dahir. How a group of twitter colleagues blew up wordle. January 2022.

[9] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, sep 2001.

[10] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023.

[11] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning, 2015.

[12] Adam Coates and Andrew Y. Ng. *Learning Feature Representations with K-Means*, pages 561–580. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[13] Kenneth A. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.

[14] Gabriel V. de la Cruz Jr au2, Yunshu Du, and Matthew E. Taylor. Pre-training neural networks with human demonstrations for deep reinforcement learning, 2019.

[15] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 29, New York, NY, USA, 2004. Association for Computing Machinery.

[16] Dibdin Emma. When it comes to wordle strategies, it's personal, February 2022.

[17] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models, 2016.

[18] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition (2nd Ed.)*. Academic Press Professional, Inc., USA, 1990.

[19] David E. Goldberg, Kalyanmoy Deb, and Hillol Kargupta. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[21] Rachel Hall. Wordle creator overwhelmed by global success of hit puzzle. January 2022.

[22] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[23] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957.

[24] Ian Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016.

[25] Josh Katz. Wordlebot: Your daily, personalized wordle score.

[26] John F Kenney. *Mathematics of statistics*. D. Van Nostrand, 1939.

[27] Donald Knuth. The computer as master mind. *J. Recreational Mathematics*, 9(1), 1976.

[28] Jonathan Lee. The new york times is finally making changes to wordle. November 2022.

[29] Raquel Magalhães. What wordle can teach us about gamification in marketing. February 2022.

[30] R. B. Marimont and M. B. Shapiro. Nearest Neighbour Searches and the Curse of Dimensionality. *IMA Journal of Applied Mathematics*, 24(1):59–70, 08 1979.

[31] James McQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967*, pages 281–297, 1967.

[32] Maximiliano Miranda, Antonio Sánchez-Ruiz, and Federico Peinado. Building non-player character behaviors by imitation using interactive case-based reasoning. 10 2020.

[33] Maximiliano Miranda, Antonio A. Sánchez-Ruiz, and Federico Peinado. Interactive explainable case-based reasoning for behavior modelling in videogames. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1263–1270, 2021.

[34] Melanie Mitchell, Stephanie Forrest, and John H. Holland. Introduction to the special issue on genetic algorithms: New theoretical and empirical results. *Artificial Intelligence Journal*, 1992.

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[36] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–33, 02 2015.

[37] Atsushi Nakano, Akihito Tanaka, and Jun'ichi Hoshino. Imitating the behavior of human players in action games. pages 332–335, 01 2006.

[38] Atsushi Nakano, Akihito Tanaka, and Junichi Hoshino. Imitating the behavior of human players in action games. In Richard Harper, Matthias Rauterberg, and Marco Combetto, editors, *Entertainment Computing - ICEC 2006*, pages 332–335, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[39] Juan Ortega, Noor Shaker, Julian Togelius, and Georgios Yannakakis. Imitating human playing styles in super mario bros. *Entertainment Computing*, 4, 01 2012.

[40] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. November 1901.

[41] Johannes Pfau, Jan David Smeddinck, and Rainer Malaka. The case for usable ai: What industry professionals make of academic ai in video games. In *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '20, page 330–334, New York, NY, USA, 2020. Association for Computing Machinery.

[42] Yu Popkov. Randomization and entropy in machine learning and data processing. *Doklady Mathematics*, 105:135–157, 06 2022.

[43] M Somerville. valid-wordle-words.txt. `https://gist.github.com/dracos/dd0668f281e685bad51479e5acaadb93`, 2021.

[44] Inc Twitter. Rules and filtering: Standard v1.1, February 2022.

[45] Diego Unzueta. Information theory applied to wordle, February 2022.

[46] Daniel Victor. Wordle is a love story, January 2022.

[47] Ruohan Wang, Carlo Ciliberto, Pierluigi Amadori, and Yiannis Demiris. Random expert distillation: Imitation learning via expert policy support estimation, 2019.

[48] Christopher Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 05 1992.

[49] Hao Wu, Yueli Li, Xiaohan Bi, Linna Zhang, Rongfang Bie, and Yingzhuo Wang. Joint entropy based learning model for image retrieval. *Journal of Visual Communication and Image Representation*, 55:415–423, 2018.

[50] Boming Xia, Xiaozhen Ye, and Adnan Abuassba. Recent research on ai in games. pages 505–510, 06 2020.

[51] Georgios N. Yannakakis and Julian Togelius. *Modeling Players*, pages 203–255. Springer International Publishing, Cham, 2018.

[52] Rong Zhou, Zhisheng Zhang, Kunyyu Peng, Yang Mi, and Xiangsheng Huang. Humanoid action imitation learning via boosting sample dqn in virtual demonstrator environment. pages 1–9, 11 2016.