Grado en Estadística
Trabajo de Fin de Grado

El ajedrez desde la minería de datos

Francisco José Molina Alonso

UNIVERSIDAD DE GRANADA

Departamento de Estadística e Investigación Operativa

Universidad de Granada

Tutor

Carlos J. Mantas Ruiz

Granada, Mes de Año

Grado en Estadística
Trabajo de Fin de Grado

El ajedrez desde la minería de datos

UNIVERSIDAD
DE GRANADA

Declaro explícitamente que el trabajo presentado es original, entendido en el sentido de que no he utilizado fuentes sin citarlas debidamente.

Francisco José Molina Alonso

Autoevaluación

Antes de elegir un TFG sentía que tenía ganas de hacer algo práctico, algo que me enseñara conocimientos que pudiera poner en práctica en el día de mañana. Puedo decir orgullosamente mientras escribo estas palabras que ese objetivo se ha cumplido. Entre todo lo que he aprendido no solamente puedo destacar qué es la minería de datos junto con algunas técnicas, sino también que he comenzado a aprender por mi propia cuenta programación en lenguaje Python (si aparece código en algún lugar de este trabajo, es porque lo que se está haciendo es con dicho lenguaje). También me siento bastante animado a seguir aprendiendo sobre minería de datos y Python, no solamente porque me han parecido temas muy interesantes y que he aplicado de manera exitosa conocimientos aprendidos durante el grado, sino porque van a ser importantes en el día de mañana.

Y también me gustaría aprovechar estas líneas para darle las gracias a mi familia por todo el apoyo recibido durante tantos años. También las gracias a mi tutor por el apoyo a lo largo de todos estos meses en el desarrollo de este trabajo.

Resumen

A día de hoy debido al desarrollo tecnológico se aplican diversas técnicas de minería de datos e inteligencia artificial a incontables partidas de ajedrez con el objetivo de extraer conocimiento que pueda servir para mejorar en ajedrez. Este va a ser el objetivo de este trabajo: ver qué condiciones tanto para las blancas como para las negras dan la victoria.

Partiendo de una base de datos de más de medio millón de partidas de ajedrez, primero se van a generar variables sobre las partidas y luego se le va a aplicar un análisis exploratorio para así posteriormente hacer un análisis estadístico básico. Una vez hecho esto, se va a continuar con la aplicación de técnicas de minería de datos explicadas en el capítulo dos (árboles de decisión, reglas de asociación y regresión logística) para así extraer conocimiento de ella. Cuando ya se hayan aplicado estas técnicas, se va a hacer un análisis conjunto priorizando qué han extraído en común.

Summary

As of today, due to technological development and advances, various data mining and artificial intelligence techniques are applied to many chess games with the aim of extracting knowledge that can help improve chess skills. This will be the objective of this work: to see what conditions lead to victory.

Starting with a database of more than half a million chess games, variables about the games will be generated, followed by an exploratory analysis, and then a basic statistical analysis. Once this is done, the application of data mining techniques explained in chapter two will continue in order to extract knowledge from it. After applying these techniques, a joint analysis will be conducted, prioritizing what these techniques have extracted in common. In the end, the final chapter will proceed with the conclusions of the work.

Índice general

1. Introducción **11**

2. Conceptos importantes **15**

2.1. Conocimientos de ajedrez 15

2.1.1. Sistema ELO 15

2.1.2. Modos de juego 17

2.1.3. ¿Qué es una apertura de ajedrez? 18

2.1.4. Plataformas online 19

2.1.5. Formato .PGN para bases de datos de partidas de ajedrez 21

2.2. Minería de datos 22

2.2.1. Árboles de decisión 22

2.2.2. Reglas de asociación 27

2.2.3. Discretización de variables 30

2.2.4. Regresión logística 30

2.2.5. Métodos de selección de variables 35

2.2.6. Librerías utilizadas en R 36

2.2.7. Librerías utilizadas en Python 36

3. Preparación de la base de datos **39**

3.1. Descripción de la base de datos 39

3.2. Transformación de variables 41

4. Análisis exploratorio de datos 43

4.1. Tratamiento de datos faltantes 43

4.2. Valores repetidos 44

4.3. Identificación de ruido 44

4.3.1. Errores tipográficos 44

4.3.2. Valores atípicos 45

4.4. Eliminación de variables innecesarias 48

5. Análisis estadístico 51

5.1. Rated Blitz Game 51

5.1.1. Análisis estadístico del elo 51

5.1.2. Variables sobre el número de veces que se ha movido cada pieza . . 53

5.1.3. Correlación entre todas las variables 55

5.1.4. Correlación diferencia de elo y probabilidad de ganar 56

5.2. Rated Rapid game 57

5.2.1. Análisis estadístico del elo 57

5.2.2. Variables sobre el número de veces que se ha movido cada pieza . . 58

5.2.3. Correlación entre todas las variables 60

5.2.4. Correlación diferencia de elo y probabilidad de ganar 60

5.3. Rated Bullet Game 61

5.3.1. Análisis estadístico del elo 61

5.3.2. Variables sobre el número de veces que se ha movido cada pieza . . 62

5.3.3. Correlación entre todas las variables 63

5.3.4. Correlación diferencia de elo y probabilidad de ganar 64

6. Minería de datos 65

6.1. Árboles de decisión 65
6.1.1. Rated Blitz Game 66
6.1.2. Rated Rapid Game 69
6.1.3. Rated Bullet Game 71
6.2. Reglas de asociación 76
6.2.1. Rated Blitz Game 76
6.2.2. Rated Rapid Game 82
6.2.3. Rated Bullet Game 86
6.3. Regresión logística 89
6.3.1. Rated Blitz Game 90
6.3.2. Rated Rapid Game 91
6.3.3. Rated Bullet Game 93

7. Análisis conjunto **95**
7.1. Rated Blitz Game 95
7.2. Rated Rapid Game 96
7.3. Rated Bullet Game 98

8. Conclusiones finales **99**

Capítulo 1

Introducción

El ajedrez en sus inicios era muy distinto a como lo conocemos a día de hoy. Aparecido en la India septentrional en el siglo VII d.C., se le conocía por diversos nombres siendo uno de los más populares *chaturanga*. Este juego tan antiguo tenía unas reglas bastante distintas a las que tiene lo que conocemos como ajedrez a día de hoy, como que no existía el enroque del rey o que los peones (conocidos como *Bhata* o *Padati*) no podían avanzar dos casillas al comienzo de la partida [**1**].

Figura 1.1: Tablero de una partida de chaturanga

Esta forma de ajedrez primitivo fue evolucionando a lo largo de la historia hasta el mítico juego que conocemos a día de hoy, siendo un punto importante cuando los árabes entre los años 632 y 651 conquistaron el imperio Sasánida. Durante estos años tuvieron contacto con el ajedrez desarrollándolo y apareciendo ajedrecistas de élite dentro del mundo árabe como *al-Adli* (ca. 800-870) que compuso el primer manual de ajedrez.

Con el avance de los siglos el juego llegó a Europa por diversas fuentes, como a través de los árabes a España en el siglo IX. Este país ha sido muy importante en la historia del ajedrez puesto que a lo largo del siglo XV se desarrollaron las reglas que dieron forma al ajedrez moderno (por ejemplo, la dama, antes llamada alferza, avanzaba únicamente una casilla y en dirección diagonal y el peón no podía dar dos pasos en el primer movimiento), reglas que se introdujeron en Valencia entre los años 1470-1490 y aparecieron en el poema valenciano *Scachs d'amor* , el documento más antiguo sobre el ajedrez moderno [**2**].

Es de destacar que el primer torneo internacional de la historia del ajedrez fue en 1575 organizado por la corte de Felipe II, donde el jugador Ruy López de Segura, considerado el mejor del mundo hasta entonces, fue derrotado por el calabrés Giovanni Leonardo da Cutro [**3**]. Como dato curioso, existe una apertura de ajedrez considerada como una de las más sólidas entre los mejores ajedrecistas actuales del mundo y es la apertura española, conocida también como Ruy López en honor a Ruy López de Segura que fue un gran jugador de ella. Es una apertura bastante sólida porque desde un primer momento se ejerce una fuerte presión sobre el caballo desarrollado de las negras y deja lista a las blancas para enrocar en su siguiente turno, dejando al rey rápidamente a salvo [**4**].

Figura 1.2: Apertura española, también conocida como Ruy López

En los siguientes siglos el ajedrez cogió bastante importancia, especialmente en la segunda mitad del siglo XX. En esta época se empezaron a desarrollar los primeros algoritmos computacionales que podían jugar al ajedrez, siendo el primero de ellos creado por el famoso matemático Alan Turing en 1950. Es interesante mencionar que estas máquinas no podían jugar al ajedrez en un principio a un nivel superior a un ser humano hasta la famosa partida ocurrida en 1996 donde el superordenador de IBM, *Deep Blue*, derrotó al entonces campeón del mundo y a día de hoy uno de los mejores jugadores de la historia, Garry Kaspárov. Esto marcó el comienzo de una era de dominancia de los ordenadores en el ajedrez [**5**].

Figura 1.3: Garry Kaspárov contra Deep Blue

¿Por qué hacer este trabajo?

Se ha mencionado antes que los ordenadores a día de hoy tienen una dominancia sobre los seres humanos a la hora de jugar al ajedrez y que ni siquiera los mejores del mundo pueden ganarles y esto es debido al avance tecnológico de las últimas décadas que ha permitido desarrollar esta clase de máquinas. Pero no hay que destacar solamente el avance de la potencia de cálculo de los ordenadores, sino también el desarrollo de diversas técnicas estadísticas y de inteligencia artifical que se han aplicado a multitud de posiciones de partidas de ajedrez para sacar patrones y conocimiento de ellas que pueda servir para mejorar en el juego y sacar una ventaja al oponente. Por poner algún ejemplo, se utilizan

redes neuronales para encontrar las mejores jugadas dada una posición de ajedrez. El objetivo aquí entonces va a ser aplicar los recientes avances en algoritmos de minería de datos para intentar extraer conocimiento de la partida.

En este trabajo se va a comenzar en el capítulo 2 explicando conceptos importantes relativos al ajedrez y de minería de datos. Se van a ver algunas técnicas, como los árboles de decisión, regresión logística y reglas de asociación. Se va a continuar con el capítulo 3 creando una base de datos funcional que aloje más de medio millón de partidas de ajedrez. Los capítulos 4 y 5 van destinados a un análisis exploratorio y estadístico de la base de datos que serán pasos previos necesarios para desarrollar los capítulos 6 y 7 que serán análisis de la base de datos utilizando las técnicas explicadas en el capítulo 2. Para terminar, el capítulo 8 serán las conclusiones finales de todo lo que se ha aplicado.

Capítulo 2

Conceptos importantes

2.1. Conocimientos de ajedrez

2.1.1. Sistema ELO

El sistema Elo es un método estadístico para calcular el resultado probable de un jugador frente a otro en juegos de suma-cero como el ajedrez. Es un sistema llamado como su creador, Arpad Elo, un húngaro-americano profesor de física
. Aunque el sistema Elo originalmente se hizo como una mejoría de un sistema anterior empleado en ajedrez llamado Harkness, se utiliza en otros deportes como el béisbol, el fútbol americano o el billar.El Elo de un jugador de ajedrez es una puntuación en forma de número que cambia dependiendo de las partidas clasificatorias que el jugador juegue. Después de cada partida, el jugador que gana toma puntos del jugador que pierde en una cantidad que depende de la diferencia de Elo al comienzo de la partida. Si el jugador con más elo gana la partida, conseguirá menos puntos que si ganara el jugador con una menor puntuación de Elo. [**6**]

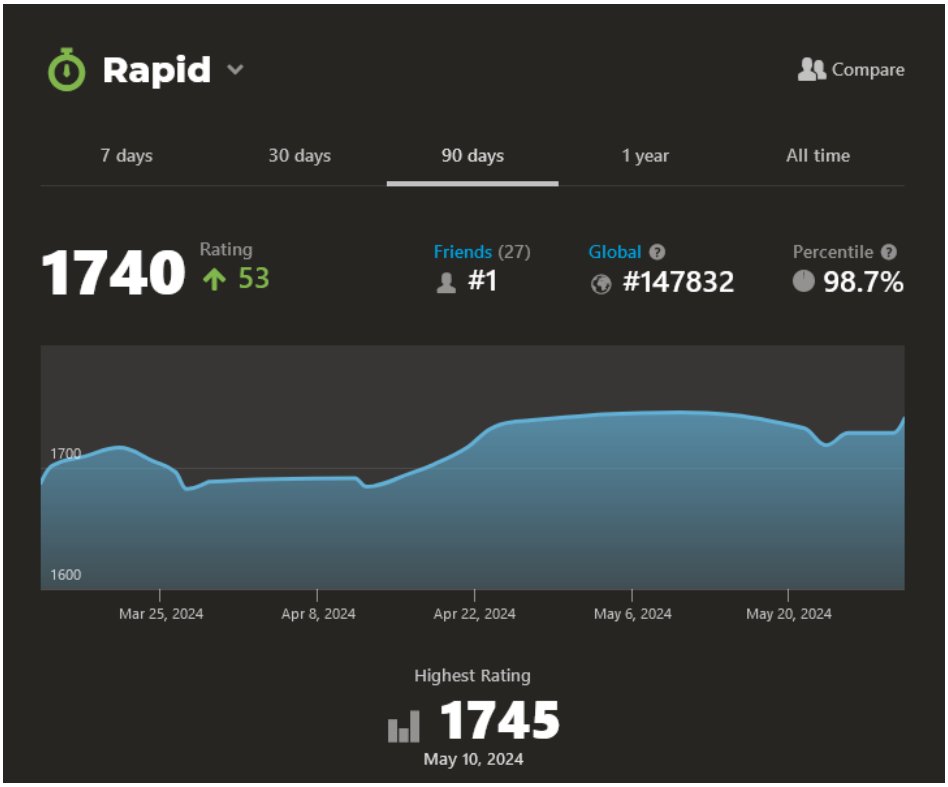

Figura 2.1: Evolución del elo a lo largo del tiempo de un jugador en chess.com

Aunque es una creencia generalizada que el sistema de puntuación Elo mide la fuerza absoluta de un ajedrecista, en realidad no es así. Lo que calcula es el resultado probable del enfrentamiento entre un jugador y otro. Por norma general, se espera que un jugador con una puntuación Elo 100 puntos superior a la de su oponente gane alrededor de 5 partidas de 8 contra ese adversario (64 %). Por otro lado, un ajedrecista de 200 puntos Elo de ventaja sobre su rival debería ganar un 75 % de ellas. [**7**]

¿Por qué es importante el sistema de puntuación Elo?

Todas las federaciones y plataformas de ajedrez importantes de todo el mundo utilizan el sistema de puntuación Elo o una variante del mismo (como el Glicko). Además, el sistema de puntuación Elo está basado en un modelo estadístico que tiene en cuenta únicamente el resultado de las partidas disputadas, lo que se considera una estimación bastante precisa al no estar basado en modelos arbitrarios o subjetivos a la hora de

determinar la fuerza de un jugador. Aunque un ajedrecista hiciera "'el más espectacular de los sacrificios"' no se tendría en cuenta en el cálculo del elo a no ser que ganase la partida.

Aún así, el propio Arpad Elo reconoció que resulta prácticamente imposible determinar la fuerza exacta de un ajedrecista. En uno de sus artículos, expone este hecho de la siguiente forma: "La medición de la puntuación Elo de un individuo podría compararse con la estimación de la posición de un corcho que se mueve hacia arriba y hacia abajo sobre la superficie del agua agitada con un palo atado a una cuerda y que se balancea con el viento. "[**8**]

2.1.2. Modos de juego

Aunque las reglas sean las mismas, existen distintos modos de de juego para jugar una partida de ajedrez que afectan al tiempo total que tiene un jugador en toda la partida. Dado que el trabajo se ha hecho con una base de datos de Lichess, se va a usar la definición de cada modo de juego que utiliza esta plataforma en la tabla 2.1 [**9**]:

Nombre	**Tiempo mínimo**	**Tiempo máximo**
Bullet	15 segundos	59 segundos
Blitz	1 minuto	6 minutos 59 segundos
Rapid	7 minutos	20 minutos
Classical	25 minutos	180 minutos
Correspondence	24 horas (por jugada)	Sin límite (por jugada)

Tabla 2.1: Tabla de tiempos en diferentes modalidades de ajedrez

Hay que añadir que el último modo de juego se diferencia del resto en que no es en tiempo real, los jugadores pueden hacer su jugada cuando quieran mientras sea dentro de un intervalo de tiempo muy grande. Antes existía una forma de jugar al ajedrez a distancia en la que los jugadores se mandaban cartas el uno al otro con las jugadas que hacían y de ahí que este modo de juego se llame Correspondence.

2.1.3. ¿Qué es una apertura de ajedrez?

En ajedrez se denomina apertura a la fase inicial del juego, en la que se desarrollan las piezas desde sus posiciones iniciales. Existen tres fases en una partida de ajedrez que son: apertura, medio juego y final. Las aperturas reciben diversos nombres como defensa siciliana, sistema Londres o apertura imperial. Existen no docenas sino cientos de aperturas diferentes que varían desde buscar un juego posicional hasta una búsqueda de tácticas salvaje.

La fase de apertura es muy importante porque el medio juego se va a derivar de ella y si en la fase de apertura un jugador ha cometido errores va a comenzar el medio juego en una posición de desventaja frente a su adversario.

A día de hoy los mejores jugadores del mundo estudian a fondo la teoría de aperturas utilizando los motores de ajedrez más potentes para terminar la primera fase de la partida en una posición conocida y así tenerlo más fácil sacarle ventaja al rival en la segunda fase de la partida [**10**].

Un ejemplo de una apertura de ajedrez puede ser la famosa defensa siciliana (vista en la figura 2.2) que se da cuando las blancas después de jugar e4 las negras responden con c5.

Figura 2.2: La defensa siciliana

El objetivo de la defensa siciliana es crear un desequilibrio en el tablero, en vez de responder con un peón en e5 al responder de esta manera lo que buscan las negras es crear una asimetría controlando la importante casilla central de d4. Es una apertura compleja muy empleada por los gran maestros cuando no quieren empatar con negras sino ganar la partida.

2.1.4. Plataformas online

Con el desarrollo tecnológico el ajedrez se ha vuelto bastante más fuerte en los últimos 20 años. Antes para jugar una partida un ajedrecista necesitaba encontrar a un oponente en persona pero con el avance de internet es muy fácil y rápido encontrar un rival contra el que jugar.

A día de hoy, el juego online de ajedrez está dominado por dos grandes plataformas online: Chess.com y Lichess. A continuación se va a hablar un poco de Lichess por su importancia en este trabajo.

Lichess es un servidor de ajedrez en internet, gratuito y de código abierto que está gestionado por una organización sin ánimo de lucro del mismo nombre [**11**] La plataforma fue fundada en 2010 por un programador francés llamado Thibault Duplessis. El software y diseño que utiliza Lichess es en su mayoría de código abierto bajo licencia AGPL.

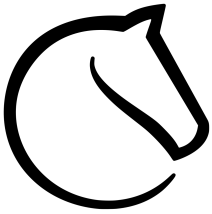

Figura 2.3: Logo de Lichess.org

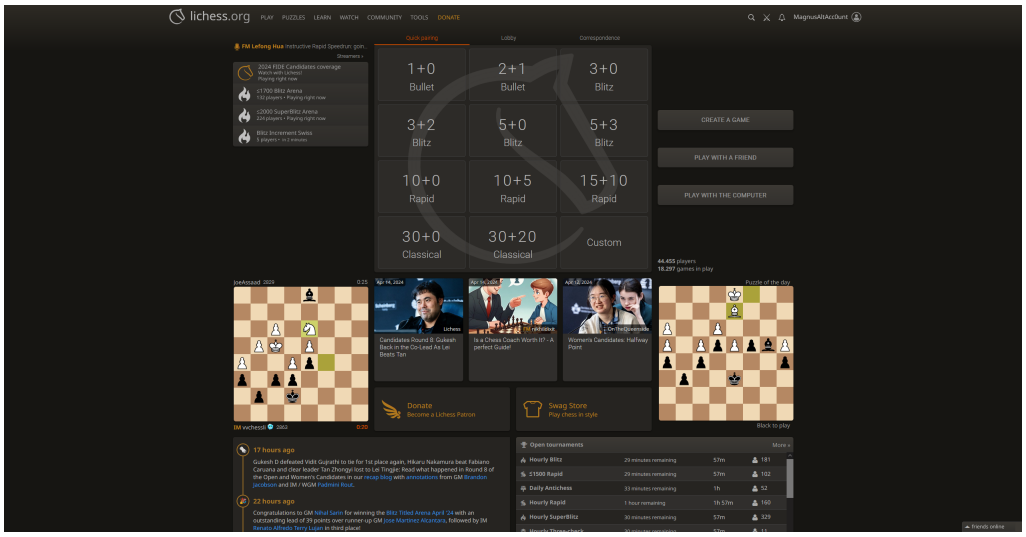

Figura 2.4: Página principal de Lichess

Aparte de poder jugar en los distintos modos de ajedrez hablados anteriormente, Lichess.org dispone de otras interesantes funciones como un repertorio de aperturas para poder aprenderlas, puzzles de ajedrez en los que practicar tácticas o incluso variantes del propio ajedrez donde las reglas del juego cambian.

2.1.5. Formato .PGN para bases de datos de partidas de ajedrez

El formato PGN (Portable Game Notation) es un estándar ampliamente utilizado para representar partidas de ajedrez en formato de texto plano. Es un formato flexible y legible por humanos que permite almacenar de manera concisa la información clave de una partida de ajedrez, incluyendo los movimientos de las piezas, los nombres de los jugadores, el resultado de la partida y comentarios adicionales.

Una partida de ajedrez en formato PGN consta de varias secciones, que pueden incluir:

- Cabecera (Header): La cabecera contiene metadatos sobre la partida, como los nombres de los jugadores, la fecha, el lugar, el resultado de la partida, el tiempo de control, el Elo de los jugadores, entre otros. Estos metadatos se almacenan en pares de etiquetas y valores, como [Event "FIDE World Championship"], [Date "2022.11.12"], [White Çarlsen, Magnus"], [Black "Nepomniachtchi, Ian"], etc.

- Movimientos (Moves): La sección de movimientos contiene la secuencia de movimientos de la partida en notación algebraica estándar. Cada movimiento se registra como una pareja de letras y números que representan la pieza que se mueve y su destino, por ejemplo, e4, Nf3, exd5, O-O, etc. Los movimientos se pueden registrar junto con información adicional, como comentarios ({...}), variantes ((...)), resultados de análisis de computadora ($...$) y anotaciones simbólicas de la partida (!, ?, !!, ?!, etc.).

- Resultado (Result): La partida termina con un resultado que indica el desenlace de la partida. Los resultados comunes incluyen 1-0 para victoria de las blancas, 0-1 para victoria de las negras, 1/2-1/2 para tablas y * para una partida sin resultado conocido.

El formato PGN es ampliamente compatible y puede ser utilizado por la mayoría de los programas de ajedrez y bases de datos de partidas. Es fácil de leer y escribir para los

humanos, lo que lo hace ideal para compartir y archivar partidas de ajedrez. Además, es lo suficientemente flexible como para admitir variaciones y extensiones, lo que lo convierte en un formato versátil para el almacenamiento de datos de ajedrez.

2.2. Minería de datos

La minería de datos es un proceso de descubrir patrones e información valiosa dentro de grandes volúmenes de datos. Considerando la evolución de la tecnología de depósito de datos y el crecimiento del Big Data, la minería de datos ha evolucionado muy rápidamente en las últimas dos décadas y es muy empleada por las empresas para transformar sus datos en conocimiento útil.

La toma de decisiones en una organización ha experimentado una mejora gracias a la minería de datos, la cual se vale del análisis de la información. Las técnicas empleadas en este proceso se dividen principalmente en dos categorías: aquellas que describen el conjunto de datos y las que pueden prever resultados mediante el uso de algoritmos de machine learning. Estos métodos son fundamentales para organizar y filtrar datos, desvelando información de gran interés que abarca desde la detección de fraudes hasta los comportamientos de los usuarios, identificación de cuellos de botella e incluso la detección de brechas de seguridad.

Cuando se combinan con herramientas de visualización y analítica de datos, estas técnicas permiten sumergirse en la minería de datos con una facilidad que antes no existía, extrayendo información importante a velocidades nunca antes vistas. Los avances en el ámbito de la inteligencia artificial están impulsando la rápida adopción de estas técnicas en diversos sectores. [**12**]

2.2.1. Árboles de decisión

Un árbol de decisión es un modelo de aprendizaje automático utilizado en estadística, minería de datos y machine learning. Un árbol de decisión es utilizado como un modelo

predictivo para predecir los valores que puede tomar una variable objetivo. Es interesante mencionar que los árboles de decisión se utilizan en muchos ámbitos como la inteligencia artificial y la economía.

Existen distintos tipos de árboles de decisión, que van a depender en parte del tipo de variable que se quiere predecir [**13**].

Elementos de un árbol de decisión

- Nodos. Cada nodo se define como el momento en el que se debe de tomar una decisión entre varias posibles, lo que implica que a medida que aumenta el número de nodos se incrementa también el número de posibles finales.
- Vectores de números. Solución final a la que se llega en función de las distintas opciones. Representan utilidades.
- Flechas. Uniones entre un nodo y otro que representan las acciones a tomar.
- Etiquetas. Están en cada nodo y flecha. Especifican la acción que se va a tomar.

Conceptos ligados

- Sobreajuste. Ocurre cuando el modelo se ajusta demasiado a los datos y al ruido de los propios datos de entrenamiento produciendo un desempeño muy bueno en datos de entrenamiento pero muy pobre en datos nuevos o no vistos.

- Poda (Prunning). La poda consiste en eliminar una rama de un nodo transformándolo en una hoja (terminal), asignándole la clasificación más común de los ejemplos de formación considerados en ese nodo.

- Validación cruzada. se realiza dividiendo el conjunto de datos en múltiples subconjuntos. Luego se entrena el modelo en varios de estos subconjuntos y se evalúa en el restante. Este proceso se repite varias veces, de modo que cada subconjunto sirve tanto como conjunto de entrenamiento como de conjunto de prueba en diferentes etapas del proceso. La forma más común de validación cruzada es la validación cruzada k-fold, donde el conjunto de datos se divide en k subconjunto.

- Coeficiente kappa. Se calcula con la siguiente fórmula [**14**]:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

  Donde P_o es la proporción de acuerdo observada entre las clasificaciones del árbol de decisión y las clasificaciones reales en el conjunto de datos de prueba (proporción en la que el árbol clasifica correctamente los datos) y P_e la proporción que cabría de esperar por pula aleatoriedad entre las reales y el árbol de decisión por pura aleatoriedad.

  El coeficiente Kappa sirve para comparar P_o y P_e y normaliza la diferencia para que se encuentre en el intervalo (-1,1). Un coeficiente de Kappa cercano a 1 indica

un acuerdo fuerte entre las clasificaciones del árbol y las reales, mientras que uno cercano al 0 una clasificación bastante aleatoria. En cambio, un coeficiente negativo significa que el árbol hace clasificaciones peores a las esperadas por puro azar.

Tipos de árboles de decisión

- Árboles de clasificación. Se dan cuando el valor que se predice toma valores discretos.
- Árboles de regresión. Este tipo de análisis se da cuando el valor que se predice es un número real.

Algunas técnicas para elaborar árboles de decisión [13]

- Algoritmo ID3 (Iterative Dichotomiser 3). Es uno de los primeros algoritmos de árbol de decisión. Funciona dividiendo el conjunto de datos en función de la característica que proporciona la mayor ganancia de información en cada nivel del árbol.

- CART. Es un algoritmo que puede ser utilizado tanto para problemas de clasificación como de regresión. Utiliza medidas como la impureza Gini para tomar decisiones sobre cómo dividir los datos en cada nodo del árbol.

- Random Forest. Técnica de conjunto que combina múltiples árboles de decisión entrenados en diferentes subconjuntos de datos y características. Luego, combina las predicciones de estos árboles para obtener una predicción final más robusta y precisa.

Métricas

Los algoritmos para construir árboles de decisión usualmente funcionan de arriba abajo, utilizando una variable para cada caso que mejor divide el conjunto de datos. Todas estas métricas se aplican a cada subconjunto de datos y los valores resultantes se combinan para proporcionar una medida de la calidad de la división.

A continuación se van a explicar dos de las métricas más populares que se emplean a día de hoy.

- Impuridad gini

  También conocido como índice Gini-Simpson, es llamado así por el matemático italiano Corrado Gini y es ampliamente empleado en los árboles de decisión. El objetivo del índice Gini es medir con qué frecuencia un elemento elegido de manera aleatoria de un conjunto estaría incorrectamente clasificado si estuviera agrupado de manera aleatoria e independiente en base a la distribución de valores a predecir en el conjunto de datos. Si da el valor cero (que es el mínimo) significaría que los valores solamente pueden tomar un valor [**14**].

  Sea un conjunto de datos con J clases y frecuencias relativas $p_i, i \in 1, 2, \ldots, J$, la probabilidad de elegir un objeto i es p_i y la probabilidad de categorizar mal a dicho objeto es

$$\sum_{k \neq i} p_k = 1 - p_i$$

Fórmula 2.5: Probabilidad de elegir un objeto i

  El índice Gini es calculado sumando los productos de estas probabilidades para cada clase:

$$\mathrm{I}_G(p) = \sum_{i=1}^{J}\left(p_i \sum_{k\neq i} p_k\right) = \sum_{i=1}^{J} p_i(1-p_i) = \sum_{i=1}^{J}(p_i - p_i^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} p_i^2 = 1 - \sum_{i=1}^{J} p_i^2.$$

Fórmula 2.6: Fórmula índice Gini

- Reducción de varianza

  La reducción de varianza se emplea frecuentemente cuando la variable que se quiere predecir es continua (árbol de regresión). La reducción de varianza se aplica a cada nodo N y se define como el total de la reducción de la varianza de la variable objetivo Y como consecuencia de dividir este nodo [**15**]:

$$I_V(N) = \frac{1}{|S|^2}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}(y_i - y_j)^2 - \left(\frac{|S_t|^2}{|S|^2}\frac{1}{|S_t|^2}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}(y_i - y_j)^2 + \frac{|S_f|^2}{|S|^2}\frac{1}{|S_f|^2}\sum_{i\in S_f}\sum_{j\in S_f}\frac{1}{2}(y_i - y_j)^2\right)$$

Fórmula 2.7: Fórmula Reducción de Varianza

Donde S, St y Sf son conjuntos antes de dividir las muestras.

2.2.2. Reglas de asociación

Las reglas de asociación son declaraciones tipo *if-then* que muestran las probabilidades de las relaciones entre transacciones de objetos en grandes bases de datos. En un nivel básico, las reglas de asociación en minería de datos implican el uso de modelos de machine learning para analizar los datos en busca de patrones. Encuentra asociaciones *if-then* que son las conocidas como reglas de asociación.

Por ejemplo, si el 80 % de la gente que compra tableros de ajedrez también son buenos jugando, está claro que existe un patrón, una asociación entre comprar un tablero de ajedrez y ser bueno jugando.

Distintos algoritmos utilizan reglas de asociación para descubrir patrones dentro de los conjuntos de datos. Es interesante comentar que la inteligencia artificial también utiliza

las reglas de asociación para generar asociaciones en bases de datos [**16**].

¿Cómo funcionan las reglas de asociación?

Una regla de asociación tiene dos partes: un antecedente (*if*) y un consecuente (*then*) por sus nombres en inglés. Un antecedente es un objeto encontrado en los datos mientras que un consecuente es una combinación encontrada en combinación por el antecedente. Las declaraciones *if-then* forman conjuntos de objetos que son las bases para calcular las reglas de asociación.

Los estadísticos buscan declaraciones *if-then* que ocurren con frecuencia. Luego buscan por el soporte en términos de con qué frecuencia ocurren esas reglas y la confianza de con qué frecuencia se cumplen.

Las reglas de asociación son típicamente creadas de conjuntos de objetos que incluyen muchas entradas y que están bien representados.

Parámetro Lift y Cobertura

- El parámetro Lift mide la importancia de una regla de asociación, comparando la probabilidad de que ambos valores (antecedente y consecuente) vayan juntos junto con la probabilidad de que sean independientes
- La Cobertura es la proporción de transacciones que tienen el antecedente de la regla

¿Qué es el soporte y confianza en minería de datos?

Las reglas de asociación son creadas buscando en los datos frecuentes declaraciones *if-then*. El soporte indica con qué frecuencia dicha regla aparece en los datos mientras que la confianza con qué frecuencia dicha regla se cumple [**18**].

Dos pasos están involucrados en generar las reglas de asociación:

- Identificación de las declaraciones *if-then* que con mayor frecuencia aparecen en un conjunto de datos. Dado el número de resultados que obtengamos, ajustar el soporte mínimo.

- Establecer mínimos de confianza para los resultados obtenidos. Por ejemplo, si dos objetos en el conjunto de datos están entrelazados más de la mitad de las veces, podría ser una asociación interesante que se podría estudiar más a fondo.

Algoritmo a priori

El algoritmo *a priori* es un algoritmo empleado en minería de datos sobre bases de datos transaccionales que sirven para encontrar de forma eficiente conjuntos de ítems frecuentes para generar reglas de asociación. Comienza buscando ítems individuales en una parte de la base de datos y continúa extendiéndolos en conjuntos de mayor tamaño siempre que se mantengan las reglas de soporte y confianza preestablecidas.

El algoritmo *a priori*, propuesto en la década de los 90 por Agrawal y Srikant, es empleado para analizar bases de datos que contienen transacciones, como registros de compras de consumidores o incluso partidas de ajedrez. Cada transacción se interpreta como un conjunto de elementos. Con un umbral definido como C, apriori identifica todos los conjuntos de elementos que son subconjuntos de al menos C transacciones en la base de datos. Utiliza un enfoque de abajo a arriba, expandiendo gradualmente conjuntos frecuentes añadiendo un elemento a la vez en lo que se conoce como generación de

candidatos, y luego verifica estos grupos contra los datos. El algoritmo finaliza cuando no puede encontrar más ampliaciones con éxito de los conjuntos previos. *A priori* utiliza una búsqueda en anchura para almacenar eficientemente los conjuntos de elementos candidatos. Genera conjuntos de candidatos de tamaño k a partir de conjuntos de tamaño k-1, y luego elimina los candidatos que contienen un subpatrón poco frecuente. Según el lema de clausura hacia abajo, el conjunto candidato contiene todos los conjuntos frecuentes de tamaño k. Después de este proceso, escanea la base de datos para identificar los conjuntos de elementos frecuentes entre los candidatos.

2.2.3. Discretización de variables

La discretización de variables es un proceso para convertir las variables continuas en variables discretas o categóricas. Esto implica dividir el rango de valores continuos en un número finito de intervalos o categorías, asignándose a cada intervalo finito o categoría su valor correspondiente de la variable que se está discretizando. Se transforma así los datos originales a una forma más manejable y apta para ciertos tipos de análisis. Existen distintos motivos por los que se discretizan las variables, como reducir el ruido en conjuntos de datos grandes o para aplicar modelos basados en categorías, como las reglas de asociación.

Existen distintos métodos para discretizar variables, como el método de *agrupación por igual frecuencia* que va a ser empleado más tarde. Este método consiste en agrupar los valores en intervalos que tienen el mismo número de observaciones. Por ejemplo, si tenemos 200 observaciones y queremos 20 intervalos, cada intervalo tendrá 10 observaciones. Una ventaja interesante a destacar de este método es que es más fácil manejar distribuciones asimétricas porque los datos podrían quedar distribuidos más equitativamente [**19**].

2.2.4. Regresión logística

La regresión logística es una técnica de análisis de datos que utiliza las matemáticas para identificar las relaciones entre dos o más variables. Luego, usa esta relación para predecir el valor de una variable basándose en las otras. Generalmente, la predicción tiene

un número limitado de resultados, como sí o no.

Por ejemplo, imagine que quiere prever si un visitante de su sitio web hará clic en el botón de pago del carrito de compras. La regresión logística analiza el comportamiento de visitantes anteriores, considerando factores como el tiempo que pasan en el sitio y la cantidad de artículos en el carrito. Puede determinar que los visitantes que permanecen más de cinco minutos en el sitio y agregan más de tres artículos al carrito suelen hacer clic en el botón de pago. Con esta información, la regresión logística puede predecir el comportamiento de un nuevo visitante en el sitio web.

¿Qué beneficios aporta la regresión logística?

La regresión logística es crucial en la inteligencia artificial y el aprendizaje automático. Los modelos de ML son programas de software que pueden entrenarse para realizar tareas complejas de procesamiento de datos sin que un ser humano tenga que intervenir. Los modelos de ML basados en regresión logística permiten a las organizaciones extraer información útil de sus datos empresariales, lo que facilita el análisis predictivo para reducir costos, aumentar la eficiencia y escalar más rápidamente. Por ejemplo, las empresas pueden identificar patrones que mejoren la retención de empleados o lleven a un diseño de productos más rentable [**20**].

Algunos beneficios del uso de la regresión logística son:

- Simplicidad

  Los modelos de regresión logística son matemáticamente menos complejos que otros métodos de ML. Por lo tanto, pueden implementarse incluso si el equipo no tiene una profunda experiencia en ML.

- Velocidad

  Los modelos de regresión logística pueden procesar grandes volúmenes de datos a una velocidad muy alta debido a su menor demanda de recursos computacionales. Esto los hace ideales para organizaciones que están comenzando con proyectos de

ML y desean obtener resultados más rápidos.

- Visibilidad

  El análisis de regresión logística proporciona a los desarrolladores una mayor visibilidad de los procesos internos del software en comparación con otras técnicas. La solución de problemas y la corrección de errores son más fáciles debido a la menor complejidad de las matemáticas empleadas.

Implementación matemática

La regresión logística analiza los datos distribuidos binomialmente como:

$$Y_i \sim B(p_i, n_i), \text{ para } i = 1, \ldots, m,$$

Fórmula 2.8: Distribución binomial

Donde Y_i en la fórmula 2.8. son ensayos de Bernoulli donde los n_i son conocidos y las probabilidades de éxito p_i son desconocidas.

Luego el modelo se obtiene a base de que cada ensayo i y el conjunto de variables explicativas/independientes puedan informar sobre la probabilidad final. Sea X_i un vector k-dimensional en la fórmula 2.9

$$p_i = \mathrm{E}\left(\frac{Y_i}{n_i}\middle| X_i\right).$$

Fórmula 2.9: Definición de los p_i

Los logaritmos de la razón de momios de las probabilidades binomiales desconocidas se modelan como sigue en la fórmula 2.10:

$$\operatorname{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}.$$

Fórmula 2.10: Cálculo de los logits

Siendo estimados los parámetros β por máxima verosimilitud.

El modelo se puede formular de manera equivalente como en 2.11.:

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i})}}$$

Fórmula 2.11: Cálculo de los p_i

El gráfico de la función logística que se muestra en la figura 2.12 tiene como variable independiente la combinación lineal $\beta_0 + \beta_1 x$ y la variable dependiente es la probabilidad estimada $p_i(x)$.

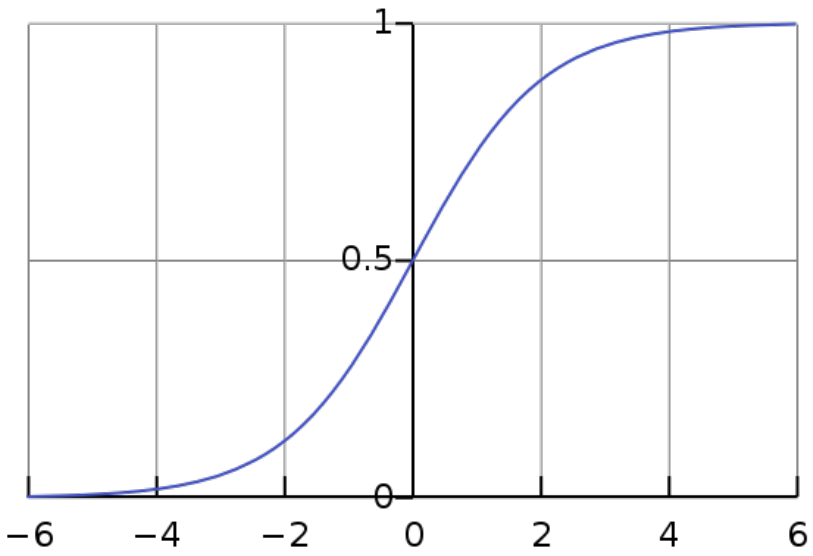

Figura 2.12: Gráfico de la función logística

Validación cruzada y regresión logística

Estos dos conceptos se han explicado antes para árboles de decisión pero también van a ser empleados en regresión logística.

La validación cruzada es un método de evaluación que para dividir el conjunto de datos en múltiples subconjuntos, entrenando el modelo en algunos de estos subconjuntos y validándolo en los restantes. Esto se repite varias veces y los resultados se promedian para

obtener una estimación más robusta del rendimiento del modelo. El coeficiente kappa, en cambio, mide la concordancia entre dos conjuntos de categorías (en el caso de la regresión logística, las predicciones del modelo y las etiquetas verdaderas), corrigiendo por el acuerdo que podría esperarse por azar. Su fórmula como cabría de esperar es la misma que en árboles de decisión [**21**].

2.2.5. Métodos de selección de variables

Information Gain Ratio

A continuación se va a explicar este método para seleccionar variables puesto que va a ser empleado más adelante. Este método (también empleado en árboles de decisión) consiste en reducir la entropía producida de dividir un conjunto con atributos α y encontrar el candidato óptimo que produce el mayor valor en la fórmula 2.13 [**22**]:

$$\mathrm{IG}(T, a) = \mathrm{H}(T) - \mathrm{H}(T|a)$$

Fórmula 2.13: Reducción de entropía

Donde T es una variable aleatoria y $H(T|\alpha)$ es la entropía de T dado el valor del atributo α. El valor information gain es igual a la entropía total de un atributo si por cada valor del atributo una única clasificación puede ser posible en la fórmula 2.14.

$$\mathrm{SplitInformation}(X) = -\sum_{i=1}^{n} \frac{\mathrm{N}(x_i)}{\mathrm{N}(x)} * \log_2 \frac{\mathrm{N}(x_i)}{\mathrm{N}(x)}$$

Fórmula 2.14: Valor de SplitInformation(X)

El valor de SplitInformation puede ser calculado en la fórmula 2.14 donde X es una variable aleatoria discreta con un conjunto de posibles valores $x_1, x_2, ..., x_i$ y $N(x_i$ el número de veces que esa x_i ocurre dividido por el total de eventos $N(x)$ donde x es el conjunto de eventos. El valor SplitInformation es un número positivo que describe el valor potencial de dividir una rama en un nodo.

Luego ya para calcular el valor Information Gain Ratio se utiliza la fórmula 2.15. Este ratio se define como $IGR(T, \alpha) = IG(T, \alpha)/SplitInformation(T)$

$$\mathrm{IGR}(T, a) = \frac{-\sum_{i=1}^{n} \mathrm{P}(T) \log \mathrm{P}(T) - (-\sum_{i=1}^{n} \mathrm{P}(T|a) \log \mathrm{P}(T|a))}{-\sum_{i=1}^{n} \frac{\mathrm{N}(t_i)}{\mathrm{N}(t)} * \log_2 \frac{\mathrm{N}(t_i)}{\mathrm{N}(t)}}$$

Fórmula 2.15: Cálculo de Information Gain Ratio

2.2.6. Librerías utilizadas en R

BigChess

La librería Bigchess en R es una herramienta poderosa para trabajar con datos relacionados con el ajedrez. Permite importar datos de partidas de ajedrez desde archivos en formatos estándar como PGN (Portable Game Notation), FEN (Forsyth–Edwards Notation) y CSV (Comma-Separated Values). Una vez importados los datos, Bigchess ofrece funciones para manipular y limpiarlos, así como para llevar a cabo análisis estadísticos y experimentos en el ámbito del ajedrez.

OneR

La librería OneR implementa diversas técnicas de machine learning, incluyendo las reglas de asociación que serán utilizadas más adelante en el trabajo.

Caret

La librería caret (Classification And REgression Training) en R es una herramienta integral para la creación de modelos de machine learning. Proporciona una amplia gama de funciones para preprocesamiento de datos, selección de características, ajuste de modelos y evaluación de rendimiento. Va a ser empleada para crear árboles de decisión.

aod

Está diseñada para análisis de datos orientados a modelos de regresión y pruebas estadísticas, particularmente en el contexto de datos binomiales y de conteo. Será empleada para crear modelos de regresión logística.

2.2.7. Librerías utilizadas en Python

Numpy

NumPy es una biblioteca fundamental para la computación numérica en Python. Proporciona una estructura de datos de matriz multidimensional eficiente, junto con una amplia gama de funciones matemáticas para operaciones de álgebra lineal, transformadas de Fourier, generación de números aleatorios y mucho más.

Pandas

Pandas es una biblioteca de análisis de datos que proporciona estructuras de datos flexibles y herramientas de manipulación de datos diseñadas para trabajar con datos tabulares y series temporales.

Chess

Chess es una biblioteca específica de Python diseñada para trabajar con el juego de ajedrez. Proporciona clases y funciones para representar y manipular tableros de ajedrez, movimientos de piezas, posiciones, partidas, aperturas y más.

Capítulo 3

Preparación de la base de datos

3.1. Descripción de la base de datos

He comentado antes que existen a día de hoy dos grandes plataformas para jugar al ajedrez a día de hoy, Chess.com y Lichess.org. Para este trabajo voy a utilizar una de las bases de datos alojadas en la plataforma Lichess, concretamente la subida en diciembre del año 2013 que contiene 579.263 partidas.

El formato de la base de datos es .pgn y al leerlo en python vemos dos variables: una llamada 'headers' que contiene información relevante sobre la partida y otra llamada "moves"que contiene en sí misma todos los movimientos de cada una de las partidas.

Ejemplo de entrada en la variable 'headers':

```
{'_tag_roster': {'Event': 'Rated Classical game', 'Site': 'https://lichess.org/83rjsno9',
     'Date': '????.??.??', 'Round': '?', 'White': 'goran123', 'Black': 'alexj2013', '
    Result': '1-0'}, '_others': {'UTCDate': '2013.11.30', 'UTCTime': '23:02:45', '
    WhiteElo': '1584', 'BlackElo': '1620', 'ECO': 'C26', 'Opening': 'Vienna Game: Stanley
     Variation, Reversed Spanish', 'TimeControl': '300+8'}}
```

Vemos varias variables sobre la partida como el nombre de la apertura, el elo de las blancas e incluso un enlace para ver la propia partida en Lichess.

```
1. e4 e5 2. Bc4 Nf6 3. Nf3 Nxe4 4. Bxf7+ Kxf7 5. Nxe5+ Kg8 6. O-O d6 7. Nc4 b5 8. Ne3 Bb7
     9. d3 Nf6 10. c3 g6 11. f4 Bg7 12. f5 Nc6 13. Qb3+ Kf8 14. fxg6 hxg6 15. Ng4 Ne5 16.
     Nxf6 Bxf6 17. d4 Ng4 18. Bg5 Kg7 19. Bxf6+ Nxf6 20. Qe6 Rhe8 21. Qh3 Re2 22. Na3
    Rxg2+ 0-1
```

Esto sería un ejemplo de una partida completa de ajedrez que podríamos introducir

en un programa relacionado con archivos .pgn y poder ver la partida de principio a final.

Las variables que contiene el archivo escritas en la variable headers son las siguientes que se pueden ver en la tabla 3.1. Cada una de ellas se ha acompañado de un ejemplo de valor que pueden tomar.

Variable	**Descripción**	**Ejemplo**
Site	Enlace a Lichess.org para ver la partida completa	https://lichess.org/83rjsno9
Date	Fecha en la que se jugó la partida	03/07/1998
White	Nombre del usuario que jugó las blancas	MagnusAlt4ccount
Black	Nombre del usuario que jugó las negras	goran123
Result	Resultado de la partida	1/2-1/2
Event	Modo de juego	Rated Blitz game
ECO	Código asociado a la apertura jugada	B23
Opening_Names	Nombre de la apertura jugada	Vienna Game: Stanley Variation
TimeControl	Tiempo disponible para cada jugador	180+0

Tabla 3.1: Detalles de la Partida de Ajedrez

Es necesario mencionar que todas estas variables son de tipo string y que posteriormente se harán las transformaciones adecuadas. También se va a combinar esta extracción de variables en Python con funciones de la librería bigchess de R. Esta librería va a crear las siguientes variables reflejadas en la tabla 3.2. Como nota aclaratoria, las variables que terminan en "moves"significan el número de veces que se ha movido esa pieza. Por ejemplo, la variable W_Q_moves cuenta el número de veces que se ha movido la reina (W de White de que el jugador es las blancas, si fuera B de Black significaría que el jugador es las negras. La letra Q hace referencia a la dama). Si no hay una aclaración de que jugador la ha movido, significa que es la suma de ambos jugadores.

Variable	Descripción	Tipo de variable
W_K_moves	Rey blanco	integer
W_O_moves	Enroque blanco	integer
W_N_moves	Caballo blanco	integer
W_Q_moves	Reina blanca	integer
W_R_moves	Torre blanca	integer
B_K_moves	Rey negro	integer
B_O_moves	Enroque negro	integer
B_N_moves	Caballo negro	integer
B_Q_moves	Dama negra	integer
B_R_moves	Alfil negro	integer
K_moves	Suma de B_K_moves y W_K_moves	integer
O_moves	Enroque negro	integer
N_moves	Caballo negro	integer
Q_moves	Dama negra	integer
R_moves	Alfil negro	integer
complete.movetext	Verifica que el set de movimientos esté completo	Booleano
Nmoves	Número de movimientos en la partida	integer
W1, W2, W3, W4, W5 W6, W7, W8, W9, W10	Variables sobre los 10 primeros movimientos de las blancas	string
B1, B2, B3, B4, B5 B6, B7, B8, B9, B10	Variables sobre los 10 primeros movimientos de las negras	string

Tabla 3.2: Variables generadas por bigchess

3.2. Transformación de variables

Result

Como se ha dicho antes, esta variable es de tipo string. Para un mejor visionado se va a hacer una ligera transformación de la variable a:

Valor	**Nuevo valor**
1/2-1/2	Empate
1-0	Victoria
0-1	Derrota

Tabla 3.3: Transformación variable Result

WhiteElo y BlackElo

Estas dos variables que representan el elo de las blancas y de las negras, respectivamente, se van a transformar a tipo entero puesto que ahora mismo son de tipo string a pesar de ser números. Se han encontrado valores faltantes que se van a tratar posteriormente, pero no se han encontrado valores extraños que han dificultado la conversión como "15h4.º "155.2".

Capítulo 4

Análisis exploratorio de datos

4.1. Tratamiento de datos faltantes

Eliminación de partidas vacías

Lo mejor sería comenzar eliminando de toda la base de datos aquellas partidas que están vacías, esto es, partidas que han terminado sin que las blancas hayan hecho ningún movimiento.

```
entradas_vacias = games['Movetext'].isnull().sum()
```

Con este código se ve que existen 275 partidas de ajedrez que han terminado sin que las blancas hayan movido por lo que lo mejor sería eliminarlas.

games = games.dropna(subset=['Movetext'])

Pasando así la base de datos de 579.263 partidas a 578.988. En este caso hacer una imputación de datos faltantes sería muy complejo puesto que habría que generar una partida completa de ajedrez y siendo solo un 0.047 % del total de datos no resulta necesario.

WhiteElo y BlackElo

Para las variables WhiteElo y BlackElo se han encontrado 257 y 144 valores faltantes, respectivamente. Se va a hacer una imputación de datos sustituyendo los valores faltantes por la media de sus respectivas variables, siendo para WhiteElo su media 1622.996 y para BlackElo 1611.2769. Sin embargo, como el elo solamente puede ser un número natural positivo se va a redondear a 1623 y a 1611, respectivamente.

No se han encontrado valores faltantes en otras variables.

4.2. Valores repetidos

El siguiente paso es explorar cuantas partidas de ajedrez existen en la base de datos que sean exactamente iguales. Hay que tener en cuenta que aunque dos partidas hayan tenido exactamente los mismos movimientos no significa que el resultado sea el mismo (los jugadores pueden acordar tablas o uno de ellos rendirse pensando que su posición es bastante inferior). Igualmente, para el objetivo de este trabajo lo mejor sería eliminar las partidas repetidas independientemente del resultado.

```
duplicados = games.duplicated(subset=['Movetext'], keep=False)
cantidad\_valores\_duplicados = duplicados.sum()
duplicados = games.duplicated(subset=['Movetext'], keep='first')
games\_sin\_duplicados = games.drop\_duplicates(subset=['Movetext'], keep='first')
```

Se tiene que había un total de 5.617 partidas duplicadas (un 0.97 % del total). Se han eliminado y ahora la base de datos ha pasado a 573.318 entradas.

4.3. Identificación de ruido

4.3.1. Errores tipográficos

Variable Event

Respecto a esta variable algunas entradas se han registrado erróneamente guardando aparte del modo de juego el enlace a Lichess para ver la partida. Por ejemplo, la entrada 16 de esta variable es: Rated Blitz tournament https://lichess.org/tournament/rfs28knk". Se arregla de la siguiente manera:

```
def eliminarenlaces(variable):
```

```
    return re.sub(r'https?://\\S+', '', variable)


games['Event'] = games['Event'].apply(eliminarenlaces)
```

No se han encontrado errores de tipografía en otras variables.

4.3.2. Valores atípicos

Distribución de la variable Event

Se van a eliminar los modos de juego de la base de datos que no lleguen mínimo al 3 % del total de partidas de ajedrez por lo que los modos de juego Rated Blitz tournament, Rated Bullet tournament, Rated Correspondence game y Rated Classical tournament van a ser eliminados y la base de datos va a pasar a 570.427 entradas. En la figura 4.1. se ve los modos de juego que se van a tener en cuenta.

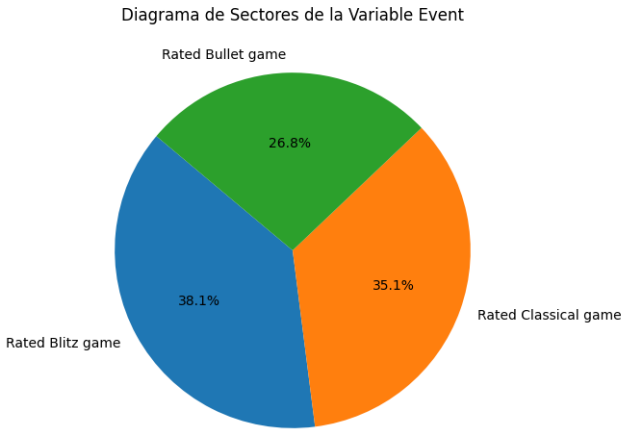

Figura 4.1: Distribución de la variable Eventos

Desproporción de jugadores

Podría darse el caso de que una minoría de jugadores tuviera una participación demasiado alta en el total de partidas. Se ha estudiado la frecuencia relativa de todos los

jugadores en cada uno de los distintos modos de juego y la participación más alta de un solo jugador es del 0.00389 % del total de partidas en el modo Rated Blitz game por lo que no será necesario hacer cambios en este aspecto.

Duración anómala de las partidas

Continuando con la distribución del número de movimientos, sería conveniente eliminar las partidas que son demasiado cortas por carecer de valor estadístico y las partidas demasiado largas por ser dudosa la intencionalidad de los jugadores.

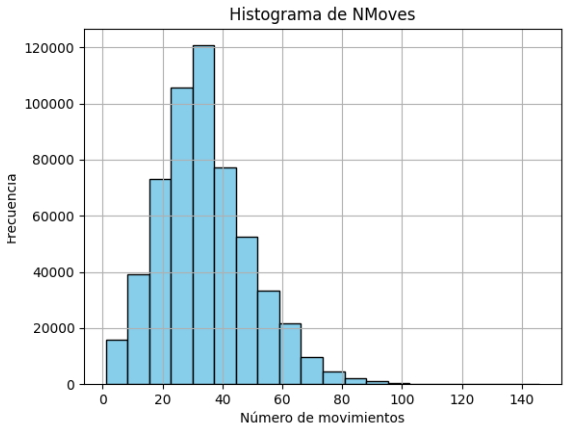

Figura 4.2: Histograma de NMoves

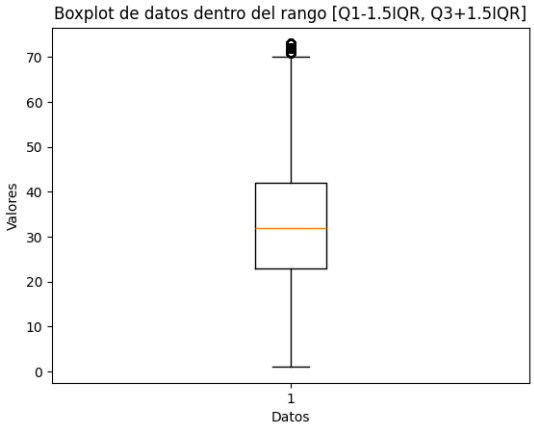

Figura 4.3: Diagrama de caja y bigotes

Observando el histograma NMoves en la figura 4.2 vemos que existe un porcentaje bastante bajo de partidas que tienen un número muy grande de movimientos. En cualquier caso, va a ser mejor examinarlo. Con el diagrama de caja y bigotes en la figura 4.2, el valor Q3+1.5IQR es igual a 73 y existen 8.442 observaciones por encima de él. No se van a eliminar todas las observaciones que superan este valor porque existen muchas partidas de ajedrez perfectamente válidas que superan los 73 movimientos. Sin embargo, teniendo en cuenta el valor Q3 + 3IQR igual a 103 se van a eliminar todas las partidas que superan esta longitud aunque sean solamente 331.

Así, la base de datos pasa de 570.427 entradas a 570.096.

También sería conveniente eliminar las partidas que sean demasiado cortas. Para ello, se va a tomar como referencia el jaque mate más rápido posible, el Mate del Loco [**23**] que es posible realizarlo en tan solo dos movimientos por lo que se van a eliminar todas las partidas con dos o menos movimientos. Así, se eliminan 1.076 partidas de la base de datos y pasamos a 569.020

Variables sobre el número de movimientos de cada pieza

La parte final de una partida de ajedrez se conoce como endgame, una fase en la que quedan muy pocas piezas en el tablero. En esta parte del juego existen patrones que se repiten a lo largo de las partidas, como por ejemplo cuando a ambos jugadores solamente les queda las reinas o una torre. Esto es una dificultad a la hora de analizar los valores extremos de estas variables porque que una partida tenga un número muy alto de movimientos de torre no la hace inválida para un análisis. Por esta razón se ha buscado solamente si alguno de estas variables es superior a su valor correspondiente en la variable NMoves, puesto que no tendría sentido una partida así.

De todas las variables que hacen referencia al número de veces que se ha movido cada pieza no se ha encontrado ninguna partida cuyo valor sea superior al número total de movimientos de la partida.

4.4. Eliminación de variables innecesarias

Se van a eliminar las siguientes variables puesto que no son necesarias para el presente trabajo:

Variable	**Motivo**
White	El nombre de usuario no afecta al resultado de la partida
Black	Misma razón que el anterior
complete.movetext	No presenta ninguna utilidad
Site	No se va a emplear el enlace a Lichess
Movetext	Por falta de potencia tecnológica no se va a utilizar
ECO	Se tienen ya variables sobre los primeros movimientos de la partida
Opening.Names	Misma razón que el anterior.
TimeControl	Un mismo modo tiene distintos tiempos

Tabla 4.1: Variables eliminadas

Y, a parte de estas variables, se van a eliminar también las siguientes por su capacidad de explicar la variable Result. Se ha utilizado el algoritmo Information Gain Ratio y se ha detectado que las variables que hacen referencia a los 10 primeros movimientos de la partida no van a ser útiles. Se ha comprobado si depende del modo de juego empleado pero todos los modos de juego han dado prácticamente los mismos valores.

Variable	Valor Gain Ratio
19 W9	0.001641
22 B10	0.001620
21 W10	0.001598
20 B9	0.001562
17 W8	0.001507
18 B8	0.001476
15 W7	0.001365
16 B7	0.001330
14 B6	0.001312
13 W6	0.001288
12 B5	0.001283
11 W5	0.001219
9 W4	0.001138
10 B4	0.001066
8 B3	0.001066
7 W3	0.001010
6 B2	0.000960
5 W2	0.000807
4 B1	0.000681
3 W1	0.000636

Tabla 4.2: Valores de Gain Ratio

Luego, las variables finales que se van a emplear en este análisis van a ser las siguientes en la tabla 4.3. En la tabla 4.4. se refleja el número total de partidas en la base de datos junto con el número total de partidas en cada modo de juego.

Variable	**Descripción**
Event	Modo de juego
Result	Resultado de la partida
NMoves	Número de movimientos en la partida
W_B_moves	Número de movimientos de los alfiles de las blancas
W_K_moves	Número de movimientos del rey de las blancas
W_N_moves	Número de movimientos del caballo de las blancas
W_O_moves	1 si las blancas han enrocado; 0 si no lo han hecho
W_Q_moves	Número de movimientos de la reina de las blancas
W_R_moves	Número de movimientos de las torres de las blancas
B_B_moves	Número de movimientos de los alfiles de las negras
B_K_moves	Número de movimientos del rey de las negras
B_N_moves	Número de movimientos de los caballos de las negras
B_O_moves	1 si las negras se han enrocado; 0 si no lo han hecho
B_Q_moves	Número de movimientos de la reina de las negras
B_R_moves	Número de movimientos de las torres de las negras
WhiteElo	Nivel de elo de las blancas
BlackElo	Nivel de elo de las negras

Tabla 4.3: Variables finalmente utilizadas

Evento	**Porcentaje**	**Total**
Rated Blitz Game	38.1 %	216,797
Rated Rapid Game	26.8 %	152,497
Rated Bullet Game	35.1 %	199,726
Total	**100 %**	**569,020**

Tabla 4.4: Número de partidas en cada modo consideradas para este trabajo

Capítulo 5

Análisis estadístico

5.1. Rated Blitz Game

5.1.1. Análisis estadístico del elo

A continuación se estudiarán los histogramas del elo de las negras junto con el elo de las blancas. También se buscará su media, desviación típica y test de normalidad. El test de normalidad elegido es el test de Lilliefors porque otros test (como el de Shapiro Wilk) se utilizan para tamaños de muestra pequeños y el de Kolmogorov-Smirnov asume conocida la media y varianza poblacional lo cual en la mayoría de casos no se conoce. Para solventar este problema, se utiliza el conocido como test Lilliefors. El test de Lilliefors asume que la media y varianza poblacionales son desconocidas, estando especialmente desarrollado para testear la normalidad [**24**]

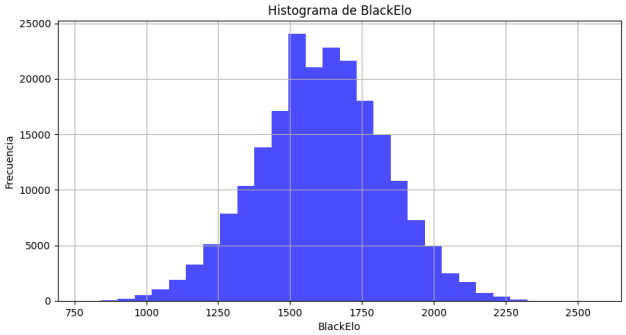

Figura 5.1: Histograma del elo de las negras para Rated Blitz Game

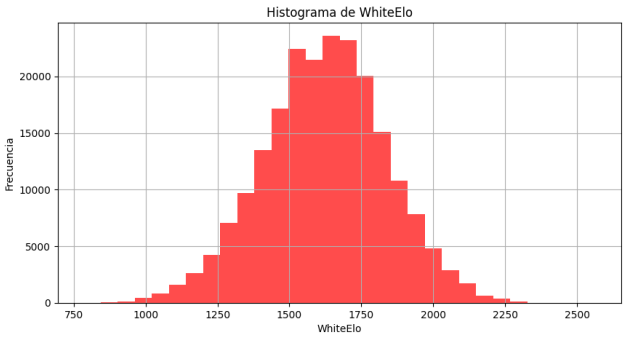

Figura 5.2: Histograma del elo de las blancas para Rated Blitz Game

Comentando las figuras 5.1 y 5.2 se observa que el histograma del elo parece seguir una distribución normal, sin embargo al calcular el p-valor del test de Lilliefors para ambas variables se tiene que el valor de ambos es aproximadamente 0 lo cual entonces se rechaza la hipótesis nula de que las variables siguen una distribución normal y se toma la alternativa, esto es, que siguen una distribución distinta. Cabe mencionar también que la media muestral para BlackElo y WhiteEo es de 1611.166 y 1622.839 respectivamente con desviaciones típicas de 225.53 y 219.3097.

5.1.2. Variables sobre el número de veces que se ha movido cada pieza

Variable	Media	Desviación Típica	P-valor Lilliefors
W_B_moves	5.38	3.06	0.00
W_K_moves	4.48	5.56	0.00
W_N_moves	6.07	3.49	0.00
W_Q_moves	4.29	3.70	0.00
W_R_moves	5.77	5.04	0.00
B_B_moves	5.19	3.10	0.00
B_K_moves	4.51	5.49	0.00
B_N_moves	5.97	3.49	0.00
B_Q_moves	4.14	3.68	0.00
B_R_moves	5.62	5.07	0.00
NMoves	35.04	15.26	0.00

Tabla 5.1: Media, desviación típica y normalidad para Rated Blitz Game

Observando la tabla 5.1. vemos que prácticamente ninguna variable de la base de datos sigue una distribución normal, lo cual es bastante esperable. El número medio de movimientos por partida es de 35 por lo que de media las partidas no son largas.

Variable	Porcentaje
Porcentaje de veces que W_O_moves es 1	80.76 %
Porcentaje de veces que B_O_moves es 1	75.11 %

Tabla 5.2: Enroque en Rated Blitz Game

En la tabla 5.2 vemos que las blancas enrocan en el 80.76 % de las partidas mientras que las negras en el 75.11 % de ellas. Esta diferencia seguramente sea debido a que las blancas comienzan antes que las negras por lo que tienen un mayor margen para poder enrocar y/o evitar que el oponente enroque.

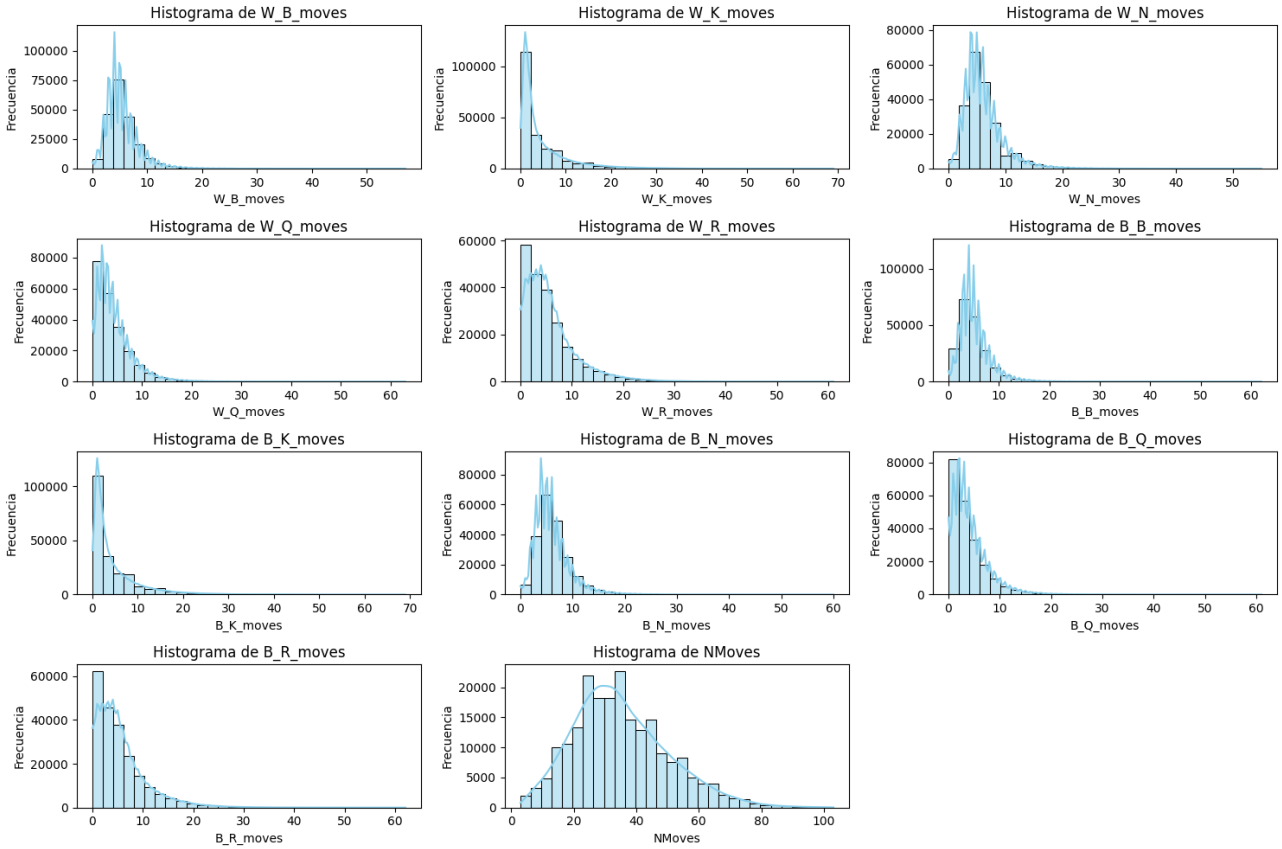

Figura 5.3: Histogramas para Rated Blitz Game

En la figura 5.3. se ha representado un histograma de todas las variables junto con la función KDE. Observando los histogramas vemos un patrón generalizado de un fuerte aumento del número de veces que se mueve cada pieza para luego un descenso (claramente por el progreso del juego, menos piezas quedan en la partida).

5.1.3. Correlación entre todas las variables

Algunos aspectos interesantes a destacar de la figura 5.4 son:

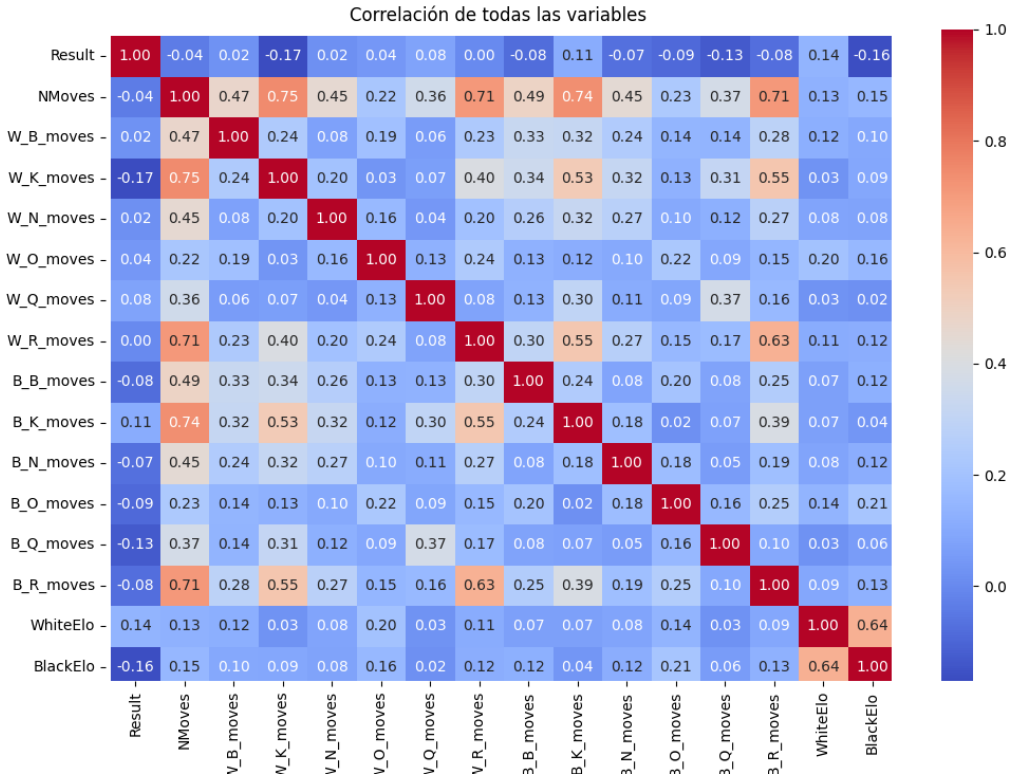

Figura 5.4: Correlación entre las variables para Rated Blitz Game

- La correlación entre W_N_moves y $NMoves$ es de 0.45, la misma que B_N_moves y $NMoves$.
- $WhiteElo$ y $BlackElo$ están correlados con 0.64. Es llamativo que no sea más alta la correlación porque el algoritmo de Lichess cuando una persona busca un oponente es buscarlo dentro de un rango respecto al elo del jugador (el intervalo es generalmente de ± 500).
- Ninguna de las variables tiene una correlación fuerte con la variable Result.

5.1.4. Correlación diferencia de elo y probabilidad de ganar

Una idea interesante de la base de datos sería ver como afecta el elo de un jugador a sus probabilidades de ganar. La correlación entre la diferencia de elo y la variable Result es de 0.305. Como era de esperar la correlación es positiva aunque no muy grande.

5.2. Rated Rapid game

5.2.1. Análisis estadístico del elo

Vamos a comenzar al igual que en el apartado anterior con un análisis estadístico del elo.

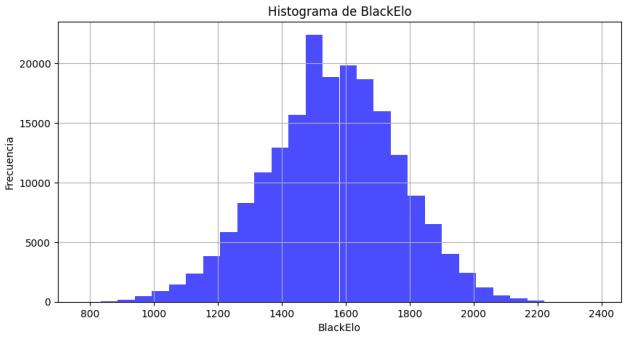

Figura 5.5: Histograma del elo de las negras para Rated Rapid Game

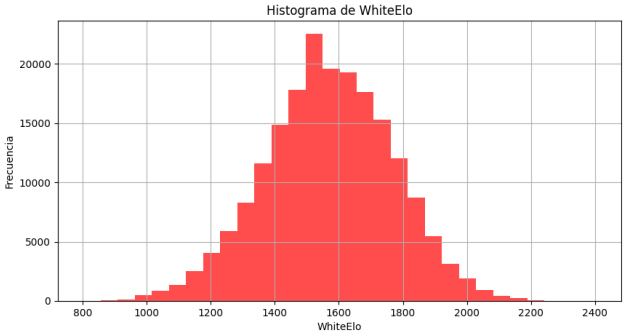

Figura 5.6: Histograma del elo de las blancas para Rated Rapid Game

Vemos en las figuras 5.5 y 5.6 que la distribución del elo es muy similar al apartado anterior. La media muestral en este caso para WhiteElo es algo inferior (1568 puntos) con una desviación típica de 199.63 mientras que para BlackElo sus valores son de 1559 y 205.017, respectivamente. Se rechazan las hipótesis de normalidad para el test de Lilliefors por tener p-valores de aproximadamente 0.

5.2.2. Variables sobre el número de veces que se ha movido cada pieza

Variable	Media	Desviación Típica	P-valor Lilliefors
W_B_moves	5.149072	3.106553	0
W_K_moves	4.212808	5.590050	0
W_N_moves	5.884090	3.636724	0
W_Q_moves	4.338952	3.985372	0
W_R_moves	5.454513	5.325030	0
B_B_moves	4.934599	3.175231	0
B_K_moves	4.433743	5.725606	0
B_N_moves	5.689167	3.559873	0
B_Q_moves	4.086279	3.881482	0
B_R_moves	5.226132	5.252080	0
NMoves	33.497310	16.494826	0

Tabla 5.3: Tabla de datos con la media, desviación típica y p-valor de Lilliefors para Rated Rapid Game

Vemos en la tabla 5.3. que los valores medios para cada variable son prácticamente los mismos excepto para NMoves, que es de casi dos movimientos menos de media por partida. Esto es algo interesante puesto que el modo Rapid es de una duración superior a las partidas Blitz lo que significa que los jugadores tienen mayor tiempo para hacer sus jugadas.

Variable	Porcentaje
Porcentaje de veces que W_O_moves es 1	72.679 %
Porcentaje de veces que B_O_moves es 1	65.666 %

Tabla 5.4: Porcentajes de veces que enroca cada jugador en partidas Rated Rapid Game

Vemos en la tabla 5.4 que es considerablemente inferior el número de veces que enroca cada jugador en las partidas. A pesar de tener un mayor tiempo para pensar cada movimiento, los jugadores deciden no enrocar.

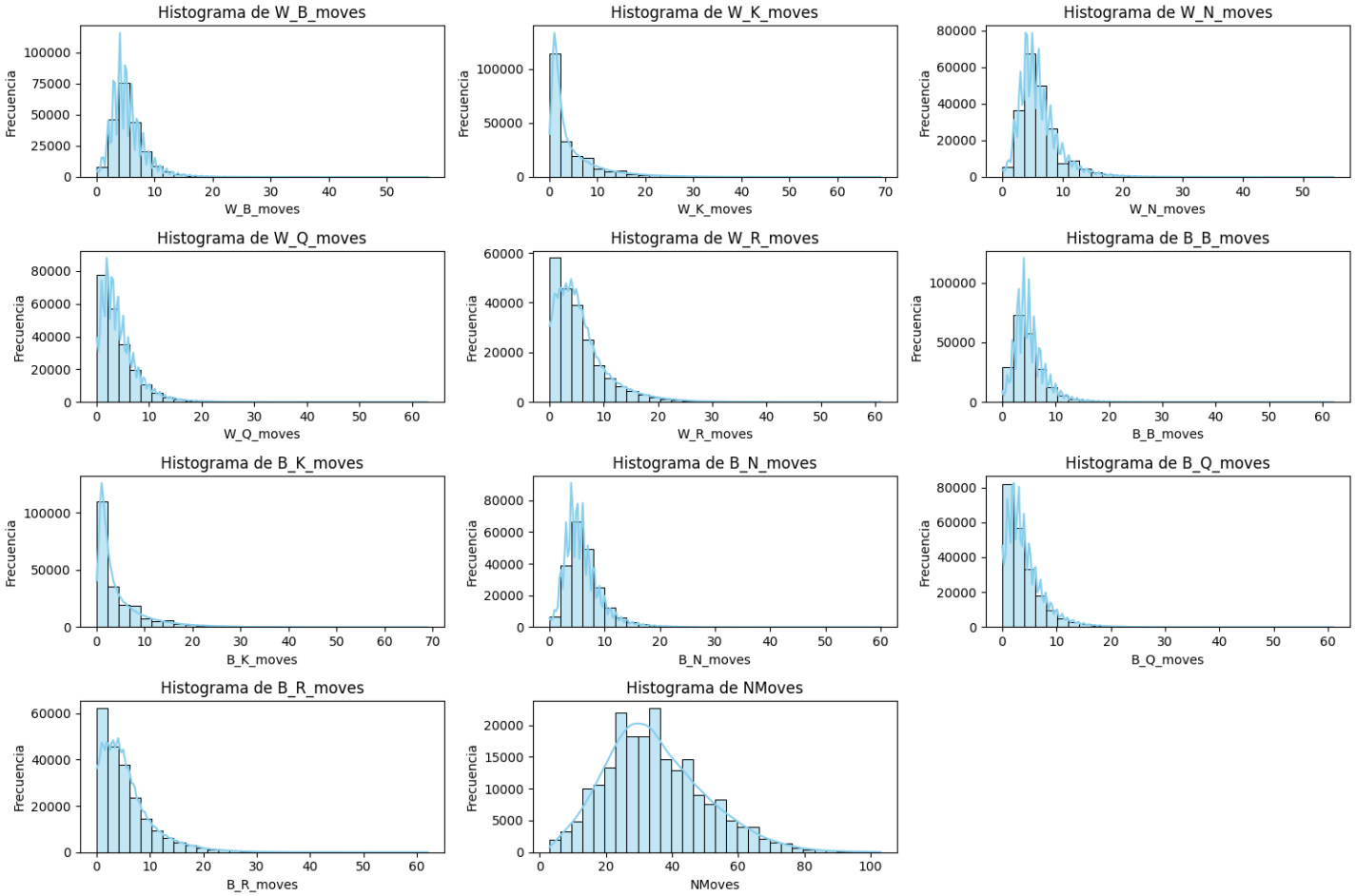

Figura 5.7: Histogramas de las variables para Rated Rapid Game

Los histogramas en 5.7 son prácticamente iguales a los del apartado anterior sin cambios destacables.

5.2.3. Correlación entre todas las variables

Haciendo un gráfico de la correlación entre todas las variables de la base de datos:

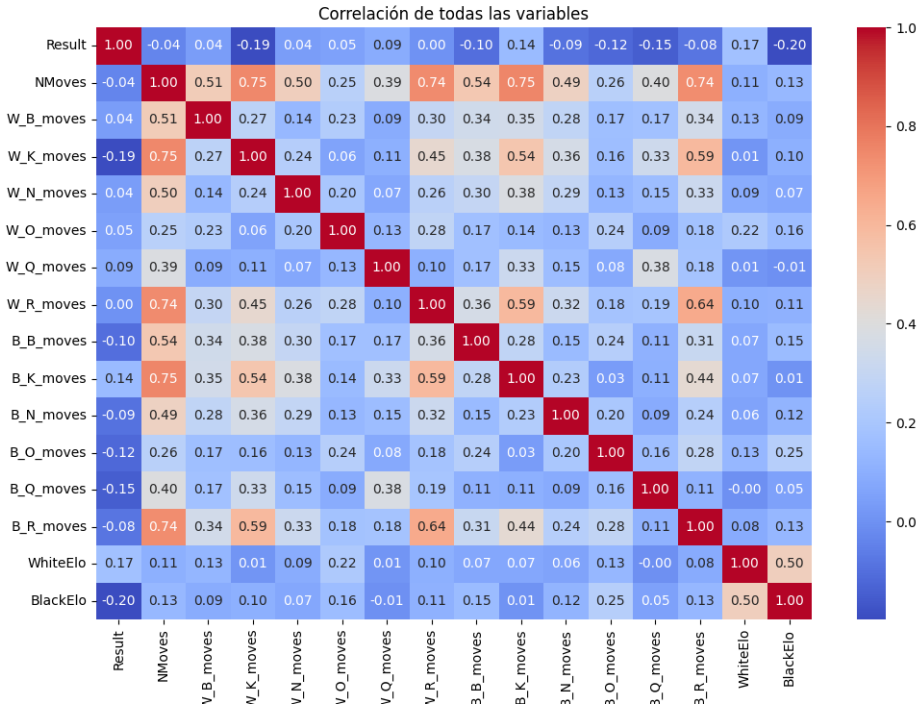

Figura 5.8: Correlación entre todas las variables para Rated Rapid Game

Las correlaciones en 5.8 entre las variables son prácticamente las mismas y no dependen del modo de juego empleado.

5.2.4. Correlación diferencia de elo y probabilidad de ganar

En este caso la correlación entre la diferencia de elo y la probabilidad de ganar es de 0.269, una correlación que al igual que en apartado anterior a pesar de ser positiva no es muy grande.

5.3. Rated Bullet Game

5.3.1. Análisis estadístico del elo

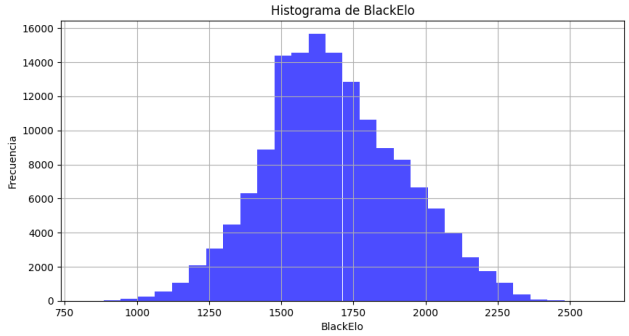

Figura 5.9: Histograma del elo de las negras para Rated Bullet Game

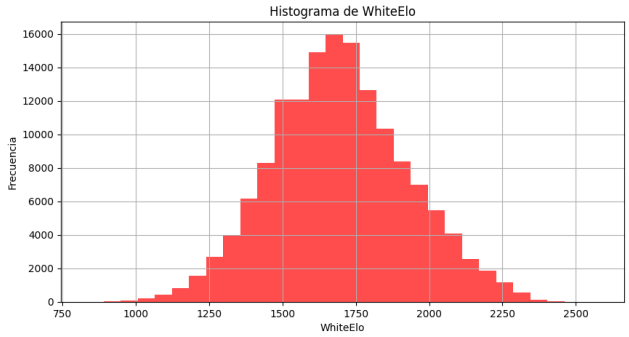

Figura 5.10: Histograma del elo de las blancas para Rated Bullet Game

Para este modo de juego se tiene una media de elo para las blancas de 1694.335 y para las negras de 1638 (siendo este el modo que posee los niveles de elo más altos) con desviaciones estándar de 231.575 y 240.235, respectivamente. Al igual que en los dos modos anteriores, tampoco se verifica la hipótesis de normalidad. Los histogramas en 5.9 y 5.10 son prácticamente idénticos a los anteriores.

5.3.2. Variables sobre el número de veces que se ha movido cada pieza

Variable	Media	Desviación Típica	P-valor Lilliefors
W_B_moves	5.162369	2.818217	0
W_K_moves	4.150239	5.097517	0
W_N_moves	5.858909	3.273006	0
W_Q_moves	3.991772	3.344657	0
W_R_moves	5.352008	4.447298	0
B_B_moves	4.974987	2.848464	0
B_K_moves	4.200050	5.085202	0
B_N_moves	5.742059	3.278670	0
B_Q_moves	3.895965	3.361031	0
B_R_moves	5.195529	4.465176	0
NMoves	33.416629	13.578543	0

Tabla 5.5: Tabla de datos con la media, desviación típica y p-valor de Lilliefors para Rated Bullet Game

Variable	Porcentaje
Porcentaje de veces que W_O_moves es 1	80 %
Porcentaje de veces que B_O_moves es 1	74.79 %

Tabla 5.6: Porcentaje de veces que se enroca para Rated Bullet Game

Viendo las tablas 5.5 y 5.6 se ve que no existen diferencias significativas respecto a los modos de juego anteriores. En la figura 5.11 vemos que los histogramas son prácticamente idénticos a los de modos de juego anteriores.

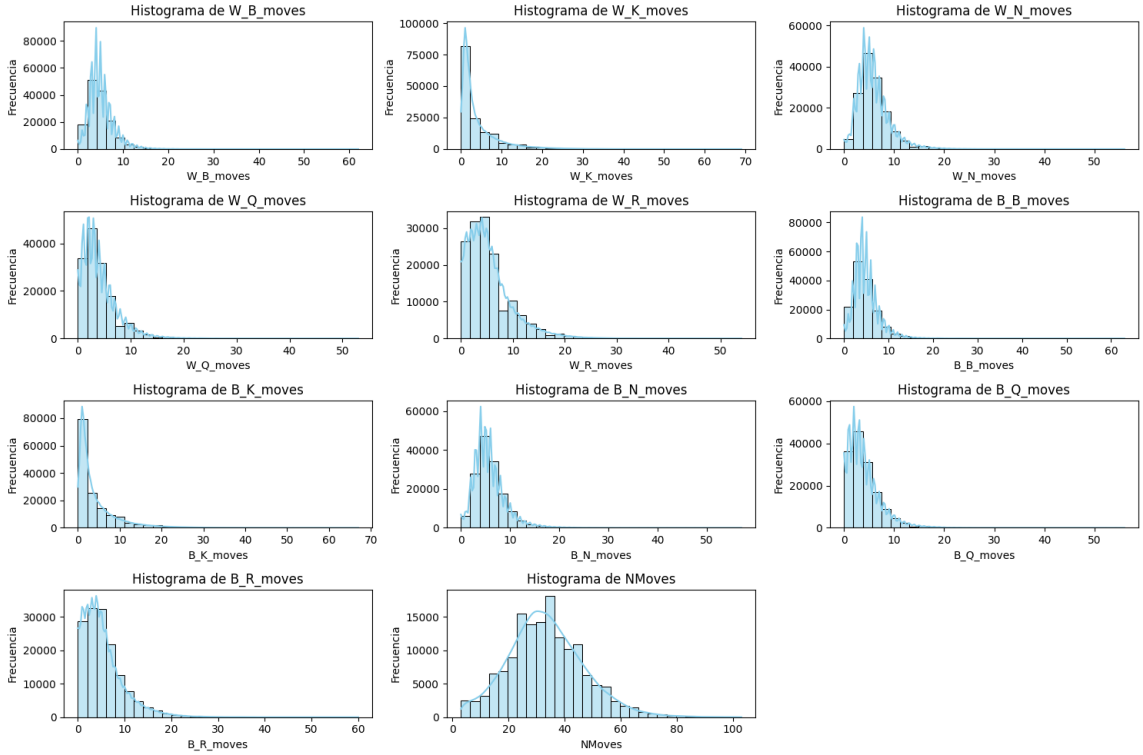

Figura 5.11: Histograma de las variables para Rated Bullet Game

5.3.3. Correlación entre todas las variables

Algunas correlaciones a destacar de la figura 5.12 son prácticamente idénticas a las de los apartados anteriores. No hay diferencias significativas en las correlaciones del número de movimientos de una pieza y el modo de juego.

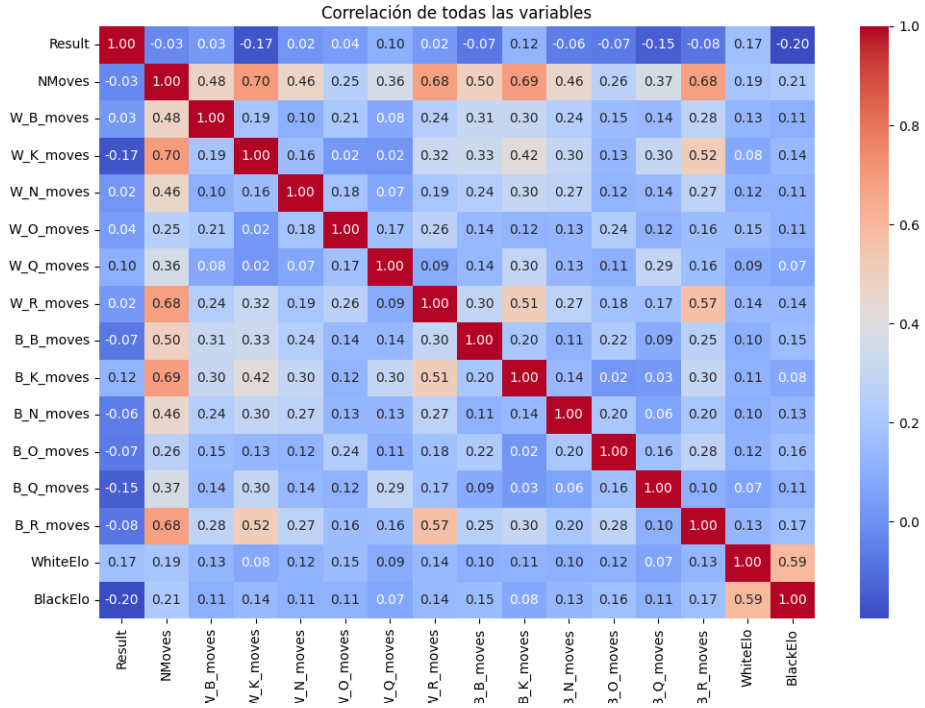

Figura 5.12: Correlaciones en partidas Rated Bullet Game

5.3.4. Correlación diferencia de elo y probabilidad de ganar

Se ha obtenido, en este caso, una correlación de 0.35. A pesar de ser la más alta hasta el momento, sigue sin ser una correlación alta. Un jugador de un elo más bajo podría tener buenas posibilidades de ganar a uno de mayor elo.

Capítulo 6

Minería de datos

En este capítulo se van a aplicar las técnicas de minería de datos que antes se han explicado en el capítulo 2. Como tenemos 3 tipos de partidas distintas (Rated Blitz Game, Rated Rapid Game y Rated Bullet Game) vamos a discriminar por tipo de partida para ver si los resultados dependen del tipo de esta. También se va a tener en cuenta en cada apartado qué condiciones dan la victoria a las blancas y qué condiciones dan la victoria a las negras.

6.1. Árboles de decisión

Se van a comenzar las técnicas de minería de datos con los árboles de decisión. Para este apartado, se ha utilizado en cada modo de juego una validación cruzada de 10 subconjuntos y una profundidad máxima de 10 nodos (utilizando luego el 100 % de los datos). Todas las variables han sido consideradas aunque luego el algoritmo ha empleado una minoría de ellas. Es importante mencionar que el algoritmo no ha predecido en ninguno de los tres modos de juego el resultado de Empate debido a que este representa sólo un 3 % de las partidas. Aún así estas partidas no se han eliminado porque sería falsificar los resultados (no tendría sentido decir que bajo ciertas condiciones las blancas tienen un 60 % de ganar puesto que esa probabilidad no ha tenido en cuenta el reducido número de partidas en las que se puede empatar, que es un resultado posible y no acorde al que se predice).

6.1.1. Rated Blitz Game

Podemos ver en la figura 6.1. el árbol de decisión asociado a este modo de juego. Es importante mencionar que este árbol de decisión tiene una precisión del 65.4 % en validación cruzada a la hora de predecir el resultado de una partida junto con un coeficiente Kappa asociado de tan solo 0.357. El árbol se ha construído con el 100 % de los datos.

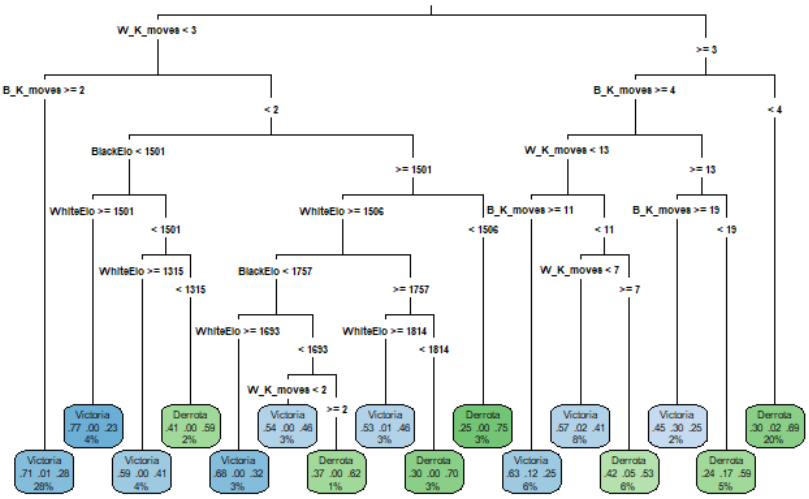

Figura 6.1: Árbol de decisión para partidas Blitz

Observando la tabla 6.1 y comenzando con las blancas, la primera regla generada, $W\_K\_moves < 3, B\_K\_moves \geq 2$ es interesante ver que para ella el algoritmo predice un 71 % de probabilidad de ganar cuando su rey se ha movido entre 0 y 2 veces y el rey de las negras se ha movido 2 o más. Esto bajo el punto de vista de un jugador de ajedrez podría ser debido a dos motivos principales: las negras han perdido tiempo moviendo demasiado al rey lo que las ha llevado a una posición de desventaja (algo propio de niveles de elo bajos) o también podría ser debido a que su rey ha estado sufriendo jaques y ha terminado en jaque mate. Si nos fijamos ahora en la victoria para las negras, vemos que exactamente lo contrario se cumple para ellas: $W\_K\_moves \geq 3, B\_K\_moves < 4$ da una probabilidad del 69 % de ganar a las negras.

Es interesante ver también que en la tabla 6.1 las reglas hacen una confirmación de la importancia del elo de un jugador a la hora de ganarle a su adversario. Por ejemplo, con la segunda regla generada $W\_K\_moves < 3, B\_K\_moves < 2, BlackElo < 1501, WhiteElo \geq 1501$ cuando tanto el rey de las blancas como el rey de las negras han realizado muy pocos movimientos (incluso el de las negras con un movimiento menos en el límite superior de su intervalo), si el elo de las blancas es superior a 1501 y el de las negras es inferior a 1501, la probabilidad de ganar para las blancas es del 77 %. Aún así, es importante mencionar que esto depende en gran medida de la dispersión de los datos: un jugador de 1501 de elo con las blancas su probabilidad de ganar va a ser inferior al 77 % pero si el jugador de las blancas tuviera de elo 2500 y el de las negras 1000, su probabilidad de ganar seguramente pase del 90 %.

Tabla 6.1: Reglas de decisión para partidas Blitz

Regla	Aciertos (%)	Total (%)	Resultado
W_K_moves < 3, B_K_moves ≥ 2	71	28	Victoria
W_K_moves < 3, B_K_moves < 2, BlackElo < 1501, WhiteElo ≥ 1501	77	4	Victoria
W_K_moves < 3, B_K_moves < 2, BlackElo < 1501, WhiteElo < 1501, WhiteElo ≥ 1315	59	4	Victoria
W_K_moves < 3, B_K_moves < 2, BlackElo < 1501, WhiteElo < 1501, WhiteElo < 1315	59	2	Derrota
W_K_moves < 3, B_K_moves < 2, BlackElo ≥ 1501, WhiteElo ≥ 1506, BlackElo < 1757, WhiteElo ≥ 1693	68	3	Victoria
W_K_moves < 3, B_K_moves < 2, BlackElo ≥ 1501, WhiteElo ≥ 1506, BlackElo < 1757, WhiteElo < 1693, W_K_moves < 2	54	3	Victoria
W_K_moves < 3, B_K_moves < 2, BlackElo ≥ 1501, WhiteElo ≥ 1506, BlackElo < 1757, WhiteElo < 1693, W_K_moves ≥ 2	62	1	Derrota
W_K_moves < 3, B_K_moves < 2, BlackElo ≥ 1501, WhiteElo ≥ 1506, BlackElo ≥ 1757, WhiteElo ≥ 1814	53	3	Victoria
W_K_moves < 3, B_K_moves < 2, BlackElo ≥ 1501, WhiteElo ≥ 1506, BlackElo ≥ 1757, WhiteElo < 1814	70	3	Derrota
W_K_moves < 3, B_K_moves < 2, BlackElo ≥ 1501, WhiteElo < 1506	75	3	Derrota
W_K_moves ≥ 3, B_K_moves ≥ 4, W_K_moves < 13, B_K_moves ≥ 11	63	6	Victoria
W_K_moves ≥ 3, B_K_moves ≥ 4, W_K_moves < 13, B_K_moves < 11, W_K_moves < 7	57	8	Victoria
W_K_moves ≥ 3, B_K_moves ≥ 4, W_K_moves < 13, B_K_moves < 11, W_K_moves ≥ 7	53	6	Derrota
W_K_moves ≥ 3, B_K_moves ≥ 4, W_K_moves ≥ 13, B_K_moves ≥ 19	45	2	Victoria
W_K_moves ≥ 3, B_K_moves ≥ 4, W_K_moves ≥ 13, B_K_moves < 19	59	5	Derrota
W_K_moves ≥ 3, B_K_moves < 4	69	20	Derrota

6.1.2. Rated Rapid Game

Vemos en la figura 6.2. el árbol de decisión para partidas Rapid. Esta vez el coeficiente Kappa asociado es de 0.357 que sigue siendo bastante bajo y la precisión es del 69 %. El árbol ha sido generado con el 100 % de los datos.

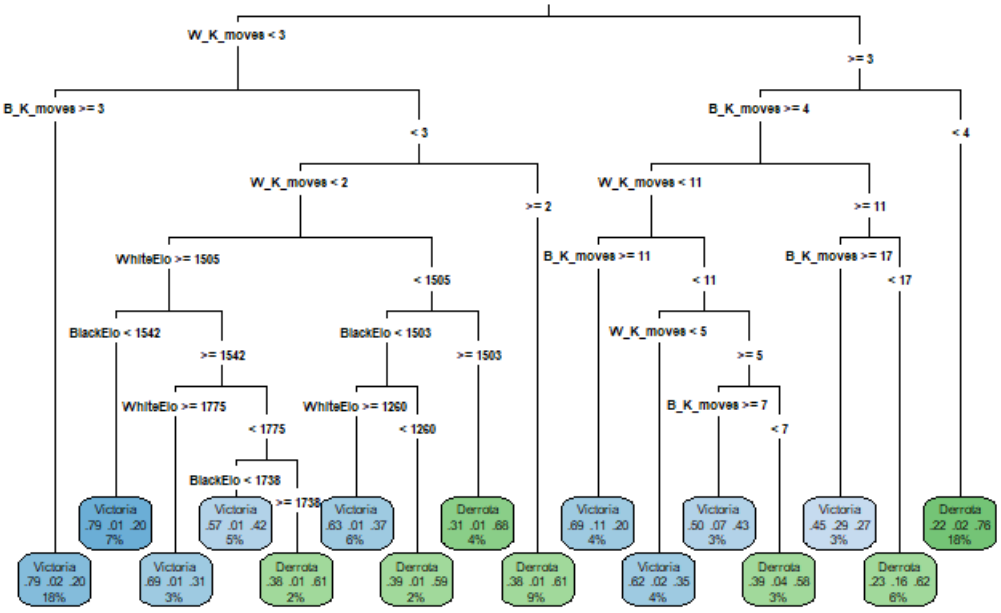

Figura 6.2: Árbol de decisión para partidas Rapid

Continuando con la tabla 6.2, podemos sacar prácticamente el mismo conocimiento que en el apartado anterior pero con distintas probabilidades. Si cogemos la misma regla que da victoria en el caso anterior para las blancas, $W\_K\_moves < 3, B\_K\_moves \geq 3$, vemos que la probabilidad de ganar pasa del 71 % al 79 %, muy probablemente debido a que las partidas de este modo de juego son bastante más largas que el anterior y eso da lugar a cometer menos errores en ajedrez. Si tenemos en cuenta también la misma regla pero en su caso para las negras, $W\_K\_moves \geq 3, B\_K\_moves < 4$, para este modo de juego la probabilidad de ganar es del 78 % mientras que la misma regla para el caso anterior es del 69 %. Respecto al elo, podemos sacar las mismas conclusiones del

apartado anterior con reglas como $W\_K\_moves < 3, B\_K\_moves < 3, W\_K\_moves < 2, WhiteElo < 1505, BlackElo \geq 1503$ que posee un 68 % de probabilidad de acierto.

Tabla 6.2: Reglas de decisión para partidas Rapid

Regla	Acierto (%)	Total (%)	Resultado
W_K_moves $<$ 3, B_K_moves $\geq$ 3	79	18	Victoria
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $<$ 2, WhiteElo $\geq$ 1505, BlackElo $<$ 1542	79	7	Victoria
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $<$ 2, WhiteElo $\geq$ 1505, BlackElo $\geq$ 1542, WhiteElo $\geq$ 1775	69	3	Victoria
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $<$ 2, WhiteElo $\geq$ 1505, BlackElo $\geq$ 1542, WhiteElo $<$ 1775, BlackElo $<$ 1738	57	5	Victoria
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $<$ 2, WhiteElo $\geq$ 1505, BlackElo $\geq$ 1542, WhiteElo $<$ 1775, BlackElo $>$ 1738	61	2	Derrota
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $<$ 2, WhiteElo $<$ 1505, BlackElo $<$ 1503, WhiteElo $\geq$ 1260	63	6	Victoria
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $<$ 2, WhiteElo $<$ 1505, BlackElo $<$ 1503, WhiteElo $<$ 1260	59	2	Derrota
W_K_moves $<$ 3, B_K_moves $<$ 3,			

Regla	Acierto (%)	Total (%)	Resultado
W_K_moves $<$ 2, WhiteElo $<$ 1505, BlackElo $\geq$ 1503	68	4	Derrota
W_K_moves $<$ 3, B_K_moves $<$ 3, W_K_moves $\geq$ 2	61	9	Derrota
W_K_moves $\geq$ 3, B_K_moves $\geq$ 4, W_K_moves $<$ 11, B_K_moves $\geq$ 11	69	4	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 4, W_K_moves $<$ 11, B_K_moves $<$ 11, W_K_moves $<$ 5	62	4	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 4, W_K_moves $<$ 11, B_K_moves $<$ 11, W_K_moves $\geq$ 5, B_K_moves $\geq$ 7	50	3	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 4, W_K_moves $<$ 11, B_K_moves $<$ 11, W_K_moves $\geq$ 5, B_K_moves $<$ 7	58	3	Derrota
W_K_moves $\geq$ 3, B_K_moves $\geq$ 4, W_K_moves $\geq$ 11, B_K_moves $\geq$ 17	45	3	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 4, W_K_moves $\geq$ 11, B_K_moves $<$ 17	62	6	Derrota
W_K_moves $\geq$ 3, B_K_moves $<$ 4	78	18	Derrota

6.1.3. Rated Bullet Game

Terminamos esta técnica de decisión con las partidas Bullet. El árbol expuesto en la figura 6.3. en este caso tiene un coeficiente kappa asociado de 0.313 y una precisión de tan solo el 62.3 % en validación cruzada, siendo estos los valores más bajos de los 3 modelos generados.

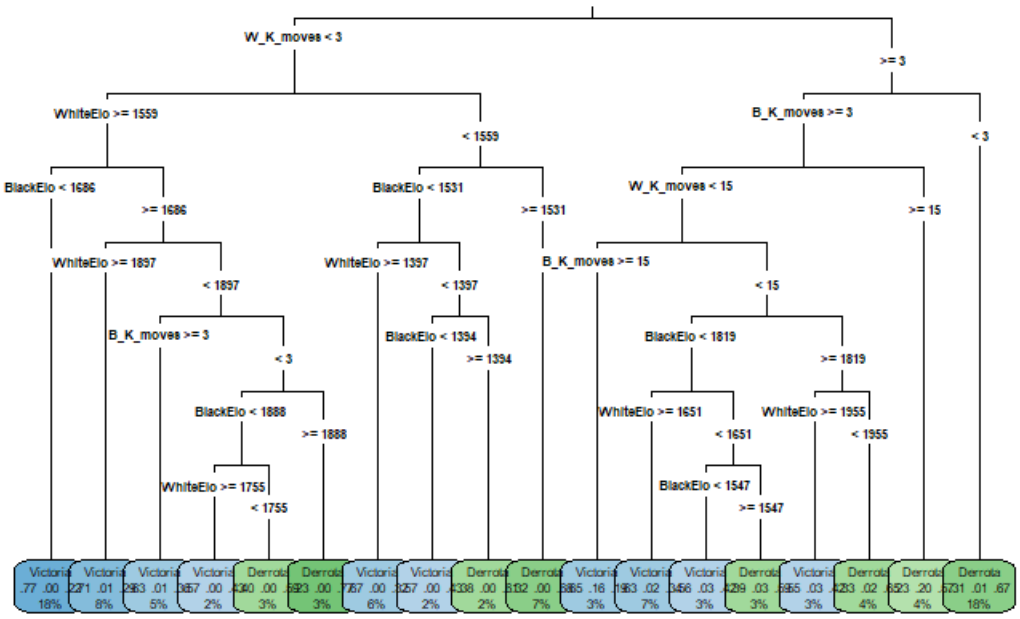

Figura 6.3: Árbol de decisión para partidas Bullet

Terminando el apartado de reglas de decisión con la tabla 6.3 ya se tienen conclusiones algo más variadas. Para empezar no existe la misma regla que se ha dado en los casos anteriores para las blancas de que si estas movían un número muy reducido de veces su rey y las negras lo movían bastante a lo largo de la partida estas ganaban con una alta probabilidad pero sí existe una regla parecida: $W\_K\_moves \geq 3, B\_K\_moves \geq 3, W\_K\_moves < 15, B\_K\_moves \geq 15$ con una probabilidad de ganar para las blancas del 65 %. Es interesante también decir que las negras sí tienen la misma regla que se ha presentado anteriormente, esto es $W\_K\_moves \geq 3, B\_K\_moves < 3$ con una probabilidad de ganar del 67 %. Vemos también que el elo influye en la partida con reglas como $W\_K\_moves < 3, WhiteElo \geq 1559, BlackElo < 1686$ con una precisión del 77 %.

Tabla 6.3: Reglas de decisión para partidas Bullet

Regla	Aciertos (%)	Total (%)	Resultado
W_K_moves < 3, WhiteElo ≥ 1559, BlackElo < 1686	77	18	Victoria
W_K_moves < 3, WhiteElo ≥ 1559, BlackElo ≥ 1686, WhiteElo ≥ 1897	70	8	Victoria
W_K_moves < 3, WhiteElo ≥ 1559, BlackElo ≥ 1686, WhiteElo < 1897, B_K_moves ≥ 3	63	5	Victoria
W_K_moves < 3, WhiteElo ≥ 1559, BlackElo ≥ 1686, WhiteElo < 1897, B_K_moves < 3, BlackElo < 1888, WhiteElo ≥ 1755	57	2	Victoria
W_K_moves < 3, WhiteElo ≥ 1559, BlackElo ≥ 1686, WhiteElo < 1897, B_K_moves < 3, BlackElo < 1888, WhiteElo < 1755	59	3	Derrota
W_K_moves < 3, WhiteElo ≥ 1559, BlackElo ≥ 1686, WhiteElo < 1897, B_K_moves < 3, BlackElo ≥ 1888	63	7	Derrota
W_K_moves < 3, WhiteElo < 1559, BlackElo < 1531, WhiteElo ≥ 1397	67	9	Victoria
W_K_moves < 3, WhiteElo < 1559, BlackElo < 1531, WhiteElo < 1397, BlackElo < 1394	57	4	Victoria
W_K_moves < 3, WhiteElo < 1559,			

Tabla 6.3: Reglas de decisión (cont.)

Regla	Aciertos (%)	Total (%)	Resultado
BlackElo $<$ 1531, WhiteElo $<$ 1397, BlackElo $\geq$ 1394	66	5	Derrota
W_K_moves $<$ 3, WhiteElo $<$ 1559, BlackElo $\geq$ 1531	67	6	Derrota
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $<$ 15, B_K_moves $\geq$ 15	65	11	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $<$ 15, B_K_moves $<$ 15, BlackElo $<$ 1819, WhiteElo $>$ 1651	63	8	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $<$ 15, B_K_moves $<$ 15, BlackElo $<$ 1819, WhiteElo $<$ 1651, BlackElo $<$ 1547	56	6	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $<$ 15, B_K_moves $<$ 15, BlackElo $<$ 1819, WhiteElo $<$ 1651, BlackElo $>$ 1547	59	7	Derrota
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $<$ 15, B_K_moves $<$ 15, BlackElo $\geq$ 1819, WhiteElo $\geq$ 1955	55	9	Victoria
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $<$ 15, B_K_moves $<$ 15, BlackElo $\geq$ 1819, WhiteElo $<$ 1955	65	8	Derrota
W_K_moves $\geq$ 3, B_K_moves $\geq$ 3, W_K_moves $\geq$ 15	57	10	Derrota

Tabla 6.3: Reglas de decisión (cont.)

Regla	Aciertos (%)	Total (%)	Resultado
W_K_moves ≥ 3, B_K_moves < 3	67	12	Derrota

6.2. Reglas de asociación

Para el apartado de reglas de asociación se va a comenzar cada modo de juego con un soporte relativamente alto y luego se va a ir afinando buscando incrementar la confianza de las reglas obtenidas para cada modo de juego. Se ha intentado que todos los modos de juego y que tanto para las negras como para las blancas tengan los mismos soportes y confianza pero esto no ha sido posible debido a la diferencia de la cantidad de reglas obtenidas (se ha encontrado que, en general, las negras necesitan soportes y mínimos de confianza más bajos para encontrar el mismo número de reglas, algo muy llamativo puesto que son las segundas en mover y comienzan en desventaja).

Es necesario aclarar que lo que se ha buscado en cada paso es qué reglas dan la victoria a las blancas y qué reglas dan victoria a las negras por lo que la columna que hace mención al Consecuente de la regla se ha eliminado al carecer de valor (su valor es siempre el mismo, es decir, Resultado=Victoria para todas las reglas de las blancas y Resultado=Derrota para todas las reglas de las negras). Sobre el empate, al igual que con las reglas de decisión no se ha eliminado a pesar de que no se ha intentado predecir porque eliminarlo aunque mejoraría la precisión del modelo sería falsear los resultados. También hay que mencionar que las variables han sido discretizadas por el método explicado en el capítulo 2.

6.2.1. Rated Blitz Game

Victoria para las blancas

Observando la tabla 6.4 y comenzando con la primera regla de todas, vemos que aquellas partidas en las que el rey de las blancas se ha movido entre 1 y 3 veces y que el rey de las negras se ha movido entre 4 y 69 veces la confianza para ganar es de 74.48. La razón por la que esto es así podría ser que el rey de las negras está sufriendo muchos jaques y la partida le queda poco para terminar. Si observamos las reglas siguientes vemos que prácticamente es la misma regla con el añadido del enroque. El valor lift es de 1.4426

lo que significa que la regla es bastante mejor que pura aleatoriedad.

Tabla 6.4: Reglas de asociación para las blancas en Blitz con soporte de 0.1 y confianza de 0.7

Antecedente	soporte	confianza	cobertura	lift
W_K_moves=[1,4), B_K_moves=[4,69]	0.11	0.74	0.15	1.44
W_K_moves=[1,4), B_K_moves=[4,69], B_O_moves=[0,1]	0.11	0.74	0.15	1.44
W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,69]	0.11	0.74	0.15	1.44
W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,69], B_O_moves=[0,1]	0.11	0.74	0.15	1.44

Continuando con la tabla 6.5. vemos que disminuyendo el soporte a 0.01 y aumentando la confianza a 0.85 obtenemos reglas distintas. Comentando la primera regla vemos que cuando la torre de las negras se mueve en el intervalo [0,3) dado que el elo de las blancas está en el intervalo [1720, 2560] y el de las negras se encuentra entre [782.1, 1510] la confianza para ganar las blancas es de casi el 90 %. Esta regla básicamente está confirmando que un jugador con un elo más alto tiene unas probabilidades de ganar muy altas (con un valor lift de 1.735, dando resultados muy superiores a lo esperados por aleatoriedad). La segunda regla es prácticamente igual pero en vez de utilizar la torre de las negras, sus alfiles.

Considerando las reglas siguientes vemos la presencia de los mismos intervalos de elo mencionados anteriormente junto con algunos añadidos (como la consideración del número de movimientos totales de la partida, expresado por la variable NMoves). Debido al funcionamiento del sistema elo dentro del ajedrez, es bastante seguro afirmar que estas reglas son una confirmación de que un jugador tenga un elo mucho más alto que su oponente hace que sus posibilidades de ganar aumenten considerablemente.

Tabla 6.5: Reglas de asociación para las blancas en Blitz con soporte 0.01 y confianza 0.85

Antecedente	soporte	confianza	cobertura	lift
B_R_moves=[0,3), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.90	0.01	1.73
B_B_moves=[0,4), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.73
NMoves=[3,27), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.72
W_K_moves=[1,4), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.02	0.89	0.02	1.72
B_O_moves=[0,1], B_R_moves=[0,3), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.90	0.01	1.73
W_O_moves=[0,1], B_R_moves=[0,3), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.90	0.01	1.73
B_B_moves=[0,4), B_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.73
W_O_moves=[0,1], B_B_moves=[0,4), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.73
NMoves=[3,27), B_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.72
NMoves=[3,27), W_O_moves=[0,1],				

Antecedente	soporte	confianza	cobertura	lift
WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.72
W_K_moves=[1,4), B_K_moves=[4,69], WhiteElo=[1.53e+03,1.72e+03), BlackElo=[782,1.51e+03]	0.02	0.87	0.02	1.69
W_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.02	0.89	0.02	1.72
W_K_moves=[1,4), W_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.02	0.89	0.02	1.72
W_O_moves=[0,1], B_O_moves=[0,1], B_R_moves=[0,3), WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.90	0.01	1.73
W_O_moves=[0,1], B_B_moves=[0,4), B_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.73
NMoves=[3,27), W_O_moves=[0,1], B_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.01	0.89	0.01	1.72
W_K_moves=[1,4), B_K_moves=[4,69], B_O_moves=[0,1], WhiteElo=[1.53e+03,1.72e+03), BlackElo=[782,1.51e+03]	0.02	0.87	0.02	1.69
W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,69], WhiteElo=[1.53e+03,1.72e+03),				

Antecedente	soporte	confianza	cobertura	lift
BlackElo=[782,1.51e+03]	0.02	0.87	0.02	1.69
W_K_moves=[1,4), W_O_moves=[0,1], B_O_moves=[0,1], WhiteElo=[1.72e+03,2.56e+03], BlackElo=[782,1.51e+03]	0.02	0.89	0.02	1.72

Victoria para las negras

Continuando con las reglas de asociación que le dan la victoria a las negras en la tabla 6.6 vemos que la primera regla es igual que la primera regla anterior con las blancas. Ahora si es el rey de las blancas el que se mueve mucho y el rey de las negras el que se mueve poco la victoria es para las negras con una confianza del 70 %. Observando las siguientes reglas vemos algunas modificaciones en ellas pero que son básicamente añadidos a esta primera regla. Por ejemplo, con la segunda regla vemos que si el rey de las blancas se mueve en un intervalo de [4, 69] movimientos, el rey de las negras en un intervalo de [1,4) y la reina de las negras en un intervalo de [5,61] movimientos la victoria es para las negras muy probablemente debido a la influencia de su reina en la partida.

Las siguientes reglas al igual que en apartados anteriores son añadidos a esta regla inicial pero teniendo en cuenta otros factores como el enroque.

Tabla 6.6: Reglas de asociación para las negras en Blitz con un soporte de 0.05 y confianza de 0.7

Antecedente	soporte	confianza	cobertura	lift
W_K_moves=[4,69], B_K_moves=[1,4)	0.10	0.71	0.14	1.57
W_K_moves=[4,69], B_K_moves=[1,4), B_Q_moves=[5,61]	0.06	0.73	0.08	1.61
W_K_moves=[4,69], B_K_moves=[1,4), B_O_moves=[0,1]	0.10	0.71	0.14	1.57
W_K_moves=[4,69], W_O_moves=[0,1], B_K_moves=[1,4)	0.10	0.71	0.14	1.57
W_K_moves=[4,69], B_K_moves=[1,4), B_O_moves=[0,1), B_Q_moves=[5,61]	0.06	0.73	0.08	1.61
W_K_moves=[4,69], W_O_moves=[0,1), B_K_moves=[1,4), B_Q_moves=[5,61]	0.06	0.73	0.08	1.61
W_K_moves=[4,69], W_O_moves=[0,1), B_K_moves=[1,4), B_O_moves=[0,1)	0.10	0.71	0.14	1.57
W_K_moves=[4,69], W_O_moves=[0,1), B_K_moves=[1,4), B_O_moves=[0,1), B_Q_moves=[5,61]	0.06	0.73	0.08	1.61

Terminando este modo de juego con la tabla 6.7 vemos rápidamente en la segunda regla una confirmación aún más fuerte de la influencia del elo a la hora de ganar la partida, dando una confianza de 0.80 para la victoria de las negras cuando el elo de las blancas se encuentra en el intervalo [784, 1530) y el de las negras en el intervalo [1700, 2560].

Más allá de la influencia del elo o de que las blancas hayan tenido que mover mucho su rey en la partida no se aprecia información relevante.

Tabla 6.7: Reglas de asociación para las negras en Blitz con un soporte de 0.01 y una confianza de 0.8

Antecedente	soporte	confianza	lift
NMoves=[3,27), W_K_moves=[4,69]	0.01	0.82	1.82
WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.02	0.81	1.78
NMoves=[3,27), W_K_moves=[4,69], B_O_moves=[0,1]	0.01	0.82	1.82

Antecedente	soporte	confianza	lift
NMoves=[3,27), W_K_moves=[4,69], W_O_moves=[0,1]	0.01	0.82	1.82
W_N_moves=[4,7), WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.01	0.81	1.80
B_K_moves=[1,4), WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.01	0.85	1.88
B_O_moves=[0,1], WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.02	0.81	1.78
W_O_moves=[0,1], WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.02	0.81	1.78
NMoves=[3,27), W_K_moves=[4,69], W_O_moves=[0,1], B_O_moves=[0,1]	0.01	0.82	1.82
W_K_moves=[4,69], B_K_moves=[1,4), WhiteElo=[1.53e+03,1.72e+03), BlackElo=[1.7e+03,2.56e+03]	0.01	0.81	1.80
W_N_moves=[4,7), B_O_moves=[0,1], WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.01	0.81	1.80
W_N_moves=[4,7), W_O_moves=[0,1], WhiteElo=[784,1.53e+03), BlackElo=[1.7e+03,2.56e+03]	0.01	0.81	1.80

6.2.2. Rated Rapid Game

Victoria para las blancas

Vemos en la tabla 6.8 que con un soporte de 0.1 y confianza mínima de 0.74 vemos que las partidas en las que el rey de las negras realiza muchos movimientos mientras que el de las blancas hace muy pocos la victoria es para las blancas con una confianza de 0.74. Parece ser que de este modo de juego vamos a sacar las mismas conclusiones que en el modo de juego anterior, incluso con valores de *lift* similares.

Tabla 6.8: Reglas de asociación para las blancas en Rapid con 0.74 de confianza y un soporte de 0.1

Antecedente	soporte	confianza	cobertura	lift
W_K_moves=[1,4), B_K_moves=[4,63]	0.12	0.79	0.15	1.56
W_K_moves=[1,4), B_K_moves=[4,63], B_O_moves=[0,1]	0.12	0.79	0.15	1.56
W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,63]	0.12	0.79	0.15	1.56
W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,63], B_O_moves=[0,1]	0.12	0.79	0.15	1.56

Observando la tabla 6.9, tenemos que incrementando la confianza a 0.85 y disminuyendo el soporte a 0.03 vemos la presencia de la reina blanca en las 4 reglas generadas con un intervalo amplio de movimientos que ha tenido. Aparte de esto, vemos que se sigue manteniendo la presencia de que el rey de las blancas se haya movido muy poco y el de las negras haya tenido que realizar muchos movimientos. También vemos la presencia del elo de las negras para todas las reglas generadas en el intervalo [780, 1480].

Tabla 6.9: Reglas de asociación para las blancas en Rapid con 0.03 de soporte y confianza 0.85

Antecedente	soporte	confianza	cobertura	lift
W_K_moves=[1,4), W_Q_moves=[5,64), B_K_moves=[4,63], BlackElo=[780,1.48e+03]	0.03	0.86	0.04	1.70
W_K_moves=[1,4), W_Q_moves=[5,64), B_K_moves=[4,63], B_O_moves=[0,1], BlackElo=[780,1.48e+03]	0.03	0.86	0.04	1.70
W_K_moves=[1,4), W_O_moves=[0,1], W_Q_moves=[5,64), B_K_moves=[4,63], BlackElo=[780,1.48e+03]	0.03	0.86	0.04	1.70
W_K_moves=[1,4), W_O_moves=[0,1], W_Q_moves=[5,64), B_K_moves=[4,63], B_O_moves=[0,1], BlackElo=[780,1.48e+03]	0.03	0.86	0.04	1.70

Victoria para las negras

Viendo la tabla 6.10 en este apartado ya estamos empezando a ver algunas diferencias de los resultados respecto al resto al resto de reglas creadas. Vemos que en este caso no se tiene en cuenta que el rey de las negras se haya tenido que mover poco o mucho (se ha probado incluso a bajar el nivel de confianza mínimo incluso hasta 0.60 y no se ha encontrado ninguna regla en la que aparezca el antecedente de que el rey del jugador estudiado se haya tenido que mover poco y el rey de las blancas se haya movido mucho a diferencia del resto de apartados) pero sí se sigue la regla del antecedente de que el rey del oponente se mueva mucho con el añadido de que la reina de las negras haya realizado un número de movimientos entre [5, 65]. Al igual que en las reglas anteriores, el enroque de ambos jugadores no parece ser muy relevante al dar el intervalo de [0,1].

Tabla 6.10: Reglas de asociación para las negras en Rapid con 0.03 de soporte y confianza 0.64

Antecedente	soporte	confianza	cobertura	lift
W_K_moves=[4,70], B_Q_moves=[5,65]	0.11	0.65	0.16	1.42
W_K_moves=[4,70], B_O_moves=[0,1], B_Q_moves=[5,65]	0.11	0.65	0.16	1.42
W_K_moves=[4,70], W_O_moves=[0,1], B_Q_moves=[5,65]	0.11	0.65	0.16	1.42
W_K_moves=[4,70], W_O_moves=[0,1], B_O_moves=[0,1], B_Q_moves=[5,65]	0.11	0.65	0.16	1.42

Terminando con la tabla 6.11 ahora sí nos aparece en una misma regla que el rey propio haga un número bajo de movimientos ([1,4)) y el del oponente un número alto ([4,70]) solo que a diferencia de los casos anteriores esta vez hemos tenido que bajar el soporte hasta 0.02 para que aparezca, sacrificando el número de partidas en las que se aplica la regla. Aparte de esto, vemos en las reglas la presencia de algunas piezas, como un número importante de movimientos de la reina de las negras (esto es bastante destacable puesto que ya estaba en las reglas anteriores).

Tabla 6.11: Reglas de asociación para las negras en Rapid con 0.02 de soporte y confianza 0.85

Antecedente	soporte	confianza	cobertura	lift
W_K_moves=[4,70], B_K_moves=[1,4), B_Q_moves=[5,65], WhiteElo=[803,1.49e+03]	0.03	0.86	0.03	1.88
W_K_moves=[4,70], W_R_moves=[2,6), B_K_moves=[1,4), WhiteElo=[803,1.49e+03]	0.02	0.85	0.02	1.86
W_K_moves=[4,70], B_B_moves=[5,55], B_K_moves=[1,4), WhiteElo=[803,1.49e+03]	0.03	0.85	0.03	1.86
W_K_moves=[4,70], B_K_moves=[1,4), B_O_moves=[0,1], B_Q_moves=[5,65], WhiteElo=[803,1.49e+03]	0.03	0.86	0.03	1.88
W_K_moves=[4,70], W_O_moves=[0,1], B_K_moves=[1,4), B_Q_moves=[5,65], WhiteElo=[803,1.49e+03]	0.03	0.86	0.03	1.88
W_K_moves=[4,70], W_R_moves=[2,6), B_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[803,1.49e+03]	0.02	0.85	0.02	1.86
W_K_moves=[4,70], W_O_moves=[0,1], W_R_moves=[2,6), B_K_moves=[1,4), WhiteElo=[803,1.49e+03]	0.02	0.85	0.02	1.86
W_K_moves=[4,70], B_B_moves=[5,55], B_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[803,1.49e+03]	0.03	0.85	0.03	1.86
W_K_moves=[4,70], W_O_moves=[0,1], B_B_moves=[5,55], B_K_moves=[1,4), WhiteElo=[803,1.49e+03]	0.03	0.85	0.03	1.86
W_K_moves=[4,70], W_O_moves=[0,1], B_K_moves=[1,4), B_O_moves=[0,1], B_Q_moves=[5,65], WhiteElo=[803,1.49e+03]	0.03	0.86	0.03	1.88
W_K_moves=[4,70], W_O_moves=[0,1], W_R_moves=[2,6), B_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[803,1.49e+03]	0.02	0.85	0.02	1.86
W_K_moves=[4,70], W_O_moves=[0,1], B_B_moves=[5,55], B_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[803,1.49e+03]	0.03	0.85	0.03	1.86

6.2.3. Rated Bullet Game

Victoria para las blancas

En la tabla 6.12 vemos una regla que antes no aparecía inmediatamente en el anterior para las negras, y es la regla de que si un jugador mueve muy poco el rey y el otro hace un número grande de movimientos, el jugador que lo ha movido muy poco tiene bastantes probabilidades de ganar.

Tabla 6.12: Reglas de asociación para las blancas en Bullet con un soporte de 0.03 y una confianza de 0.7

Antecedente	**Soporte**	**Confianza**	**Cobertura**	**Lift**
{W_K_moves=[1,4), B_K_moves=[4,67]}	0.12	0.71	0.17	1.36
{W_K_moves=[1,4), B_K_moves=[4,67], B_O_moves=[0,1]}	0.12	0.71	0.17	1.36
{W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,67]}	0.12	0.71	0.17	1.36
{W_K_moves=[1,4), W_O_moves=[0,1], B_K_moves=[4,67], B_O_moves=[0,1]}	0.12	0.71	0.17	1.36

Con la tabla 6.13 se tiene que bajando el soporte ahora hasta 0.03 y subiendo la confianza a 0.8 vemos en la primera regla una que no nos aparecía anteriormente para este modo de juego, y es que un jugador de un elo mucho más alto va a tener una probabilidad de ganar considerablemente más alta, añadiendo incluso en este caso que las torres negras se hayan movido en un intervalo de [0,2] movimientos (podría ser un indicativo de que los jugadores de las negras de un elo bastante inferior al de su oponente tienden a hacer un desarrollo tardío).

Observando las siguientes reglas, vemos modificaciones de esta pero con el añadido del movimiento de otras piezas como que el rey de las blancas haya realizado pocos movimientos.

Tabla 6.13: Reglas de asociación para las blancas en Bullet con un soporte de 0.03 y una confianza de 0.8

Antecedente	**Soporte**	**Confianza**	**Cobertura**	**Lift**
{B_R_moves=[0,3), WhiteElo=[1.59e+03,1.78e+03), BlackElo=[825,1.57e+03]}	0.03	0.80	0.04	1.54
{W_K_moves=[1,4), WhiteElo=[1.78e+03,2.58e+03], BlackElo=[1.57e+03,1.77e+03]}	0.04	0.80	0.05	1.54
{B_O_moves=[0,1], B_R_moves=[0,3), WhiteElo=[1.59e+03,1.78e+03), BlackElo=[825,1.57e+03]}	0.03	0.80	0.04	1.54
{W_O_moves=[0,1], B_R_moves=[0,3), WhiteElo=[1.59e+03,1.78e+03), BlackElo=[825,1.57e+03]}	0.03	0.80	0.04	1.54
{W_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[1.78e+03,2.58e+03], BlackElo=[1.57e+03,1.77e+03]}	0.04	0.80	0.05	1.54
{W_K_moves=[1,4), W_O_moves=[0,1], WhiteElo=[1.78e+03,2.58e+03], BlackElo=[1.57e+03,1.77e+03]}	0.04	0.80	0.05	1.54
{W_O_moves=[0,1], B_O_moves=[0,1], B_R_moves=[0,3), WhiteElo=[1.59e+03,1.78e+03), BlackElo=[825,1.57e+03]}	0.03	0.80	0.04	1.54
{W_K_moves=[1,4), W_O_moves=[0,1], B_O_moves=[0,1], WhiteElo=[1.78e+03,2.58e+03], BlackElo=[1.57e+03,1.77e+03]}	0.04	0.80	0.05	1.54

Victoria para las negras

En este caso con la tabla 6.14 volvemos a ver la regla repetida tanto aquí como en árboles de decisión de los movimientos de los reyes.

Tabla 6.14: Reglas de asociación para las negras en Bullet con un soporte de 0.1 y una confianza de 0.66

Antecedente	**Soporte**	**Confianza**	**Cobertura**	**Lift**
{W_K_moves=[4,69], B_K_moves=[1,4]}	0.10	0.66	0.15	1.45
{W_K_moves=[4,69], B_K_moves=[1,4), B_O_moves=[0,1]}	0.10	0.66	0.15	1.45
{W_K_moves=[4,69], W_O_moves=[0,1], B_K_moves=[1,4]}	0.10	0.66	0.15	1.45
{W_K_moves=[4,69], W_O_moves=[0,1], B_K_moves=[1,4), B_O_moves=[0,1]}	0.10	0.66	0.15	1.45

En 6.15 bajando el soporte a 0.03 y subiendo la confianza a 0.75 vemos otra vez la aparición del elo. Aparte de esto, el algoritmo introduce la presencia de algunas piezas como la torre o la reina, piezas que podrían haber afectado bastante el resultado de la partida.

Tabla 6.15: Reglas de asociación para las negras en Bullet con un soporte de 0.03 y una confianza de 0.75

Antecedente	**Soporte**	**Confianza**	**Cobertura**	**Lift**
{B_K_moves=[1,4), WhiteElo=[1.59e+03,1.78e+03), BlackElo=[1.77e+03,2.6e+03]}	0.03	0.76	0.05	1.65

Antecedente	**Soporte**	**Confianza**	**Cobertura**	**Lift**
{B_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[1.59e+03,1.78e+03), BlackElo=[1.77e+03,2.6e+03]}	0.03	0.76	0.05	1.65
{W_O_moves=[0,1], B_K_moves=[1,4), WhiteElo=[1.59e+03,1.78e+03), BlackElo=[1.77e+03,2.6e+03]}	0.03	0.76	0.05	1.65
{W_K_moves=[4,69], B_K_moves=[1,4), B_Q_moves=[4,56], BlackElo=[1.77e+03,2.6e+03]}	0.03	0.75	0.04	1.65
{W_O_moves=[0,1], B_K_moves=[1,4), B_O_moves=[0,1], WhiteElo=[1.59e+03,1.78e+03), BlackElo=[1.77e+03,2.6e+03]}	0.03	0.76	0.05	1.65
{W_K_moves=[4,69], B_K_moves=[1,4), B_O_moves=[0,1], B_Q_moves=[4,56], BlackElo=[1.77e+03,2.6e+03]}	0.03	0.75	0.04	1.65
{W_K_moves=[4,69], W_O_moves=[0,1], B_K_moves=[1,4), B_Q_moves=[4,56], BlackElo=[1.77e+03,2.6e+03]}	0.03	0.75	0.04	1.65
{W_K_moves=[4,69], W_O_moves=[0,1], B_K_moves=[1,4), B_O_moves=[0,1], B_Q_moves=[4,56], BlackElo=[1.77e+03,2.6e+03]}	0.03	0.75	0.04	1.65

6.3. Regresión logística

Se va a terminar la parte de minería de datos utilizando la regresión logística para intentar extraer conocimiento de la base de datos y luego con la validación cruzada medir la generalización del algoritmo con la base de datos y comprobar si existe sobreajuste.

Para obtener los coeficientes se emplea el 100 % de los datos

6.3.1. Rated Blitz Game

Comenzando con el apartado de Rated Blitz Game, habiéndose hecho la tabla inferior con validación cruzada se ha obtenido una precisión del 71 %.

Variable	**Estimate**	**Std. Error**	**z value**	**Pr**$(>\|z\|)$
(Intercept)	-5.662e-02	4.229e-02	-1.339	0.18067
NMoves	5.988e-02	2.041e-03	29.344	$<$ 2e-16
W_B_moves	1.590e-02	2.506e-03	6.344	2.24e-10
W_K_moves	-9.348e-02	2.288e-03	-40.854	$<$ 2e-16
W_N_moves	6.822e-03	2.271e-03	3.004	0.00267
W_O_moves	4.101e-02	1.379e-02	2.973	0.00295
W_Q_moves	5.193e-02	2.346e-03	22.138	$<$ 2e-16
W_R_moves	5.683e-04	2.258e-03	0.252	0.80131
B_B_moves	-1.020e-01	2.491e-03	-40.960	$<$ 2e-16
B_K_moves	5.614e-02	2.361e-03	23.781	$<$ 2e-16
B_N_moves	-9.234e-02	2.251e-03	-41.028	$<$ 2e-16
B_O_moves	-5.492e-02	1.282e-02	-4.286	1.82e-05
B_Q_moves	-1.314e-01	2.331e-03	-56.378	$<$ 2e-16
B_R_moves	-8.426e-02	2.228e-03	-37.809	$<$ 2e-16
WhiteElo	4.490e-03	3.487e-05	128.750	$<$ 2e-16
BlackElo	-4.446e-03	3.428e-05	-129.685	$<$ 2e-16

Tabla 6.16: Estimación de los coeficientes del modelo de regresión para Rated Blitz Game

Estudiando los pesos de la tabla 6.16, vemos que por ejemplo, para la variable B_K_moves su valor es 5.614e-02 lo que significa que por cada unidad que se incrementa la variable B_K_moves se espera que el logaritmo de odds de la variable Result se incremente en promedio $e^{5.614\times10^{-2}} = 1.0577$. Es muy llamativo ver que la variable W_R_moves es la única que no es estadísticamente significativa aunque dado su bajo valor en la columna Estimate no es algo relevante en la información extraída del modelo.

El p-valor es casi 0 lo que significa que el modelo en conjunto sí es significativo. Se han comparado los resultados del modelo antes y después de eliminar las partidas en las que los jugadores empatan y los resultados son los mismos por lo que por simplicidad se

ha optado por quedarse con el modelo sin Empates.

Es importante también recalcar que aunque todos los pesos sean bajos, unos son claramente bastante mayor que otros. Por ejemplo, para B_Q_moves su peso es de $-1.314 * 10^{-1}$ que es claramente bastate superior al de W_Q_moves de $5.193 * 10^{-2}$, concretamente 2.53 veces superior. Esto nos indica que si el resto de variables del modelo se mantienen constantes, al aumentar el valor de B_Q_moves (ya que su peso es significativamente grande respecto al resto y negativo) tiende a que el valor de clase de salida sea cero (pierden las blancas).

6.3.2. Rated Rapid Game

Como en las técnicas anteriores, se continuará con el modo de juego Rated Rapid Game. Se ha generado un modelo de regresión logística utilizando el 100 % de los datos (la precisión en validación cruzada es del 73.8 %) y su p-valor asociado es casi 0. El modelo no presenta ningún cambio si se eliminan las partidas en las que se empata por lo que se han dejado eliminadas por simplicidad.

Variable	**Estimate**	**Std. Error**	**z value**	**Pr**$(>\|z\|)$
(Intercept)	-8.299e-02	4.874e-02	-1.703	0.08862
NMoves	5.066e-02	2.225e-03	22.769	$<$ 2e-16
W_B_moves	3.527e-02	2.780e-03	12.683	$<$ 2e-16
W_K_moves	-1.295e-01	2.615e-03	-49.521	$<$ 2e-16
W_N_moves	1.749e-02	2.506e-03	6.980	2.95e-12
W_O_moves	6.459e-02	1.342e-02	4.813	1.49e-06
W_Q_moves	5.511e-02	2.495e-03	22.087	$<$ 2e-16
W_R_moves	-7.799e-03	2.495e-03	-3.126	0.00177
B_B_moves	-1.135e-01	2.753e-03	-41.244	$<$ 2e-16
B_K_moves	1.177e-01	2.710e-03	43.430	$<$ 2e-16
B_N_moves	-9.415e-02	2.459e-03	-38.292	$<$ 2e-16
B_O_moves	-1.026e-01	1.277e-02	-8.032	9.63e-16
B_Q_moves	-1.256e-01	2.490e-03	-50.444	$<$ 2e-16
B_R_moves	-7.022e-02	2.461e-03	-28.536	$<$ 2e-16
WhiteElo	4.118e-03	3.538e-05	116.384	$<$ 2e-16
BlackElo	-4.030e-03	3.490e-05	-115.471	$<$ 2e-16

Tabla 6.17: Estimación de los coeficientes del modelo de regresión

Continuando con la tabla 6.17, vemos en ella los coeficientes estimados para el modelo de regresión en Rated Rapid Game con unas estimaciones para las variables similares a las anteriores, aunque llama la atención que se ha mantenido el signo del valor estimado para todas ellas (probablemente debido a que cuando un jugador mueve una pieza, de media, ese movimiento le empuja a ganar).

Es muy destacable también que ahora la variable W_R_moves sí es estadísticamente significativa a diferencia del modelo anterior, aunque sea una de las variables con los pesos más bajos. Al igual que en la tabla anterior, vemos que las variables W_K_moves y B_K_moves tienen los pesos más altos y que, por ejemplo en el primer caso, su peso de $-1.295 * 10^{-}01$ es 16 veces más grande que el de W_R_moves. La regresión logística al igual que las técnicas anteriores dan un claro peso a las variables W_K_moves y B_K_moves bastante más grande que el resto, y si el resto de variables se mantuvieran constantes a excepción de una de estas dos, el resultado de la partida tendería hacia el signo de la variable elegida.

6.3.3. Rated Bullet Game

Terminando con el modo de juego Rated Bullet Game, al igual que en los casos anteriores se va a utilizar validación cruzada (que en este caso la precisión ha sido del 69 %) y luego se generan los coeficientes con el 100 % de los datos.

Variable	**Estimate**	**Std. Error**	**z value**	**Pr**$(>\|z\|)$
(Intercept)	-2.436e-01	5.024e-02	-4.849	1.24e-06
NMoves	7.883e-02	2.531e-03	31.146	$<$ 2e-16
W_B_moves	9.544e-03	3.139e-03	3.040	0.00236
W_K_moves	-7.639e-02	2.786e-03	-27.418	$<$ 2e-16
W_N_moves	4.122e-03	2.816e-03	1.464	0.14316
W_O_moves	-7.147e-02	1.668e-02	-4.284	1.84e-05
W_Q_moves	7.258e-02	2.926e-03	24.801	$<$ 2e-16
W_R_moves	1.904e-02	2.800e-03	6.802	1.03e-11
B_B_moves	-1.201e-01	3.145e-03	-38.190	$<$ 2e-16
B_K_moves	1.952e-03	2.839e-03	0.688	0.49176
B_N_moves	-1.064e-01	2.803e-03	-37.949	$<$ 2e-16
B_O_moves	2.311e-02	1.553e-02	1.489	0.13660
B_Q_moves	-1.728e-01	2.925e-03	-59.076	$<$ 2e-16
B_R_moves	-1.176e-01	2.785e-03	-42.226	$<$ 2e-16
WhiteElo	4.862e-03	3.971e-05	122.431	$<$ 2e-16
BlackElo	-4.769e-03	3.858e-05	-123.617	$<$ 2e-16

Tabla 6.18: Estimación de los coeficientes del modelo de regresión

Terminamos este apartado con la tabla 6.18., donde podemos ver los coefiicientes del modelo de regreresión logísitica para Rated Bullet Game. En este modelo vemos como a diferencia de los modelos anteriores que la variable del enroque de las negras deja de ser estadísticamente significativa, junto con el rey de las negras (B_K_moves), que curiosamente no podemos decir lo mismo con el rey de las blancas (W_K_moves). También vemos como a diferencia de los modelos anteriores la variable que hace referencia al número de veces que ha movido su caballo las blancas (W_N_moves) no es estadísticamente significativa. En cualquier caso, los pesos de estas variables al igual que en los modelos anteriores siguen siendo bastante bajos, pero con importantes diferencias entre ellos como

en el caso de B_R_moves que tiene uno de los pesos más altos que comparado con uno de los más bajos (como el de W_N_moves) resulta que es casi 30 veces superior.

La diferencia de que unas variables hayan dejado de ser significativas comparadas con los modos anteriores podría ser porque la diferencia de tiempo es muy grande como para cometer los mismos movimientos en una posición dada. En una partida con tiempos más bajos, los errores son más proclives y evitar que tu rival enroque es más fácil.

Capítulo 7

Análisis conjunto

En el siguiente capítulo se va a hacer una puesta en común de la información que se ha extraído de las tres técnicas de minería de datos que se han aplicado anteriormente. Se va a tener también un poco en cuenta las diferencias entre los distintos métodos, pero se va a dar una prioridad mucho más grande a lo que dicen en común.

7.1. Rated Blitz Game

Comenzando con las partidas Blitz, del árbol de decisión obtenemos la siguiente regla: $W\_K\_moves < 3, B\_K\_moves \geq 2$ que con un 71 % de aciertos da la victoria a las blancas en un 28 % de las partidas (un número bastante grande en el que se podría aplicar). Esto es muy similar a la regla de asociación $W\_K\_moves = [1, 4), B\_K\_moves = [4, 69]$ que da la victoria a las blancas con un soporte de 0.11 y una confianza de 0.74, solo que la cobertura de esta última regla es de solo el 15 % de las partidas. Si tenemos en cuenta los resultados de la regresión logística, vemos que la variable B_K_moves tiene un valor positivo lo que significa que incrementos de esta variable acercan a las blancas a la victoria aunque en este caso el incremento es bastante pequeño. También podemos extraer conocimiento con el nivel de elo emparejando la regla del árbol de decisión $W\_K\_moves < 3, B\_K\_moves < 2, BlackElo < 1501, WhiteElo \geq 1501$ que le da la victoria a las blancas en un 77 % de las partidas (aunque su aplicabilidad es más limitada, de tan solo el 4 %) con la regla de asociación $W\_K\_moves = [1, 4), WhiteElo = [1.72e + 03, 2.56e + 03], BlackElo =$

$[782, 1.51e+03]$ que le da la victoria a las blancas con una confianza del 89 % solo que una cobertura también muy reducida (0.01).

Continuando con las negras para el mismo modo de juego, tenemos la regla de decisión $W\_K\_moves \geq 3, B\_K\_moves < 4$ que da la victoria a las negras en un 69 % de las partidas con un 20 % de ellas en las que se puede aplicar. Es comparable con la regla de asociación $W\_K\_moves = [4, 69], B\_K\_moves = [1, 4)$ que da la victoria a las negras con una confianza de 71 % con una cobertura de 0.14. Teniendo en cuenta también lo mencionado anteriormente de que la variable WhiteElo es positiva en regresión logística y BlackElo negativa, existe también en esta ocasión la información de que para las negras cuando su rey se mueve muy poco y el de las blancas mucho, sus probabilidades de ganar aumentan considerablemente. Una de las diferencias entre las reglas de asociación y de las reglas de decisión es que estas últimas no tienen en cuenta los movimientos de las piezas a diferencia de las reglas de asociación, que estas por ejemplo han considerado los movimientos de la reina en algunas de ellas.

Hemos extraído en este trabajo que las partidas en las que las blancas hacen muy pocos movimientos de rey cuando las negras hacen un número considerable de ellos las blancas tienden a ganar la partida.

7.2. Rated Rapid Game

Considerando las partidas Rapid, del árbol de decisión obtenemos la misma regla para las blancas que en el apartado anterior: $W\_K\_moves < 3, B\_K\_moves \geq 3$ que pasa de un 71 % de aciertos a un 79 % de ellos en un 18 % de las partidas, regla que se asemeja a la siguiente del apartado de reglas de asociación: $W\_K\_moves = [1, 4), B\_K\_moves = [4, 63]$ que da la victoria a las blancas con un soporte de 0.79 y cobertura de 0.15. Si tenemos en cuenta también el elo, es más difícil sacar conclusiones aisladas para esto dada su mezcla con los movimientos en los árboles de decisión pero aún así, tenemos la siguiente regla:

$W\_K\_moves < 3, B\_K\_moves < 3, W\_K\_moves < 2, WhiteElo \geq 1505, BlackElo < 1542$ con un 79 % de victoria para las blancas. Esta regla es interesante porque establece un límite superior para BlackElo que es más grande que el inferior de WhiteElo pero da la victoria a las blancas probablemente debido a que un jugador puede tener buenas probabilidades de ganar la partida contra un jugador de un nivel muy ligeramente superior si este último juega con negras. Esta regla se asociaría con la siguiente de reglas de asociación: $NMoves = [3, 27), WhiteElo = [1.72e + 03, 2.56e + 03], BlackElo = [782, 1.51e + 03]$ que da la victoria a las blancas en un 79 % de las partidas aunque con un soporte del 0.01. Aún así, es un poco difícil de comparar esta regla con la asociada a árboles de decisión porque esta última no incluye la variable NMoves. Teniendo en cuenta ahora la regresión logística, esta favorece las blancas para las variables WhiteElo y B_K_moves y favorece las negras en las variables BlackElo y W_K_moves.

Teniendo en cuenta ahora la victoria para las negras, se le da importancia al elo con reglas como $W\_K\_moves < 3, WhiteElo < 1559, BlackElo \geq 1531$ que da la victoria a las negras en un 67 % de las partidas. Esta regla es interesante no solamente porque afirma la importancia del elo para ganar como en apartados anteriores, sino porque incluso si el rey de las blancas se ha movido menos de 3 veces, las negras ganan la partida reforzando así la importancia de un elo supeior en el resultado. Esta regla es muy similar a la regla de asociación $B\_K\_moves = [1, 4), WhiteElo = [784, 1.53e+03), BlackElo = [1.7e+03, 2.56e+03]$ que da la victoria a las negras con una confianza de 0.85. Si vemos los pesos generados por la regresión logística mencionados en el párrafo anterior, incrementos en el elo de las negras se asocia a mayores probabilidades para estas de ganar la partida (aunque como se ha mencionado en el capítulo anterior, son incrementos bastante ligeros).

7.3. Rated Bullet Game

Terminando con partidas Bullet, la regla para las blancas sobre el elo $W\_K\_moves < 3, WhiteElo \geq 1559, BlackElo < 1686$ con un 77 % de aciertos da la misma información que en los apartados anteriores respecto al elo. Sin embargo, existen aquí ya ciertas diferencias respecto al número de veces que mueve el rey las negras, sin estar del todo claro como afecta esto a las probabilidades de ganar de las blancas. Considerando por ejemplo la regla $W\_K\_moves = [1, 4), B\_K\_moves = [4, 67]$ que con una confianza de 0.71 da la victoria a las blancas en una cobertura del 17 %. y también se puede decir lo mismo de la importancia del elo con reglas como $W\_K\_moves = [1, 4), WhiteElo = [1.78e + 03, 2.58e + 03], BlackElo = [1.57e + 03, 1.77e + 03]$ que da la victoria a las blancas con una confianza de 0.80. De la regresión logística se puede comentar que los pesos asociados a B_K_moves, W_K_moves, BlackElo y WhiteElo son los mismos que para los modos de juego anteriores.

Con la victoria para las negras, se ve una clara importancia del elo en árboles de decisión con $W\_K\_moves < 3, WhiteElo < 1559, BlackElo \geq 1531$ que da un porcentaje de victorias del 67 %. Al igual que en el párrafo anterior, no se ve una influencia tan fuerte de un mayor número de movimientos del rey enemigo a excepción de cuando esta cantidad de movimientos es muy grande con $W\_K\_moves \geq 3, B\_K\_moves \geq 3, W\_K\_moves \geq 15$ que da la victoria a las negras en un 57 % de las partidas. Si nos vamos a las reglas de asociación, al igual que antes la importancia clara del movimiento de los reyes vuelve con reglas como $W\_K\_moves = [4, 69], B\_K\_moves = [1, 4]$ que da la victoria a las negras con una confianza de 0.66 y cobertura de 0.15. Si nos vamos al elo en reglas de asociación, tenemos $B\_K\_moves = [1, 4), WhiteElo = [1.59e + 03, 1.78e + 03), BlackElo = [1.77e + 03, 2.6e + 03]$ con confianza de 0.76 y cobertura de 0.05.

Capítulo 8

Conclusiones finales

El objetivo inicial de este trabajo era aprender no solamente técnicas de minería de datos sino todos los pasos, todo el procedimiento necesario tanto antes como después de aplicarlas para extraer conocimiento de una base de datos, aparte de empezar a aprender programación en python que pueda servirme en el día de mañana. En el momento de escribir estas líneas, puedo afirmar que estos objetivos se han completado con éxito.

Comenzamos eligiendo una base de datos de partidas de ajedrez. La elección de la base de datos no fue fácil porque en un principio no estaba claro si usar una base de datos de partidas de ajedrez de jugadores aficionados o de jugadores profesionales (se terminó eligiendo una base de jugadores aficionados porque al cometer estos más errores las conclusiones podrían ser más interesantes). Una vez elegida la base de datos, aprendí en python como transformarla desde un archivo .pgn a un dataframe funcional, en parte utilizando la librería chess.

Hecho esto, comencé a hacer el paso previo a aplicar los algoritmos de análisis exploratorio. Los conocimientos adquiridos en la carrera fueron útiles y ha sido muy interesante la decisión de eliminar las partidas con muy pocos movimientos en base al Mate del Loco (que como se mencionó era el jaque mate con el menor número de movimientos posible) en vez de simplemente eliminar partidas por lo que simplemente diga una medida estadística como el IQR.

Después de hacer el análisis exploratorio y análisis estadístico, me puse a aprender sobre las técnicas de minería de datos aplicadas en este trabajo. Empecé a investigar

sobre el tema por con la guía de mi tutor y los conocimientos que consideré importantes sobre ello los plasmé en el tema 2. Aprendí también como seleccionar variables con el algoritmo InformationGainRatio y la importancia de discretizar.

Finalmente, apliqué las técnicas de minería de datos sacando las conclusiones plasmadas en los capítulos 7 y 8. Sobre esto, finalmente no se sacaron conclusiones relevantes más allá de los aspectos triviales de la importancia del elo o que las partidas en las que un jugador hace muchos movimientos de su rey mientras que el otro muy pocos tiende a ganar el que realiza pocos.

Bibliografía

[1] Historia del ajedrez, Harold Murray (1913)

[2] https://www.houseofstaunton.com/history-of-chess

[3] https://www.mindmentorz.com/blog/2019/8/8/the-5-most-historical-instances-in-chess

[4] https://www.chess.com/article/view/the-ruy-lopez-chess-opening-explained

[5] https://www.chess.com/article/view/computers-and-chess—a-history

[6] Curso completo de ajedrez, Miguel Ilescas (2019)

[7] https://www.chess.com/es/terms/sistema-puntuacion-elo-ajedrez

[8] https://www.chess.com/es/terms/sistema-puntuacion-elo-ajedrez#how-does-elo-work

[9] https://lichess.org/

[10] Aperturas modernas en ajedrez, Nick de Firmian (2019)

[11] https://es.wikipedia.org/wiki/Lichess

[12] https://www.chess.com/terms/chess-pgn

[13] Decision tree methods: applications for classification and prediction, Yan-yan SONG et al., 2015

[14] https://thenewstack.io/cohens-kappa-what-it-is-when-to-use-it-and-how-to-avoid-its-pitfalls/

[15] https://dataheadhunters.com/academy/exploring-the-limits-of-decision-trees-depth-bias-and-variance/

[16] https://en.wikipedia.org/wiki/Association_rule_learning

[17] https://www.techtarget.com/searchbusinessanalytics/definition/association-rules-in-data-mining

[18] https://medium.com/@24littledino/association-mining-support-association-rules-and-confidence-60132a37e355

[19] https://medium.com/@24littledino/association-mining-support-association-rules-and-confidence-60132a37e355

[20] https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica

[21] https://medium.com/@lily_su/logistic-regression-accuracy-cross-validation-58d9eb58d6e6

[22] https://en.wikipedia.org/wiki/Information_gain_ratio

[23] https://es.wikipedia.org/wiki/Mate_del_loco

[24] https://vivaelsoftwarelibre.com/test-de-kolmogorov-smirnov-en-r/