Análisis de partidas de ajedrez mediante Data Mining

Rubén Márquez
Minería de datos
Universidad de Castilla-La Mancha
Ruben.Marquez@alu.uclm.com

Alberto Velasco
Minería de datos
Universidad de Castilla-La Mancha
Alberto.Velasco1@alu.uclm.com

Diego Pedregal
Minería de datos
Universidad de Castilla-La Mancha
Diego.Pedregal@alu.uclm.com

Antonio Manjavacas
Minería de datos
Universidad de Castilla-La Mancha
Antonio.Manjavacas@alu.uclm.com

1 Introducción

La minería de datos se define como un proceso utilizado para extraer, analizar e identificar patrones a partir de conjuntos masivos de datos. Un área donde esta disciplina ha alcanzado una gran relevancia en las últimas décadas ha sido en el contexto del ajedrez, donde, desde los primeros casos como *DeepBlue* [1], se han elaborado múltiples sistemas de inteligencia artificial basados en el análisis de partidas con distintos fines: jugar contra humanos, jugar contra otras máquinas, predecir resultados, analizar partidas, etc.

En base a la utilidad de la minería de datos en el contexto del ajedrez, hemos decidido aplicar técnicas de extracción de conocimiento sobre datos de partidas disponibles en la web.

En este documento se definirán los aspectos iniciales del problema a abordar: fuente de datos, descripción de estos y objetivos que se pretenden conseguir. Posteriormente, se abordará el proceso de recopilación y tratamiento de los datos, para finalmente concluir con los resultados obtenidos en base a los objetivos propuestos.

Tanto los datos como el código empleados son de libre acceso y pueden encontrarse en el siguiente repositorio: https://github.com/manjavacas/Data-Mining.

2 Fuente de datos

Para realizar este proyecto se ha elegido como fuente de datos el conjunto de partidas de ajedrez de la web *lichess.org* [2] correspondientes al mes de **septiembre de 2019**. De este conjunto de partidas se han obtenido las partidas correspondientes a los 200 mejores jugadores de la clasificación en modo clásico.

La extracción selectiva de los datos por fecha y mejores jugadores se ha llevado a cabo haciendo uso de la API `berserk` [3]. Todos los datos recopilados son de dominio público sin derechos reservados.

3 Descripción de los datos

El formato de cada uno de los registros del dataset es el siguiente:

- `Event`: evento en el cual se enmarca la partida.
- `Site`: enlace a la partida.
- `White`: jugador con blancas.
- `Black`: jugador con negras.
- `Result`: resultado de la partida.
- `UTCDate`: fecha UTC.
- `UTCTime`: hora UTC.
- `WhiteElo`: puntuación ELO del jugador con blancas.
- `BlackElo`: puntuación ELO del jugador con negras.
- `WhiteRatingDiff`: puntos obtenidos por blancas.
- `BlackRatingDiff`: puntos obtenidos por negras.
- `WhiteTitle`, `BlackTitle`: abreviación de la titulación de los jugadores (según estándar FIDE).
- `ECO`: código estándar de la apertura utilizada.
- `Opening`: apertura utilizada.
- `TimeControl`: control de tiempo utilizado.
- `Termination`: motivo de victoria.
- `Lista de jugadas` incluyendo evaluación y tiempo de reloj para cada movimiento.

Aunque estos son los campos que se recogen en la mayoría de los datasets en ajedrez, existen campos adicionales permitidos por el formato `PGN` (Portable Game Notation) utilizado. Información más

detallada sobre cada uno de los campos puede consultarse en http://www.saremba.de/chessgml/standards/pgn/pgn-complete.htm#c9.1.1.

4 Formulación de hipótesis

Partiendo del conjunto de datos elegido, se pretenden contrastar las siguientes hipótesis:

H1) Es posible pronosticar el resultado de un conjunto de partidas de ajedrez en base a la información disponible acerca de los jugadores involucrados.

H2) Es posible definir los factores de éxito que influyen en el devenir de una partida de ajedrez entre dos jugadores.

H3) Es posible definir el estilo de juego de un jugador en base a su histórico de partidas jugadas.

5 Objetivos

En base a este conjunto de datos, se pretenden abordar los siguientes objetivos aplicando las diferentes fases del **proceso KDD** [4]:

1. Caracterizar a los jugadores en base su histórico de partidas estableciendo un perfil de juego para cada uno de ellos (en base a su agresividad, aperturas, etc.).
2. Tratar de pronosticar el resultado de un conjunto de partidas entre unos jugadores predefinidos.

6 Target data

De cara a llevar a cabo el contraste de nuestras hipótesis, se procedió con la definición del conjunto de **datos objetivo** ("tarjeta de datos" o *target data*) compuesto por las siguientes características:

- `USER_ID`: identificador de usuario (`String`).
- `GAME_LINK`: enlace a la partida (`String`).
- `ELO`: puntuación ELO del jugador (`int`).
- `COLOUR`: color de piezas del usuario (`String`).
- `OPENING`: apertura (y variación) utilizada en la partida (`String`).
- `RESULTADO`: modo de finalización de la partida (`int`):
  - 0 = gana_blanco
  - 1 = tablas
  - 2 = gana_negro
- `MOVEMENTS`: número de movimientos de la partida (`int`).
- `TOTAL_TIME`: duración de la partida en segundos (`double`).
- `TOTAL_TIME_PLAYER`: número de segundos invertido por el jugador (`double`).
- `EARLY_TIMES_MEAN`: tiempo medio entre movimientos en el *early game* (`double`).
- `EARLY_TIMES_MEDIAN`: mediana de los tiempos entre movimientos en el *early game* (`double`).
- `EARLY_TIMES_VARIANCE`: varianza de tiempos entre movimientos en el *early game* (`double`).
- `EARLY_TIMES_MAX`: máximo tiempo empleado entre movimientos en el *early game* (`double`).
- `EARLY_TIMES_MIN`: mínimo tiempo empleado entre movimiento en el *early game* (`double`).
- `MID_TIMES_MEAN`: tiempo medio entre movimientos en el *mid game* (`double`).
- `MID_TIMES_MEDIAN`: mediana de los tiempos entre movimientos en el *mid game* (`double`).
- `MID_TIMES_VARIANCE`: varianza de tiempos entre movimientos en el *mid game* (`double`).
- `MID_TIMES_MAX`: máximo tiempo empleado entre movimientos en el *mid game* (`double`).
- `MID_TIMES_MIN`: mínimo tiempo empleado entre movimiento en el *mid game* (`double`).
- `END_TIMES_MEAN`: tiempo medio entre movimientos en el *end game* (`double`).
- `END_TIMES_MEDIAN`: mediana de los tiempos entre movimientos en el *end game* (`double`).
- `END_TIMES_VARIANCE`: varianza de tiempos entre movimientos en el *end game* (`double`).
- `END_TIMES_MAX`: máximo tiempo empleado entre movimientos en el *end game* (`double`).
- `END_TIMES_MIN`: mínimo tiempo empleado entre movimiento en el *end game* (`double`).
- `POINTS_BALANCE`: balance de puntos al final de la partida (`int`).
- `TAKEN_BALANCE`: balance de piezas al final de la partida (`int`).
- `AGGRESSIVENESS`: nivel de agresividad del jugador en el rango [0,5]. Se mide a partir de los siguientes factores (`double`):
  - `EARLY_TAKEN`: número de piezas comidas en el *early game* (primer tercio de la partida).
    - Alto = +1
    - Medio = +0.5
    - Bajo = +0
  - `AGRESSIVE_OPENING`: la apertura utilizada es agresiva para el color de piezas del jugador:
    - True = +2
    - False = +0

- CASTLING: el jugador se enroca a lo largo de la partida:
  - True = +0
  - False = +2

6 Preproceso y normalización

Una vez extraído nuestro conjunto de datos objetivo, se procedió con su tratamiento preliminar:

- El preproceso de los datos consistió, inicialmente, en la **eliminación de partidas con campos nulos**. La existencia de dichos valores nulos se debía en la totalidad de los casos a juegos demasiado cortos, ocasionados por eventos excepcionales como la rendición de alguno de los jugadores al principio de la partida.

- Una segunda medida llevada a cabo fue la **eliminación de partidas que contenían tiempos negativos**: en *lichess*, un jugador puede ofrecer tiempo a su contrincante de forma voluntaria, provocando que algunos de los registros del dataset que almacenan la información acerca de los tiempos de la partida contengan valores negativos. Así, debido a la imposibilidad de conocer *a priori* el tiempo regalado, nos vimos obligados a prescindir de estas partidas, si bien el número de casos de estas características no resultó ser significante.

- Los campos que reflejaban el color (`COLOUR`) de los jugadores y la apertura (`OPENING`) utilizada fueron **transformados en categóricos** y pertinentemente codificados, dando lugar a los campos `OPENING_ENC` y `COLOUR_ENC`.

- Finalmente, se realizó una **normalización de los datos** haciendo uso de la clase `MinMaxScaler` ofrecida por la librería de Python `sklearn`.

7 Visualización y *clustering*

Una vez los datos fueron convenientemente preparados para su utilización, el objetivo inicial perseguido fue la caracterización de los diferentes tipos de jugadores presentes en nuestro dataset mediante agrupamiento o *clustering*. Trataríamos de identificar jugadores similares a partir de las características consideradas.

Primero, con el fin de limitar el número de características empleadas, se realizó una **reducción de la dimensionalidad** de los datos mediante PCA (*Principal Component Analysis*). De nuevo, se empleó la clase `PCA` que ofrece la librería `sklearn` para llevar a cabo este proceso. El número de componentes considerado fue 3, y los resultados obtenidos los que mostrados en las Imágenes 1 y 2.

```
[ Explained variance ratio ]
[0.62322601 0.22088364 0.06225892]
0.9063685719687522
```

Imagen 1. Ratio de varianzas obtenido mediante PCA

```
[ Relation between PCA components and features ]
                      PC-0      PC-1      PC-2
elo                -0.110880 -0.006919  0.288907
opening_enc        -0.000983 -0.999766  0.014340
result             -0.989784  0.001753 -0.033313
movements          -0.000406  0.000115  0.443249
total_time_player   0.001063  0.015091  0.730605
early_times_median -0.001556  0.002331  0.090972
early_times_max    -0.000368  0.004323  0.229377
early_times_min    -0.001495  0.000307  0.014759
mid_times_median   -0.001311  0.011402  0.302232
mid_times_max       0.000450  0.003064  0.076533
mid_times_min      -0.005953  0.004159  0.075692
end_times_median    0.001603  0.002938  0.055031
end_times_max       0.003802 -0.000395  0.132851
end_times_min      -0.001926  0.000521  0.010731
points_balance     -0.089260 -0.000120  0.011578
```

Imagen 2. PCA para la selección de componentes principales

A partir de los resultados obtenidos se consideró el uso de las variables `RESULT`, `OPENING_ENC` y `TOTAL_TIME_PLAYER` para realizar el agrupamiento.

El algoritmo de *clustering* elegido fue *k-means*. De cara a decidir el número de *clusters* óptimo se hizo uso de los coeficientes *Distortion* (Figura 1) y *Silhouette* (Figura 2) para un intervalo de entre 1 y 35 *clusters*, decidiéndonos finalmente por un número de conjuntos `K = 9` (`Distortion`=247.02; `Silhouette`=0.414).

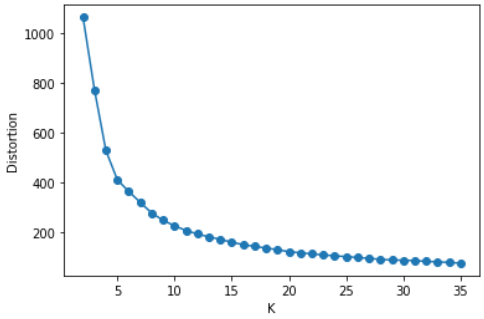

Figura 1. *Distortion* por número de *clusters*

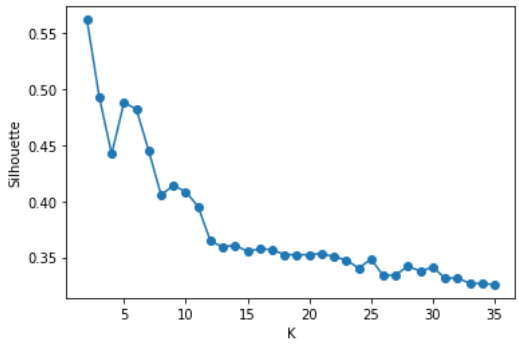

Figura 2. *Silhouette* por número de *clusters*

Los *clusters* obtenidos son los mostrados en las Figuras 3 y 4. Estos pueden interpretarse como las diferentes caracterizaciones (en términos de *apertura utilizada* y *tiempo empleado*) de los conjuntos de jugadores que han *ganado*, *perdido* o *empatado*.

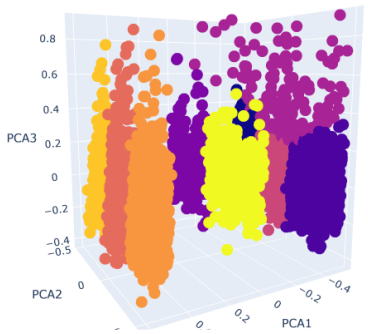

Figura 3. **Visualización de los *clusters* (1)**

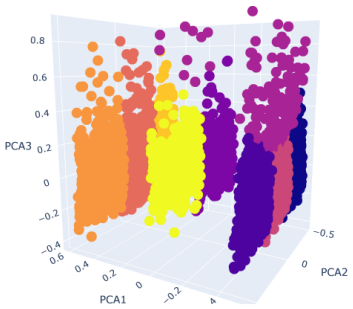

Figura 4. **Visualización de los *clusters* (2)**

8 Modelo de predicción

Una vez llevada a cabo la caracterización de los jugadores presentes en nuestro dataset, se procedió a la construcción del **modelo de predicción** empleado para tratar de conocer el resultado de las partidas en base a sus características.

La herramienta utilizada para este cometido fue BigML [5], una extendida plataforma dedicada a proyectos relacionados tratamiento de datos, modelos y técnicas aprendizaje. El proceso de construcción de nuestro modelo fue el siguiente:

- Primero, se cargaron los datos objetivo (**source**) en la plataforma. Se realizaron configuraciones menores como la selección de idioma del dataset a crear o la elección del punto como separador en números decimales.
- A partir de dichos datos se generó el **dataset**. Sobre dicho dataset se realizaron labores de filtrado para descartar variables que no serían empleadas por nuestro futuro modelo, concretamente: IDs de cada uno de los registros (generados por BigML), IDs de los usuarios y nombre de la apertura (pues ya contamos con su codificación).
- Finalmente, llevamos a cabo la construcción del **modelo**, siendo la variable objetivo establecida el **resultado** de la partida.

Mientras que la Imagen 3 refleja una visualización general del árbol, la Imagen 4 muestra las variables más relevantes identificadas por la herramienta a la hora de predecir el resultado.

Como puede observarse, el balance de puntos (21.82%) y piezas (13.68%), así como el ELO (14.09 % blanco; 12.46% negro) de los jugadores son factores especialmente significativos en el devenir de una partida de ajedrez. A su vez, el número de movimientos de la partida (9.05%) también supone una relevancia considerable.

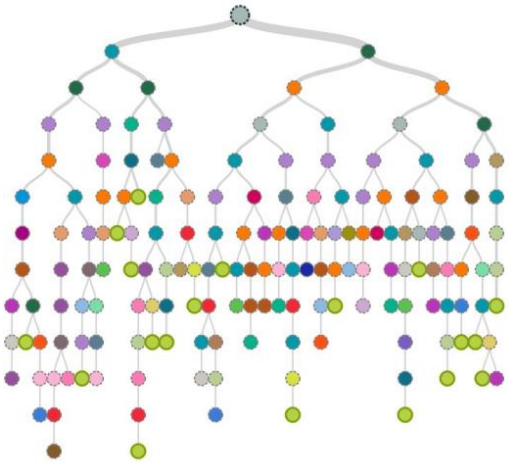

Imagen 3. **Árbol de decisión generado**

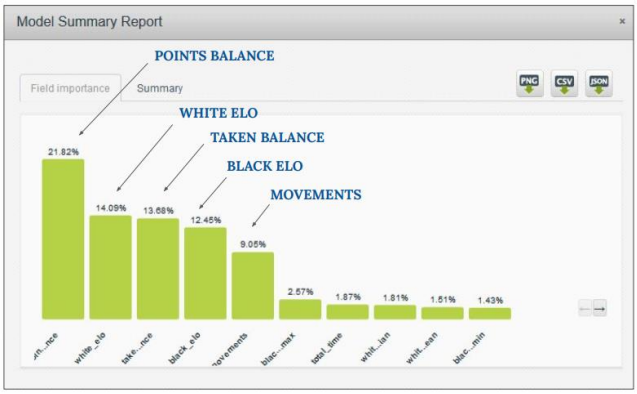

Imagen 4. **Relevancia de los diferentes campos**

9 Evaluación del modelo

Una vez generado nuestro modelo, decidimos evaluarlo con el fin de probar su capacidad de predicción. Para llevar a cabo dicha evaluación, elegimos como conjunto de datos de prueba las partidas correspondientes a los 200 mejores jugadores en modo clásico de *lichess* en el mes de **octubre de 2019**. Tanto las condiciones de extracción de los datos como las características recopiladas fueron las mismas que para los datos de entrenamiento: únicamente se modificaron las fechas de extracción de las partidas desde la API.

Si comparamos las dimensiones del dataset de entrenamiento (*train data*) y prueba (*test data*), en el primero contamos con 4227 registros para generar el modelo, mientras que el segundo contenía 2450 para evaluarlo.

Así, haciendo uso de las herramientas proporcionadas por BigML, se procedió a la evaluación del modelo a partir de estos datos prueba. Los resultados paramétricos y las diferentes matrices de confusión para cada uno de los resultados posibles: ganan blancas, tablas o ganan negras, son los mostrados en las Imágenes 5,6 y 7, respectivamente.

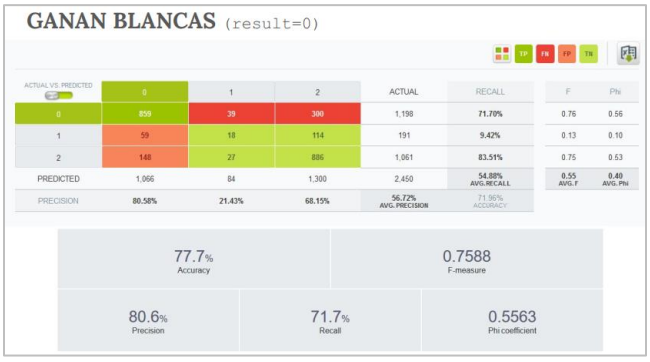

Imagen 5. **Evaluación del modelo para *ganan blancas* (result=0)**

Imagen 6. **Evaluación del modelo para *tablas* (result=1)**

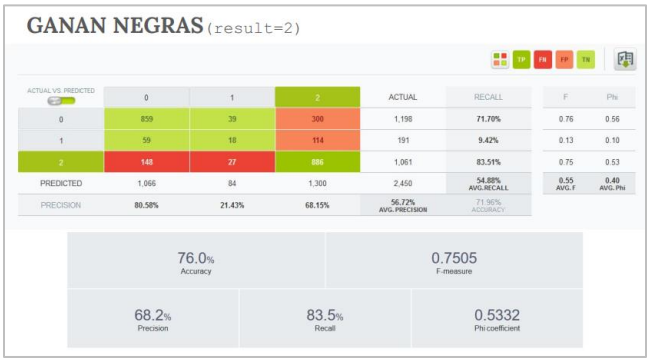

Imagen 7. **Evaluación del modelo para *ganan negras* (result=2)**

El resumen de los resultados obtenidos se detalla en la Tabla 1. Podemos observar buenos resultados en términos de *accuracy* y *precision* a la hora de predecir victorias de blancas o negras. En el caso de las tablas, aunque el valor *accuracy* es alto, la precisión (*precision*) es mucho menor que en el resto de resultados, algo entendible si estudiamos la naturaleza de este fenómeno, mucho menos común y, en ocasiones, producido por motivos ajenos a las características del propio juego.

Tabla 1. **Resultados de la evaluación para cada resultado**

	Accuracy (%)	Precision (%)	Recall (%)	F-Measure	Phi coefficient
Ganan blancas	77.7	80.6	71.7	0.7588	0.5563
Tablas	90.2	21.4	9.4	0.13	0.0958
Ganan negras	76	68.2	83.5	0.7505	0.5332

10 Discusión sobre los resultados

Tras evaluar nuestro modelo, discutiremos cómo han sido abordadas cada una de las hipótesis/objetivos de nuestro trabajo.

- Con respecto a ***H1*** (pronóstico del resultado de un conjunto de partidas en base a los datos recopilados), esta hipótesis ha sido abordada mediante el modelo generado a partir del conjunto de datos objetivo. Dicho modelo ha demostrado una alta capacidad de predicción, especialmente a la hora de predecir victorias de blancas o negras.
- Atendiendo a ***H2*** (identificación de las variables más influyentes en el modelo), también hemos identificado las variables más relevantes en el devenir de una partida; estas son: balance de puntos y piezas, ELO de los jugadores y número de movimientos.
- Finalmente, para abordar ***H3*** (definición del estilo de juego de los jugadores) se han empleado técnicas de *clustering* para el agrupamiento y caracterización de los diferentes perfiles de jugadores presentes en nuestro conjunto de datos.

11 Conclusiones y trabajo futuro

En este proyecto se ha abordado el estudio de partidas de ajedrez tomando como referencia el proceso KDD para la extracción de conocimiento a partir de conjuntos de datos.

Se han llevado a cabo labores de recopilación y preprocesamiento de los datos correspondientes a las partidas de ajedrez, así como a los jugadores. A partir de dichos datos, hemos tratado de caracterizar los perfiles de los jugadores mediante técnicas de *clustering* y hemos generado un modelo para tratar de pronosticar el resultado de partidas de ajedrez en base a sus características, obteniendo unos resultados significativos y un modelo ajustado al propósito.

Dentro del trabajo futuro que podría abordarse sobre esta base, destacaríamos la búsqueda de otros posibles factores de predicción de las partidas con una mayor influencia de conocimiento experto o la posibilidad de realizar dichas predicciones en tiempo real.

REFERENCIAS

[1] Campbell, M., Hoane, A. J., & Hsu, F. (2002). Deep Blue. Artificial Intelligence, 134(1-2), 57–83. doi:10.1016/s0004-3702(01)00129-1

[2] https://database.lichess.org/

[3] https://berserk.readthedocs.io/en/master/

[4] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, 39(11), 27–34. doi:10.1145/240455.240464

[5] https://bigml.com/