
Proyecto 1

Análisis de clasificación de victoria en partidas de ajedrez en línea

PRESENTAN

ING. RICARDO RAMIREZ ISLAS

MARIA GUADALUPE LOMELI PLASCENCIA

ING. ALEJANDRO NOEL HERNÁNDEZ GUTÍERREZ

MATERIA

MODELADO PREDICTIVO

IMPARTE

DR. RIEMANN RUÍZ CRUZ



ITESO

INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES
DE OCCIDENTE

MAESTRÍA EN CIENCIA DE DATOS
OCTUBRE 2023

Índice

Introducción	3
Definición del problema	5
Preparación de los datos	6
Análisis exploratorio	6
Extracción de datos	14
Tratamiento de valores faltantes y limpieza	14
Codificación de variables categóricas	15
Análisis de valores atípicos	16
Análisis de la asimetría de datos	17
Criterio de Correlación	18
Aplicación de los modelos de clasificación	20
Discusión de resultados	24
Apéndice A	27

Introducción

En este proyecto se tiene una base de datos en la que el objetivo es realizar clasificación, se aplican los algoritmos de máquinas de soporte vectorial (SVM) y regresión logística. Las SVM tienen como objetivo encontrar un hiperplano que separe el conjunto de datos de la mejor manera posible, esto siempre y cuando la separación sea fácilmente lineal; en caso de que la separación no sea evidentemente lineal se usa la función kernel para transformar las características de las observaciones y asignarlas a un espacio dimensional diferente que permita hacer ajustes a diferentes situaciones. Existen diferentes tipos de kernel, en este caso se realizaron pruebas con el kernel lineal, polinomial y radial.

Kernel lineal: puede ser utilizado como el producto normal de dos observaciones. El producto entre dos vectores es la suma de la multiplicación de cada par de valores de entrada. Una función kernel lineal es recomendable si la separación lineal de los datos es sencilla.

$$K(x, xi) = \text{sum}(x)(xi)$$

Kernel polinomial: es una forma más generalizada del núcleo lineal. Considera tanto las características dadas por las muestras de entrada para determinar su similitud, como también las combinaciones de éstas. Con “n” características originales y “d” grados de polinomio produce n^d características expandidas.

$$K(x, xi) = 1 + \text{sum}[(x)(xi)]^d$$

Kernel de base radial o RBF: recibe también el nombre de kernel gaussiano. RBF puede mapear un espacio de entrada en un espacio dimensional infinito. Gamma es un parámetro que va de 0 a 1. Un valor más alto de gamma se ajusta perfectamente en el conjunto de datos de entrenamiento, lo que provoca un ajuste excesivo. Gamma igual a 0.1 se considera un buen valor por defecto.

$$K(x, xi) = e^{-\gamma \text{sum}(x-xi)^2}$$

Corresponde a una proyección en un espacio de dimensiones finitas. Se caracteriza porque

$$K(x, xi) = e^0 = 1 \quad \forall x \in X, \text{ luego } \|x\| = 1$$

También sucede que $k(x; z) > 0$ para todo $x, z \in X$ como consecuencia de esto el ángulo entre cualquier par de imágenes es menor que 2, de tal forma que las imágenes de los vectores del espacio de entrada caen dentro de una región restringida del espacio de características.

El segundo de los algoritmos utilizados es la regresión logística, este es similar al modelo de regresión lineal, pero tiene la característica esencial de funcionar para variables dicotómicas, hay que considerar la necesidad de realizar un ajuste en los valores para poder aplicar regresión logística. Se aplica la regularización Lasso como un método de reducción y selección de variables. En la regresión logística se aplica la regularización Lasso.

Lasso corresponde a la abreviación de Least Absolute Shrinkage and Selection Operator, que se traduce como operador de selección y contracción mínima absoluta. Se trata de un método de reducción y selección de variables para la interpretación del modelo de regresión lineal. Es muy aplicada en el aprendizaje automático.

La regularización Lasso tiene como objetivo obtener el subconjunto de predictores que minimice el error de predicción para una variable de respuesta cuantitativa. Hace esto al imponer una variable de restricción en los parámetros del modelo que hace que los coeficientes de regresión de algunas variables se reduzcan a cero.

Definición del problema

En esta sección se va a definir el problema y el conjunto de datos.

La base de datos que se analiza se llama Online Chess Match Prediction que se encuentra en <https://www.kaggle.com/code/dhruvsikka/online-chess-match-prediction/notebook>.

Se tiene información de juegos de ajedrez en línea, con 16 características que mencionan desde el índice para identificar el número de partida, o el tiempo que hay entre un movimiento y otro hasta la secuencia de movimientos realizados; esto además de la columna que indica quién ganó el juego o si hubo un empate. Así el objetivo que se tiene al realizar el procesamiento de los datos es estimar, dadas las características, quién será el ganador en una partida de ajedrez en línea.

Preparación de los datos

Análisis exploratorio

La base de datos cuenta en total con 17 columnas y 20058 observaciones, se tienen variables de tipo entero, objeto y booleanos, como se puede verificar en el archivo *exploratorio.py*.

Las características de las variables se describen más a detalle en la siguiente tabla:

Nombre	Tipo	Valores faltantes	Valores únicos	Valor mínimo	Valor máximo
game_id	Entero	no	20058	1	20058
rated	Booleano	no	2	0	1
turns	Entero	no	211	1	349
victory_status	Objeto	no	4	Draw	Resign
winner	Objeto	no	3	Black	White
time_increment	Objeto	no	400	0+12	90+8
white_id	Objeto	no	9438	--jim--	zzzimon
white_rating	Entero	no	1516	784	2700
black_id	Objeto	no	9331	-0olo0-	zztopillo
black_rating	Entero	no	1521	789	2723
moves	Objeto	no	18920		
opening_code	Objeto	no	365	A00	E98
opening_moves	Entero	no	23	1	28
opening_fullname	Objeto	no	1477	Alekhine Defense	Zukertort Opening: Wade Defense
opening_shortcode	Objeto	no	128	Alekhine Defense	Zukertort Opening
opening_response	Objeto	18851	3	No disponible	No disponible
opening_variation	Objeto	5660	615	No disponible	No disponible

A continuación, se presenta la descripción de las variables:

1. game_id: Identificador único para cada partida de ajedrez.
2. rated: Indica si la partida está clasificada (True) o no clasificada (False). En los juegos de ajedrez en línea, las partidas clasificadas afectan el rating de los jugadores.
3. turns: Es el número total de movimientos realizados en la partida, en este caso va desde 1 hasta 349.
4. victory_status: Indica el estado de la victoria una vez concluida la partida. Puede tomar diferentes valores, en el caso de esta base de datos dichos valores son "mate" (jaque mate), "resign" (rendición), "outoftime" (agotamiento de tiempo), "draw" (empate).

5. winner: Indica el jugador que ganó la partida. En la base de datos están presentes los valores "white" (blanco), "black" (negro) o "draw" (empate).
6. time_increment: Incremento de tiempo en segundos después de cada movimiento.
7. white_id: Identificación única del jugador blanco.
8. white_rating: Rating del jugador blanco en el momento de la partida.
9. black_id: Identificación única del jugador negro.
10. black_rating: Rating del jugador negro en el momento de la partida.
11. moves: Lista de movimientos realizados en la partida.
12. opening_code: Código que identifica la apertura de ajedrez jugada.
13. opening_moves: Número de movimientos iniciales realizados en la apertura.
14. opening_fullname: Nombre completo de la apertura.
15. opening_shortname: Nombre abreviado de la apertura.
16. opening_response: Respuesta a la apertura por parte del oponente.
17. opening_variation: Variación específica de la apertura.

En cuanto a los valores faltantes, destacan las variables que se encuentran en las dos últimas columnas, opening_response y opening_variation en las cuales faltan 18851 y 5660 respectivamente.

En un primer momento, en cuanto a las columnas a considerar para el análisis, se tomarán en cuenta las que sean más relevantes para clasificar las observaciones como corresponde a la variable winner, es decir, winner es la variable de salida, considerando a las demás como las variables que van a influir en el resultado correspondiente a la variable dependiente.

Las columnas que son candidatas para ser eliminadas en un primer momento y antes de realizar algún análisis son: las dos últimas "opening_response" y "opening_variation" dada la cantidad de datos faltantes; la columna de "moves", la cual contiene la secuencia de movimientos realizados en la partida de ajedrez y tiene extensión muy variable además de que representa dificultad para ser codificada; las columnas "white_id" y "black_id" ya que para ganar no resulta relevante el número de identificación.

En ajedrez, la "apertura" hace referencia a la fase inicial del juego en la cual los jugadores mueven sus piezas para desarrollar sus posiciones y prepararse para la parte intermedia y el final del juego. Comienza desde el primer movimiento y generalmente termina después de aproximadamente 10 a 20 movimientos realizados por cada jugador.

La elección de la apertura es una decisión estratégica muy importante en el ajedrez, esto se debe a que puede influir en el desarrollo futuro del juego. Cada movimiento de apertura tiene un nombre específico y se caracteriza por una secuencia particular de movimientos de apertura.

En el contexto de la base de datos "Online chess match prediction", las variables "opening_fullname", "opening_shortname" y "opening_variation" se refieren a diferentes aspectos del movimiento de apertura utilizada en una partida de ajedrez. Aquí las diferencias entre ellas:

1. opening_fullname (Nombre completo de la apertura):

Proporciona el nombre completo de la apertura utilizada en la partida, al inicio. Por ejemplo, si se jugó la "Apertura Española", esta variable contendría el nombre completo de esa apertura.

2. opening_shortcode (Nombre abreviado de la apertura):

Proporciona una forma abreviada del nombre de la apertura. Es una versión concisa del nombre completo. Por ejemplo, para la "Apertura Española", el nombre abreviado podría ser "Ruy López".

3. opening_variation (Variación específica de la apertura):

Indica la variación específica de la apertura jugada en la partida. La apertura puede tener varias variantes, y esta variable especifica cuál de esas variantes se utilizó en la partida. Por ejemplo, dentro de la "Apertura Española", hay diferentes variantes como la "Defensa Morphy" o la "Defensa Steinitz".

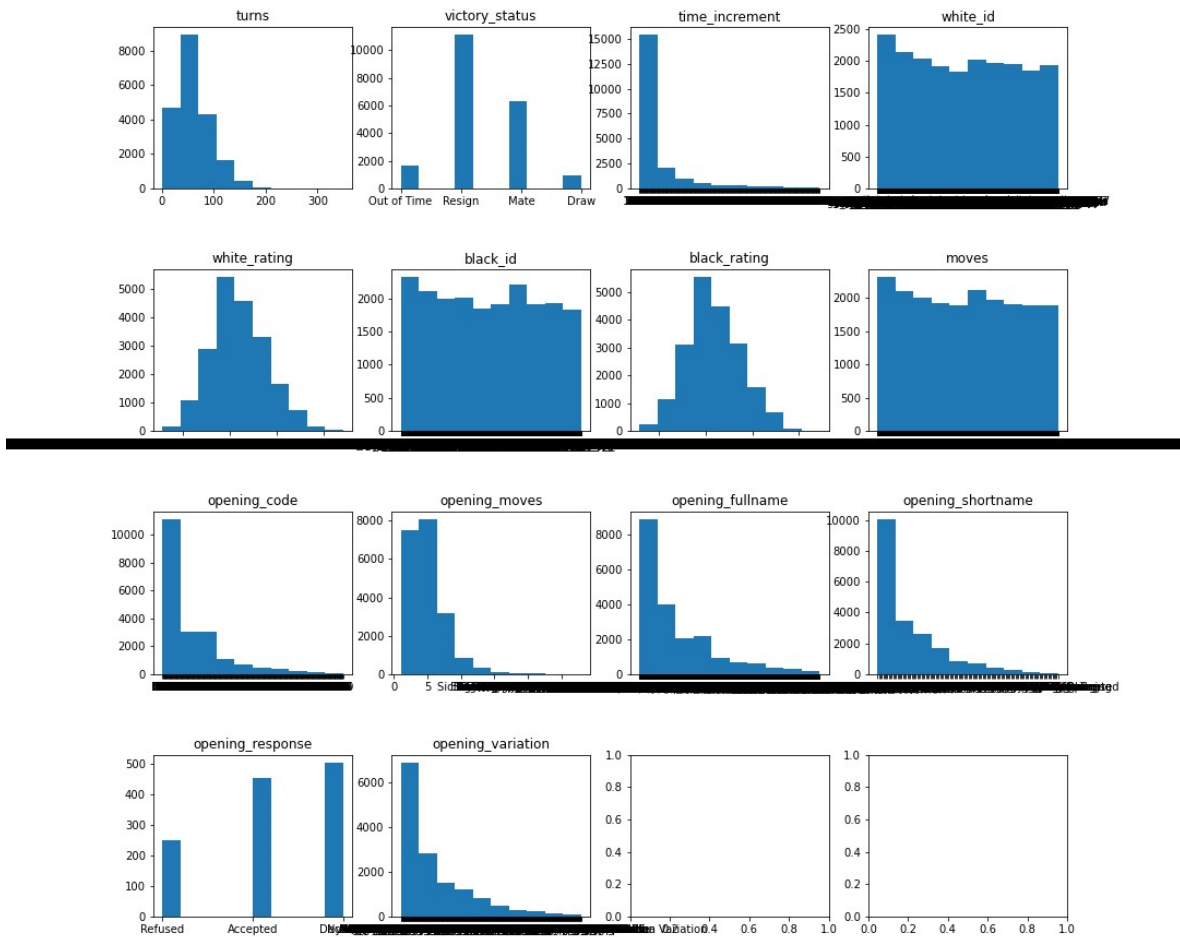
En otras palabras, mientras que "opening_fullname" proporciona el nombre completo de la apertura, "opening_shortcode" muestra una versión abreviada del nombre, y "opening_variation" se centra en la variante específica de la apertura jugada en la partida de ajedrez. Las variables mencionadas permiten identificar y categorizar las aperturas utilizadas en las partidas de ajedrez en el conjunto de datos.

Dado lo anterior, se anticipa que será necesario valorar si en realidad se pueden eliminar las dos columnas finales por la simple razón de que faltan datos o será necesario realizar algún tratamiento a éstas e incluirlas si es que se encuentra que tienen relevancia para determinar el triunfo en el juego.

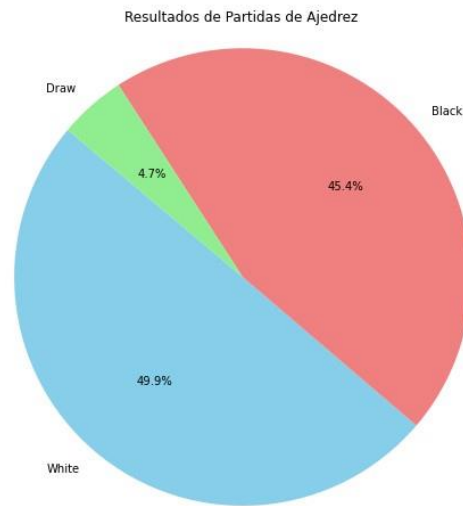
Gráficos de la base de datos.

La siguiente gráfica muestra el histograma de cada una de las variables, en estos histogramas se omitió la variable que corresponde al índice de cada juego (game_id), también la que indica que la partida está clasificada o no (rated) que es una variable booleana, así como la de salida (winner). Se puede apreciar que algunas de estas variables pueden adquirir una gran cantidad de valores de tal forma que no es posible visualizar las etiquetas en el eje horizontal de la gráfica, aunque se hagan ajustes en el área de graficación. Algunas por simple observación, se puede afirmar que muestran sesgo en su distribución como time_increment aunque aún no se realizan análisis. En la gráfica de opening_response se aprecia que los valores en el eje vertical son mucho menores que en el resto de las gráficas, indica datos faltantes como se verá más adelante.

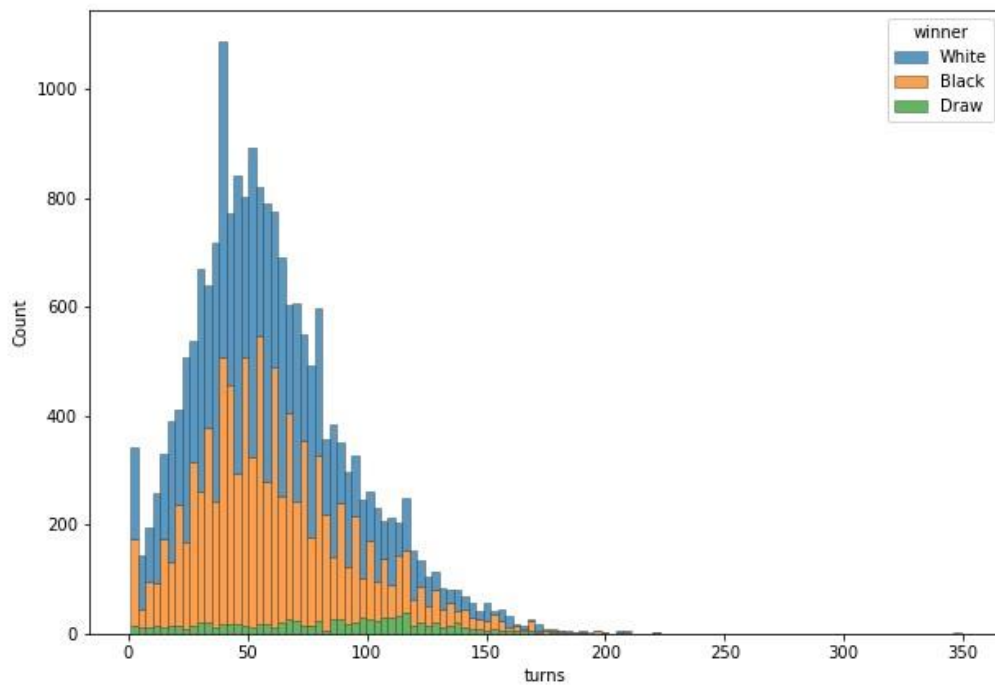
Distribución de las variables



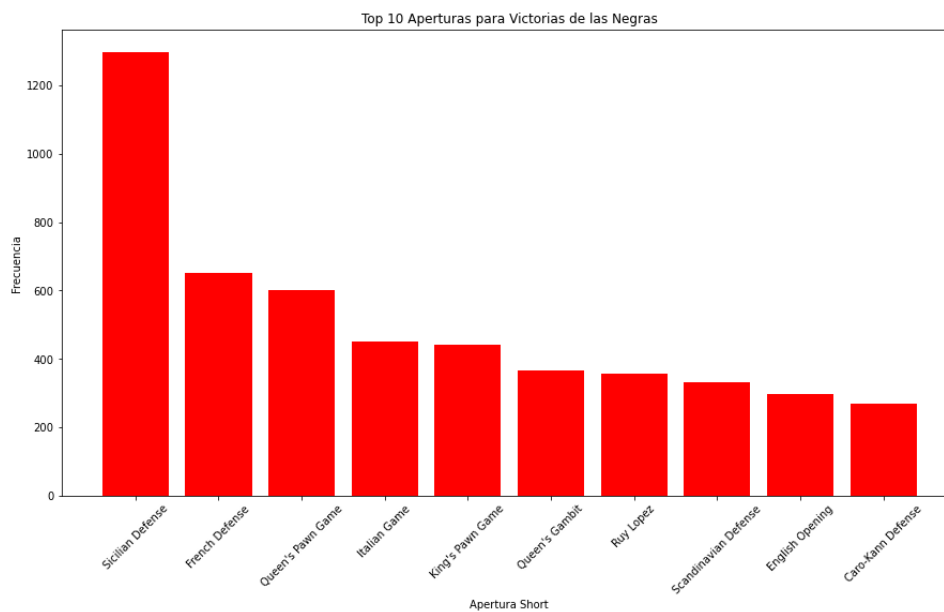
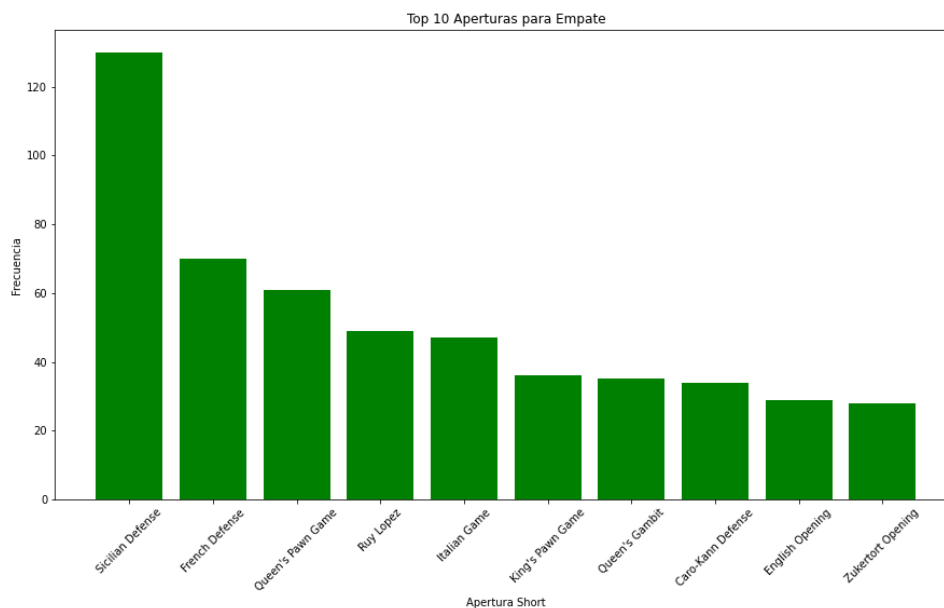
En la siguiente gráfica, se muestra el porcentaje de partidas ganadas por las fichas blancas, las de color negro y los empates, se puede apreciar que este último evento se presenta con mucho menor frecuencia que los otros dos. Al jugar una partida de ajedrez online menos del 5% serán empate.

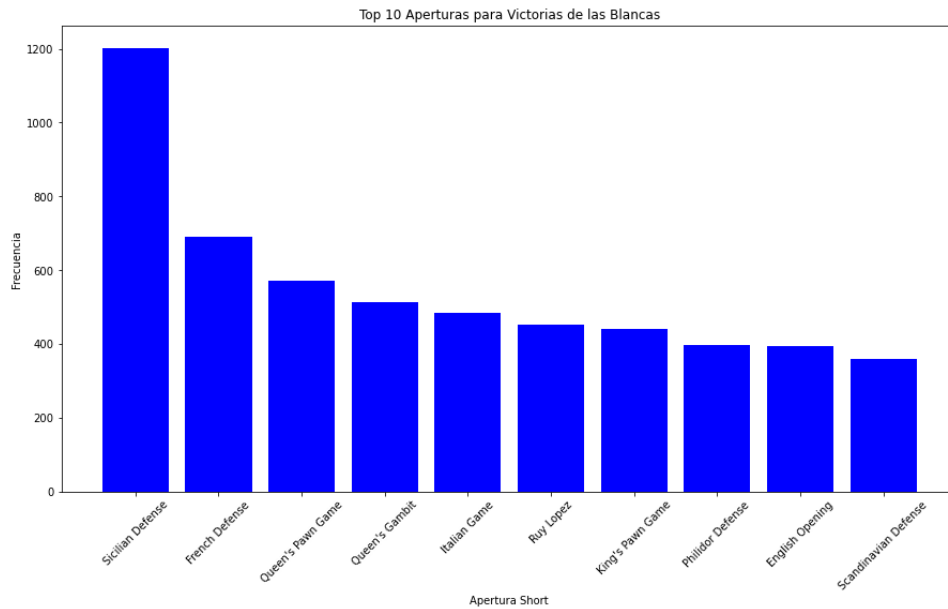


En la gráfica que se muestra a continuación se permite apreciar que la mayoría de las partidas de ajedrez tienen entre 25 y 75 movimientos, también se puede apreciar que en el caso de los empates se mantiene más estable la variación en el número de movimientos realizados.

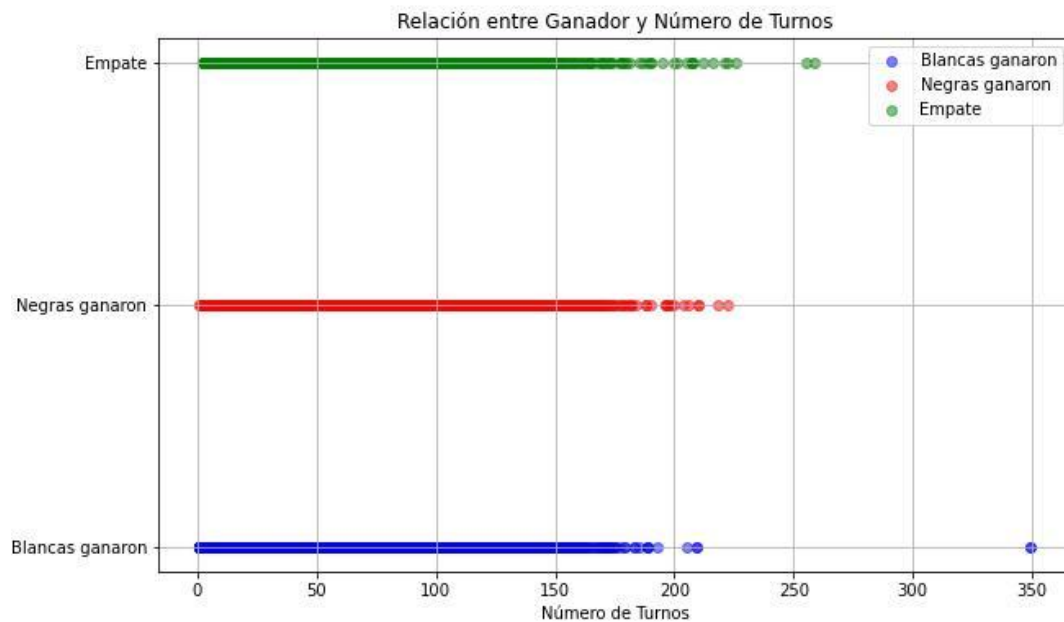


En las tres siguientes gráficas se muestran los movimientos de apertura para los casos de empate, que ganen las fichas negras y que ganen las fichas blancas. Se puede apreciar que hay tres aperturas que son muy populares, puesto que para cualquiera de las opciones de desenlace del juego están presentes, estas son Sicilian Defense, French Defense y Queen's Pawn Game.





En cuanto a ganador y número de turnos no se muestran diferencias significativas a simple vista en la gráfica, hay que mencionar que en empate hay dos observaciones que se alejan un poco de la distribución. Otra observación que se aleja es en el caso que ganan las fichas de color blanco, se tiene una sola observación que se muestra muy alejada del resto, lo que se puede interpretar como una partida en la que hubo quizá algún cambio de estrategia o movimientos inesperados por parte de uno de los oponentes que hizo que el juego se prolongara y finalizara con la victoria para las fichas de color blanco.



En el gráfico que relaciona las victorias con los estados de la victoria, se puede apreciar que las menos son las victorias que se obtienen porque uno de los participantes se excedió en el tiempo y eso le dio el triunfo al oponente, lo cual está marcado en color verde en la gráfica. En un tono de color naranja se tiene las victorias contundentes que son las de jaque mate. La mayoría de los ganadores, obtienen este resultado porque el contrincante así lo reconoce como se muestra en la opción Resign que se muestra en color amarillo. También podemos apreciar que muchas menos partidas, con respecto a las demás, en donde se finaliza con empate.



Extracción de datos

Los siguientes pasos del preprocesamiento de datos se realizaron en el archivo de Python: “Practica preprosesamiento.py”

Para el procesamiento de datos, se decidió tomar las muestras totales con escalamiento de datos en algunas corridas, mientras que en otras se decidió no realizar escalamiento, pero si hacer un re-muestreo pasando de 20,058 muestras a 2,000 muestras seleccionadas aleatoriamente.

Para entender más acerca cada uno de los archivos utilizados (original y transformados) favor de revisar el Apéndice A en este documento.

Tratamiento de valores faltantes y limpieza

- La variable “opening_variation” contaba con 28% de datos faltantes. En base a análisis de la misma se concluyó que aquellas muestras en blanco eran porque no tenían “opening variation” (variación inicial). Esto se pudo observar con ayuda de la variable “opening_fullname”. Se decidió completar aquellas muestras en blanco con “traditional opening” (inicio tradicional)
- Las variables “white_id” y “black_id” fueron removidas del dataset. Estas columnas son nombres de usuarios que no influyen en nuestra variable de salida “winner”
- La variable “moves” fue removida del dataset. Son movimientos concatenados que no pueden procesarse o no generan valor para nuestra variable de respuesta, por la representación de la información.
- La variable “opening_response” está con el 93% de datos faltantes. Se decidió no incluirla en el análisis.

Codificación de variables categóricas

Se decidió usar `LabelEncoder()` de la librería `sklearn` para transformar las variables categóricas y codificarlas para poderlas procesar. Las variables transformadas fueron:

- `victory_status`
- `winner`
- `time_increment`
- `opening_code`
- `opening_fullname`
- `opening_shortcode`
- `opening_variation`
- `rated`

De esta manera se obtuvo un primer dataset con las variables sin datos faltantes y con transformaciones de variables categóricas codificadas.

RangeIndex: 2000 entries, 0 to 1999

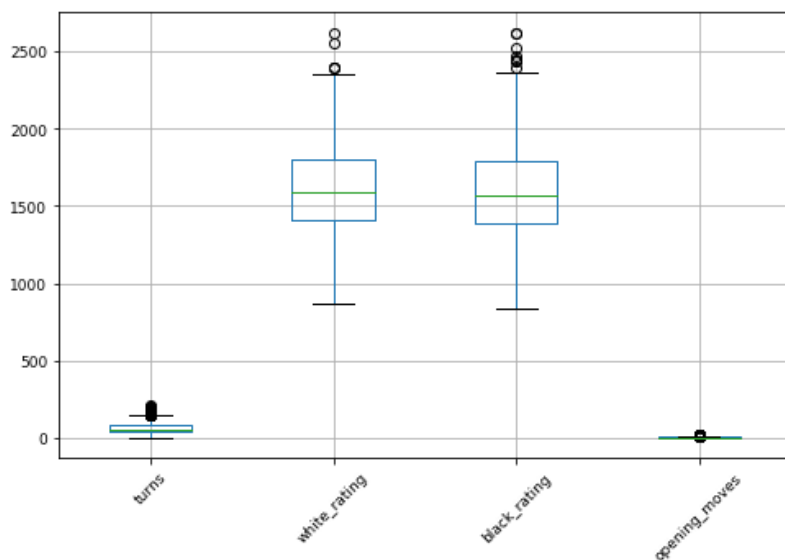
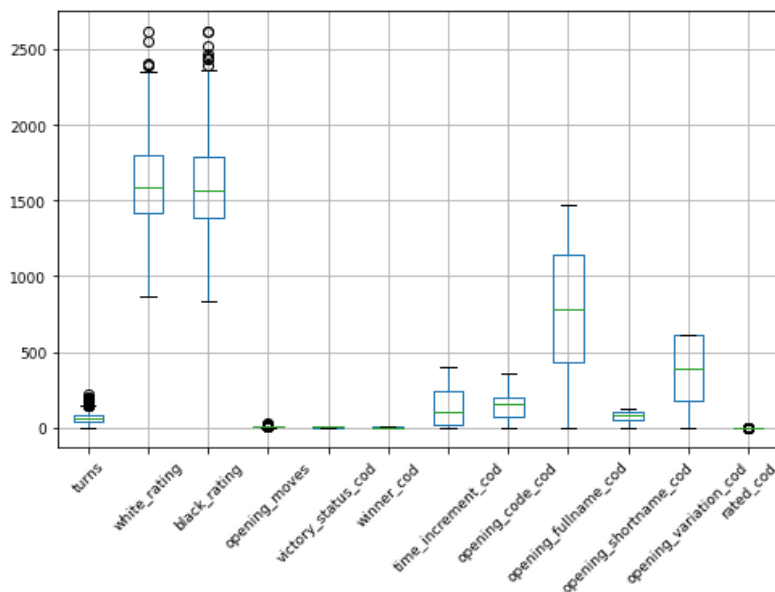
Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	game_id	2000 non-null	int64
1	turns	2000 non-null	int64
2	white_rating	2000 non-null	int64
3	black_rating	2000 non-null	int64
4	opening_moves	2000 non-null	int64
5	victory_status_cod	2000 non-null	int32
6	winner_cod	2000 non-null	int32
7	time_increment_cod	2000 non-null	int32
8	opening_code_cod	2000 non-null	int32
9	opening_fullname_cod	2000 non-null	int32
10	opening_shortcode_cod	2000 non-null	int32
11	opening_variation_cod	2000 non-null	int32
12	rated_cod	2000 non-null	int64

dtypes: int32(7), int64(6)

Análisis de valores atípicos

Se analizaron las variables del dataset y se observan outliers en las variables numéricas “turns”, “white_raiting”, “black_raiting”, “opening_moves”



Se procedió a la remoción de los outliers en las 4 variables cuantitativas mostradas en el box plot de arriba. En total se removieron 130 muestras que representan solo el 6% de los 2000 datos. Esto nos da como resultado un dataset con 1,870 muestras sin outliers.

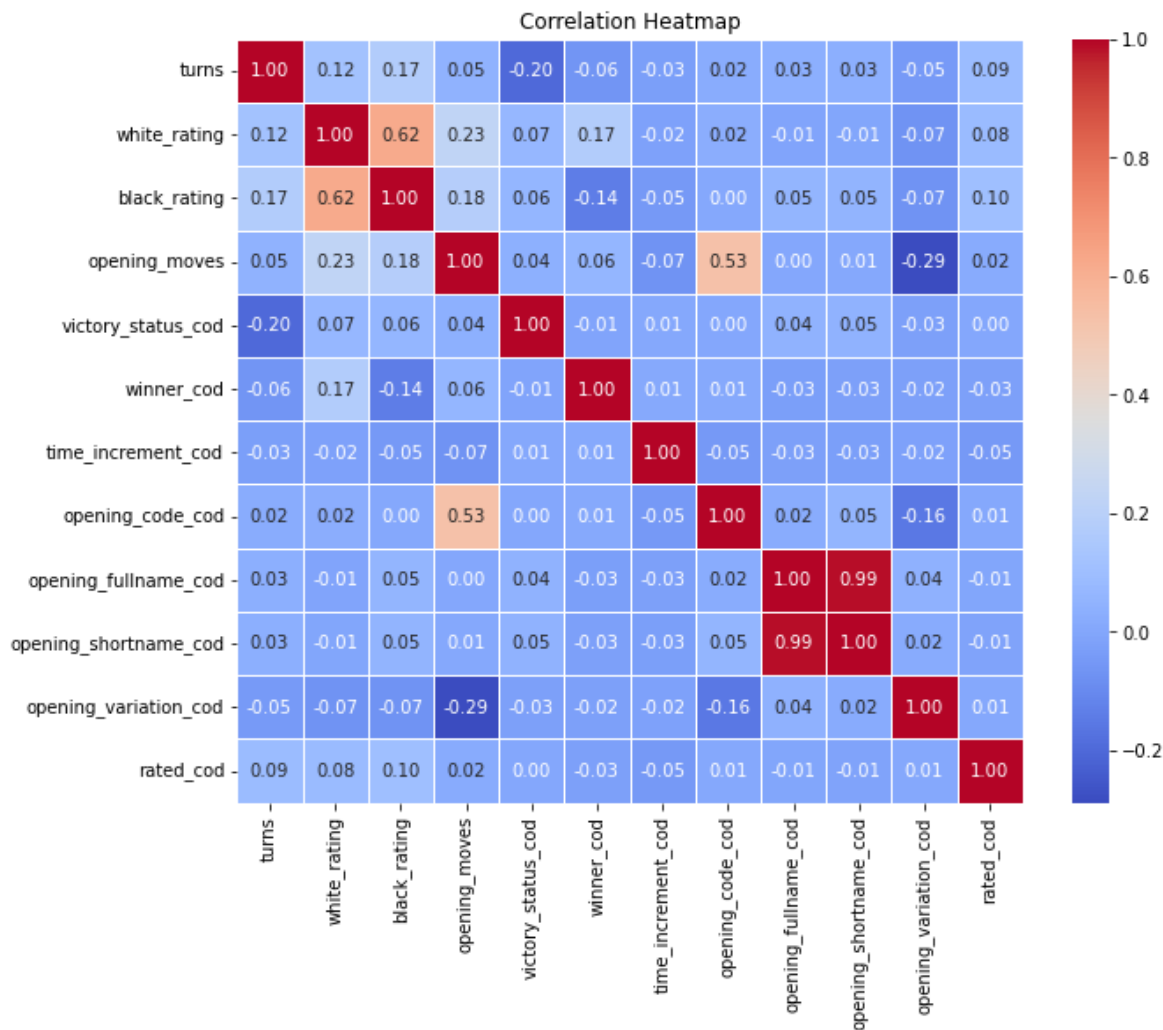
Análisis de la asimetría de datos

Se realizó análisis de asimetría usando la librería pandas sobre los datos sin outliers. Los resultados para cada variable concluyen que todas las variables están dentro del rango de -1.5 a 1.5 por lo que ninguna de ellas requiere transformación para mitigar asimetría.

Index	0
turns	0.51437
white_rating	0.184358
black_rating	0.178775
opening_moves	0.571839
victory_status_cod	-0.523309
winner_cod	-0.121072
time_increment_cod	0.795591
opening_code_cod	0.0496014
opening_fullname_cod	-0.105364
opening_shortcode	-0.389626
opening_variation_cod	-0.376158
rated_cod	-1.49703

Criterio de Correlación

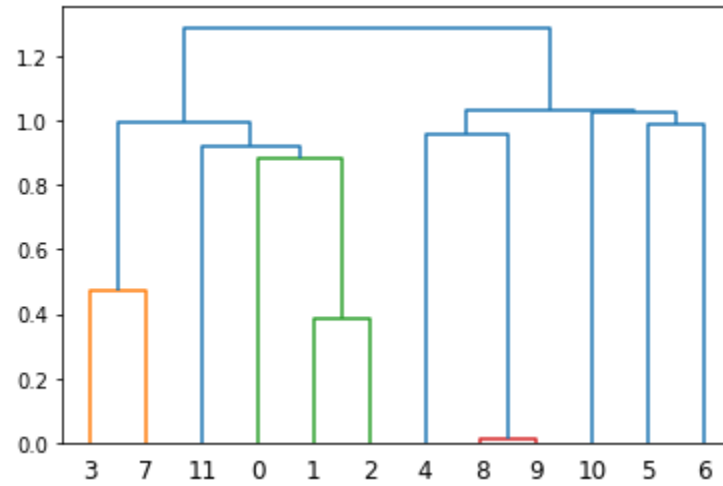
Se realizó el análisis de correlación para identificar aquellas variables altamente correlacionadas y así evitar problemas de estabilidad durante la corrida de algoritmos. Primeramente, se generó un Heat-map de Correlación el cual se puede apreciar a continuación:



Se puede deducir del Heat Map de Correlación lo siguiente:

- las variables “opening_fullname_cod” y “opening_shortcode” muestran alta correlación. Se decide dejar la variable “opening_shortcode” para procesamiento.
- las variables “white_raiting” y “black_raiting” muestran alta correlación. Se decide únicamente tomar “white_raiting” para procesamiento.
- las variables “opening_code_cod” y “opening_moves” también muestran fuerte correlación, por lo que se deja únicamente la variable “opening_moves” para procesamiento.

Lo anterior se puede reforzar al analizar el algoritmo de clustering en donde se puede apreciar los clústeres de las variables relacionadas:



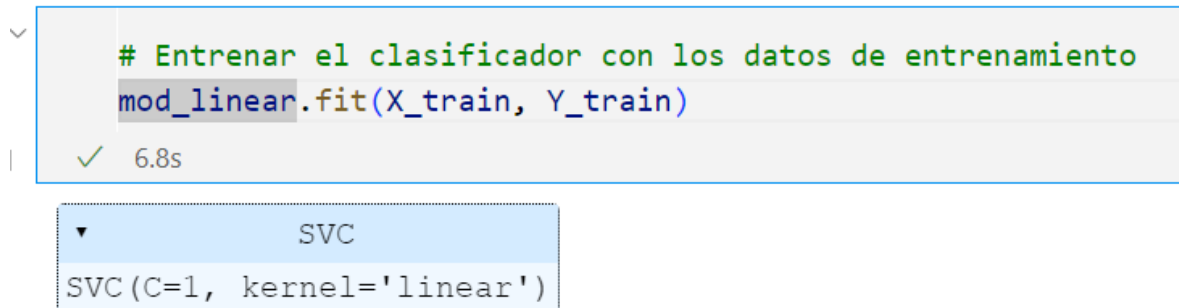
La visualización arriba refuerza las conclusiones obtenidas anteriormente sobre la alta correlación en las variables 8 y 9 (“opening_fullname_cod” y “opening_shortcode”), seguido de 1 y 2 (“white_rating” y “black_rating”) y finalmente 3 y 7 (“opening_code_cod” y “opening_moves”).

Aplicación de los modelos de clasificación

En el archivo ***model_execution originales.ipynb*** se procesan los datos originales ya codificados que fueron limpiados únicamente con la eliminación de *user_id*, *White_id*, *Black_id* y *Opening_response*. Estas variables se quitaron porque no aportaban nada o porque estaban casi vacías (Limpieza básica).

El objetivo de esta ejecución es probar los modelos sin muchos cambios en los datos, ningún procesamiento de outliers o corrección de distribución. Los resultados se incluyen en la sección Discusión de resultados.

Un problema que se presentó fue que la máquina de soporte lineal con las 20058 tardó más de 8 hrs. Al escalar los datos logramos un entrenamiento en 6.8 segundos.



```
# Entrenar el clasificador con los datos de entrenamiento
mod_linear.fit(X_train, Y_train)
```

✓ 6.8s

▼ SVC

SVC(C=1, kernel='linear')

Se ejecutaron los siguientes modelos:

- Regresión logística (0.3 s)
- SVM Lineal (6.8 s)
- SVM Polinomial (6.6 s)
- SVM RBF (6.5 s)
- Xgboost (.9s) (Aunque no se vio en clase se usa para comparar y con el conocimiento previo de que los modelos de árboles son buenos para clasificar)

En el archivo ***model_execution_df1.ipynb*** se utiliza el conjunto de datos *df1_2000_cleaned.csv* detallado en el Apéndice A y se entrenan y evalúan los siguientes modelos.

- Regresión logística
- SVM Lineal
- SVM Polinomial
- SVM RBF
- Xgboost

Las características de este dataframe son las siguientes:

Subsampling 2000 muestras + eliminación de outliers

- Datos originales.
- Randomly Resample to 2000
- Variable "opening_variation" completada
- Las variables categoricas han sido codificadas para procesamiento de datos.
- Sin datos faltantes
- Columnas removidas:
 - white_id
 - black_id
 - moves
 - opening_response

En el archivo ***model_execution_df2.ipynb*** se utiliza el conjunto de datos df2_2000_cleaned.csv detallado en el Apéndice A y se entrenan y evalúan los siguientes modelos.

- Regresión logística
- SVM Lineal
- SVM Polinomial
- SVM RBF
- Xgboost

Las características de este dataframe son las siguientes:

- Datos originales.
- Randomly Resample to 2000
- Variable "opening_variation" completada
- Las variables categoricas han sido codificadas para procesamiento de datos.
- Sin datos faltantes
- Columnas removidas:
 - white_id
 - black_id
 - moves
 - opening_response
 - opening_fullname_cod

En el archivo ***model_execution_df3.ipynb*** se utiliza el conjunto de datos df3_2000_cleaned.csv detallado en el Apéndice A y se entrenan y evalúan los siguientes modelos.

- Regresión logística

- SVM Lineal
- SVM Polinomial
- SVM RBF
- Xgboost

Las características de este dataframe son las siguientes:

- Datos originales.
- Randomly Resample to 2000
- Variable "opening_variation" completada
- Las variables categoricas han sido codificadas para procesamiento de datos.
- Sin datos faltantes
- Columnas removidas:
 - white_id
 - black_id
 - moves
 - opening_response
 - opening_shortcode
 - opening_variation_cod

En el archivo ***model_execution_no_corr.ipynb*** se utiliza el conjunto de datos df_2000_cleanned_after_correlation_.csv detallado en el Apéndice A y se entrenan y evalúan los siguientes modelos.

- Regresión logística
- SVM Lineal
- SVM Polinomial
- SVM RBF
- Xgboost

Las características de este dataframe son las siguientes:

Igual a df1 pero removiendo columnas correlacionadas:

- opening_fullname_cod se removió
- black_raiting se removió
- opening_code_cod se removió

Discusión de resultados

A continuación, se presenta la tabla con los diferentes files utilizados (escenarios), modelos y métricos obtenidos:

CSV (escenario)	Algoritmo	1. Accuracy 2. Precisión 3. Recall	1. Accuracy 2. Precisión 3. Recall
		Prueba	Entrenamiento
df1_original_cod.csv	Regresión logística	0.681	0.663
		0.682	0.664
		0.681	0.663
	SVM Lineal	0.672	0.666
		0.675	0.668
		0.672	0.666
	SVM Polinomial	0.568	0.569
		0.589	0.595
		0.568	0.569
	SVM RBF	0.665	0.696
		0.666	0.698
		0.665	0.696
	Xgboost	0.884	0.954
		0.884	0.954
		0.884	0.954
df_2000_cleaned_after_correlation_csv	SVM Lineal	0.579	0.615
		0.578	0.615
		0.579	0.615
	SVM Polinomial	0.499	0.555
		0.476	0.534
		0.499	0.555
	SVM RBF	0.520	1.000
		0.323	1.000
		0.520	1.000
	Regresión Logística	0.586	0.614
		0.583	0.614
		0.586	0.614
df1_2000_cleaned.csv	SVM Lineal	0.643	0.670
		0.644	0.669
		0.643	0.670
	SVM Polinomial	0.613	0.617
		0.593	0.591
		0.613	0.617
	SVM RBF	0.508	1.000
		0.258	1.000
		0.508	1.000
	Regresión Logística	0.652	0.668
		0.654	0.667
		0.652	0.668

df2_2000_cleaned.csv	SVM Lineal	0.658	0.668
		0.657	0.668
		0.658	0.668
	SVM Polinomial	0.608	0.622
		0.586	0.594
		0.608	0.622
	SVM RBF	0.513	1.000
		0.751	1.000
		0.513	1.000
	Regresión Logística	0.672	0.674
		0.672	0.674
		0.672	0.674
df3_2000_cleaned.csv	SVM Lineal	0.661	0.660
		0.660	0.660
		0.661	0.660
	SVM Polinomial	0.640	0.604
		0.613	0.578
		0.640	0.604
	SVM RBF	0.522	1.000
		0.708	1.000
		0.522	1.000
	Regresión Logística	0.670	0.662
		0.669	0.662
		0.670	0.662

Entre las cosas que más destacan en esta tabla, es que con el escenario “df1_original_cod.csv”, donde se utilizó la cantidad total de muestras y variables escaladas, se obtuvo mejores métricas de prueba y entrenamiento para la mayoría de los modelos. Dentro de los modelos lineales, se puede apreciar que con la Regresión Logística se obtuvo un test accuracy de 0.68, seguido de SVM Lineal con un test accuracy de 0.67. Como opción adicional se decidió correr el modelo “Xgboost” tras comprobar que los modelos lineales tradicionales no cumplieron con un accuracy alto, los resultados muestran un test accuracy de 0.88. Estos resultados obtenidos fueron los mejores métricos obtenidos.

Entre los modelos que tienen características que destacan se encuentran los SVM con la implementación del kernel RBF, ya que muestran resultados de Precisión, Accuracy y Recall de 1 en todos los casos con 2000 muestras lo cual evidentemente muestra que existe overfittig, entre las causas para esto podría ser el valor de gamma utilizado, que fue de $1/n$, donde n indica el número de columnas; otra de las causas posibles es el escalamiento de los datos; también puede ser que alguna de las variables consideradas sea irrelevante o se esté ocasionando ruido en los datos al momento de hacer los subconjuntos de la base de datos. Todas estas posibilidades serían temas para un análisis posterior.

Respecto al submuestreo, algo que destaca es que el mejor modelo obtenido con 2000 muestras se logra en el df2_2000_cleaned, en el cual se removieron en total 5 características. Dos de las cuales opening_fullname_cod quedó representado por 2 variables de menos clases: opening_shortcode y opening_variation_cod. Con esto podemos plantear la idea de que con menos clases funciona mejor la máquina de soporte vectorial.

Respecto a las métricas utilizadas para medir la efectividad del modelo, llama la atención que en el `df_2_2000_cleaned` y en el `df_2_2000_cleaned`, precisión es significativamente más alta que la accuracy y el recall. Esto puede indicar que la configuración en esos modelos es mejor para predecir los verdaderos positivos que fueron realmente correctos, o cuando se predice bien quién gana y resulta que ganó.

Apéndice A

En esta sección se encuentra el detalle de cada uno de los datasets usados como escenarios base de las corridas de los modelos usados en este proyecto:

#	Nombre del archivo	Descripción	Variables
1	df1_2000	<ul style="list-style-type: none"> Datos originales. Randomly Resample to 2000 Variable "opening_variation" completada Las variables categoricas han sido codificadas para procesamiento de datos. Sin datos faltantes Columnas removidas: <ul style="list-style-type: none"> white_id black_id moves opening_response 	0 game_id 1 turns 2 white_rating 3 black_rating 4 opening_moves 5 victory_status_cod 6 winner_cod 7 time_increment_cod 8 opening_code_cod 9 opening_fullname_cod 10 opening_shortcode_cod 11 opening_variation_cod 12 rated_cod
2	df2_2000	<ul style="list-style-type: none"> Datos originales. Randomly Resample to 2000 Variable "opening_variation" completada Las variables categoricas han sido codificadas para procesamiento de datos. Sin datos faltantes Columnas removidas: <ul style="list-style-type: none"> white_id black_id moves opening_response opening_fullname_cod 	0 game_id 1 turns 2 white_rating 3 black_rating 4 opening_moves 5 victory_status_cod 6 winner_cod 7 time_increment_cod 8 opening_code_cod 9 opening_shortcode_cod 10 opening_variation_cod 11 rated_cod
3	df3_2000	<ul style="list-style-type: none"> Datos originales. Randomly Resample to 2000 Variable "opening_variation" completada Las variables categoricas han sido codificadas para procesamiento de datos. Sin datos faltantes Columnas removidas: <ul style="list-style-type: none"> white_id black_id moves opening_response 	0 game_id 1 turns 2 white_rating 3 black_rating 4 opening_moves 5 victory_status_cod 6 winner_cod 7 time_increment_cod 8 opening_code_cod 9 opening_fullname_cod 10 rated_cod

		<ul style="list-style-type: none"> • opening_shortcode • opening_variation_cod 	
4	df1_2000_cleaned	Igual a df1_2000 pero sin outliers	
5	df2_2000_cleaned	Igual a df2_2000 pero sin outliers	
6	df3_2000_cleaned	Igual a df3_2000 pero sin outliers	
7	df_2000_cleaned_after_correlation	Igual a df1_2000_cleaned pero removiendo columnas correlacionadas: <ul style="list-style-type: none"> • opening_fullname_cod se removió • black_raiting se removió • opening_code_cod se removió 	