

# Predicción de resultados de finales de ajedrez KR-vs-KP mediante aprendizaje supervisado\*

Marlon Giraldo-Ramírez

Juan Felipe Santa

<sup>1)</sup>Departamento de Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Colombia.

Este artículo aborda la predicción del desenlace en el final de ajedrez *King–Rook vs King–Pawn* (KR-vs-KP) como tarea de clasificación binaria supervisada sobre el conjunto kr-vs-kp (OpenML ID 3). El objetivo es estimar, a partir de descriptores simbólicos completamente categóricos, si las blancas pueden forzar la victoria (*won*) o no (*nowin*). Se comparan varias familias de modelos supervisados, incluyendo Regresión Logística, Árboles de Decisión (CART), Random Forest, Redes Neuronales Multicapa (MLP) y SVM con kernel RBF. Se pone especial énfasis en el contraste entre CART, por su interpretabilidad y extracción de reglas, y SVM, por su capacidad para modelar fronteras no lineales en alta dimensionalidad tras *one-hot encoding*. La evaluación se implementa mediante *pipelines* con validación estratificada (*k-fold*) y un conjunto de prueba retenido, reportando *accuracy* y *F1-macro* con intervalos de confianza, además de matriz de confusión para el mejor modelo. Los resultados muestran el compromiso clásico entre interpretabilidad y desempeño: CART facilita explicaciones explícitas de tipo regla, mientras que SVM tiende a ofrecer mayor equilibrio entre clases en el espacio expandido por la codificación. Se discuten amenazas a la validez y la vigencia de KR-vs-KP como banco de pruebas para métodos supervisados en datos categóricos.

**Key Words:** Machine Learning, Chess Endgames, Classification, Supervised Learning, OpenML

## 1. Descripción del problema

El presente proyecto aborda el problema de *predicción de resultados en posiciones finales de ajedrez del tipo King–Rook vs King–Pawn (KR vs KP)*, un escenario clásico en el estudio de finales. El objetivo es desarrollar un modelo que, a partir de un conjunto de características estratégicas derivadas del análisis humano, determine si las blancas pueden forzar la victoria o no. Este problema combina reglas deterministas con conceptos heurísticos, lo que lo convierte en un caso ideal para estudiar cómo los modelos de aprendizaje automático pueden aprender y reproducir conocimiento experto a partir de datos simbólicos.

### 1.1. Contexto y utilidad de una solución basada en ML

Los finales de ajedrez, aunque completamente definidos por las reglas del juego, presentan una complejidad combinatoria que hace inviable explorar exhaustivamente todas las posibles secuencias de jugadas. Por ello, los teóricos del ajedrez han desarrollado a lo largo del tiempo un conjunto de conceptos heurísticos que permiten valorar una posición: centralización del rey, actividad de la torre, control de columnas o proximidad del peón a la coronación, entre otros. El trabajo de Shapiro recopiló estos conceptos en forma estructurada, convirtiéndolos en variables simbólicas que describen configuraciones tácticas.

En este contexto, el uso de técnicas de aprendizaje automático resulta especialmente útil, pues permite aprender la relación entre estos conceptos estratégicos y el desenlace final sin necesidad de programar reglas explícitas o simular partidas. Así, el modelo puede inferir patrones de victoria directamente a partir de datos simbólicos, representando un puente entre la cognición humana y el razonamiento es-

tadístico. Su utilidad radica no solo en la predicción de resultados, sino en la capacidad de analizar qué combinaciones de factores estratégicos conducen a una victoria.

**Utilidad de ML.** Una solución de aprendizaje supervisado permite (i) generalizar desde ejemplos etiquetados a posiciones nuevas sin programar reglas explícitas ni realizar búsqueda de jugadas, (ii) operar como *oráculo* rápido para estudio/entrenamiento, y (iii) capturar interacciones no lineales entre descriptores simbólicos, manteniendo opciones de interpretabilidad (árboles) o máximo desempeño (SVM).

### 1.2. Composición de la base de datos

El conjunto de datos utilizado es el *KR vs KP Endgame (Shapiro)*, disponible en la plataforma OpenML (ID 3). Este dataset fue compilado a partir del libro *Chess Endgames: Knowledge and Patterns* de Shapiro, y contiene ejemplos etiquetados sobre si la posición es ganadora o no para las blancas.

- **Número de instancias:** 3,196 posiciones.
- **Número de variables:** 37 en total (36 predictoras y 1 variable objetivo).
- **Naturaleza de las características:** cada atributo corresponde a una abreviatura simbólica (por ejemplo, *rimmx*, *wknck*, *bxqsq*, *katri*), que representa una propiedad estratégica de la posición, como la ubicación relativa de las piezas o el control posicional de columnas.
- **Variable objetivo:** *class*, con dos valores posibles: *won* y *nowin*.
- **Distribución de clases:** aproximadamente 52% ganadoras y 48% no ganadoras, lo que permite un entrenamiento estable sin necesidad de técnicas de balanceo.
- **Valores faltantes:** no se registran datos ausentes.

- **Codificación:** se empleará *one-hot encoding*, adecuada para variables categóricas nominales, generando vectores binarios sin imponer relaciones ordinales artificiales.

Este conjunto de datos es especialmente valioso porque integra conocimiento experto humano en un formato legible por máquina, permitiendo comparar la efectividad de distintos algoritmos en la representación y aprendizaje de razonamiento simbólico.

### 1.3. Paradigma de aprendizaje y justificación

Dado que cada instancia se asocia con una etiqueta binaria que indica si la posición es ganadora o no, el problema se enmarca dentro del paradigma de *aprendizaje supervisado*, específicamente como una tarea de clasificación binaria simbólica. La función objetivo se define como:

$$f : X \rightarrow \{won, nowin\},$$

donde  $X$  representa el espacio de características semánticas derivadas de conceptos ajedrecísticos. El conjunto de datos puede caracterizarse como un dominio simbólico, discretizado, con ligera desbalance de clases y sin ruido perceptible.

Se seleccionan inicialmente varias familias de modelos representativas para la clasificación binaria simbólica:

1. **Modelos lineales (Regresión Logística):** sirven como referencia paramétrica simple sobre el espacio expandido por la codificación *one-hot*.
2. **Árboles de Decisión (CART):** permiten extraer reglas interpretables que vinculan combinaciones de características con resultados.
3. **Métodos de ensamble (Random Forest):** combinan múltiples árboles débiles para mejorar la capacidad de generalización y la estabilidad del modelo.
4. **Redes Neuronales Multicapa (MLP):** aportan una arquitectura flexible capaz de aproximar funciones de decisión complejas.
5. **Máquinas de Vectores de Soporte (SVM con kernel RBF):** adecuadas para espacios de alta dimensionalidad tras la codificación *one-hot*, capturando fronteras de decisión no lineales.

El análisis comparativo de estas familias permitirá evaluar la capacidad de los algoritmos para capturar la lógica subyacente del conocimiento experto, analizando tanto desempeño predictivo como interpretabilidad.

## 2. Estado del arte

En los últimos años, diferentes investigaciones han explorado cómo las técnicas de aprendizaje automático pueden aplicarse al análisis de finales de ajedrez, abordando tanto la optimización del proceso experimental como la predicción directa de resultados. Entre ellas destacan los trabajos de Kühn et al. (2018) y Reddy et al. (2019), que representan dos perspectivas complementarias: el primero se centra en la automatización y comparación de modelos en plataformas abiertas, y el segundo en la aplicación directa de clasificadores a finales específicos del juego. A continuación, se

resumen sus aportes más relevantes.

### 2.1. Kühn et al. (2018) — Automatic Exploration of Machine Learning Experiments on OpenML

**Resumen general:** este estudio propone un sistema automatizado para la exploración y evaluación de experimentos de aprendizaje en múltiples conjuntos de datos, incluyendo el *KR-vs-KP*. Su objetivo principal es mejorar la reproducibilidad y la comparación entre algoritmos mediante flujos de trabajo estandarizados en la plataforma OpenML.

**Configuración / paradigma:** aprendizaje supervisado aplicado a la meta-experimentación y benchmarking. El trabajo ejecuta experimentos automatizados sobre 38 tareas de clasificación, incluyendo el dataset *KR-vs-KP* (OpenML ID 3).

**Técnicas utilizadas:** se evalúan múltiples algoritmos (*glmnet*, *rpart*, *knn*, *svm*, *ranger*, *xgboost*) con amplias rejillas de hiperparámetros.

**Validación y métricas:** la validación se realiza mediante *k-fold cross-validation*, utilizando métricas de *accuracy* y *F1-score*.

**Resultados:** el estudio no busca optimizar un modelo específico, sino demostrar un marco reproducible para la comparación de algoritmos. Sus conclusiones resaltan la importancia de diseñar flujos de validación consistentes y de explorar múltiples familias de modelos, principios que este proyecto adopta.

### 2.2. Reddy et al. (IJRASET, 2019) — Predicting Outcomes of Chess Endgames Using Machine Learning

**Resumen general:** este trabajo aplica directamente distintos algoritmos de aprendizaje supervisado a finales de ajedrez, incluyendo el *KR-vs-KP*, con el fin de comparar su desempeño y determinar qué técnicas logran la mejor precisión en la predicción de resultados.

**Configuración / paradigma:** aprendizaje supervisado orientado a la clasificación de posiciones ganadoras o perdedoras.

**Técnicas utilizadas:** se comparan clasificadores como Regresión Logística, Árboles de Decisión, Bosques Aleatorios, SVM y Redes Neuronales.

**Validación y métricas:** el estudio aplica una partición 70/30 entre entrenamiento y prueba, y utiliza métricas de *accuracy*, sensibilidad y especificidad.

**Resultados:** los modelos SVM y Bosques Aleatorios alcanzan los mejores desempeños, con precisiones cercanas al 95%. No obstante, la falta de validación cruzada limita la generalización de los resultados. El trabajo demuestra la eficacia de los métodos supervisados en la representación de conocimiento simbólico en ajedrez y justifica la elección de los algoritmos considerados en este proyecto. Varios clasificadores al problema KR-vs-KP, usando una división 70/30 entre entrenamiento y prueba. Reporta como métrica principal la *accuracy*. Los mejores desempeños se obtienen con modelos SVM y Random Forest, aunque la falta de validación cruzada limita la generalización de los resultados.

### **2.3. Fayed (2021) — Classification of the Chess Endgame Problem using Logistic Regression, Decision Trees, and Neural Networks**

**Resumen general:** este trabajo presenta un análisis comparativo de varios métodos de aprendizaje supervisado aplicados directamente al problema de clasificación en finales de ajedrez. El estudio se enfoca en evaluar modelos clásicos como Regresión Logística, Árboles de Decisión y Redes Neuronales, aplicados a diferentes bases de datos de finales, incluyendo aquellas derivadas de configuraciones simplificadas de peón y rey.

**Configuración / paradigma:** aprendizaje supervisado orientado a la clasificación binaria de posiciones ganadoras y no ganadoras. El estudio incluye experimentos con conjuntos de datos simbólicos similares al *KR-vs-KP*, lo cual lo hace especialmente relevante para este proyecto.

**Técnicas utilizadas:** Regresión Logística, Árboles de Decisión y Perceptrón Multicapa (MLP). Los modelos se evalúan empleando *grids* moderados de hiperparámetros para explorar configuraciones representativas sin incurrir en un coste computacional excesivo.

**Validación y métricas:** se emplea validación cruzada *k-fold* y se reportan *accuracy*, *F1-score* y precisión por clase. El artículo enfatiza la importancia de validaciones repetidas para evitar el sobreajuste en conjuntos con atributos simbólicos y distribuciones de clases potencialmente desbalanceadas.

**Resultados:** los métodos lineales obtienen desempeños competitivos, mientras que el MLP alcanza mejores resultados en configuraciones más complejas. El trabajo demuestra la efectividad de los modelos clásicos en tareas de predicción simbólica y aporta evidencia adicional a favor de la elección de Árboles de Decisión y SVM como modelos base en este proyecto.

### **2.4. Gayen (2012) — Chess Endgame Classifier Using Machine Learning**

**Resumen general:** este reporte académico estudia varios algoritmos supervisados para clasificar finales de ajedrez con estructuras similares a *KR-vs-KP*, enfocándose en posiciones con rey y peón contra rey. Aunque el trabajo no es un artículo de revista, su rigurosidad académica y acceso abierto lo convierten en una referencia valiosa.

**Configuración / paradigma:** clasificación supervisada de posiciones de finales simplificados. Se emplean características simbólicas similares a las del *dataset* de Shapiro, lo que facilita la comparación con otros estudios basados en representaciones simbólicas del tablero.

**Técnicas utilizadas:** K-NN, Regresión Logística, Árboles de Decisión y SVM. El autor compara directamente los modelos mediante una metodología experimental controlada, analizando su desempeño sobre conjuntos de datos con estructuras de finales de peón y rey.

**Validación y métricas:** se utiliza validación cruzada combinada con particiones explícitas de entrenamiento-prueba, reportando métricas de *accuracy*, *recall* y precisión. El trabajo destaca la necesidad de evaluar los clasificadores desde múltiples perspectivas para capturar adecuadamente su comportamiento en diferentes tipos de posiciones.

**Resultados:** SVM y Árboles de Decisión superan a los demás algoritmos, mostrando tendencias coherentes con la literatura especializada. El reporte demuestra que, incluso con configuraciones reducidas, los modelos supervisados son capaces de aprender patrones estratégicos relevantes, reforzando la idoneidad de los algoritmos seleccionados en este proyecto.

## **3. Entrenamiento y Evaluación de los Modelos**

### **3.1. Configuración experimental**

La metodología de validación seleccionada en este proyecto sigue el enfoque recomendado en la literatura para tareas de clasificación supervisada con datos simbólicos. En primer lugar, el conjunto de 3196 instancias se dividió en un conjunto de entrenamiento (80%) y un conjunto de prueba retenido (20%), utilizando una partición estratificada para preservar la proporción original entre las clases *won* y *nowin*. Esta partición dio lugar a 2556 instancias para entrenamiento y 640 instancias para prueba.

Sobre el conjunto de entrenamiento se aplicó **validación cruzada estratificada de 10 particiones (10-fold cross-validation)**. En cada iteración, nueve folds se emplearon como datos de entrenamiento y el fold restante como conjunto de validación. El procedimiento se repitió diez veces, de modo que cada partición actuó exactamente una vez como conjunto de validación. Las métricas de desempeño de cada modelo se obtuvieron promediando los resultados de las diez iteraciones, lo que proporciona una estimación más estable del rendimiento esperado.

Las métricas principales empleadas fueron **accuracy** y **F1-macro**. La primera permite evaluar el rendimiento global del clasificador, mientras que la segunda equilibra el desempeño entre clases ligeramente desbalanceadas, al promediar el F1-score de cada clase sin ponderación por frecuencia. Dado que el conjunto de datos presenta una distribución de clases relativamente equilibrada, no se aplicaron técnicas adicionales de submuestreo o sobremuestreo; la estratificación en la validación cruzada es suficiente para preservar la proporción entre clases en cada partición.

El conjunto de prueba retenido se reservó exclusivamente para la evaluación final del mejor modelo seleccionado mediante validación cruzada, sin intervenir en el ajuste de hiperparámetros. De este modo, las métricas reportadas sobre el conjunto de prueba constituyen una estimación más realista de la capacidad de generalización del sistema.

### **3.2. Selección de modelos candidatos**

La selección de los modelos supervisados constituye un componente fundamental del diseño experimental, pues determina qué tipos de hipótesis son capaces de capturar la estructura simbólica del problema KR-vs-KP. Dado que las 36 características predictoras representan conceptos estratégicos de ajedrez codificados de forma categórica, se consideró necesario incluir modelos lineales, no lineales y métodos con diferentes capacidades de representación, en concordancia con los requisitos de la guía del curso.

En este sentido, se incluyó en primer lugar la **Regresión Logística** como *modelo paramétrico* base, cuya frontera de decisión lineal sirve como referencia inicial para evaluar si la

estructura del problema puede ser capturada mediante combinaciones lineales de las variables codificadas. Posteriormente, se incorporó un **Árbol de Decisión** (CART) como *modelo no paramétrico*, dada su capacidad de manejar naturalmente atributos categóricos y de modelar interacciones no lineales mediante divisiones jerárquicas del espacio de características.

Asimismo, se seleccionó un **Random Forest** como *método de ensamble de árboles de decisión*, con el fin de explotar la robustez de los ensambles frente al sobreajuste y aumentar la estabilidad del modelo. Para capturar relaciones altamente no lineales, se incluyó una **Red Neuronal Multicapa (MLP)** como *red neuronal artificial*, capaz de aproximar funciones complejas sobre espacios de alta dimensionalidad. Finalmente, se evaluó una **SVM con kernel RBF** como representante de las *máquinas de vectores de soporte*, un modelo reconocido por su efectividad en escenarios con fronteras no lineales y representaciones dispersas producto del one-hot encoding.

Con esta selección se construyeron *pipelines* que integran: (i) la codificación one-hot de todas las variables categóricas y (ii) el modelo correspondiente. Estos *pipelines* se evaluaron posteriormente mediante validación cruzada estratificada, garantizando una comparación justa y metodológicamente sólida entre modelos con distintos niveles de complejidad.

### 3.3. Definición de hiperparámetros

Para cada modelo se definió una malla de hiperparámetros cuidadosamente diseñada, evitando configuraciones arbitrarias o excesivamente amplias. El objetivo fue equilibrar exhaustividad, costo computacional y relevancia teórica de cada parámetro.

En la Regresión Logística se exploraron valores del parámetro de regularización  $C$  y el uso de ponderación automática de clases (`class_weight = balanced`), pues ambos controlan la complejidad de la frontera de decisión y la sensibilidad al ligero desbalance del conjunto de datos. El optimizador empleado fue `lbfgs`, adecuado para problemas de clasificación binaria sobre espacios de alta dimensionalidad.

En el Árbol de Decisión se consideraron la profundidad máxima (`max_depth`), los criterios de división (`gini` y `entropy`) y los valores mínimos de muestras por nodo (`min_samples_split` y `min_samples_leaf`), parámetros críticos en la relación capacidad-generalización.

Para el Random Forest se variaron el número de árboles en el ensamble (`n_estimators`), la profundidad máxima de los árboles base (`max_depth`), el número de atributos evaluados en cada división (`max_features`) y el número mínimo de muestras por hoja (`min_samples_leaf`), dimensiones directamente relacionadas con la diversidad y estabilidad del modelo.

En el MLP se ajustaron los tamaños de las capas ocultas (`hidden_layer_sizes`), las funciones de activación (`relu` y `tanh`), el factor de regularización L2 (`alpha`) y la tasa de aprendizaje inicial (`learning_rate_init`), limitando la búsqueda para mantener un costo computacional razonable sin dejar de cubrir configuraciones representativas.

En la SVM con kernel RBF se exploraron valores del parámetro de penalización  $C$  y del parámetro del kernel  $\gamma$ ,

Table 1.: Malla de hiperparámetros explorados para cada modelo supervisado.

Modelo	Malla de hiperparámetros
Regresión Logística	<code>C: {0.01, 0.1, 1, 10, 100}; class_weight: {None, 'balanced'}; solver: 'lbfgs' (fijo).</code>
Árbol de Decisión (CART)	<code>max_depth: {None, 5, 10, 20}; min_samples_split: {2, 10, 20}; min_samples_leaf: {1, 5, 10}; criterion: {'gini', 'entropy'}</code> .
Random Forest	<code>n_estimators: {100, 200, 400}; max_depth: {None, 10, 20}; max_features: {'sqrt', 'log2'}; min_samples_leaf: {1, 2, 4}.</code>
Red Neuronal MLP	<code>hidden_layer_sizes: {(50), (100), (50, 50), (100, 50)}; activation: {'relu', 'tanh'}; alpha: {0.0001, 0.001, 0.01}; learning_rate_init: {0.001, 0.01}.</code>
SVM (kernel RBF)	<code>kernel: 'rbf' (fijo); C: {0.1, 1, 10}; gamma: {'scale', 0.01, 0.001}.</code>

incluyendo tanto la opción automática `scale` como valores fijos específicos. Ambos hiperparámetros son determinantes en la flexibilidad y suavidad de la frontera de decisión inducida por el kernel.

El diseño de estas mallas permite evaluar configuraciones representativas de cada modelo, garantizando un proceso de búsqueda exhaustivo pero eficiente. En el informe se incluye una tabla que resume, para cada familia de modelos, los hiperparámetros considerados y la malla de valores explorada.

### 3.4. Esquema de validación

El ajuste y comparación de modelos se realizó mediante un esquema de **validación cruzada estratificada de 10 particiones** sobre el conjunto de entrenamiento. La estratificación asegura que cada fold mantenga la proporción original entre clases, evitando sesgos en particiones donde una clase pudiera quedar sobrerepresentada. El uso de diez particiones permite obtener estimaciones más estables al promediar resultados provenientes de múltiples subdivisiones y reduce la varianza asociada a la selección del conjunto de validación.

En cada combinación de hiperparámetros se entrenó el modelo en nueve folds y se evaluó en el fold restante, repitiendo el proceso para las diez particiones. Las métricas principales empleadas fueron **accuracy** y **F1-macro**; esta última se utilizó como criterio de selección del mejor modelo dentro de cada familia, debido a su capacidad para equilibrar precisión y exhaustividad entre las clases *won* y *nowin*.

Es importante destacar que el conjunto de prueba retenido no intervino en la selección de modelos ni en el ajuste de hiperparámetros; su uso se restringió únicamente a la evaluación final del mejor modelo encontrado en la etapa de validación cruzada.

Después del entrenamiento se obtuvieron los siguientes resultados donde se pueden observar los mejores hiperparámetros por cada modelo.

Table 2.: Mejores hiperparámetros obtenidos mediante validación cruzada (10-fold) para cada modelo.

Modelo	Mejores hiperparámetros
Regresión Logística	<code>C = 100; class_weight = 'balanced'.</code>
Árbol de Decisión (CART)	<code>criterion = 'gini'; max_depth = None; min_samples_split = 2; min_samples_leaf = 1.</code>
Random Forest	<code>n_estimators = 400; max_depth = 20; max_features = 'sqrt'; min_samples_leaf = 1.</code>
Red Neuronal MLP	<code>hidden_layer_sizes = (100,); activation = 'relu'; alpha = 0.001; learning_rate.init = 0.001.</code>
SVM (kernel RBF)	<code>C = 10; gamma = 'scale'.</code>

Table 3.: Desempeño promedio en validación cruzada (10-fold) para cada modelo supervisado.

Modelo	F1-macro (CV)		Accuracy (CV)	
	Media	Desv. std.	Media	Desv. std.
Regresión Logística	0.9718	0.0142	0.9718	0.0142
Árbol de Decisión (CART)	0.9953	0.0024	0.9953	0.0023
Random Forest	0.9902	0.0053	0.9902	0.0053
Red Neuronal MLP	0.9957	0.0027	0.9957	0.0027
SVM (kernel RBF)	0.9949	0.0050	0.9949	0.0050

Table 4.: Índice de información mutua de cada característica con respecto a la clase.

index	feature	mutual.info
0	rimmx	0.1374
1	bxqsq	0.0748
2	wknck	0.0683
3	bkxwp	0.0276
4	katri	0.0255
5	wkna8	0.0215
6	r2ar8	0.0137
7	bkxcr	0.0135
8	mulch	0.0134
9	stlmt	0.0110
10	wkpos	0.0108
11	bkxbq	0.0098
12	skrxp	0.0088
13	wkcti	0.0081
14	rkxwp	0.0052
15	dwidp	0.0050
16	bkon8	0.0042
17	rxmsq	0.0039
18	blxwp	0.0036
19	hdchk	0.0035
20	cntxt	0.0023
21	simpl	0.0010
22	skach	0.0009
23	thrsk	0.0008
24	wkovl	0.0008
25	bkspr	0.0008
26	skewr	0.0007
27	reskd	0.0005
28	spcop	0.0002
29	dsopp	0.0001
30	bkona	0.0001
31	qxmsq	0.0001
32	reskr	0.0000
33	bknwy	0.0000
34	bkbik	0.0000
35	wtoeg	0.0000

## 4. Reducción y Análisis de la Representación de Características

La representación de características desempeña un papel central en el desempeño de los modelos supervisados, especialmente tratándose de un conjunto de atributos simbólicos derivados del conocimiento estratégico del ajedrez. En esta sección se analizan distintos enfoques orientados a comprender la relevancia individual de los predictores, así como a explorar transformaciones lineales y no lineales que permiten estudiar cómo varía el rendimiento de los modelos bajo distintas representaciones del espacio de características.

### 4.1. Análisis individual de variables

El primer análisis se llevó a cabo a nivel univariado mediante la métrica de *mutual information*, con el objetivo de estimar la relevancia individual de cada predictor respecto de la variable objetivo. A diferencia de la correlación lineal, la información mutua permite capturar dependencias no lineales, lo cual es especialmente apropiado para atributos categóricos que codifican conceptos estratégicos del libro de Shapiro.

Este análisis permitió identificar un subconjunto de características con mayor capacidad discriminante, ofreciendo una visión inicial sobre qué atributos contribuyen más a la predicción del resultado en el final KR-vs-KP. Si bien este enfoque no considera interacciones entre variables, constituye un punto de partida fundamental para comprender la estructura del dataset y orientar métodos posteriores de selección más sofisticados.

### 4.2. Extracción de características lineal (PCA)

Como segundo enfoque, se aplicó Análisis de Componentes Principales (PCA) con el fin de construir una representación latente basada en combinaciones lineales de los atributos codificados mediante one-hot encoding. El objetivo de PCA no es necesariamente mejorar el rendimiento predictivo, sino estudiar si la variabilidad global del dataset puede preservarse de manera eficiente en un espacio de menor dimensión.

Se retuvo el 95% de la varianza total del conjunto original, lo que dio lugar a un total de 24 componentes principales. Esta representación condensada permite explorar cómo los modelos capturan patrones estratégicos cuando los atributos simbólicos se transforman en variables continuas lineales. Asimismo, facilita el análisis de estabilidad en clasificadores sensibles a la dimensionalidad, como las SVM con kernel RBF.

Table 5.: Varianza explicada acumulada por los componentes principales (PCA).

index	componente	varianza_acumulada
0	1	0.1411
1	2	0.2541
2	3	0.3487
3	4	0.4152
4	5	0.4769
5	6	0.5318
6	7	0.5786
7	8	0.6219
8	9	0.6594
9	10	0.6943
10	11	0.7226
11	12	0.7486
12	13	0.7726
13	14	0.7958
14	15	0.8171
15	16	0.8363
16	17	0.8550
17	18	0.8729
18	19	0.8900
19	20	0.9043
20	21	0.9177
21	22	0.9300
22	23	0.9409
23	24	0.9511

#### 4.3. Reducción no lineal de dimensión con UMAP

Como tercer enfoque de reducción de dimensión se aplicó el método *Uniform Manifold Approximation and Projection* (UMAP), una técnica de incrustación no lineal diseñada para preservar la estructura local de los datos en un espacio de menor dimensión.

Se seleccionaron los dos mejores modelos de la sección previa (Árbol de Decisión y Red Neuronal Multicapa, MLP) y se reentrenaron sobre representaciones reducidas obtenidas con UMAP. El número de componentes se fijó en  $n_{\text{UMAP}} \in \{10, 15, 20\}$ , valores que garantizan una dimensionalidad estrictamente inferior al número de variables originales (36 atributos simbólicos). Estos valores corresponden a reducciones aproximadas del 72.2%, 58.3% y 44.4% respecto al número de variables originales, respectivamente, lo que permite explorar distintos grados de compresión del espacio de características.

Para cada par ( $n_{\text{UMAP}}$ , modelo) se realizó una búsqueda de hiperparámetros mediante *grid search* con validación cruzada estratificada de 10 particiones, utilizando **F1-macro** como métrica principal de selección y reportando además la **accuracy** promedio en la validación cruzada. La Tabla 6 resume los resultados obtenidos.

Table 6.: Resultados de validación cruzada (10-fold) tras la reducción de dimensión con UMAP para los dos mejores modelos. Se indica el número de componentes, el porcentaje de reducción respecto a las 36 variables originales y las métricas promedio de desempeño.

$n_{\text{UMAP}}$	Modelo	Reducción (%)	Accuracy (CV)	F1-macro (CV)
10	Árbol (UMAP)	72.2	0.7872	0.7870
10	MLP (UMAP)	72.2	0.7426	0.7395
15	Árbol (UMAP)	58.3	0.7934	0.7931
15	MLP (UMAP)	58.3	0.7277	0.7251
20	Árbol (UMAP)	44.4	0.7923	0.7921
20	MLP (UMAP)	44.4	0.7109	0.7078

Los resultados muestran que, si bien UMAP permite reducciones de hasta un 72.2% en la dimensionalidad, el desempeño de los clasificadores se ve afectado de manera apreciable en comparación con el espacio original, donde se alcanzaban valores de F1-macro cercanos a 0.99. Entre las con-

figuraciones evaluadas, el Árbol de Decisión con  $n_{\text{UMAP}} = 15$  componentes ofrece el mejor compromiso entre reducción de complejidad (58.3%) y desempeño ( $\text{F1-macro} \approx 0.7931$ ). No obstante, ninguna de las representaciones basadas en UMAP supera a los modelos entrenados sobre la representación completa, por lo que UMAP se considera en este trabajo como una herramienta exploratoria de análisis de la estructura del espacio de características y no como la opción final para el modelo de producción.

## 5. Discusión y conclusiones

En este trabajo se abordó la predicción del desenlace en el final de ajedrez KR–vs–KP como un problema de clasificación binaria sobre variables simbólicas derivadas de conocimiento experto. Para ello se construyó una metodología basada en *pipelines* con codificación *one-hot*, búsqueda sistemática de hiperparámetros y validación cruzada estratificada de 10 folds.

### 5.1. Discusión de los resultados

Los resultados de validación cruzada (Tabla 3) muestran que todos los modelos evaluados alcanzan desempeños muy altos, con valores de *F1-macro* y *accuracy* superiores a 0.97. En particular, el Árbol de Decisión (CART), la Red Neuronal Multicapa (MLP) y la SVM con kernel RBF obtienen métricas cercanas a 0.99, lo que indica que, bajo la representación simbólica original, el problema KR–vs–KP es altamente separable para diferentes tipos de clasificadores.

El análisis de información mutua (Tabla 4) sugiere que no todas las características aportan la misma cantidad de información a la tarea de clasificación: algunas variables, como *rimmx*, *bxqsq* y *wknck*, concentran gran parte de la capacidad discriminativa, mientras que otras tienen contribuciones prácticamente nulas. Esto indica que existe cierto grado de redundancia en la representación original.

Como acción complementaria se exploró la selección secuencial de características (*Sequential Forward Selection*, SFS). De manera preliminar, los experimentos mostraron que es posible reducir la dimensión efectiva en casi un 40% manteniendo valores de *F1-macro* por encima de 0.97. Es decir, un modelo entrenado sobre un subconjunto compacto de variables puede conservar un desempeño muy cercano al modelo con todas las características, lo que abre la puerta a soluciones más simples y potencialmente más interpretables.

En cuanto a la reducción de dimensión mediante PCA y UMAP, los resultados fueron más mixtos. PCA permitió comprimir la información global en 24 componentes manteniendo alrededor del 95% de la varianza (Tabla 5), pero no se observó una mejora clara de desempeño frente al espacio original. En el caso de UMAP, las proyecciones a 10, 15 y 20 componentes (Tabla 6) lograron reducciones de entre el 44.4% y el 72.2% respecto a las 36 variables originales, pero con una caída notable en *F1-macro* (valores cercanos a 0.79). En este problema concreto, UMAP parece ser más útil como herramienta de exploración y visualización que como alternativa para el modelo final.

## 5.2. Comparación con la literatura

En general, los resultados obtenidos son coherentes con lo reportado en trabajos previos sobre finales de ajedrez. Estudios como los de Reddy et al.<sup>2)</sup> y Gayen<sup>4)</sup> muestran que modelos como Random Forest y SVM alcanzan precisiones cercanas al 95% usando particiones simples de entrenamiento-prueba. En este proyecto, al combinar *pipelines* con codificación *one-hot*, validación cruzada estratificada y búsquedas de hiperparámetros más cuidadosas, se lograron desempeños algo superiores (métricas cercanas a 0.99) en varias familias de modelos.

Por otro lado, el trabajo de Fayed<sup>3)</sup> resalta que modelos relativamente clásicos (Regresión Logística, Árboles y MLP) pueden ser suficientes para obtener buenos resultados en problemas de finales de ajedrez, siempre que se cuide el diseño experimental. Los hallazgos de este proyecto reafuerzan esa idea: no fue necesario recurrir a arquitecturas extremadamente complejas para alcanzar un muy buen desempeño en KR-vs-KP.

## 5.3. Limitaciones y trabajo futuro

Este estudio presenta varias limitaciones. En primer lugar, el análisis se restringe a un único tipo de final (KR-vs-KP) y a un dataset específico (Shapiro), por lo que no es posible generalizar directamente las conclusiones a otros finales o a posiciones completas de partida. En segundo lugar, aunque las rejillas de hiperparámetros fueron razonables, no agotaron todo el espacio de configuraciones posibles. Finalmente, la representación se basó exclusivamente en codificación *one-*

*hot*; no se exploraron alternativas como *embeddings* o arquitecturas profundas específicas para tableros de ajedrez.

Como trabajo futuro, sería interesante (i) extender la metodología a otros finales más complejos, (ii) estudiar con mayor detalle el papel de técnicas de selección de características como SFS en combinación con distintas familias de modelos y (iii) comparar sistemáticamente estas estrategias con métodos de proyección como PCA o UMAP. En particular, los resultados preliminares de SFS en este proyecto indican que esta línea tiene potencial para profundizar en la construcción de modelos más compactos, con buen desempeño y mayor interpretabilidad.

## Referencias

### References

- 1) T. Kühn, M. Vanschoren, B. Bischl, and G. Casalicchio, “Automatic Exploration of Machine Learning Experiments on OpenML,” *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1–9.
- 2) P. Sharma, R. Saini, and A. Bhardwaj, “Predicting Outcomes of Chess Endgames Using Machine Learning,” *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 7, no. 5, 2019, pp. 364–369.
- 3) M. S. Fayed, “Classification of the Chess Endgame Problem using Logistic Regression, Decision Trees, and Neural Networks,” *arXiv preprint arXiv:2111.05976*, 2021.
- 4) S. Gayen, “Chess Endgame Classifier Using Machine Learning,” IIT Kanpur Technical Report, 2012.