

# Sistema Inteligente de detección de errores tácticos en Ajedrez



UNIVERSIDAD  
DE GRANADA

Miguel Moles Mestre

Tutores:

Francisco Javier Melero Rus

Francisco Herrera Triguero

Universidad de Granada

25 de julio de 2023

# Presentación

- 1 Introducción
- 2 Errores
- 3 Características
- 4 Clustering
- 5 Validación
- 6 Caso Práctico
- 7 Conclusiones
- 8 Fin

# Ajedrez



# Historia de los motores de ajedrez



**Figure:** Mac Hack VI  
(1967)

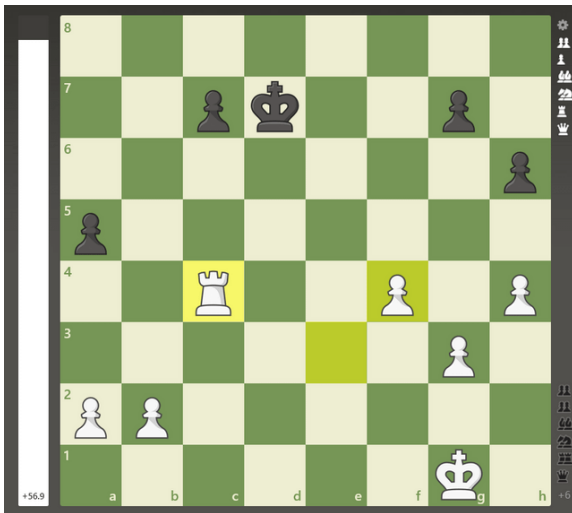


**Figure:** Kasparov vs  
32 motores(1985)

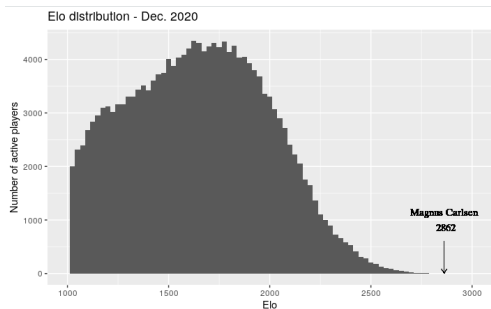


**Figure:** Kasparov vs  
Deep Blue (1997)

# Motores en la actualidad



# IA y ajedrez: ¿Ahora qué?



 Aimchess

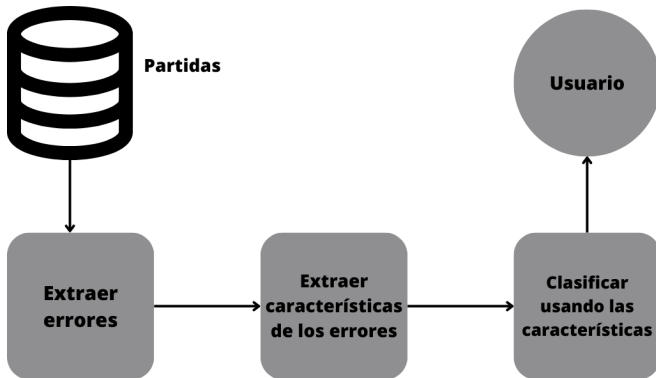


# Clasificar errores de ajedrez

- 1 Localizar los fallos comunes en el juego de una persona
  - para mejorar
  - para un futuro oponente
- 2 Obtener posiciones similares a los errores
- 3 Estudiar la existencia de tipos disjuntivos de posiciones

# Objetivos

Usar **aprendizaje automático no supervisado** para clasificar errores de ajedrez.





# Software Usado

Lenguaje de programación



Librerías generales



Extracción de errores

AsyncIO

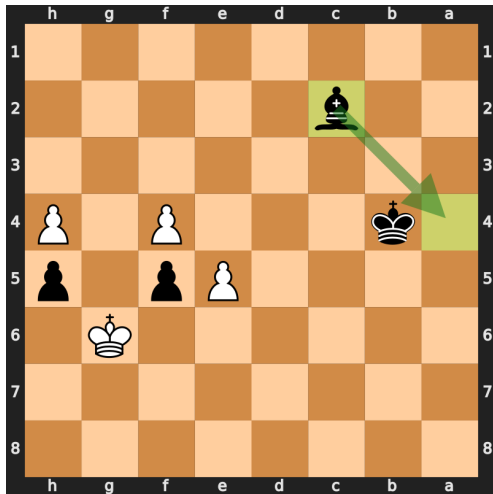
Extracción de características



Clustering de errores



# Error de Ajedrez

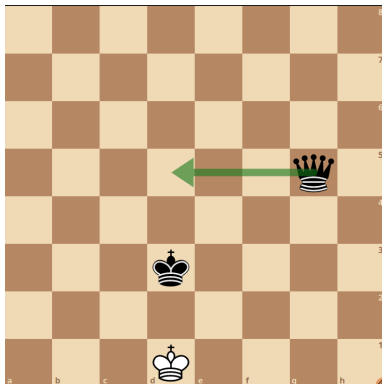


Evaluaciones de Stockfish:

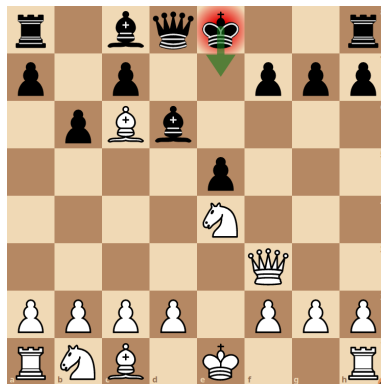
- 1 Antes: -0.1
- 2 Ahora: 8.2
- 3 Evaluación del movimiento: -8.3

# Errores no deseados

## Posiciones muy ganadas



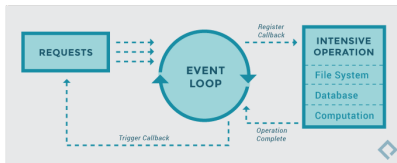
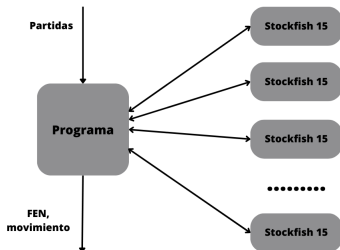
## Posiciones muy perdidas



# Programa de extracción: objetivos

- Velocidad
- Calidad de análisis
- Búsquedas cortas

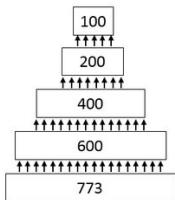
# Paralelización



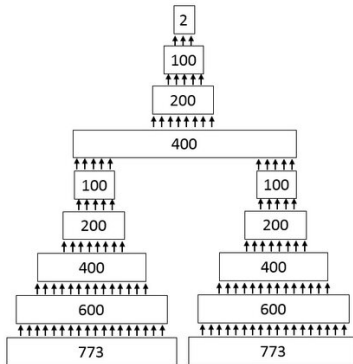
# Requisitos para clustering

- Medir semejanzas
- **Distancias**
- **Vector de características**

# DeepChess



Stage 1: DBN (Pos2Vec)



Stage 2: Supervised Training (DeepChess)

# Mov2Vec y Dif2Vec

## Penúltima y última capa

$$\sum_{i=1}^{100} (w_{0,i} * out_i) (>, <) \sum_{i=1}^{100} (w_{1,i} * out_i)$$

### Mov2Vec:

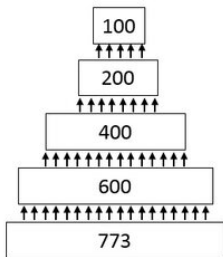
$$(w_{0,i} * out_i | i = 1 \dots 100) \cup (w_{1,i} * out_i | i = 1 \dots 100)$$

### Dif2Vec:

$$((w_{0,i} - w_{1,i}) * out_i | i = 1 \dots 100)$$



# Pos2Vec 1 y 2



## 1 Pos2Vec1

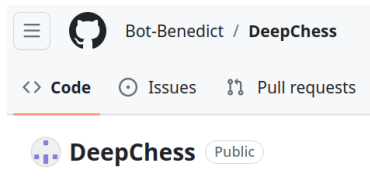
- Antes del entrenamiento completo

## 2 Pos2Vec2

- Después del entrenamiento completo

# Implementación

- 1 Crear modelos vacíos
- 2 Introducir pesos
- 3 Guardar los 4 modelos



# Base de datos

- Lichess, media 2400
- Extraídos 501.858 errores
- Transformados con mov2vec, dif2vec, pos2vec1 y pos2vec2

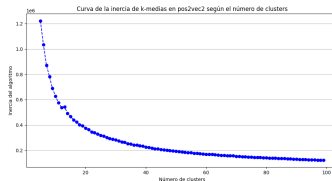
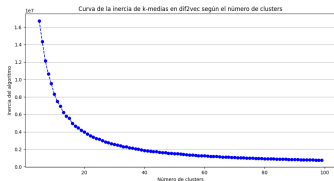
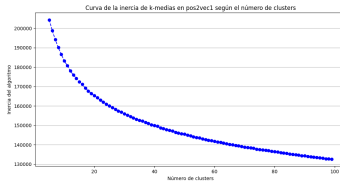
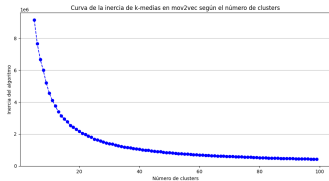
# Características 0

	Porcentaje no 0	Caract. no 0
mov2vec	10.1%	30 de 200
dif2vec	10.1%	15 de 100
pos2vec1	31.9%	85 de 100
pos2vec2	6.21%	48 de 100

# Acumular características

	Caract. acumuladas	Caract. finales
mov2vec	6 → 2	26
dif2vec	3 → 1	13
pos2vec1	35 → 2	52
pos2vec2	33 → 1	16

# Método del codo para el número de clusters

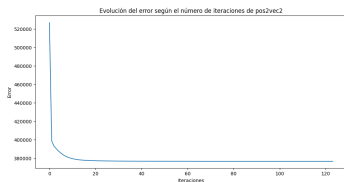
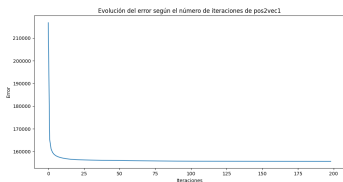
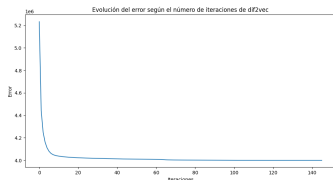
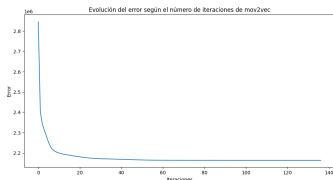


# Número de clusters final

mov2vec	20
dif2vec	20
pos2vec1	30
pos2vec2	20

# K-medias

Hasta **absoluta convergencia** o 200 iteraciones.





# Etiquetas tácticas

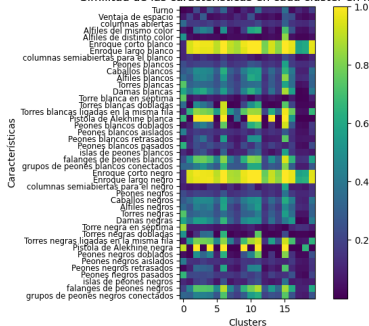
- 38 etiquetas
- Motivos tácticos en ambos bandos

## Medida de calidad de separación:

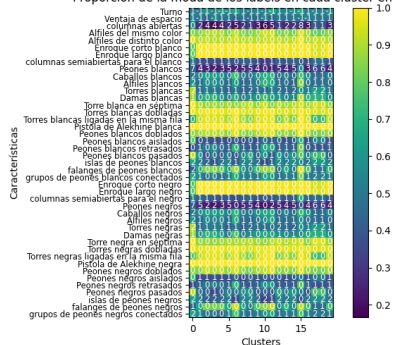
$$Punt(etiq, cluster) = \max_{x \in etiq} \left( \frac{pIntracluster_x - pTotal_x}{1 - pTotal_x} \right)$$

# Mov2Vec

Similitud de las características en cada cluster en mov2vec

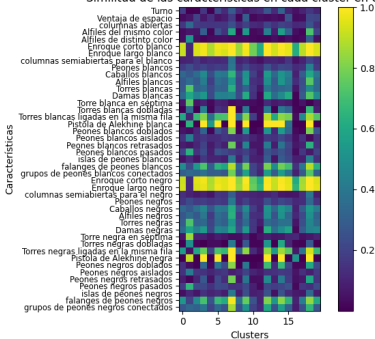


Proporción de la moda de los labels en cada cluster en mov2vec

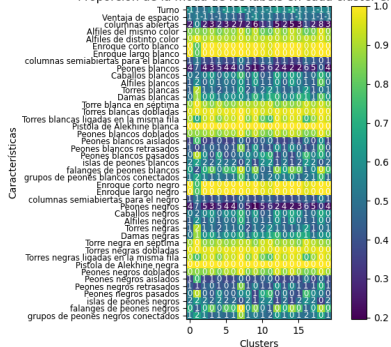


# Dif2Vec

Similitud de las características en cada cluster en dif2vec

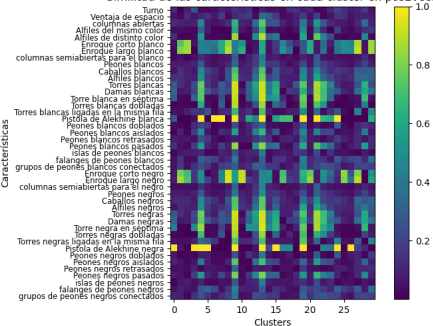


Proporción de la moda de los labels en cada cluster en dif2vec

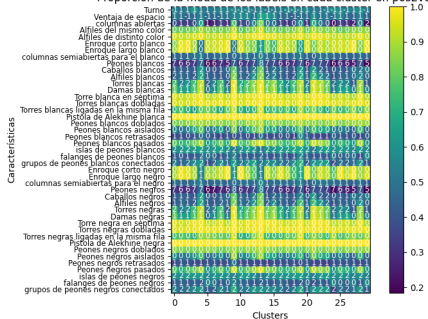


# Pos2Vec1

Similitud de las características en cada cluster en pos2vec1

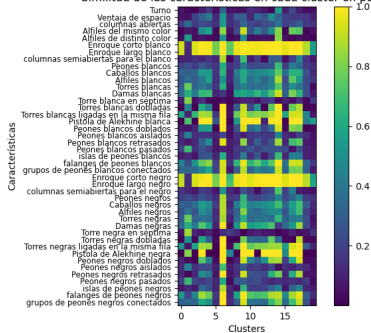


Proporción de la moda de los labels en cada cluster en pos2vec1

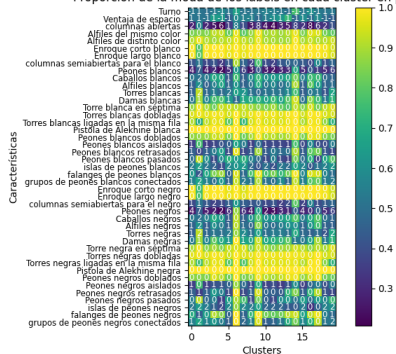


# Pos2Vec2

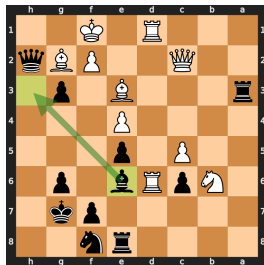
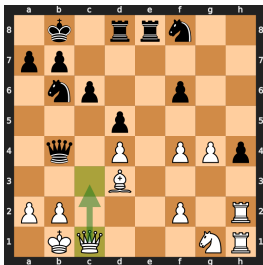
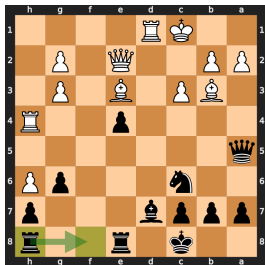
Similitud de las características en cada cluster en pos2vec2



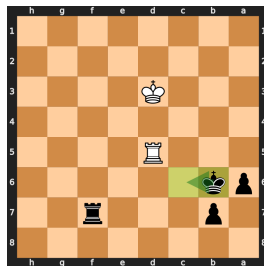
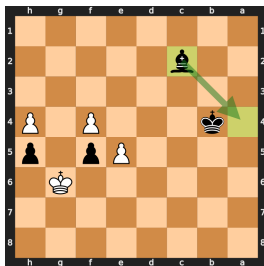
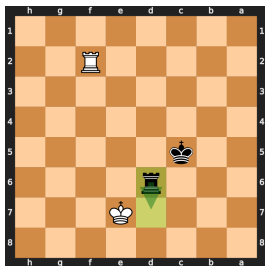
Proporción de la moda de los labels en cada cluster en pos2vec2



# Cluster 9, Dif2Vec



# Cluster 16, Dif2Vec



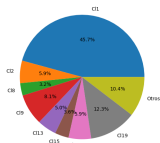
# Datos disponibles

- 1 **William Steinitz** (1836-1900)
- 2 590 partidas
- 3 3020 errores extraídos

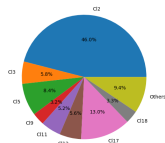


# Proporciones de clusters

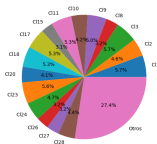
Porcentaje de pertenencia de los datos a los clusters de mov2vec



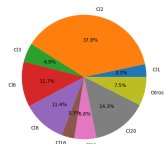
Porcentaje de pertenencia de los datos a los clusters de dif2vec



Porcentaje de pertenencia de los datos a los clusters de pos2vec1



Porcentaje de pertenencia de los datos a los clusters de pos2vec2



## Resultados prometedores

- Clusters que separan bien
- La mayoría de ellos mejorable

Pos2Vec2 > Mov2Vec y Dif2Vec >>> Pos2Vec1

Redes neuronales: difícil obtener información de ellas

DeepChess y Stockfish no coinciden mucho

# Trabajo futuro

## Experimentar:

- Otras bases de datos
- Diferentes definiciones de error
- Diferentes distancias
- Más preprocesado
- Otros algoritmos de clustering

Implementar una interfaz para un usuario

Usar más etiquetas de tácticos

Mejorar la interpretación de los casos prácticos

# ¡Gracias por su atención!