Tabla de contenidos

Abstracto

En el Ejercicio 1 realizamos un estudio de clustering sobre el dataset de 734 Pokémon, con el objetivo de comprobar si existen agrupaciones "naturales" según características (estadísticas base y otras variables derivadas) y si estas se alinean con los tipos oficiales. Para ello, construimos un conjunto de variables numéricas a partir del JSON, incluyendo las estadísticas de combate (hp, atk, def, spa, spd, spe) y otras variables (n_moves, n_types, n_abilities, hasEvo). En primer lugar, realizamos un análisis exploratorio de datos (EDA) para estudiar las diferentes variables, sus estadísticas básicas, distribuciones, valores faltantes y atípicos, así como las correlaciones entre variables numéricas y la relación entre los tipos de Pokémon y sus estadísticas base, entre otras técnicas, con el fin de obtener una mayor comprensión y una mejor visión global del conjunto de datos. A partir de este EDA, se llevó a cabo el preprocesamiento necesario para escalar las variables numéricas.

Una vez hecho esto, entrenamos y comparamos los modelos K-Means y DBSCAN bajo distintas configuraciones de valores (k, eps, min_samples), evaluando la calidad de los agrupamientos mediante el Silhouette Score y el método del codo. En K-Means se exploraron diferentes valores de k (2…10), seleccionando k = 2 como opción óptima en base a métricas de evaluación como el método del codo y, principalmente, el Silhouette Score, donde el modelo presentó un valor de ≈ 0.271 para k = 2. En DBSCAN se utilizó el gráfico de distancias k-NN como guía para definir el rango de valores de eps a probar y se evaluaron combinaciones junto a distintos valores de min_samples, obteniendo como mejor configuración eps = 2.5 y min_samples = 3, que genera 2 clústeres, 27 puntos de ruido y un Silhouette Score de ≈ 0.211.

El análisis muestra que los clústeres se explican sobre todo por diferencias en estadísticas. Las mayores separaciones aparecen en las estadísticas atk, spa, spd, def, hp y spe. Además, hasEvo presenta una diferencia de medias de ≈ 0.93, lo que indica una separación casi total entre Pokémon con evolución y sin evolución entre clústeres. Al contrastar los grupos con los tipos oficiales, no se observa una alineación fuerte: los clústeres mezclan tipos y reflejan más bien perfiles funcionales basados en las estadísticas base de los Pokémon.

En el Ejercicio 2 usamos agrupamiento jerárquico aglomerativo para agrupar Pokémon del dataset (734 en total) según sus características. Primero preparamos los datos: convertimos los tipos a variables numéricas (one-hot) y escalamos las variables para que todas cuenten de forma parecida en las distancias. Después probamos varios criterios de enlace (single, complete, average, ward) y distintos valores de k, comparándolos con métricas como Silhouette, Davies–Bouldin y Calinski–Harabasz. La mejor Silhouette aparece con complete/average y k=2 (≈0.314), pero al revisar el resultado vemos que no es práctico: crea un grupo enorme (733 Pokémon) y otro de 1 solo, que es Mew, un caso extremo porque tiene n_moves=235 y estadísticas muy "uniformes". Para obtener grupos más útiles definimos un modelo funcional centrado en las stats de combate y quitamos n_moves, y entonces elegimos ward con k=3, apoyándonos en el dendrograma y en que los tamaños quedan más equilibrados (313, 311 y 110). Por último comparamos los clusters con los tipos oficiales y vemos que los grupos mezclan tipos, lo que encaja con que el clustering está separando más bien roles por estadísticas. En el análisis final, Zapdos y Raichu caen en el mismo cluster (1), mientras ZapdosGalar cae en otro (2), indicando que sus perfiles funcionales no son iguales aunque algunos compartan rasgos o tipo.

1 Introducción

En el Ejercicio 1 el propósito es realizar un agrupamiento general del dataset y responder a tres cuestiones principales: (1) evaluar si el algoritmo encuentra una estructura clara de agrupaciones, (2) comprobar si los grupos obtenidos se corresponden con los tipos oficiales de Pokémon o si, por el contrario, mezclan tipos y reflejan similitud por perfil estadístico, y (3) analizar qué variables son las que más influyen en la separación entre grupos para poder describirlos de forma comprensible. Esta comparación con los tipos es relevante porque el tipo es una clasificación conocida, pero no necesariamente determina por sí solo que dos Pokémon tengan características de combate parecidas.

En el Ejercicio 2 aplicamos agrupamiento jerárquico aglomerativo al conjunto de datos de Pokémon con el objetivo de descubrir grupos de Pokémon "parecidos" sin usar etiquetas previas. La idea es construir una jerarquía: al principio cada Pokémon es su propio grupo y, poco a poco, el algoritmo va uniendo los más similares hasta formar un árbol. Esto es especialmente útil porque no solo obtenemos una partición final, sino también un dendrograma, que permite ver la estructura completa de uniones y decidir de forma razonada cuántos grupos tiene sentido considerar. A nivel práctico, nos interesa comprobar dos cosas. Primero, qué pasa cuando probamos distintos criterios de enlace (single, complete, average, ward) y varios números de clusters: queremos elegir una configuración que tenga sentido según métricas internas y según la interpretación visual del dendrograma. Segundo, queremos ver si los grupos encontrados se parecen a los tipos oficiales (Eléctrico, Agua, etc.) o no, ya que dos Pokémon de tipos distintos pueden cumplir un rol parecido si sus estadísticas son similares. Para cerrar, analizamos casos (Zapdos, ZapdosGalar y Raichu) y comprobamos en qué cluster cae cada uno y qué nos dice eso sobre su similitud funcional según las características usadas.

2 Ejercicio 1

2.1 Descripción del ejercicio y sus objetivos

El Ejercicio 1 tiene como objetivo aplicar técnicas de clustering al conjunto de datos de Pokémon para identificar agrupaciones sin usar etiquetas durante el entrenamiento. El objetivo no es únicamente obtener clústeres, sino evaluar si existe una estructura clara de grupos e interpretar qué representa cada grupo en términos de características de los Pokémon.

En este ejercicio se persiguen tres metas principales. En primer lugar, comprobar si a partir de las variables disponibles (principalmente estadísticas base y otras características derivadas) emergen clusters coherentes y bien definidos. En segundo lugar, analizar si esos grupos tienen relación con una clasificación conocida del dominio, como los tipos oficiales de Pokémon, o si el agrupamiento responde más bien a similitudes funcionales basadas en atributos de combate (estadísticas). Por último, se busca identificar qué características contribuyen más a la formación de los clústers.

Para lograrlo, tras estudiar y preparar los datos, se realizan experimentos con dos modelos de clustering con enfoques distintos: K-Means y DBSCAN. En ambos casos se prueban diferentes configuraciones de hiperparámetros (valores de *k* en K-Means y combinaciones de *eps* y

min_samples en DBSCAN) para observar cómo cambia la estructura de los clústeres obtenidos. Los resultados se comparan y evalúan con métricas de evaluación, principalmente el Silhouette Score, con el objetivo de seleccionar el modelo y la configuración que mejor cumplan los objetivos del ejercicio y, además, permitan responder de forma analítica si los grupos se relacionan con los tipos oficiales, qué variables separan con más fuerza a los clústeres y cómo describir y etiquetar distintos grupos de manera interpretable.

2.2 Descripción del conjunto de datos

Tras importar los datos y parsearlos a un dataframe, se utilizaron las funciones df.head(), df.info() y df.describe() para obtener una visión global inicial del conjunto de datos: estructura general, número de instancias y variables, tipos de dato, presencia de valores faltantes y estadísticas descriptivas de las variables numéricas. La función df.head() ofrece una visión general del dataframe generado, mostrando las primeras 5 instancias completas donde obtenemos una primera vista de la estructura del dataframe:

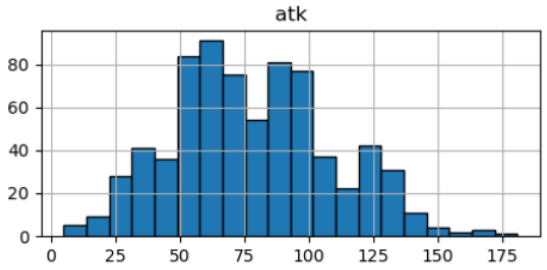
poke_id	hasEvo	hp	atk	def	spa	spd	spe	n_types	n_abilities	n_moves	types	abilities	learnset
abomasnow	0	90	92	75	92	85	60	2	2	57	[Grass, Ice]	[Snow Warning, Soundproof]	[lightscreen, avalanche, endure, powdersnow, r...
abra	1	25	20	15	105	55	90	1	3	57	[Psychic]	[Synchronize, Inner Focus, Magic Guard]	[lightscreen, wonderroom, allyswitch, sunnyday...
absol	0	65	130	60	75	60	75	1	3	73	[Dark]	[Pressure, Super Luck, Justified]	[sunnyday, endure, detect, assurance, darkpuls...
accelgor	0	80	70	40	100	60	145	1	3	49	[Bug]	[Hydration, Sticky Hold, Unburden]	[toxicspikes, endure, raindance, yawn, substit...
aegislash	0	60	50	140	50	140	60	2	1	45	[Steel, Ghost]	[Stance Change]	[sunnyday, autotomize, irondefense, endure, ra...

El dataset contiene 734 instancias (Pokémon) y 13 variables. De estas, 10 son numéricas (int64) y 3 son categóricas (object). Además, ninguna variable presenta valores faltantes, lo que permite trabajar sin necesidad de imputación. Las variables del dataframe son las siguientes:

- **hasEvo**: variable binaria (0/1) que indica si el Pokémon tiene evolución.
- **hp, atk, def, spa, spd, spe**: estadísticas base de combate.
- **types**: lista con el/los tipo(s) del Pokémon - variable categórica (object).
- **abilities**: lista con sus habilidades - variable categórica (object).
- **learnset**: lista de movimientos que puede aprender - variable categórica (object).
- **n_types, n_abilities, n_moves**: variables derivadas.

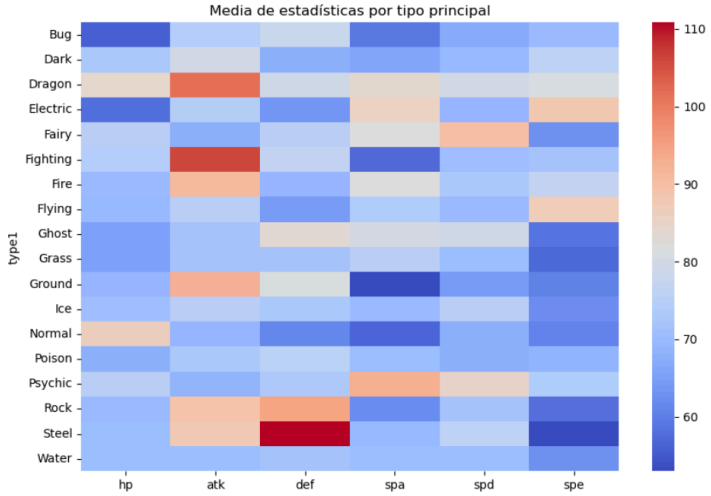
La función df.describe() permite visualizar estadísticas básicas de las variables numéricas como media, desviación estándar, cuartiles, máximo y mínimo. Se observan rangos amplios (hp va de 1 a 255, atk de 5 a 181), reflejando una alta variabilidad. En cuanto a las variables derivadas, n_types toma valores entre 1 y 2 (los Pokémons tienen uno o dos tipos), n_abilities entre 1 y 4 (los Pokémons poseen entre 1 y 4 habilidades) y n_moves que presenta una media de ≈ 53.17 y un máximo de 235. Sin embargo, no se aprecian errores sistemáticos en el conjunto de datos.

Se continuó con un estudio de distribuciones de las variables numéricas y sus correlaciones mediante histogramas y heatmaps/gráficos de caja, respectivamente. Como ejemplo, en la distribución de ataque (atk) se observa que la mayor parte de los Pokémon se concentran aproximadamente entre 50 y 110, con una media ≈ 78.54 y una mediana = 75, lo que sugiere una distribución moderadamente simétrica y concentrada en valores medios, con la presencia de valores atk más elevados (hacia la derecha).



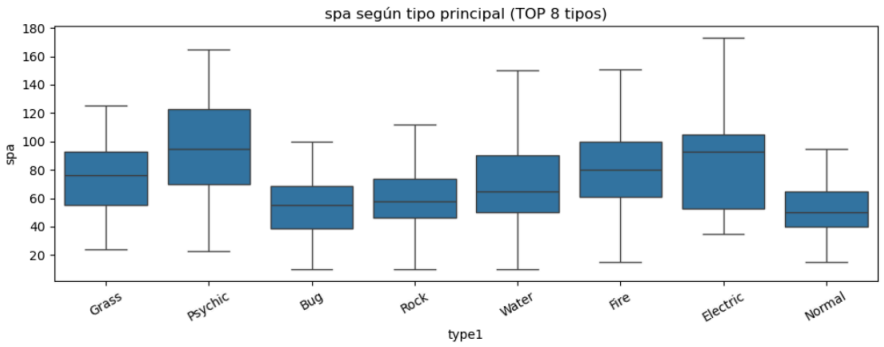
Aunque no se puedan incluir todas las distribuciones en este informe, se llevó a cabo el mismo proceso para todas las variables numéricas en el cuaderno.

Además, para explorar la relación entre estadísticas base (hp, atk, def, spa, spd, spe) y los tipos, se utilizó un heatmap de las medias de cada estadística base por tipo.



Fighting destaca por un ataque (atk) medio alto (≈110), mientras que el tipo Steel presenta una defensa (def) media elevada (≈110). Electric presenta valores relativamente altos en velocidad (spe) y Psychic valores altos en ataque especial (spa), lo que sugiere que el tipo puede estar asociado a ciertos perfiles estadísticos, aunque existe también solapamiento entre tipos.

También se utilizaron gráficos de caja para estudiar la relación entre el tipo principal y las estadísticas base, trabajando con los 8 tipos más representativos. En el ejemplo para ataque especial (spa), se observa que tipos como Psychic y Electric presentan valores centrales más altos (≈90) que Normal, Bug o Rock (≈60). En Electric y Psychic se aprecia una mayor amplitud mientras que en Normal o Bug la distribución es compacta y se concentra en valores bajos.



Este mismo procedimiento se aplicó en el cuaderno al resto de estadísticas base con un análisis correspondiente de cada una para comparar las estadísticas según el tipo.

2.3 Preparación de datos

En cuanto al tratamiento de valores faltantes, no fue necesario aplicar ninguna técnica de imputación, ya que el análisis inicial realizado confirma que ninguna variable presenta valores nulos. Todas las columnas cuentan con 734 valores no nulos. En cuanto al tratamiento de valores atípicos, aunque el dataset presenta algunos valores atípicos, no presentan indicios de errores sistemáticos o incoherencias en la estructura de los datos. Los valores más elevados corresponden a Pokémon con características especialmente destacadas. Dado que estos valores forman parte del contexto de Pokémon y del problema, además de que pueden aportar información relevante al proceso de clustering, no se aplicó ningún tratamiento específico.

El conjunto de datos incluye tres variables categóricas: types, abilities y learnset. En este ejercicio se decidió no codificarlas para incorporarlas al entrenamiento de los modelos de clustering. Aunque sería posible aplicar técnicas como one-hot encoding, estas variables son multivalor (un Pokémon puede tener varios tipos, habilidades o movimientos) y, en el caso de habilidades y movimientos, su codificación produciría un espacio de alta dimensionalidad. Esto dificulta definir una métrica de distancia que sea realmente interpretable y puede introducir ruido en algoritmos basados en distancia como K-Means o DBSCAN. La información de estas variables se resume mediante las variables derivadas sin aumentar la dimensionalidad.

Dado que los algoritmos de clustering utilizados en este ejercicio se basan en el cálculo de distancias, fue necesario aplicar un escalado (StandardScaler) de las variables numéricas para evitar que aquellas con rangos mayores dominaran el proceso de agrupamiento. Este escalado transforma cada variable para que tenga media 0 y desviación típica 1, asegurando que todas contribuyan de forma comparable al cálculo de distancias.

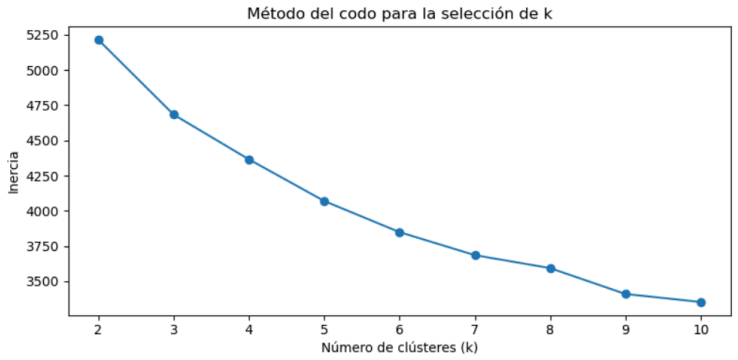
2.4 Experimentos

2.4.1 K-Means

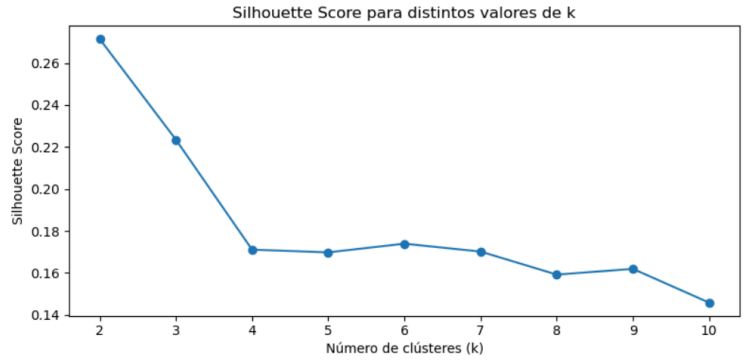
En esta sección se aplica K-Means con el objetivo de explorar cómo se agrupan los Pokémon a partir de su representación numérica. K-Means es un método de clustering que busca formar grupos minimizando la distancia entre los puntos y el centro (centroide) de su clúster.

K-Means es que requiere fijar previamente el número de clústeres, k, que indica cuántos grupos se desean obtener. Para llevar a cabo la implementación del modelo, se evaluaron distintos valores de k entre 2 y 10 (ambos incluidos). Este rango permite explorar desde agrupaciones muy generales (pocos clústeres) hasta particiones más detalladas, evitando a la vez valores excesivamente altos que dificultan la interpretación.

Como primer método de evaluación se utilizó el método del codo, basado en la inercia (suma de distancias cuadradas dentro de los clústeres). Esta métrica disminuye al aumentar k, pero el objetivo es identificar el punto a partir del cual añadir más clústeres aporta mejoras cada vez menores. En los resultados se observa una reducción pronunciada de la inercia para valores bajos, especialmente entre k = 2 y k = 4, y una mejora más suave a partir de ese rango. Esto sugiere un posible punto de inflexión alrededor de k = 4–5, aunque el método del codo por sí solo no determina un único valor óptimo. Los resultados se muestran en el gráfico abajo.



Se complementó el análisis de evaluación con el Silhouette Score, que evalúa la cohesión interna de cada clúster y la separación entre clústeres (valores más altos implican mejor estructura). En este caso, el valor máximo del Silhouette Score se obtiene para k = 2 (≈ 0.271), y a partir de ahí disminuye de forma notable. Se seleccionó k = 2 como configuración final al ofrecer el mejor equilibrio observado entre cohesión y separación. La evaluación mediante Silhouette Score se muestra en el gráfico abajo.



A partir de esta evaluación, con k = 2 se entrenó el modelo definitivo y se obtuvo una partición de dos clústeres con tamaños 313 y 421. Se analizaron los centroides (escala original), que representan la media de cada grupo. Los resultados se muestran en la siguiente tabla:

	hp	atk	def	spa	spd	spe	hasEvo	n_types	n_abilities	n_moves
0	52.495208	57.322684	56.281150	51.581470	52.501597	52.376997	0.955272	1.440895	2.527157	47.105431
1	83.954869	94.315914	89.365796	86.795724	87.543943	78.513064	0.019002	1.622328	2.237530	57.686461

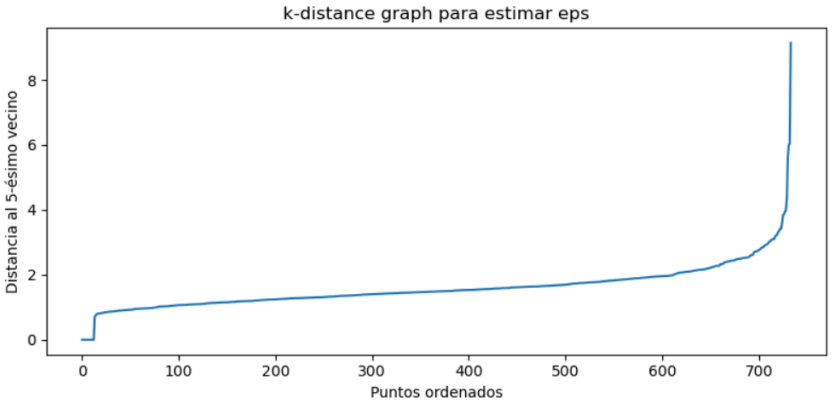
El clúster 0 presenta medias más bajas en todas las estadísticas base (por ejemplo, hp ≈ 52.5, atk ≈ 57.3, spe ≈ 52.4) y un valor medio alto de hasEvo ≈ 0.96, lo que sugiere un grupo formado mayoritariamente por Pokémon con evolución pendiente y perfil estadístico más modesto. En cambio, el clúster 1 muestra valores medios significativamente mayores en las estadísticas (hp ≈ 84.0, atk ≈ 94.3, def ≈ 89.4, spa ≈ 86.8, spe ≈ 78.5) y un valor medio muy bajo de hasEvo ≈ 0.02, indicando un grupo compuesto principalmente por Pokémon ya evolucionados.

El Silhouette Score (≈ 0.27) refleja una separación moderada. Existe estructura en los datos y diferencias claras entre perfiles medios, aunque con cierto solapamiento entre grupos.

2.4.2 DBSCAN

En esta sección se aplica DBSCAN, un algoritmo de clustering basado en densidad. A diferencia de K-Means, no requiere fijar el número de clústeres. DBSCAN identifica automáticamente regiones densas y forma clústeres a partir de ellas.

DBSCAN depende de dos hiperparámetros: eps y min_samples. Para seleccionar un rango de valores de eps, se utilizó el k-distance graph, calculando la distancia al 5º vecino más cercano (k = 5) para todos los puntos y ordenándolos de menor a mayor.



En la parte inicial de la curva se observa un crecimiento casi lineal (distancias relativamente pequeñas), lo que indica que la mayoría de instancias se encuentran en regiones de densidad similar. Hacia el final aparece un crecimiento exponencial, indicando puntos mucho más aislados. Este cambio de pendiente es el punto de referencia para estimar un eps razonable, ya que separa regiones densas de zonas dispersas. Se definió un rango de eps alrededor de esos valores para experimentar de forma controlada (1.8, 2.0, 2.2, 2.5 y 3.0).

Se entrenó el modelo a partir de estas configuraciones y se registraron para cada configuración el número de clústeres resultante, la cantidad de puntos clasificados como ruido y el Silhouette Score (cuando el modelo producía al menos dos clústeres) para evaluar el modelo.

	eps	min_samples	n_clusters	n_noise	silhouette
0	1.8	3	9	80	0.103596
1	1.8	5	6	106	0.117911
2	1.8	10	4	149	0.109593
3	2.0	3	9	61	0.110107
4	2.0	5	7	73	0.131979
5	2.0	10	4	100	0.132951
6	2.2	3	4	41	0.120733
7	2.2	5	3	51	0.119333
8	2.2	10	1	66	NaN
9	2.5	3	2	27	0.210873
10	2.5	5	2	28	0.210714
11	2.5	10	1	36	NaN
12	3.0	3	1	8	NaN
13	3.0	5	1	10	NaN
14	3.0	10	1	12	NaN

A partir de los resultados, la configuración más adecuada fue eps = 2.5, ya que produce una estructura con dos clústeres y una cantidad de ruido reducida. En ese valor, min_samples = 3 y min_samples = 5 ofrecen resultados muy similares, pero se seleccionó min_samples = 3 al presentar un Silhouette Score ligeramente superior (≈ 0.210873 frente a ≈ 0.210714).

2.4.1 Evaluación

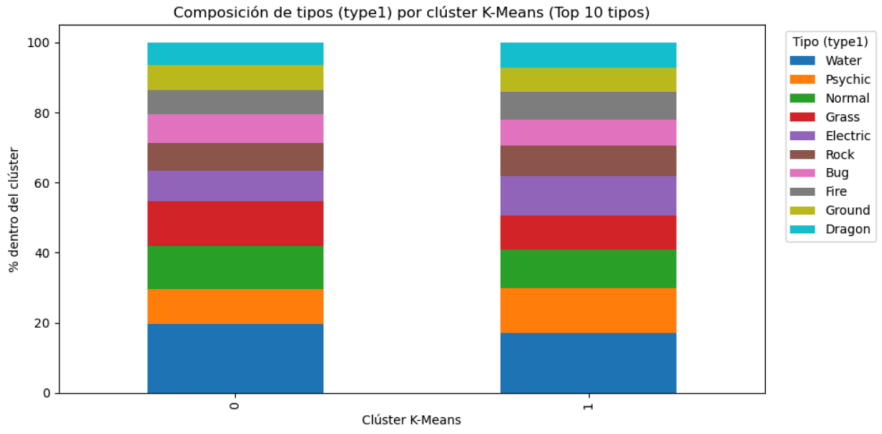
Para evaluar ambos enfoques se tomaron sus configuraciones de mejor rendimiento: K-Means con k = 2 y DBSCAN con eps = 2.5 y min_samples = 3. Ambos modelos producen dos clústeres, pero con diferencias. K-Means asigna todas las instancias a un clúster (313 y 421 Pokémon), mientras que DBSCAN también identifica 27 puntos como ruido. En términos de interpretabilidad, utilizando el modelo K-Means los clústeres pueden describirse fácilmente mediante sus centroides, lo que facilita comparar perfiles medios y entender qué variables diferencian los grupos. Los centroides muestran una separación clara entre un grupo con estadísticas base más bajas y un grupo con estadísticas más altas.

Desde el punto de vista cuantitativo, el Silhouette Score refuerza la idea de K-Means como el modelo más adecuado para este conjunto de datos: K-Means (k = 2) alcanza un valor ≈ 0.271, superior al de DBSCAN en su mejor configuración (≈ 0.211). Esto indica una mejor combinación de cohesión interna y separación entre clústeres en el caso de K-Means.

2.5 Preguntas

2.5.1 Alineación de los clústeres con los tipos oficiales

Para analizar si los clústeres obtenidos con el modelo seleccionado (K-Means, k = 2) se alinean con los tipos elementales oficiales, se estudió la distribución del tipo principal (type1, definido como el primer tipo de la lista types) dentro de cada clúster. Se representó gráficamente la composición porcentual de los 10 tipos más frecuentes en cada clúster:

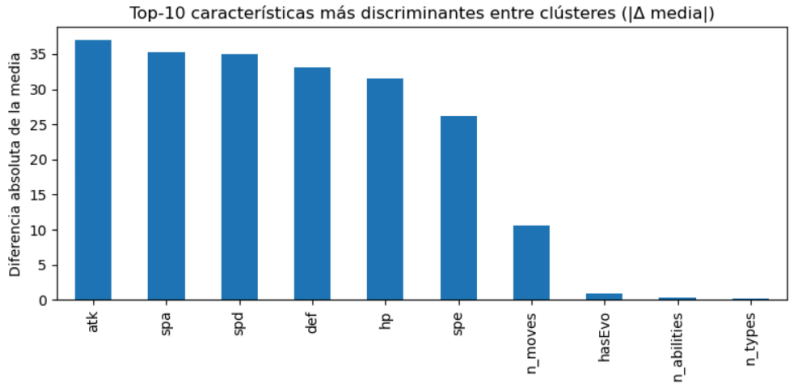


Los resultados muestran que ambos clústeres presentan distribuciones muy parecidas, sin una concentración marcada de tipos en un clúster concreto, las variaciones no son lo suficientemente grandes como para indicar una separación clara por el tipo elemental.

En conjunto, esta evidencia sugiere que los grupos generados por el algoritmo no se alinean principalmente con los tipos oficiales, y que el tipo no actúa como un predictor dominante del agrupamiento. Esto es coherente con el planteamiento del experimento, ya que los tipos no se incluyeron como variables de entrenamiento; los clústeres se formaron a partir de estadísticas base y variables derivadas. Aun así, las ligeras variaciones observadas podrían indicar una relación indirecta entre tipo y clúster (en la medida en que ciertos tipos tienden a compartir perfiles estadísticos), pero dicha relación no es lo bastante fuerte como para explicar por sí sola la estructura encontrada. En consecuencia, el agrupamiento parece responder sobre todo al perfil estadístico global de los Pokémon, más que a su clasificación elemental.

2.5.2 Características que más contribuyen a la formación de los clústeres

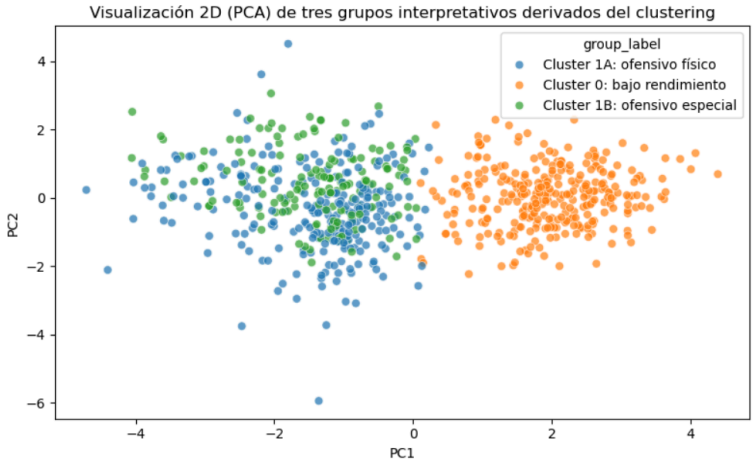
A continuación queremos entender qué variables dentro del conjunto de datos contribuyen más a la formación de los clústeres. Se calcularon las medias de cada característica por clúster y se representaron mediante un heatmap en escala original. Además, se calculó el ranking de diferencias absolutas de medias y se representó con un gráfico de barras, con el objetivo de identificar de forma directa qué características son más discriminantes. A continuación mostramos el gráfico de barras con las diferencias de las medias de cada clúster ya que la información del heatmap ya se presenta en la sección 2.4.1.



El resultado una vez más sugiere que la separación entre clústeres está dominada principalmente por las estadísticas base. Las variables con mayor capacidad discriminante son atk, spa, spd, def, hp y spe, con diferencias amplias de aproximadamente 26–37 puntos, muy por encima del resto de variables.

2.5.3 Descripción y etiquetado de tres grupos interpretables

El modelo seleccionado (K-Means con k = 2) produce dos clústeres con una separación marcada por las estadísticas base. Para obtener tres grupos, se definió un tercer grupo a partir del clúster de mayor rendimiento en función del estilo ofensivo, comparando atk y spa. De este modo se obtienen tres grupos, un grupo de bajo rendimiento y dos subgrupos dentro del clúster fuerte (ofensivo físico y ofensivo especial). Se utilizó PCA para visualizar estos grupos:



El Cluster 0: bajo rendimiento se caracteriza por medias más bajas en todas las estadísticas base (aproximadamente en el rango 50–60 para hp, atk, def, spa, spd y spe) y por un valor medio de hasEvo cercano a 1. El clúster de alto rendimiento se divide en dos particiones que comparten un nivel general de estadísticas elevado y un hasEvo cercano a 0, pero difieren en el tipo de características ofensivas. El Cluster 1A: ofensivo físico agrupa Pokémon en los que atk ≥ spa, mostrando un énfasis mayor en Ataque físico. El Cluster 1B: ofensivo especial agrupa aquellos con spa > atk, con un perfil más orientado a Ataque especial. Esta subdivisión permite distinguir estilos de combate diferentes con etiquetas semánticas claras.

3 Ejercicio 2

3.1 Descripción del ejercicio y sus objetivos

En este ejercicio aplicamos agrupamiento jerárquico aglomerativo para descubrir grupos de Pokémon "parecidos" sin usar etiquetas. Empezamos con cada Pokémon como un grupo y unimos los más similares hasta formar una jerarquía que podemos visualizar con un dendrograma. El objetivo es obtener clusters y también entender cómo se forman y decidir cuántos grupos tiene sentido cortar, apoyándonos en métricas (Silhouette, Davies–Bouldin y Calinski–Harabasz) y la interpretación del dendrograma. Además, el ejercicio pide comprobar si los grupos encontrados se parecen a los tipos oficiales (type1/type2) o si aparecen grupos funcionales (por ejemplo, atacantes rápidos, tanques defensivos, etc.) definidos más por estadísticas que por el tipo. Por último, analizamos casos concretos (Zapdos, ZapdosGalar y Raichu) para ver en qué cluster cae cada uno y qué nos dice eso sobre su similitud funcional.

3.2 Descripción del conjunto de datos

El dataset utilizado contiene 734 Pokémon y 13 variables, 3 variables categóricas (name, type1, type2) y 10 numéricas (hasEvo, n_types, n_abilities, n_moves y las estadísticas hp, atk, def, spa, spd, spe). No aparecen valores nulos en las columnas. En los descriptivos numéricos se observa que las stats tienen rangos amplios (por ejemplo hp llega a 255 y spe llega a 200), y destaca especialmente n_moves, con media ≈ 53.17 pero máximo 235. Esto es importante porque en clustering jerárquico, valores extremos pueden comportarse como outliers y condicionar mucho la separación de grupos.

3.3 Preparación de datos

Como el clustering jerárquico se basa en distancias, primero convertimos el dataset a una matriz numérica usable y hacemos que las variables sean comparables. En una primera preparación incluimos: estadísticas (hp, atk, def, spa, spd, spe), variables auxiliares (hasEvo, n_types, n_abilities, n_moves) y además codificamos type1 y type2 con one-hot encoding. Con esto se obtiene una matriz con 47 variables (stats + auxiliares + columnas binarias de tipos).

Aplicamos StandardScaler para escalar las variables. Si no escalamos, una variable con un rango grande puede dominar la distancia y "mandar" sobre el clustering. Los valores quedan en formato tipo z-score (positivos/negativos según estén por encima o debajo de la media), de modo que cada variable contribuye de manera más equilibrada.

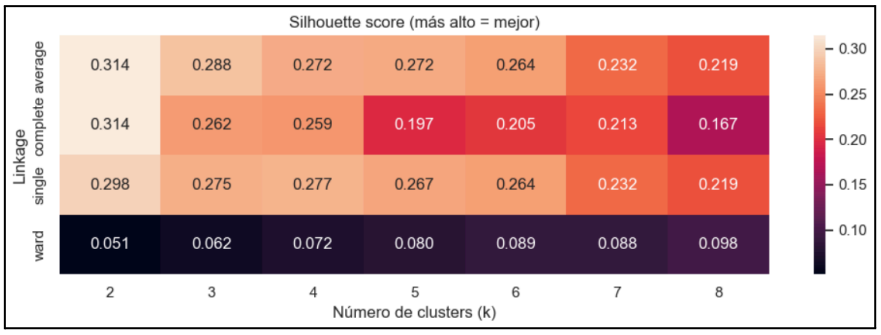
Más adelante, al ver que n_moves introduce un comportamiento extremo, se hace una segunda preparación orientada a "roles funcionales": se construye un modelo centrado en stats de combate y variables auxiliares suaves, eliminando n_moves.

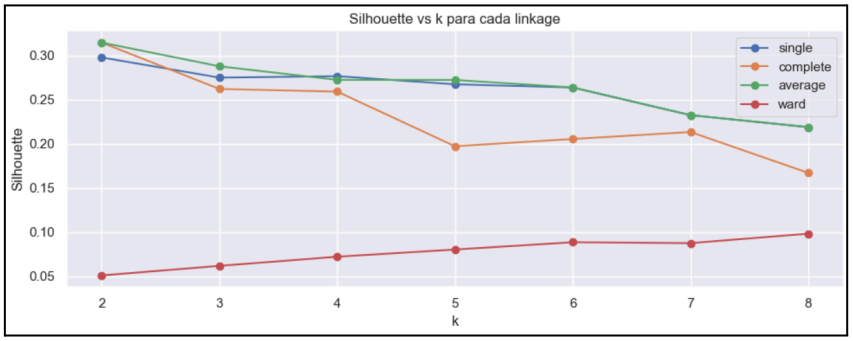
3.4 Experimentos

3.4.1 Definición del experimento

El experimento principal consiste en probar agrupamiento jerárquico aglomerativo variando dos cosas. El criterio de enlace (linkage) y 2) el número de clusters k. Se han evaluado los linkages: single, complete, average y ward, y se ha probado un rango de k de 2 a 8. Para comparar configuraciones se han usado tres métricas internas:

- Silhouette (más alto = mejor)
- Davies–Bouldin (más bajo = mejor)
- Calinski–Harabasz (más alto = mejor)



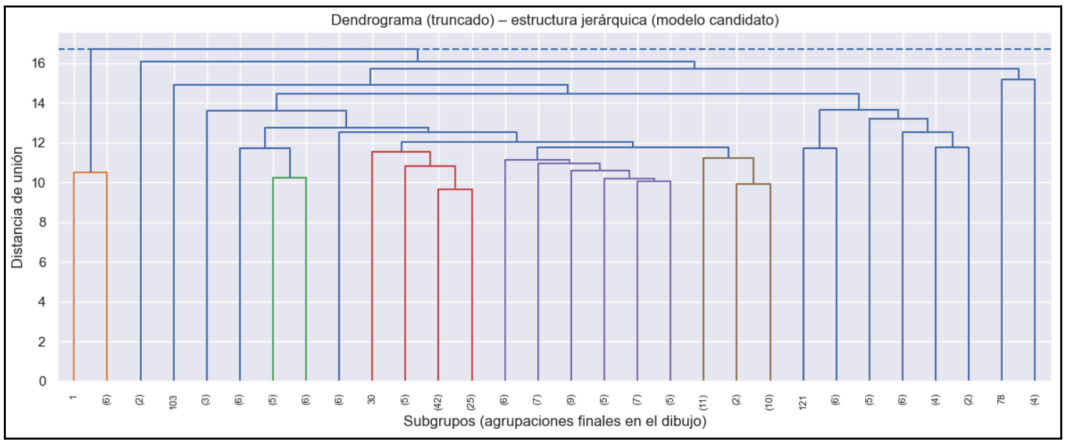


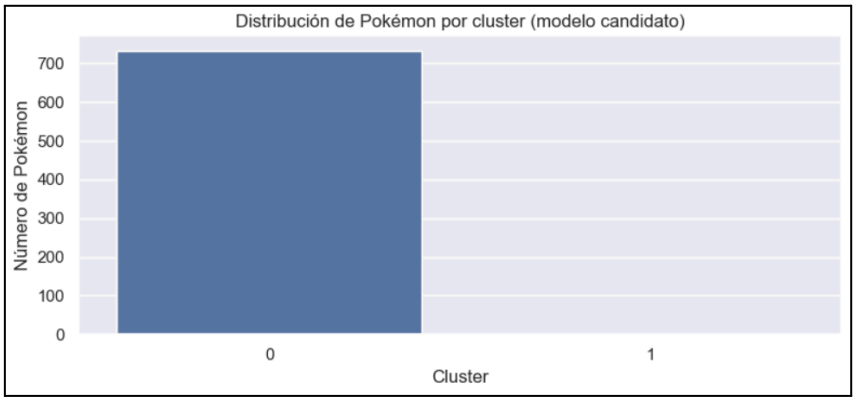
Estas dos figuras se usan porque permiten ver rápidamente qué combinaciones (linkage, k) rinden mejor y cómo cambia la calidad al aumentar el número de grupos.

3.4.2 Evaluación del experimento

Con la primera representación (stats + auxiliares + tipos one-hot, incluyendo n_moves), el mejor resultado por métricas aparece con complete y k=2, con Silhouette ≈ 0.314469 y Davies–Bouldin ≈ 0.548707 (empata con average en Silhouette). Esto se aprecia claramente en el heatmap y las curvas: el valor máximo está en k=2 para complete/average.

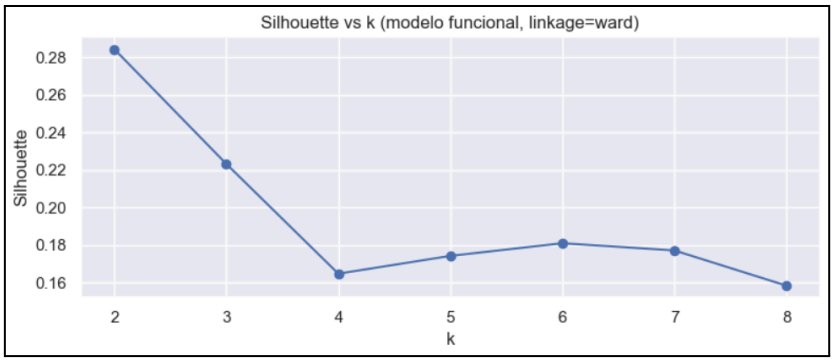
Sin embargo, al mirar la distribución real de clusters, aparece un problema importante: con k=2 el modelo crea un cluster con 733 Pokémon y otro con 1 Pokémon. Es decir, la partición existe, pero no es útil para analizar "varios grupos" porque básicamente está separando un outlier.

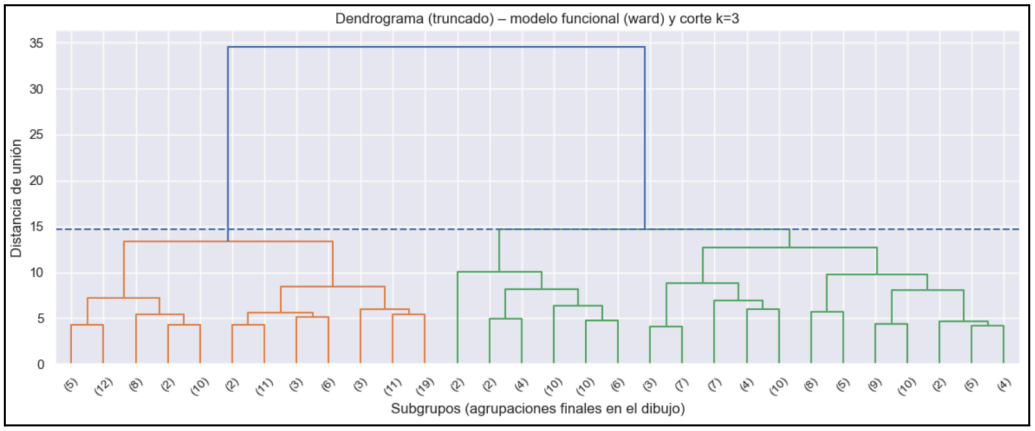


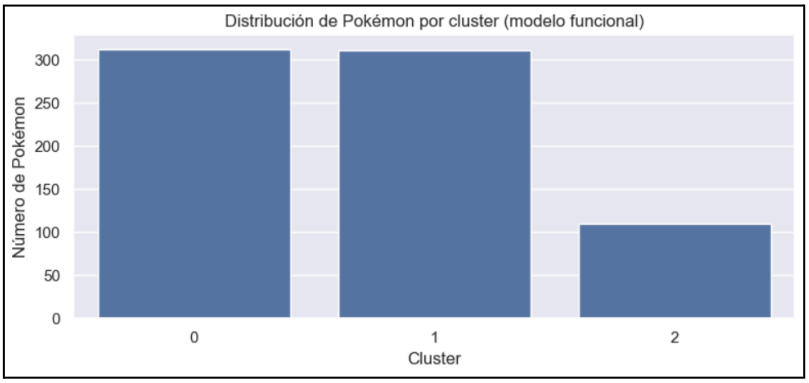


La identificación del outlier confirma lo anterior: el Pokémon aislado es Mew, con n_moves = 235 y stats perfectamente equilibradas (100 en todo), lo que lo hace extremadamente diferente del resto en la dimensión de movimientos. Esto explica por qué el dendrograma del modelo candidato muestra una separación fuerte y por qué el corte en dos clusters termina siendo "resto del mundo vs Mew".

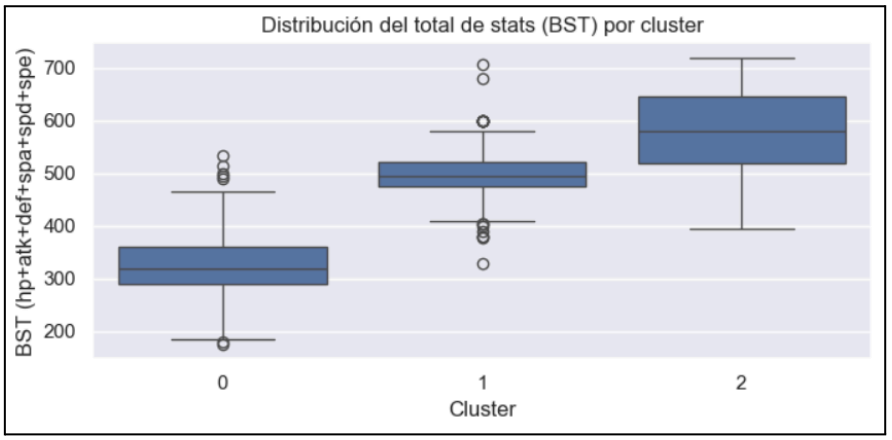
Para obtener grupos interpretables, se plantea un modelo funcional eliminando n_moves y centrándonos en stats de combate y variables auxiliares más estables. Con este enfoque se prueba ward y se analiza Silhouette para k=2..8. Aunque k=2 da el mayor Silhouette (≈ 0.2842), se elige k=3 porque el dendrograma permite un corte claro en tres ramas y porque el reparto final es equilibrado y fácil de interpretar.







En este modelo funcional, la distribución final es: 313 Pokémon en el cluster 0, 311 en el cluster 1 y 110 en el cluster 2, lo cual ya permite comparar perfiles y sacar conclusiones. Finalmente, para interpretar los clusters como "niveles funcionales", se analiza el BST (suma de hp+atk+def+spa+spd+spe) con un boxplot. Este gráfico es útil porque muestra diferencias claras de nivel global entre clusters y también la dispersión interna.



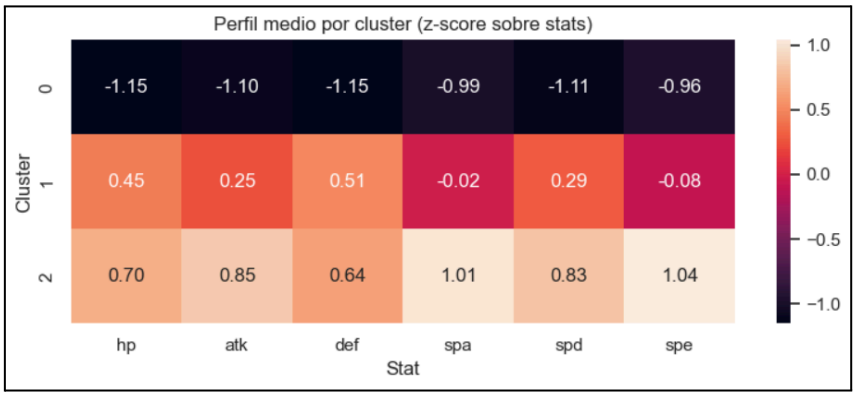
3.5 Preguntas

Distribución de Pokémon en los grupos:
Con el modelo candidato (complete, k=2) la distribución es extremadamente desbalanceada: 733 vs 1. Esto se ve claramente y se explica porque el cluster de tamaño 1 corresponde a Mew, un outlier dominado por n_moves=235. Por tanto, aunque ese modelo optimiza Silhouette, no sirve para un análisis de grupos amplio. Con el modelo funcional (ward, k=3) la distribución es mucho más útil y equilibrada: 313 (cluster 0), 311 (cluster 1) y 110 (cluster 2). Esto se observa en la, donde ya se pueden comparar tamaños sin que haya clusters "vacíos" o de un solo elemento.

Número óptimo de clusters según dendrograma y justificación:
Si solo mirásemos las métricas del primer experimento, el "mejor" sería k=2 (Silhouette ≈ 0.314 con complete/average), pero el dendrograma y la distribución revelan que ese corte está separando básicamente un outlier. En cambio, en el modelo funcional el dendrograma muestra un corte razonable en k=3, y el reparto 313/311/110 respalda que los grupos son consistentes. Además, el boxplot de BST refuerza que con k=3 aparecen perfiles globales distintos (un cluster claramente más alto en BST, uno intermedio y uno más bajo).

¿Aparecen grupos funcionales distintos a los tipos oficiales?
Sí. Al cruzar clusters con type1 se ve que ningún cluster es "un tipo puro". En el heatmap de proporciones se aprecia que los clusters mezclan tipos, lo que indica que el agrupamiento está capturando más bien roles por estadísticas y no las categorías oficiales. Aun así, se observan concentraciones: por ejemplo, el cluster 2 (el más pequeño) tiene una presencia relativamente alta de Psychic y Dragon (en recuento: Psychic=16 y Dragon=13 dentro de ese cluster), lo cual cuadra con que ese cluster tiende a agrupar perfiles más potentes (también visible por el BST).

Análisis de Zapdos, ZapdosGalar y Raichu (clusters y similitud funcional). En el modelo funcional, Zapdos cae en el cluster 1, Raichu también cae en el cluster 1, y ZapdosGalar cae en el cluster 2. Esto sugiere que Zapdos y Raichu comparten un perfil funcional más parecido entre sí (aunque no tengan el mismo rol exacto), mientras que ZapdosGalar se agrupa con un conjunto diferente, coherente con su perfil más orientado a ataque físico (atk=125) y con un "nivel" global más alto en BST.



Se ve además cómo cada uno se compara con la media de su cluster: Zapdos destaca en spa=125, Raichu destaca en spe=110 (más rápido que la media de su grupo), y ZapdosGalar se separa en atk frente a un perfil medio distinto. En conjunto, esto apoya la idea de que el clustering está separando por perfil de estadísticas (función/rol) más que por el tipo.

3 Conclusiones

3.1 Conclusiones generales

En este informe se ha comprobado que el clustering y el clustering jerárquico aglomerativo permite identificar patrones útiles en el conjunto de datos de Pokémon sin necesidad de etiquetas. La comparación entre enfoques mostró que K-Means ofrece una solución más estable e interpretable que DBSCAN, apoyada además por un mejor rendimiento según métricas internas (Silhouette). Los resultados indican que la estructura descubierta está dominada por el perfil estadístico global y por rasgos estructurales como hasEvo, más que por una separación directa por tipos elementales. Además de asignar grupos, permite visualizar la estructura completa con un dendrograma. Esto ayuda a justificar el número de clusters de forma más razonada, ya que se pueden identificar saltos grandes en la distancia de unión. El ejercicio deja claro que en clustering no basta con mirar una métrica: hay que comprobar si la solución es interpretable y si los grupos que salen tienen sentido. El método funciona como herramienta de análisis exploratorio, pero es sensible a la representación de datos y outliers.

3.2 Conclusiones específicas sobre los ejercicios

En el ejercicio 1 se compararon K-Means y DBSCAN mediante diferentes configuraciones de hiperparámetros y métricas de evaluación como Silhouette Score. K-Means mostró su mejor rendimiento con k=2 (Silhouette≈0.271), mientras que DBSCAN alcanzó su mejor configuración con eps = 2.5 y min_samples = 3 (Silhouette≈0.211) e identificó 27 puntos de ruido. En conjunto, K-Means ofreció una estructura más estable y una interpretación más directa (centroides), por lo que se seleccionó como modelo final. La interpretación del mejor modelo indica que los clústeres se explican principalmente por el perfil estadístico global. Un grupo con estadísticas medias más bajas frente a otro con valores significativamente más altos, sugiriendo una separación casi total entre Pokémon con evolución pendiente y sin evolución. Al contrastar los clústeres con los tipos principales no se observó una alineación fuerte.

En el Ejercicio 2, al probar distintos linkages y valores de k, el mejor resultado inicial por Silhouette se obtuvo con complete/average y k=2 (≈0.314). Sin embargo, al revisar la distribución se vio que esa solución era poco útil porque generaba un reparto 733 vs 1, aislando a Mew como outlier (n_moves=235 y stats muy homogéneas). Esto muestra como un valor alto de Silhouette puede corresponder a una separación “real”, pero no necesariamente a una segmentación que describe varios grupos. Para obtener clusters más interpretables se adoptó un modelo funcional con ward, eliminando n_moves y centrando la distancia en stats de combate y variables auxiliares. Con este enfoque se eligió k=3, apoyado por el dendrograma del modelo funcional y por una distribución equilibrada de tamaños. Las figuras de análisis refuerzan la interpretación: el boxplot de BST sugiere tres niveles claros de potencia global, y el heatmap de perfil medio deja un cluster con stats globalmente por debajo de la media (cluster 0), uno intermedio (cluster 1) y otro claramente superior en casi todas las stats (cluster 2).

En la comparación con tipos oficiales , los clusters no se corresponden con un único tipo: se mezclan tipos dentro de cada grupo, lo que apoya la idea de que el clustering está capturando roles funcionales basados en stats, más que categorías oficiales. Por último, el caso de estudio refuerza esta lectura. Zapdos y Raichu quedan en el mismo cluster (1), mientras ZapdosGalar cae en otro (2). También se muestra que Zapdos destaca por su spa=125, Raichu por su spe=110, y ZapdosGalar por su atk=125, lo que explica por qué el algoritmo los separa.

Bibliografía

1. G. James, D. Witten, T. Hastie y R. Tibshirani, An Introduction to Statistical Learning with Applications in R (2.ª edición), Springer, 2021.

2. A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd Edition), O'Reilly Media, 2019.

3. W. McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and IPython (2nd Edition), O'Reilly Media, 2017.

4. J. VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media, 2016.

5. Canvas de Aprendizaje Estadístico y Data Mining, Apuntes de la asignatura, Universidad Francisco de Vitoria, plataforma Canvas, 2025.