

# Laboratorio 7: ¡Atrapa el mejor modelo!

## Ensamblés de Votación con PokeAPI

Duración: 2 horas    Formato: Competencia por equipos con rotaciones y kahoots Pokemon

**Objetivo general:** Desarrollar un flujo completo de aprendizaje automático con datos provenientes de una API pública, construyendo un dataset desde cero, entrenando modelos base y creando un **ensamble de votación** sin librerías externas, integrando teoría, implementación y trabajo colaborativo bajo rotaciones estructuradas.

## Estructura y dinámica de trabajo

- Los equipos estarán conformados por **4 integrantes**.
- Cada equipo trabajará con **2 computadoras fijas**:
  - **Computadora A (Infraestructura)**: encargada del consumo de la PokéAPI, la construcción del dataset de 500 Pokémon y la creación del repositorio público en GitHub con su respectivo README.
  - **Computadora B (Modelado)**: dedicada al entrenamiento de los clasificadores base y a la implementación de los tres tipos de votadores (dura, suave y ponderada) sin usar herramientas preconstruidas de ensamble.
- Los **integrantes rotarán de lugar** a lo largo de la práctica; las computadoras permanecen fijas.
- En cada rotación se realizará un **kahoot temático de Pokémon**, que otorgará puntos extra durante la competencia.
- El **cuestionario teórico** sobre ensambles de votación estará disponible durante toda la práctica y deberá ser respondido dentro del cuaderno de trabajo.
- Los **modelos base no pueden ser ensambles potenciadores** (por ejemplo, no se permitirá el uso de AdaBoost, Gradient Boosting, XGBoost, LightGBM ni Random Forest). Los clasificadores deberán ser modelos individuales como regresión logística, árbol de decisión, KNN, SVM, entre otros.
- El **desempeño final** se evaluará según la exactitud (*accuracy*) del ensamble sobre un conjunto de prueba (**test set**) proporcionado por el profesor.
- Todo el material necesario para el laboratorio (instrucciones, plantilla del cuaderno y recursos auxiliares) estará disponible en un **repositorio público de solo lectura**, accesible en:

<https://github.com/JAlbertoAlonso/poke-ensambles>

# Rotaciones y cronograma general

Tiempo	Actividad	Participantes
0:00–0:10	Bienvenida, explicación de la dinámica y formación de equipos.	Todos
0:10–0:35	Primera rotación: trabajo inicial en infraestructura y modelado.	2 por computadora
0:35–0:40	Kahoot 1 (dinámica Pokémon).	Todos
0:40–1:05	Segunda rotación: intercambio de lugares entre integrantes.	2 por computadora
1:05–1:10	Kahoot 2 (dinámica Pokémon).	Todos
1:10–1:35	Tercera rotación: nueva distribución de integrantes.	2 por computadora
1:35–1:40	Kahoot 3 (dinámica Pokémon).	Todos
1:40–1:55	Cuarta rotación y cierre de tareas principales.	2 por computadora
1:55–2:00	Kahoot final y registro de entregas.	Todos

## Elementos integradores de la práctica

### 1. Construcción del dataset (Infraestructura)

- Cada equipo deberá consumir la **PokéAPI** para recopilar información de al menos **500 Pokémon**.
- La API a utilizar es la **PokéAPI**, disponible públicamente en:

<https://pokeapi.co/api/v2/pokemon/>

Cada Pokémon puede consultarse mediante su identificador numérico, por ejemplo:

<https://pokeapi.co/api/v2/pokemon/1>

devuelve la información correspondiente a Bulbasaur. Los equipos deberán automatizar las solicitudes para recolectar al menos 500 registros distintos y construir su propio dataset tabular.

- Las variables utilizadas como características (*features*) del modelo serán:

`height, weight, base_experience, type_main.`

- Las variables utilizadas para construir la etiqueta (no visibles para el modelo) serán:

`hp, attack, defense, special_attack, special_defense, speed.`

- La etiqueta binaria se definirá según la fuerza total del Pokémon:

$power\_score = hp + attack + defense + special\_attack + special\_defense + speed$

$$strong = \begin{cases} 1, & \text{si } power\_score > P_{75}(power\_score) \\ 0, & \text{en otro caso.} \end{cases}$$

- El dataset final deberá guardarse en formato `.csv` y subirse al repositorio público del equipo.
- El archivo `.csv` entregable deberá contener, como mínimo, las siguientes columnas:

1. `name`
2. `height`
3. `weight`
4. `base_experience`
5. `type_main`
6. `hp`
7. `attack`
8. `defense`
9. `special_attack`
10. `special_defense`
11. `speed`
12. `power_score`
13. `strong`

Las primeras once columnas corresponden a las variables extraídas desde la API, mientras que las dos últimas (`power_score` y `strong`) deben generarse a partir del cálculo y umbral definidos en esta sección. Cada fila representará un Pokémon único.

## 2. Entrenamiento de modelos base (Modelado)

- Se deberán entrenar al menos tres clasificadores distintos utilizando `scikit-learn`.
- Los modelos base no pueden ser ensambles potenciadores ni métodos de bosque aleatorio.
- Cada modelo debe ser entrenado, evaluado y documentado dentro del cuaderno de trabajo.

## 3. Ensambls de votación: Explicación teórica

Los métodos de ensamble de votación combinan las predicciones de varios clasificadores base para obtener una decisión final más robusta. Sea un conjunto de  $M$  modelos  $\{h_1, h_2, \dots, h_M\}$  que predicen una etiqueta binaria  $y \in \{0, 1\}$  a partir de una entrada  $x$ .

- **Votación dura (majority voting):** cada clasificador emite una votación  $h_j(x) \in \{0, 1\}$ , y la clase final se determina por mayoría:

$$\hat{y} = \begin{cases} 1, & \text{si } \sum_{j=1}^M h_j(x) > \frac{M}{2} \\ 0, & \text{en otro caso.} \end{cases}$$

Este método asume que todos los clasificadores tienen el mismo peso y que los errores individuales son independientes.

- **Votación suave (soft voting):** en lugar de usar decisiones binarias, cada modelo produce una probabilidad  $p_j(y = 1|x)$ . La predicción final se obtiene al promediar las probabilidades:

$$\hat{y} = \begin{cases} 1, & \text{si } \frac{1}{M} \sum_{j=1}^M p_j(y = 1|x) \geq 0.5 \\ 0, & \text{en otro caso.} \end{cases}$$

Este método suele ser más estable porque aprovecha la confianza de cada modelo.

- **Votación ponderada (weighted voting):** asigna un peso  $w_j$  a cada clasificador en función de su desempeño (por ejemplo, su exactitud en validación):

$$\hat{y} = \begin{cases} 1, & \text{si } \frac{\sum_{j=1}^M w_j p_j(y=1|x)}{\sum_{j=1}^M w_j} \geq 0.5 \\ 0, & \text{en otro caso.} \end{cases}$$

Los pesos permiten dar mayor influencia a los modelos más confiables.

El principio fundamental es que, si los clasificadores cometen errores de manera no correlacionada, la combinación por votación tiende a reducir la varianza del sistema y mejorar la capacidad de generalización.

## 4. Evaluación y competencia

- Cada equipo recibirá un conjunto de prueba (**test set**) proporcionado por el profesor, el cual deberá ser utilizado exclusivamente para evaluar el desempeño de su ensamble final.
- El desempeño se medirá mediante la **exactitud (accuracy)** del ensamble final. Esta métrica representa la proporción de predicciones correctas sobre el total de ejemplos del conjunto de prueba:

$$\text{Accuracy} = \frac{\text{Número de predicciones correctas}}{\text{Número total de ejemplos}}$$

Cuanto mayor sea la exactitud, mejor será la posición del equipo en el ranking de la competencia.

- Se elaborará un **ranking de equipos** según la exactitud alcanzada por cada ensamble. Los resultados serán presentados al finalizar la sesión de laboratorio.
- Cada equipo deberá entregar también un archivo de **predicciones finales** en formato **.csv**, correspondiente a su votador final evaluado sobre el conjunto de prueba proporcionado.

El archivo deberá tener el siguiente formato estandarizado:

```
id, y_pred, team_name, model
```

donde:

- **id**: identificador numérico o índice del Pokémon dentro del conjunto de prueba.
- **y\_pred**: predicción binaria del ensamble (1 para Pokémon fuerte, 0 para Pokémon débil).
- **team\_name**: nombre oficial del equipo (solo completado en la primera fila).

- `model`: nombre o tipo de votador utilizado (solo completado en la primera fila, por ejemplo: `VotacionPonderada`, `VotacionSuave`, o `VotacionDura`).

Ejemplo de formato válido:

```
id,y_pred,EquipoBulbasaur,VotacionPonderada
0,1,,
1,0,,
2,1,,
3,1,,
4,0,,
...
```

Este archivo será utilizado por el profesor para calcular la métrica de exactitud (*accuracy*) y determinar la posición de cada equipo en el ranking final. El incumplimiento del formato implicará la exclusión automática del ranking de competencia.

- La calificación de competencia es independiente de la evaluación académica del laboratorio, pero ambas influyen en el reconocimiento final del desempeño del equipo.

## 5. Cuestionario teórico (en el cuaderno)

Durante toda la práctica, los equipos deberán responder dentro del cuaderno las siguientes preguntas teóricas relacionadas con los ensambles de votación.

1. ¿Qué ventajas ofrece combinar varios clasificadores mediante un sistema de votación?
2. Explica la diferencia conceptual entre votación dura y votación suave.
3. ¿Qué tipo de problemas puede presentar una votación por mayoría simple?
4. ¿En qué casos un ensamble puede tener peor desempeño que un modelo individual?
5. ¿Por qué es importante la diversidad entre los modelos base en un ensamble?
6. Define el principio de independencia de alternativas irrelevantes y su relación con los sistemas de votación.
7. ¿Qué diferencia existe entre una votación ponderada y una votación suave?
8. ¿Cómo afecta el sesgo y la varianza de los modelos base al resultado del ensamble?
9. ¿Qué implicaciones tiene combinar modelos que tienden a cometer los mismos errores?
10. Explica por qué los métodos de ensamble no contradicen el teorema del *no free lunch*.

## Entregables

- Enlace al repositorio público en GitHub que contenga:
  1. Dataset de 500 Pokémon obtenido desde la PokéAPI.
  2. Notebook con el desarrollo completo (consumo de API, modelos, ensambles, respuestas teóricas y resultados).

3. Archivo de predicciones finales sobre el test set.
  4. README claro, estructurado y con formato profesional.
- El profesor proporcionará un **notebook plantilla** para unificar el formato de entrega.

## Evaluación

### Rúbrica académica (100 pts)

Criterio	Ponderación
Consumo correcto de la API y construcción del dataset de 500 Pokémon	20%
Entrenamiento y documentación de los modelos base	20%
Construcción de los tres ensambles (dura, suave, ponderada)	35%
Cuestionario teórico (10 preguntas)	15%
Claridad del README y presentación general del repositorio	10%
<b>Total</b>	<b>100%</b>

### Competencia (Ranking por exactitud del ensamble)

Lugar	1°	2°	3°	4°	5° o menor	Bonus
<b>Puntaje</b>	100	80	60	40	20	+20 mejor README, +30 mejor cuaderno

La puntuación de competencia es independiente de la calificación académica, pero será anunciada en tiempo real durante la sesión del laboratorio.

Al finalizar el laboratorio, los equipos deberán entregar su enlace de GitHub con todos los elementos solicitados. El desempeño del ensamble en el conjunto de prueba determinará la posición final en el ranking.