

Predicción de Combate Pokémon: Análisis y Desarrollo de un Modelo Predictivo

Tecnologías utilizadas

- Python (Pandas, Seaborn, Matplotlib)
- Scikit-learn (RandomForestClassifier, métricas)
- Joblib (serialización de modelos)
- Flask (backend web)

Introducción

El presente proyecto tiene como objetivo desarrollar un sistema que permita predecir el ganador en un combate entre dos Pokémon. Para ello, se trabaja con datos históricos de combates, así como con las características y estadísticas individuales de cada Pokémon. La predicción se realiza mediante un modelo computacional entrenado con información real, y se presenta a través de una interfaz web interactiva que muestra las estadísticas de los Pokémon y el resultado de la predicción.

Descripción de los Datos

Se utilizan dos conjuntos principales de datos en formato CSV:

1. **pokemon.csv**: Contiene información básica y estadísticas de cada Pokémon. Sus campos principales son:
 - #: Número o ID del Pokémon
 - Name: Nombre del Pokémon
 - Type 1: Tipo primario (por ejemplo, fuego, agua)
 - Type 2: Tipo secundario (puede estar vacío)
 - HP, Attack, Defense, Sp. Atk, Sp. Def, Speed: Estadísticas de combate
 - Generation: Generación a la que pertenece
 - Legendary: Indica si es un Pokémon legendario (verdadero o falso)
2. **combats.csv**: Registra enfrentamientos entre dos Pokémon, identificando el ganador. Sus campos son:
 - First_pokemon: Nombre del primer Pokémon participante
 - Second_pokemon: Nombre del segundo Pokémon participante
 - Winner: Nombre del Pokémon ganador del combate

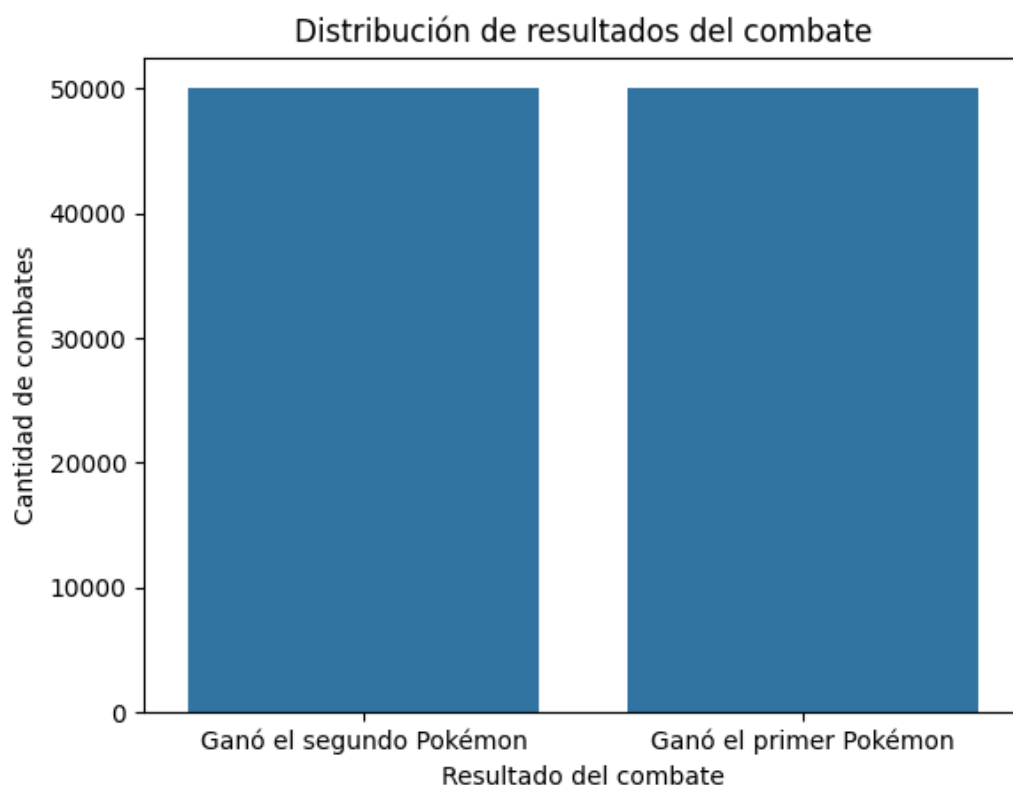
Metodología

Obtención y Limpieza de los Datos

Este dataset fue obtenido de [Kaggle](#). Dado que la cantidad de datos no es muy grande, se realizó una revisión manual en Excel. Solo se detectó un dato faltante: el nombre de un Pokémon tipo lucha de primera generación. Mediante descarte y comparación con los nombres existentes, se identificó e incorporó correctamente el Pokémon faltante.

Generación de Datos Simétricos

Los archivos CSV se cargan y procesan con Pandas para facilitar el análisis y el entrenamiento del modelo. Se utilizó el script *InvertirDataSet.py* para generar un archivo llamado *combats_simetrico.csv*, que contiene las mismas batallas del archivo original pero con los Pokémon invertidos (es decir, *First_pokemon* y *Second_pokemon* intercambiados). Esto permite al modelo aprender de manera simétrica y evitar sesgos por el orden en que se presentan los combatientes.



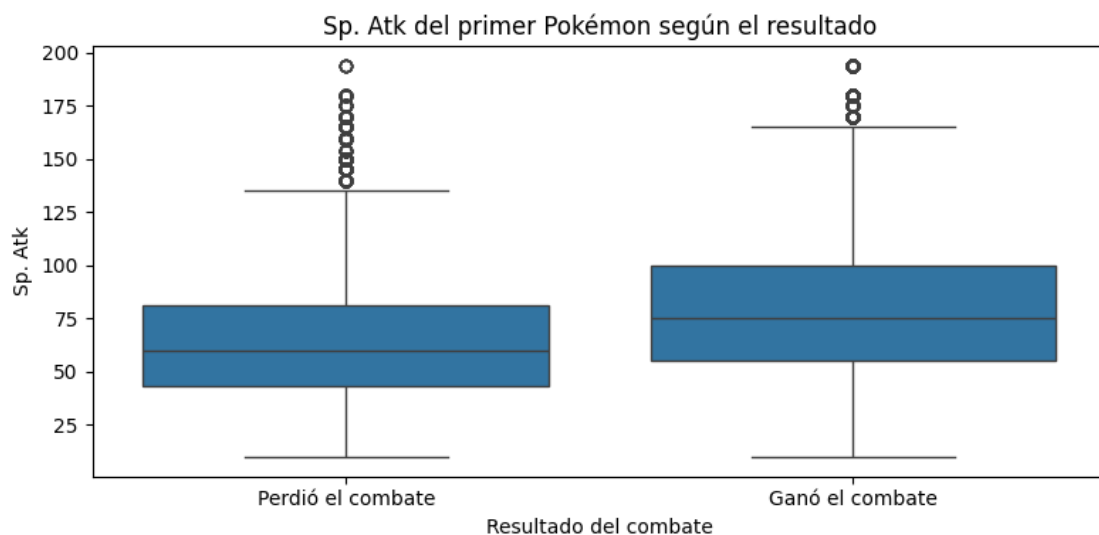
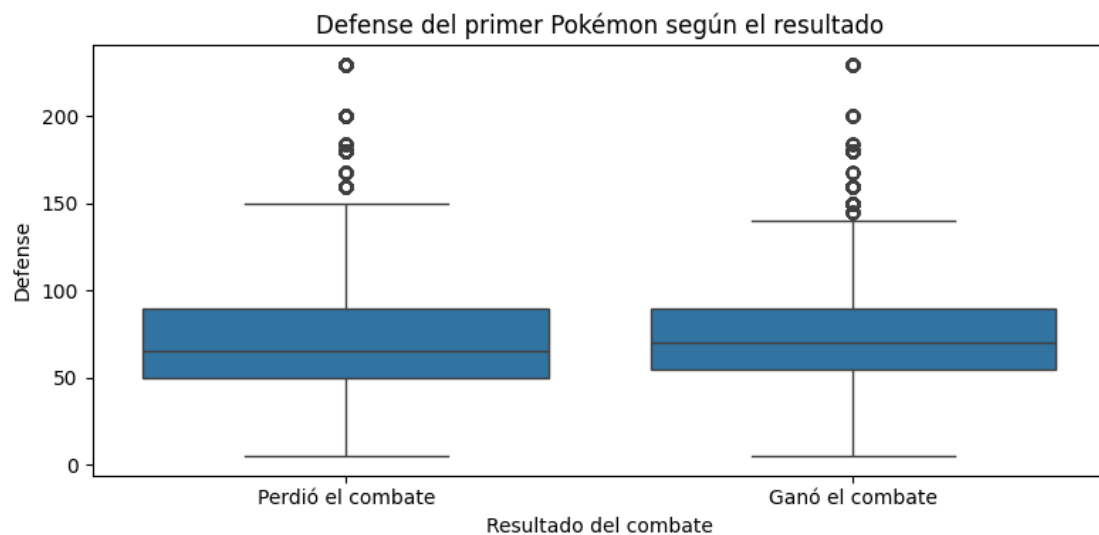
Visualización de Datos y Modelo Predictivo

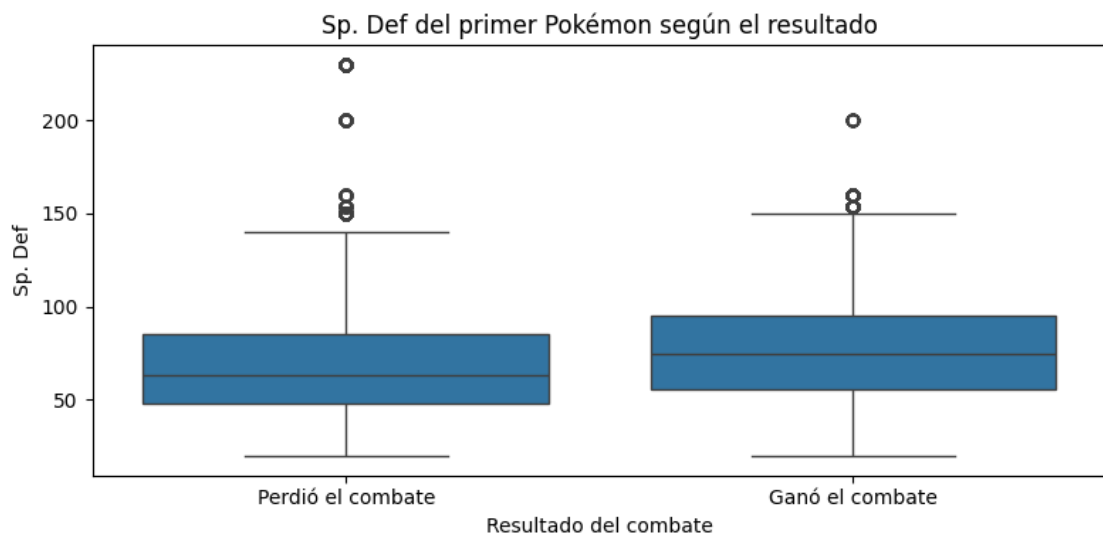
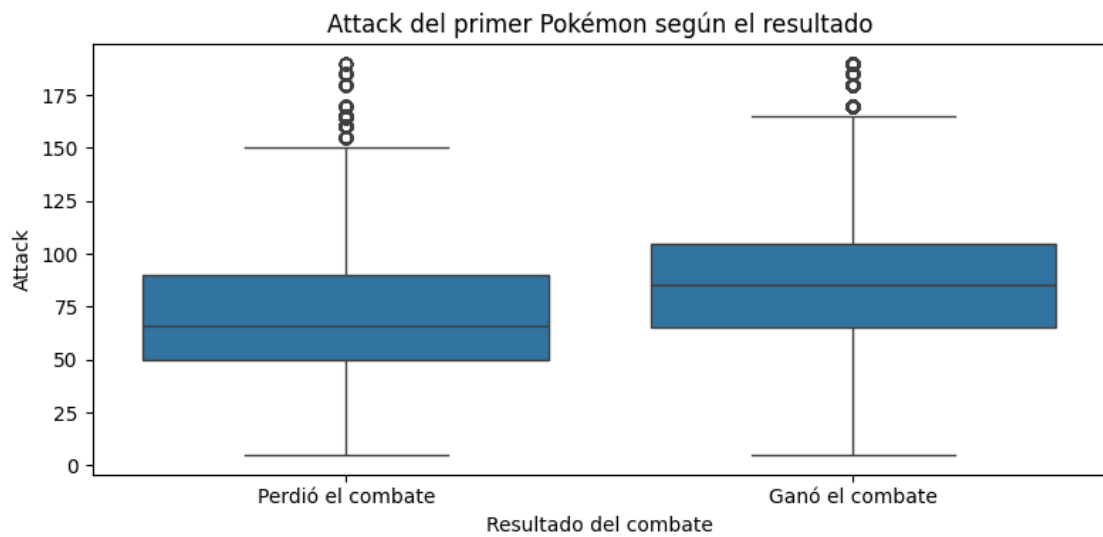
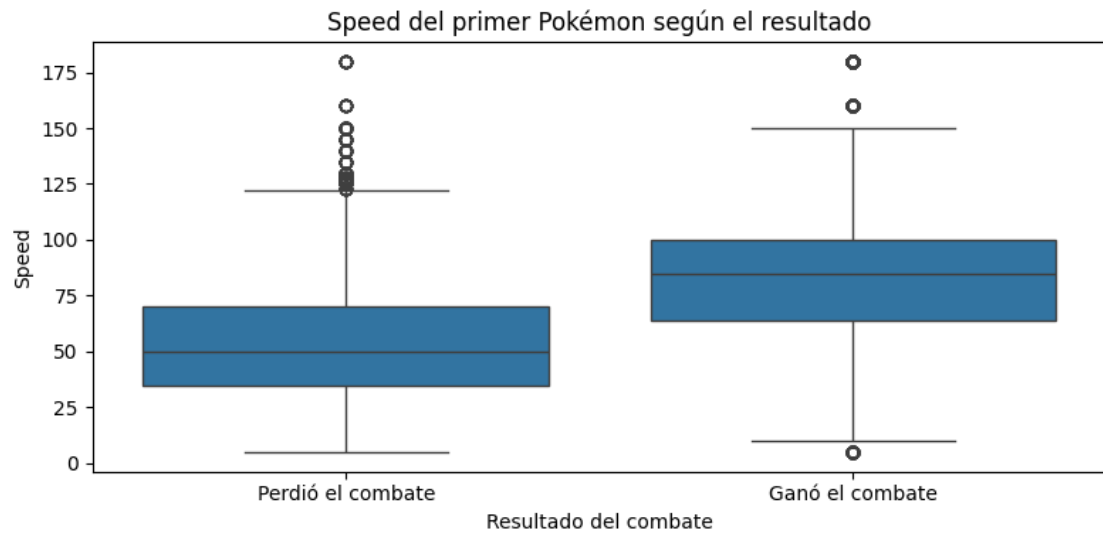
El script *ModelTrain.py* realiza las siguientes tareas:

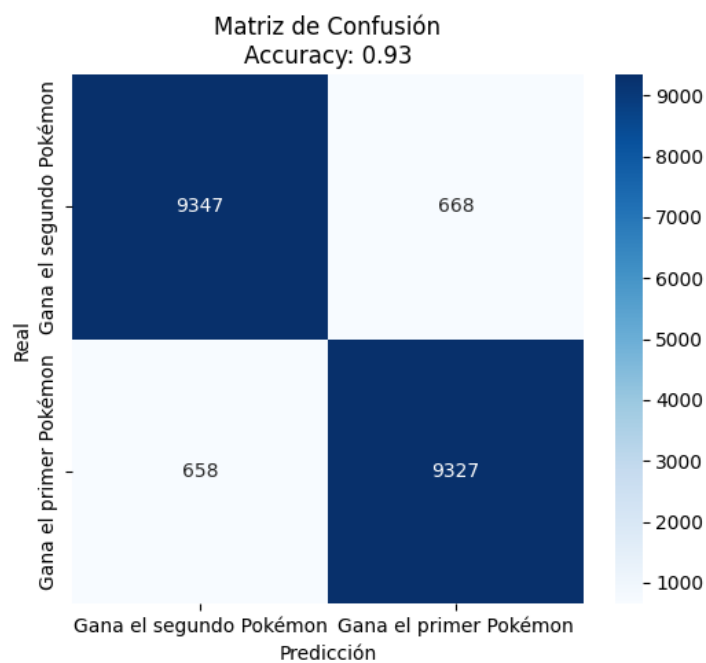
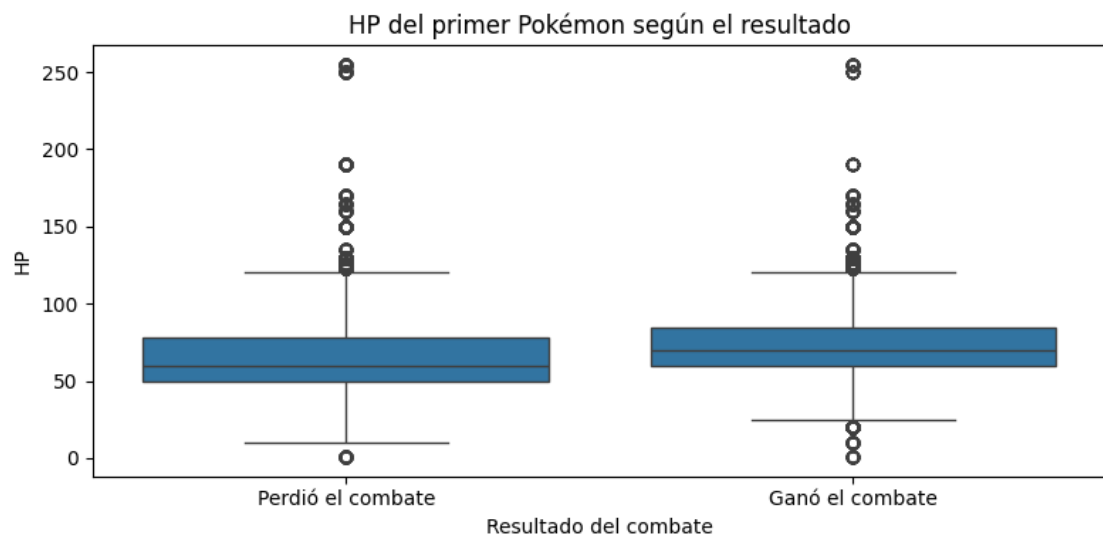
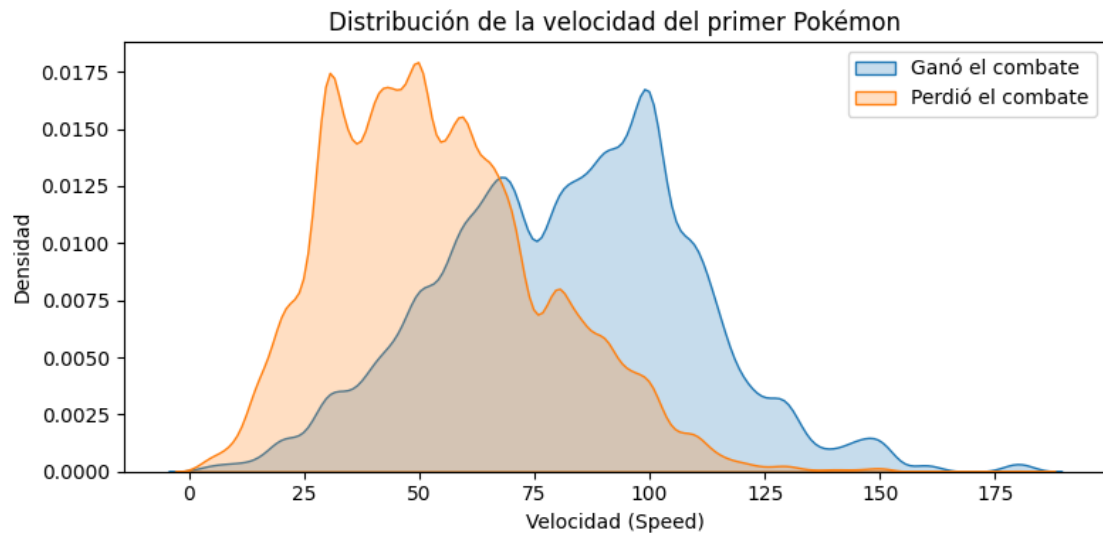
- Carga y prepara los datos, incluyendo la generación de simetría en los combates.

- Fusiona las estadísticas de los Pokémon y aplica codificación one-hot para las variables categóricas.
- Visualiza la distribución de las variables y la relación con la variable objetivo (el ganador).
- Entrena un modelo Random Forest para la predicción del ganador.
- Evalúa el modelo mostrando métricas como la exactitud, la matriz de confusión y un reporte detallado.
- Guarda el modelo entrenado como *pokemon_rf.pkl* y exporta los resultados en *predicciones_vs_real.csv* para su uso posterior.

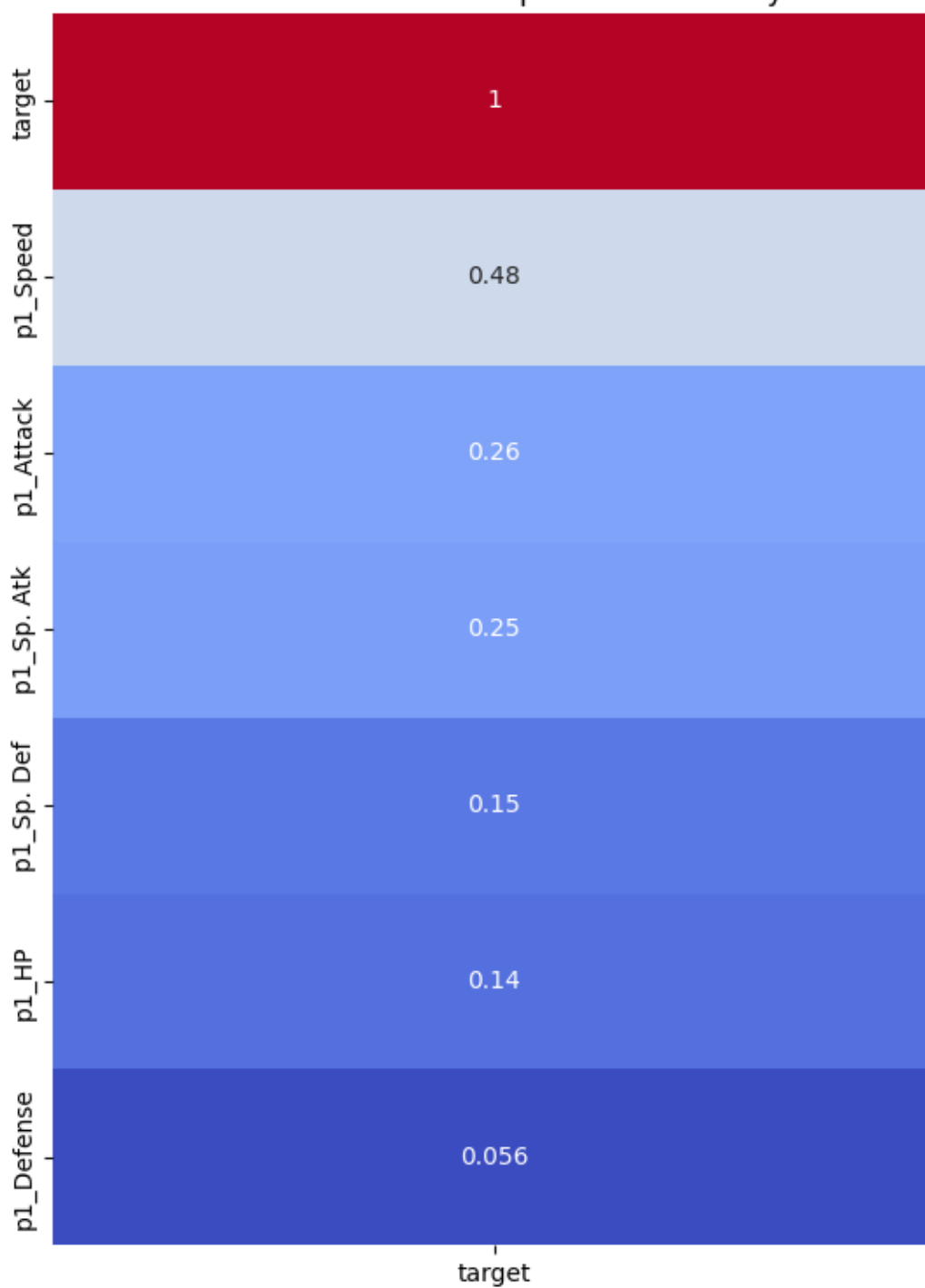
El modelo predictivo se basa en datos históricos de combates, utilizando los archivos *pokemon.csv* y *combats_simetrico.csv*. Emplea las características de ambos Pokémon para estimar la probabilidad de victoria y determinar un ganador probable.







Correlación entre estadísticas del primer Pokémon y el resultado

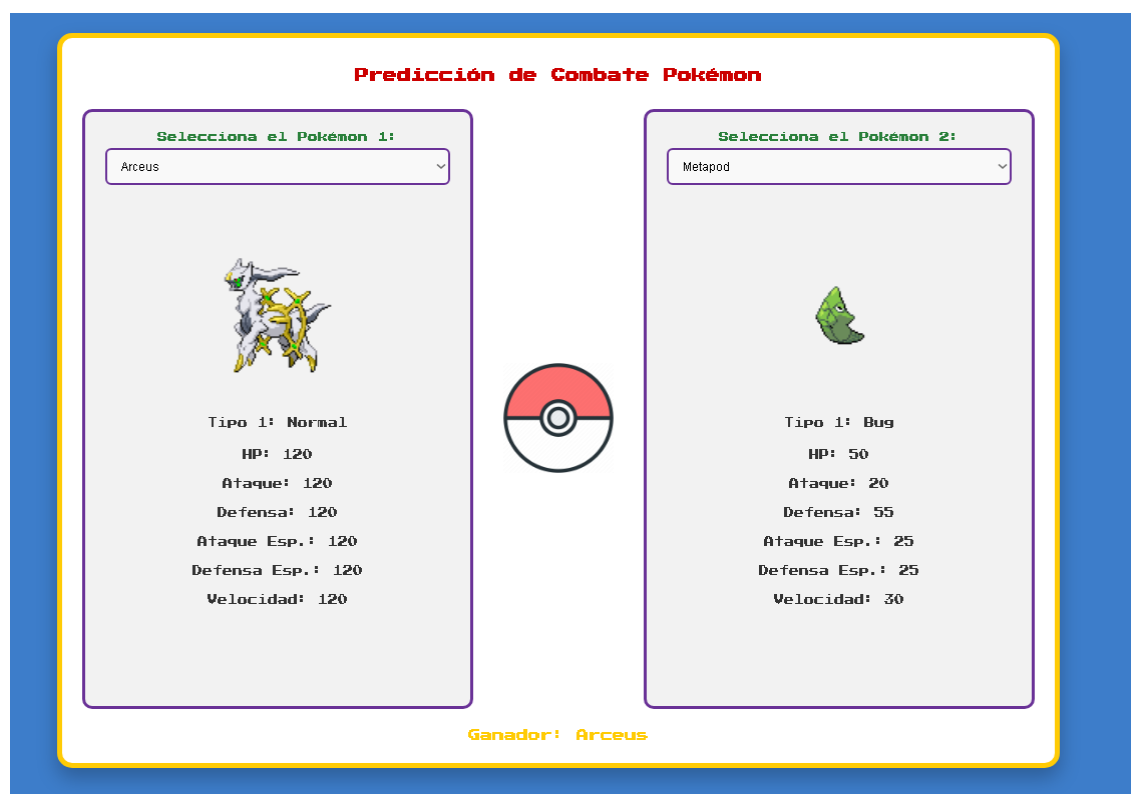


	precision	recall	f1-score	support
Gana el segundo Pokémon	0.93	0.93	0.93	10015.0
Gana el primer Pokémon	0.93	0.93	0.93	9985.0
accuracy	0.93	0.93	0.93	0.93
macro avg	0.93	0.93	0.93	20000.0
weighted avg	0.93	0.93	0.93	20000.0

Interfaz de Usuario

Se desarrolló una interfaz web con HTML, CSS y JavaScript, donde el usuario puede seleccionar dos Pokémon desde menús desplegables con los nombres disponibles.

Al seleccionar cada Pokémon, el sistema muestra sus estadísticas detalladas y la imagen correspondiente. Al presionar el botón central, se realiza una llamada al backend que ejecuta la predicción y muestra el resultado del combate.



Implementación Técnica

Frontend

- **HTML/CSS:** Diseño con estética retro para emular el estilo clásico de Pokémon.

- **JavaScript:** Manejo dinámico de la carga de estadísticas e imágenes desde el backend y la API pokeapi.co, además de enviar las peticiones para la predicción.

Backend

El backend está desarrollado en Python con Flask y maneja la lógica del servidor, la interacción con los datos y la comunicación con la interfaz web.

Flask(Python)—app.py:

La aplicación Flask expone los endpoints necesarios para la interacción con el frontend:

- **Endpoint de estadísticas:** Consulta las estadísticas detalladas de un Pokémon específico, obtenidas del archivo *pokemon.csv*, y devuelve la información en formato JSON para su visualización.
- **Endpoint de predicción:** Recibe los nombres de dos Pokémon, consulta el modelo previamente entrenado y cargado en memoria (*pokemon_rf.pkl*), y devuelve el ganador probable basándose en los datos históricos de combate y las características de ambos Pokémon. El modelo se carga una sola vez al iniciar la aplicación para optimizar el rendimiento y permitir respuestas rápidas en cada predicción.

Conclusión

Este proyecto integra datos históricos y estadísticos de Pokémon para desarrollar un modelo predictivo funcional y accesible a través de una interfaz intuitiva. La combinación de machine learning con un frontend interactivo permite a los usuarios simular combates y conocer probabilidades de victoria basadas en datos reales.

Notas

El modelo tiene áreas claras de mejora, ya que no incluye factores como:

- Las características otorgadas por la tabla de tipos a las estadísticas de los Pokémon.
- El nivel de los Pokémon.
- Variaciones en estadísticas por naturaleza, IV's o EV's. Por lo tanto, es probable que existan discrepancias evidentes para conocedores expertos en batallas Pokémon.
- Aquí el repositorio: <https://github.com/SansSloth/Modelo-Predictivo-de-Combates-Pok-mon>

Jair Jaime Hernandez Gonzalez