

Práctica programación

By:

Eduardo Hortelano,
Lucía Parreño,
Pablo De la Iglesia.

Carga de datos (Wikidex)

- La primera parte de la carga de datos la hemos realizado a través de web scrabbing usando la wikidex.
- Hemos realizado dos funciones, una de ellas en la que dada una generación y un tipo de pokemon, devuelve todos los pokemons de ese tipo y esa generación (en esta función es de donde obtenemos la información de la wikidex)
- La segunda función simplemente especifica las generaciones de las cuales queremos los pokemons (en nuestro caso las dos primeras) y llama a la primera función para crear la lista de pokemons.

Carga de datos (Pokeapi)

- La segunda parte de la carga de datos la hemos realizado con peticiones a una api (pokeapi)
- Primero tenemos una función que dada una lista de entradas devuelve la misma lista de entradas si están en inglés o un texto de error si no está en inglés
- La segunda función utiliza la api para obtener todos los movimientos que pueden aprender todos los pokemons. Almacena en listas el nombre del movimiento, su descripción y la potencia del movimiento (realmente almacena 1 si tiene potencia y 0 si no la tiene para poder diferenciar si el movimiento es de ataque o defensa).
- La tercera función crea una base de datos con las listas creadas anteriormente.
- Por último tenemos una función que dado un pokemon devuelve una base de datos con todos los posibles movimientos que puede aprender ese pokemon (la estructura de la base de datos es la misma que la anterior).

Equipo defensivo

- Las dos primeras funciones nos permiten obtener las stats defensivas de una lista de pokemons
- La tercera función nos crea un diccionario en el que cada pokemon de una lista es la key y sus stats son los values. Devuelve dicho diccionario pero con las keys ordenadas de mayor a menor según las stats
- Por último la función que devuelve los seis mejores pokemons según las stats 'defense', 'hp', y 'attack'

Equipo defensivo

1	pokemon	defense	hp	attack
2	Shuckle	230	20	10
3	Forretress	140	75	90
4	Scizor	100	70	130
5	Pinsir	100	65	125
6	Pineco	90	50	65
7	Scyther	80	70	110

Equipo ofensivo

- La primera función busca todos los movimientos que puede aprender un tipo de pokemon (en nuestro caso bicho). Crea un dataframe con tres columnas (nombre, power y ataque base)
- La tercera función nos crea el equipo ofensivo calculando el poder de un pokemon (Poder= Ataque especial * daño base el ataque más potente del tipo) y devuelve un diccionario con los 6 pokemons con más ataque.

Equipo ofensivo

1	pokemon	special-attack	attack	power
2	Venomoth	90	65	10800
3	Butterfree	90	45	10800
4	Yanma	75	65	9000
5	Parasect	60	95	7200
6	Ariados	60	90	7200
7	Forretress	60	90	7200

Sinergia de equipos

- La primera función identifica los tipos elementales exactos de un Pokémon concreto consultando la API.
- La segunda función evalúa la versatilidad de los candidatos: cuenta cuántos tipos tiene cada uno y selecciona los 6 que aportan mayor variedad al equipo.
- Por último, la función de sinergia unifica los mejores candidatos defensivos y ofensivos en un solo grupo para generar el equipo final con la mejor cobertura de tipos posible.

NLP y Clasificador (vectorización de los dataframes)

- La primera función actúa como un filtro inteligente: limpia el texto eliminando palabras vacías extrae solo la "esencia" de cada descripción(sustantivos y adjetivos en su forma raíz).
- Las funciones de vocabulario analizan todas las descripciones disponibles para crear un diccionario personalizado con las palabras más frecuentes y relevantes del mundo Pokémon.
- Finalmente, las funciones de vectorización traducen el lenguaje humano a lenguaje matemático: convierten cada descripción textual en una secuencia numérica (vector) que el modelo de Inteligencia Artificial puede entender y procesar.

NLP y Clasificador (entrenamiento del modelo y clasificación)

- La primera función entrena el modelo de Inteligencia Artificial: analiza los datos ya vectorizados para aprender patrones y establecer la relación matemática entre las descripciones y su categoría (Ofensivo o Defensivo).
- La segunda función aplica el aprendizaje: utiliza el modelo ya entrenado para predecir la clase de cada movimiento y añade esta nueva información a nuestros datos para poder compararla con la clasificación original.

Equipo final

- Dado un pokemon se elige el primer movimiento defensivo y los tres mejores movimientos ofensivos. Se devuelven estos 4 mejores movimientos en un dataframe.
- Por último se crea el equipo final con los mejores pokemons, sus tipos, sus movimientos, de que clase son (ofensivo o defensivo) y la predicción que ha habido para cada movimiento.

Equipo final

0	Shuckle	bug and rock	steel-roller double-edge meteor-beam string-shot	Offensive Offensive Offensive Defensive	Offensive Offensive Offensive Defensive
1	Forretress	bug and steel	explosion self-destruct hyper-beam counter	Offensive Offensive Offensive Defensive	Offensive Offensive Defensive Defensive
2	Scizor	bug and steel	hyper-beam giga-impact steel-beam swords-dance	Offensive Offensive Offensive Defensive	Defensive Defensive Offensive Defensive
3	Scyther	bug and flying	hyper-beam giga-impact skull-bash swords-dance	Offensive Offensive Offensive Defensive	Defensive Defensive Offensive Defensive
4	Venomoth	bug and poison	hyper-beam giga-impact solar-beam whirlwind	Offensive Offensive Offensive Defensive	Defensive Defensive Offensive Defensive
5	Butterfree	bug and flying	hyper-beam giga-impact solar-beam whirlwind	Offensive Offensive Offensive Defensive	Defensive Defensive Offensive Defensive