Analiza zbioru danych Pokemon dataset

Zbiór danych składa się z 721 instancji. Każda z instancji odpowiada jednemu Pokemonowi. Zbiór został stworzony przy pomocy zbiorów danych dostępnych na stronach: bulbapedia, pokemon.com, pokemondb. Ich łączna zawartość dostępna jest na stronie Kaggle.

Do analizy zbioru użyty został Python, z Jupyter Notebookiem wykorzystując biblioteki matplotlib, seaborn oraz pandas, który dostępny jest także w środowisku R i możliwe jest bezpośrednie przekształcanie dataframes z jednego języka do drugiego.

Zbiór ten jest używany podczas uczenia dzieci statystyki.

Atrybuty każdej instancji to:

ID: numer pożądkowy,

Name: imię pokemona,

Type 1: każdy pokemon posiada przynajmniej jeden typ (np. ognisty, wodny, itp.),

Type 2: niektóre pokemony posiadają dodatkowy drugi typ,

Total: suma wszystkich statystyk pokemona,

HP: liczba punktów życia pokemona,

Attack: modyfikator ataku pokemona,

Defense: modyfikator defensywny pokemona,

SP Atk: modyfikator ataku specjalnego pokemona,

SP Def: modyfikator defensywy specjalnej pokemona,

Speed: prędkość pokemona,

Generation: generacja pokemonów, z których pochodzi dana instancja.

Na początku każdej analizy dobrze jest wypisać przynajmniej część danych aby lepiej poznać dane z jakimi się pracuje. Po załadowaniu zbioru poprzez funkcję *pandas.read_csv('pokemon.csv')* zbiór danych dostępny jest w postaci dataframe. Dataframe posiada wiele funkcji pozwalających na wizualizację zbioru, jedną z najczęściej używanych do wyżej opisanej funkcji jest *df.head()*.

```
In [22]: pokemon.head()
```

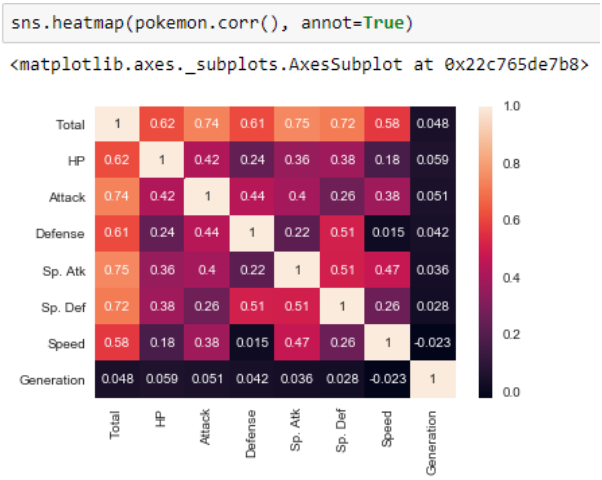
Out[22]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1

Rysunek 1 Wypisanie kilku pierwszych instancji zbioru za pomocą metody df.head()

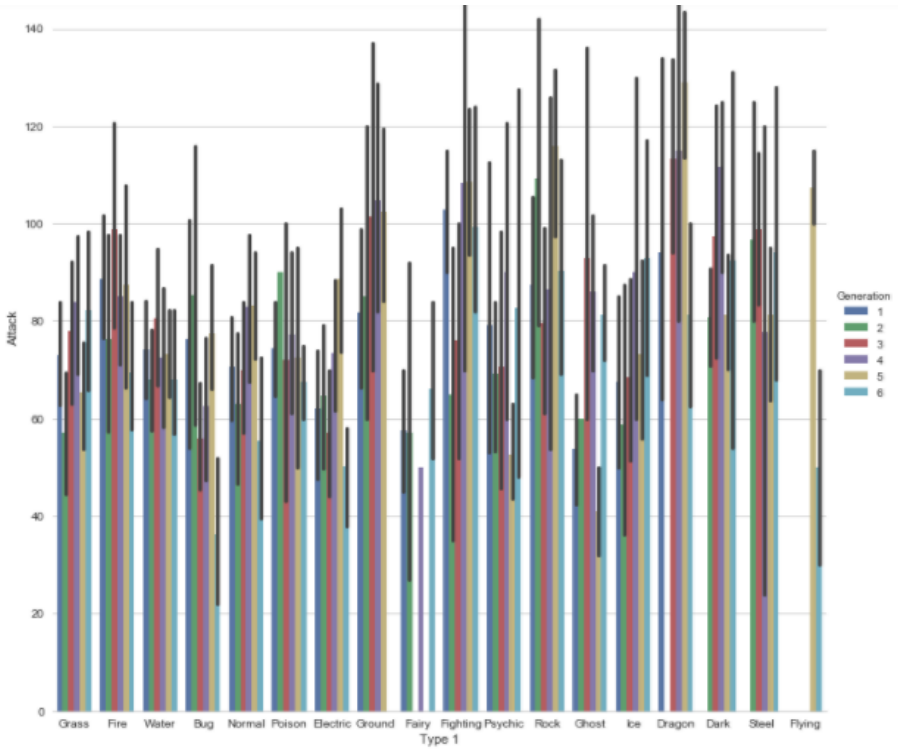
Na rysunku 1 widzimy, że niektóre dane są w formie numerycznej, a niektóre w formie kategorii. Ważną informacją, którą można zauważyć jest potencjalny problem z kolumną *Type 2*. Jak widać w niektórych obserwacjach może tam wystąpić NaN, który może być problematyczny w analizie i uczeniu maszynowym na danych. Inną bardzo pomocną metodą

jest *df.describe()*, pokazujący takie informacje jak rozłożenie danych, czy wariancje i odchylenia danych.



Rysunek 2 Mapa korelacji pomiędzy wszystkimi atrybutami.

Dzięki mapie korelacji widzimy wiele zależności. Między innymi to że Atak i Atak specjalny pokemona wpływa znacząco na łączną statystykę pokemona (Total). Innym ciekawym faktem jest to, że wraz ze wzrostem prędkności pokemona (Speed), statystyka defensywana (Defence) jest zazwyczaj niska co można wysnioskować z corelacji na poziomie 0.015.



Rysunek 3 Analiza statystyki ataku ze względu na pogrupowanie po typach oraz generacjach.

Zachęcam do analizy wykresów w notebooku, ze względu na złożoność wykresu i małej formie w tym dokumencie, co nie sprzyja jego analizie. W powyższej analizie ze zmienną grupującą porównane zostały Ataki pokemonów dla każdego z typów ze względu na generacje w jakiej występują. Jak widać po czarnym pasku, odchylenia są dosyć spore jednakże jednoznacznie widać, że typ Fairy cechuje się zazwyczaj najmniejszym czynnikiem ataku, a typ Dragon najwyższym.