

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙖🕮🙐—

****

**BỘ MÔN: KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI : PHÂN LOẠI POKEMON**

**Trương Nhật Nam – 3119410262**

**GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN THANH PHƯỚC**

Thành phố Hồ Chí Minh, tháng 05 năm 2024

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc166385754)

[CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI 2](#_Toc166385755)

[CHƯƠNG II. MÔ TẢ DỮ LIỆU 3](#_Toc166385756)

[**1. NGUỒN GỐC CỦA BỘ DỮ LIỆU** 3](#_Toc166385757)

[**2. SỐ LƯỢNG MẪU VÀ THUỘC TÍNH** 3](#_Toc166385758)

[CHƯƠNG III. TIỀN XỬ LÝ DỮ LIỆU 4](#_Toc166385759)

[**1. CÁC THƯ VIỆN** 4](#_Toc166385760)

[**2. XỬ LÝ DỮ LIỆU** 5](#_Toc166385761)

[CHƯƠNG IV. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ 10](#_Toc166385762)

[**1. PHÂN TÍCH THỐNG KÊ MÔ TẢ** 10](#_Toc166385763)

[**2. PHÂN TÍCH ĐƠN BIẾN** 11](#_Toc166385764)

[**2.1. PHÂN TÍCH ĐƠN BIẾN VỚI CÁC BIẾN PHÂN LOẠI** 12](#_Toc166385765)

[**2.2. PHÂN TÍCH ĐƠN BIẾN VỚI CÁC BIẾN SỐ** 15](#_Toc166385766)

[**3. PHÂN TÍCH ĐA BIẾN** 20](#_Toc166385767)

[CHƯƠNG V. KHAI PHÁ DỮ LIỆU 23](#_Toc166385768)

[**1. CÁC MÔ HÌNH HỌC MÁY** 23](#_Toc166385769)

[**1.1. K-NEAREST NEIGHBORS (KNN)** 23](#_Toc166385770)

[**1.2. SUPPORT VECTOR MACHINE (SVM)** 24](#_Toc166385771)

[**1.3. RANDOM FOREST (RF)** 25](#_Toc166385772)

[**1.4. K-MEANS CLUSTERING** 25](#_Toc166385773)

[**2. CÁC CÂU HỎI CẦN GIẢI QUYẾT** 26](#_Toc166385774)

[CHƯƠNG VI. ĐÁNH GIÁ VÀ CHỌN THUẬT TOÁN 27](#_Toc166385775)

[**1. MÔ HÌNH KNN** 27](#_Toc166385776)

[**2. MÔ HÌNH SVC** 28](#_Toc166385777)

[**3. MÔ HÌNH RANDOM FOREST** 29](#_Toc166385778)

[**4. KIỂM THỬ MÔ HÌNH VÀ TRẢ LỜI CÂU HỎI** 31](#_Toc166385779)

[CHƯƠNG VII. KẾT QUẢ VÀ THẢO LUẬN 33](#_Toc166385780)

[CHƯƠNG VIII. KẾT LUẬN 34](#_Toc166385781)

[TÀI LIỆU THAM KHẢO 35](#_Toc166385782)

# LỜI MỞ ĐẦU

Trong thời đại số hóa ngày nay, dữ liệu đã trở thành một tài nguyên vô cùng quý giá, mở ra những cơ hội không ngừng cho việc phân tích và khai phá thông tin. Trong lĩnh vực khoa học máy tính, khai phá dữ liệu là quá trình tìm kiếm thông tin hữu ích từ dữ liệu lớn, phức tạp, đa dạng, và thường là không có cấu trúc. Trong báo cáo này, chúng ta sẽ khám phá một trong những ứng dụng phổ biến của khai phá dữ liệu - đó là phân loại dữ liệu.

Tiến bộ công nghệ đã tạo ra một nguồn dữ liệu khổng lồ từ nhiều lĩnh vực khác nhau, từ tài chính đến y tế và giáo dục. Dữ liệu này chứa đựng những thông tin quý báu có thể giúp chúng ta hiểu rõ hơn về mô hình hoạt động của nhiều hệ thống phức tạp. Việc phân loại dữ liệu từ các nguồn này có thể giúp chúng ta đưa ra những quyết định thông minh, dựa trên việc phân tích dữ liệu một cách hiệu quả.

Trong khi việc phân loại Pokémon có thể trông như một ứng dụng giải trí, thì nó cũng đòi hỏi sự áp dụng của các phương pháp và thuật toán khai phá dữ liệu mạnh mẽ. Bằng cách sử dụng các kỹ thuật như học máy và mạng nơ-ron, chúng ta có thể xây dựng các mô hình có khả năng phân loại Pokémon dựa trên các thuộc tính và đặc điểm của chúng. Các mô hình này có thể giúp người chơi hiểu rõ hơn về ưu và nhược điểm của từng loài Pokémon, từ đó tối ưu hóa chiến thuật trong trò chơi.

Khai phá dữ liệu phân loại Pokémon không chỉ mang lại giá trị giải trí mà còn mở ra một cánh cửa mới cho sự hiểu biết về cách mà khoa học máy và khai phá dữ liệu có thể được áp dụng vào các lĩnh vực khác nhau. Điều này là một minh chứng rõ ràng cho sức mạnh của dữ liệu và khả năng biến chúng thành thông tin hữu ích trong xã hội ngày nay.

# CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI

**Đề tài**

Phân loại Pokemon

Tập dữ liệu sử dụng: “Pokemon Pokedex with Stats”.

Trong bài này nhóm chọn tập dữ liệu “Pokemon Pokedex with Stats” trên Kaggle:

<https://www.kaggle.com/datasets/arnavvvvv/pokemon-pokedex>

**Bối cảnh**

Anh A, một người đam mê trò chơi Pokémon, đang tìm kiếm dữ liệu về các loài Pokémon để thú vị hóa trải nghiệm chơi game của mình. Tuy nhiên, với sự đa dạng và phong phú của thế giới Pokémon, việc tìm kiếm và tổng hợp dữ liệu có thể là một thách thức.

Anh A muốn tìm hiểu về sự tương quan giữa các chỉ số (máu, tấn công, tốc độ,…), phân loại pokemon đó thuộc vào lớp nào (tấn công, phòng thủ, tốc độ,…) và đó có phải là pokemon huyền thoại hay không.

Trong vấn đề này, bài toán cần giải quyết được phân loại pokemon đó có phải là pokemon huyền thoại không.

**Mục đích**

**Đánh giá và so sánh giữa các loại pokemon:** thông qua các chỉ số của pokemon như máu, công phép, công vật lý, tốc độ,… từ đó có thể xác định lớp của pokemon đó.

**Phân tích các chỉ số ảnh hưởng đến 1 loại pokemon:** thông qua các chỉ số của pokemon như máu, công phép, công vật lý, tốc độ,… từ đó có thể xác định pokemon đó thuộc loại (huyền thoại, sử thi, rất hiếm, hiếm, thường, không hiếm) nào.

**Biến dữ liệu thô thành thông tin hữu ích:** từ cụm dữ liệu thô ta có thể khai phá, tích hợp dữ liệu thành các dữ liệu trực quan hơn.

# CHƯƠNG II. MÔ TẢ DỮ LIỆU

## **1. NGUỒN GỐC CỦA BỘ DỮ LIỆU**

Dataset:.(<https://www.kaggle.com/datasets/arnavvvvv/pokemon-pokedex>)

Tập dữ liệu "Pokemon Pokedex with Stats" được tạo bởi KUMAR ARNAV và được đăng tải trên trang web Kaggle - một nơi tập trung các tập dữ liệu và cuộc thi khoa học dữ liệu trực tuyến. Tập dữ liệu này chứa thông tin về hơn 1000 loại pokemon và bao gồm số lượng, tên, hệ cũng như các chỉ số cơ bản: HP, Tấn công, Phòng thủ, Tấn công đặc biệt, Phòng thủ đặc biệt và Tốc độ.

Tập dữ liệu này được tạo ra nhằm mục đích phục vụ cho các hoạt động phân loại pokemon và dự đoán loại pokemon. Đây là tập dữ liệu thích hợp cho các nhà nghiên cứu trong lĩnh vực khoa học dữ liệu, học máy và trí tuệ nhân tạo.

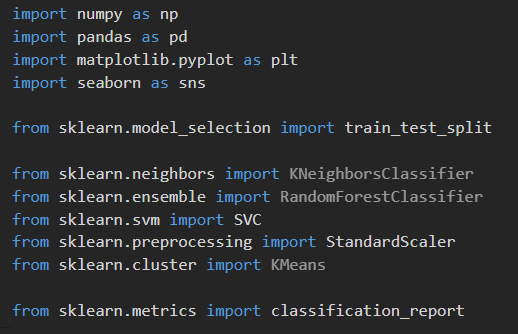
## **2. SỐ LƯỢNG MẪU VÀ THUỘC TÍNH**

Tập dữ liệu bao gồm 13 cột và 1044 hàng, trong đó:

|  |  |
| --- | --- |
| **#** | Số thứ tự của pokemon |
| **Name** | Tên của pokemon |
| **Type 1** | Hệ thứ nhất của pokemon |
| **Type 2** | Hệ thứ hai của pokemon |
| **Total** | Tổng chỉ số của pokemon |
| **HP** | Máu của pokemon |
| **Attack** | Tấn công vật lý của pokemon |
| **Defense** | Phòng thủ vật lý của pokemon |
| **Sp. Atk** | Tấn công phép thuật/đặc biệt của pokemon |
| **Sp. Def** | Phòng thủ phép thuật/đặc biệt của pokemon |
| **Speed** | Tốc độ của pokemon |
| **Generation** | Thế hệ của pokemon |
| **Legendary** | Phải pokemon huyền thoại không(TRUE/FALSE) |

# CHƯƠNG III. TIỀN XỬ LÝ DỮ LIỆU

## **1. CÁC THƯ VIỆN**

****

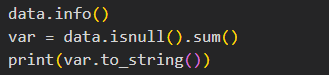
* **numpy (imported as np):** Thư viện Python cho tính toán khoa học và toán học trên mảng đa chiều. Nó cung cấp một đối tượng mảng đa chiều, các chức năng để làm việc với mảng này, cũng như các công cụ để làm việc với ma trận.
* **pandas (imported as pd):** Thư viện Python cung cấp cấu trúc dữ liệu và công cụ phân tích dữ liệu dễ sử dụng. Nó cung cấp các đối tượng như DataFrame để lưu trữ và xử lý dữ liệu dưới dạng bảng.
* **matplotlib.pyplot (imported as plt):** Thư viện trực quan hóa dữ liệu trong Python, cho phép tạo ra các biểu đồ, đồ thị, hình vẽ, vv.
* **seaborn (imported as sns):** Thư viện trực quan hóa dữ liệu dựa trên matplotlib, giúp tạo ra các biểu đồ phức tạp một cách dễ dàng hơn và cải thiện tính thẩm mỹ của biểu đồ.
* **sklearn.model\_selection.train\_test\_split:** Hàm từ thư viện scikit-learn được sử dụng để chia dữ liệu thành các tập huấn luyện và tập kiểm tra để đánh giá hiệu suất của mô hình máy học.
* **sklearn.neighbors.KNeighborsClassifier:** Một trong các mô hình học máy cơ bản, dựa trên việc tìm kiếm các điểm láng giềng gần nhất của một điểm dữ liệu để dự đoán lớp của nó.
* **sklearn.ensemble.RandomForestClassifier:** Một mô hình học máy dựa trên phương pháp rừng ngẫu nhiên, trong đó nhiều cây quyết định được huấn luyện và kết hợp để tạo ra dự đoán.
* **sklearn.svm.SVC:** Mô hình này là một phần của module svm trong scikit-learn và được sử dụng để phân loại các điểm dữ liệu sao cho khoảng cách từ các điểm dữ liệu tới các ranh giới là lớn nhất.
* **sklearn.preprocessing.StandardScaler:** Đây là một phương pháp trong việc tiền xử lý dữ liệu, được sử dụng để chuẩn hóa dữ liệu bằng cách loại bỏ trung bình và chia cho độ lệch chuẩn của mỗi tính năng.
* **sklearn.cluster.KMeans**: Đây là một thuật toán phân cụm phổ biến trong học máy không giám sát. Thuật toán này tìm cách chia tập dữ liệu thành các nhóm (cụm) sao cho các điểm trong cùng một nhóm có tính chất tương tự nhau và các điểm giữa các nhóm có tính chất khác biệt lớn.
* **sklearn.metrics.classification\_report:** Hàm từ thư viện scikit-learn được sử dụng để hiển thị báo cáo phân loại, bao gồm các chỉ số đánh giá hiệu suất của mô hình phân loại như precision, recall và f1-score.

## **2. XỬ LÝ DỮ LIỆU**

***- Đọc dữ liệu***

****

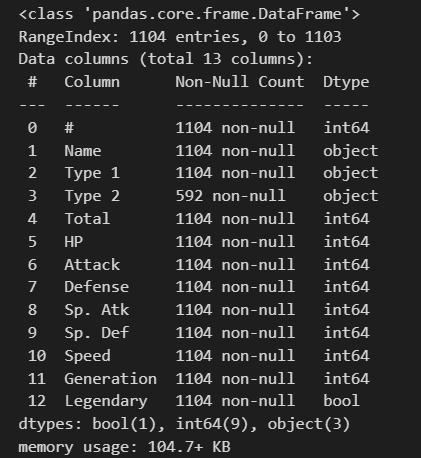
***- Tổng quan về dữ liệu***

******

Chúng ta sử dụng phương thức **data.info()** để hiển thị thông tin tổng quan về các cột trong DataFrame bao gồm:

* Tên các cột và số lượng giá trị không null trong mỗi cột.
* Tổng số lượng hàng và cột trong bộ dữ liệu.
* Kiểu dữ liệu của mỗi cột.
* Số lượng bộ nhớ đã sử dụng.

Sau khi thực hiện lệnh **data.info()** ta sẽ được kết quả như hình:



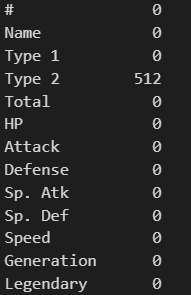
***Kết quả trên biểu diễn:*** trong 13 cột dữ liệu có 3 cột name, type 1, type 2 là string, legendary là boolen và các cột còn lại là int.

***- Xử lý các giá trị thiếu***

Để xử lý dữ liệu khuyết trong bộ dữ liệu, ta thực hiện lệnh **data.isnull().sum()** trong thư viện pandas để kiểm tra xem bộ dữ liệu có bị khuyết hay không.

****

Sau khi thực hiện ta được kết quả như hình bên dưới. Có thể nhận thấy tổng số lượng giá trị trống ở cột Type 2 = 512 điều đó có nghĩa là sẽ có 512 pokemon chỉ có 1 hệ. Dữ liệu thiếu này không gây ảnh hưởng đến DataFrame.

****

***- Xử lý dữ liệu bị trùng***

Để kiểm tra giá trị trùng ta sử dụng lệnh **data.duplicated().sum()** tìm ra giá trị bị trùng

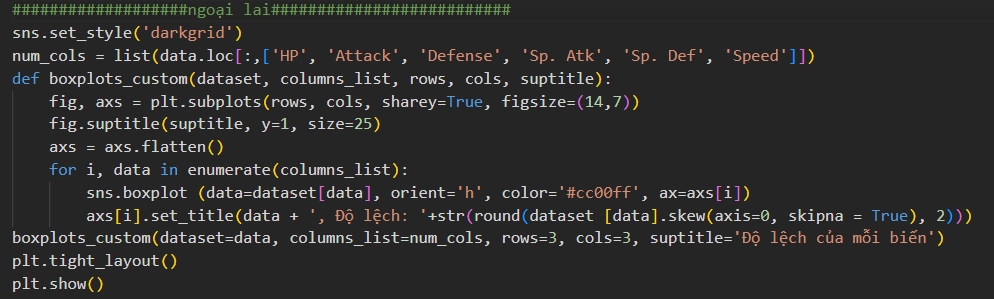


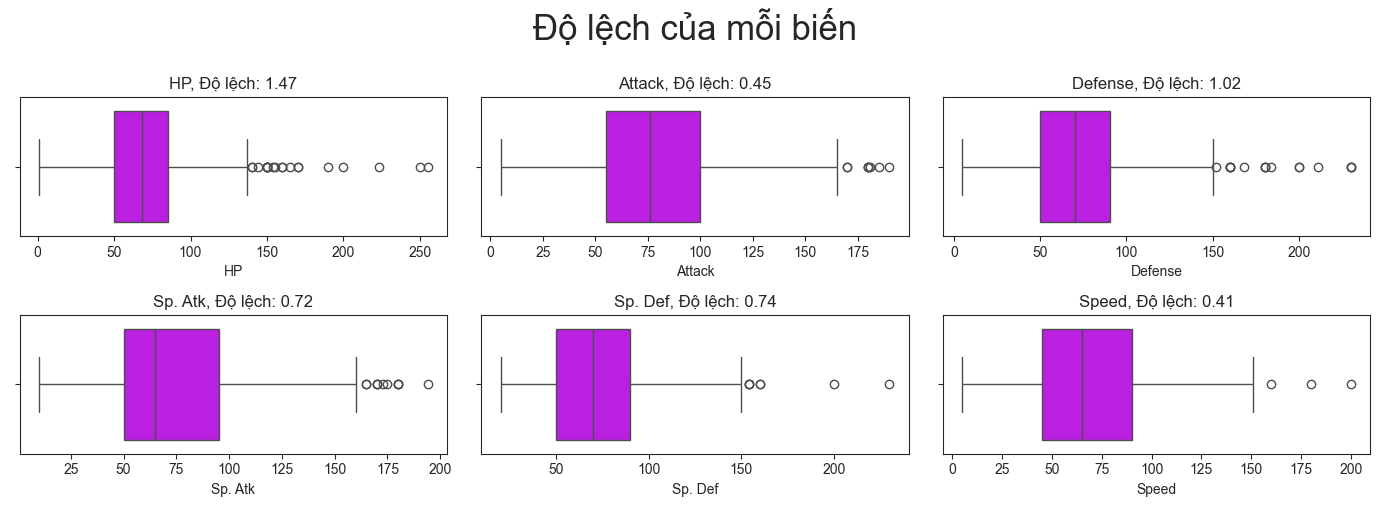
Sau khi thực hiện ta được kết quả như hình bên dưới. Có thể nhận thấy tổng số lượng giá trị bị trùng là 0 điều đó có nghĩa là không có giá trị nào trùng lặp. Do DataFrame không có giá trị trùng nên số lượng dòng của DataFrame vẫn giữ nguyên.



***- Xử lý dữ liệu ngoại lai***

Trước khi xử lý ta tiến hành kiểm tra các giá trị ngoại lai bằng cách tim ra độ lệch của từng biến bằng biểu đồ boxplots



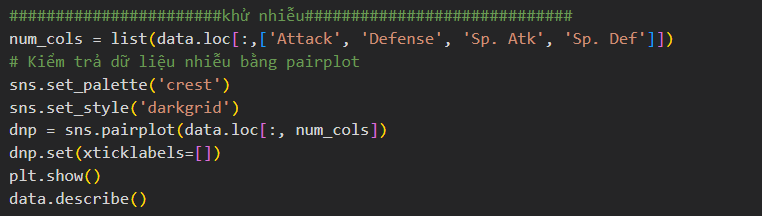


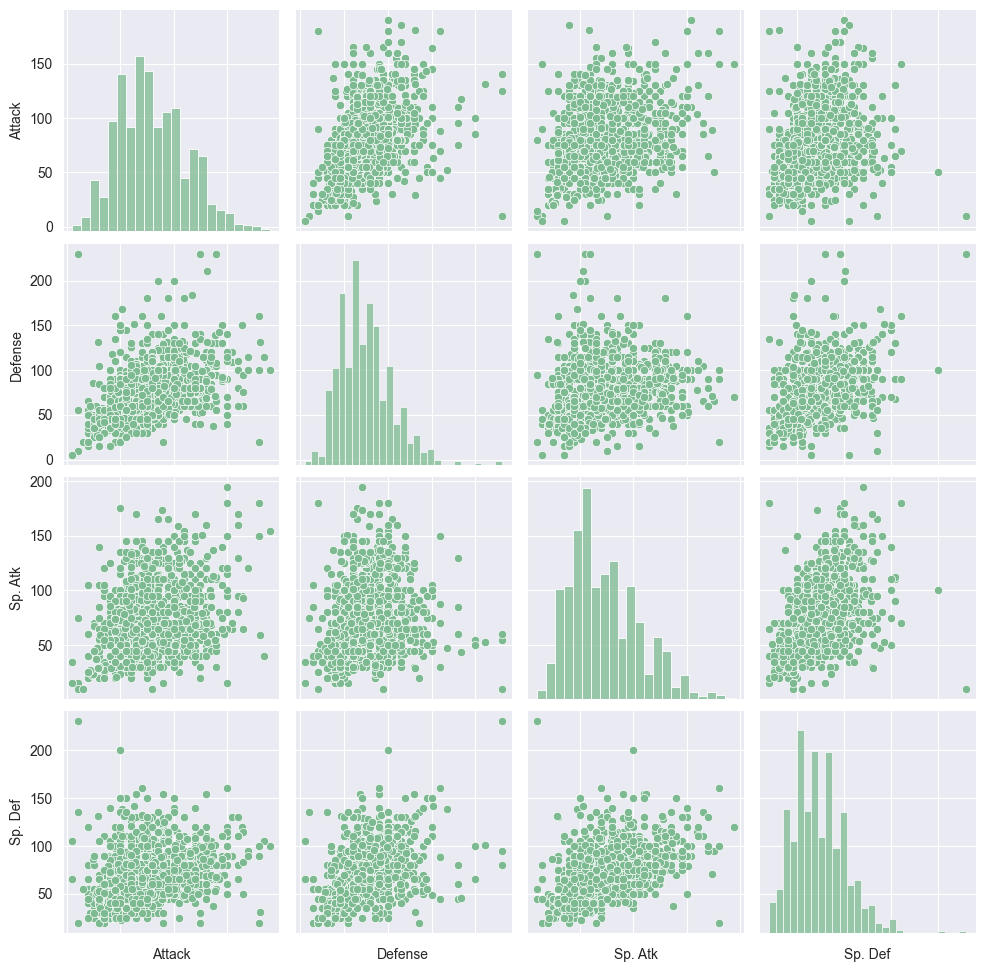
Quá trình xử lý ngoại lai có tác dụng rất lớn đến quá trình phân tích và khai phá dữ liệu bao gồm:

1. **Cải thiện chất lượng dữ liệu:** Loại bỏ các các giá trị ngoại lai giúp cải thiện chất lượng dữ liệu bằng cách giảm thiểu các giá trị bất thường và không chính xác.
2. **Tăng tính chính xác trong phân tích:** Loại bỏ các giá trị ngoại lai giúp tăng tính chính xác trong phân tích dữ liệu, đặc biệt là khi sử dụng các kỹ thuật thống kê như tính trung bình, độ lệch chuẩn và phân phối.
3. **Giảm thiểu ảnh hưởng của nhiễu:** Các giá trị ngoại lai có thể được tạo ra bởi nhiễu trong quá trình thu thập dữ liệu. loại bỏ chúng giúp giảm bớt ảnh hưởng của nhiễu trên kết quả.
4. **Cải thiện hiệu suất và độ tin cậy của mô hình:** Khi sử dụng dữ liệu dẵ được xử lý ngoại lai huấn luyện mô hình, nó có thể giúp cái thiện hiệu suất và độ tin cậy của mô hình.

***- Xử lý dữ liệu nhiễu***

Để xử lý dữ liệu nhiễu ta tiến hành kiểm tra dữ liệu bằng biểu đồ pairplot





Như hình trên ta có thể thấy không có bất kí dữ liệu nhiễu nào

# CHƯƠNG IV. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

## **1. PHÂN TÍCH THỐNG KÊ MÔ TẢ**

Để giúp người dùng có cái nhìn tổng quan về tập dữ liệu “Pokemon Pokedex with Stats”, chúng ta tiến hành quá trình phân tích thống kê mô tả dữ liệu bằng hàm **data.describe().**

****

***Quá trình phân tích thống kê mô tả dữ liệu hỗ trợ chúng ta như sau:***

* Biết Đánh giá trung tâm và phân bố của dữ liệu: Bằng cách tính toán giá trị trung bình, trung vị và phân vị của dữ liệu, chúng ta có thể đánh giá được nơi tập trung chính và sự phân phối của dữ liệu.
* Hiểu rõ độ biến động của dữ liệu: Phân tích độ lệch của dữ liệu giúp ta hiểu được mức độ đồng đều dữ liệu và nhận biết sự biến động trong tập dữ liệu.
* Phát hiện giá trị ngoại lai và bất thường: Qua việc phát hiện các giá trị ngoại lai và bất thường, ta có thể xác định những dữ liệu cần được kiểm tra và loại bỏ để đảm bảo tính chính xác của kết quả phân tích.
* Xác định phân phối của dữ liệu: Hiểu rõ về phân phối của dữ liệu giúp ta lựa chọn mô hình phù hợp nhất cho bài toán dự đoán.
* Tóm tắt thông tin thống kê cơ bản: Tổng hợp các thông tin thống kê cơ bản như mean, median và quantiles giúp trình bày kết quả phân tích một cách trực quan và dễ hiểu cho người đọc.

Cụ thể sau khi thực hiện lệnh **data.describe()** ta thu được kết quả phân tích thống kê như sau:



Trong đó:

* **count:** Số lượng mẫu (số lượng pokemon) trong tập dữ liệu.
* **mean:** Giá trị trung bình của các cột trong tập dữ liệu.
* **std:** Độ lệch chuẩn của các cột trong tập dữ liệu.
* **min:** giá trị nhỏ nhất của các cột trong tập dữ liệu.
* **25%, 50%, 75%:** Các tứ phân vị của các cột trong tập dữ liệu.
* **max:** Giá trị lớn nhất của các cột trong tập dữ liệu.

Các thống kê mô tả này giúp ta có cái nhìn tổng quan về phân bố và đặc điểm của các biến trong tập dữ liệu “Pokemon Pokedex with Stats”. Chúng ta có thể sử dụng chúng để phân tích và so sánh các biến với nhau, từ đó đưa ra những kết luận và nhận định về dữ liệu.

## **2. PHÂN TÍCH ĐƠN BIẾN**

Phân tích đơn biến là một kỹ thuật phân tích dữ liệu thống kê đơn giản nhưng rất hữu ích trong quá trình khai thác thông tin từ dữ liệu. Nó được sử dụng để tìm hiểu tính chất của một biến số cụ thể, bao gồm các thông tin về phân bố tần suất, trung bình, phương sai, độ lệch chuẩn và các giá trị thống kê khác.

Ví dụ, trong nghiên cứu "Phân loại Pokemon", phân tích đơn biến có thể được sử dụng để tìm hiểu phân loại pokemon huyền thoại trong tập dữ liệu.

Tuy nhiên, phân tích đơn biến chỉ tập trung vào một biến cụ thể mà không xem xét các mối quan hệ giữa các biến khác trong tập dữ liệu. Do đó, trong quá trình phân loại pokemon, các kỹ thuật phân tích đa biến sẽ được sử dụng để khám phá các mối quan hệ giữa biến huyền thoại và các biến khác, như HP, Attack, Defense,... để tạo ra các mô hình phân loại chính xác hơn.

### **2.1. PHÂN TÍCH ĐƠN BIẾN VỚI CÁC BIẾN PHÂN LOẠI**

Các biến phân loại bao gồm: Type 1 và Type 2

Để phân tích các biến phân loại ta có thể sử dụng đồ thị Pie :

Biểu đồ pie (hay còn gọi là biểu đồ tròn) là một loại biểu đồ thể hiện tỷ lệ phần trăm của các thành phần trong một tập dữ liệu. Biểu đồ pie thường được sử dụng để biểu diễn các thành phần của một tập dữ liệu có tổng bằng 100%.

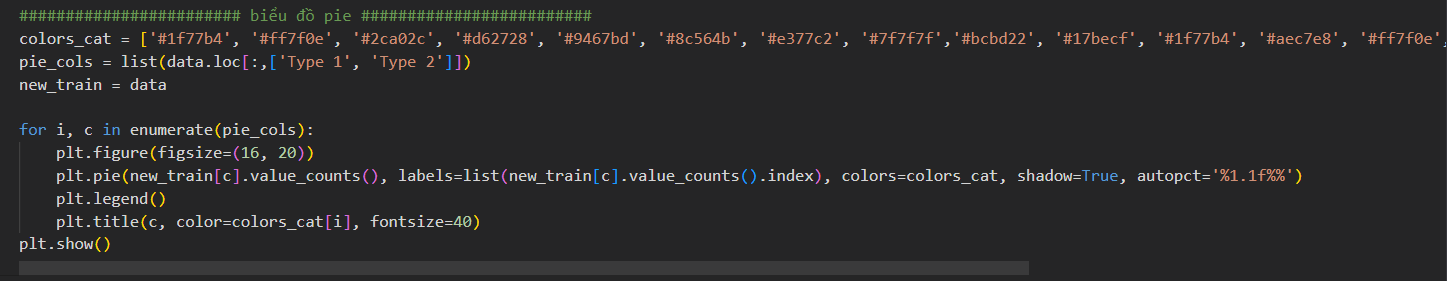
Trong biểu đồ pie, các thành phần được thể hiện dưới dạng các miếng hình tròn, với mỗi miếng biểu thị cho một thành phần của tập dữ liệu. Kích thước của mỗi miếng được xác định bởi tỷ lệ phần trăm của thành phần đó trong tổng số các thành phần của tập dữ liệu.

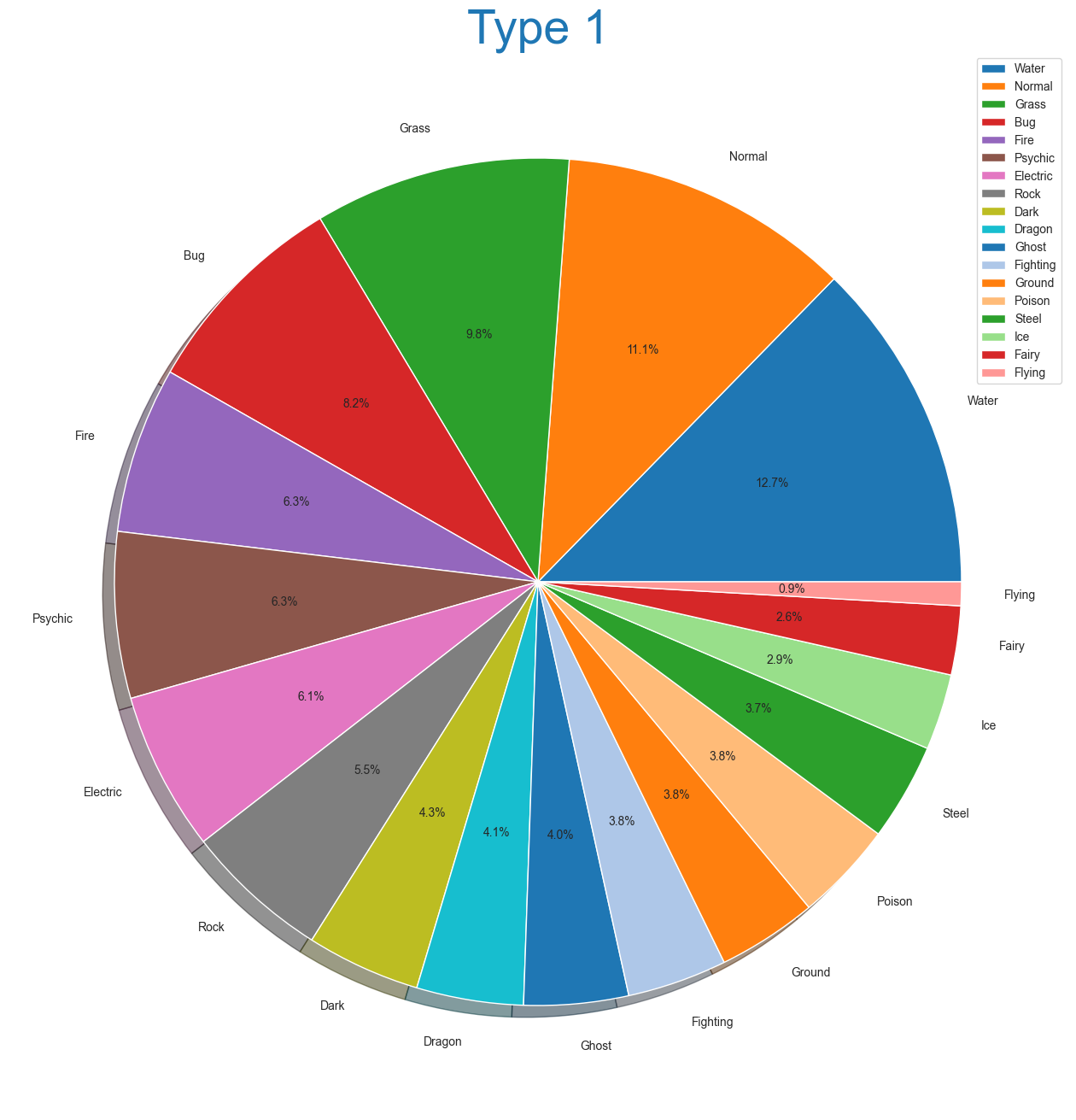
Các thành phần trong biểu đồ pie bao gồm:

1. Miếng: đại diện cho tỷ lệ phần trăm của mỗi thành phần trong tập dữ liệu.
2. Nhãn: là tên hoặc thông tin về mỗi thành phần, thường được đặt ở trung tâm của miếng tương ứng.
3. Chú thích: là một bảng liệt kê tên và giá trị phần trăm của các thành phần, được đặt ở phía dưới hoặc bên cạnh biểu đồ.

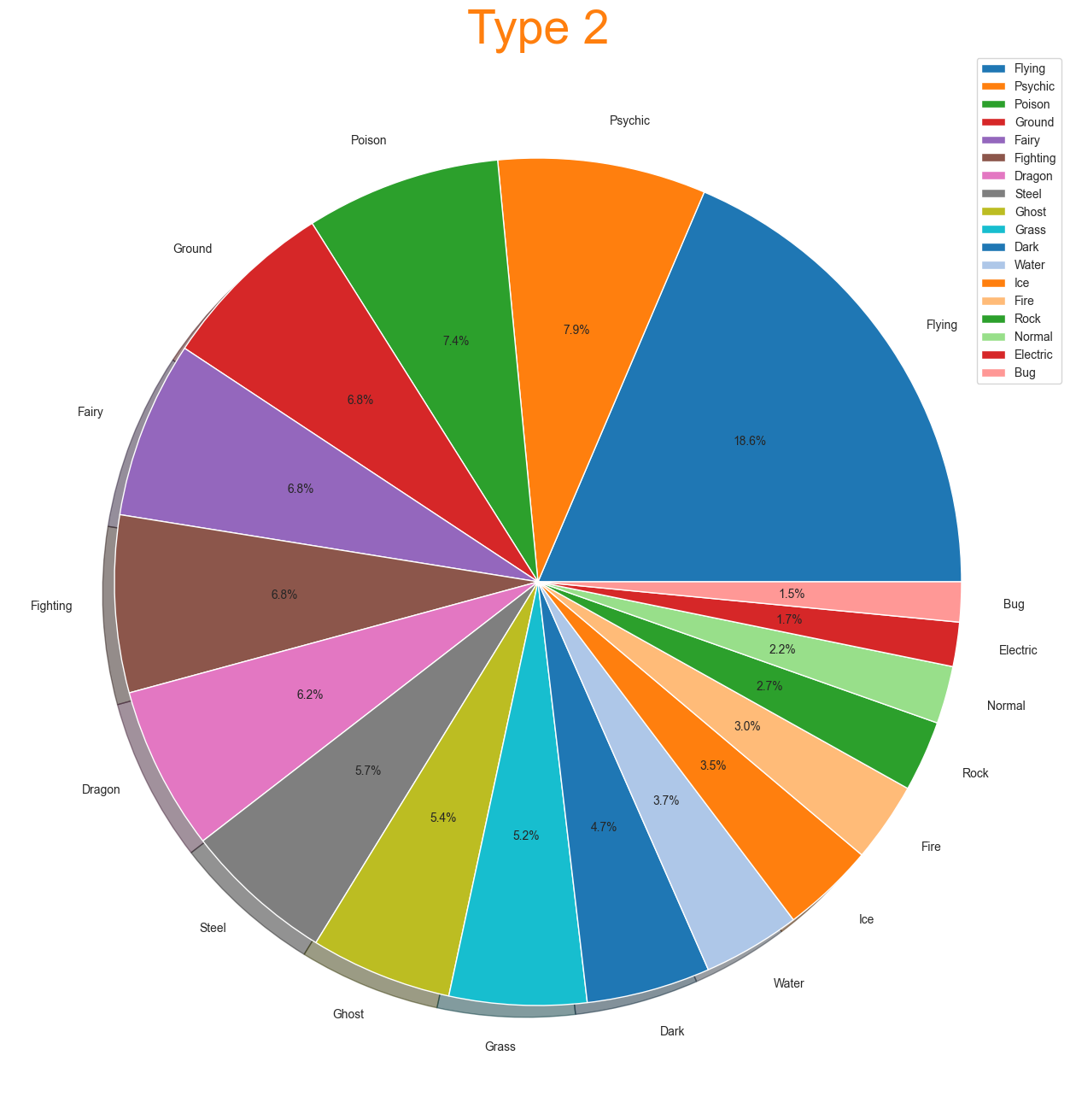
Việc sử dụng biểu đồ pie giúp cho người đọc có thể dễ dàng nhận ra các tỷ lệ phần trăm giữa các thành phần trong tập dữ liệu, từ đó giúp cho việc so sánh và phân tích dữ liệu trở nên dễ dàng và nhanh chóng hơn.

Cụ thể ta dùng câu lệnh dưới đây để tạo biểu đồ Pie cho từng biến của DataFrame “Pokemon Pokedex with Stats”:

****

****

Biểu đồ cho thấy phần lớn pokemon là hệ nước, thường, cỏ, bọ và thấp nhất là hệ bay chỉ chiếm 0.9%

****

Biểu đồ cho thấy phần lớn pokemon có hệ thứ 2 là hệ bay, và hệ thứ 2 ít phổ biến nhất là hệ bọ

* **Qua 2 biểu đồ trên ta nhận xét được rằng đa phần pokemon có hệ chủ đạo không thể bay nên phần lớn hệ thứ 2 được bổ sung vào là hệ bay.**

### **2.2. PHÂN TÍCH ĐƠN BIẾN VỚI CÁC BIẾN SỐ**

KDEplot là viết tắt của "Kernel Density Estimate Plot". Đây là một loại đồ thị biểu diễn mật độ xác suất của một biến ngẫu nhiên. KDEplot thường được sử dụng trong khai phá dữ liệu để hiển thị mức độ phân bố của dữ liệu.

Các thành phần của đồ thị KDEplot bao gồm:

* Trục x: Biến đại diện cho giá trị của dữ liệu.
* Trục y: Biến đại diện cho mật độ xác suất của dữ liệu.
* Đường cong: Biểu diễn mật độ xác suất của dữ liệu. Đường cong này thường có hình dạng trơn và liên tục.
* Màu sắc: Thường được sử dụng để đại diện cho mức độ phân bố của dữ liệu. Các màu sắc khác nhau thường được sử dụng để đại diện cho các giá trị khác nhau của biến.

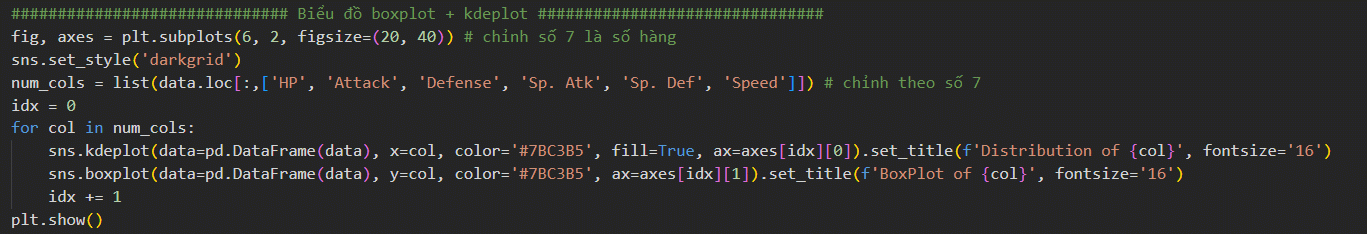
**BoxPlot:**

Boxplot là một loại đồ thị thường được sử dụng trong khai phá dữ liệu để biểu diễn phân phối của một tập dữ liệu và các giá trị ngoại lai.

Các thành phần của đồ thị Boxplot bao gồm:

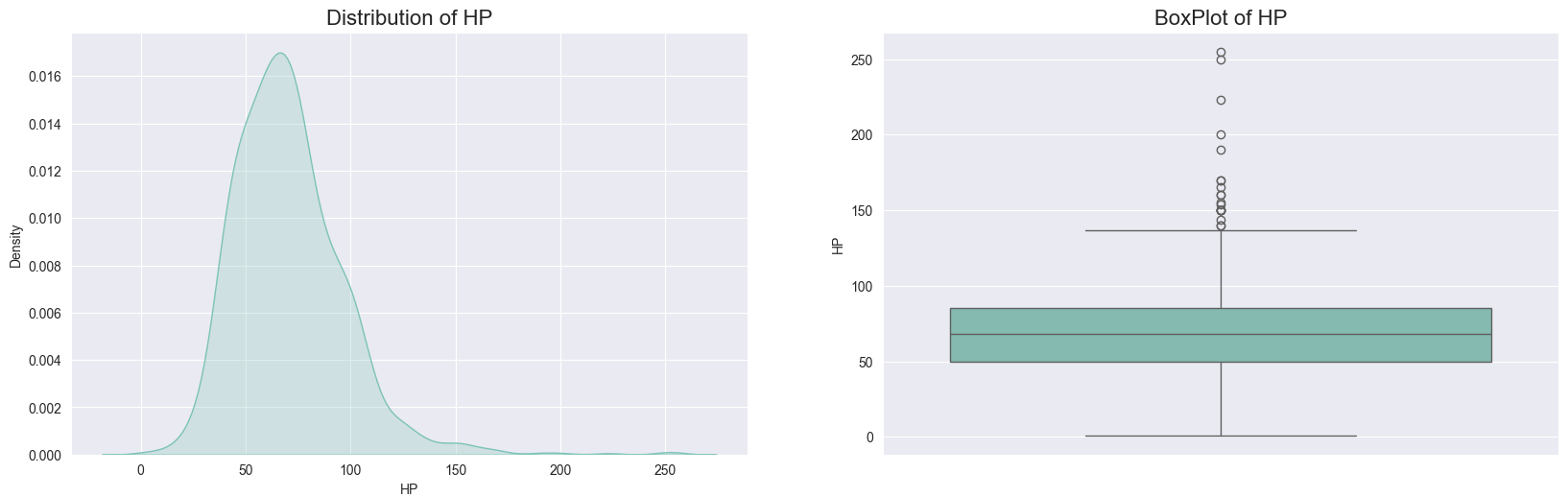
* Hộp (box): Biểu diễn phạm vi của các giá trị trong dữ liệu. Đường kẻ ngang bên trong hộp là giá trị trung vị (median), hai đường kẻ dọc phía trên và phía dưới hộp biểu diễn các phân vị thứ 25 và thứ 75.
* Đường thẳng nằm ngang (whiskers): Biểu diễn khoảng cách từ giá trị tối thiểu và giá trị tối đa trong dữ liệu đến biên của hộp. Đôi khi, chỉ các giá trị trong khoảng whiskers được coi là dữ liệu bình thường.
* Điểm ngoại lai (outliers): Các giá trị nằm ngoài khoảng whiskers được coi là giá trị ngoại lai và được biểu diễn dưới dạng các điểm trên đồ thị.

Cụ thể ta dùng các câu lệnh dưới đây để tạo biểu đồ KDE Plot và BoxPlot cho từng biến của DataFrame “Pokemon Pokedex with Stats”:



Sau khi thực hiện đoạn code trên ta thu được các biểu đồ KDE Plot và Box Plot dưới đây:

**HP**



**Đồ thị KDE PLot**

            Biểu đồ KDE Plot cho thấy pokemon có HP từ 50 – 70 có tỷ trọng xuất hiện nhiều nhất.

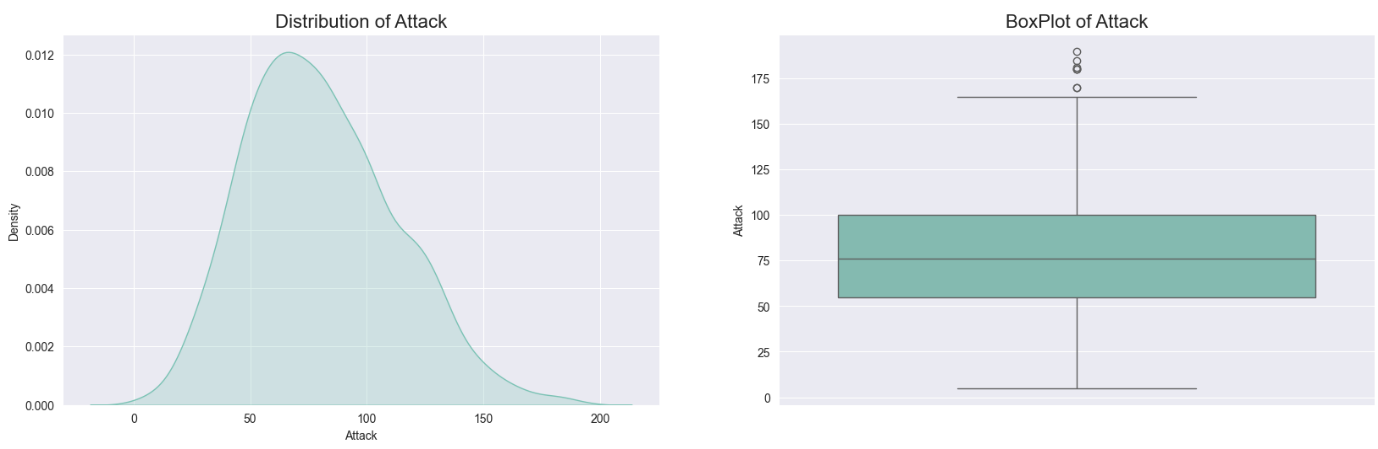
**Đồ thị Box Plot**

●       Giá trị trung vị (median): 70.655

●       Phạm vi giá trị (range): khoảng cách giữa giá trị nhỏ nhất và lớn nhất là khoảng 1 đến 255

●       Hộp (box) của biểu đồ bao phủ khoảng từ 50 đến 85 tương đương khoảng giá trị từ Q1 đến Q3 (tức là 25% đến 75% của tập dữ liệu)

**Attack**



**Đồ thị KDE PLot**

            Biểu đồ KDE Plot cho thấy pokemon có Attack từ 50 – 80 có tỷ trọng xuất hiện nhiều nhất.

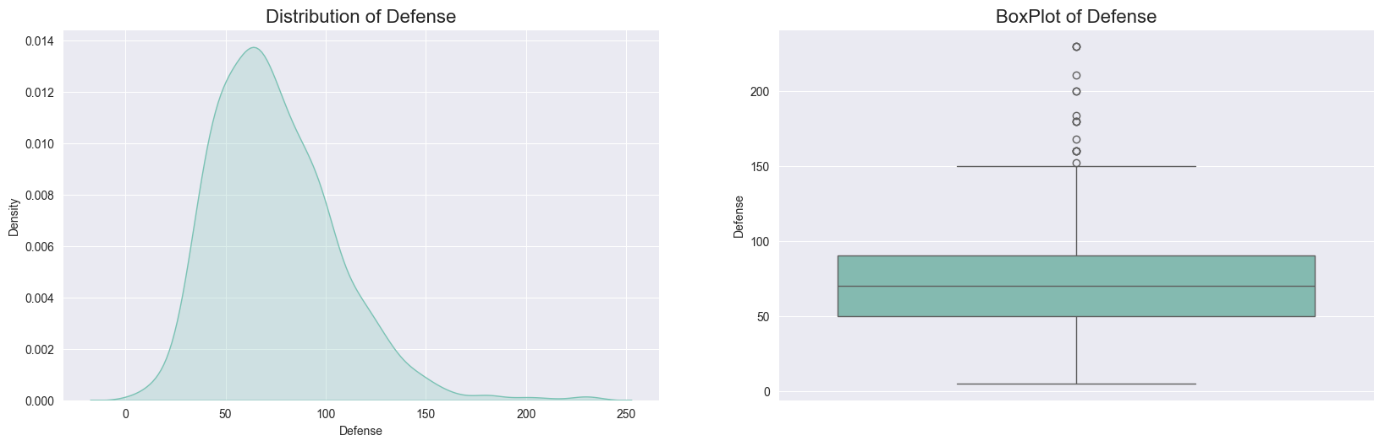
**Đồ thị Box Plot**

●       Giá trị trung vị (median): 80.22

●       Phạm vi giá trị (range): khoảng cách giữa giá trị nhỏ nhất và lớn nhất là khoảng 5 đến 190.

●       Hộp (box) của biểu đồ bao phủ khoảng từ 55 đến 100 tương đương khoảng giá trị từ Q1 đến Q3 (tức là 25% đến 75% của tập dữ liệu)

**Defense**



**Đồ thị KDE PLot**

            Biểu đồ KDE Plot cho thấy pokemon có Defense từ 50 – 75 có tỷ trọng xuất hiện nhiều nhất.

**Đồ thị Box Plot**

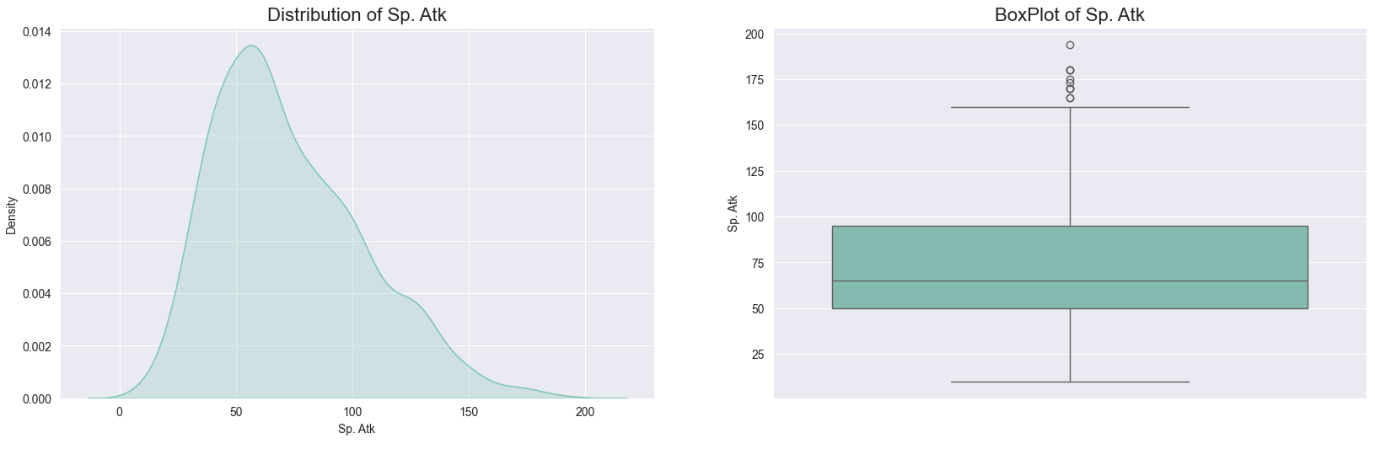
●       Giá trị trung vị (median): 74.5

●       Phạm vi giá trị (range): khoảng cách giữa giá trị nhỏ nhất và lớn nhất là khoảng 5 đến 230.

●       Hộp (box) của biểu đồ bao phủ khoảng từ 50 đến 90.25 tương đương khoảng giá trị từ Q1 đến Q3 (tức là 25% đến 75% của tập dữ liệu).

●      Có một số ngoại lệ (outliers) nằm ở phía trên của biểu đồ, cho thấy rằng có một vài **pokemon** có phòng thủ cao hơn tương đối so với phần còn lại.

**Sp. Atk**



**Đồ thị KDE PLot**

            Biểu đồ KDE Plot cho thấy pokemon có Sp. Atk từ 40 – 75 có tỷ trọng xuất hiện nhiều nhất.

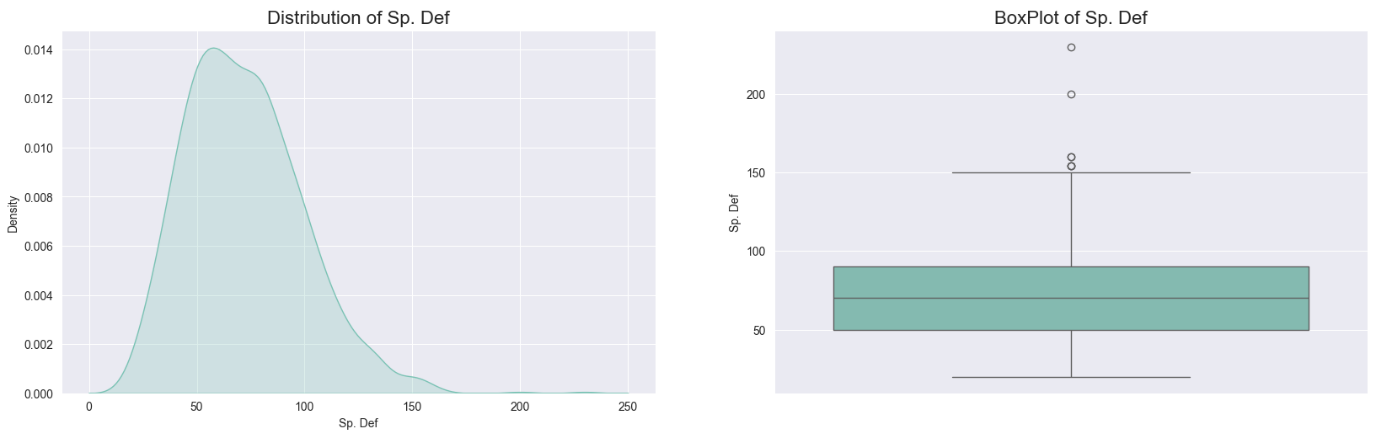
**Đồ thị Box Plot**

●       Giá trị trung vị (median): 72.951

●       Phạm vi giá trị (range): khoảng cách giữa giá trị nhỏ nhất và lớn nhất là khoảng 10 đến 194.

●       Hộp (box) của biểu đồ bao phủ khoảng từ 50 đến 95 tương đương khoảng giá trị từ Q1 đến Q3 (tức là 25% đến 75% của tập dữ liệu).

**Sp. Def**



**Đồ thị KDE PLot**

            Biểu đồ KDE Plot cho thấy pokemon có Sp. Def từ 45 – 90 có tỷ trọng xuất hiện nhiều nhất.

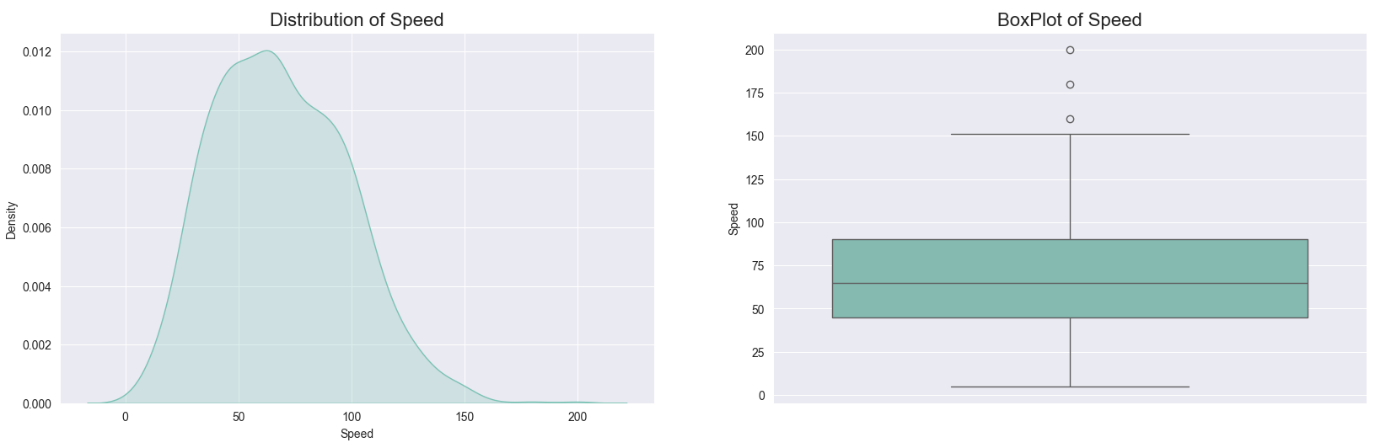
**Đồ thị Box Plot**

●        Giá trị trung vị (median): 72

●       Phạm vi giá trị (range) : khoảng cách giữa giá trị nhỏ nhất và lớn nhất là khoảng 20 đến 230.

●       Hộp (Box) của biểu đồ bao phủ khoảng từ 50 đến 90 tương dương khoảng giá trị từ Q1 đến Q3 ( tức 25% đến 75%)

**Speed**



**Đồ thị KDE PLot**

            Biểu đồ KDE Plot cho thấy pokemon có Speed từ 30 – 95 có tỷ trọng xuất hiện nhiều nhất.

**Đồ thị Box Plot**

●       Giá trị trung vị (median): 65.994

●       Phạm vi giá trị (range): khoảng cách giữa giá trị nhỏ nhất và lớn nhất là khoảng 5 đến 200.

●       Hộp (box) của biểu đồ bao phủ khoảng từ 50 đến 90 tương đương khoảng giá trị từ Q1 đến Q3 (tức là 25% đến 75% của tập dữ liệu).

## **3. PHÂN TÍCH ĐA BIẾN**

Phân tích đa biến (Multivariate analysis) là một kỹ thuật quan trọng trong khai thác dữ liệu và phân tích thống kê dùng để tìm hiểu mối quan hệ giữa nhiều biến đối với nhau. Trong khai thác dữ liệu, phân tích đa biến được sử dụng để tìm ra các mẫu và quy luật ẩn trong tập dữ liệu lớn. Phân tích đa biến có thể giúp chúng ta xác định các biến quan trọng trong tập dữ liệu, đo lường sự tương quan giữa các biến, và dự đoán kết quả dựa trên các biến đầu vào.

Phân tích đa biến được áp dụng rộng rãi trong nhiều lĩnh vực, chẳng hạn như kinh tế học, y học, khoa học xã hội, học máy và khoa học dữ liệu. Nó cung cấp các công cụ để xác định các mối quan hệ phức tạp giữa các biến và cung cấp thông tin về sự phụ thuộc giữa chúng. Có nhiều phương pháp phân tích đa biến khác nhau được sử dụng trong khai thác dữ liệu, bao gồm phân tích hồi quy tuyến tính, phân tích hồi quy logistic, phân tích nhân tố, phân tích thành phần chính, phân tích định tính, và phân tích chuỗi thời gian. Mỗi phương pháp có ưu điểm và hạn chế riêng, và cần phải được lựa chọn phù hợp với dữ liệu và mục đích nghiên cứu cụ thể.

Phân tích đa biến giúp cải thiện độ chính xác và độ tin cậy của các mô hình dự đoán và cung cấp thông tin quan trọng để giải thích các kết quả. Nó cho phép chúng ta hiểu được cách các biến tương tác với nhau và ảnh hưởng lẫn nhau trong quá trình quyết định.

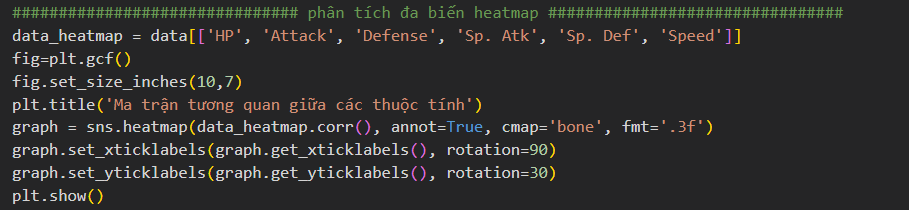
***- Sử dụng heatmap***

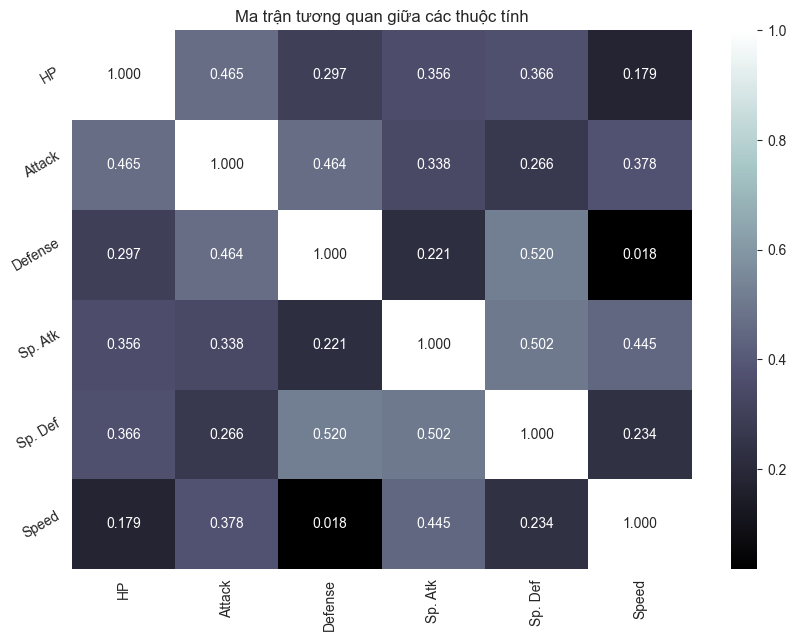
Biểu đồ Heatmap là một công cụ thường được sử dụng để minh họa mối quan hệ giữa các biến trong một tập dữ liệu. Trong việc khai thác dữ liệu, heatmap thường được áp dụng để hình dung sự tương quan giữa các biến và phát hiện các mẫu tiềm ẩn trong dữ liệu. Thông thường, heatmap được tạo ra từ một bảng dữ liệu, trong đó các cột tương ứng với các biến và các hàng tương ứng với các quan sát. Mỗi ô trong bảng dữ liệu được sắp xếp màu sắc dựa trên giá trị của nó, với màu sáng hoặc tối biểu thị mức độ tương quan giữa hai biến. Thông thường, các giá trị lớn hơn thường được biểu diễn bằng màu sáng, trong khi các giá trị nhỏ hơn thường được biểu diễn bằng màu tối.

Heatmap là một công cụ hữu ích trong việc phân tích dữ liệu, đặc biệt là khi xử lý dữ liệu lớn. Nó giúp chúng ta dễ dàng nhận biết các mẫu và quy luật tiềm ẩn trong dữ liệu, cũng như xác định các biến quan trọng. Ngoài ra, heatmap cũng giúp chúng ta phát hiện các giá trị bất thường hoặc lỗi trong dữ liệu, nhờ sự tương phản màu sắc của chúng.

Ngoài ra, biểu đồ heatmap cũng có thể được sử dụng để trực quan hóa dữ liệu cho mục đích trình bày và thuyết trình. Trong việc trình bày dữ liệu, biểu đồ heatmap giúp chúng ta dễ dàng hiểu và trình bày dữ liệu một cách rõ ràng và thú vị hơn.

Sau đây chúng ta sẽ tiến hành vẽ đồ thị heatmap để phân tích sự tương quan giữa các thuộc tính trong dataframe “Pokemon Pokedex with Stats”.



****

Biểu đồ này cho ta thấy tính tương quan giữa các thuộc tính của bộ dữ liệu “Pokemon Pokedex with Stats”. Trong đó mỗi giá trị trong bảng dữ liệu được đại diện bằng một màu sắc, điều này giúp người dùng dễ dàng phát hiện ra các mối quan hệ và xu hướng giữa các biến dữ liệu khác nhau. Đồ thị **heatmap** cho phép người dùng xác định các cặp biến có tính tương quan cao nhất và thấp nhất, tương quan dương và tương quan âm. Trong đồ thị **heatmap** này tính tương quan dương cao nhất được biểu thị màu trắng, giá trị tương quan dương cao vừa có màu xám nhạt , giá trị tương quan dương trung bình màu xám đậm và giá trị tương quan dương thấp hoặc âm có màu đen.

Cụ thể trong bài này ta xét trị tuyệt đối độ tương quan của chúng nằm trong khoảng:

* **0.75 – 1:** Độ tương quan rất cao.
* **0.5 – 0.75:** Độ tương quan cao.
* **0.25 – 0.5:** Độ tương quan trung bình.
* **0 – 0.25:** Độ tương quan thấp.

Trong đó với những giá trị thấp dưới 0.1 hầu như không co độ tương quan giữa các thuộc tính. Với các giá trị tương quan cao trên 0.9 nếu chúng ta biết trước giá trị của một thuộc tính, ta hầu như có thể suy ra giá trị của thuộc tính còn lại.

# CHƯƠNG V. KHAI PHÁ DỮ LIỆU

## **1. CÁC MÔ HÌNH HỌC MÁY**

### **1.1. K-NEAREST NEIGHBORS (KNN)**

Thuật toán K-Nearest Neighbors (KNN) là một thuật toán phân loại trong lĩnh vực khai phá dữ liệu và học máy. Ý nghĩa của KNN là dự đoán lớp của một điểm dữ liệu mới dựa trên K láng giềng gần nhất trong tập dữ liệu huấn luyện.

Các bước của thuật toán KNN như sau:

Chọn số lượng láng giềng gần nhất (K) mà bạn muốn xem xét.

Đo khoảng cách từ điểm dữ liệu mới đến tất cả các điểm dữ liệu trong tập huấn luyện.

Chọn K điểm có khoảng cách gần nhất từ điểm dữ liệu mới.

Đếm số lượng điểm thuộc từng lớp trong K láng giềng gần nhất.

Gán lớp cho điểm dữ liệu mới bằng cách lựa chọn lớp phổ biến nhất trong K láng giềng gần nhất.

Ý nghĩa của thuật toán KNN là tìm hiểu cách dữ liệu được phân bố và dự đoán lớp của các điểm dữ liệu mới dựa trên sự giống nhau với các láng giềng gần nhất. KNN có sự đơn giản và linh hoạt trong việc áp dụng cho nhiều bài toán khác nhau và không yêu cầu quá trình huấn luyện phức tạp. Tuy nhiên, nó cũng có nhược điểm là tính toán khoảng cách và yêu cầu lưu trữ toàn bộ tập dữ liệu huấn luyện.

Đặc điểm quan trọng của thuật toán K-Nearest Neighbors (KNN) bao gồm:

Đơn giản và dễ hiểu: KNN là một thuật toán đơn giản và dễ hiểu trong lĩnh vực học máy. Nó không yêu cầu quá trình huấn luyện phức tạp và không cần xây dựng mô hình dự đoán cụ thể.

Linh hoạt và áp dụng rộng: KNN có thể áp dụng cho cả bài toán phân loại và hồi quy, tùy thuộc vào loại nhãn mục tiêu mà bạn đang dự đoán. Nó cũng có khả năng làm việc với dữ liệu số, dạng văn bản và dữ liệu hạng mục.

Tính toán khoảng cách: KNN sử dụng các phép đo khoảng cách (như khoảng cách Euclid hoặc khoảng cách Mahalanobis) để xác định sự giống nhau giữa các điểm dữ liệu. Việc tính toán khoảng cách là một phần quan trọng của thuật toán.

Số lượng láng giềng (K): Thuật toán KNN sử dụng K láng giềng gần nhất để xác định lớp của điểm dữ liệu mới. Việc chọn giá trị K phù hợp có thể ảnh hưởng đến kết quả dự đoán và độ chính xác của thuật toán.

"Lười" (lazy algorithm): KNN là một thuật toán "lười" vì nó không học một mô hình dự đoán cụ thể trong quá trình huấn luyện. Thay vào đó, nó lưu trữ toàn bộ tập dữ liệu huấn luyện và sử dụng nó trực tiếp trong quá trình dự đoán.

Quản lý dữ liệu: Thuật toán KNN yêu cầu lưu trữ toàn bộ tập dữ liệu huấn luyện trong bộ nhớ để tính toán khoảng cách và xác định lớp của điểm dữ liệu mới. Điều này có thể ảnh hưởng đến hiệu suất và tốc độ tính toán của thuật toán đối với các tập dữ liệu lớn.

Đặc điểm quan trọng này giúp KNN trở thành một công cụ đơn giản và linh hoạt cho việc phân loại và dự đoán dữ liệu dựa trên sự tương đồng giữa các láng giềng gần nhất.

### **1.2. SUPPORT VECTOR MACHINE (SVM)**

Thuật toán Support Vector Machine (SVM) là một thuật toán phân loại và hồi quy trong lĩnh vực khai phá dữ liệu và học máy. Ý nghĩa của SVM là tạo ra một đường ranh giới tối ưu để phân chia các điểm dữ liệu vào các lớp khác nhau.

Đặc điểm quan trọng của SVM bao gồm:

Tìm đường ranh giới tối ưu: SVM tìm kiếm đường ranh giới tối ưu nhất để phân chia các điểm dữ liệu vào các lớp khác nhau. Đường ranh giới này được xác định bằng cách tìm ra siêu phẳng (hyperplane) có khoảng cách xa nhất với các điểm dữ liệu gần nhất của các lớp.

Hỗ trợ vector: SVM sử dụng hỗ trợ vector để xác định các điểm dữ liệu quan trọng trong việc xác định đường ranh giới. Các hỗ trợ vector là các điểm dữ liệu nằm gần nhất với đường ranh giới tối ưu và có vai trò quan trọng trong việc xác định hình dạng và vị trí của đường ranh giới.

Kernel trick: SVM sử dụng kernel trick để biến đổi không gian dữ liệu ban đầu thành một không gian có tính chất tuyến tính. Việc biến đổi này giúp SVM có khả năng xử lý các vấn đề phân loại phi tuyến và tăng độ chính xác của thuật toán.

1. Tính chất margin lớn: SVM tạo ra đường ranh giới có margin lớn nhất, tức là khoảng cách xa nhất giữa đường ranh giới và các điểm dữ liệu gần nhất. Tính chất này giúp giảm thiểu overfitting và cung cấp khả năng tổng quát hóa tốt cho dữ liệu mới.
2. Quản lý dữ liệu: SVM chỉ cần lưu trữ các hỗ trợ vector để xác định đường ranh giới tối ưu, do đó không yêu cầu lưu trữ toàn bộ tập dữ liệu huấn luyện. Điều này giúp SVM phù hợp với việc làm việc với các tập dữ liệu lớn.

SVM là một thuật toán mạnh mẽ và đa dụng có thể áp dụng cho các bài toán phân loại và hồi quy. Đường ranh giới tối ưu của SVM giúp xác định một quy luật phân chia dữ liệu một cách tốt nhất và độ chính xác của thuật toán.

### **1.3. RANDOM FOREST (RF)**

Random forest là một thuật toán machine learning được sử dụng cho bài toán phân loại và dự đoán. Là một phương pháp tập hợp (ensemble method), random forest kết hợp nhiều cây quyết định để đưa ra dự đoán cuối cùng.

Random forest bắt đầu bằng việc tạo ra một tập dữ liệu con ngẫu nhiên từ dữ liệu ban đầu (Bootstrap aggregating hay còn gọi là bagging). Sau đó, nó sẽ xây dựng nhiều cây quyết định (decision tree) bằng cách chọn ngẫu nhiên ra một tập con của các đặc trưng (features) của tập dữ liệu, và chỉ tính toán kết quả trên các tập con này. Kết quả dự đoán được tính bằng cách tính trung bình hoặc phụ thuộc vào mục tiêu của bài toán.

Điều đáng chú ý là, random forest có khả năng giảm thiểu overfitting (tạm dịch là quá khớp) vì nó sử dụng nhiều cây quyết định cùng phân loại, tránh được sự chênh lệch trong dữ liệu. Bên cạnh đó, random forest cũng có thể ứng dụng được cho các tập dữ liệu có kích thước lớn và số lượng đặc trưng cao.

Tuy nhiên, random forest cũng có một số nhược điểm như tốn thời gian và tài nguyên để huấn luyện mô hình, đặc biệt là khi tập dữ liệu càng lớn, số lượng cây quyết định cần được tạo ra càng nhiều. Ngoài ra, một số trường hợp có thể gặp phải hiện tượng bias khi các cây quyết định quá giống nhau.

### **1.4. K-MEANS CLUSTERING**

K-Means Clustering là một thuật toán phân cụm phổ biến trong Machine Learning và Data Mining. Mục tiêu của thuật toán là phân nhóm dữ liệu không gắn nhãn thành các cụm, sao cho các điểm trong mỗi cụm có tính chất tương tự và khác biệt so với các cụm khác.

Thuật toán hoạt động theo cách lặp đi lặp lại. Ban đầu, nó chọn ngẫu nhiên K điểm dữ liệu làm điểm trung tâm ban đầu cho các cụm. Sau đó, nó lặp lại hai bước: gán mỗi điểm dữ liệu vào cụm gần nhất dựa trên khoảng cách và cập nhật lại điểm trung tâm của các cụm dựa trên các điểm dữ liệu mới được gán vào đó. Quá trình này tiếp tục cho đến khi không có sự thay đổi đáng kể nào trong các điểm trung tâm hoặc một số lượng tối đa vòng lặp đã được đạt được.

Một điểm quan trọng của K-Means là hàm mục tiêu, là tổng bình phương khoảng cách giữa mỗi điểm dữ liệu và điểm trung tâm của cụm mà nó thuộc về. Thuật toán cố gắng giảm thiểu hàm mục tiêu này bằng cách tìm cách phân chia dữ liệu thành các cụm có khoảng cách trung bình nhỏ nhất.

Mặc dù K-Means có thể dễ dàng triển khai và hiệu quả với các tập dữ liệu lớn, nhưng nó cũng có nhược điểm. Ví dụ, K-Means dễ bị ảnh hưởng bởi lựa chọn ngẫu nhiên của các điểm tâm ban đầu và không hiệu quả khi dữ liệu không phân tán đều. Nó cũng không phù hợp cho các cụm có kích thước hoặc hình dạng không đồng nhất.

K-Means được ứng dụng rộng rãi trong nhiều lĩnh vực như xử lý ảnh (ví dụ: phân loại màu), phân loại văn bản (nhóm các văn bản có nội dung tương tự lại với nhau), và quảng cáo trực tuyến (nhóm khách hàng dựa trên hành vi truy cập trang web).

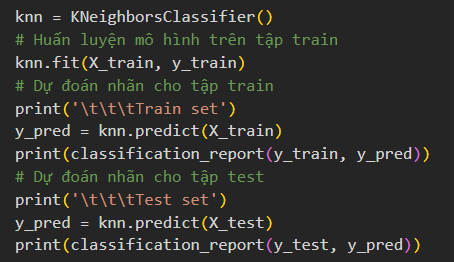
## **2. CÁC CÂU HỎI CẦN GIẢI QUYẾT**

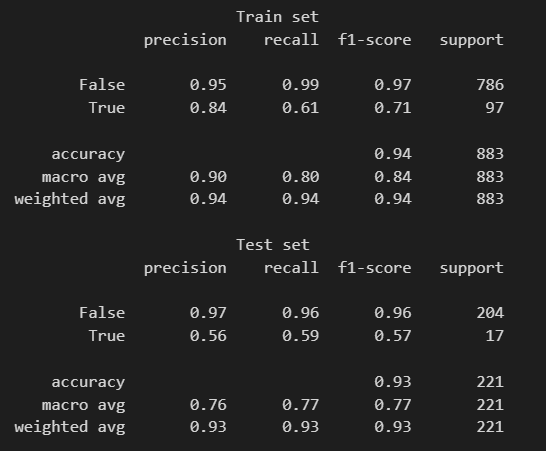
- Câu hỏi 1: phân loại đó có phải pokemon huyền thoại không ?

- Câu hỏi 2: phân cụm pokemon theo tính chất máu, tấn công, phòng thủ, tốc độ ?

# CHƯƠNG VI. ĐÁNH GIÁ VÀ CHỌN THUẬT TOÁN

## **1. MÔ HÌNH KNN**





Dựa vào hình trên ta có thể thấy:

Training set core : 0.94

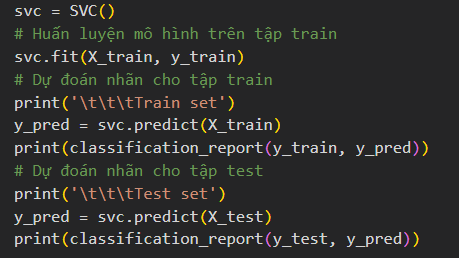
Test set core : 0.93

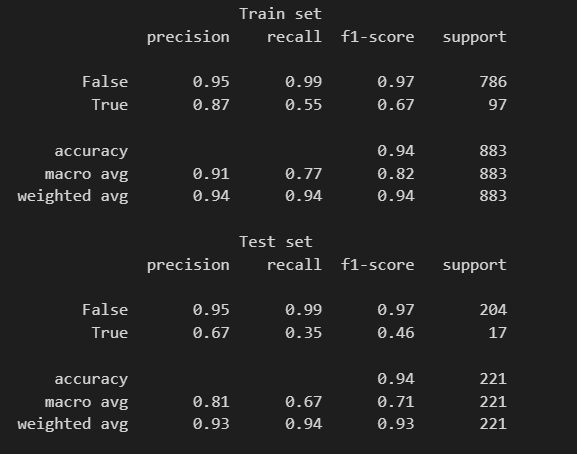
Training set score: Đây là điểm số đánh giá hiệu suất của mô hình trên tập huấn luyện. Giá trị điểm số được cung cấp là 0.94. Điểm số này cho biết mô hình đạt được hiệu suất khá cao trong việc dự đoán nhãn lớp trên tập huấn luyện.

Test set score: Đây là điểm số đánh giá hiệu suất của mô hình trên tập kiểm tra. Giá trị điểm số được cung cấp là 0.93. Điểm số này càng gần 1, cho thấy mô hình có khả năng dự đoán chính xác rất tốt trên tập kiểm tra, tức là mô hình có khả năng tổng quát hóa tốt và hiệu suất cao trên dữ liệu mới.

Điểm số của cả tập huấn luyện và tập kiểm tra đều rất cao, lần lượt là 0.94 và 0.93. Điều này cho thấy mô hình có khả năng học tốt từ dữ liệu huấn luyện và áp dụng kiến thức đã học để dự đoán chính xác trên dữ liệu mới. Sự chênh lệch nhỏ giữa điểm số của tập huấn luyện và tập kiểm tra (0.94 và 0.93) cho thấy mô hình không bị overfitting nghiêm trọng, tức là nó không chỉ "gỡ" nhãn lớp từ dữ liệu huấn luyện mà còn có khả năng tổng quát hóa đối với dữ liệu mới.

## **2. MÔ HÌNH SVC**

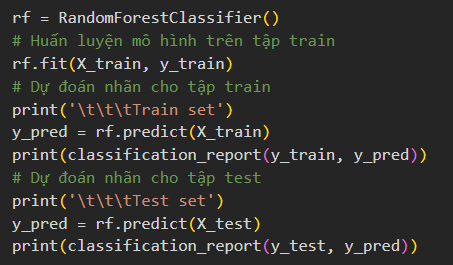


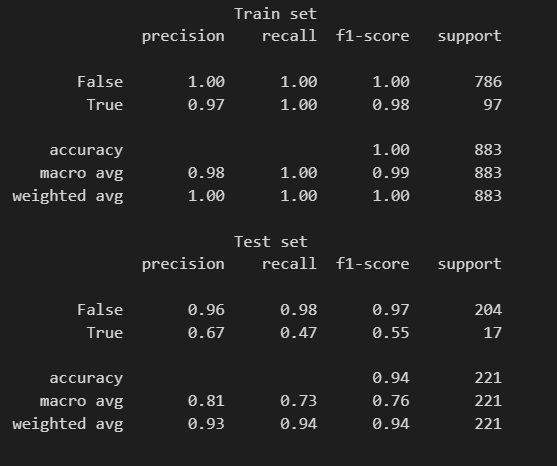


Trong khi áp dụng mô hình này, việc đánh giá dữ liệu đã cho thấy một kết quả ấn tượng khi cả tập huấn luyện và tập kiểm thử đều đạt được độ chính xác cao đến mức 0.94. Điều này không chỉ cho thấy mô hình đã học được từ dữ liệu huấn luyện mà còn cho thấy tính tổng quát và khả năng áp dụng của mô hình trên dữ liệu mới.

Bằng cách đánh giá kết quả này, khả năng có thể tiếp tục cải thiện và tối ưu hóa mô hình để đảm bảo tính đáng tin cậy và ứng dụng thực tiễn của nó trong các ứng dụng thực tế.

## **3. MÔ HÌNH RANDOM FOREST**



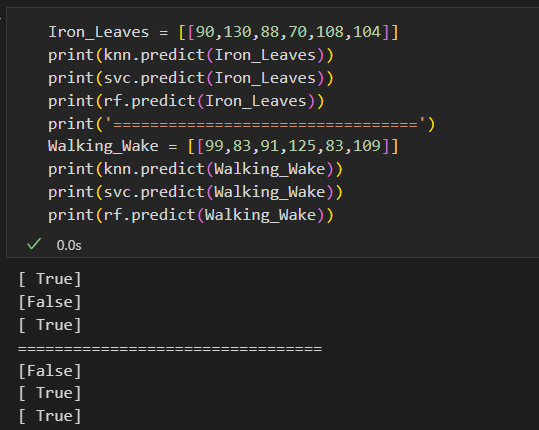


Trong mô hình này, kết quả đánh giá dữ liệu đã cho thấy một hiệu suất rất ấn tượng khi tập huấn luyện đạt độ chính xác 100% khi sử dụng mô hình học máy Random Forest. Điều này đề xuất rằng mô hình đã hoàn toàn học được các mẫu và quy luật trong dữ liệu huấn luyện và có khả năng hoạt động tốt trên dữ liệu mới.

Tuy nhiên, sự hoàn hảo đáng ngạc nhiên này cũng đặt ra một số câu hỏi về tính tổng quát của mô hình và khả năng xử lý dữ liệu nhiễu hoặc không đại diện. Cần phải thận trọng trong việc đánh giá tính ứng dụng thực tế của mô hình trong các tình huống mới, vì có thể mô hình đã "quá tinh chỉnh" để phù hợp hoàn toàn với dữ liệu huấn luyện, dẫn đến hiện tượng overfitting.

Tuy nhiên, sự hoàn hảo của kết quả này cũng đánh dấu một cột mốc quan trọng trong quá trình phát triển mô hình và cung cấp một cơ sở mạnh mẽ cho các nghiên cứu và ứng dụng tương lai. Đồng thời, nó cũng là một lời khẳng định về tiềm năng của phương pháp học máy Random Forest trong việc xử lý dữ liệu và dự đoán.

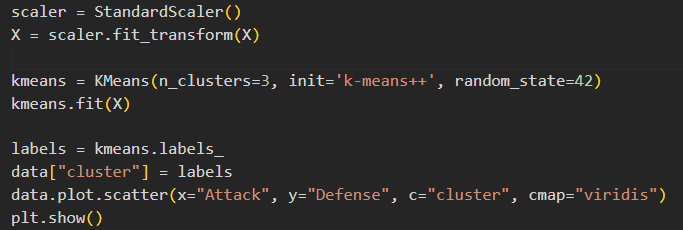
## **4. KIỂM THỬ MÔ HÌNH VÀ TRẢ LỜI CÂU HỎI**

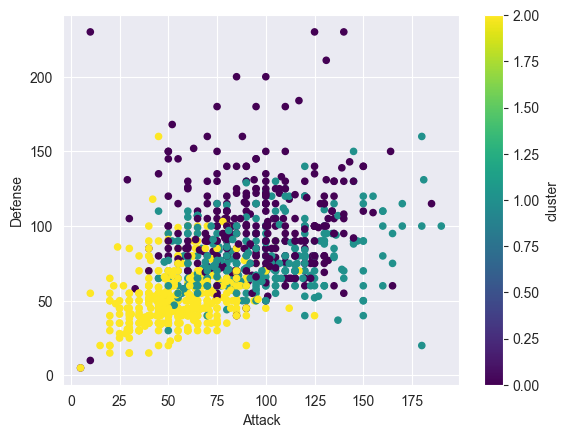
****

- Câu hỏi 1:

Khi áp dụng 3 mô hình trên để phân loại Pokemon huyền thoại thì trong cả 2 loại pokemon huyền thoại được áp dụng test thì mô hình Random Forest đều cho kết quả chính xác.

- Câu hỏi 2: để phân cụm các loại pokemon thiên về tấn công hoặc phòng thủ, sử dụng mô hình KMeans





Với những cụm vàng, những loại pokemon đó yếu về tấn công hoặc phòng thủ hoặc yếu cả 2, với những cụm xanh thì đó là những pokemon có thể sử dụng, còn đối với màu tím thì đó là các loại pokemon có mức công thủ cân bằng toàn diện.

# CHƯƠNG VII. KẾT QUẢ VÀ THẢO LUẬN

Trong báo cáo này, đã sử dụng bộ dữ liệu Pokémon để áp dụng các mô hình học máy trong Python nhằm mục đích phân loại các Pokémon dựa trên các đặc điểm cơ bản như loại, thuộc tính và các chỉ số như HP, tấn công, phòng thủ, tốc độ và nhiều hơn nữa. Trước khi huấn luyện mô hình, quá trình tiền xử lý dữ liệu là bước quan trọng đảm bảo tính chuẩn xác và hiệu quả của mô hình. Thực hiện các bước tiền xử lý bao gồm xử lý dữ liệu thiếu, chuẩn hóa dữ liệu và mã hóa các biến phân loại.

Sau khi tiền xử lý dữ liệu, tiến hành huấn luyện và đánh giá ba mô hình học máy phổ biến: Random Forest, Support Vector Machine (SVM) và K-Nearest Neighbors (KNN). Kết quả của việc đánh giá mô hình trên tập kiểm tra đã cho thấy Random Forest đạt độ chính xác cao nhất, với train là 100% và test là 94%, theo sau là SVM với 94% và KNN với 93%.

Trong quá trình thảo luận, nhận thấy mô hình Random Forest đã cho kết quả tốt nhất trong số các mô hình được thử nghiệm. Điều này có thể được giải thích bởi tính linh hoạt và khả năng tránh overfitting của phương pháp ensemble này. Tuy nhiên, mô hình SVM cũng cho kết quả tốt và có thể cần điều chỉnh siêu tham số để tối ưu hiệu suất. Kết quả của mô hình KNN cũng khá ổn định, tuy nhiên nó có thể bị ảnh hưởng bởi lựa chọn của số lượng hàng xóm.

Tổng kết lại, trong báo cáo này, đã thực hiện phân loại các Pokémon bằng cách áp dụng các mô hình học máy. Mặc dù đã đạt được kết quả khá tốt, việc thử nghiệm và điều chỉnh các phương pháp tiền xử lý dữ liệu cũng như siêu tham số của mô hình có thể là cách tiếp cận tiếp theo để cải thiện hiệu suất của mô hình.

# CHƯƠNG VIII. KẾT LUẬN

Trong phần kết luận, việc áp dụng các mô hình học máy vào việc phân loại Pokémon đã mang lại những kết quả đáng chú ý. Mặc dù mô hình Random Forest đã cho thấy hiệu suất tốt nhất trong số các mô hình được thử nghiệm, nhưng việc so sánh và đánh giá các mô hình khác cũng là một phần quan trọng trong quá trình này.

Ngoài ra, báo cáo cũng đã đề cập đến một số cải tiến có thể được thực hiện trong tương lai để cải thiện hiệu suất của mô hình. Điều này có thể bao gồm việc thử nghiệm các phương pháp tiền xử lý dữ liệu khác nhau, tối ưu hóa siêu tham số của các mô hình, hoặc thậm chí thử nghiệm các kiến trúc mô hình mới để xem xét khả năng phân loại tốt hơn.

Bên cạnh việc áp dụng mô hình học máy vào việc phân loại Pokémon, còn có nhiều hướng nghiên cứu và phát triển tiềm năng khác có thể được khám phá trong tương lai. Ví dụ, việc sử dụng các mô hình học sâu như mạng nơ-ron sẽ có thể giúp tăng cường khả năng phân loại và dự đoán, đặc biệt là khi có sẵn một lượng lớn dữ liệu về các Pokémon.

Trong tương lai, nếu có thêm dữ liệu hoặc thông tin về các Pokémon, việc mở rộng phạm vi của bài nghiên cứu này có thể mang lại những cái nhìn mới mẻ và sâu sắc hơn về việc phân loại Pokémon bằng các phương pháp học máy. Điều này có thể hỗ trợ cho các nhà nghiên cứu và nhà phát triển trong việc hiểu rõ hơn về đặc điểm và khả năng của từng loại Pokémon, từ đó tạo ra các chiến lược huấn luyện và phát triển mạnh mẽ hơn trong trò chơi Pokémon và các ứng dụng liên quan.

# TÀI LIỆU THAM KHẢO

[1] Data Mining Concepts and Techniques - Fourth Edition - Jiawei Han Jian Pei Hang, 2022

[2]Pokémon Database. "Pokémon Database". (5/2023). Từ: <https://pokemondb.net/>

[3]Arnavvvvv. "Pokémon Pokedex". (5/2023). Từ: <https://www.kaggle.com/datasets/arnavvvvv/pokemon-pokedex/>

[4]OpenAI