ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

BÀI THU HOẠCH GIỮA KỲ

HỌC PHẦN: PHÂN TÍCH DỮ LIỆU

CHỦ ĐỀ: TRỰC QUAN HOÁ DỮ LIỆU (DATA VISUALIZATION)

THÀNH VIÊN:

Lê Minh Phúc – 3119411051-DCT119C1

Nguyễn Công Thọ – 3119411077-DCT119C1

Lê Công Minh - 3119411037-DCT119C2

GVHD: PGS.TS Nguyễn Tuấn Đăng

NĂM HỌC: 2023-2024/HK2

TP. HỒ CHÍ MINH, THÁNG 4 NĂM 2024

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH, SƠ ĐỒ

DANH MỤC CHỮ VIẾT TẮT

KDE	:	Kernel Density Estimation

MỞ ĐẦU

Trực quan hoá dữ liệu là một trong những bước quan trọng trong công việc của một nhà phân tích dữ liệu. Bởi vì trong thực tế thì dữ liệu thô thường dưới dạng bảng biểu hay văn bản có cấu trúc rất phức tạp đôi lúc thậm chí không rõ cấu trúc cụ thể nên việc trực quan hoá sẽ biến dữ liệu thô phức tạp trở thành các hình ảnh sinh động dễ dàng nắm bắt được cấu trúc, quy luật, xu hướng của dữ liệu hơn. Từ đó có thể phân tích, đưa ra những quyết định đúng đắn và nhanh chóng góp phần nâng cao hiệu suất làm việc của cá nhân, doanh nghiệp cũng như lợi thế cạnh tranh tăng lên với khả năng ra quyết định nhanh chóng dựa trên dữ liệu thu được.

Đối với nhà khoa học dữ liệu, kỹ sư dữ liệu, chuyên viên phân tích dữ liệu thì làm việc với ngôn ngữ lập trình Python để tương tác với dữ liệu nói chung cũng như trực quan hoá dữ liệu nói riêng là một kỹ năng không thể thiếu. Ngôn ngữ lập trình Python cũng cung cấp cho người dùng một số thư viện dùng để trực quan hoá dữ liệu. Do dó, nhiệm vụ đầu tiên của nhóm chúng em trong đề tài này là nghiên cứu các thư viện trực quan được Python hỗ trợ.

Ngoài nghiên cứu những lý thuyết trọng tâm của các thư viện trực quan hoá dữ liệu với Python kể trên thì mục đích tiếp theo của nhóm chúng em trong đề tài này chính là thực hiện các tình huống thực nghiệm áp dụng các kiến thức, kỹ thuật đã nghiên cứu vào một phần bộ dữ liệu Youtube trending được cung cấp tại Kaggle (cụ thể là tập tin xu hướng video tại khu vực Canada – Cavideos.csv). Qua đó, nhóm chúng em có thể củng cố kiến thức thông qua các tình huống thực tế.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Định nghĩa trực quan hoá dữ liệu

Trực quan hoá dữ liệu (data visualization) là bước khởi đầu trong việc phân tích dữ liệu nhằm hướng tới việc hiểu và truyền đạt thông tin một cách dễ dàng hơn. Nó trình bày thông tin và dữ liệu dưới dạng hình ảnh đồ hoạ (graphical form) bằng cách sử dụng các [4]yếu tố thị giác (visual elements) như biểu đồ, đồ thị, … . Việc trực quan hoá có thể giúp người phân tích dữ liệu có thể tìm hiểu được quy luật (pattern), xu hướng (trend) cũng như những ngoại lệ (outlier), sự phân bố (distribution) và các mối quan hệ (relationship) của dữ liệu. Trực quan hoá dữ liệu là một phương pháp hiệu quả để xử lý một số lượng lớn bộ dữ liệu.

Trực quan hoá dữ liệu thường được sử dụng trong nhiều lĩnh vực như khoa học dữ liệu, thống kê, kinh doanh, giáo dục và nhiều lĩnh vực khác để giúp người dùng nắm bắt thông tin và hiểu sâu hơn về các xu hướng, mối quan hệ và biến động trong dữ liệu.

1.2. Mục đích của trực quan hoá dữ liệu

Việc trực quan hoá dữ liệu đóng vai trò quan trọng vì khi chúng ta làm tốt việc trực quan hoá dữ liệu sẽ giúp chúng ta loại bỏ tình trạng nhiễu loạn dữ liệu làm nổi bật các thông tin hữu ích và nói lên được dữ liệu mà chúng ta đang có nói lên thông điệp gì. Theo một chuyên gia về trực quan hoá dữ liệu - Edward R. Tufte cho rằng[3] "Cái hay vượt trội của đồ họa là cho người xem thấy được nhiều ý tưởng nhất trong thời gian ngắn nhất, mà lại ít tốn giấy mực nhất, trong không gian nhỏ nhất."

Trong môi trường kinh doanh, tốc độ đưa ra quyết định là yếu tố then chốt. Những câu hỏi và chỉ dấu đã biết sẽ không đòi hỏi quá nhiều thời gian để tìm câu trả lời. Tuy nhiên, khi xử lý những tình huống chưa biết, bạn cần có giải pháp phù hợp để khám phá những dữ liệu có sẵn theo cách ít tốn thời gian hơn.

Vậy trực quan hoá dữ liệu có những mục đích sau:

- Biến đổi dữ liệu phức tạp khó hiểu thành dễ hiểu hơn: Đối với dữ liệu thô trong thực tế thường có dạng là bảng biểu hay văn bản rất phức tạp và khó hiểu. Việc trực quan hoá giúp biến đổi dữ liệu dưới dạng bảng hay văn bản thành hình ảnh, biểu đồ khiến cho người phân tích có thể dễ dàng nắm được quy luật, xu hướng, mô hình cũng như sự tương quan của dữ liệu
- Phát hiện thông tin chi tiết giúp người phân tích dễ dàng đưa ra quyết định: Nhờ trực quan hoá sang dạng hình ảnh, biểu đồ mà người phân tích có thể quan sát được những sự bất thường cũng như các xu hướng tiềm ẩn tồn tại trong dữ liệu. Qua đó, giúp người phân tích có thể đưa ra các quyết định sáng suốt dựa trên dữ liệu nhằm nâng cao hiệu quả kinh doanh của doanh nghiệp cũng như hiệu quả làm việc, tiết kiệm thời gian phân tích hơn.
- Nâng cao hiệu quả truyền đạt thông tin: Thông tin được truyền đạt của dữ liệu sau khi trực quan hoá sẽ dễ hiểu hơn. Bên cạnh đó, thông tin được truyền đạt cũng sẽ trở nên trực quan và sinh động hơn giúp nâng cao khả năng ghi nhớ và thu hút của người xem
- Kích thích khả năng tư duy sáng tạo: Việc trực quan hoá giúp cho người xem có thể quan sát dữ liệu dưới nhiều góc độ. Qua đó kích thích tư duy sáng tạo tìm ra giải pháp mới để giải quyết vấn đề.

1.3. Các kỹ thuật trực quan hoá dữ liệu với ngôn ngữ lập trình python

Để trực quan hoá dữ liệu, ngôn ngữ lập trình Python đã hỗ trợ một số thư viện giúp cho việc trực quan hoá dữ liệu dễ dàng hơn. Một số thư viện trực quan hoá dữ liệu phổ biến là Matplotlib, Seaborn và Bokeh

Đặc điểm chính của Matplotlib, Seaborn và Bokeh

Table 1. Bảng mô tả đặc điểm chính của Matplotlib, Seaborn và Bokeh

Thư viện	Đặc điểm
Matplotlib	[4]Matplotlib là thư viện hỗ trợ của ngôn ngữ Python được dùng để thể hiện việc trực quan hoá dữ liệu phổ biến nhất. Nó là nền tảng của các thư viện trực quan hoá nâng cao như Seaborn. Trong đó nó cung cấp các hàm tiện ích và dễ dàng sử dụng cho việc tạo hình ảnh và biểu đồ
Seaborn	[4]Seaborn là thư viện mã nguồn mở của Python dùng cho việc trực quan hoá và thống kê với độ tương tác và tính thu hút ở mức cao. Seaborn được xây dựng dựa trên nền tảng của Matplotlib và các tính năng đơn giản hơn, dễ hiểu hơn dễ thu hút và tương tác hơn.
Bokeh	[4]Bokeh là một thư viện trực quan hoá có tính tương tác, chất lượng cao, linh hoạt, tập trung và mạnh mẽ hơn cho một khối lượng lớn dữ liệu. Nó cung cấp các biểu đồ, sơ đồ, bố cục tương tác, phong phú, và bảng điều khiển cho các trình duyệt web hiện đại.

1.3.1. Các kỹ thuật trực quan hoá với Matplotlib

Cách cài đặt Matplotlib:

Table 2. Bảng mô tả cách thức cài đặt Matplotlib trên Python và Anaconda

Môi trường	Câu lệnh cài đặt
Python	pip install matplotlib

Python 3	pip3 install matplotlib
Anaconda	conda install matplotlib

Với Matplotlib thì chúng ta cần nạp module pyplot của thư viện Matplotlib thông qua câu lệnh:

import matplotlib.pyplot as plt

Một số loại biểu đồ được hỗ trợ tại Matplotlib

Table 3. Bảng mô tả một số loại biểu đồ hỗ trợ bởi Matplotlib

Loại biểu đồ	Hàm hỗ trợ	Định nghĩa
Biểu đồ đường	plt.plot()[1]	Là dạng biểu đồ biểu diễn dạng đường biểu thị cho 2 biến số. Nó là tập hợp tất cả các điểm dữ liệu cùng nằm trên một đoạn (Xem ví dụ ở hình 2)
Biểu đồ cột	plt.bar()[1]	Là công cụ trực quan hoá dùng để so sánh giá trị của nhiều nhóm. Biểu đồ cột có thể biểu diễn theo chiều dọc (vertical) và chiều ngang (horizontal).(Ở ví dụ tại hình 4, biểu đồ cột được biểu diễn theo chiều ngang gọi là horizontal bar plot)
Biểu đồ tròn	plt.pie()[1]	Là dạng biểu đồ có dạng là đường tròn và được chia thành nhiều mảnh phân biệt. Mỗi mảnh tượng trưng cho một tỉ lệ của giá trị mà mình muốn trình bày sao cho tổng giá trị của các mảnh là 100% (Xem ví dụ ở hình 3)

Biểu đồ Scatter	plt.scatter()[1]	Là dạng biểu đồ cho phép vẽ các điểm dữ liệu bằng cách sử dụng hệ trục toạ độ Descartes (Cartesian coordinates) để biểu diễn các số liệu (Xem ví dụ ở hình 1)
Biểu đồ histogram	plt,hist()[1]	Là dạng biểu đồ mô tả sự phân bố của các biến số (numeric variable). Việc tạo một biểu đồ histogram thì chúng ta cần sử dụng hàm hist(). Biểu đồ sẽ thể hiện sự phân bố xác suất của một biến liên tục. Tuy nhiên, biểu đồ histogram chỉ thể hiện được 1 biến đơn lẻ trong khi đó biểu đồ cột có thể thể hiện hai biến (Xem ví dụ ở hình 5)

Dưới đây là một số hình ảnh liên quan đến các dạng biểu đồ được tạo bởi thư viện Matplotlib

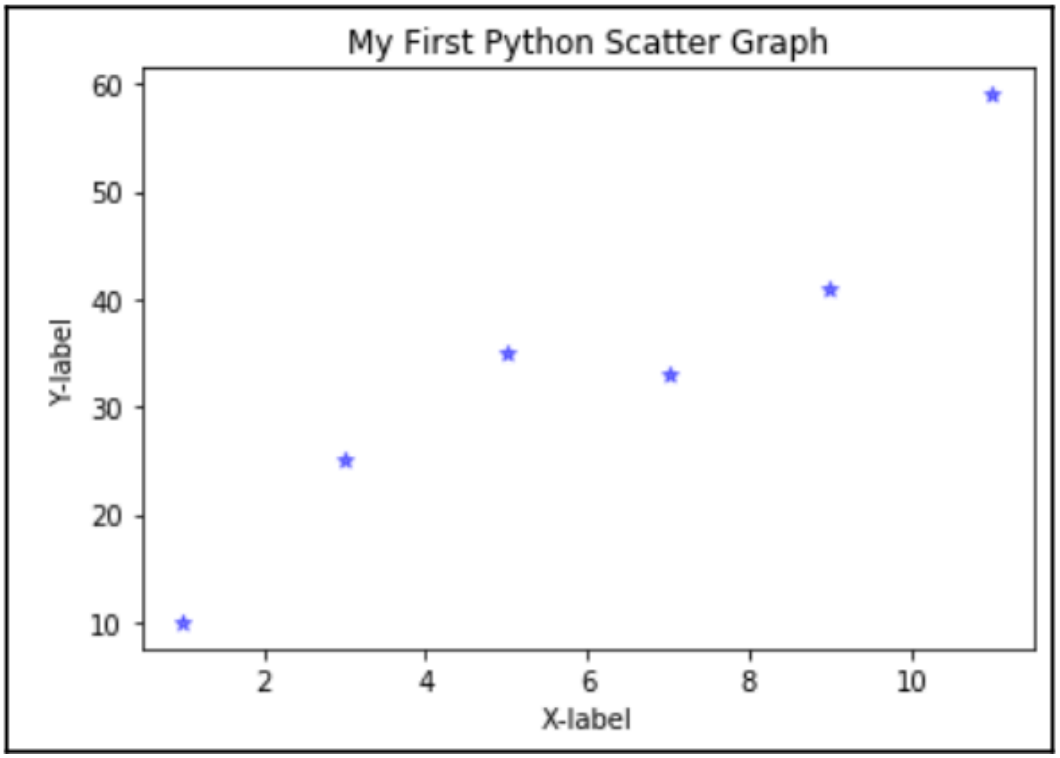


Figure 1. Ví dụ về hình dạng của biểu đồ scatter trên Matplotlib[4]

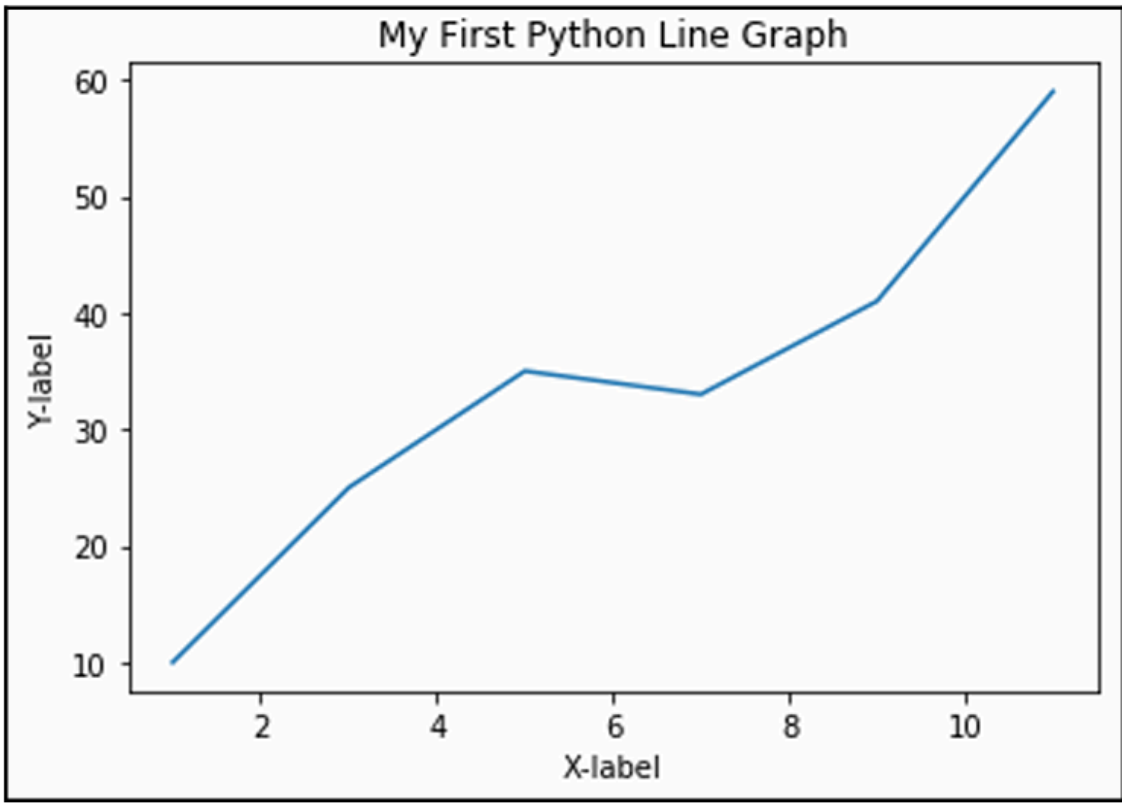


Figure 2. Ví dụ về hình dạng cũa biểu đồ đường trên Matplotlib[4]

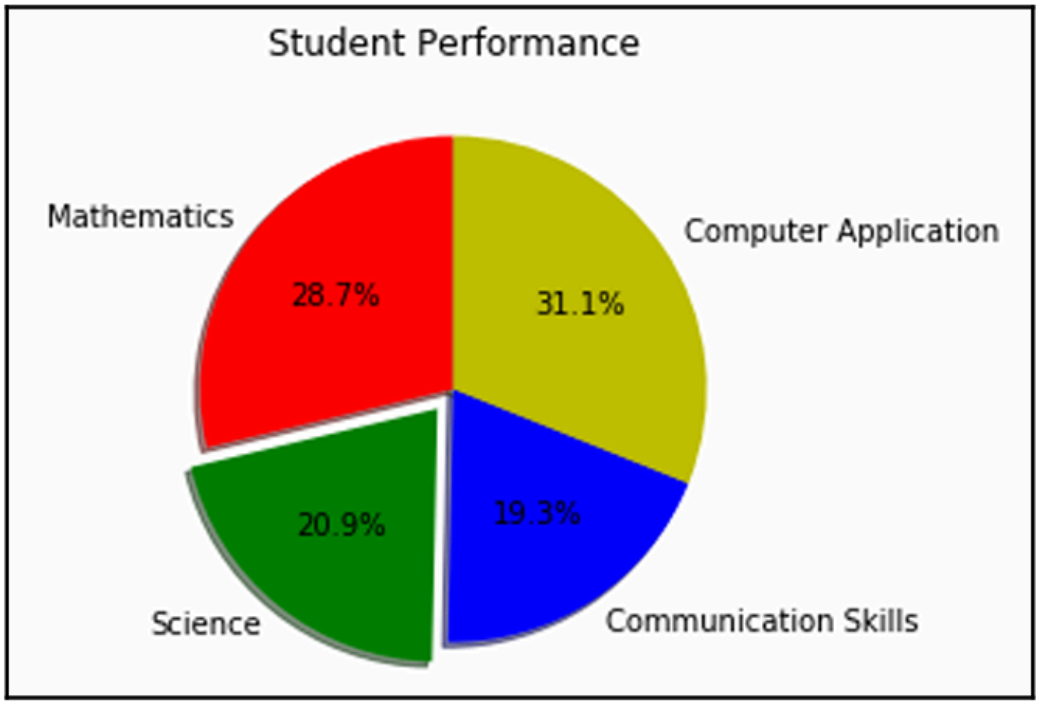


Figure 3. Ví dụ về biểu đồ tròn trên Matplotlib[4]

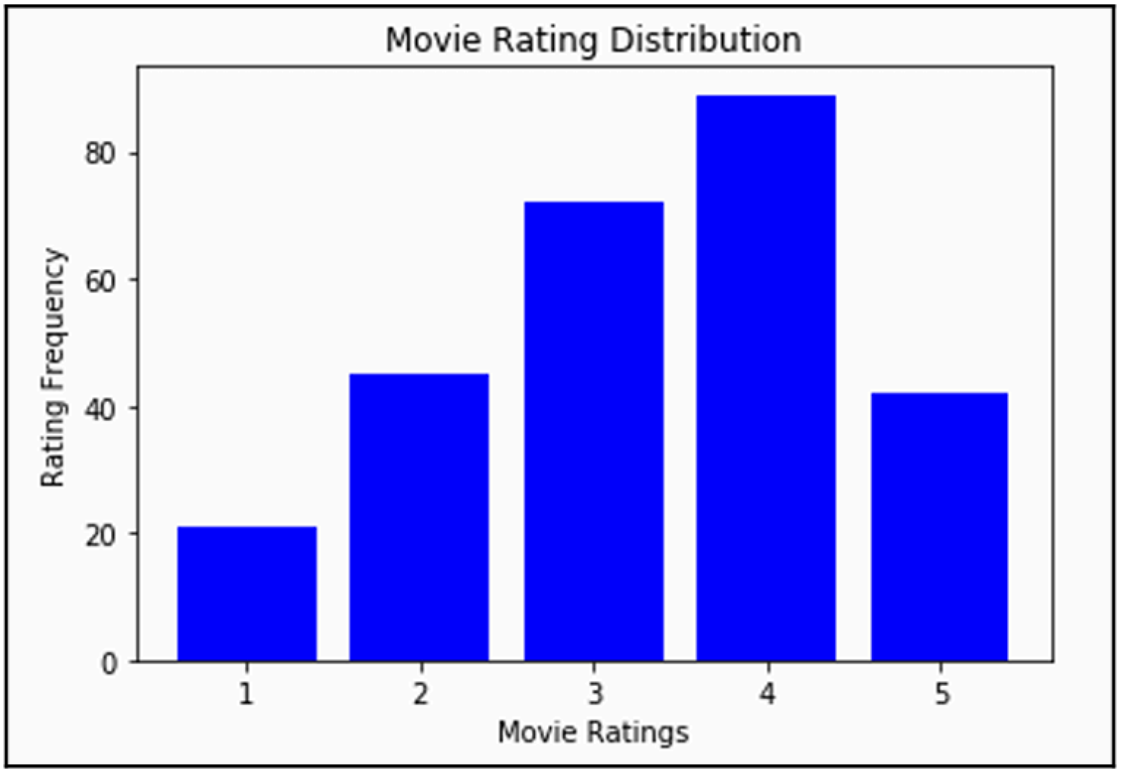


Figure 4. Ví dụ về biểu đồ cột được tạo bởi Matplotlib[4]

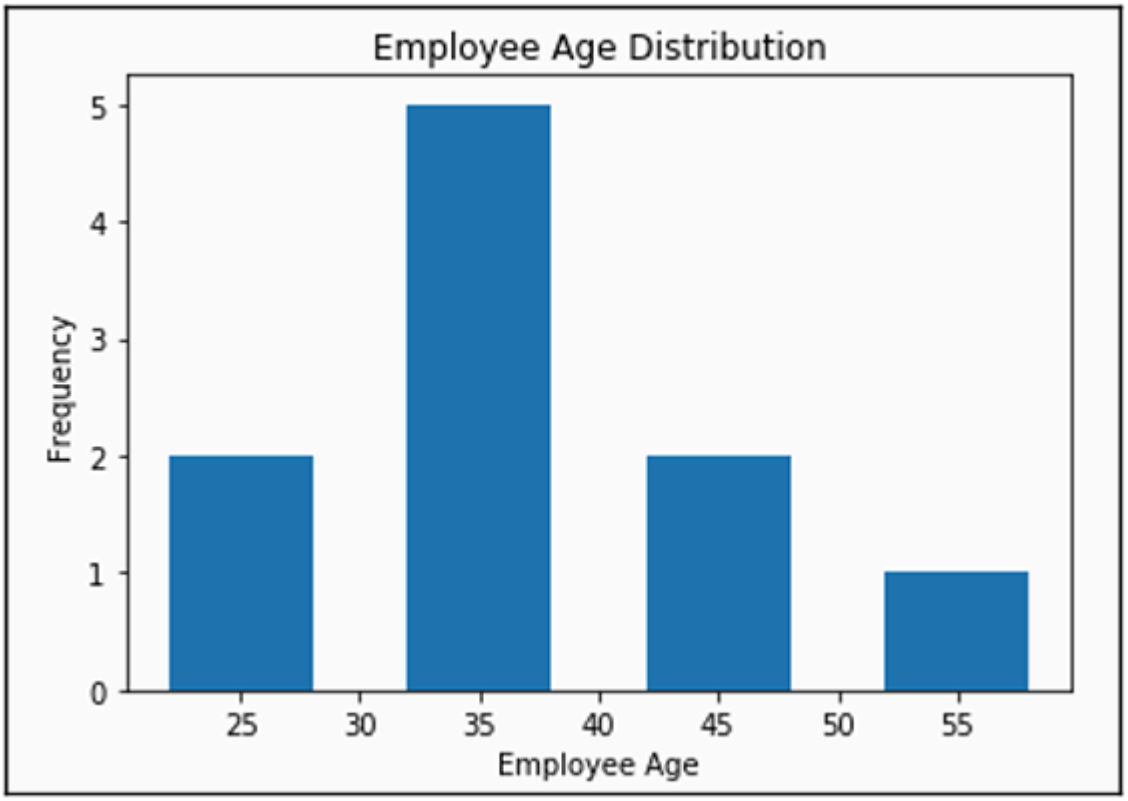


Figure 5. Biểu đồ Histogram được tạo bởi Matplotlib[4]

1.3.2. Các kỹ thuật trực quan hoá với Seaborn

Cách cài đặt Seaborn trên môi trường Python và Anaconda

Table 4. Bảng mô tả cách thức cài đặt thư viện Seaborn trên môi trường Python và Anaconda

Môi trường	Câu lệnh cài đặt
Python	pip install seaborn
Python 3	pip3 install seaborn
Anaconda	conda install seaborn

Để có thể sử dụng được Seaborn thì cần nạp thư viện Seaborn theo câu lệnh sau:

import seaborn as sns

Các đồ thị của Seaborn được chia thành các nhóm sau:

- Biểu đồ quan hệ[2] (Relational plot): Biểu đồ quan hệ được sử dụng trong việc trực quan hoá các quan hệ thống kê. Một hàm phổ biến được sử dụng chính là relplot(). Nó là hàm trực quan hoá thống kê các mối

quan hệ ở mức tập trung vào hình ảnh (figure-level). Hai hướng phổ biến của dạng biểu đồ này là biểu đồ phân bố điểm (scatter plot) và biểu đồ đường (line plot)

- Scatter plot (xem hình 6 để minh hoạ): là dạng mặc định của relplot(). Chúng ta có thể sử dụng scatter plot bằng cách sử dụng hàm scatterplot() hoặc hàm relplot(kind = 'scatter')
- Line plot (xem hình 7 để minh hoạ): Với một số tập dữ liệu, chúng ta muốn hiểu sự thay đổi của một biến theo thời gian, hoặc một biến liên tục tương tự. Trong trường hợp này, lựa chọn tốt là vẽ một biểu đồ đường (biểu đồ theo thời gian). Trong Seaborn có hàm linepolot() để tạo biểu đồ đường. Tuy nhiên, chúng ta vẫn có thể sử dụng hàm relplot(kind = 'line') để tạo biểu đồ đường

- Biểu đồ phân bổ (Distribution plot): Là dạng biểu đồ trực quan hoá sự phân bổ của dữ liệu cũng là bước nên làm để có thể hiểu được dữ liệu được phân bố như thế nào. Qua đó có thể biết được phạm vi của dữ liệu, xu hướng của dữ liệu, độ lệch và các ngoại lai (outliers). Seaborn cũng hỗ trợ một số hàm để trực quan hoá sự phân bổ dữ liệu và chúng được chia thành 2 nhóm:
  - Nhóm tập trung vào hình ảnh (figure-level) gồm: histplot(), kdeplot()
    - Biểu đồ histogram[2] là dạng tiếp cận cơ bản và cũng là mặc định của nhóm biểu đồ phân bổ. Một đồ thị histogram là một đồ thị mô tả sự phân bổ của các biến số trên các ngăn rời rạc giống như một biểu đồ cột. Cách sử dụng biểu đồ histogram là gọi thẳng hàm displot()
    - Biểu đồ [2],[4]KDE (Kernel Density Estimation): Thay vì sử dụng các ngăn rời rạc như histogram, biểu đồ KDE làm trơn tru các quan sát bằng nhân Gaussian, tạo ra ước tính mật độ liên tục. Cách để tạo biểu đồ KDE thì có thể sử dụng hàm displot(kind = 'kde'). Hình 13 là một ví dụ mô tả hình dạng của biểu đồ KDE
  - Nhóm tập trung vào giá trị trên trục số (axis-level): jointplot(), pairplot()
    - Joint plot[2]: là một hình ảnh trực quan nhiều bảng; nó cho thấy mối quan hệ hai biến và phân phối các biến riêng lẻ trong một biểu đồ (xem thêm ví dụ ở hình 14). Cách để tạo một Joint plot thì ta có thể dùng hàm jointplot()

        - Pair plot[4]: Là sự kết hợp giữa một biểu đồ histogram và biểu đồ phân bổ sử dụng biểu đồ scatter. Chúng ta có thể sử dụng hàm pairplot() để tạo một pairplot
- Biểu đồ phân loại[2] (Categorical plot); Là dạng biểu đồ dùng để trực quan hoá dữ liệu phân loại. Các loại biểu đồ phổ biến cho nhóm biểu đồ này gồm
    - Nhóm phân loại theo sự phân bổ dữ liệu: box plot và violin plot
        - Box plot (biểu đồ hộp): [4]là một trong những biểu đồ tốt nhất để hiểu được sự phân bố của mỗi biến với các tứ phân vị của nó. Nó có thể nằm ngang hoặc dọc. Nó cũng cho thấy mức tối thiểu và mức tối đa và các giá trị ngoại lệ trong dữ liệu. Seaborn đã hỗ trợ hàm boxplot() để người dùng có thể tạo một biểu đồ hộp. Một ví dụ về hình dạng của biểu đồ hộp được thể hiện ở hình 10
        - Violin plot (biểu đồ violin): Là biểu đồ có dạng giống như một chiếc đàn violin (xem ví dụ hình 11). [4]Đồng thời nó là dạng kết hợp giữa biểu đồ KDE và biểu đồ hộp, nó giúp cho việc hiểu, phân tích sự phân bổ của dữ liệu trở nên dễ hiểu hơn. Hàm violinplot() được seaborn hỗ trợ người dùng nhằm có thể tạo ra một biểu đô violin.
    - Nhóm phân loại theo ước lượng: biểu đồ cột và biểu đồ đếm
        - Biểu đồ cột:[4] được dùng để xác định mối quan hệ của các biến liên tục và có phân loại. Hàm barplot() của Seaborn cung cấp tính năng tạo một biểu đồ cột. Hình 15 là hình minh hoạ cho một biểu đồ cột với seaborn
        - Biểu đồ đếm:[4] là một dạng biểu đồ cột đặc biệt. Nó cho biết tần suất của từng biến phân loại. Nó còn có thể hiểu theo cách khác là biểu đồ histogram dành cho biến phân loại.[4] So với Matplotlib thì cần gom nhóm cho từng nhóm dữ liệu rồi tính tần suất với Seaborn thì chỉ cần 1 dòng mã lệnh chính là áp dụng hàm countplot() để tạo một biểu đồ đếm.
- Biểu đồ hồi quy (Regression plot): là dạng đồ thị dùng để ước lượng tính phù hợp của mô hình hồi quy [2](estimating regression fit). Các biểu đồ hồi quy trong seaborn chủ yếu nhằm mục đích bổ sung hướng dẫn trực quan giúp nhấn mạnh các mẫu trong tập dữ liệu trong quá trình phân tích dữ liệu thăm dò. Điều đó có nghĩa là bản thân seaborn không

phải là một gói phân tích thống kê. Hàm regplot() và hàm lmplot() là các hàm được seaborn hỗ trợ trong việc tạo biểu đồ hồi quy. Hình 8 mô tả hình dạng của biểu đồ hồi quy được tạo bằng hàm lmplot()

- Heatmap: [4]cung cấp biểu diễn lưới hai chiều. Ô riêng lẻ của lưới chứa một giá trị của ma trận. Chức năng bản đồ nhiệt cũng cung cấp chú thích trên mỗi ô. Để có thể tạo một heatmap thì seaborn đã cung cấp hàm heatmap với đầu vào là một ma trận dữ liệu. Hình dạng của heatmap có thể quan sát ở ví dụ hình 12.

Dưới đây là một số hình ảnh liên quan đến các dạng biểu đồ được tạo bởi thư viện Seaborn:

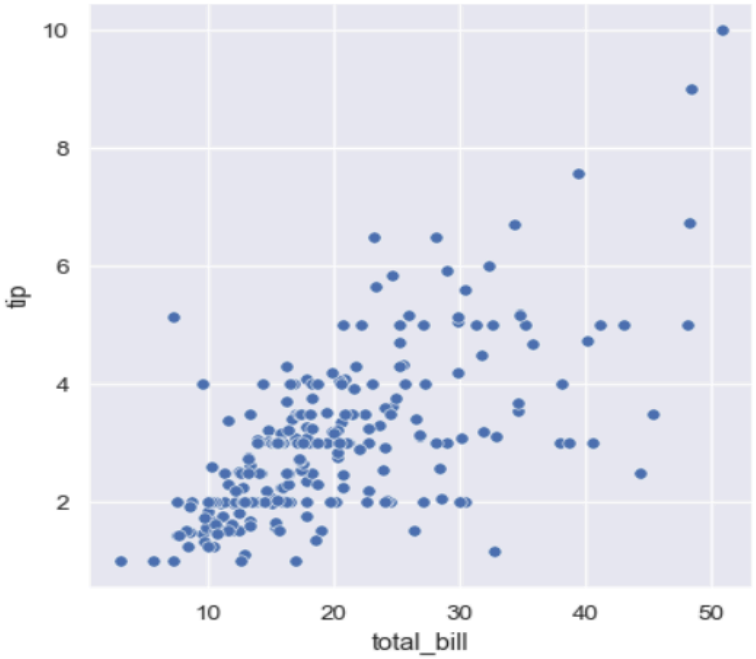


Figure 6. Hình ảnh minh hoạ biểu đồ scatter được tạo bởi Seaborn

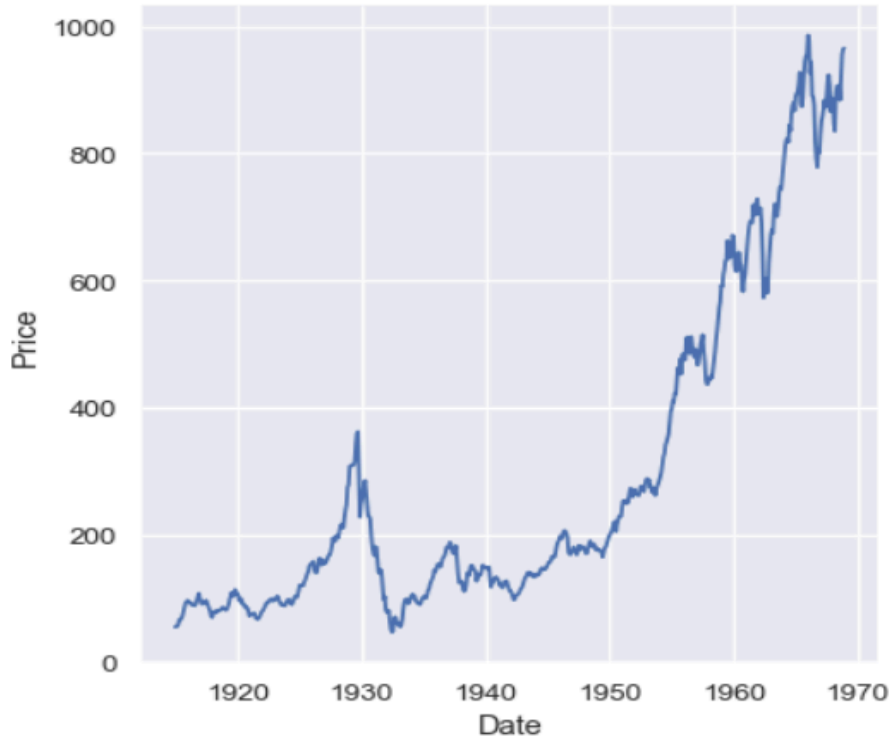


Figure 7. Hình ảnh mô tả biểu đồ đường được tạo bởi Seaborn

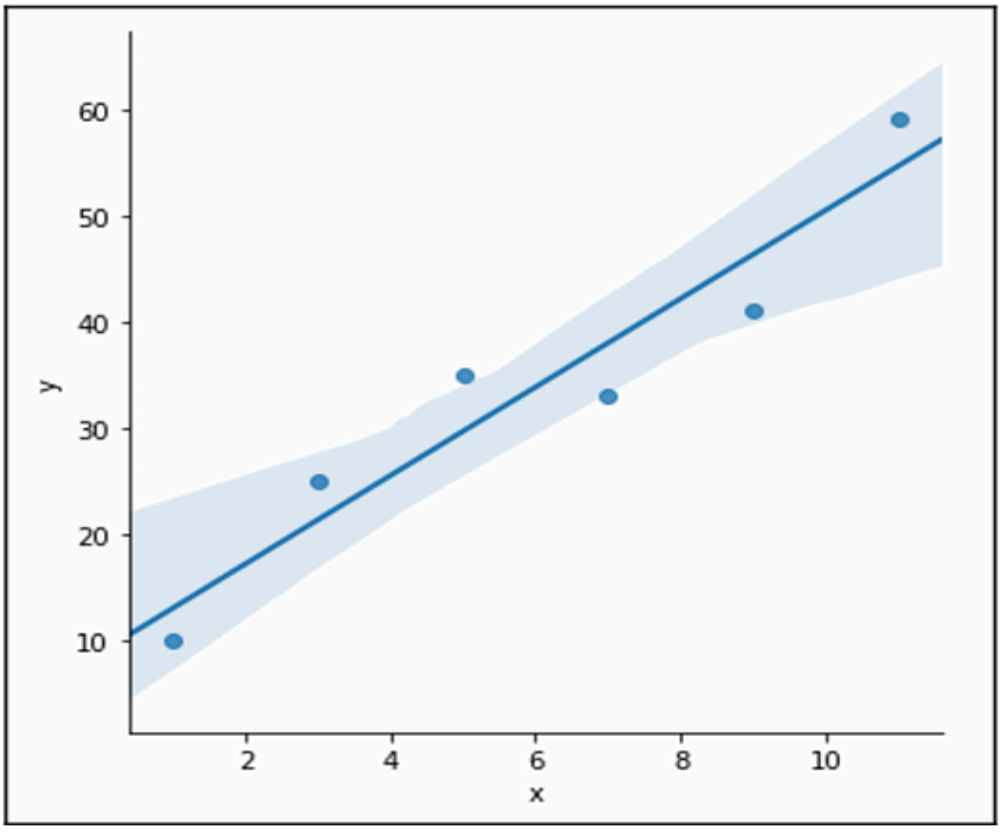


Figure 8. Hình ảnh biểu đồ lmplot được tạo bởi Seaborn

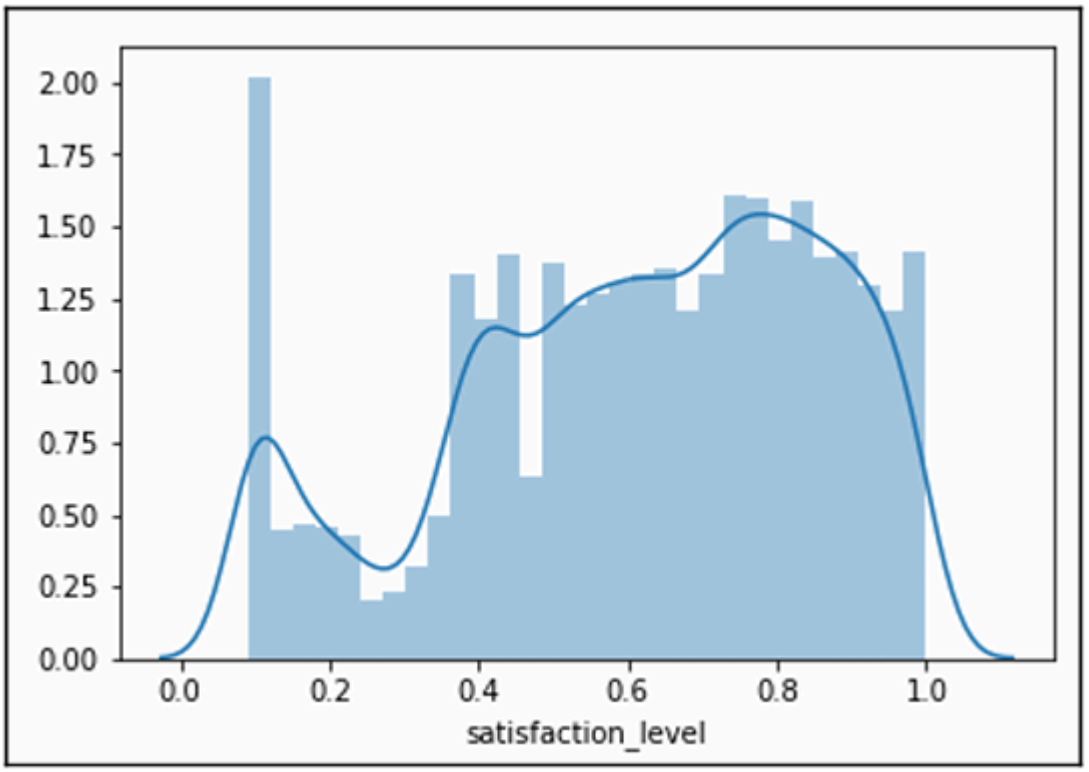


Figure 9. Hình ảnh biểu đồ phân bố (displot) được tạo bởi Seaborn

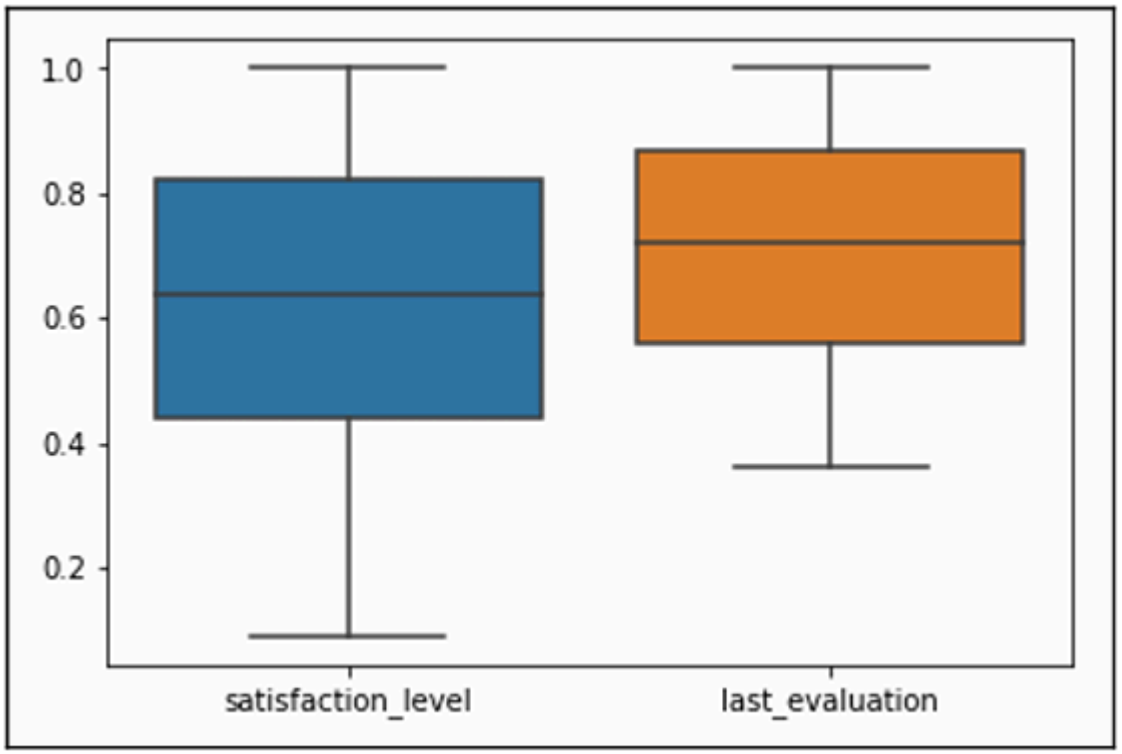


Figure 10. Hình ảnh mô tả biểu đồ hộp được tạo bởi Seaborn

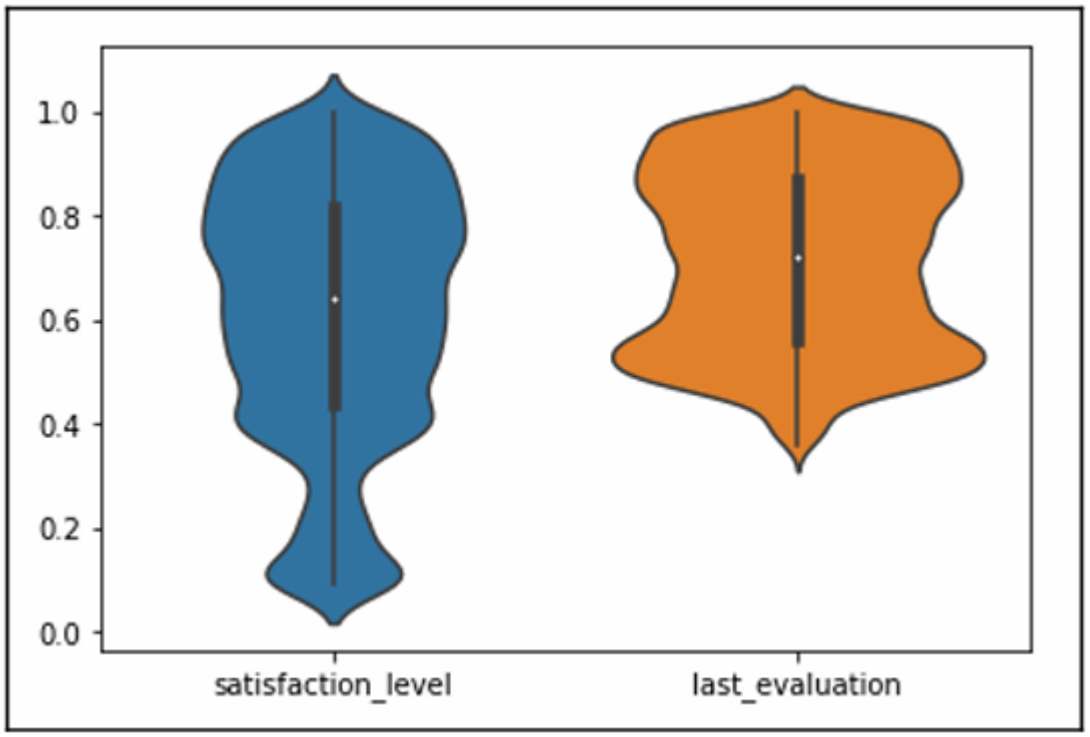


Figure 11. Biểu đồ violin được tạo bởi Seaborn[4]

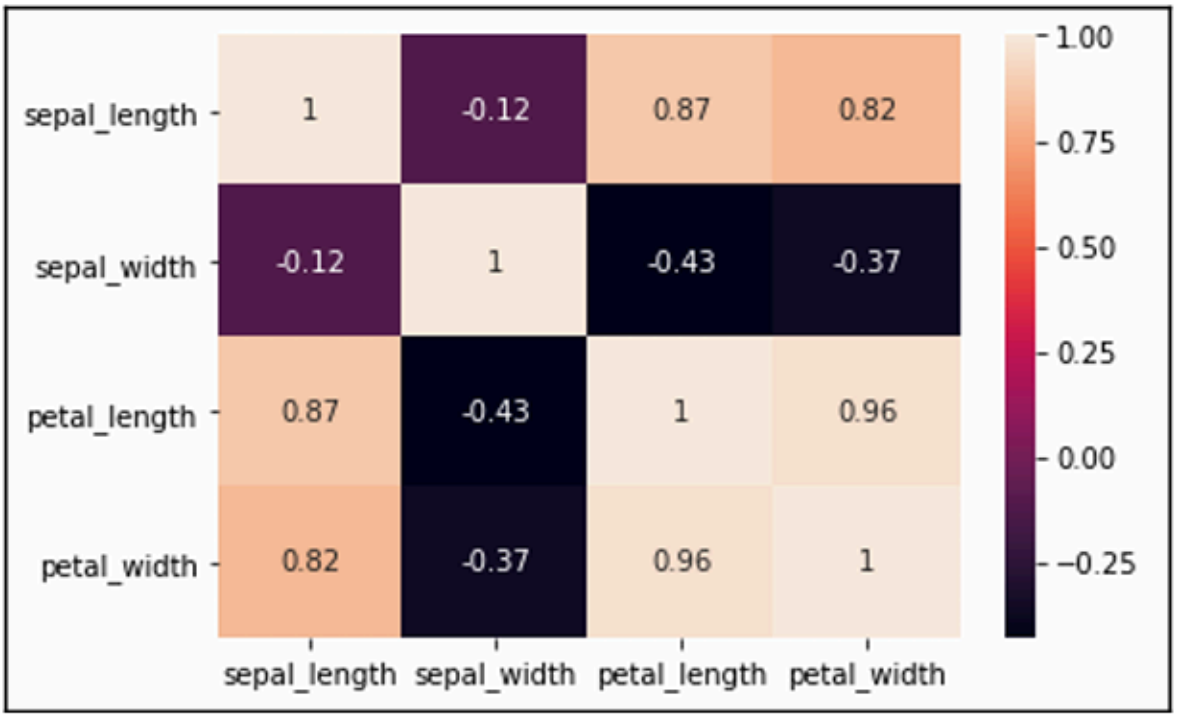


Figure 12. Biểu đồ heatmap[4]

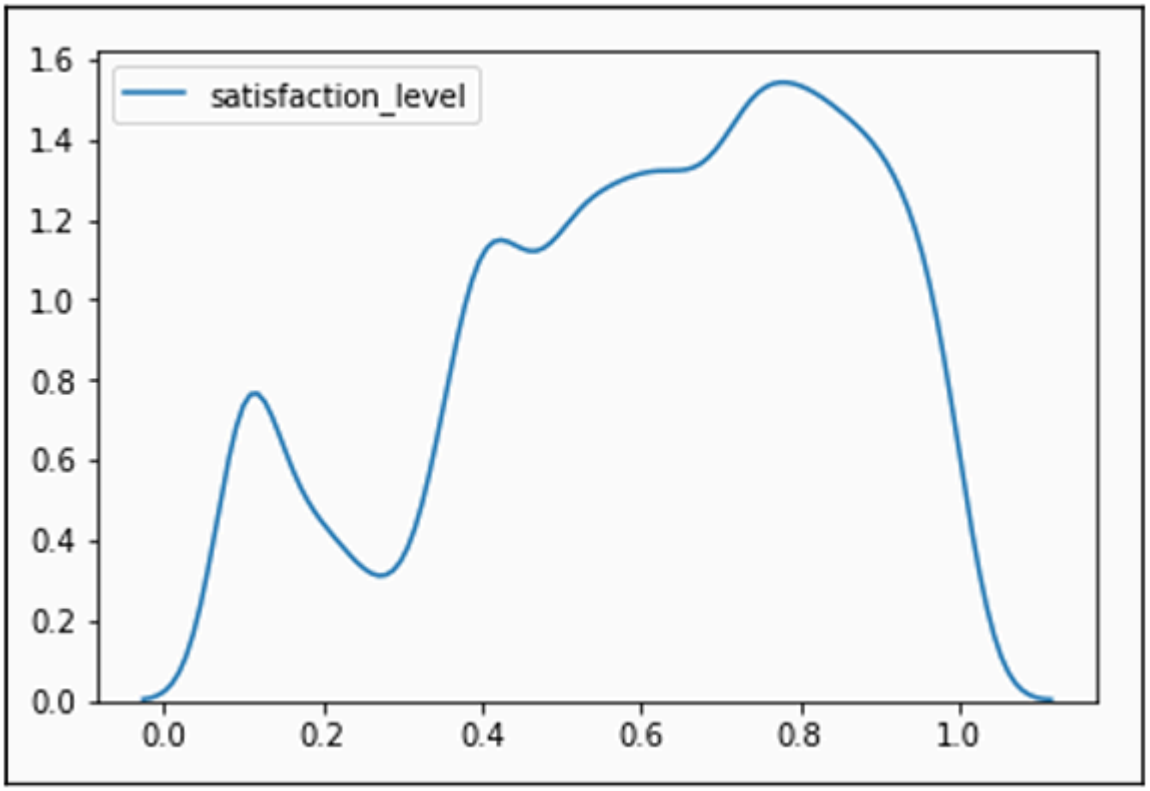


Figure 13. Biểu đồ ước lượng mật độ (KDE)[4]

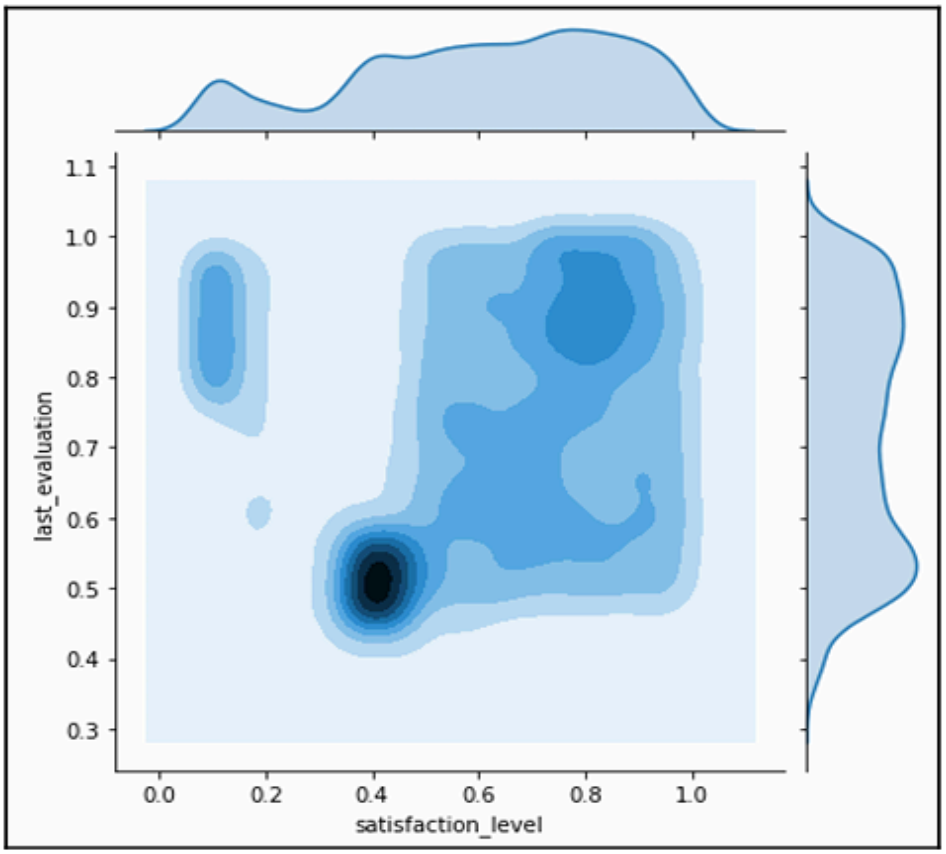


Figure 14. Biểu đồ Joint plot[4]

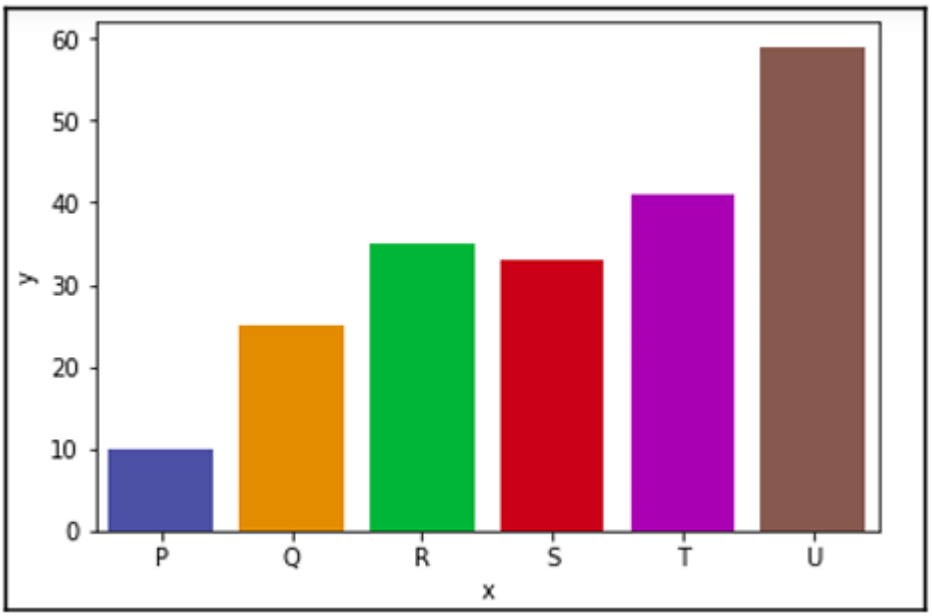


Figure 15. Biểu đồ cột được tạo bởi Seaborn[4]

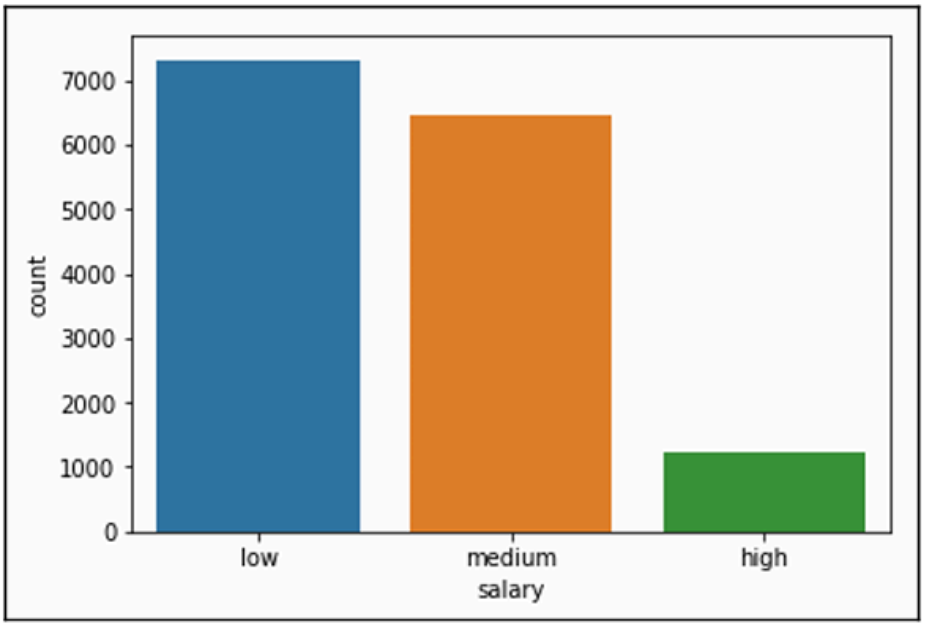


Figure 16. Biểu đồ đếm[4]

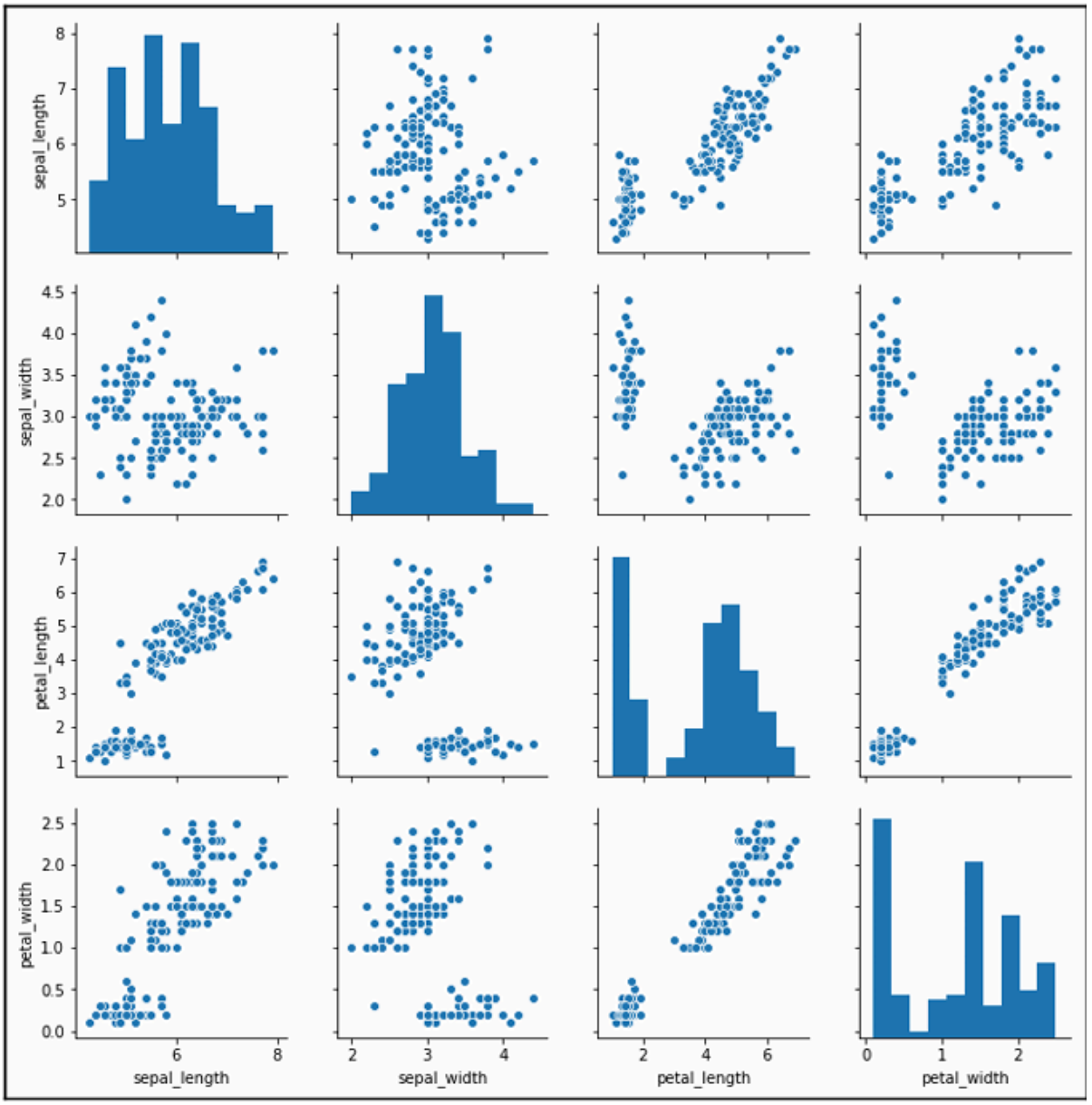


Figure 17. Biểu đồ pair plot

1.3.3. Các kỹ thuật trực quan hoá với Bokeh

Cách cài đặt Bokeh

Table 5. Bảng mô tả cách thức cài đặt Bokeh với Python và Anaconda.

Môi trường	Câu lệnh cài đặt
Python	pip install bokeh
Anaconda	conda install bokeh

Khác với matplotlib và seaborn, Việc sử dụng bokeh để trực quan hoá dữ liệu thì cần nạp các hàm chức năng của thư viện sau từ module bokeh.plotting:

- Hàm figure để tạo biểu đồ bằng cách sử dụng câu lệnh:

  from bokeh.plotting import figure

- Hàm show để hiển thị kết quả của biểu đồ với câu lệnh:

  from bokeh.plotting import show

- Cuối cùng là hàm output_notebook dùng để xác định đầu ra của đoạn mã nguồn sẽ là một biểu đồ được hiển thị trên Jupyter notebook bằng cách sử dụng câu lệnh:

  from bokeh.plotting import output_notebook

Bên cạnh đó, Bokeh cũng hỗ trợ một số dạng biểu đồ để có thể thực hiện trực quan hoá từ cơ bản đến nâng cao như sau:

Table 6. Một số dạng biểu đồ căn bản với Bokeh

Dạng đồ thị	Hàm hỗ trợ
Đồ thị điểm với kí hiệu chấm tròn (circle)	figure.circle()
Đồ thị glyph (line)	figure.line()

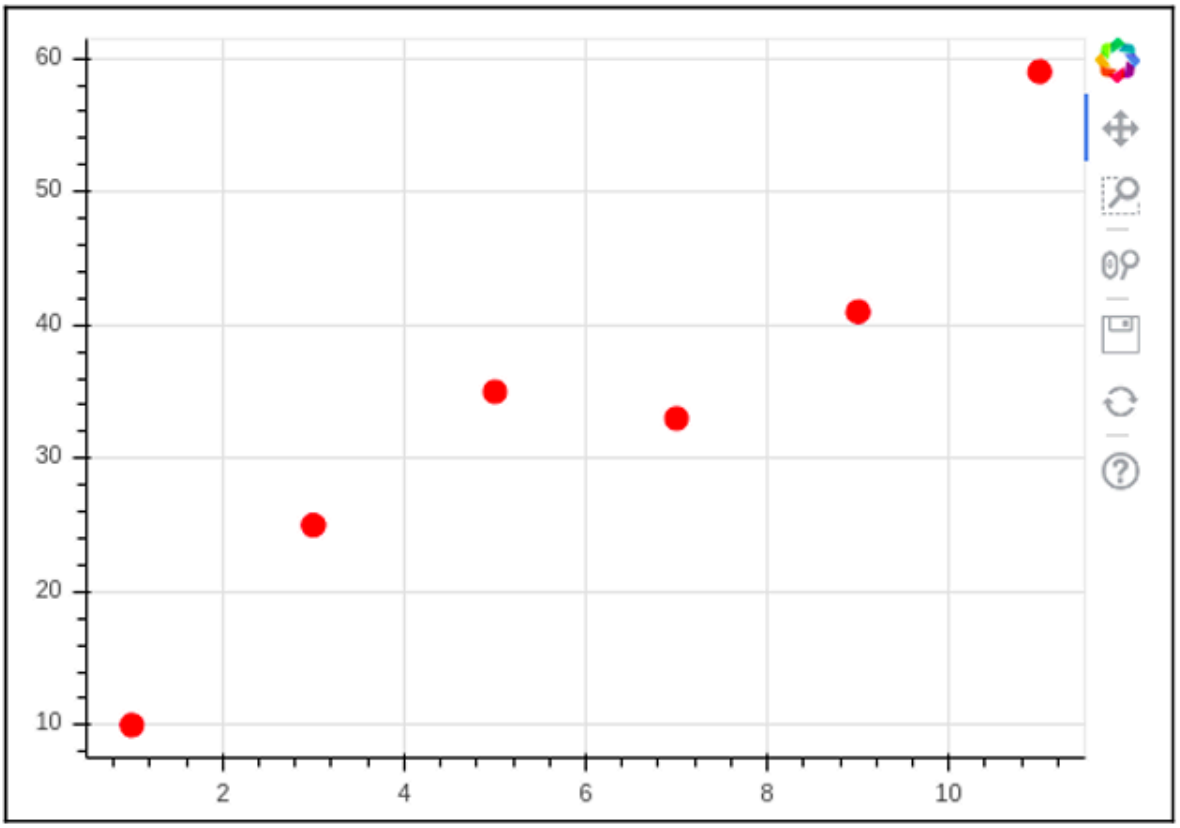


Figure 18. Biểu đồ điểm tròn được tạo bởi bokeh

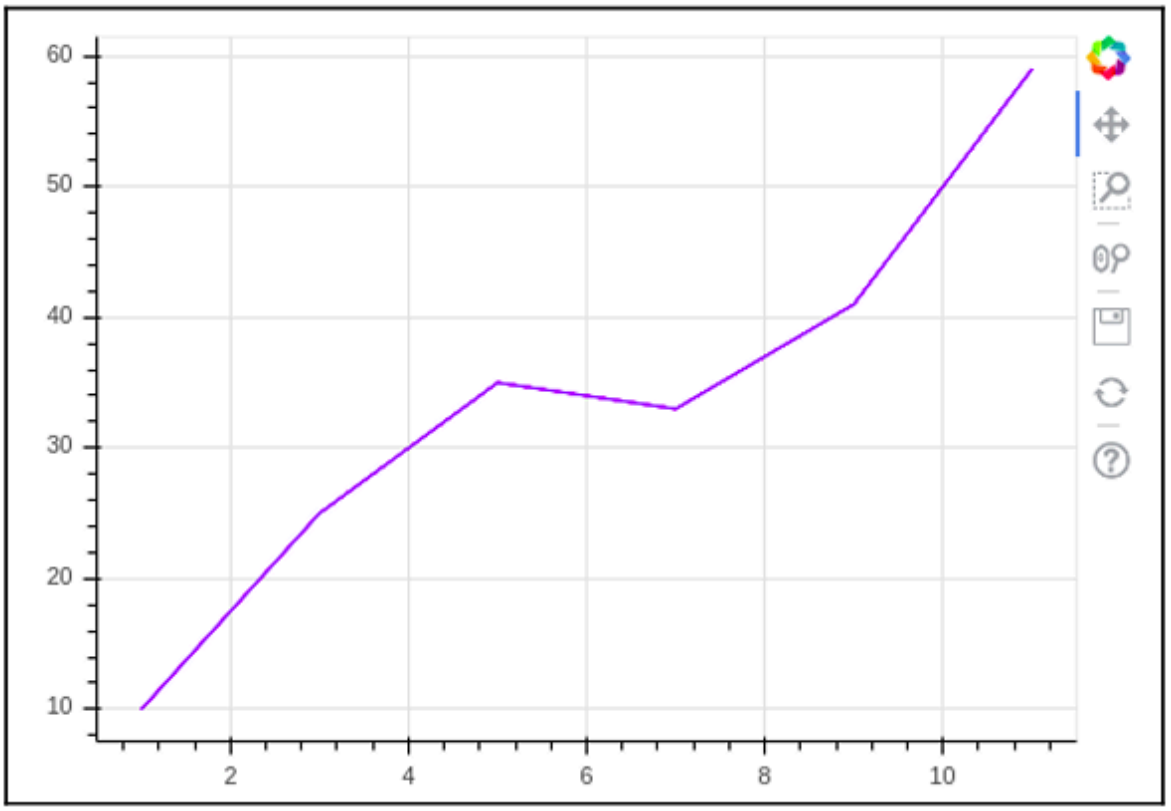


Figure 19. Biểu đồ đường được tạo bởi bokeh

Bokeh layout:[4] Bokeh có đề xuất một số layout nhằm tổ chức sắp xếp các biểu đồ và widget. Layout có thể tổ chức nhiều biểu đồ vào trong 1 panel khiến cho việc trực quan hoá trở nên có tính tương tác hơn. Chúng ta cũng có thể điều chỉnh kích thước, biểu đồ và các widget dựa trên kích thước của panel

Một số loại layout thông dụng

Table 7. Bảng mô tả các dạng layout của Bokeh

Loại layout	Định nghĩa
Row layout	Là dạng layout hiển thị các biểu đồ trong 1 hàng ngang (Xem ví dụ hình 20)
Column layout	Là dạng layout hiển thị các biểu đồ trong 1 cột dọc (Xem ví dụ hình 21)
Nested layout	Là sự kết hợp giữa row layout và column layout (Xem ví dụ hình 22)
Grid layout	Với layout, người dùng có thể sắp xếp các biểu đồ dưới dạng một ma trận (Xem ví dụ hình 23)

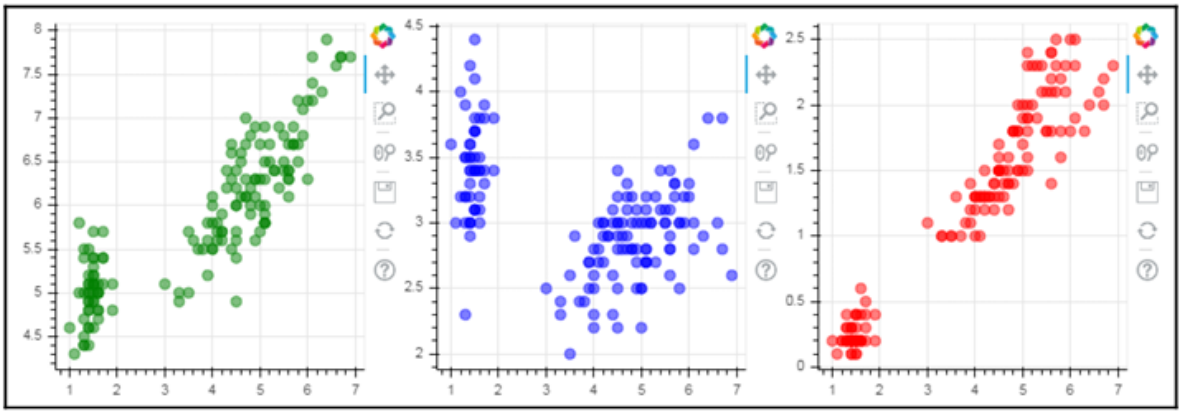


Figure 20. Hình ảnh mô tả row layout[4]

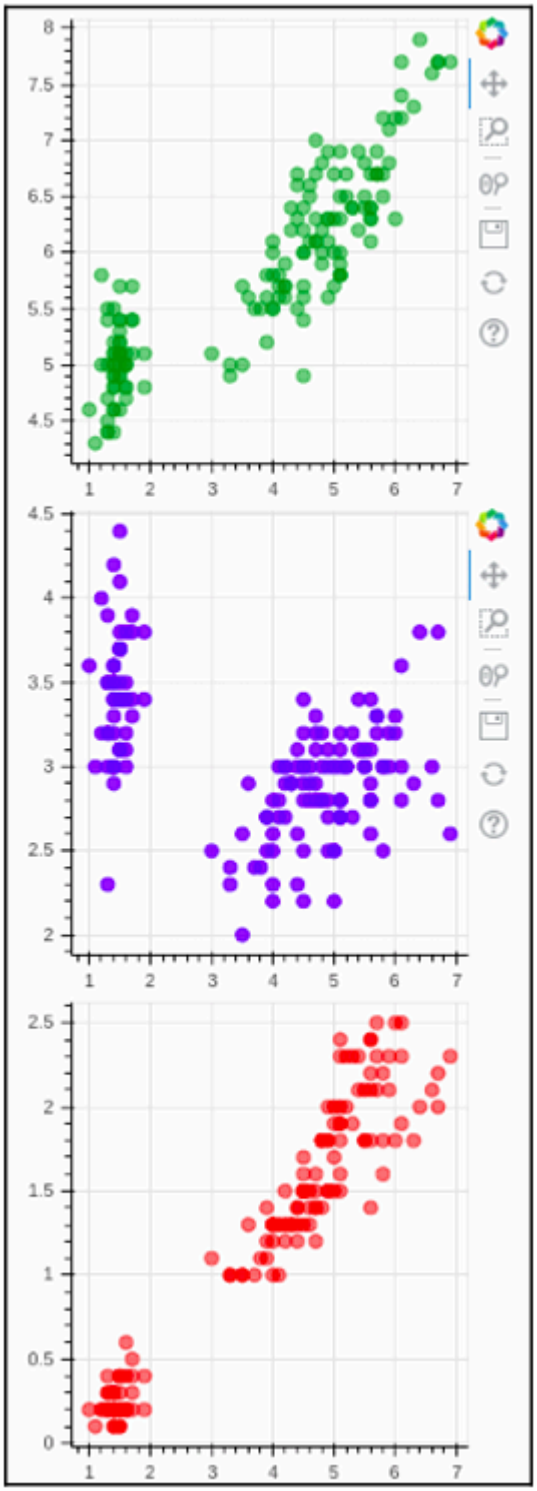


Figure 21. Hình ảnh mô tả column layout[4]

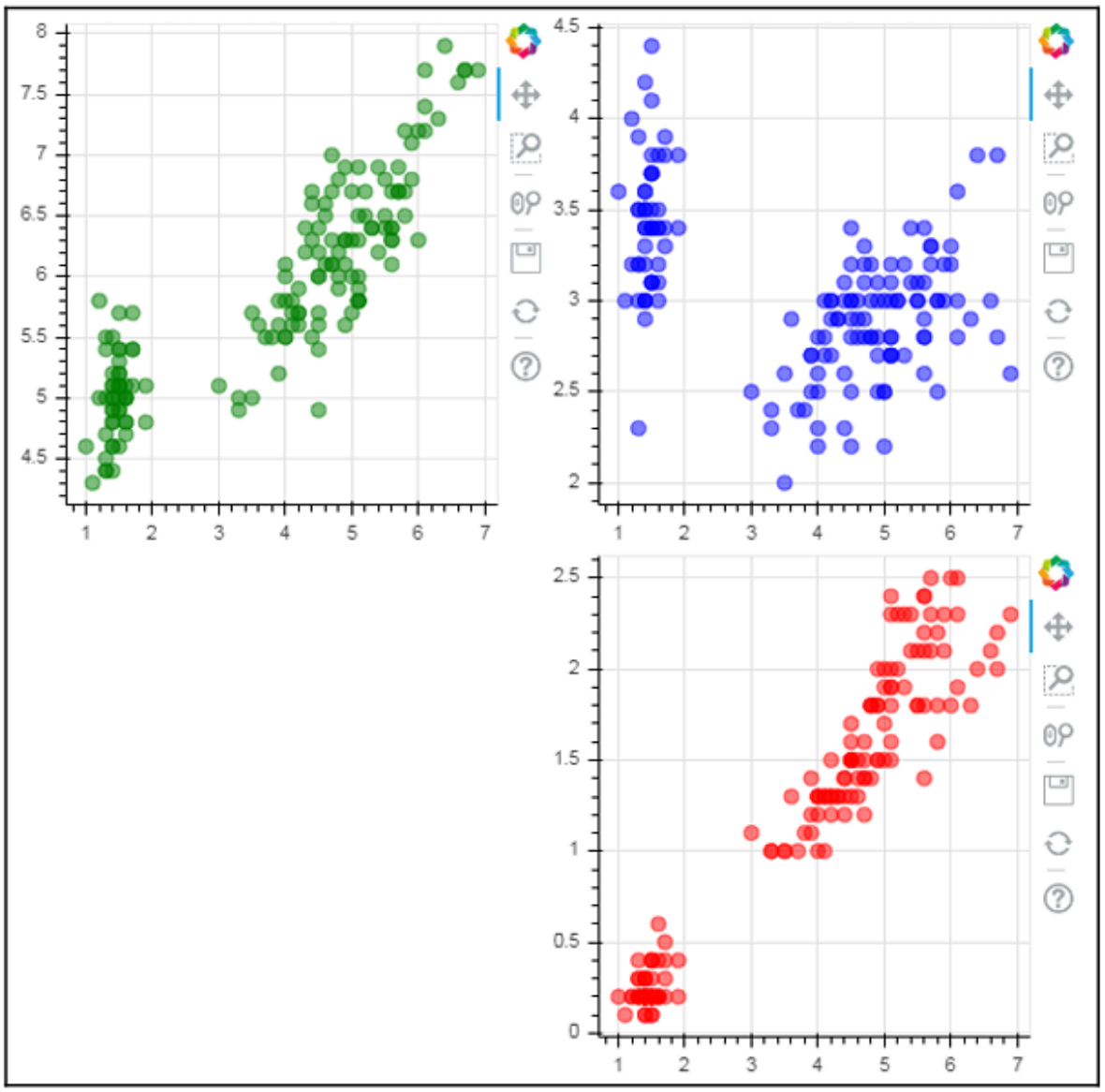


Figure 22. Hình ảnh mô tả nested layout[4]

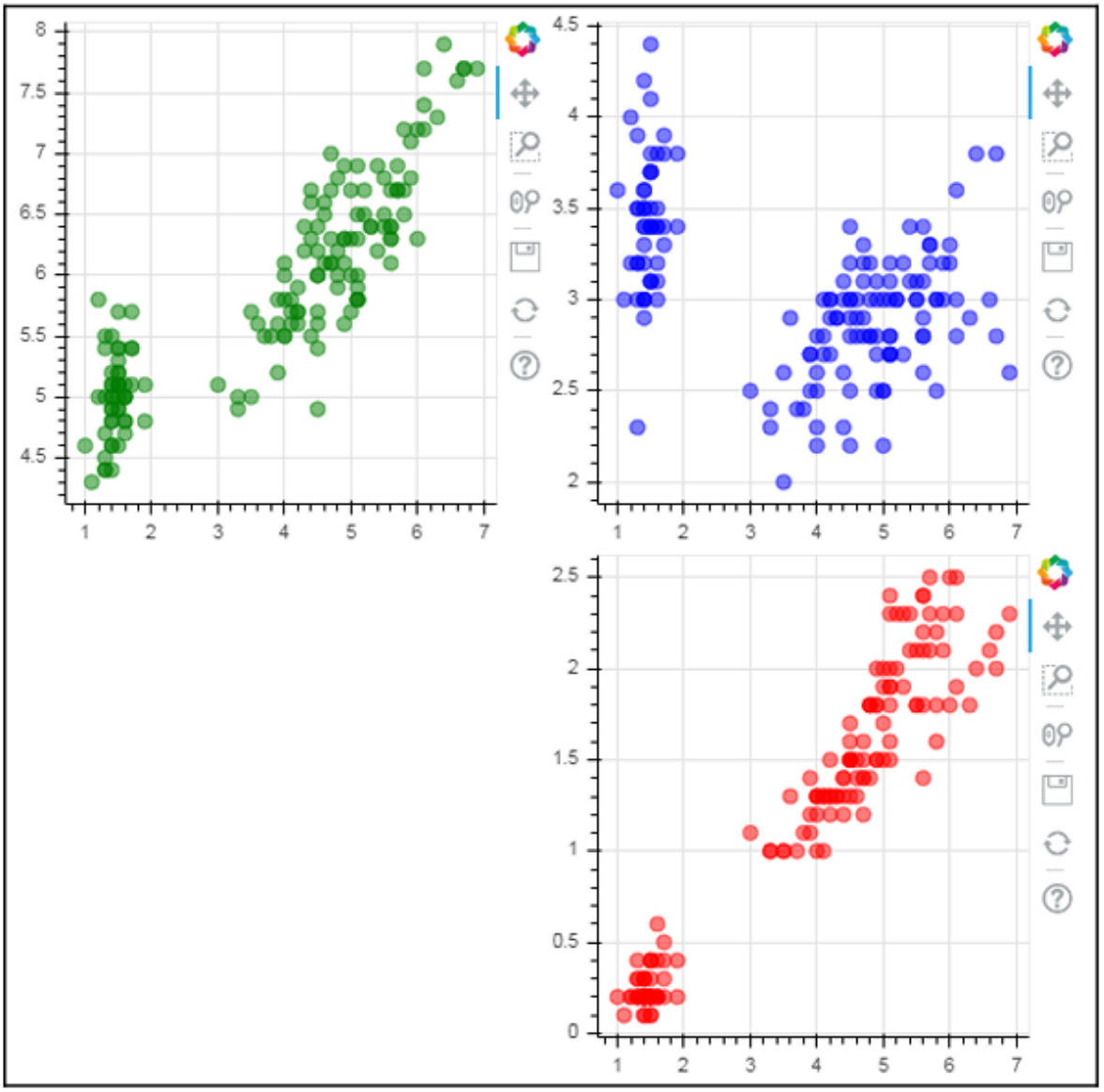


Figure 23. Hình ảnh mô tả đa đồ thị sử dụng grid layout[4]

Tính tương tác của Bokeh (Interaction): [4]Bokeh cho phép tương tác với chú thích của đồ thị. Phần chú thích này có thể ẩn hoặc hiện bằng cách nhấn vào biểu đồ glyph. Để có thể thực hiện tích năng này cần kích hoạt thuộc tính click_policy của module figure.legend.

Câu lệnh kích hoạt thuộc tính click_policy để ẩn hiện bảng chú thích của biểu đồ như sau:

- Hiện bảng chú thích: fig.legend.click_policy = “mute”
- Ẩn bảng chú thích: fig.legend.click_policy = “hide”

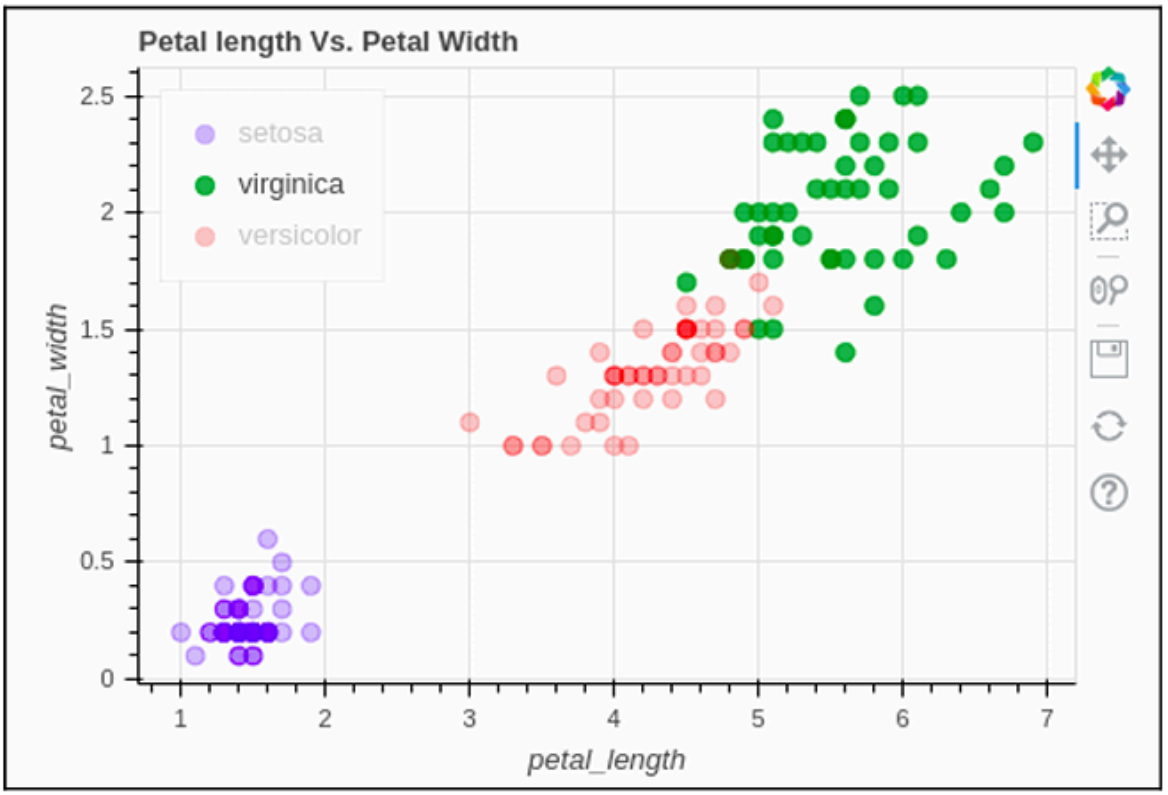


Figure 24. Hình ảnh ví dụ cho tính năng ẩn hiện bảng chú thích với bokeh[4]

Chú thích biểu đồ với Bokeh (Annotation): Bokeh cũng có khả năng cung cấp một số tính năng để chú thích các thông tin bổ sung trong việc trực quan hoá dữ liệu. Nó giúp cho người xem có thể theo dõi thông tin.

Một số thuộc tính được cung cấp để chú thích thông tin:

Thuộc tính	Định nghĩa
Titles	Là thuộc tính cho biết tên của biểu đồ
Axis labels	Là thuộc tính xác định tên của các trục toạ độ trong biểu đồ
Legends	Là thuộc tính chú thích các biến số dựa theo màu sắc, hình dáng giúp người xem có thể liên kết các đặc trưng của từ đó giúp làm sáng tỏ một cách dễ dàng hơn

Color bars	Là thuộc tính xác định màu dựa trên bảng màu được cung cấp bởi lớp ColorMapper

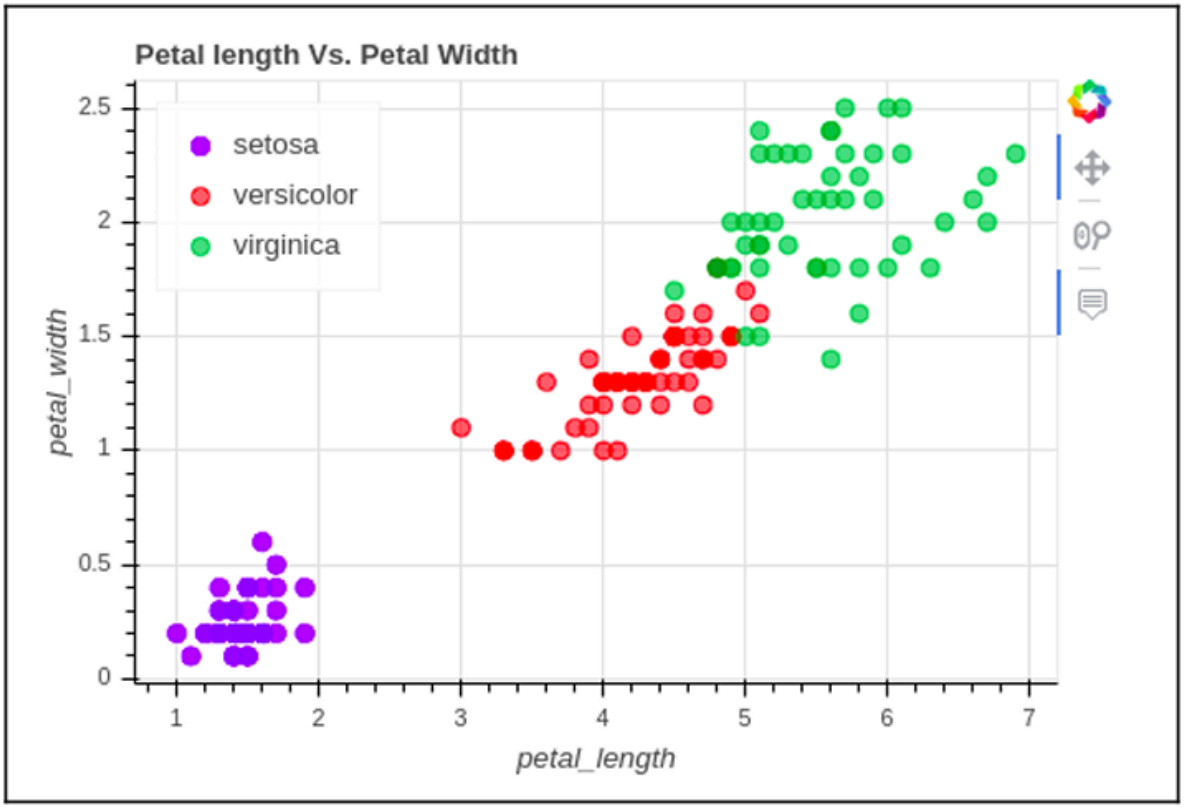


Figure 25. Hình ảnh mô tả các loại chú thích trên một biểu đồ với bokeh[4]

Bokeh hover tool (công cụ di chuyển chuột):[4] Là công cụ hiển thị thông tin liên quan bất cứ khi nào con trỏ chuột được đặt trên một khu vực cụ thể. Bokeh có hỗ trợ Hover tool bằng lớp HoverTool trong module bokeh.model. Để có thể sử dụng thì người dùng có thể nạp lớp HoverTool theo cú pháp:

from bokeh.model import HoverTool

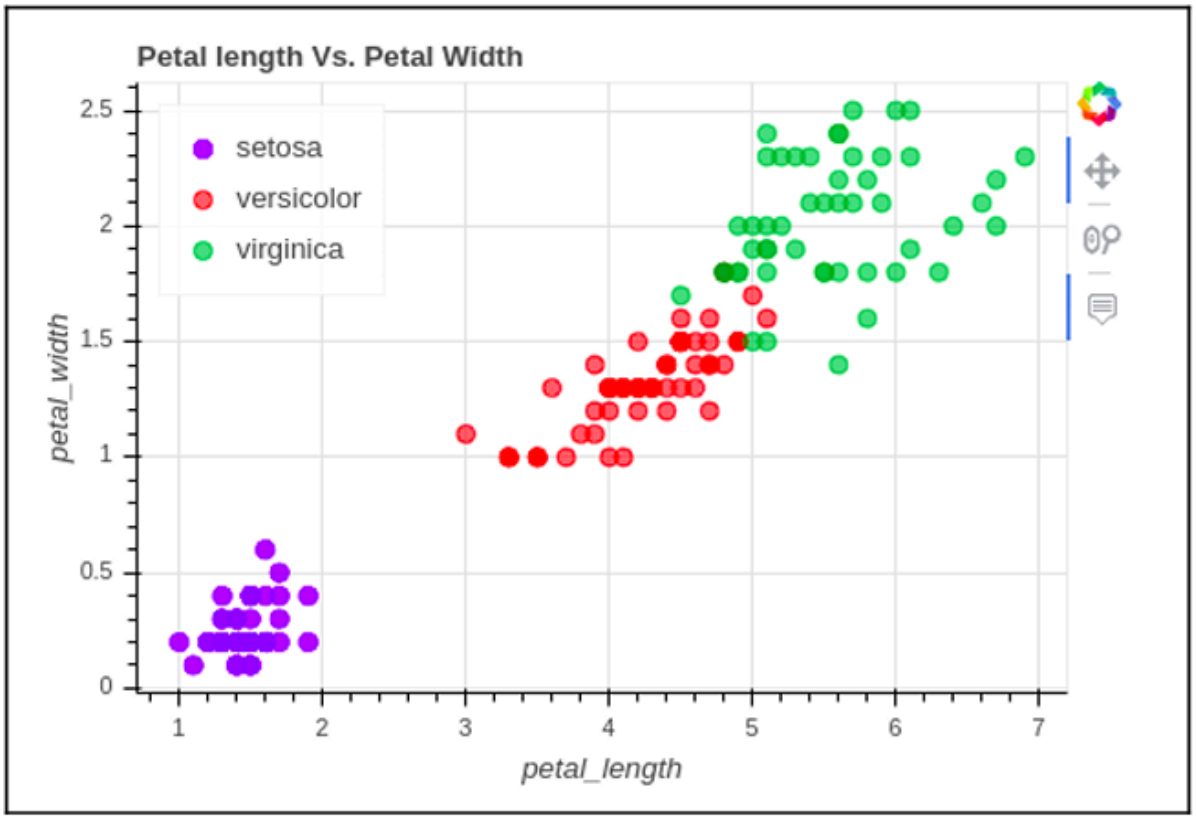


Figure 26. Hình ảnh ví dụ cho bokeh hover tool

Bokeh Tab Panel: [4]Là công cụ cho phép hiển thị nhiều biểu đồ và layout trên cùng một khung cửa sổ. Để có thể sử dụng Tab Panel thì Bokeh đã hỗ trợ hai lớp Tabs và lớp Panel trong module bokeh.model.widgets

- Lớp Panel:
  - Lớp Panel cho phép nạp các đồ thị vào trong cùng một Panel
  - Cách sử dụng:
    - Nạp lớp Panel theo cú pháp: from bokeh.model.widgets import Panel
    - Gọi constructor của lớp Panel với tham số child là biểu đồ được tạo ra bởi bokeh.plotting.figure, tham số title là tên của đồ thị
- Lớp Tabs:
  - Lớp Tabs dùng để tạo một cửa sổ nhằm nạp các Panel chứa các đồ thị
  - Cách sử dụng:
    - Nạp lớp Tabs theo cú pháp: from bokeh.model.widgets import Tabs
    - Gọi constructor của lớp Tabs với thsm số tabs là danh sách các Panel

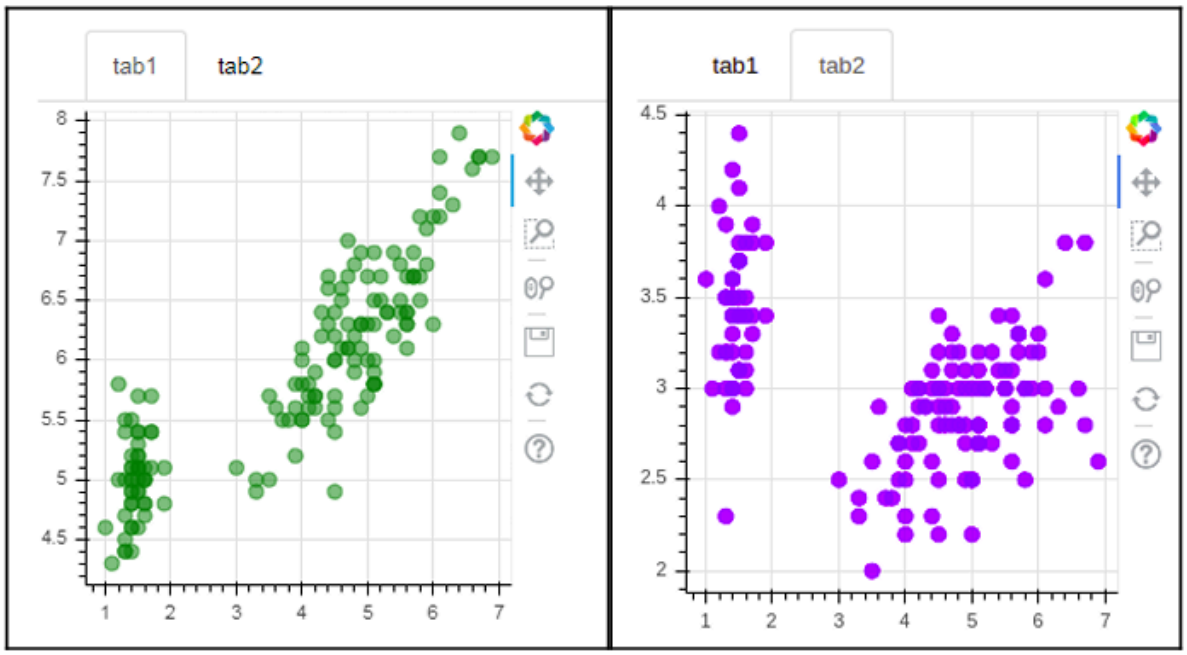


Figure 27. Hình ảnh minh hoạ tính năng Tab panel của bokeh

CHƯƠNG 2: GIỚI THIỆU BỘ DỮ LIỆU VÀ KẾT QUẢ THỰC NGHIỆM

2.1. Môi trường thực nghiệm

Nhóm chúng em sử dụng Google Colab để tiến hành thực nghiệm vì yếu tố teamwork và tính công khai giữa các thành viên trong nhóm nhờ vào khả năng quản lý bộ mã nguồn và tập dữ liệu của Google Drive.

Về vấn đề đọc dữ liệu để tiến hành thực nghiệm thì chúng em sử dụng kết hợp giữa pypark và pandas để tiến hành đọc và xử lý dữ liệu trước khi tiến hành trực quan hoá. Tuy nhiên, thư viện pyspark không được trang bị sẵn trên Colab nên mỗi khi thực nghiệm thì đều phải chạy câu lệnh sau để cài đặt pyspark:

```
%pip install pyspark
```

2.2. Giới thiệu bộ dữ liệu Youtube trending

2.2.1 Giới thiệu tổng quan bộ dữ liệu

Dữ liệu được lấy về từ nền tảng cộng đồng khoa học trực tuyến Kaggle. Bộ dữ liệu gồm 45560 video thịnh hành trên nền tảng youtube từ tháng 11 năm 2017 đến tháng 6 năm 2018. Các thông tin chi tiết bao gồm ID video, ngày video trở thành xu hướng, tiêu đề, tên kênh, mã danh mục, số lượt xem, số lượt thích và nhiều thông tin khác.

2.2.2 Tiền xử lý tập tin CAvideos.csv

Tiền xử lý dữ liệu: Quá trình tiền xử lý bao gồm các bước sau đây:

- Chuẩn hóa dữ liệu: chuyển đổi các giá trị dữ liệu sang đúng kiểu dữ liệu phù hợp để phân tích dữ liệu một cách chính xác nhất, chúng ta cần xác định các trường dữ liệu và chọn kiểu dữ liệu phù hợp dựa trên tính chất của từng trường.

```
root
 |-- video_id: string (nullable = true)
 |-- trending_date: string (nullable = true)
 |-- title: string (nullable = true)
 |-- channel_title: string (nullable = true)
 |-- category_id: string (nullable = true)
 |-- publish_time: string (nullable = true)
 |-- tags: string (nullable = true)
 |-- views: string (nullable = true)
 |-- likes: string (nullable = true)
 |-- dislikes: string (nullable = true)
 |-- comment_count: string (nullable = true)
 |-- thumbnail_link: string (nullable = true)
 |-- comments_disabled: string (nullable = true)
 |-- ratings_disabled: string (nullable = true)
 |-- video_error_or_removed: string (nullable = true)
 |-- description: string (nullable = true)
```

Figure 28. Schema của tập dữ liệu trước khi chuẩn hoá

```
root
 |-- video_id: string (nullable = true)
 |-- trending_date: date (nullable = true)
 |-- title: string (nullable = true)
 |-- channel_title: string (nullable = true)
 |-- category_id: integer (nullable = true)
 |-- publish_time: string (nullable = true)
 |-- tags: string (nullable = true)
 |-- views: integer (nullable = true)
 |-- likes: integer (nullable = true)
 |-- dislikes: integer (nullable = true)
 |-- comment_count: integer (nullable = true)
 |-- thumbnail_link: string (nullable = true)
 |-- comments_disabled: string (nullable = true)
 |-- ratings_disabled: string (nullable = true)
 |-- video_error_or_removed: string (nullable = true)
 |-- description: string (nullable = true)
```

Figure 29. Schema của dữ liệu sau khi chuẩn hoá

- Loại bỏ các giá trị nhiễu: Loại bỏ các giá trị nhiễu, các giá trị bị mất hoặc bị sai sót.

video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes
\nCast : Imran Ab...	NULL	Mehboob Sultan	Farah Zeba	Faisal Bali	Ikhyar Khan	Areej Fatima	NULL	NULL	NULL
\nCast : Imran Ab...	NULL	Mehboob Sultan	Farah Zeba	Faisal Bali	Ikhyar Khan	Areej Fatima	NULL	NULL	NULL
\nCast : Imran Ab...	NULL	Mehboob Sultan	Farah Zeba	Faisal Bali	Ikhyar Khan	Areej Fatima	NULL	NULL	NULL
\nSemolina pasta ...	NULL	malloreddus	lorighittas	cencioni	capunti	strascinati	NULL	NULL	NULL
\nSemolina pasta ...	NULL	malloreddus	lorighittas	cencioni	capunti	strascinati	NULL	NULL	NULL
\nSemolina pasta ...	NULL	malloreddus	lorighittas	cencioni	capunti	strascinati	NULL	NULL	NULL
\nEvoking memorie...	NULL	the Duchess of C...	as she smiled to...	Prince William s...	delighted" and j...	Prince William s...	NULL	NULL	NULL
\nEvoking memorie...	NULL	the Duchess of C...	as she smiled to...	Prince William s...	delighted" and j...	Prince William s...	NULL	NULL	NULL
\nKate	NULL	is said to be ge...	Carole Middleton	63	was spotted pick...	who was born on ...	NULL	NULL	NULL
Kate Middleton	NULL	was spotted in K...	two	in recent days.T...	you wouldn't kno...	according to the...	NULL	NULL	NULL
The Royal Wedding...	NULL	36	and Prince Harry	33. Guests were ...	with short instr...	morning coats	NULL	NULL	NULL
The Royal Wedding...	NULL	36	and Prince Harry	33. Guests were ...	with short instr...	morning coats	NULL	NULL	NULL
The Royal Wedding...	NULL	36	and Prince Harry	33. Guests were ...	with short instr...	morning coats	NULL	NULL	NULL

Figure 30. Hình ảnh mô tả một số dòng của bộ dữ liệu trong đó có vài dòng bị nhiễu do thiếu dữ liệu

Dữ liệu sau khi tiền xử lý ta còn lại 39586 giá trị.

2.3. Thực nghiệm trực quan hoá dữ liệu với tập tin CAvideos.csv của bộ dữ liệu

2.3.1. Trực quan hoá dữ liệu với Matplotlib

- Biểu đồ đường(line chart)

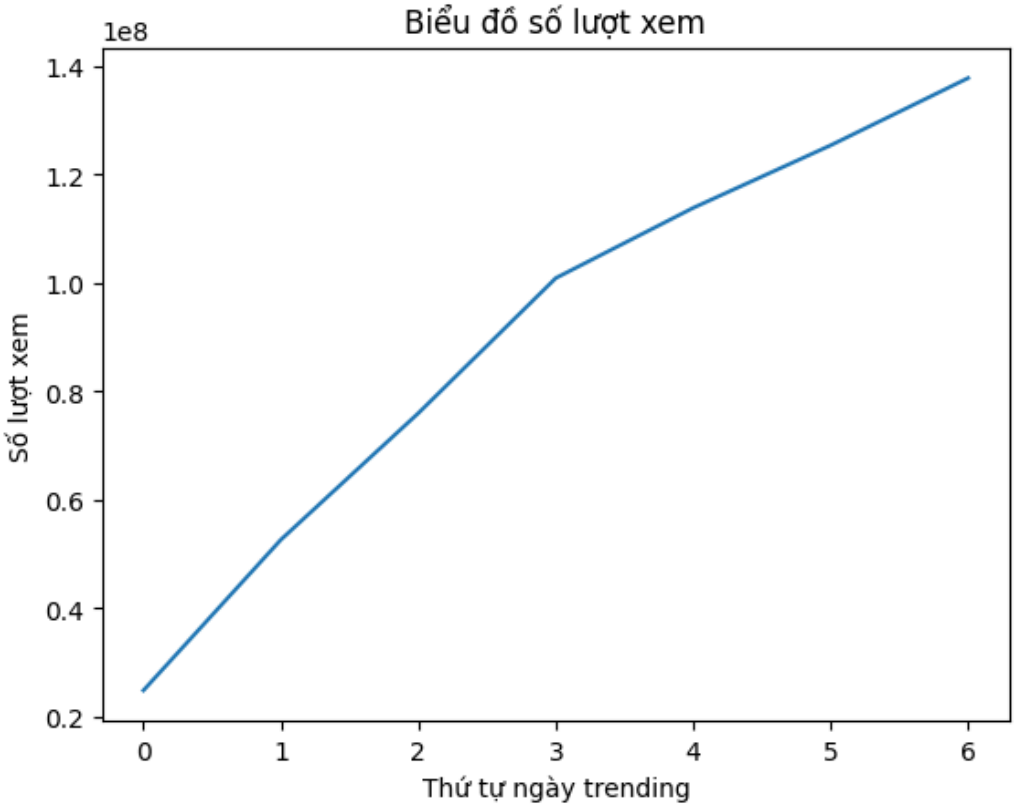


Figure 31. Biểu đồ đường mô tả số lượt xem trong 7 ngày trending

Với biểu đồ line này sẽ cho chúng ta thấy sự thay đổi của số lượt xem của video có video_id là 'FlsCjmMhFmw' theo thứ tự ngày trending. Nếu giá trị trục x tăng, điều đó có nghĩa là video được trending vào các ngày tiếp theo theo thứ tự tăng dần. Biểu đồ line này có thể giúp chúng ta nhận ra xu hướng tăng giảm của lượt xem của video theo thời gian.

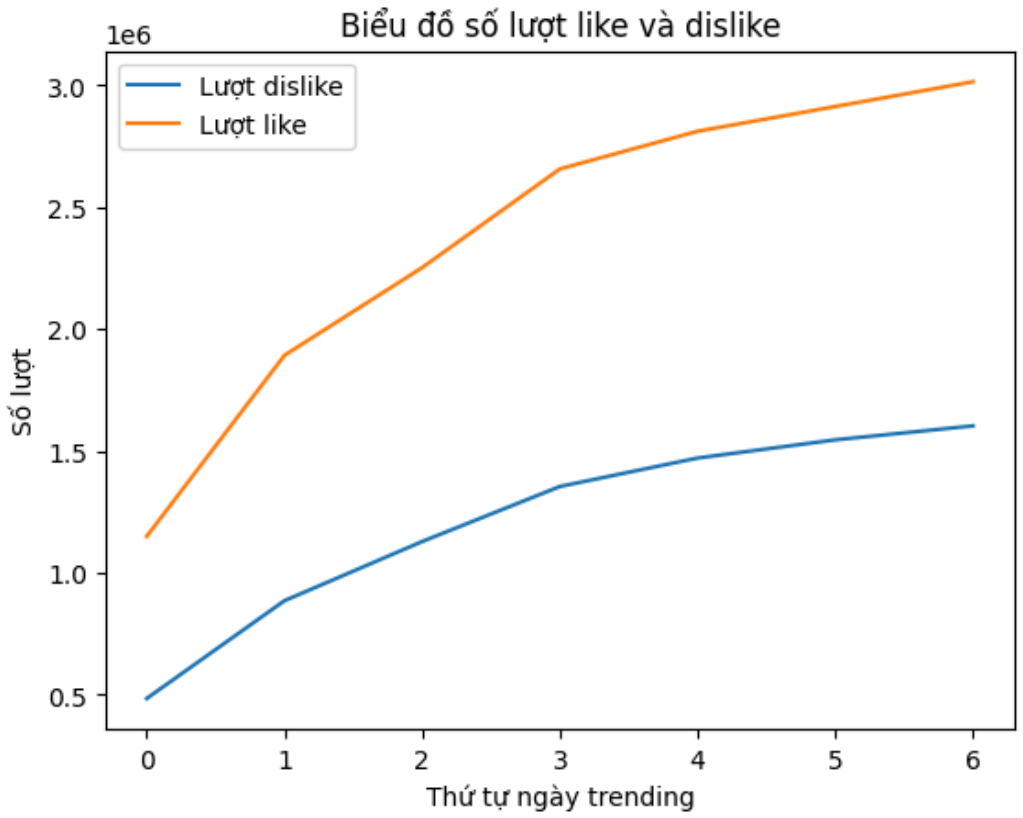


Figure 32. Biểu đồ mô tả số lượng like và dislike theo thứ tự 7 ngày trending

Với biểu đồ line này sẽ giúp chúng ta so sánh sự thay đổi của số lượt like và dislike của video theo thứ tự ngày trending. Nếu giá trị trục x tăng, điều đó có nghĩa là video được trending vào các ngày tiếp theo theo thứ tự tăng dần. Biểu đồ này có thể giúp chúng ta nhận ra xu hướng tương quan giữa lượt like và dislike của video theo thời gian.

- Biểu đồ phân bổ điểm (Scatter plot)

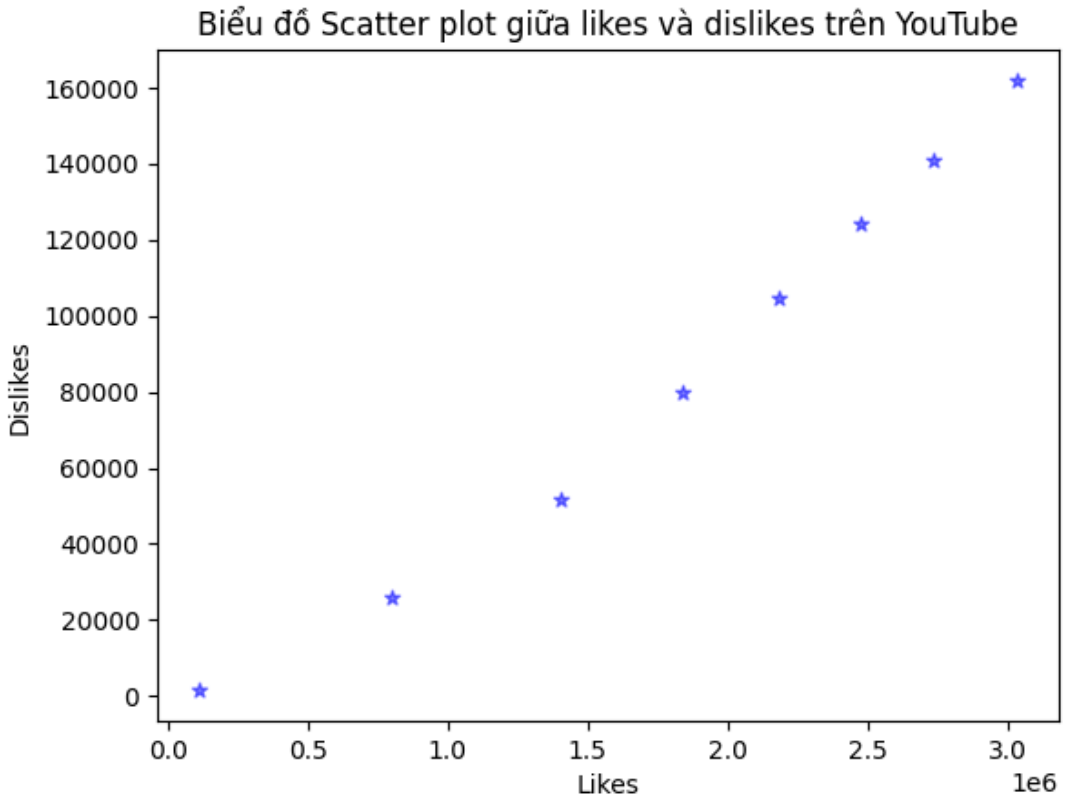


Figure 33. Biểu đồ scatter mô tả mối quan hệ giữa số lươt like và dislike trong một video

Với Biểu đồ Scatter plot này sẽ cho chúng ta thấy mối quan hệ giữa số lượt like và số lượt dislike của video có video_id là 'VYOjWnS4cMY'. Mỗi điểm trên biểu đồ đại diện cho một video trong bộ dữ liệu, và vị trí của điểm trên trục x (Likes) và trục y (Dislikes) thể hiện mối quan hệ giữa hai thuộc tính này. Nếu các điểm có xu hướng tập trung tại một vị trí gần nhau trên biểu đồ, điều đó có thể cho thấy mối tương quan giữa số lượt like và dislike của các video.

- Biểu đồ tròn (pie plot)

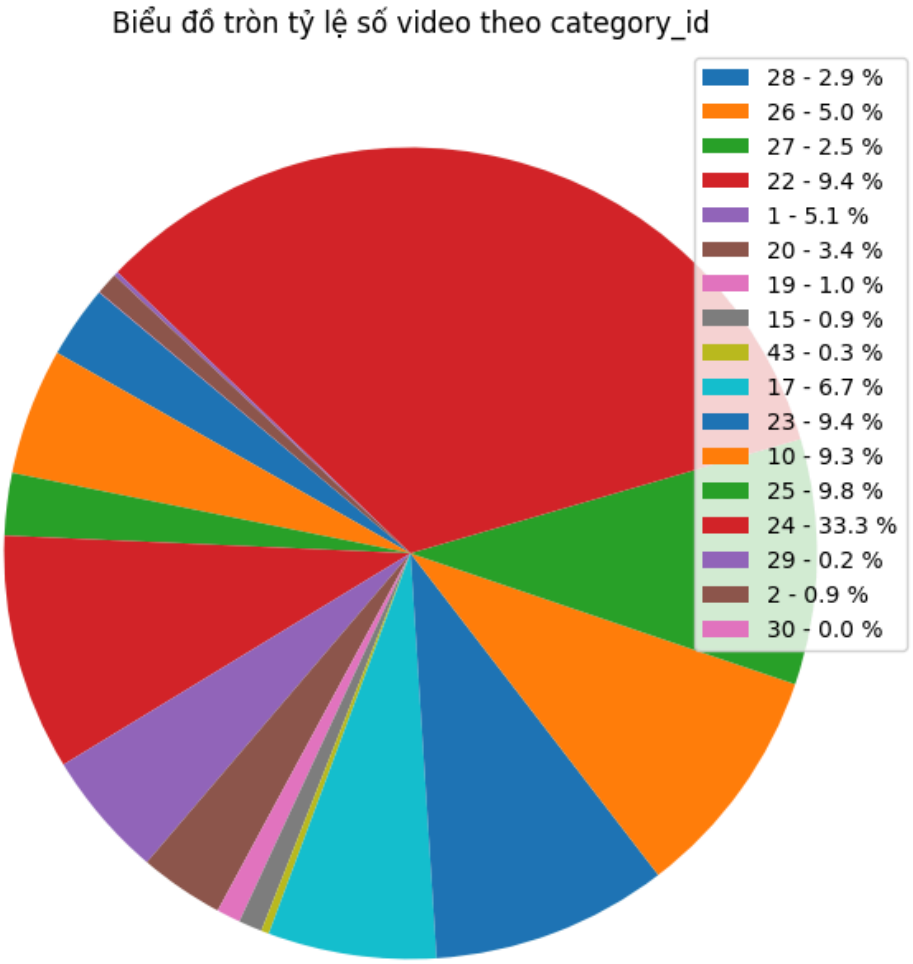


Figure 34. Biểu đồ tròn mô tả tỷ lệ số lượng video theo category ID

Với biểu đồ tròn này sẽ giúp chúng ta hiểu tỷ lệ số lượng video theo từng category_id trong bộ dữ liệu. Mỗi miền trên biểu đồ sẽ biểu thị một category_id cùng với tỷ lệ phần trăm của số lượng video trong category_id đó so với tổng số video. Chú thích trên biểu đồ cung cấp thông tin chi tiết về từng miền.

- Biểu đồ cột (bar plot)

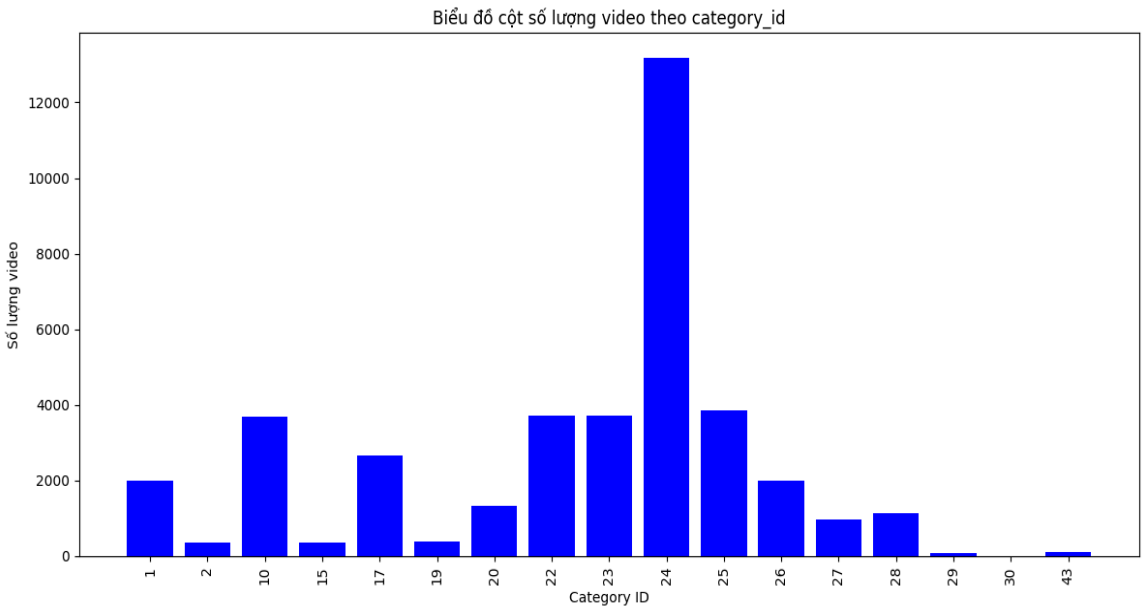


Figure 35. Biểu đồ cột mô tả số lượng video theo category ID

Với biểu đồ cột này sẽ giúp chúng ta hiểu rõ hơn về phân phối số lượng video theo từng category_id trong bộ dữ liệu. Mỗi cột trên biểu đồ biểu thị số lượng video của mỗi category_id, và nhãn trên trục x sẽ cho biết category_id tương ứng. Qua biểu đồ này, chúng ta có thể dễ dàng so sánh số lượng video giữa các category_id khác nhau.

2.3.2. Thực nghiệm bộ dữ liệu với Bokeh

- Biểu đồ chấm tròn:

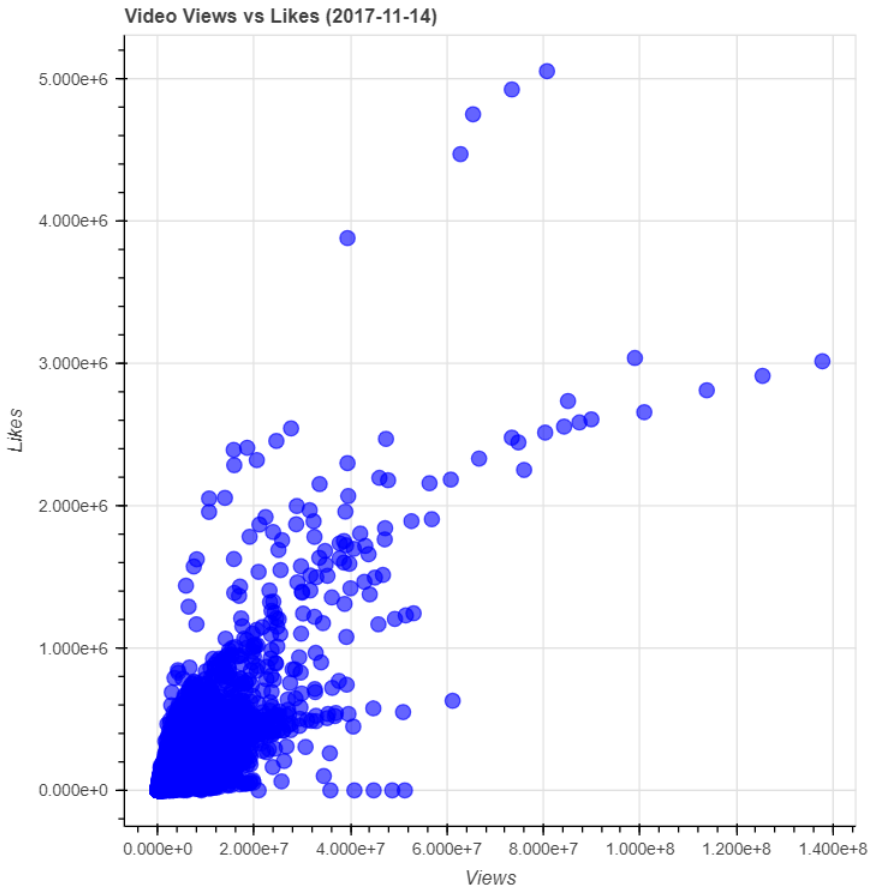


Figure 36. Biểu đồ phân bổ giữa số lượng views và số lượng like trong ngày 14/11/2017

Với biểu đồ chấm tròn đã mô tả sự phân bố tương quan giữa số lượt xem và số lượt like của các video có trong bộ dữ liệu trong ngày 17/11/2017. Từ biểu đồ ta thấy, việc phân bố lượt xem và lượt like của các video trong ngày này tập trung ở trong khoảng 0 đến 20 triệu lượt xem thì về phía lượt like tập trung khoảng từ 0 đến 1 triệu lượt like

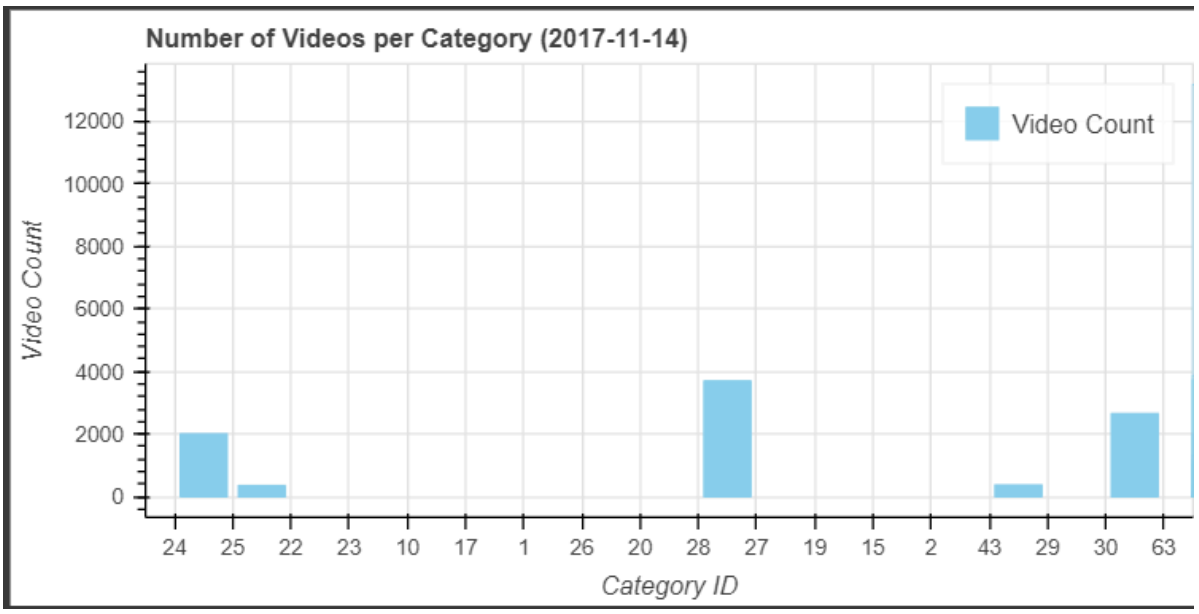
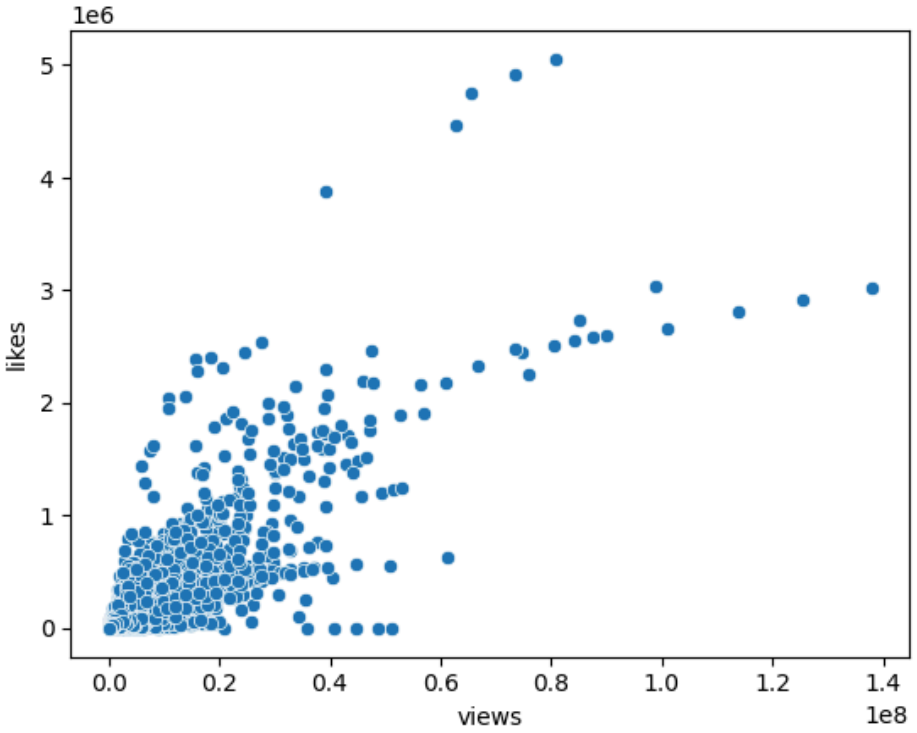


Figure 37. Biểu đồ cột mô tả số lượng video theo category được ghi nhận trong 1 ngày (ngày 14/11/2017)

Biểu đồ cột hình 37 cho thấy số lượng video và loại video (quản lý theo ID) được lên xu hướng trong ngày 14/11/2017 cho thấy có nhiều loại (category) không có video nào được lên xu hướng và loại video có mã số 26 là có nhiều video nhất với gần 4000 video.

2.3.3. Thực nghiệm trực quan hoá với Seaborn



-

Figure 38. Biểu đồ scatter biểu diễn sự tương quan giữa lượt view và lượt like được tạo bởi Seaborn

- lmplot

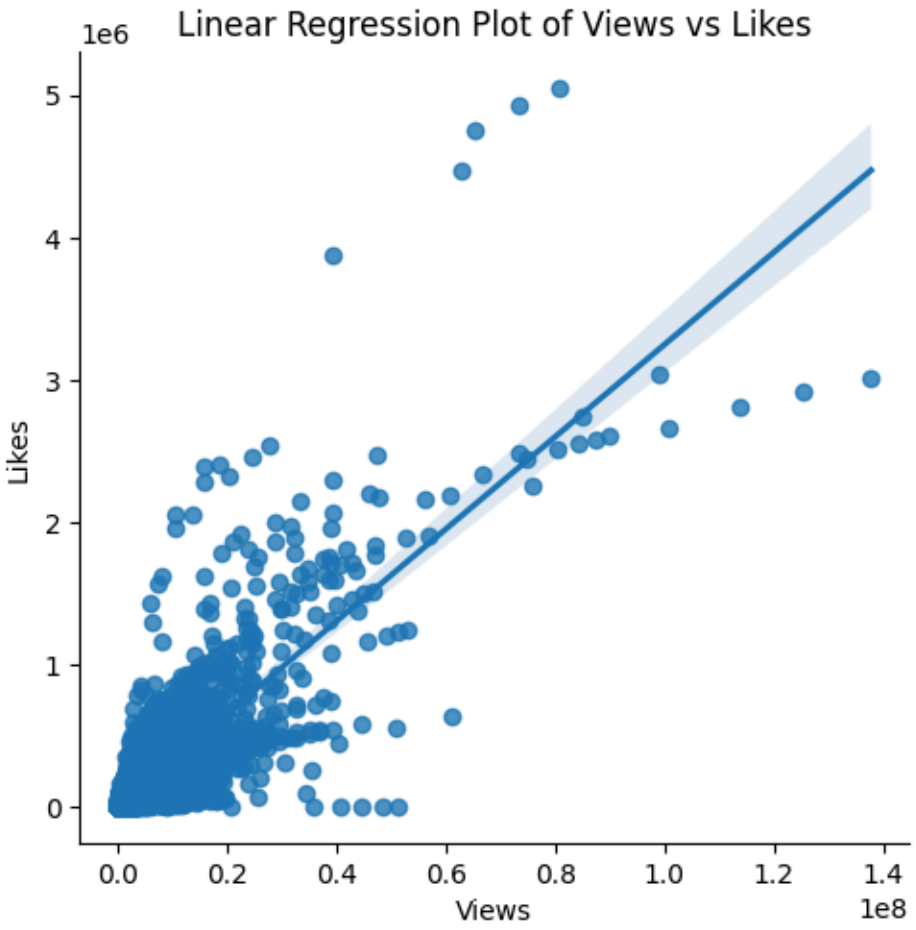


Figure 39. Biểu đồ Implot mô tả sự phân bổ số lượt like và lượt view

- Bar plot

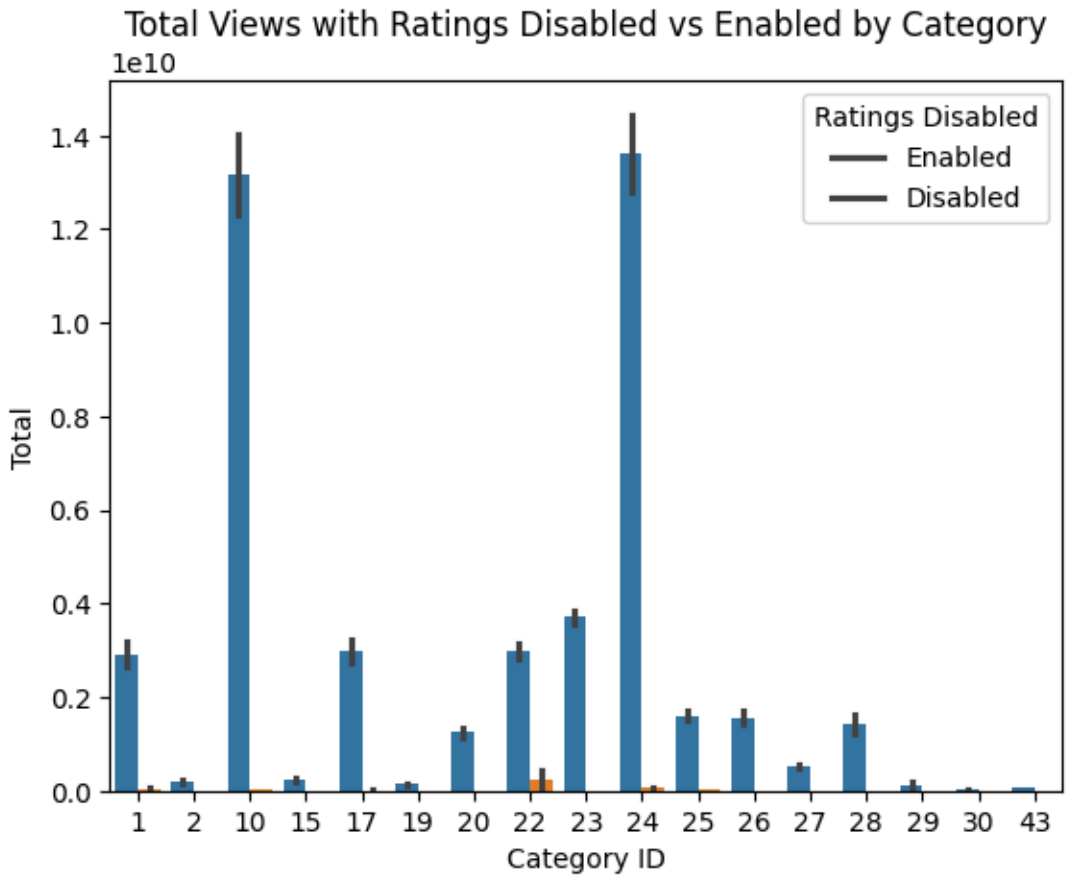


Figure 40. Biểu đồ cột đếm tổng lượt view của các video cho phép đánh giá và không cho phép đánh giá được sắp xếp theo loại

- Pair plot

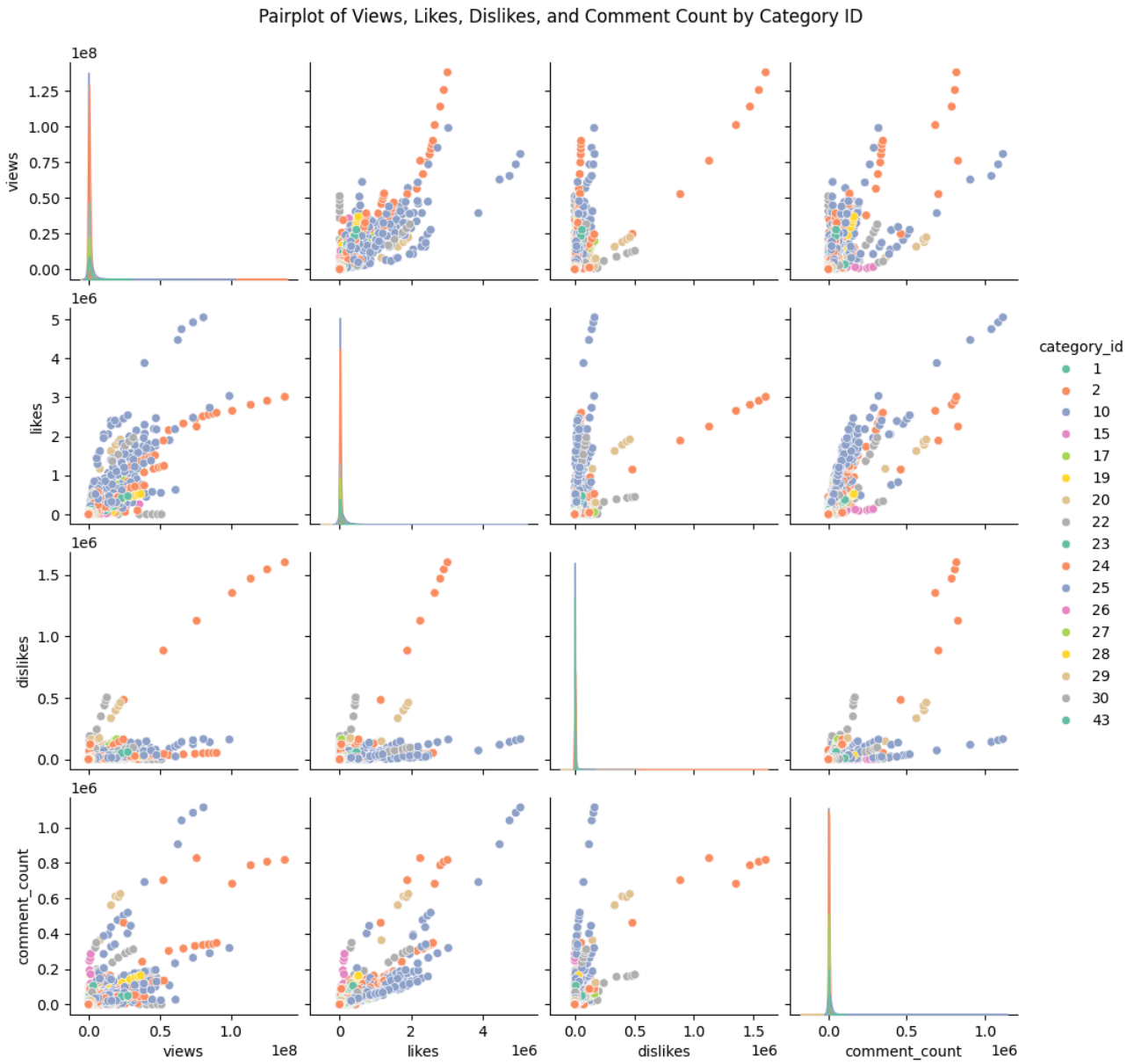


Figure 41. Biểu đồ pair plot hiển thị sự phân bổ lượt view, like, dislike và comment theo ID loại video

- Box plot

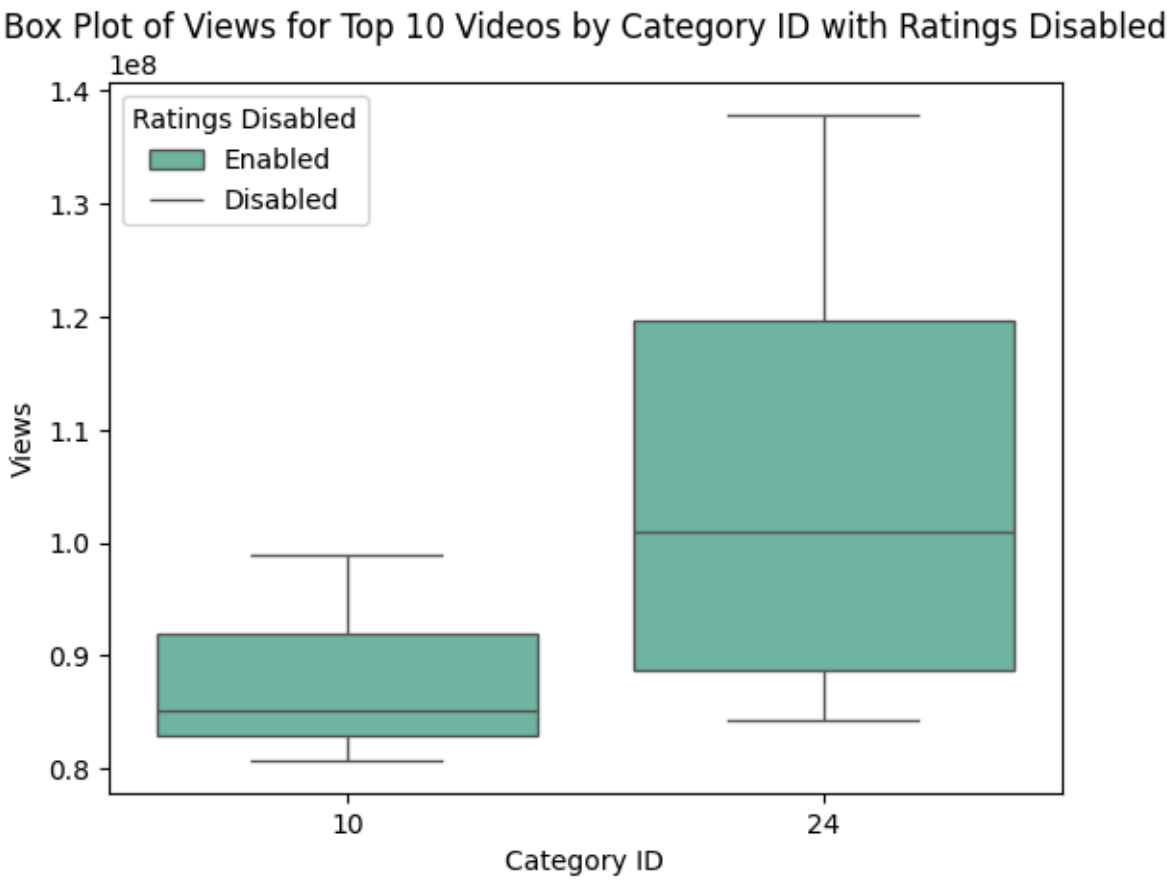


Figure 42. Biểu đồ hộp mô tả lượt view của top 10 video theo ID loại video mà bị tắt đánh giá

- KDE plot

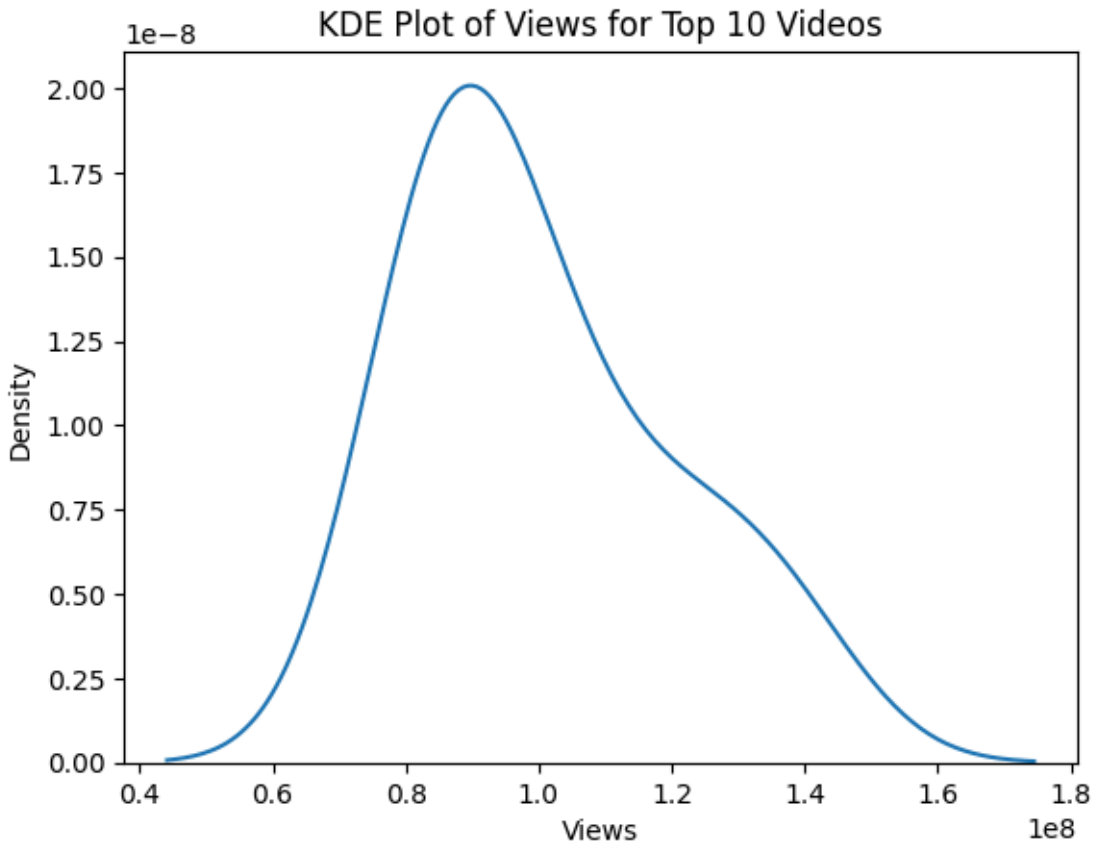


Figure 43. Biểu đồ KDE mô tả sự phân bổ mật độ số lượt view của top 10 video

- Violin plot

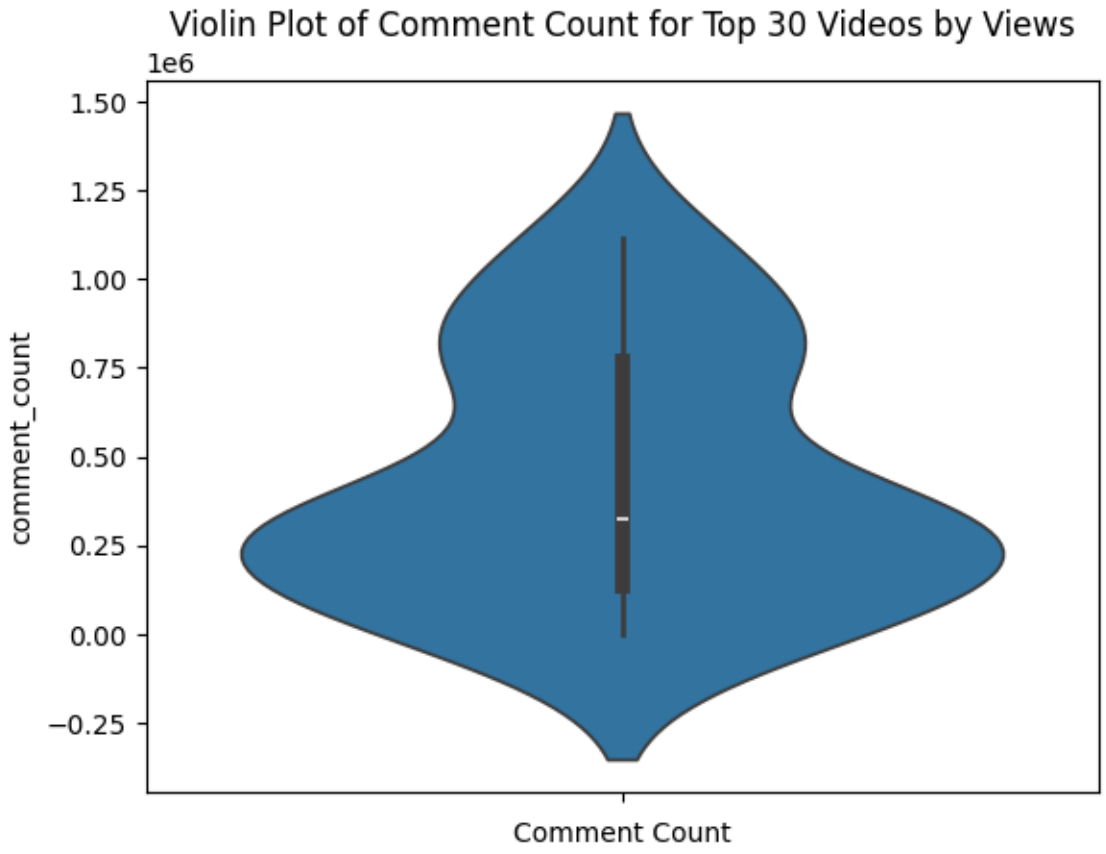


Figure 44. Biểu đồ violin mô tả số comment của top 30 video

CHƯƠNG 3: KẾT LUẬN

Thông qua việc nghiên cứu lý thuyết về các kỹ thuật trực quan hoá dữ liệu với các thư viện của Python như Matplotlib, Seaborn và Bokeh nhóm chúng em đã nắm bắt được cơ bản làm thế nào để tạo được một số dạng biểu đồ mà mỗi thư viện hỗ trợ. Qua đó, có thể áp dụng các kỹ thuật vào trong việc phân tích trực quan hoá với một bộ dữ liệu thực tế như Youtube trending.

Mặt khác, để có thể trực quan hoá tốt thì việc xử lý dữ liệu ban đầu rất cần thiết vì dữ liệu thô ban đầu có cấu trúc tương đối phức tạp, kiểu dữ liệu khi đọc tập tin có thể bị sai sót. Đôi lúc còn có trường hợp dữ liệu không đầy đủ (hiện tượng missing data). Do đó, để phân tích thống kê trực quan hoá thì nhóm chúng em cũng đã áp dụng các kỹ thuật aggregating (tập hợp dữ liệu) cũng như handling missing data (xử lý dữ liệu khuyết thiếu) để giải quyết các vấn đề của dữ liệu bị nhiễu ảnh hưởng đến thống kê và trực quan.

TÀI LIỆU THAM KHẢO

[1] "Matplotlib," [Online]. Available: https://matplotlib.org/stable/tutorials/pyplot.html. [Accessed 7 April 2024].

[2] "Seaborn," [Online]. Available: https://seaborn.pydata.org/. [Accessed 7 April 2024].

[3] "Zoho Analytics," [Online]. Available: https://www.zoho.com/vi/analytics/what-is-data-visualization.html. [Accessed 3 April 2024].

[4] A. Navlani, A. Fandango and I. Idris, "Chapter 5: Data visualization," in *Python Data Analysis 3rd edition*, Mumbai, Packt, 2021, pp. 135-189.

PHỤ LỤC

Phụ lục 1: Bộ mã nguồn thực nghiệm chương trình

- Mount Google drive

```
from google.colab import drive
drive.mount('/content/drive')

base_path = '/content/drive/MyDrive/PTDL/data_and_code'
```

- Cài đặt Pyspark trên Colab vì môi trường của Google colab không có sẵn thư viện Pyspark để thực nghiệm

```
%pip install pyspark
```

- Tạo Spark Session

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import month,year,max,sum, col,to_date
import matplotlib.pyplot as plt

def spark_session(app_name):
    spark = SparkSession.builder \
```

```
        .appName(app_name)\
        .getOrCreate()
    # To make sure that we are working with the correct version of Spark
    print("Running PySpark version: ", spark.version)
    return spark
spark = spark_session('assignment')
```

- Đọc dữ liệu với pyspark

```
dataset_path = base_path+ '/dataset/youtube_trending/CAvideos.csv' #dataset
path string
data = spark.read.csv(dataset_path, header=True)
data.printSchema()
```

- Tiền xử lý dữ liệu

```
data = data.na.drop()
data = data.filter(data["category_id"].cast("int").isNotNull())
data = data.withColumn("category_id", col("category_id").cast("integer"))
data = data.withColumn("views", col("views").cast("integer"))
data = data.withColumn("likes", col("likes").cast("integer"))
data = data.withColumn("dislikes", col("dislikes").cast("integer"))
data = data.withColumn("comment_count",
col("comment_count").cast("integer"))
```

- Trực quan hoá dữ liệu với matplotlib
    + Biểu đồ đường(line chart)

```
import numpy as np
import matplotlib.pyplot as plt
filtered_data = data.filter(data.video_id ==
'FlsCjmMhFmw').orderBy('trending_date')
views = filtered_data.select('views').rdd.flatMap(lambda x: x).collect()
x = np.linspace(0, len(views) - 1, len(views))
plt.plot(x, views)
plt.title('Biểu đồ số lượt xem')
plt.xlabel('Thứ tự ngày trending')
plt.ylabel('Số lượt xem')
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
filtered_data = data.filter(data.video_id == 'FlsCjmMhFmw')
sorted_data = filtered_data.orderBy('trending_date')
```

```
dislikes = sorted_data.select('dislikes').rdd.flatMap(lambda x: x).collect()
likes = sorted_data.select('likes').rdd.flatMap(lambda x: x).collect()
x = np.linspace(0, len(dislikes) - 1, len(dislikes))
plt.plot(x, dislikes, label='Lượt dislike')
plt.plot(x, likes, label='Lượt like')
plt.title('Biểu đồ số lượt like và dislike')
plt.xlabel('Thứ tự ngày trending')
plt.ylabel('Số lượt')
plt.legend()
plt.show()
```

+ Biểu đồ phân bổ(Scatter plot)

```
import numpy as np
import matplotlib.pyplot as plt
num_rows = data.count()
filtered_data = data.filter(data.video_id == 'VYOjWnS4cMY')
sorted_data = filtered_data.orderBy('trending_date')
likes = sorted_data.select('likes').rdd.flatMap(lambda x: x).collect()
dislikes = sorted_data.select('dislikes').rdd.flatMap(lambda x: x).collect()
plt.scatter(likes, dislikes, color='blue', marker='*', alpha=0.5)
plt.title('Biểu đồ Scatter plot giữa likes và dislikes trên YouTube')
plt.xlabel('Likes')
plt.ylabel('Dislikes')
plt.show()
```

+ Biểu đồ tròn (pie plot)

```
import matplotlib.pyplot as plt
category_counts = data.groupBy('category_id').count().collect()
category_counts_dict = {row['category_id']: row['count'] for row in
category_counts}
video_counts = list(category_counts_dict.values())
category_labels = list(category_counts_dict.keys())
plt.figure(figsize=(8, 8))
patches, texts = plt.pie(video_counts, startangle=140, textprops={'color':
"white"})
percent_labels = ['{0} - {1:1.1f} %'.format(i, j / sum(video_counts) * 100) for
i, j in zip(category_labels, video_counts)]
plt.legend(patches, percent_labels, loc="best")
plt.title('Biểu đồ tròn tỷ lệ số video theo category_id')
plt.show()
```

+ Biểu đò cột (bar plot)

```
import matplotlib.pyplot as plt
```

```
category_counts = data.groupBy('category_id').count().collect()
category_counts_dict = {row['category_id']: row['count'] for row in
category_counts}
sorted_category_counts = sorted(category_counts_dict.items())
category_labels, video_counts = zip(*sorted_category_counts)
plt.figure(figsize=(12, 6))
plt.bar(range(len(category_labels)), video_counts, color='blue')
plt.xticks(range(len(category_labels)), category_labels, rotation=90)
plt.title('Biểu đồ cột số lượng video theo category_id')
plt.xlabel('Category ID')
plt.ylabel('Số lượng video')
plt.tight_layout()
plt.show()
```

- Trực quan hoá dữ liệu với bokeh

    + Biểu đồ scatter (với bokeh)

```
from bokeh.plotting import figure, show, output_notebook
dataframe_pandas = data.toPandas()
output_notebook()
p = figure(title="Video Views vs Likes (2017-11-14)",
x_axis_label="Views", y_axis_label="Likes")
p.circle(dataframe_pandas['views'], dataframe_pandas['likes'], size=10,
color="blue", alpha=0.6)
show(p)
```

    + Biểu đồ cột (với bokeh)

```
# Assuming 'category_id' column exists in dataframe_pandas
video_count_per_category = dataframe_pandas['category_id'].value_counts()
from bokeh.plotting import figure, show, output_notebook
output_notebook()
# Use a bar chart function for visualization
x_range = video_count_per_category.index.astype(str).tolist()
p = figure(
    x_range=x_range,  # List of category IDs
    height=300,
    title="Number of Videos per Category (2017-11-14)",
    toolbar_location=None,  # Remove toolbar for cleaner visuals (optional)
    tools=""  # Hide zooming/panning tools (optional)
)

# Create vertical bars with category names on x-axis and video count on
```

```
y-axis
p.vbar(
    x=video_count_per_category.index,
    top=video_count_per_category.values,
    width=0.8,
    legend_label="Video Count",
    bottom=0,  # Set bottom to 0 for starting bars from the axis
    color='skyblue'  # Set bar color (optional)
)

# Customize axis labels
p.xaxis.axis_label = "Category ID"
p.yaxis.axis_label = "Video Count"

# Display the chart
show(p)
```

- Trực quan hoá dữ liệu với seaborn
    + Biểu đồ scatter

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt


# Đọc dữ liệu từ tệp CSV bằng Pandas

dataset_path = base_path+ '/youtube_trending/CAvideos.csv'

data = pd.read_csv(dataset_path)


# Hiển thị 5 dòng đầu tiên của dữ liệu để kiểm tra

print(data.head())


# Ví dụ về việc sử dụng Seaborn để tạo biểu đồ

sns.scatterplot(x="views", y="likes", data=data)


# Hiển thị biểu đồ
```

```
plt.show()
```

+ lmplot

```
sns.lmplot(data=data, x='views', y='likes')
plt.xlabel('Views')
plt.ylabel('Likes')
plt.title('Linear Regression Plot of Views vs Likes')
plt.show()
```

+ Bar plot

```
sns.barplot(data=data, x='category_id', y='views', hue='ratings_disabled', estimator=sum)
plt.xlabel('Category ID')
plt.ylabel('Total')
plt.title('Total Views with Ratings Disabled vs Enabled by Category')
plt.legend(title='Ratings Disabled', labels=['Enabled', 'Disabled'])
plt.show()
```

+ Pair plot

```
sns.pairplot(data=data, vars=['views', 'likes', 'dislikes', 'comment_count'], hue='category_id', diag_kind='kde', palette='Set2')
plt.suptitle('Pairplot of Views, Likes, Dislikes, and Comment Count by Category ID', y=1.02)
plt.show()
```

+ Box plot

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Sắp xếp DataFrame theo số lượt xem giảm dần
sorted_data = data.sort_values(by='views', ascending=False)

# Chọn 10 video có số lượt xem cao nhất
top_10_videos = sorted_data.head(10)

# Tạo biểu đồ hộp của số lượt xem cho 10 video có lượt xem cao nhất, phân loại theo 'category_id' và sử dụng màu sắc khác nhau cho 'ratings_disabled'
sns.boxplot(data=top_10_videos, x='category_id', y='views', hue='ratings_disabled', palette='Set2')

# Đặt nhãn và tiêu đề cho biểu đồ
plt.xlabel('Category ID')
plt.ylabel('Views')
plt.title('Box Plot of Views for Top 10 Videos by Category ID with Ratings Disabled')

# Hiển thị chú thích
plt.legend(title='Ratings Disabled', labels=['Enabled', 'Disabled'])

# Hiển thị biểu đồ
plt.show()
```

+ KDE plot

```
# Sắp xếp DataFrame theo số lượt xem giảm dần
sorted_data = data.sort_values(by='views', ascending=False)
```

```
# Chọn 10 video có số lượt xem cao nhất
top_10_videos = sorted_data.head(10)

# Tạo KDE plot cho cột 'views' của 10 video có lượt xem cao nhất
sns.kdeplot(data=top_10_videos['views'])

# Đặt nhãn và tiêu đề cho biểu đồ
plt.xlabel('Views')
plt.ylabel('Density')
plt.title('KDE Plot of Views for Top 10 Videos')

# Hiển thị biểu đồ
plt.show()
```

+ Violin plot

```
# Sắp xếp DataFrame theo số lượt xem giảm dần
sorted_data = data.sort_values(by='views', ascending=False)

# Chọn 30 video có số lượt xem cao nhất
top_30_videos = sorted_data.head(30)

# Tạo biểu đồ violin cho cột 'comment_count' của 30 video có lượt xem cao nhất
sns.violinplot(data=top_30_videos['comment_count'])

# Đặt nhãn và tiêu đề cho biểu đồ
plt.xlabel('Comment Count')
plt.title('Violin Plot of Comment Count for Top 30 Videos by Views')
```

```
# Hiển thị biểu đồ
plt.show()
```

Phụ lục 2: Bảng phân chia công việc

STT	Công việc	Người thực hiện
1	Viết báo cáo phần mở đầu và chương cơ sở lý thuyết	Nguyễn Công Thọ
2	Viết báo cáo phần giới thiệu bộ dữ liệu và tiền xử lý dữ liệu	Lê Công Minh
3	Thực hiện thực nghiệm với Matplotlib và báo cáo kết quả	Lê Công Minh
4	Thực hiện thực nghiệm với Seaborn và báo cáo kết quả	Lê Minh Phúc
5	Thực hiện thực nghiệm với Bokeh và báo cáo kết quả	Nguyễn Công Thọ
6	Viết kết luận và chỉnh sửa báo cáo theo quy định	Lê Minh Phúc