**BÁO CÁO NHỎ PHẦN ĐỒ THỊ**

**1.Hiển Thị Biểu Đồ**

def hien\_thi\_full\_screen(fig):

full\_screen = tk.Toplevel()

full\_screen.title("Xem biểu đồ toàn màn hình")

full\_screen.state("zoomed")

canvas = FigureCanvasTkAgg(fig, master=full\_screen)

canvas.draw()

canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=True)

tk.Button(full\_screen, text="Thoát", command=full\_screen.destroy).pack(pady=10)

* 1. **Giải Thích Chi Tiết**

Hàm hien\_thi\_full\_screen(fig) dùng để hiển thị biểu đồ toàn màn hình trong một cửa sổ giao diện người dùng (UI) sử dụng thư viện Tkinter và matplotlib. Sau đây là giải thích chi tiết từng dòng

1.1.1. def hien\_thi\_full\_screen(fig):

* Đây là phần định nghĩa hàm hien\_thi\_full\_screen, nhận một tham số đầu vào fig.
* fig: Là một đối tượng biểu đồ (Figure) được tạo bởi thư viện matplotlib.

1.1.2. full\_screen = tk.Toplevel()

* Tạo một cửa sổ mới (con) trong giao diện chính của Tkinter bằng lớp Toplevel.
* full\_screen: Đại diện cho cửa sổ này. Nó cho phép hiển thị nội dung (biểu đồ) độc lập với cửa sổ chính.

1.1.3. full\_screen.title("Xem biểu đồ toàn màn hình")

* Thiết lập tiêu đề của cửa sổ mới là "Xem biểu đồ toàn màn hình"

1.1.4. full\_screen.state("zoomed")

* Thiết lập trạng thái của cửa sổ con thành "zoomed", tức là mở toàn màn hình trên hệ điều hành.

1.1.5. canvas = FigureCanvasTkAgg(fig, master=full\_screen)

* FigureCanvasTkAgg: Là một lớp của matplotlib cho phép nhúng biểu đồ (Figure) vào trong giao diện Tkinter.
* fig: Biểu đồ cần hiển thị.
* master=full\_screen: Gắn biểu đồ này vào cửa sổ full\_screen.

1.1.6. canvas.draw()

* Gọi phương thức draw() để vẽ và hiển thị biểu đồ fig lên canvas.

1.1.7. canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=True)

* get\_tk\_widget(): Trả về widget Tkinter được nhúng từ FigureCanvasTkAgg.
* pack(fill=tk.BOTH, expand=True):
* fill=tk.BOTH: Yêu cầu widget lấp đầy cả chiều ngang và chiều dọc trong bố cục.
* expand=True: Cho phép widget mở rộng khi cửa sổ được thay đổi kích thước.

1.1.8. tk.Button(full\_screen, text="Thoát", command=full\_screen.destroy).pack(pady=10)

* text="Thoát": Hiển thị chữ "Thoát" trên nút.
* command=full\_screen.destroy: Khi nhấn nút, cửa sổ full\_screen sẽ bị đóng.
* pack(pady=10): Bố trí nút với khoảng cách trên và dưới là 10 pixel

**2. Vẽ Biểu Đồ Cột Chồng**

def ve\_do\_thi\_cot():

for widget in khung\_4.winfo\_children():

widget.destroy()

if 'Student Country' in du\_lieu.columns and 'Type of Answer' in du\_lieu.columns:

grouped\_data = du\_lieu.groupby(['Student Country', 'Type of Answer']).size().unstack(fill\_value=0)

fig, ax = plt.subplots(figsize=(8, 5))

grouped\_data.plot(kind='bar', stacked=True, ax=ax, color=['red', 'green'])

ax.set\_title('Biểu đồ cột chồng thể hiện Student Country và Type of Answer')

ax.set\_xlabel('Student Country')

ax.set\_ylabel('Số lượng')

ax.legend(title='Type of Answer', labels=['Sai (0)', 'Đúng (1)'])

canvas = FigureCanvasTkAgg(fig, master=khung\_4)

canvas.draw()

canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=True)

tk.Button(khung\_4, text="Phóng to", command=lambda: hien\_thi\_full\_screen(fig)).pack(pady=10)

**2.1. Giải Thích Chi Tiết**

Hàm ve\_do\_thi\_cot() dùng để vẽ biểu đồ cột chồng (stacked bar chart) trong giao diện Tkinter dựa trên dữ liệu từ DataFrame du\_lieu. Sau đây là giải thích chi tiết từng dòng

2.1.1. for widget in khung\_4.winfo\_children(): widget.destroy()

* khung\_4: Là một khung (Frame) trong giao diện Tkinter dùng để chứa biểu đồ.
* khung\_4.winfo\_children(): Lấy danh sách các widget (thành phần giao diện con) hiện có trong khung\_4.
* widget.destroy(): Xóa từng widget con trong khung\_4.

2.1.2. if 'Student Country' in du\_lieu.columns and 'Type of Answer' in du\_lieu.columns:

* Kiểm tra xem DataFrame du\_lieu có chứa hai cột Student Country và Type of Answer hay không

2.1.3. grouped\_data = du\_lieu.groupby(['Student Country', 'Type of Answer']).size().unstack(fill\_value=0)

* du\_lieu.groupby(['Student Country', 'Type of Answer']): Nhóm dữ liệu theo hai cột Student Country và Type of Answer.
* .size(): Đếm số lượng phần tử trong từng nhóm.
* unstack(fill\_value=0):
* fill\_value=0: Điền giá trị 0 cho các ô trống.

2.1.4. fig, ax = plt.subplots(figsize=(8, 5))

* figsize=(8, 5): Kích thước biểu đồ là 8 inch chiều ngang và 5 inch chiều cao.

2.1.5. grouped\_data.plot(kind='bar', stacked=True, ax=ax, color=['red', 'green'])

* kind='bar': Vẽ biểu đồ dạng cột.
* stacked=True: Các cột được chồng lên nhau để thể hiện tổng số lượng.
* ax=ax: Vẽ biểu đồ trên trục đã tạo ở bước trước.
* color=['red', 'green']: Đặt màu cho các phần cột tương ứng với các giá trị của Type of Answer.
* Màu đỏ cho giá trị "Sai (0)".
* Màu xanh lá cho giá trị "Đúng (1)".

2.1.6. ax.set\_title('Biểu đồ cột chồng thể hiện Student Country và Type of Answer')

* Đặt tiêu đề cho biểu đồ

2.1.7. ax.set\_xlabel('Student Country')

* Đặt nhãn cho trục hoành là "Student Country".

2.1.8. ax.set\_ylabel('Số lượng')

* Đặt nhãn cho trục tung là "Số lượng".

2.1.9. ax.legend(title='Type of Answer', labels=['Sai (0)', 'Đúng (1)'])

* title='Type of Answer': Đặt tiêu đề cho phần chú thích (legend).
* labels=['Sai (0)', 'Đúng (1)']: Gắn nhãn cho các phần chú thích.
* "Sai (0)": Đại diện cho phần màu đỏ.

"Đúng (1)": Đại diện cho phần màu xanh lá.

2.1.10. canvas = FigureCanvasTkAgg(fig, master=khung\_4)

* FigureCanvasTkAgg: Nhúng biểu đồ fig (tạo bởi matplotlib) vào trong khung khung\_4 của giao diện Tkinter.

2.1.11. canvas.draw()

* Vẽ biểu đồ trên canvas.

2.1.12. canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=True)

* get\_tk\_widget(): Lấy widget Tkinter tương ứng với canvas.
* pack(fill=tk.BOTH, expand=True): Bố trí widget để lấp đầy cả chiều ngang và chiều dọc của khung\_4.

2.1.13. tk.Button(khung\_4, text="Phóng to", command=lambda: hien\_thi\_full\_screen(fig)).pack(pady=10)

* Tạo một nút bấm trong khung\_4.
* text="Phóng to": Hiển thị chữ "Phóng to" trên nút.
* command=lambda: hien\_thi\_full\_screen(fig): Khi bấm nút, gọi hàm hien\_thi\_full\_screen(fig) để mở biểu đồ ở chế độ toàn màn hình.
* pack(pady=10): Bố trí nút với khoảng cách trên và dưới là 10 pixel.

**3.Vẽ Biểu Đồ Tròn**

def ve\_bieu\_do\_tron():

for widget in khung\_4.winfo\_children():

widget.destroy()

if 'Question Level' in du\_lieu.columns:

data = du\_lieu['Question Level'].value\_counts()

fig, ax = plt.subplots(figsize=(6, 6))

ax.pie(data, labels=data.index, autopct='%1.1f%%', startangle=90)

ax.set\_title("Biểu đồ tròn thể hiện Question Level")

canvas = FigureCanvasTkAgg(fig, master=khung\_4)

canvas.draw()

canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=True)

tk.Button(khung\_4, text="Phóng to", command=lambda: hien\_thi\_full\_screen(fig)).pack(pady=10)

**3.1.Giải Thích Chi Tiết**

3.1.1. for widget in khung\_4.winfo\_children(): widget.destroy()

* khung\_4: Là một khung (Frame) trong giao diện Tkinter dùng để chứa biểu đồ.
* khung\_4.winfo\_children(): Trả về danh sách tất cả các widget (thành phần giao diện con) hiện có trong khung\_4.
* widget.destroy(): Xóa từng widget con khỏi khung\_4.

3.1.2. if 'Question Level' in du\_lieu.columns:

* Kiểm tra xem DataFrame du\_lieu có chứa cột Question Level hay không.

3.1.3. data = du\_lieu['Question Level'].value\_counts()

* du\_lieu['Question Level']: Truy xuất cột Question Level từ DataFrame du\_lieu.
* value\_counts(): Đếm số lượng xuất hiện của từng giá trị trong cột Question Level.

3.1.4. fig, ax = plt.subplots(figsize=(6, 6))

* figsize=(6, 6): Kích thước biểu đồ là 6 inch chiều ngang và 6 inch chiều cao (tạo biểu đồ vuông

3.1.5. ax.pie(data, labels=data.index, autopct='%1.1f%%', startangle=90)

* data: Dữ liệu để hiển thị (số lượng của từng giá trị trong Question Level).
* labels=data.index: Nhãn tương ứng cho từng phần của biểu đồ, sử dụng các giá trị trong Question Level.
* autopct='%1.1f%%': Hiển thị tỷ lệ phần trăm trên mỗi phần của biểu đồ, định dạng với 1 chữ số thập phân.
* starrangle=90: Bắt đầu vẽ biểu đồ từ góc 90 độ (hướng trên cùng), giúp hiển thị dễ nhìn hơn.

3.1.6. ax.set\_title("Biểu đồ tròn thể hiện Question Level")

* Đặt tiêu đề cho biểu đồ là "Biểu đồ tròn thể hiện Question Level"

3.1.7. canvas = FigureCanvasTkAgg(fig, master=khung\_4)

* FigureCanvasTkAgg: Nhúng biểu đồ fig (tạo bởi matplotlib) vào trong khung khung\_4 của giao diện Tkinter.
* master=khung\_4: Gắn canvas vào khung\_4.

3.1.8. canvas.draw()

* Gọi phương thức draw() để vẽ biểu đồ fig lên canvas.

3.1.9. canvas.get\_tk\_widget().pack(fill=tk.BOTH, expand=True)

* get\_tk\_widget(): Lấy widget Tkinter tương ứng với canvas.
* pack(fill=tk.BOTH, expand=True):
* fill=tk.BOTH: Lấp đầy cả chiều ngang và chiều dọc của khung\_4expand=True: Cho phép widget mở rộng khi khung thay đổi kích thước.

3.1.10. tk.Button(khung\_4, text="Phóng to", command=lambda: hien\_thi\_full\_screen(fig)).pack(pady=10)

* Tạo một nút bấm trong khung khung\_4.
* text="Phóng to": Hiển thị chữ "Phóng to" trên nút.
* command=lambda: hien\_thi\_full\_screen(fig): Khi bấm nút, gọi hàm hien\_thi\_full\_screen(fig) để mở biểu đồ ở chế độ toàn màn hình.
* pack(pady=10): Bố trí nút với khoảng cách trên và dưới là 10 pixel.