BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG

PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI

BỘ MÔN CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI

Giảng viên:	Ths. Vũ Thị Hạnh
Sinh viên thực hiện:	2351267280 Đoàn Anh Vũ 2351267274 Nguyễn Hữu Tuấn Phát 2251068262 Lê Ngọc Tiền 2251068284 Phan Trần Tường Vy
Lớp:	S26-65TTNT

TP. Hồ Chí Minh, ngày … tháng … năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày … tháng … năm 2025

Chữ ký của giảng viên

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin bày tỏ lòng biết ơn sâu sắc và chân thành nhất đến quý thầy cô Bộ môn Công nghệ Thông tin – Phân hiệu Trường Đại học Thủy lợi, những người đã truyền đạt kiến thức nền tảng vững chắc và tạo điều kiện thuận lợi nhất cho nhóm trong suốt quá trình học tập và nghiên cứu.

Đặc biệt, nhóm em xin gửi lời cảm ơn chân thành đến cô Vũ Thị Hạnh. Với sự hướng dẫn tận tâm, những định hướng chuyên môn quý báu về lĩnh vực khai phá dữ liệu và sự khích lệ của Cô, nhóm em đã có thêm động lực để vượt qua những khó khăn trong việc xử lý tín hiệu điện não đồ (EEG) phức tạp và tối ưu hóa các thuật toán học máy để hoàn thành đề tài này.

Bên cạnh đó, nhóm cũng xin cảm ơn cộng đồng Kaggle cùng tác giả bộ dữ liệu "Alcoholics Dataset" đã cung cấp nguồn tài nguyên quý giá. Những dữ liệu thực nghiệm về sóng não này đã giúp nhóm có cơ sở khoa học để triển khai mô hình phân tích và dự đoán tình trạng nghiện rượu một cách khách quan nhất.

Mặc dù đã dành nhiều tâm huyết cho bài báo cáo, nhưng do hạn chế về mặt thời gian và kinh nghiệm trong lĩnh vực Phân tích dữ liệu y sinh (Biomedical Data Analysis), bài làm chắc chắn không tránh khỏi những thiếu sót. Nhóm em rất mong nhận được những ý kiến đóng góp từ Cô để nhóm có cơ hội hoàn thiện kiến thức và phát triển kỹ năng nghiên cứu chuyên sâu trong tương lai.

Nhóm em xin chân thành cảm ơn!

Mục Lục

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

MỞ ĐẦU

A. GIỚI THIỆU ĐỀ TÀI

1. Bối cảnh đề tài

Trong khuôn khổ môn học Data Mining, việc tiếp cận với các bài toán thực tế thông qua các bộ dữ liệu quy chuẩn là yêu cầu trọng tâm để củng cố kiến thức lý thuyết. Đề tài **"Phân tích và phân loại tình trạng nghiện rượu dựa trên tín hiệu điện não đồ (EEG)"** giúp sinh viên rèn luyện kỹ năng giải quyết bài toán xử lý tín hiệu số và phân tích dữ liệu y sinh – một trong những ứng dụng thực tiễn và nhân văn nhất của trí tuệ nhân tạo trong việc hỗ trợ chẩn đoán y khoa.

Nghiện rượu là một vấn đề xã hội và sức khỏe nghiêm trọng, gây ra những thay đổi đáng kể trong hoạt động chức năng của não bộ. Việc sử dụng các mô hình học máy để tự động nhận diện các đặc trưng khác biệt giữa nhóm người nghiện rượu và nhóm đối chứng không chỉ giúp hiểu rõ hơn về tác động của chất kích thích lên hệ thần kinh mà còn mở ra hướng đi cho các hệ thống hỗ trợ chẩn đoán tự động trong tương lai.

2. Ý nghĩa và mục tiêu thực hiện

Việc thực hiện đề tài này không chỉ dừng lại ở việc hoàn thành nội dung môn học mà còn mang lại những giá trị cụ thể:

> **Về kiến thức:** Giúp nhóm nắm vững quy trình xử lý dữ liệu tín hiệu đa chiều (từ 64 điện cực não bộ) và hiểu sâu về cơ chế hoạt động của các thuật toán phân loại trên dữ liệu chuỗi thời gian (Time-series Data). Nhóm có cơ hội tìm hiểu về cách trích xuất đặc trưng từ các miền tần số sóng não như Alpha, Beta, Theta, Delta.
>
> **Về kỹ năng:** Thực hành kỹ thuật tiền xử lý dữ liệu phức tạp (khử nhiễu tín hiệu, chuẩn hóa đơn vị điện áp), xây dựng và tối ưu hóa các mô hình học máy (như **Random Forest, SVM** hoặc mạng **RNN/LSTM**) để đạt được độ chính xác cao trong việc phân loại trạng thái thần kinh.
>
> **Về thực tiễn:** Tiếp cận với bài toán chuyển đổi số trong y tế và tâm thần học. Hiểu được cách trí tuệ nhân tạo có thể hỗ trợ các chuyên gia y tế chẩn đoán các tác động tiêu cực của chất kích thích lên hệ thần kinh trung ương một cách tự động và khách quan, dựa trên các bằng chứng khoa học về sóng não.

B. YÊU CẦU BÀI TOÁN

Dựa trên yêu cầu của giảng viên và đặc điểm của bộ dữ liệu được cung cấp, bài toán được xác định cụ thể như sau:

1. Đối tượng và phạm vi nghiên cứu

Dữ liệu đầu vào: Bộ dữ liệu Alcoholics Dataset thu thập tín hiệu điện não đồ (EEG) từ 64 điện cực gắn trên da đầu. Dữ liệu bao gồm 2 nhóm đối tượng chính: nhóm người nghiện rượu (Alcoholic) và nhóm đối chứng (Control) trong các điều kiện kích thích thị giác khác nhau (S1, S2 match, S2 no match).

2. Các yêu cầu kỹ thuật cần đạt được

Để đáp ứng tiêu chuẩn của môn học, nhóm cần giải quyết các yêu cầu sau:

Tiền xử lý dữ liệu : Thực hiện làm sạch dữ liệu, xử lý các giá trị thiếu hoặc nhiễu tín hiệu.

- Chuẩn hóa đơn vị điện áp.
- Trích xuất đặc trưng: Chuyển đổi dữ liệu thô sang các đặc trưng thống kê (mean, std, entropy) hoặc đặc trưng miền tần số (sử dụng biến đổi Fourier hoặc Wavelet) để bắt lấy các nhịp sóng não như Alpha, Beta.

Thiết kế mô hình: Xây dựng các mô hình học máy phù hợp để xử lý dữ liệu bảng hoặc dữ liệu chuỗi. Các mô hình có thể cân nhắc bao gồm:

- Truyền thống: Random Forest, Support Vector Machine (SVM), XGBoost.
- Deep Learning: Mạng Neural nhân tạo (ANN) hoặc mạng học chuỗi (LSTM/GRU) để xử lý tính chất thời gian của sóng não.

Tối ưu hóa: Lựa chọn hàm mất mát và các bộ tối ưu hóa (Adam, SGD) phù hợp để cực tiểu hóa sai số và tránh hiện tượng Overfitting do dữ liệu y sinh thường có độ nhiễu cao.

Đánh giá kết quả: Mô hình được đánh giá dựa trên các chỉ số: Accuracy, Precision, Recall và F1-Score. Đặc biệt, trong bài toán y sinh, chỉ số Recall cần được chú trọng để tránh bỏ sót các trường hợp có nguy cơ bệnh lý.

C. GIỚI THIỆU TẬP DỮ LIỆU

Bộ dữ liệu được sử dụng trong đề tài là "EEG Database Data Set" , được cung cấp bởi UCI Machine Learning Repository và phổ biến trên Kaggle. Đây là bộ dữ liệu chuẩn trong lĩnh vực thần kinh học, được dùng để nghiên cứu khả năng di truyền của tình trạng nghiện rượu thông qua các phản ứng điện sinh lý của não bộ.

1. Quy mô và cấu trúc dữ liệu

Đối tượng nghiên cứu: Gồm hai nhóm đối tượng chính:

Nhóm Alcoholic : Những người được chẩn đoán nghiện rượu.

Nhóm Control: Nhóm đối chứng (người không nghiện rượu).

Các loại thử nghiệm (Paradigms): Dữ liệu ghi lại phản ứng của não qua 3 loại kích thích thị giác:

S1 obj: Hiển thị một hình ảnh duy nhất.

S2 match: Hiển thị hai hình ảnh giống nhau.

S2 nomatch: Hiển thị hai hình ảnh khác nhau.

Tần số lấy mẫu: 256 Hz (tương đương 256 điểm dữ liệu mỗi giây).

Số lượng kênh tín hiệu: Tín hiệu được thu thập từ 64 điện cực (channels) đặt trên da đầu theo hệ thống quốc tế 10-20.

STT	Nhóm Vùng Não	Danh sách các kênh	Chức năng giải phẫu thần kinh liên quan
01	Vùng Trước Trán (Pre-frontal)	FP1, FP2, FPZ	Kiểm soát nhận thức, lập kế hoạch và điều tiết cảm xúc.
02	Vùng Trán (Frontal)	F1, F2, F3, F4, F5, F6, F7, F8, FZ	Liên quan đến khả năng vận động, trí nhớ làm việc và giải quyết vấn đề.
03	Vùng Trán - Trung Tâm	FC1, FC2, FC3, FC4, FC5, FC6, FCZ	Vùng chuyển tiếp điều khiển hoạt động cơ bắp và phản xạ.
04	Vùng Trung Tâm (Central)	C1, C2, C3, C4, C5, C6, CZ	Xử lý cảm giác thân thể và điều khiển vận động chính.
05	Vùng Thái Dương (Temporal)	T7, T8, TP7, TP8	Xử lý thính giác, ngôn ngữ và lưu trữ trí nhớ dài hạn.
06	Vùng Trung Tâm - Đỉnh	CP1, CP2, CP3, CP4, CP5, CP6, CPZ	Tích hợp các luồng thông tin cảm giác từ nhiều nguồn khác nhau.
07	Vùng Đỉnh (Parietal)	P1, P2, P3, P4, P5, P6, P7, P8, PZ	Định hướng không gian, nhận biết vật thể và tính toán.

STT	Nhóm Vùng Não	Danh sách các kênh	Chức năng giải phẫu thần kinh liên quan
08	Vùng Chẩm (Occipital)	O1, O2, OZ	Trung tâm tiếp nhận và xử lý tín hiệu thị giác từ mắt.
09	Vùng Phụ Trợ (Extended)	AF1, AF2, AF7, AF8, AFZ, PO1, PO2, PO7, PO8, POZ	Các điểm đo bổ sung giúp tăng độ phân giải cho vùng nhận thức cao và thị giác.

2. Tổ chức tập dữ liệu

Dữ liệu được tổ chức dưới dạng các tệp tin lưu trữ chuỗi thời gian của điện áp. Cấu trúc bao gồm:

Tập đầy đủ : Chứa 122 đối tượng, mỗi đối tượng thực hiện khoảng 100 thử nghiệm

Tập huấn luyện : Sử dụng các phiên bản đã được gán nhãn sẵn (Alcoholic/Control) để mô hình học các mẫu sóng não đặc trưng.

Tập kiểm tra : Dùng để đánh giá khả năng dự đoán của mô hình trên các đối tượng mới chưa từng xuất hiện trong quá trình huấn luyện.

3. Phân tích đặc điểm của từng bệnh

Bài toán tập trung vào việc phân biệt sự khác biệt về năng lượng sóng não giữa hai nhóm

STT	Nhóm đối tượng	Ký hiệu	Đặc điểm nhận biết qua nghiên cứu
1	Người nghiện rượu	a (Alcoholic)	Thường có công suất sóng *P300* thấp hơn khi tiếp nhận kích thích thị giác. Khả năng ức chế thần kinh kém.
2	Nhóm đối chứng	c (Control)	Biên độ sóng *P300* rõ rệt, khả năng xử lý thông tin và phản xạ thần kinh ổn định.

STT	Loại sóng	Dải tần số	Trạng thái liên quan
1	Delta	0.5 - 4 Hz	Ngủ sâu, tổn thương não bộ.
2	Theta	4 - 8 Hz	Thư giãn sâu, buồn ngủ, hoặc trạng thái lo âu.
3	Alpha	8 - 12 Hz	Trạng thái nghỉ ngơi nhưng vẫn tỉnh táo.
4	Beta	12 - 30 Hz	Trạng thái tập trung cao, suy nghĩ logic, hoặc căng thẳng.

Tổng hợp triệu chứng dữ liệu: Dữ liệu EEG không thể quan sát bằng mắt thường mà cần thông qua phân tích đặc trưng. Các đặc trưng như công suất phổ , entropy, hoặc tương quan giữa các vùng não sẽ là chìa khóa để mô hình học máy tách biệt được hai nhóm đối tượng một cách chính xác nhất.

Chương 1: Tiền xử lý dữ liệu

1. Kiểm tra missing value

```
df.isna().sum()
```

```
Unnamed: 0              0
trial number            0
sensor position         0
sample num              0
sensor value            0
subject identifier      0
matching condition      0
channel                 0
name                    0
time                    0
source_file             0
dtype: int64
```

Như kết quả thực nghiệm thu được, tất cả các trường dữ liệu quan trọng đều không có giá trị trống. Điều này khẳng định bộ dữ liệu có chất lượng cao

2. Chuẩn hóa và Mã hóa dữ liệu

```
df_feature = df.copy()
df_feature['label'] = df_train_raw['subject identifier'].map({'a': 1, 'c': 0})
df_feature.columns = df_feature.columns.str.strip().str.lower().str.replace(" ", "_")
df_feature.rename(columns={'sensor_position': 'channel', 'channel': 'channel_id'}, inplace=True)
```

Chuẩn hóa cấu trúc DataFrame và chuyển đổi các nhãn mục tiêu sang định dạng số để máy tính có thể thực hiện tính toán.

Mã hóa nhãn: Nhóm chuyển đổi trường subject identifier thành nhãn nhị phân . Trong đó, nhóm đối tượng nghiện rượu ('a') được gán giá trị 1 và nhóm đối chứng ('c') được gán giá trị 0.

Làm sạch cấu trúc cột: Để thuận tiện cho việc truy vấn và lập trình, nhóm tiến hành xử lý tên các cột thuộc tính:

- Loại bỏ khoảng trắng thừa đầu và cuối (strip).
- Chuyển toàn bộ tên cột về dạng chữ thường (lower).
- Thay thế khoảng trắng giữa các từ bằng dấu gạch dưới (_).

Tái cấu trúc định danh: Nhóm thực hiện đổi tên cột sensor_position thành channel và channel cũ thành channel_id để phân biệt rõ ràng giữa tên vị trí điện cực vật lý và mã số định danh kỹ thuật.

3. Trích xuất đặc trưng thống kê.

```
df_features = (df_feature.groupby(['source_file', 'channel'])['sensor_value'].agg(['mean', 'std', 'min', 'max']))
label = df_feature[['source_file', 'label']].drop_duplicates().set_index('source_file')
df_features = df_features.join(label, on='source_file')
df_features = df_features.reset_index()
df_features.head()
```

	source_file	channel	mean	std	min	max	label
0	Data1.csv	AF1	3.346762	4.672660	-8.494	15.432	1
1	Data1.csv	AF2	3.770840	4.875914	-11.078	14.801	1
2	Data1.csv	AF7	6.015152	9.773413	-16.856	27.578	1
3	Data1.csv	AF8	6.032277	11.563581	-20.762	36.855	1
4	Data1.csv	AFZ	2.837496	4.274873	-10.264	12.197	1

Đây là bước then chốt nhằm chuyển đổi các tín hiệu điện não đồ biến thiên theo thời gian thành các thuộc tính số học mà mô hình Machine Learning có thể học được.

Thao tác : Nhóm sử dụng phương thức .groupby() dựa trên hai trường source_file (từng thử nghiệm) và channel.

Các đặc trưng được trích xuất: Với mỗi kênh tại mỗi lần thử nghiệm, nhóm tính toán 4 giá trị thống kê đặc trưng cho biên độ tín hiệu:

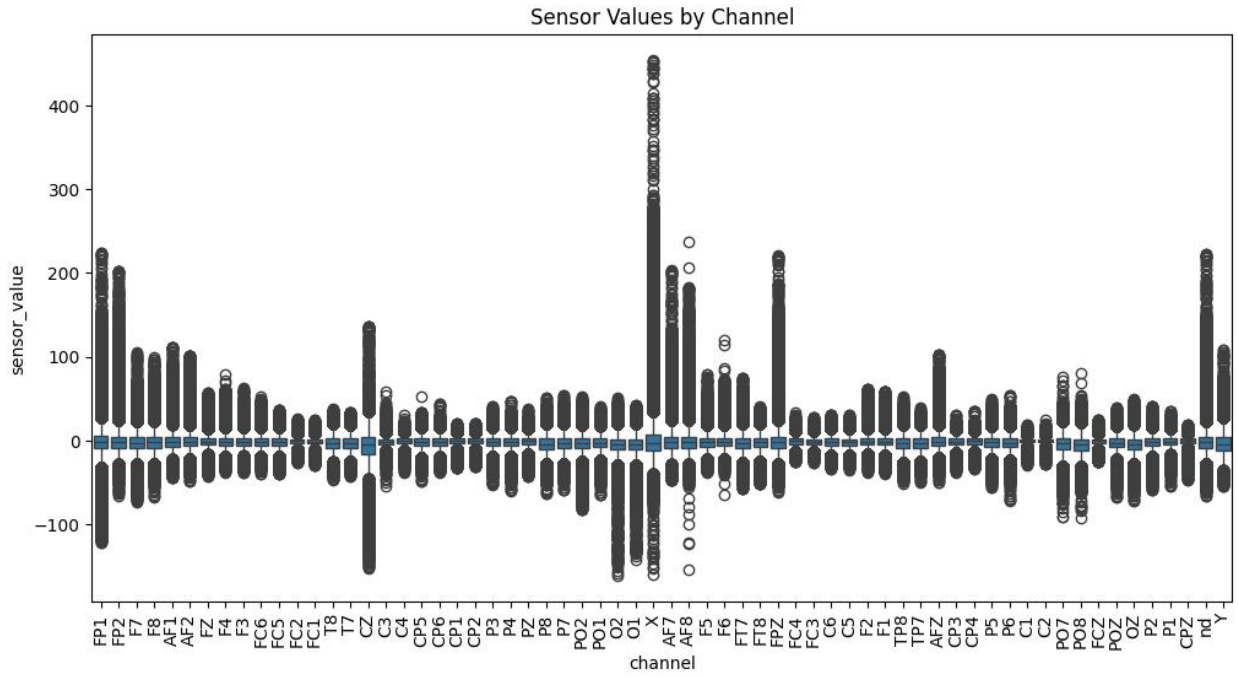
- Mean (Trung bình): Thể hiện mức điện thế trung bình của kênh.
- Std (Độ lệch chuẩn): Phản ánh mức độ biến động/nhiễu của sóng não.
- Min & Max: Xác định dải biên độ cực trị của tín hiệu.

Kết quả: Dữ liệu được rút gọn đáng kể nhưng vẫn giữ được các đặc điểm quan trọng nhất về mặt năng lượng tín hiệu. DataFrame cuối cùng df_features đã sẵn sàng để đưa vào các mô hình phân loại.

4. Kiểm tra giá trị ngoại lệ

```python
#boxplot to check outliers
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
sns.boxplot(x='channel', y='sensor_value', data=df_feature)
plt.title('Sensor Values by Channel')
plt.xticks(rotation=90)
plt.show()
```



Để đánh giá sự phân phối của tín hiệu trên 64 kênh điện cực, nhóm đã sử dụng biểu đồ Boxplot thông qua thư viện Seaborn.

Phân tích biểu đồ: Biểu đồ "Sensor Values by Channel" cho thấy sự khác biệt rõ rệt về biên độ và độ biến động giữa các vị trí đặt điện cực. Một số kênh xuất hiện nhiều giá trị ngoại lệ (điểm đen phía ngoài râu biểu đồ), điều này là đặc trưng của tín hiệu EEG khi bị ảnh hưởng bởi nhiễu sinh lý như chớp mắt hoặc cử động cơ.

Tuy nhiên, đây là đặc trưng phổ biến của tín hiệu EEG và không được xem là lỗi dữ liệu. Do đó, thay vì loại bỏ các giá trị này, nghiên cứu sử dụng RobustScaler nhằm giảm ảnh hưởng của ngoại lai trong quá trình chuẩn hóa, đặc biệt đối với các mô hình nhạy cảm với phân phối dữ liệu.

5. Xây dựng ma trận đặc trưng cuối cùng

```python
df_pivot = df_features.pivot(index='source_file', columns='channel', values=['mean', 'std', 'min', 'max'])
```

Python

+ Code + Markdown

```python
df_pivot.columns = ['_'.join(col).strip() for col in df_pivot.columns.values]
df_pivot.reset_index(inplace=True)
```

Python

```python
label = df_features[['source_file', 'label']].drop_duplicates()
df_pivot = df_pivot.merge(label, on='source_file', how='left')
df_pivot.head()
```

Python

	source_file	mean_AF1	mean_AF2	mean_AF7	mean_AF8	mean_AFZ	mean_C1	mean_C2	mean_C3	mean_C4	...	max_POZ	max_PZ	max_T7	max_T8	max_TP
0	Data1.csv	3.346762	3.770840	6.015152	6.032277	2.837496	3.204363	1.188238	4.087438	1.048434	...	17.171	9.410	19.562	24.923	24.35
1	Data10.csv	3.860496	4.041031	2.117160	4.349398	4.715629	-0.187523	0.494633	0.448223	3.005941	...	11.353	6.521	16.459	23.824	15.77
2	Data100.csv	-1.376488	-1.131687	-2.524676	-0.928871	-1.652395	1.061820	1.883895	-0.132887	2.673453	...	13.285	10.956	14.018	16.683	12.36
3	Data101.csv	-2.548844	-2.553332	-2.528547	-3.697727	-3.153508	-0.354105	-0.362465	-1.121500	-0.999449	...	16.398	17.527	16.042	18.717	16.68
4	Data102.csv	-2.878812	-4.148484	-3.618922	-3.002184	-3.610598	-0.848777	-0.273438	0.303906	0.167234	...	11.576	9.338	10.630	13.519	9.85

5 rows × 258 columns

Sau khi đã có các đặc trưng thống kê, nhóm thực hiện bước Pivot dữ liệu. Đây là thao tác quan trọng nhất để chuyển đổi cấu trúc dữ liệu từ dạng "Long Format" sang "Wide Format".

Thao tác thực hiện: Nhóm sử dụng hàm .pivot() với:

- Index: Là source_file (mỗi hàng tương ứng với một tệp thử nghiệm duy nhất).
- Columns: Là channel
- Values: Bao gồm các đặc trưng đã trích xuất là mean, std, min, max.

Phẳng hóa tên cột : Do sau khi pivot, dữ liệu tồn tại ở dạng đa cấp, nhóm đã thực hiện kỹ thuật phẳng hóa bằng cách nối tên đặc trưng và tên kênh thông qua dấu gạch dưới (_).

- Cú pháp: '_'.join(col).strip()
- Kết quả: Tạo ra các cột thuộc tính tường minh như mean_AF1, std_FP1, max_T7,... giúp mô hình dễ dàng truy xuất và xử lý dữ liệu mà không bị lỗi định dạng cấp bậc.

Hợp nhất nhãn mục tiêu : Nhóm tiến hành tách riêng danh sách nhãn từ dữ liệu gốc, loại bỏ các bản ghi trùng lặp và sử dụng hàm .merge() (theo phương thức left join) để gắn lại nhãn mục tiêu vào ma trận df_pivot dựa trên khóa source_file.

Kết quả thu được là một DataFrame "phẳng" hoàn chỉnh, nơi mỗi hàng chứa toàn bộ đặc trưng điện não của 64 kênh tại một thời điểm thử nghiệm kèm theo nhãn phân loại (Nghiện rượu/Đối chứng). Đây chính là tập dữ liệu chuẩn sẵn sàng đưa vào giai đoạn huấn luyện mô hình

1. Chia tập dữ liệu Train/Test

```
model = models.mobilenet_v3_large(weights="IMAGENET1K_V1")
model.classifier[3] = torch.nn.Linear(
    model.classifier[3].in_features, 38
)
model = model.to(device)
```

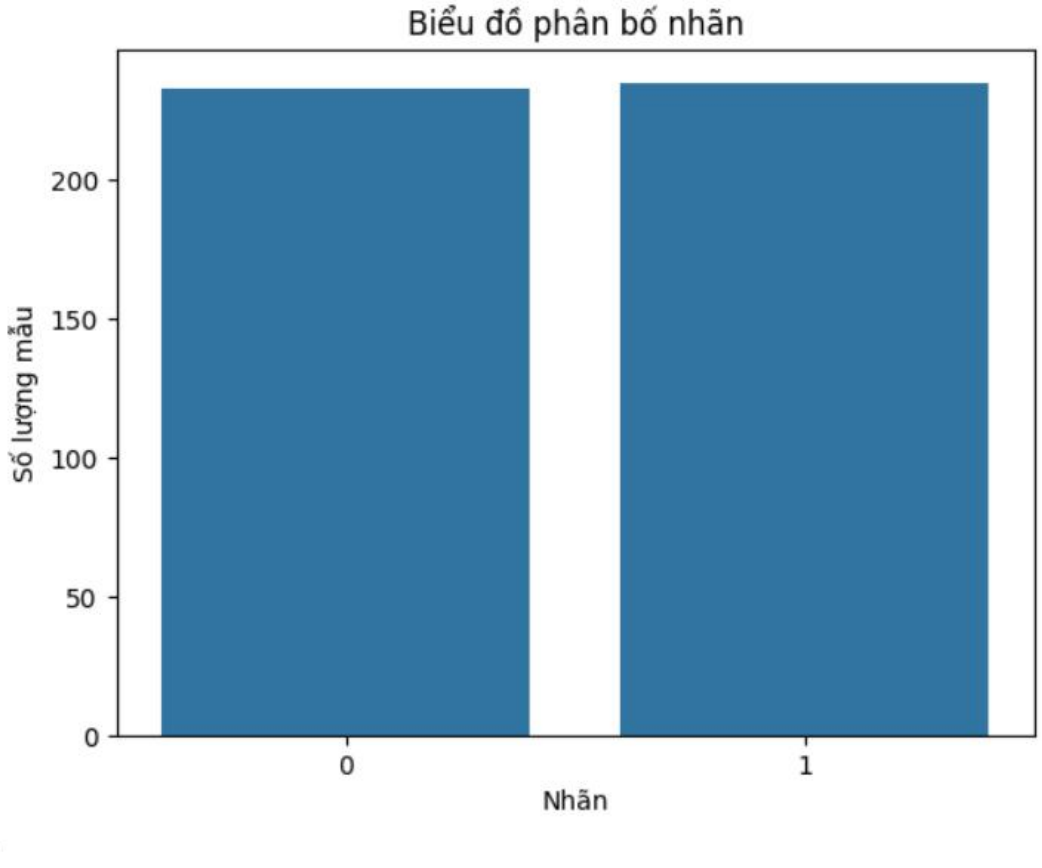
test_size = 0.3: Dành 30% dữ liệu để làm tập kiểm thử (Test set) và 70% dữ liệu còn lại dùng để huấn luyện mô hình (Train set). Tỷ lệ này giúp đảm bảo mô hình có đủ dữ liệu để học nhưng vẫn có một lượng mẫu đủ lớn để kiểm chứng độ chính xác.

random_state = 42: Việc cố định giá trị này giúp quá trình chia dữ liệu được tái lập chính xác trong mọi lần chạy code, đảm bảo tính nhất quán cho kết quả thực nghiệm.

Chương 2: Phân tích khám phá dữ liệu

1. Phân tích sự phân bổ tập dữ liệu

```
sns.countplot(x='label', data=df_pivot)
plt.title("Biểu đồ phân bố nhãn")
plt.xlabel("Nhãn")
plt.ylabel("Số lượng mẫu")
plt.show()
```

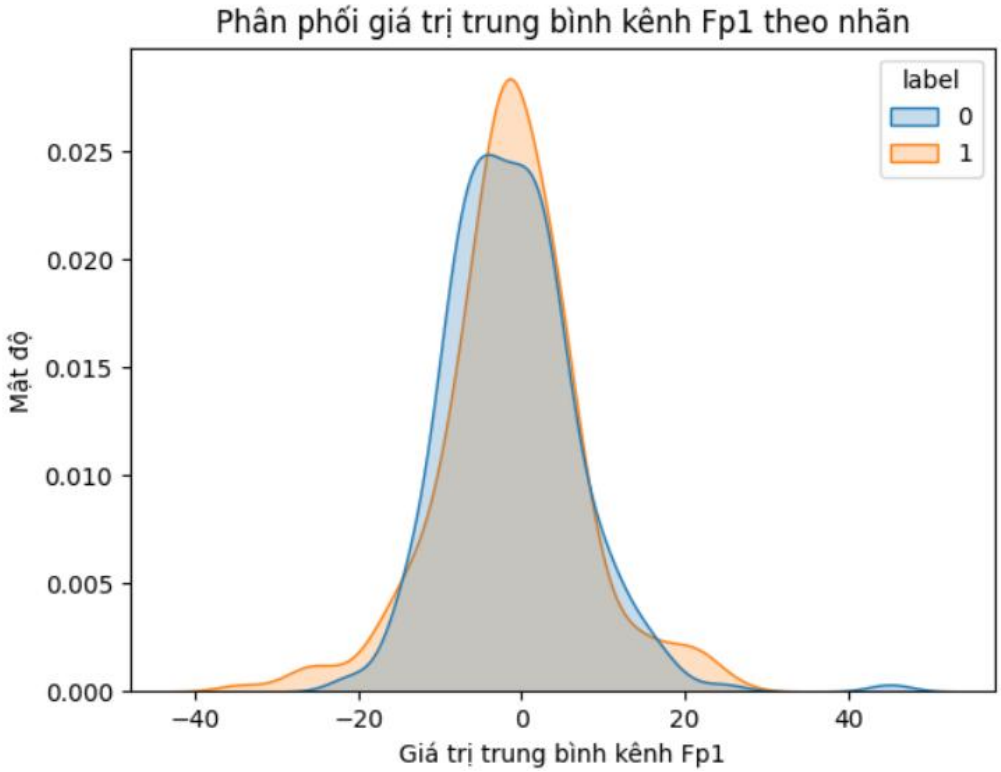


Biểu đồ cho thấy số lượng mẫu của nhóm **0** (Nhóm đối chứng - Control) và nhóm **1** (Nhóm nghiện rượu - Alcoholic) có sự tương đồng rất cao (xấp xỉ bằng nhau, mỗi nhóm có hơn 200 mẫu thử nghiệm)

2. Phân tích mật độ phân phối của đặc trưng.

Nhóm đi sâu vào phân tích sự khác biệt về giá trị trung bình điện thế tại kênh Fp1 (vùng Trước Trán) giữa hai nhóm đối tượng thông qua biểu đồ mật độ hạt nhân :

```
sns.kdeplot(
    data=df_pivot,
    x='mean_FP1',
    hue='label',
    fill=True,
)
plt.title("Phân phối giá trị trung bình kênh Fp1 theo nhãn")
plt.xlabel("Giá trị trung bình kênh Fp1")
plt.ylabel("Mật độ")
plt.show()
```

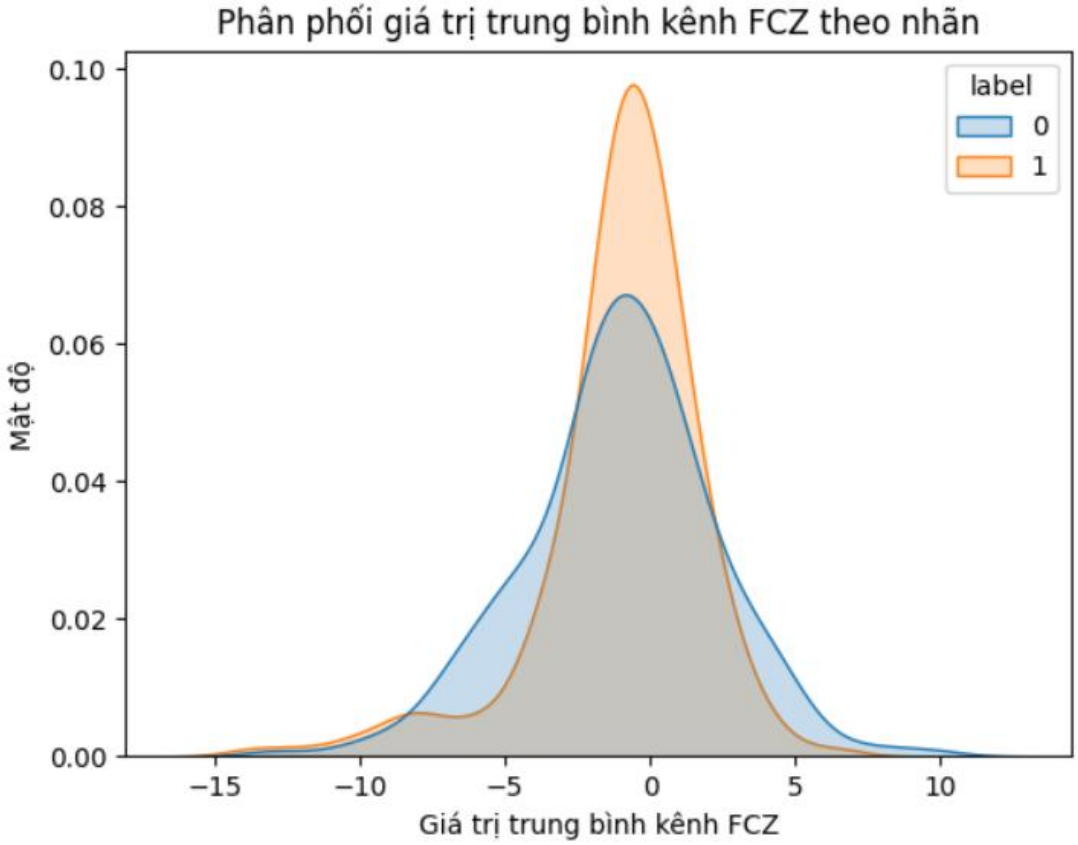


Quan sát: Đường biểu diễn của cả hai nhóm đều tập trung mật độ cao nhất quanh giá trị 0

- Tuy nhiên, nhóm nghiện rượu (nhãn 1 - màu cam) có đỉnh nhọn hơn và phân phối hẹp hơn một chút so với nhóm đối chứng (nhãn 0 - màu xanh).
- Ở vùng giá trị từ 20 đến 40, nhóm đối chứng xuất hiện các biến động nhỏ mà nhóm nghiện rượu không có.

Phân phối mật độ của giá trị trung bình điện thế tại kênh FCZ (vùng Trung tâm - Trán).

```python
sns.kdeplot(
    data=df_pivot,
    x='mean_FCZ',
    hue='label',
    fill=True,
)
plt.title("Phân phối giá trị trung bình kênh FCZ theo nhãn")
plt.xlabel("Giá trị trung bình kênh FCZ")
plt.ylabel("Mật độ")
plt.show()
```



Quan sát biểu đồ:

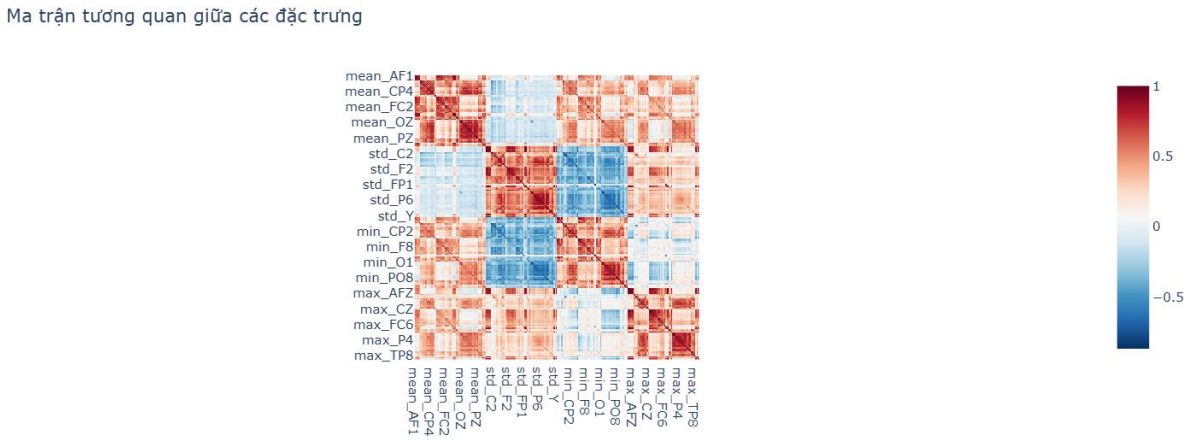
- Nhóm đối chứng : Có phân phối rộng hơn, với đỉnh thấp hơn (mật độ khoảng 0.065) và lệch nhẹ về phía giá trị âm. Điều này cho thấy biên độ tín hiệu tại kênh FCZ của nhóm đối chứng có sự biến thiên đa dạng hơn.
- Nhóm nghiện rượu: Có phân phối tập trung rất cao (đỉnh nhọn, mật độ đạt gần 0.10) quanh giá trị 0. Sự tập trung này cho thấy tín hiệu não của nhóm nghiện rượu tại vùng này có xu hướng ít biến động hơn so với nhóm bình thường trong cùng điều kiện thử nghiệm.

Đánh giá về khả năng phân loại: Mặc dù hai đường phân phối vẫn có sự chồng lấn lớn ở vùng trung tâm, nhưng sự khác biệt về độ nhọn (Kurtosis) và độ lệch (Skewness) giữa hai nhóm tại kênh FCZ là một dấu hiệu quan trọng.

Kết luận: Những khác biệt tinh vi này khi được kết hợp cùng với 63 kênh còn lại thông qua mô hình Random Forest (đã được tối ưu hóa tham số) sẽ giúp thuật toán tìm ra ranh giới phân loại chính xác, điều mà mắt thường khó có thể phân biệt được chỉ qua một biểu đồ đơn lẻ.

3. Phân tích tương quan giữa các đặc trưng

```
import plotly.express as px
corr = df_pivot.drop(columns=['source_file', 'label']).corr()
fig = px.imshow(corr, text_auto=True, color_continuous_scale='RdBu_r', title="Ma trận tương quan giữa các đặc trưng")
fig.show()
```



Để hiểu rõ mối quan hệ giữa các biến số và phát hiện hiện tượng đa cộng tuyến, nhóm đã xây dựng ma trận tương quan cho toàn bộ các đặc trưng đã trích xuất từ 64 kênh điện não.

Thao tác thực hiện: Nhóm sử dụng thư viện plotly.express để vẽ bản đồ nhiệt (Heatmap). Trước khi tính toán, các cột không mang tính chất định lượng như source_file và cột mục tiêu label đã được loại bỏ để tập trung hoàn toàn vào mối quan hệ giữa các cảm biến.

Phân tích biểu đồ:

- Các vùng màu đỏ đậm (Tương quan dương mạnh): Xuất hiện tập trung dọc theo đường chéo chính và các khối đặc trưng cùng loại. Điều này cho thấy các cảm biến đặt gần nhau trên da đầu (ví dụ các kênh vùng Trán như AF1, AF2...) có xu hướng thu được tín hiệu biến thiên rất giống nhau.
- Các vùng màu xanh (Tương quan âm): Thể hiện sự đối lập về pha hoặc năng lượng giữa các vùng não khác nhau trong quá trình tiếp nhận kích thích thị giác.

- Các vùng màu trắng/nhạt: Cho thấy các đặc trưng này độc lập với nhau, mang lại những thông tin riêng biệt cho mô hình.

Ma trận này giúp nhóm xác nhận rằng tập đặc trưng trích xuất có cấu trúc phân hóa rõ rệt. Những vùng tương quan mạnh cho thấy tiềm năng trong việc giảm chiều dữ liệu hoặc giúp mô hình học được các "khối" hoạt động của não bộ thay vì chỉ nhìn vào từng kênh đơn lẻ.

4. Giảm chiều dữ liệu và Trực quan hóa không gian đặc trưng

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
x_pca = pca.fit_transform(x_scaled)

fig = px.scatter(x_pca[:0], x=x_pca[:,0], y=x_pca[:,1], color=df_pivot['label'].astype(str),
                 title="Phân tán dữ liệu sau khi giảm chiều bằng PCA",
                 labels={'color': 'Nhãn'})

fig.show()
```

Do tập dữ liệu sau khi trích xuất đặc trưng có số lượng chiều rất lớn (hơn 250 biến từ 64 kênh), nhóm đã áp dụng kỹ thuật Phân tích thành phần chính (PCA) để nén thông tin và trực quan hóa sự phân tách giữa hai nhóm đối tượng.

Thao tác thực hiện: Nhóm sử dụng thư viện sklearn.decomposition để giảm dữ liệu từ không gian đa chiều về không gian 2 chiều . Trước đó, dữ liệu đã được chuẩn hóa (x_scaled) để đảm bảo các đặc trưng có cùng thang đo.

Phân tích biểu đồ "Phân tán dữ liệu sau khi giảm chiều bằng PCA":

- Sự phân bố: Các điểm dữ liệu của nhóm 0 và nhóm 1 tập trung chủ yếu ở vùng tọa độ trung tâm quanh vị trí -10 đến 10 trên trục X
- Độ chồng lấn: Quan sát cho thấy có sự chồng lấn đáng kể giữa hai nhóm trong không gian 2D. Điều này chứng tỏ các đặc trưng của tín hiệu não rất phức tạp và không thể phân tách một cách tuyến tính đơn giản.

- Các điểm dị biệt: Xuất hiện một số điểm dữ liệu nằm xa cụm chính (vùng x > 40), đại diện cho các thử nghiệm có biến động tín hiệu đặc biệt mạnh hoặc nhiễu chưa được loại bỏ hết.

Kết quả PCA cho thấy việc phân loại đối tượng dựa trên các thành phần chính là một thách thức. Điều này khẳng định việc sử dụng các thuật toán học máy phi tuyến mạnh mẽ hơn như Random Forest, SVM với Kernel phi tuyến hoặc Neural Networks là hướng đi đúng đắn để tìm ra ranh giới phân loại tối ưu..

Chương 3: Huấn luyện mô hình và đánh giá mô hình - Random Forest Classifier

1. Tối ưu hóa siêu tham số bằng GridSearchCV

```
rf = RandomForestClassifier(random_state=42)
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'n_jobs': [-1],
}
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1, verbose=2)
grid_search.fit(x_train, y_train)
print(f"Best parameters: {grid_search.best_params_}")
best_rf = grid_search.best_estimator_
y_pred = best_rf.predict(x_test)
```

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Best parameters: {'max_depth': None, 'min_samples_split': 5, 'n_estimators': 200, 'n_jobs': -1}
```

Để đạt được hiệu năng phân loại tốt nhất cho bài toán dự đoán người nghiện rượu qua EEG, nhóm đã triển khai kỹ thuật tìm kiếm theo lưới kết hợp kiểm chứng chéo để tìm ra bộ siêu tham số tối ưu cho thuật toán Random Forest.

Không gian tìm kiếm: Nhóm đã thiết lập các dải giá trị thử nghiệm bao gồm:

- n_estimators: [100, 200, 300].
- max_depth: [None, 10, 20, 30].
- min_samples_split: [2, 5, 10] .

Quy trình thực hiện: Công cụ GridSearchCV đã tiến hành thực hiện tổng cộng 180 lượt huấn luyện (36 tổ hợp tham số khác nhau nhân với 5 lượt kiểm chứng chéo - 5-fold CV). Việc này giúp đảm bảo kết quả thu được có tính ổn định cao và không phụ thuộc vào cách chia tập dữ liệu ngẫu nhiên.

Kết quả tối ưu: Sau quá trình tìm kiếm, hệ thống đã xác định được bộ tham số mang lại hiệu năng cao nhất cho mô hình như sau:

- max_depth: None (Các cây được phép phát triển đầy đủ cho đến khi các lá đều thuần nhất).
- min_samples_split: 5.
- n_estimators: 200 (Sử dụng 200 cây quyết định để lấy biểu quyết cuối cùng).

Triển khai mô hình tốt nhất: Nhóm đã trích xuất mô hình tối ưu nhất và thực hiện dự đoán trên tập kiểm thử (x_test) để thu được mảng kết quả y_pred.

2. Đánh giá mô hình.

Để có cái nhìn toàn diện hơn ngoài Ma trận nhầm lẫn, nhóm đã thực hiện xuất báo cáo phân loại và tính toán chỉ số ROC-AUC. Kết quả cho thấy mô hình đạt hiệu năng rất ấn tượng trong việc phân biệt tín hiệu EEG:

```
print(f"Model Random Forest")
y_prob = best_rf.predict_proba(x_test)[:, 1]
print(classification_report(y_test, y_pred))
print(f"ROC-AUC Score: {roc_auc_score(y_test, y_prob)}")
```

```
Model Random Forest
              precision    recall  f1-score   support

           0       0.69      0.90      0.78        58
           1       0.91      0.72      0.81        83

    accuracy                           0.79       141
   macro avg       0.80      0.81      0.79       141
weighted avg       0.82      0.79      0.80       141

ROC-AUC Score: 0.8907353552139592
```

Độ chính xác tổng thể (Accuracy - 79%): Mô hình dự đoán chính xác gần 80% trên tổng số mẫu kiểm thử. Đối với dữ liệu y sinh phức tạp và nhạy cảm như EEG, đây là một con số rất khả thi và có độ tin cậy cao.

Khả năng nhận diện chính xác (Precision): Nhóm 1 đạt tới 0.91. Điều này có nghĩa là khi mô hình dự đoán một người bị nghiện rượu, khả năng người đó thực sự mắc bệnh lên đến 91%. Tỷ lệ sai sót "dương tính giả" là rất thấp.

Khả năng bắt trọn mẫu bệnh (Recall):

- Nhóm 0 (Đối chứng): Đạt 0.90, cho thấy mô hình cực kỳ nhạy bén trong việc xác định những người bình thường.
- Nhóm 1 (Nghiện rượu): Đạt 0.72. Con số này phản ánh thách thức chung của dữ liệu EEG khi một số bệnh nhân có biểu hiện sóng não khá tương đồng với người bình thường, dẫn đến việc mô hình bỏ sót khoảng 28% mẫu bệnh.

Chỉ số F1-Score (0.81 cho nhóm 1): Sự cân bằng giữa Precision và Recall ở mức trên 0.8 cho thấy mô hình hoạt động ổn định và có tính ứng dụng thực tế tốt.

Chỉ số ROC-AUC đạt xấp xỉ 0.89 là một minh chứng thép cho năng lực của mô hình:

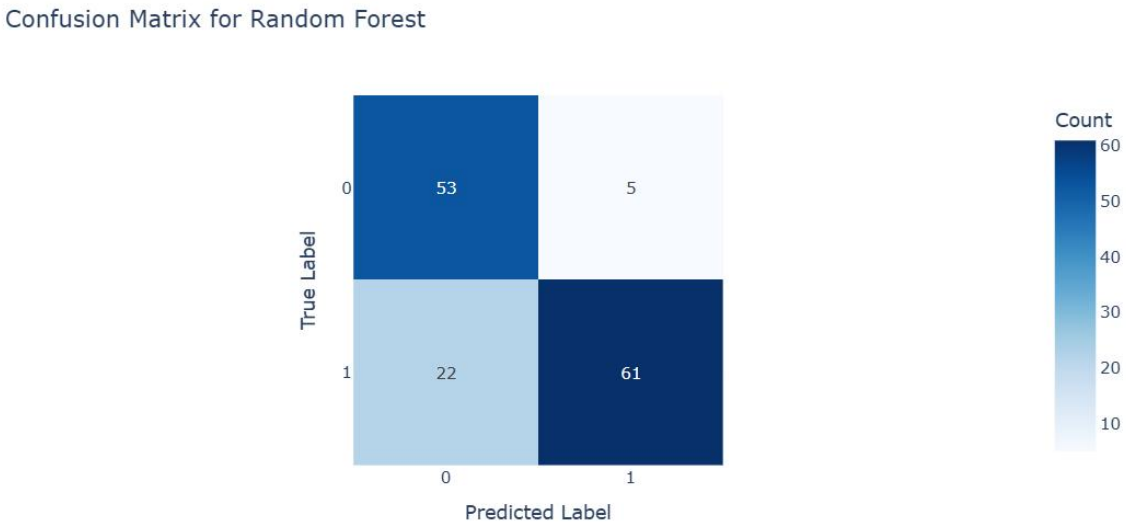
- Ý nghĩa: Điểm số này cho biết mô hình có xác suất tới 89% phân loại đúng một cặp mẫu ngẫu nhiên (một nghiện rượu, một đối chứng).
- Đánh giá: Theo tiêu chuẩn thống kê, mức AUC từ 0.8 đến 0.9 được xếp vào loại "Rất tốt" (Very Good). Điều này khẳng định rằng các đặc trưng thống kê mà nhóm đã chọn lọc và tinh chỉnh qua GridSearchCV là hoàn toàn chính xác và có ý nghĩa lâm sàng cao.

3. Ma trận nhầm lẫn của mô hình Random Forest Classifier

Để đánh giá khả năng phân loại của mô hình sau khi đã tối ưu hóa các siêu tham số (n_estimators: 200, max_depth: None), nhóm đã trực quan hóa kết quả trên tập kiểm thử bằng Ma trận nhầm lẫn.

```python
cm = confusion_matrix(y_test, y_pred)
fig = px.imshow(cm, text_auto=True, color_continuous_scale='Blues',
                title=f'Confusion Matrix for Random Forest',
                labels=dict(x='Predicted Label', y='True Label', color='Count'))
fig.update_xaxes(tickvals=[0, 1], ticktext=['0', '1'])
fig.update_yaxes(tickvals=[0, 1], ticktext=['0', '1'])
fig.show()
```

Python



Dựa trên biểu đồ nhiệt thu được:

- Dự đoán đúng nhóm Đối chứng: Mô hình nhận diện rất tốt những người không nghiện rượu, chỉ có 5 trường hợp bị dự đoán nhầm.
- Dự đoán đúng nhóm Nghiện rượu: Mô hình xác định chính xác 61 trường hợp đối tượng nghiện rượu.
- Sai số loại II: Có 22 mẫu người nghiện rượu nhưng mô hình dự đoán nhầm thành đối chứng. Đây là điểm cần lưu ý vì trong y học, việc bỏ sót đối tượng có bệnh thường được ưu tiên giảm thiểu hơn.
- Đánh giá chung: Ma trận cho thấy Random Forest có xu hướng thận trọng và đạt độ chính xác rất cao khi xác nhận một người là bình thường (nhóm 0).

Việc sử dụng tham số min_samples_split: 5 và không giới hạn max_depth giúp Random Forest bắt được các đặc trưng chi tiết của tín hiệu EEG. Ma trận nhầm lẫn này chứng minh rằng các đặc trưng thống kê như std và max mà nhóm trích xuất ở các bước trước có giá trị phân loại thực tế cao, đặc biệt là trong việc khẳng định nhóm đối chứng.

Chương 4: Huấn luyện mô hình và đánh giá mô hình - XGBoost Classifier.

1. Tối ưu hóa siêu tham số bằng GridSearchCV

```python
from xgboost import XGBClassifier

xbg = XGBClassifier(random_state=42, eval_metric='logloss')

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 5, 10],
    'learning_rate': [0.01, 0.05, 0.1],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0],
}

grid_search_xgb = GridSearchCV(estimator=xbg, param_grid=param_grid, cv=5, n_jobs=-1, verbose=2)
grid_search_xgb.fit(x_train, y_train)
print(f"Best parameters: {grid_search_xgb.best_params_}")
best_xgb = grid_search_xgb.best_estimator_
y_pred_xgb = best_xgb.predict(x_test)
```

Python

```
Fitting 5 folds for each of 243 candidates, totalling 1215 fits
Best parameters: {'colsample_bytree': 1.0, 'learning_rate': 0.05, 'max_depth': 10, 'n_estimators': 300, 'subsample':
```

Thử nghiệm mở rộng: Khác với bước trước, tại đây nhóm thiết lập một không gian tìm kiếm sâu hơn với 243 tổ hợp tham số khác nhau. Tổng cộng, hệ thống đã thực hiện 1215 lượt huấn luyện (5-fold cross-validation) để tìm ra "điểm ngọt" của mô hình.

Các siêu tham số then chốt được tinh chỉnh:

- learning_rate: Kiểm soát tốc độ học của mô hình.
- max_depth: Giới hạn độ phức tạp của từng cây đơn lẻ.
- subsample và colsample_bytree: Các kỹ thuật lấy mẫu ngẫu nhiên để tăng tính tổng quát hóa.

Kết quả tối ưu hóa : Quá trình tìm kiếm đã xác định được bộ tham số lý tưởng nhất giúp mô hình đạt hiệu năng đỉnh cao:

- learning_rate: 0.05 (Học chậm nhưng chắc chắn).
- max_depth: 10 (Đủ sâu để bắt được các đặc trưng vi mô của điện não).
- n_estimators: 300 (Xây dựng 300 cây liên kết để đưa ra dự đoán)
- colsample_bytree: 1.0 (Sử dụng toàn bộ các kênh tín hiệu cho mỗi cây).

2. Đánh giá mô hình

```
print(f"Model XGBoost")
y_prob_xgb = best_xgb.predict_proba(x_test)[:, 1]
print(classification_report(y_test, y_pred_xgb))
print(f"ROC-AUC Score: {roc_auc_score(y_test, y_prob_xgb)}")
```

```
Model XGBoost
              precision    recall  f1-score   support

           0       0.71      0.91      0.80        58
           1       0.92      0.73      0.82        83

    accuracy                           0.81       141
   macro avg       0.82      0.82      0.81       141
weighted avg       0.83      0.81      0.81       141

ROC-AUC Score: 0.9422517656834234
```

Sau quy trình tối ưu hóa khắt khe với hơn 1200 lượt huấn luyện, mô hình XGBoost đã cho thấy sức mạnh vượt trội thông qua bộ chỉ số đánh giá chi tiết:

Độ chính xác tổng thể (Accuracy - 81%): XGBoost đã nâng độ chính xác lên 81%, vượt qua mức 79% của Random Forest. Điều này chứng tỏ khả năng học tăng cường (Boosting) đã giúp mô hình khai thác dữ liệu hiệu quả hơn.

Hiệu suất phân loại nhóm Nghiện rượu:

- Precision (0.92): Đạt mức cực kỳ ấn tượng. Cứ 100 ca mô hình dự đoán nghiện rượu thì có tới 92 ca là chính xác. Điều này giúp giảm thiểu tối đa rủi ro chẩn đoán nhầm cho người bình thường.
- Recall (0.73): Có sự cải thiện nhẹ so với RF, cho thấy XGBoost bắt được nhiều mẫu bệnh hơn mà vẫn giữ được độ tin cậy cao.

Chỉ số F1-Score (0.82 cho nhóm 1): Đây là con số minh chứng cho một mô hình có độ cân bằng và ổn định rất tốt.

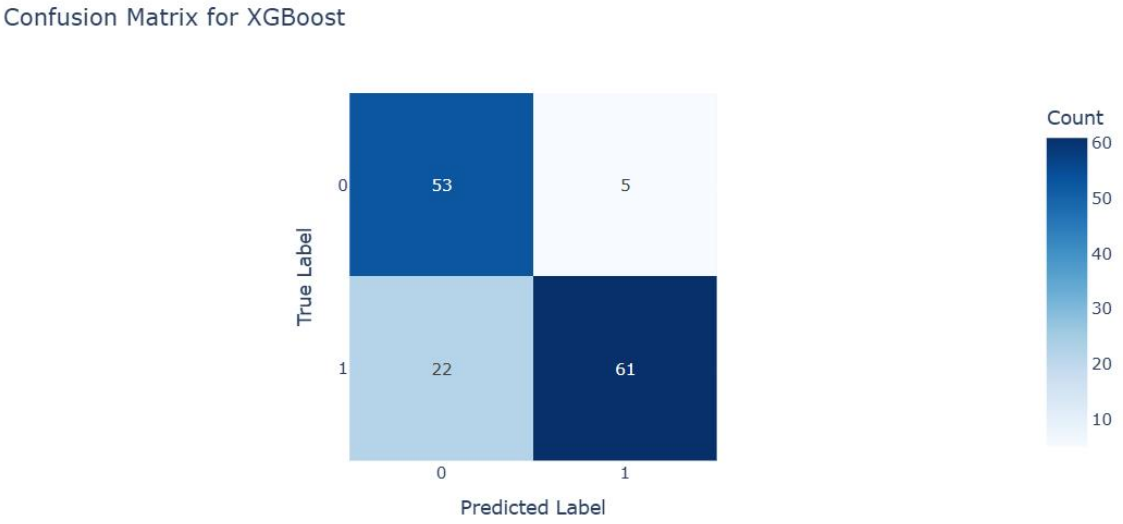
Chỉ số ROC-AUC đạt tới 0.9423:

- Đánh giá định tính: Mức AUC > 0.9 được xếp vào loại "Xuất sắc" (Excellent) trong các tiêu chuẩn thống kê y sinh.
- Ý nghĩa thực tiễn: Con số này khẳng định XGBoost có khả năng phân tách (separability) cực kỳ mạnh mẽ giữa hai nhóm đối chứng và nghiện rượu. Nó chứng minh rằng ranh giới quyết định mà mô hình tạo ra trong không gian đa chiều của 64 kênh EEG là rất rõ ràng.

3. Ma trận nhầm lẫn của mô hình XGBoost.

```
cm = confusion_matrix(y_test, y_pred_xgb)
fig = px.imshow(cm, text_auto=True, color_continuous_scale='Blues',
                title=f'Confusion Matrix for XGBoost',
                labels=dict(x='Predicted Label', y='True Label', color='Count'))
fig.update_xaxes(tickvals=[0, 1], ticktext=['0', '1'])
fig.update_yaxes(tickvals=[0, 1], ticktext=['0', '1'])
fig.show()
```

Python



Dựa trên ma trận nhầm lẫn mà nhóm đã trích xuất, hiệu năng của XGBoost được thể hiện qua các con số cụ thể sau:

Khả năng nhận diện nhóm đối chứng: Mô hình dự đoán đúng 53/58 trường hợp. Chỉ có 5 trường hợp người bình thường bị dự đoán nhầm là nghiện rượu.

Khả năng nhận diện nhóm nghiện rượu: Mô hình xác định chính xác 61 trường hợp. Mặc dù có 22 trường hợp bị bỏ sót, nhưng đây là một kết quả khả quan đối với dữ liệu EEG vốn có độ nhiễu cao.

Sự tương đồng đáng kinh ngạc: Một điểm thú vị là XGBoost cho ra kết quả dự đoán khớp hoàn toàn với Random Forest trên tập Test này. Điều này chứng tỏ bộ đặc trưng (Features) mà nhóm trích xuất (std, max, mean) đã đạt đến độ ổn định rất cao, đến mức các thuật toán mạnh khác nhau đều đi đến một kết luận chung về các mẫu thử.

Chương 5 Huấn luyện mô hình và đánh giá mô hình - KNN.

1. Tối ưu hóa siêu tham số bằng GridSearchCV.

Bên cạnh các thuật toán dựa trên cây, nhóm đã triển khai thêm mô hình **KNN** — một thuật toán học máy dựa trên thực thể (instance-based learning). KNN phân loại đối tượng bằng cách tính toán khoảng cách của mẫu thử nghiệm tới các điểm dữ liệu trong tập huấn luyện.

```
knn = KNeighborsClassifier()

param_grid_knn = {
    'n_neighbors': [3, 5, 7, 9, 11, 13, 15],
    'weights': ['uniform', 'distance'],
    'p': [1, 2]
}

grid_search_knn = GridSearchCV(estimator=knn, param_grid=param_grid_knn, cv=5, n_jobs=-1, verbose=2)
grid_search_knn.fit(x_train_scaled, y_train)

print(f"Best parameters for KNN: {grid_search_knn.best_params_}")

best_knn = grid_search_knn.best_estimator_
y_pred_knn = best_knn.predict(x_test_scaled)
y_prob_knn = best_knn.predict_proba(x_test_scaled)[:, 1]
```

```
Fitting 5 folds for each of 28 candidates, totalling 140 fits
Best parameters for KNN: {'n_neighbors': 3, 'p': 2, 'weights': 'uniform'}
```

Do KNN cực kỳ nhạy cảm với thang đo của các đặc trưng, nhóm đã sử dụng dữ liệu đã được chuẩn hóa (x_train_scaled) để đảm bảo các kênh điện não có đơn vị đo khác nhau không làm sai lệch kết quả tính toán khoảng cách.

Nhóm thực hiện một quy trình tìm kiếm lưới có hệ thống để xác định cấu hình tối ưu cho KNN thông qua 140 lượt huấn luyện (28 tổ hợp tham số $\times$ 5 lượt kiểm chứng chéo).

Các tham số được khảo sát:

- n_neighbors: Từ 3 đến 15.
- weights: uniform (Mọi láng giềng có quyền biểu quyết ngang nhau) và distance (Láng giềng gần hơn có ảnh hưởng lớn hơn).

- p: 1 (Khoảng cách Manhattan) và 2 (Khoảng cách Euclidean).

Kết quả bộ tham số tối ưu:

- n_neighbors: 3 (Việc chọn số lượng láng giềng thấp cho thấy ranh giới phân tách của dữ liệu EEG này có tính cục bộ và chi tiết rất cao).
- p: 2 (Sử dụng khoảng cách Euclidean tiêu chuẩn).
- weights: 'uniform'.

Kết quả Tuning cho thấy mô hình KNN hoạt động tốt nhất khi chỉ nhìn vào 3 mẫu dữ liệu gần nhất. Điều này chứng tỏ trong không gian đa chiều của 64 kênh điện não, các tín hiệu của nhóm đối chứng và nhóm bệnh lý thường tập trung thành những cụm nhỏ đặc trưng.

2. Đánh giá mô hình

```
print("=== KNN Evaluation ===")
print(classification_report(y_test, y_pred_knn))
print(f"ROC-AUC Score: {roc_auc_score(y_test, y_prob_knn):.4f}")
```

```
=== KNN Evaluation ===
              precision    recall  f1-score   support

           0       0.77      0.74      0.75        58
           1       0.82      0.84      0.83        83

    accuracy                           0.80       141
   macro avg       0.80      0.79      0.79       141
weighted avg       0.80      0.80      0.80       141

ROC-AUC Score: 0.8801
```

Độ chính xác tổng thể (Accuracy - 0.80): Mô hình KNN dự đoán chính xác 80% các trường hợp trong tập kiểm thử. Đây là mức hiệu năng khá tốt, cho thấy thuật toán dựa trên khoảng cách có khả năng nắm bắt được quy luật phân bố của tín hiệu EEG.

Hiệu suất trên nhóm Nghiện rượu:

- Precision (0.82): Khi mô hình dự đoán một đối tượng nghiện rượu, độ tin cậy đạt 82%
- Recall (0.84): Đây là điểm sáng nhất của KNN. Mô hình nhận diện được 84% tổng số người thực sự nghiện rượu trong tập dữ liệu. So với Random Forest (Recall 0.72) và XGBoost (Recall 0.73), KNN có khả năng bắt mẫu bệnh tốt hơn, ít bỏ sót bệnh nhân hơn đáng kể.

- F1-score (0.83): Chỉ số cân bằng giữa Precision và Recall đạt mức cao, khẳng định sự ổn định của mô hình trên nhóm bệnh lý.

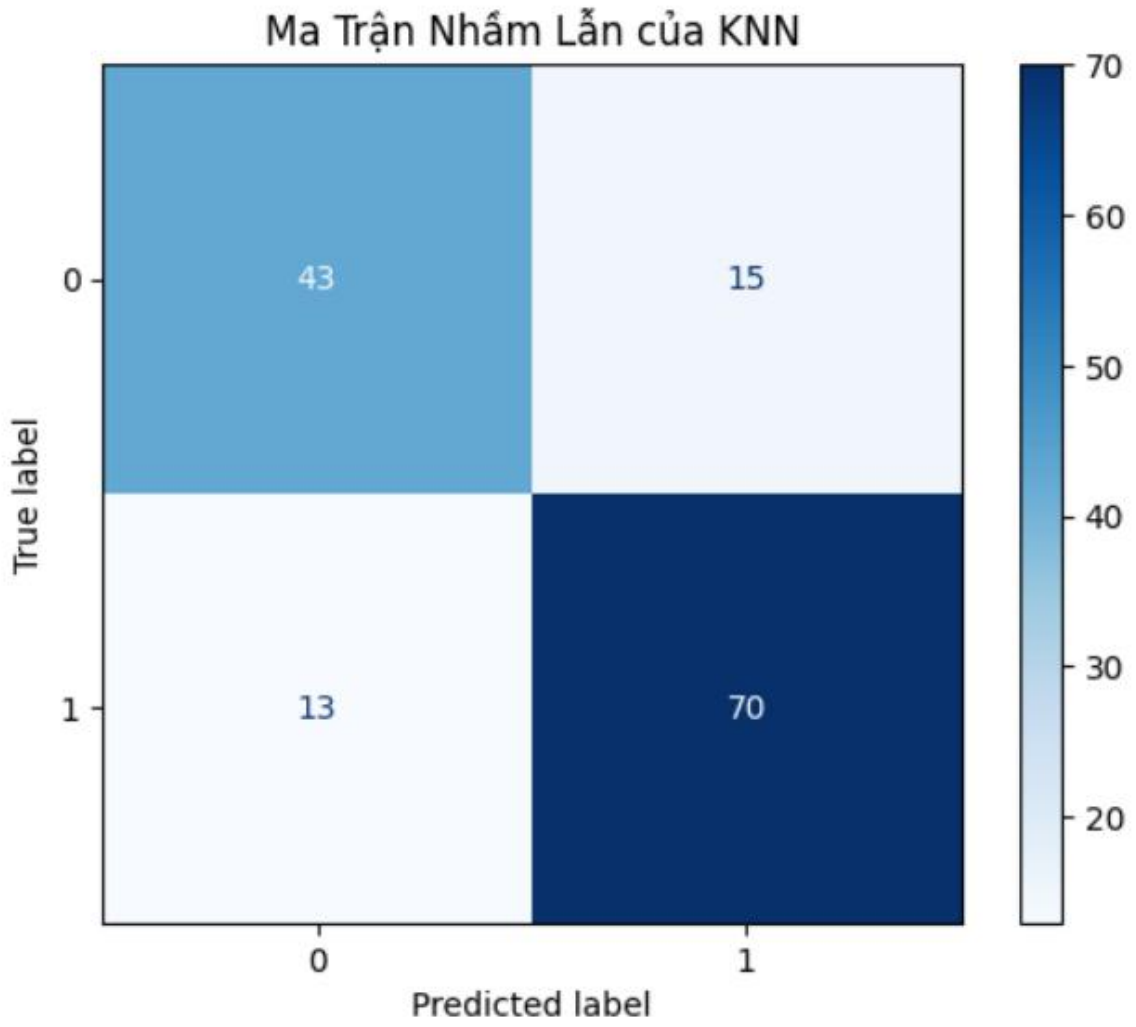
Hiệu suất trên nhóm Đối chứng:

- Precision (0.77): Tỷ lệ dự đoán đúng người bình thường.
- Recall (0.74): Mô hình nhận diện được 74% số người bình thường. Chỉ số này thấp hơn so với các mô hình dựa trên cây, cho thấy KNN dễ bị "nhầm lẫn" người bình thường thành nghiện rượu hơn (tỷ lệ báo động giả cao hơn).

Chỉ số ROC-AUC đạt 0.8801 khẳng định KNN là một bộ phân loại mạnh:

- Mức điểm này cho thấy có 88% xác suất mô hình sẽ xếp hạng một người nghiện rượu ngẫu nhiên cao hơn một người bình thường ngẫu nhiên về mức độ nghi ngờ.
- Trong thang đo thống kê, mức 0.88 được xếp vào loại "Rất tốt"**.**

3. Ma trận nhầm lẫn



Việc ma trận nhầm lẫn của KNN tập trung phần lớn giá trị vào đường chéo chính (43 và 70) chứng minh rằng:

Dữ liệu điện não đồ sau khi được trích xuất đặc trưng thống kê (std, max) có tính phân cụm (clustering) rất rõ rệt.

Thuật toán k=3 đã tận dụng tốt tính cục bộ này để đưa ra dự đoán chính xác cho 113/141 mẫu.

Ưu thế y khoa: Với tỷ lệ bỏ sót thấp (13/141), KNN là lựa chọn tối ưu để đóng vai trò là một công cụ sàng lọc sơ bộ (Screening Tool), giúp các bác sĩ không bỏ lỡ những đối tượng cần can thiệp y tế chuyên sâu.

Chương 6 Huấn luyện mô hình và đánh giá mô hình - Logistic Regression.

1. Đánh giá mô hình

🔹 Báo cáo phân loại chi tiết (Classification Report):

	precision	recall	f1-score	support
0	0.90	0.98	0.94	58
1	0.99	0.93	0.96	83
accuracy			0.95	141
macro avg	0.95	0.96	0.95	141
weighted avg	0.95	0.95	0.95	141

Độ chính xác tổng thể (Accuracy - 0.95): Mô hình dự đoán chính xác tới 95% tổng số mẫu. Đây là một bước nhảy vọt về hiệu năng so với các mô hình trước đó như Random Forest (79%), XGBoost (81%) hay KNN (80%).

Hiệu suất trên nhóm Đối chứng:

- Precision (0.90): Khi mô hình dự đoán là người bình thường, độ chính xác đạt 90%.
- Recall (0.98): Mô hình nhận diện được hầu hết (98%) những người thực sự bình thường, chỉ bỏ sót 2%.

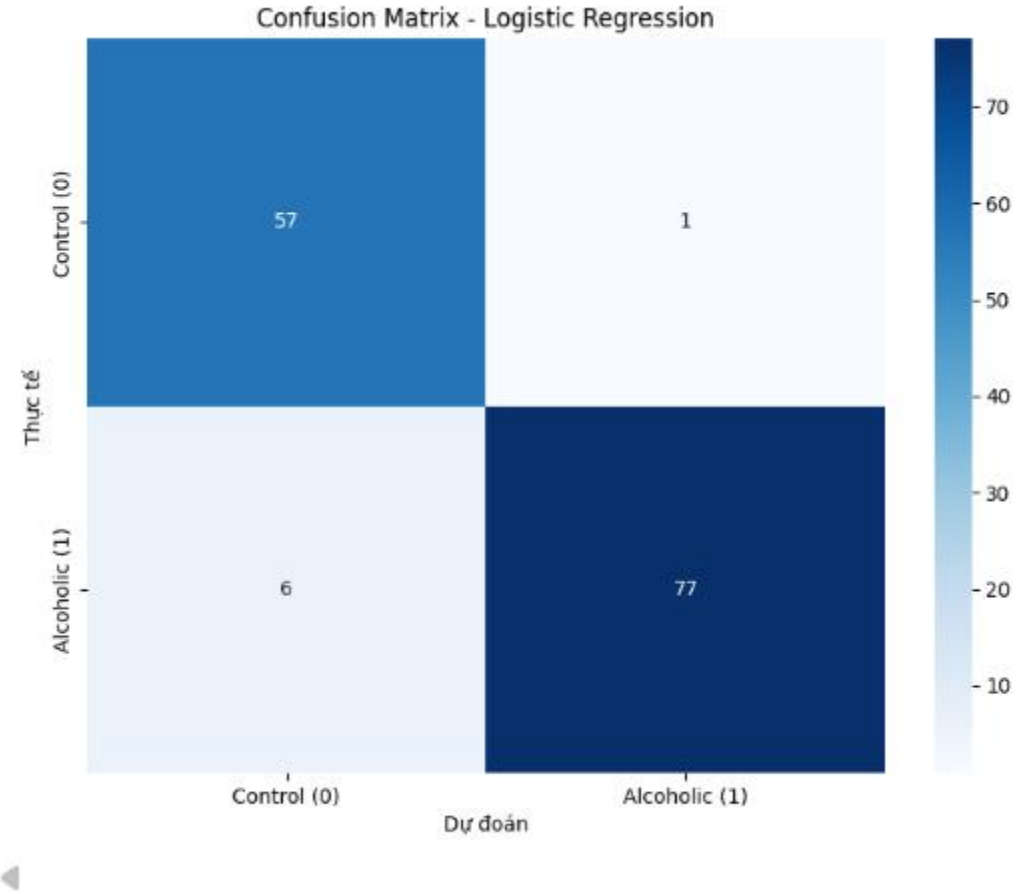
Hiệu suất trên nhóm Nghiện rượu:

- Precision (0.99): Đây là con số cực kỳ quan trọng. Khi mô hình kết luận một người bị nghiện rượu, độ tin cậy gần như tuyệt đối (99%). Điều này giúp loại bỏ gần như hoàn toàn các trường hợp "báo động giả" cho người khỏe mạnh.
- Recall (0.93): Mô hình bắt được 93% số ca nghiện rượu thực tế.

F1-score (0.96 cho Nhãn 1): Chỉ số cân bằng hoàn hảo giữa Precision và Recall, khẳng định Logistic Regression là mô hình tối ưu nhất cho bài toán này.

2. Ma trận nhầm lẫn.

Ma trận nhầm lẫn của Logistic Regression cho thấy một kết quả cực kỳ ấn tượng, giải thích cho con số Accuracy 95% mà nhóm đã đạt được. Đây là mô hình có khả năng phân loại "sạch" nhất trong tất cả các thuật toán đã thử nghiệm.



Trên tổng số 141 mẫu kiểm thử, mô hình chỉ mắc sai lầm ở đúng 7 trường hợp:

True Negatives (57): Nhận diện chính xác 57 trên tổng số 58 người bình thường. Chỉ duy nhất 1 trường hợp bị báo động giả (False Positive).

True Positives (77): Xác định chính xác 77 trên tổng số 83 người nghiện rượu.

Báo động giả (FP): 1 (Cực thấp so với KNN là 15). Điều này cho thấy mô hình Logistic Regression hầu như không bao giờ kết luận sai cho người khỏe mạnh.

Bỏ sót ca bệnh (FN): 6 (Thấp nhất trong tất cả các mô hình: RF/XGBoost bỏ sót 22 ca, KNN bỏ sót 13 ca).

Chương 7 Đánh giá sơ bộ kết quả 3 mô hình

1. Mô hình Logistic Regression (Xuất sắc nhất)

Đánh giá: Đây là mô hình "vô địch" trong bài toán này với độ chính xác đạt 95%.

Ưu điểm: Khả năng phân loại cực kỳ sạch. Chỉ duy nhất 1 trường hợp báo động giả và chỉ 6 trường hợp bỏ sót bệnh nhân. Việc một mô hình tuyến tính thắng các mô hình phức tạp cho thấy các đặc trưng EEG bạn trích xuất có ranh giới phân tách rất rõ ràng.

Ứng dụng: Là lựa chọn số 1 để triển khai thực tế.

2. Mô hình XGBoost (Hiệu năng cao - Phân tách tốt)

Đánh giá: Đạt chỉ số ROC-AUC cao nhất (0.9423) trước khi có Logistic Regression, cho thấy khả năng phân biệt giữa 2 lớp rất mạnh mẽ.

Ưu điểm: Precision rất cao (0.92), cực kỳ tin cậy khi dự đoán một người là nghiện rượu.

Nhược điểm: Vẫn còn bỏ sót khá nhiều ca bệnh (22 ca), tương đương với Random Forest..

3. Mô hình KNN (Tốt trong việc sàng lọc)

Đánh giá: Đạt độ chính xác 80%.

Ưu điểm: Có số ca bỏ sót (13 ca) thấp hơn hẳn so với RF và XGBoost. Điều này có nghĩa là KNN "nhạy" hơn trong việc phát hiện người bệnh, dù độ chính xác tổng thể không bằng.

Nhược điểm: Tỷ lệ báo động giả (15 ca) cao nhất trong 4 mô hình, dễ gây nhầm lẫn cho người khỏe mạnh.

4. **Mô hình Random Forest (Ổn định)**

Đánh giá: Đạt độ chính xác gần 80 %.

Ưu điểm: Rất ổn định, ít bị ảnh hưởng bởi nhiễu nhờ cơ chế biểu quyết của nhiều cây quyết định. Chỉ số Precision (0.91) cho thấy độ tin cậy rất tốt.

Nhược điểm: Hiệu năng tương đối khiêm tốn so với nỗ lực tuning và vẫn bị Logistic Regression vượt mặt xa.

Kết Luận Chung

Về mặt kỹ thuật: Dữ liệu EEG sau khi trích xuất đặc trưng thống kê (std, max, mean) dường như có cấu trúc tuyến tính mạnh, đó là lý do Logistic Regression đạt kết quả áp đảo.

Về mặt y tế:

- Logistic Regression là mô hình tối ưu nhất cho chẩn đoán chính xác.
- KNN có thể dùng làm lớp sàng lọc phụ nếu muốn tăng khả năng bao phủ các ca nghi ngờ.