



Trường Đại học Cần Thơ
Trường Công nghệ thông tin và Truyền thông

BÁO CÁO MÔN HỌC: CTH621

Phân tích dữ liệu

Thực hiện bởi:

Võ Thanh Luân	Nhóm 6	M2524011	TRƯỞNG NHÓM
Ngô Quốc Thanh		M2524023	THÀNH VIÊN
Trần Minh Trí		M2524025	THÀNH VIÊN

GVHD: PGS.TS. Nguyễn Thanh Hải

Học kỳ 2, năm học 2024-2025
Ngày 13 tháng 7 năm 2025

MỤC LỤC

Mục lục	ii
Danh sách Hình	iv
Danh sách Bảng	vi
1 Cơ sở lý thuyết	1
1.1 Giới thiệu các thông số thống kê	1
1.1.1 Mean (Trung bình)	1
1.1.2 Median (Trung vị)	1
1.1.3 Mode (Giá trị xuất hiện nhiều nhất)	3
1.1.4 Percentile (Phân vị)	3
1.1.5 Max and Min (Giá trị lớn nhất và nhỏ nhất)	5
1.1.6 Range (Khoảng biến thiên)	6
1.1.7 Variance (Phương sai)	7
1.1.8 Standard Deviation (Độ lệch chuẩn)	8
1.1.9 Coefficient of Variation - CV (Hệ số biến thiên)	10
1.1.10 Interquartile Range – IQR (Khoảng tứ phân vị)	11
1.2 Giới thiệu các biểu đồ trực quan	13
1.2.1 Histogram (Biểu đồ tần suất)	13
1.2.2 Box Plot (Biểu đồ hộp)	14
1.2.3 Scatter Plot (Biểu đồ phân tán)	15
1.2.4 Heatmap (Biểu đồ nhiệt)	16
1.2.5 Pie Plot (Biểu đồ tròn)	17
1.2.6 Bar Plot (Biểu đồ cột)	18
1.2.7 Line Plot (Biểu đồ đường)	19
1.3 Giới thiệu các mô hình máy học	20
1.3.1 Random Forest (Rừng ngẫu nhiên)	20
1.3.2 Gradient Boosting (Tăng cường độ dốc)	23
1.3.3 XGBoost (Extreme Gradient Boosting)	27
1.3.4 Logistic Classification (Hồi quy Logistic)	31
1.3.5 K-Nearest Neighbors - KNN (Làng giềng gần nhất)	33
1.3.6 K-Means (Thuật toán phân cụm K trung bình)	35
1.3.7 Spectral Clustering (Phân cụm phô)	36
1.3.8 Gaussian Mixture Model - GMM (Mô hình hỗn hợp Gauss)	37
1.3.9 LSTM (Long Short-Term Memory)	39
1.4 Giới thiệu các bộ chuẩn hóa dữ liệu	43
1.4.1 Chuẩn hóa dữ liệu với Min-Max Scaler	43
1.4.2 Chuẩn hóa dữ liệu với Standard Scaler	44
1.4.3 Chuẩn hóa dữ liệu với Robust Scaler	45
1.4.4 Chuẩn hóa dữ liệu với MaxAbsScaler	46
1.4.5 Chuẩn hóa dữ liệu với Quantile Transformer	47
1.5 Giới thiệu các chỉ số đánh giá hiệu suất mô hình	48
1.5.1 Accuracy (Độ chính xác)	48
1.5.2 F1-score	49

1.5.3	Precision	49
1.5.4	Recall	50
1.5.5	ROC AUC	50
1.5.6	Mean Squared Error (MAE)	51
1.5.7	Mean Absolute Error (MAE)	51
1.5.8	Root Mean Squared Error (RMSE)	52
1.5.9	Mean Absolute Percentage Error (MAPE)	53
1.5.10	Adjusted Rand Index (ARI)	54
2	Phân tích dữ liệu	56
2.1	Dữ liệu dạng bảng	56
2.1.1	Predict Student's Dropout and Academic Success	56
2.1.2	MAL show Dataset	80
2.1.3	Extrovert vs. Introvert Behavior Data	96
2.2	Dữ liệu dạng chuỗi thời gian	109
2.2.1	Daily Climate Delhi	109
2.2.2	Dữ liệu lượt khám chữa bệnh tỉnh Sóc Trăng	118
2.2.3	Dữ liệu giá chứng khoán Hòa Phát Group	127
2.3	Dữ liệu dạng ảnh	137
2.3.1	SIIM-FISABIO-RSNA COVID-19	137
2.3.2	Outdoor-fire dataset	144
Tài liệu tham khảo		156

DANH SÁCH HÌNH VẼ

1.1	Minh họa: Bagging	21
1.2	Minh họa: Random Forest [22]	21
1.3	Tương quan Out-of-bag error và test error	22
1.4	Minh họa: Boosting	23
1.5	Minh họa thuật toán AdaBoost, quá trình kết hợp các mô hình yếu [30]	25
1.6	Minh họa: Gradient boosting [20]	26
1.7	Độ lỗi qua các iteration - GB [15]	27
1.8	Minh họa cấu trúc của một mạng neural. [6]	39
1.9	Mô hình mạng neural hồi quy. [16]	41
1.10	Cấu trúc của LSTM. [32]	42
2.1	Số lượng sinh viên theo tình trạng	68
2.2	Tuổi khi nhập học theo tình trạng	68
2.3	Phân phối số môn tích lũy hk 1 nhóm Dropout	69
2.4	Phân phối số môn tích lũy hk 1 nhóm Graduate	70
2.5	Phân phối số môn tích lũy hk 2	71
2.6	Phân bố điểm trung bình sinh viên	71
2.7	Phân bố điểm trung bình sinh viên (to phỏng đoán 9-20)	72
2.8	Liên hệ giữa tình trạng nợ học phí và tình trạng sinh viên	73
2.9	Histogram 'Curricular units 1st sem (grade)' gốc	74
2.10	Histogram 'Curricular units 1st sem (grade)' qua Min-Max Scaler	75
2.11	Histogram 'Curricular units 1st sem (grade)' qua Standard Scaler	75
2.12	Histogram 'Curricular units 1st sem (grade)' qua Robust Scaler	76
2.13	Histogram 'Curricular units 1st sem (grade)' qua Max ABS Scaler	76
2.14	Histogram 'Curricular units 1st sem (grade)' qua Quantile Scaler	77
2.15	Phân phối điểm đánh giá trung bình	85
2.16	Số lượng người xem theo điểm	86
2.17	Số lượt yêu thích theo điểm	86
2.18	Số lượt yêu thích trung bình các chủ đề chính	87
2.19	Số lượt yêu thích trung bình các thẻ loại/đối tượng khán giả	88
2.20	Phân phối nhãn điểm đánh giá	90
2.21	Phân phối nhãn các cột qualitative	99
2.22	Tương quan 'Stage_fear' và 'Personality'	100
2.23	Tương quan 'Drained_after_socializing' và 'Personality'	100
2.24	Phân phối trung bình số giờ ở một mình theo tính cách	101
2.25	Phân phối trung bình số giờ ở một mình theo tính cách	101
2.26	Tương quan 'Social_event_attendance' và 'Friends_circle_size' theo nhãn 'Personality'	102
2.27	Nhiệt độ trung bình từng ngày	112
2.28	Nhiệt độ trung bình	112
2.29	Phân phối nhiệt độ trung bình	113
2.30	Nhiệt độ trung bình các tháng	113
2.31	Tương quan nhiệt độ - độ ẩm	114
2.32	Kết quả trên tập test Linear Regression	116
2.33	Kết quả trên tập test RF Regressor	116

2.34 Kết quả trên tập test XGB Regressor	117
2.35 Số lượt khám ngày	120
2.36 Số lượt khám trung bình tuần	120
2.37 Số lượt khám trung bình tuần 2025	121
2.38 Số lượt khám trung bình tuần 2024	121
2.39 Số lượt khám trung bình tuần 2023	121
2.40 Số lượt khám	122
2.41 Phân phối số lượt khám	122
2.42 Số lượt khám qua các tháng	123
2.43 Kết quả trên tập test Linear Regression	125
2.44 Kết quả trên tập test RF Regressor	125
2.45 Kết quả trên tập test XGB Regressor	126
2.46 Giá đóng cửa theo thời gian	129
2.47 Giá hình nền theo thời gian - 365 ngày gần nhất	129
2.48 Khối lượng khớp lệnh theo thời gian - 365 ngày gần nhất	130
2.49 Giá đóng cửa	130
2.50 Phân phối giá đóng cửa	131
2.51 Giá đóng cửa 2 năm gần nhất cùng các đường MA	131
2.52 Giá đóng cửa 2 năm gần nhất cùng dải Bollinger bands	132
2.53 RSI giá đóng cửa 2 năm gần nhất	132
2.54 MACD giá đóng cửa 2 năm gần nhất	132
2.55 Kết quả trên tập test Linear Regression	134
2.56 Kết quả trên tập test RF Regressor	135
2.57 Kết quả trên tập test XGB Regressor	136
2.58 Hình ảnh X-quang có phát hiện vùng lạ	137
2.59 Hình ảnh X-quang không phát hiện vùng lạ	138
2.60 Số lượng hình ảnh mỗi nhãn	138
2.61 Phân phối cường độ pixel trung bình	139
2.62 Độ tương phản theo nhãn	139
2.63 Tương quan cường độ và tương phản pixel với loại nhãn	140
2.64 K-Means. ARI = 0.0039	141
2.65 Spectral Clustering. ARI = 0.0046	142
2.66 Gaussian Mixture. ARI = -0.0095	143
2.67 Hình ảnh không lửa	144
2.68 Hình ảnh có đám cháy	145
2.69 Số lượng hình ảnh mỗi nhãn	145
2.70 Phân phối độ phân giải ảnh	146
2.71 Phân phối cường độ trung bình ảnh	147
2.72 Phân phối cường độ trung bình các màu trong ảnh có đám cháy	148
2.73 Phân phối cường độ trung bình các màu trong ảnh không có cháy	148
2.74 Độ tương phản kênh xanh dương theo nhãn	149
2.75 Độ tương phản kênh xanh lá theo nhãn	149
2.76 Độ tương phản kênh đỏ theo nhãn	150
2.77 K-Means. ARI = 0.2899	151
2.78 Spectral Clustering. ARI = 0.0838	152
2.79 Gaussian Mixture. ARI = 0.4966	153

DANH SÁCH BẢNG

1	Danh mục các ký hiệu viết tắt	ix
2.1	Một phần bảng dữ liệu student dropout	57
2.2	Thống kê dữ liệu một số đặc trưng dữ liệu sinh viên	67
2.3	Biểu diễn thống kê cột 'Curricular units 1st sem (grade)' qua các scaler	74
2.4	Kết quả Logistic Classification - Phân lớp tình trạng sinh viên	79
2.5	Kết quả Random Forest Classifier - Phân lớp tình trạng sinh viên	79
2.6	Kết quả XGBoost Classifier - Phân lớp tình trạng sinh viên	79
2.7	So sánh kết quả các mô hình - Dự đoán tình trạng sinh viên	80
2.8	Một phần bảng dữ liệu MAL dataset	81
2.9	Thống kê dữ liệu một số đặc trưng dữ liệu MAL dataset	84
2.10	Kết quả Logistic Classification - Phân lớp đánh giá	91
2.11	Kết quả Random Forest Classifier - Phân lớp đánh giá	91
2.12	Kết quả XGBoost Classifier - Phân lớp đánh giá	91
2.13	So sánh kết quả các mô hình (và nghiên cứu liên quan) - Phân lớp đánh giá	92
2.14	Kết quả Logistic Classification - Phân lớp độ phổ biến	93
2.15	Kết quả Random Forest Classifier - Phân lớp phổ biến	93
2.16	Kết quả XGBoost Classifier - Phân lớp độ phổ biến	94
2.17	So sánh kết quả các mô hình - Phân lớp độ phổ biến	94
2.18	Kết quả Logistic Classification - Phân lớp độ yêu thích	95
2.19	Kết quả Random Forest Classifier - Phân lớp yêu thích	95
2.20	Kết quả XGBoost Classifier - Phân lớp độ yêu thích	95
2.21	So sánh kết quả các mô hình - Phân lớp độ yêu thích	96
2.22	Một phần bảng dữ liệu Behavior dataset	97
2.23	Thống kê dữ liệu một số đặc trưng dữ liệu Behavior dataset	98
2.24	Kết quả Logistic Classification - Phân lớp tính cách	104
2.25	Kết quả Random Forest Classifier - Phân lớp tính cách	104
2.26	Kết quả XGBoost Classifier - Phân lớp tích cách	104
2.27	So sánh kết quả các mô hình (và nghiên cứu liên quan) - Phân lớp tính cách	105
2.28	Kết quả Logistic Classification - Phân lớp 'Drained after socializing' . .	105
2.29	Kết quả Random Forest Classifier - Phân lớp 'Drained after socializing'	106
2.30	Kết quả XGBoost Classifier - Phân lớp 'Drained after socializing' . . .	106
2.31	So sánh kết quả các mô hình - Phân lớp 'Drained after socializing' . . .	106
2.32	Kết quả Logistic Classification - Phân lớp 'Stage fear'	107
2.33	Kết quả Random Forest Classifier - Phân lớp 'Stage fear'	107
2.34	Kết quả XGBoost Classifier - Phân lớp 'Stage fear'	108
2.35	So sánh kết quả các mô hình - Phân lớp 'Stage fear'	108
2.36	Một phần bảng dữ liệu Daily Climate Delhi	109
2.37	Thống kê dữ liệu một số đặc trưng dữ liệu Daily Climate Delhi	111
2.38	Kết quả Linear Regression	115
2.39	Kết quả Random Forest Regressor	116
2.40	Kết quả XGBoost Regressor	117
2.41	So sánh kết quả các mô hình (và nghiên cứu liên quan)	117

2.42 Một phần bảng dữ liệu Khám chữa bệnh ST	118
2.43 Thống kê dữ liệu một số đặc trưng dữ liệu Lượt khám chữa bệnh ST	119
2.44 Kết quả Linear Regression	124
2.45 Kết quả Random Forest Regressor	125
2.46 Kết quả XGBoost Regressor	126
2.47 So sánh kết quả các mô hình	126
2.48 Một phần bảng dữ liệu giá cổ phiếu HPG	127
2.49 Thống kê dữ liệu một số đặc trưng dữ liệu giá chứng khoán HPG	128
2.50 Kết quả Linear Regression	134
2.51 Kết quả Random Forest Regressor	135
2.52 Kết quả XGBoost Regressor	135
2.53 So sánh kết quả các mô hình	136
2.54 So sánh kết quả các mô hình	143
2.55 So sánh kết quả các mô hình	153

LỜI CẢM ƠN

Trong suốt hơn hai tháng thực hiện nghiên cứu, nhóm chúng em đã nhận được sự quan tâm, giúp đỡ tận tình từ quý thầy cô và bạn bè. Nhờ sự chia sẻ kinh nghiệm, truyền đạt kiến thức quý báu ấy, nhóm đã có được nền tảng vững chắc để hoàn thành báo cáo.

Đặc biệt, nhóm xin gửi lời cảm ơn sâu sắc đến thầy PGS.TS. Nguyễn Thanh Hải. Thầy đã tận tình hướng dẫn, đồng hành cùng nhóm trong suốt quá trình nghiên cứu. Nhóm vô cùng trân trọng những ý kiến đóng góp quý báu, sự chỉ dẫn tận tâm cũng như tài liệu tham khảo hữu ích mà thầy đã cung cấp, giúp nhóm hoàn thiện bài báo cáo một cách tốt nhất.

Dù đã nỗ lực hết mình, nhưng bài báo cáo vẫn không tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự thông cảm, góp ý từ thầy để tiếp tục hoàn thiện hơn trong tương lai. Cuối cùng, nhóm xin kính chúc thầy cô luôn dồi dào sức khỏe, thành công trong sự nghiệp giảng dạy và nghiên cứu.

Trân trọng!

LỜI CAM ĐOAN

Nhóm chúng em xin cam đoan báo cáo là kết quả nghiên cứu của nhóm trong suốt thời gian qua. Tất cả số liệu, kết quả phân tích trong đề tài đều do nhóm tự tìm hiểu, nghiên cứu một cách khách quan, trung thực, có nguồn gốc rõ ràng và chưa từng được công bố dưới bất kỳ hình thức nào.

Nhóm chúng em cam kết nghiên cứu này không sao chép từ bất kỳ công trình nào khác. Mọi tài liệu tham khảo đều được trích dẫn đầy đủ theo quy định. Nhóm xin chịu hoàn toàn trách nhiệm nếu có bất kỳ sai sót hay sự không trung thực nào trong quá trình thực hiện đề tài. Nghiên cứu của nhóm là trung thực, không sao chép từ các nghiên cứu khác. Có trích dẫn đầy đủ các tài liệu có tham khảo....

DANH MỤC BẢNG VIẾT TẮT

Bảng 1: Danh mục các ký hiệu viết tắt

Chữ viết tắt	Chữ đầy đủ	Điễn giải
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
BoW	Bag of Words	Mô hình túi từ
CART	Classification and Regression Trees	Cây quyết định phân loại và hồi quy
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CV	Coefficient of Variation	Hệ số biến thiên
EM	Expectation-Maximization	Thuật toán EM trong GMM
FNN	Feedforward Neural Network	Mạng truyền thẳng
GB	Gradient Boosting	Tăng cường độ dốc
GBM	Gradient Boosting Machine	Máy học tăng cường độ dốc
GDP	Gross Domestic Product	Tổng sản phẩm quốc nội
GMM	Gaussian Mixture Model	Mô hình hỗn hợp Gauss
HPG	Hoa Phat Group	Mã cổ phiếu của Hòa Phát Group
IQR	Interquartile Range	Khoảng tứ phân vị
KDE	Kernel Density Estimate	Ước lượng mật độ nhân
KNN	K-Nearest Neighbors	Láng giềng gần nhất
LSTM	Long Short-Term Memory	Mạng ghi nhớ dài-ngắn hạn
MAE	Mean Absolute Error	Sai số tuyệt đối trung bình
MAL	My Anime List	CSDL hoạt hình Nhật Bản
MAPE	Mean Absolute Percentage Error	Sai số phần trăm tuyệt đối trung bình
MSE	Mean Squared Error	Sai số bình phương trung bình
OOB	Out-of-Bag	Dữ liệu ngoài túi (Random Forest)
PCA	Principal Component Analysis	Phân tích thành phần chính
RF	Random Forest	Rừng ngẫu nhiên
RMSE	Root Mean Squared Error	Căn bậc hai của sai số bình phương trung bình
RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
RSNA	Radiological Society of North America	Hiệp hội X-quang Bắc Mỹ
SD	Standard Deviation	Độ lệch chuẩn
SGD	Stochastic Gradient Descent	Gradient descent ngẫu nhiên
SIIM	Society for Imaging Informatics in Medicine	Hội Tin học Hình ảnh trong Y học
STM	Short-Term Memory	Trí nhớ ngắn hạn
SVM	Support Vector Machine	Máy vector hỗ trợ
XGBoost	Extreme Gradient Boosting	Tăng cường độ dốc cực đại

TÓM TẮT

Môn học Phân tích Dữ liệu cung cấp cái nhìn toàn diện về các phương pháp phân tích dựa trên dữ liệu với nhiều loại dữ liệu khác nhau như dữ liệu dạng bảng, chuỗi thời gian và dữ liệu hình ảnh. Thông qua các ứng dụng thực tiễn và tập dữ liệu thực tế, sinh viên được học cách áp dụng các kỹ thuật thống kê và thuật toán học máy để khai thác thông tin và đưa ra quyết định dựa trên dữ liệu. Với dữ liệu dạng bảng, các phương pháp phân loại và hồi quy được sử dụng để phân tích kết quả học tập, đặc điểm tính cách và hành vi người dùng. Dữ liệu chuỗi thời gian như khí hậu, lịch sử khám chữa bệnh và giá cổ phiếu được phân tích bằng các mô hình dự báo và phát hiện bất thường. Dữ liệu hình ảnh, bao gồm ảnh chụp y tế và dữ liệu phát hiện cháy, được xử lý qua mạng nơ-ron tích chập (CNN) để thực hiện các tác vụ phân loại và phân đoạn. Môn học nhấn mạnh vào tiền xử lý dữ liệu, trực quan hóa, đánh giá mô hình và diễn giải kết quả, giúp sinh viên trang bị kỹ năng phân tích dữ liệu thực tế.

Từ khóa: *Phân tích dữ liệu, thống kê, máy học,*

CHƯƠNG 1

CƠ SỞ LÝ THUYẾT

1.1 Giới thiệu các thông số thống kê

1.1.1 Mean (Trung bình)

Mean (trung bình số học) là giá trị đại diện cho mức trung bình của một tập hợp số liệu. Đây là một trong những chỉ số trung tâm phổ biến nhất trong thống kê.

- Trung bình cộng được tính theo công thức:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1.1)$$

Trong đó:

- n : Số lượng phần tử trong tập dữ liệu
- x_i : Giá trị của phần tử thứ i

Ví dụ: Cho tập dữ liệu: [3], [5], [7], [10]

$$\bar{x} = \frac{3 + 5 + 7 + 10}{4} = \frac{25}{4} = 6.25$$

1.1.2 Median (Trung vị)

Median (hay còn gọi là trung vị) là giá trị nằm ở giữa của một tập hợp dữ liệu đã được sắp xếp theo thứ tự tăng dần hoặc giảm dần. Nó chia tập dữ liệu thành hai nửa bằng nhau, với một nửa các giá trị nhỏ hơn hoặc bằng trung vị và một nửa các giá trị lớn hơn hoặc bằng trung vị.

Với một dãy số đã được sắp xếp tăng dần, ký hiệu $x_{(1)}, x_{(2)}, \dots, x_{(n)}$, trung vị được tính như sau:

Trường hợp 1: Nếu n là số lẻ

$$\text{Median} = x_{\left(\frac{n+1}{2}\right)} \quad (1.2)$$

Trường hợp 2: Nếu n là số chẵn

$$\text{Median} = \frac{1}{2} \left(x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)} \right) \quad (1.3)$$

Ví dụ: Nếu bạn có một tập dữ liệu như sau: 2, 3, 5, 7, 9, thì trung vị là 5, vì nó nằm ở giữa và có hai giá trị nhỏ hơn (2, 3) và hai giá trị lớn hơn (7, 9).

Nếu tập dữ liệu có số lượng giá trị chẵn, trung vị là trung bình cộng của hai giá trị ở giữa sau khi đã sắp xếp.

Ví dụ, nếu tập dữ liệu là 2, 3, 5, 7, thì trung vị là $\frac{3+5}{2} = 4$.

Khác biệt giữa trung vị và trung bình cộng: Trung bình cộng (mean) là tổng của tất cả các giá trị trong tập dữ liệu chia cho số lượng giá trị, còn trung vị là giá trị ở giữa.

Ưu điểm của Trung vị

- Ít bị ảnh hưởng bởi giá trị ngoại lệ (outliers):** Khác với trung bình cộng, trung vị không bị kéo lệch bởi các giá trị quá lớn hoặc quá nhỏ trong tập dữ liệu.
- Phù hợp với phân bố không chuẩn:** Trung vị là đại lượng thích hợp để mô tả xu hướng trung tâm của các tập dữ liệu có phân bố lệch hoặc không đối xứng.
- Đơn giản và dễ tính:** Việc xác định trung vị chỉ cần sắp xếp dữ liệu theo thứ tự và chọn giá trị giữa, nên không yêu cầu các phép tính phức tạp.

Ứng dụng của Trung vị

- Trong thống kê mô tả:** Được sử dụng để mô tả xu hướng trung tâm trong các báo cáo thống kê, đặc biệt là khi dữ liệu không tuân theo phân phối chuẩn.
- Trong lĩnh vực tài chính:** Trung vị được dùng để tính toán mức lợi nhuận điển hình, giá trị tài sản đầu tư, hoặc thu nhập hộ gia đình để tránh bị ảnh hưởng bởi các giá trị bất thường.
- Trong y học:** Thường được sử dụng để mô tả các chỉ số sinh học như huyết áp, cholesterol... giúp phản ánh tình trạng sức khỏe điển hình trong dân số mà không bị ảnh hưởng bởi các ca bệnh hiếm.
- Trong khoa học dữ liệu và máy học:** Trung vị được dùng để xử lý dữ liệu có phân phối lệch, phát hiện điểm bất thường, hoặc phân chia tập dữ liệu thành các phần có kích thước và đặc tính tương tự nhau (như chia thành tứ phân vị).

1.1.3 Mode (Giá trị xuất hiện nhiều nhất)

Mode là giá trị xuất hiện nhiều nhất trong một tập hợp dữ liệu. Nó là một trong ba đại lượng đo xu hướng trung tâm, cùng với Trung bình (Mean) và Trung vị (Median).

- Mode được tính theo công thức:

$$\text{Mode} = \arg \max_{x_i} (\text{Frequency}(x_i)) \quad (1.4)$$

- Trong thống kê, Mode là giá trị được quan sát thấy tần suất xuất hiện nhiều nhất trong một tập hợp dữ liệu.
- Đối với phân phối chuẩn, Mode có cùng giá trị với giá trị trung bình và trung vị.
- Trong nhiều trường hợp, giá trị của Mode sẽ khác với giá trị trung bình.

Ưu điểm của Mode

- Dễ hiểu và dễ tính toán.
- Không bị ảnh hưởng bởi các giá trị ngoại lai.
- Dễ dàng xác định trong các dữ liệu chưa được gộp và phân phối có tần số rời rạc.
- Hữu ích cho dữ liệu định tính.
- Có thể tính toán với bảng tần số không giới hạn.
- Có thể được xác định bằng hình ảnh.

Nhược điểm của Mode

- Không được định nghĩa rõ ràng.
- Không tạo thành dựa trên tất cả các giá trị trong tập dữ liệu.
- Chỉ ổn định cho số lượng giá trị nhiều và sẽ không được xác định rõ ràng nếu dữ liệu chỉ có một số lượng nhỏ các giá trị.
- Không có khả năng sử dụng tính toán thêm.
- Đôi khi dữ liệu có một hoặc nhiều Mode hoặc không có Mode nào cả.

1.1.4 Percentile (Phân vị)

Phân vị thứ P là giá trị chia tập dữ liệu đã được sắp xếp thành 100 phần bằng nhau, sao cho $P\%$ các quan sát có giá trị nhỏ hơn hoặc bằng giá trị này.

Công thức tính vị trí phân vị

Vị trí lý thuyết của phân vị thứ P trong dãy dữ liệu được sắp xếp tăng dần được xác định bằng công thức:

$$L_P = \frac{P}{100}(n + 1) \quad (1.5)$$

Trong đó:

-
- L_P : Vị trí lý thuyết của phân vị thứ P trong tập dữ liệu đã được sắp xếp (có thể là số thập phân).
 - P : Giá trị phân vị cần tính, thuộc khoảng từ 0 đến 100.
 - n : Tổng số phần tử trong tập dữ liệu.
 - $\frac{P}{100}$: Tỷ lệ phần trăm tương ứng với phân vị thứ P .
 - $n + 1$: Hệ số điều chỉnh để tăng độ chính xác khi nội suy giữa các phần tử.

Trường hợp nội suy

Nếu L_P không phải là số nguyên, ta thực hiện nội suy tuyến tính giữa hai phần tử gần nhất trong tập dữ liệu đã sắp xếp.

Giả sử $L_P = k + d$, trong đó $k \in \mathbb{Z}$, $0 < d < 1$, thì:

$$\text{Percentile}_P = x_k + d \cdot (x_{k+1} - x_k) \quad (1.6)$$

Với:

- x_k : Phần tử ở vị trí thứ k ,
- x_{k+1} : Phần tử kế tiếp sau x_k ,
- d : Phần thập phân của L_P .

Ví dụ minh họa

Giả sử tập dữ liệu đã được sắp xếp như sau:

$$2, 4, 6, 8, 10, 12, 14, 16, 18, 20$$

Với $n = 10$, ta tính phân vị thứ 25:

$$L_{25} = \frac{25}{100}(10 + 1) = 2.75$$

Nội suy giữa giá trị ở vị trí 2 và 3:

$$\text{Percentile}_{25} = 4 + 0.75 \cdot (6 - 4) = 5.5$$

Ứng dụng của phân vị

- Xếp hạng học sinh theo điểm số.

- Đánh giá chỉ số cơ thể trong y tế (BMI, chiều cao...).
- Phân tích dữ liệu để phát hiện ngoại lệ.
- Trực quan hóa phân bố dữ liệu qua biểu đồ hộp (boxplot) với các tứ phân vị (Q1, Q2, Q3).

1.1.5 Max and Min (Giá trị lớn nhất và nhỏ nhất)

Trong thống kê mô tả, hai giá trị quan trọng giúp xác định độ bao phủ của dữ liệu là giá trị lớn nhất và giá trị nhỏ nhất.

Định nghĩa

- Giá trị lớn nhất (Maximum - Max):** là phần tử có giá trị cao nhất trong tập dữ liệu.
- Giá trị nhỏ nhất (Minimum - Min):** là phần tử có giá trị thấp nhất trong tập dữ liệu.

Ký hiệu và công thức

Cho tập dữ liệu gồm n phần tử: $\{x_1, x_2, \dots, x_n\}$, ta có:

- Giá trị lớn nhất (Max):**

$$\max(x_i) = \max\{x_1, x_2, \dots, x_n\}$$

- Giá trị nhỏ nhất (Min):**

$$\min(x_i) = \min\{x_1, x_2, \dots, x_n\}$$

Ví dụ minh họa

Cho tập dữ liệu sau:

$$7, 12, 5, 9, 14, 6$$

Ta có:

$$\begin{aligned}\max(x_i) &= 14 \\ \min(x_i) &= 5\end{aligned}$$

Ứng dụng trong thống kê và học máy

-
- **Tính phạm vi (Range):**

$$\text{Range} = \max(x_i) - \min(x_i)$$

- **Chuẩn hóa dữ liệu (Min-Max normalization):**

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Phương pháp này giúp đưa dữ liệu về khoảng giá trị $[0, 1]$, thường được sử dụng trong học máy để cải thiện hiệu suất huấn luyện mô hình.

- **Phát hiện giá trị ngoại lệ (Outliers):** Các giá trị quá xa so với min hoặc max có thể bị xem là bất thường nếu kết hợp với các phân vị (Q_1, Q_3) và khoảng từ phân vị (IQR).

1.1.6 Range (Khoảng biến thiên)

Phạm vi là một trong những thước đo đơn giản nhất của độ phân tán trong thống kê mô tả. Nó được tính bằng hiệu số giữa giá trị lớn nhất và giá trị nhỏ nhất trong tập dữ liệu.

Công thức tính

$$\text{Range} = \max(x_i) - \min(x_i) \quad (2.4)$$

Trong đó:

- $\max(x_i)$: Giá trị lớn nhất trong tập dữ liệu $\{x_1, x_2, \dots, x_n\}$.
- $\min(x_i)$: Giá trị nhỏ nhất trong tập dữ liệu.
- Range: Phạm vi của dữ liệu, thể hiện độ trai rộng giữa hai giá trị cực trị.

Đặc điểm

- Phạm vi rất dễ tính toán và giúp hình dung nhanh độ biến thiên của dữ liệu.
- Tuy nhiên, phạm vi "rất nhạy cảm với các giá trị ngoại lệ (outliers)", vì nó chỉ xét đến hai giá trị cực trị.
- Phù hợp sử dụng trong những trường hợp dữ liệu không có ngoại lệ rõ rệt hoặc cần ước lượng nhanh.

Ví dụ minh họa

Cho tập dữ liệu:

$$4, 7, 9, 10, 15$$

Ta có:

$$\max(x_i) = 15, \quad \min(x_i) = 4$$

$$\Rightarrow \text{Range} = 15 - 4 = 11$$

1.1.7 Variance (Phương sai)

Phương sai là một đại lượng quan trọng dùng để đo mức độ phân tán (biến thiên) của dữ liệu so với giá trị trung bình. Nó cho biết các quan sát trong tập dữ liệu cách xa trung bình cộng bao nhiêu, tính theo bình phương.

Ký hiệu và công thức

Cho một tập dữ liệu $\{x_1, x_2, \dots, x_n\}$, với trung bình cộng là \bar{x} , ta có:

- **Phương sai mẫu (Sample variance)** được tính theo công thức:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (1.7)$$

- **Phương sai tổng thể (Population variance):**

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (1.8)$$

Trong đó:

- s^2 : phương sai mẫu.
- σ^2 : phương sai tổng thể.
- n : số phần tử trong mẫu hoặc tổng thể.
- x_i : giá trị quan sát thứ i .
- \bar{x} : trung bình mẫu.
- μ : trung bình tổng thể (nếu biết).

Giải thích ý nghĩa

Phương sai đo lường trung bình bình phương khoảng cách từ mỗi điểm dữ liệu đến trung bình. Giá trị phương sai càng lớn cho thấy dữ liệu phân tán rộng, phương sai càng nhỏ cho thấy dữ liệu tập trung gần trung bình.

Ví dụ minh họa

Cho tập dữ liệu:

$$x = \{2, 4, 6, 8, 10\}$$

Tính trung bình:

$$\bar{x} = \frac{2 + 4 + 6 + 8 + 10}{5} = 6$$

Tính phương sai mẫu:

$$s^2 = \frac{1}{5-1} [(2-6)^2 + (4-6)^2 + (6-6)^2 + (8-6)^2 + (10-6)^2] = \frac{1}{4}(16+4+0+4+16) = \frac{40}{4} = 10$$

Ứng dụng

- Đo lường độ biến thiên trong dữ liệu.
- So sánh mức độ rủi ro trong tài chính (phân tán lợi nhuận).
- Là thành phần cơ bản để tính độ lệch chuẩn, hệ số tương quan, kiểm định giả thuyết.
- Dùng trong nhiều mô hình thống kê và học máy (hồi quy tuyến tính, PCA, v.v.).

1.1.8 Standard Deviation (Độ lệch chuẩn)

Độ lệch chuẩn là một thước đo thống kê phổ biến nhằm xác định mức độ phân tán của dữ liệu quanh giá trị trung bình. Nó là căn bậc hai của phương sai, giúp đưa đơn vị đo về cùng đơn vị với dữ liệu gốc, từ đó dễ diễn giải hơn.

Công thức

- **Độ lệch chuẩn mẫu (Sample standard deviation):**

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (1.9)$$

- Độ lệch chuẩn tổng thể (Population standard deviation):

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (1.10)$$

Trong đó:

- s : Độ lệch chuẩn mẫu.
- σ : Độ lệch chuẩn tổng thể.
- n : Số phần tử trong mẫu hoặc tổng thể.
- x_i : Giá trị quan sát thứ i .
- \bar{x} : Trung bình mẫu.
- μ : Trung bình tổng thể.

Ý nghĩa

Độ lệch chuẩn cho biết mức độ phân tán trung bình của các điểm dữ liệu so với trung bình cộng:

- Độ lệch chuẩn **càng nhỏ** → dữ liệu **tập trung quanh trung bình**.
- Độ lệch chuẩn **càng lớn** → dữ liệu **phân tán rộng hơn**.

Ví dụ minh họa

Cho tập dữ liệu:

$$x = \{2, 4, 6, 8, 10\}$$

Trung bình:

$$\bar{x} = \frac{2 + 4 + 6 + 8 + 10}{5} = 6$$

Tính độ lệch chuẩn mẫu:

$$s = \sqrt{\frac{1}{4} [(2-6)^2 + (4-6)^2 + (6-6)^2 + (8-6)^2 + (10-6)^2]} = \sqrt{\frac{40}{4}} = \sqrt{10} \approx 3.16$$

Ứng dụng

-
- Được sử dụng rộng rãi trong thống kê mô tả để đo lường sự biến động.
 - Là cơ sở cho nhiều phương pháp kiểm định giả thuyết (t-test, ANOVA...).
 - Giúp đánh giá độ rủi ro trong tài chính, ví dụ như độ biến động của giá cổ phiếu.
 - Được sử dụng trong học máy và khai phá dữ liệu để chuẩn hóa dữ liệu.

1.1.9 Coefficient of Variation - CV (Hệ số biến thiên)

Hệ số biến thiên là một chỉ số thống kê đo lường mức độ phân tán tương đối của dữ liệu so với trung bình cộng. Nó biểu thị độ lệch chuẩn dưới dạng phần trăm của giá trị trung bình.

Công thức

- **Hệ số biến thiên của mẫu:**

$$CV = \frac{s}{\bar{x}} \times 100\% \quad (1.11)$$

- **Hệ số biến thiên của tổng thể:**

$$CV = \frac{\sigma}{\mu} \times 100\% \quad (1.12)$$

Trong đó:

- CV : Hệ số biến thiên (thường biểu thị bằng phần trăm).
- s : Độ lệch chuẩn mẫu.
- \bar{x} : Trung bình mẫu.
- σ : Độ lệch chuẩn tổng thể.
- μ : Trung bình tổng thể.

Ý nghĩa

- CV đo lường mức độ biến động của dữ liệu **tương đối** so với trung bình.
- Hữu ích khi so sánh độ phân tán của các tập dữ liệu có đơn vị đo hoặc giá trị trung bình khác nhau.
- CV càng cao \rightarrow dữ liệu biến động càng lớn.
- CV càng thấp \rightarrow dữ liệu ổn định và tập trung hơn quanh trung bình.

Ví dụ minh họa

Cho tập dữ liệu sau:

$$x = \{2, 4, 6, 8, 10\}$$

Ta đã biết:

$$\bar{x} = 6$$

$$s = \sqrt{10} \approx 3.16$$

Khi đó, hệ số biến thiên:

$$CV = \frac{3.16}{6} \times 100\% \approx 52.67\%$$

Ứng dụng

- So sánh độ biến động của các chỉ số kinh tế, tài chính, năng suất sản xuất giữa các nhóm khác nhau.
- Đánh giá độ ổn định của dữ liệu trong nghiên cứu thí nghiệm, kiểm tra chất lượng, phân tích định lượng.
- Trong tài chính, CV được sử dụng để đánh giá rủi ro tương đối của các khoản đầu tư.

1.1.10 Interquartile Range – IQR (Khoảng tứ phân vị)

Khoảng tứ phân vị (IQR – Interquartile Range) là một thước đo thống kê thể hiện phạm vi phân bố của nhóm dữ liệu trung bình, bằng hiệu số giữa tứ phân vị thứ ba (Q_3) và tứ phân vị thứ nhất (Q_1).

Công thức tính IQR

$$IQR = Q_3 - Q_1 \quad (1.13)$$

Trong đó:

- Q_1 : Tứ phân vị thứ nhất (25% số quan sát nhỏ hơn hoặc bằng).
- Q_3 : Tứ phân vị thứ ba (75% số quan sát nhỏ hơn hoặc bằng).
- IQR: Khoảng tứ phân vị, biểu thị phạm vi của 50% dữ liệu trung tâm.

Ý nghĩa của IQR

- IQR đo lường độ phân tán của dữ liệu theo cách **ít bị ảnh hưởng bởi các giá trị ngoại lệ (outliers)** so với độ lệch chuẩn.
- Thường dùng để xác định phạm vi dữ liệu “bình thường” và phát hiện ngoại lệ.

Phát hiện ngoại lệ bằng IQR

Các điểm dữ liệu được xem là ngoại lệ nếu nằm ngoài khoảng:

$$[Q_1 - 1.5 \cdot \text{IQR}, \quad Q_3 + 1.5 \cdot \text{IQR}]$$

Ví dụ minh họa

Cho tập dữ liệu sau (đã được sắp xếp):

$$x = \{1, 3, 5, 7, 9, 11, 13, 15, 17\}$$

Ta xác định:

- $Q_1 = 5$ (tứ phân vị thứ nhất)
- $Q_3 = 13$ (tứ phân vị thứ ba)

Khi đó:

$$\text{IQR} = Q_3 - Q_1 = 13 - 5 = 8$$

Các giá trị ngoại lệ khoảng:

$$[Q_1 - 1.5 \cdot \text{IQR}, \quad Q_3 + 1.5 \cdot \text{IQR}] = [5 - 12, \quad 13 + 12] = [-7, \quad 25]$$

→ Không có ngoại lệ trong tập dữ liệu này.

Ứng dụng

- Dùng trong biểu đồ hộp (boxplot) để mô tả sự phân bố dữ liệu.
- Phân tích độ biến thiên của dữ liệu mà không bị ảnh hưởng bởi các giá trị cực đoan.
- Hữu ích trong kiểm tra chất lượng, dữ liệu sinh học, tài chính và học máy.

1.2 Giới thiệu các biểu đồ trực quan

1.2.1 Histogram (Biểu đồ tần suất)

Biểu đồ Histogram là một biểu đồ hình cột dùng để biểu diễn phân bố tần suất hoặc tần số của một tập dữ liệu định lượng. Histogram chia dữ liệu thành các khoảng (còn gọi là **bins**) và thể hiện số lượng phần tử (hoặc tần suất) rơi vào mỗi khoảng.

Các thành phần chính của biểu đồ Histogram

- **Trục hoành (trục x):** thể hiện các **khoảng giá trị** (bins) — được chia đều hoặc không đều tùy vào dữ liệu.
- **Trục tung (trục y):** thể hiện **số lượng** (tần số) hoặc **tần suất tương đối** trong mỗi bin.
- **Cột (bars):** chiều cao của mỗi cột thể hiện số phần tử thuộc bin đó.

Cách xây dựng Histogram

1. Xác định giá trị nhỏ nhất và lớn nhất trong tập dữ liệu.
2. Chia dữ liệu thành k khoảng (bins) theo công thức như Sturges:

$$k = \lceil \log_2 n + 1 \rceil$$

trong đó n là số quan sát.

3. Tính độ rộng mỗi khoảng (bin width):

$$\text{Bin width} = \frac{\text{Max} - \text{Min}}{k}$$

4. Đếm số lượng phần tử rơi vào mỗi bin.
5. Vẽ biểu đồ cột với chiều cao tương ứng.

Ví dụ minh họa

Cho tập dữ liệu:

$$x = \{5, 7, 8, 9, 10, 10, 11, 13, 13, 15, 16, 18, 20\}$$

Số phần tử: $n = 13$ Tính số bin theo công thức Sturges:

$$k = \lceil \log_2 13 + 1 \rceil = \lceil 3.7 + 1 \rceil = 5$$

Khoảng giá trị: [5, 20], độ rộng bin:

$$\text{Bin width} = \frac{20 - 5}{5} = 3$$

Các bin: [5 – 8), [8 – 11), [11 – 14), [14 – 17), [17 – 20]

Đếm số phần tử trong từng bin và vẽ biểu đồ cột.

Ứng dụng của Histogram

- Phân tích dạng phân bố dữ liệu: chuẩn, lệch trái, lệch phải.
- Kiểm tra sự tồn tại của ngoại lệ hoặc cụm giá trị bất thường.
- Là bước đầu tiên trong phân tích thống kê mô tả và trực quan hóa dữ liệu.

1.2.2 Box Plot (Biểu đồ hộp)

Biểu đồ hộp (Box plot), còn gọi là biểu đồ hộp-tia (box-and-whisker plot), là một công cụ trực quan hóa dữ liệu rất hiệu quả trong thống kê mô tả. Nó cho phép mô tả phân bố của tập dữ liệu theo các tóm tắt tứ phân vị, đồng thời phát hiện các giá trị ngoại lệ (outliers).

Các thành phần của biểu đồ hộp

Một biểu đồ hộp gồm các thành phần chính:

- **Q1 (Tứ phân vị thứ nhất)** – 25% dữ liệu nhỏ hơn hoặc bằng.
- **Q2 (Trung vị – Median)** – 50% dữ liệu nhỏ hơn hoặc bằng.
- **Q3 (Tứ phân vị thứ ba)** – 75% dữ liệu nhỏ hơn hoặc bằng.
- **IQR (Interquartile Range)** – $Q_3 - Q_1$, phạm vi của 50% dữ liệu trung tâm.
- **Whiskers (tia)** – thể hiện phạm vi dữ liệu không bị xem là ngoại lệ:

$$\text{Lower whisker} = Q_1 - 1.5 \cdot \text{IQR}$$

$$\text{Upper whisker} = Q_3 + 1.5 \cdot \text{IQR}$$

- **Outliers (ngoại lệ)** – các điểm nằm ngoài khoảng trên.

Ý nghĩa biểu đồ hộp

- Thể hiện trực quan độ phân tán, độ lệch và tính đối xứng của dữ liệu.
- So sánh phân bố giữa nhiều nhóm.
- Phát hiện nhanh các giá trị ngoại lệ.

Ví dụ minh họa

Giả sử có tập dữ liệu:

$$x = \{1, 3, 4, 5, 6, 7, 8, 10, 12, 15\}$$

Các giá trị thống kê:

$$Q_1 = 4$$

$$Q_2 = 6$$

$$Q_3 = 10$$

$$\text{IQR} = Q_3 - Q_1 = 6$$

$$\text{Whisker dưới} = 4 - 1.5 \cdot 6 = -5$$

$$\text{Whisker trên} = 10 + 1.5 \cdot 6 = 19$$

Vì tất cả giá trị thuộc khoảng $[-5, 19]$, không có ngoại lệ trong tập dữ liệu.

Ứng dụng

- So sánh sự phân bố dữ liệu giữa nhiều nhóm (ví dụ: điểm kiểm tra giữa các lớp).
- Phân tích nhanh độ lệch (skewness) và độ biến thiên.
- Dùng trong các biểu đồ thống kê y tế, tài chính, sản xuất và học máy.

1.2.3 Scatter Plot (Biểu đồ phân tán)

Biểu đồ phân tán (Scatter Plot) là một biểu đồ dạng điểm, dùng để biểu diễn mối quan hệ giữa hai biến số định lượng. Mỗi điểm trên biểu đồ biểu thị một cặp giá trị (x_i, y_i) của hai biến.

Cấu trúc của biểu đồ

- Trục hoành (Ox):** biểu diễn biến độc lập (hoặc biến đầu vào) x .
- Trục tung (Oy):** biểu diễn biến phụ thuộc (hoặc biến đầu ra) y .
- Các điểm dữ liệu:** mỗi điểm (x_i, y_i) là một quan sát trong tập dữ liệu.

Ý nghĩa

- Cho phép xác định trực quan mối quan hệ giữa hai biến:

-
- Quan hệ tuyến tính dương (cùng tăng).
 - Quan hệ tuyến tính âm (một tăng, một giảm).
 - Không có mối quan hệ rõ ràng.
- Phát hiện xu hướng, cụm dữ liệu, điểm bất thường (outliers).

Ví dụ minh họa

Cho tập dữ liệu gồm chiều cao (cm) và cân nặng (kg) của 5 người:

Chiều cao (x)	Cân nặng (y)
160	50
165	55
170	60
175	66
180	72

Khi vẽ các điểm (x, y) này trên mặt phẳng tọa độ, ta có thể thấy một xu hướng tuyến tính dương — tức là người cao hơn thường nặng hơn.

Ứng dụng

- Kiểm tra mối tương quan giữa hai biến trong phân tích thống kê và hồi quy.
- Trực quan hóa dữ liệu trong các lĩnh vực như kinh tế, sinh học, kỹ thuật.
- Dùng để phát hiện ngoại lệ hoặc cấu trúc đặc biệt trong dữ liệu.

1.2.4 Heatmap (Biểu đồ nhiệt)

Biểu đồ nhiệt (Heatmap) là một hình thức trực quan hóa dữ liệu trong đó các giá trị được biểu diễn thông qua màu sắc. Mỗi ô vuông trong biểu đồ tương ứng với một giá trị số trong bảng dữ liệu, với màu sắc càng đậm hoặc thay đổi sắc độ thể hiện giá trị càng lớn hoặc nhỏ.

Thành phần chính của biểu đồ Heatmap

- **Trục hoành và trục tung:** đại diện cho hai chiều của ma trận dữ liệu (ví dụ: biến hoặc nhóm).
- **Mỗi ô (cell):** biểu thị một giá trị số cụ thể được ánh xạ thành màu sắc.
- **Thang màu (color scale):** chỉ thị giá trị nhỏ đến lớn bằng các gam màu từ nhạt đến đậm, lạnh đến nóng...

Ứng dụng của Heatmap

- Phân tích tương quan giữa các biến (corr matrix).
- Biểu diễn cường độ hoạt động, tần suất, mật độ (density).
- Phân tích dữ liệu lớn có cấu trúc ma trận như trong học máy, sinh học, tài chính.
- So sánh giá trị giữa các nhóm, hạng mục trong một bảng nhiều chiều.

Ví dụ minh họa

Giả sử ta có bảng tương quan giữa các biến:

$$\begin{bmatrix} 1.00 & 0.85 & 0.30 \\ 0.85 & 1.00 & 0.60 \\ 0.30 & 0.60 & 1.00 \end{bmatrix}$$

Biểu đồ nhiệt sẽ dùng các màu khác nhau để thể hiện độ tương quan (càng gần 1, màu càng đậm).

1.2.5 Pie Plot (Biểu đồ tròn)

Biểu đồ tròn (Pie Plot) là một dạng biểu đồ hình tròn dùng để biểu diễn tỷ lệ phần trăm (%) của các thành phần trong một tổng thể. Mỗi phần hình quạt tương ứng với một hạng mục và có diện tích tỉ lệ với tỷ trọng của hạng mục đó.

Đặc điểm chính

- Tổng các phần của biểu đồ luôn bằng 100% hoặc 360° .
- Kích thước của mỗi phần được tính theo:

$$\theta_i = \frac{x_i}{\sum x_i} \cdot 360^\circ$$

Trong đó:

- x_i : giá trị của nhóm thứ i ,
- θ_i : góc của nhóm thứ i trong biểu đồ.
- Mỗi phần thường có màu sắc và nhãn riêng để dễ phân biệt.

Ví dụ minh họa

Xét tỉ lệ sinh viên theo ngành học trong một lớp:

- Kỹ thuật: 40 sinh viên

-
- Kinh tế: 30 sinh viên
 - Sư phạm: 20 sinh viên
 - Luật: 10 sinh viên

Tổng số sinh viên: 100

- Kỹ thuật: 40% $\rightarrow 144^\circ$
- Kinh tế: 30% $\rightarrow 108^\circ$
- Sư phạm: 20% $\rightarrow 72^\circ$
- Luật: 10% $\rightarrow 36^\circ$

Ứng dụng

- Hiển thị thành phần tỷ lệ của các nhóm dữ liệu rời rạc.
- Thường dùng trong báo cáo tài chính, thống kê khảo sát, thống kê dân số.
- Thích hợp với dữ liệu ít nhóm (dưới 6 nhóm để dễ đọc).

1.2.6 Bar Plot (Biểu đồ cột)

Biểu đồ cột (Bar Plot) là một trong những dạng biểu đồ thống kê phổ biến, dùng để thể hiện và so sánh số lượng, tần suất hoặc giá trị giữa các nhóm hạng mục rời rạc (danh mục).

Thành phần của biểu đồ cột

- **Trục hoành (trục x):** biểu diễn các hạng mục (categories), ví dụ: các nhóm tuổi, khoa, khu vực, năm...
- **Trục tung (trục y):** biểu diễn số lượng hoặc giá trị tương ứng của từng hạng mục.
- **Cột (bars):** chiều cao của mỗi cột tương ứng với giá trị của hạng mục đó.

Công thức (nếu dùng dữ liệu đếm)

Chiều cao cột = Tần suất hoặc giá trị tương ứng $= x_i$

Ví dụ minh họa

Giả sử thống kê số lượng sinh viên ở các khoa như sau:

- CNTT: 120
- Kinh tế: 80
- Y: 60
- Môi trường: 40

Ứng dụng của biểu đồ cột

- So sánh số lượng hoặc giá trị giữa các nhóm rời rạc.
- Dùng trong báo cáo thống kê, trực quan hóa dữ liệu, khảo sát thị trường,...
- Kết hợp với màu sắc hoặc phân nhóm để thể hiện thêm chiều thông tin.

1.2.7 Line Plot (Biểu đồ đường)

Biểu đồ đường (Line Plot hoặc Line Chart) là một loại biểu đồ dùng để hiển thị dữ liệu dạng chuỗi (theo thời gian hoặc thứ tự) bằng cách nối các điểm dữ liệu bằng các đoạn thẳng. Đây là công cụ hiệu quả để biểu diễn xu hướng thay đổi của dữ liệu theo thời gian hoặc các giai đoạn.

Thành phần của biểu đồ

- **Trục hoành (Ox):** biểu diễn chuỗi thời gian hoặc thứ tự quan sát (ngày, tháng, năm, phiên, v.v.).
- **Trục tung (Oy):** biểu diễn giá trị tương ứng tại từng thời điểm.
- **Các điểm dữ liệu (data points):** là các cặp (x_i, y_i) .
- **Đường nối (lines):** nối liên tiếp các điểm dữ liệu để thể hiện xu hướng.

Ứng dụng

- Phân tích xu hướng theo thời gian (biến động giá, dân số, nhiệt độ, doanh thu...).
- So sánh sự thay đổi giữa nhiều biến theo thời gian.
- Trực quan hóa dữ liệu chuỗi thời gian trong thống kê, tài chính, khoa học, và học máy.

Ví dụ minh họa

Xét số lượng khách truy cập website theo tháng:

Tháng	Số lượt truy cập (nghìn)
1	15
2	18
3	21
4	25
5	24
6	27

Biểu đồ đường sẽ nối các điểm tương ứng: $(1, 15), (2, 18), \dots, (6, 27)$

1.3 Giới thiệu các mô hình máy học

1.3.1 Random Forest (Rừng ngẫu nhiên)

1.3.1.1 Khái niệm: Bootstrapping và Bagging

Bootstrapping là một phương pháp nổi tiếng trong thống kê được giới thiệu bởi Bradley Efron vào năm 1979 [10]. Phương pháp này thực hiện lặp lại nhiều lần:

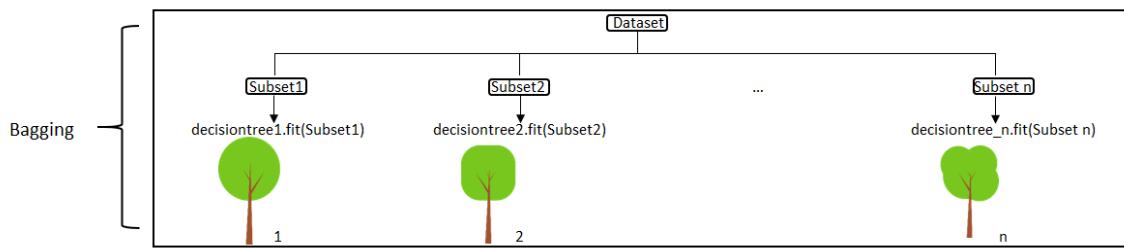
- Lấy ra một mẫu có hoàn lại từ mẫu ban đầu
- Ước tính các tham số mong muốn từ mẫu lấy được

Từ đó, ước lượng được các tham số mong muốn từ mẫu ban đầu. Phương pháp này chủ yếu dùng để ước lượng lỗi chuẩn (standard errors), độ lệch (bias) và tính toán khoảng tin cậy (confidence interval) cho các tham số.

Bootstrap Aggregating (Bagging) là thuật toán kết hợp các mô hình máy học. Thuật toán sử dụng phương pháp Bootstrap để tạo nhiều bộ dữ liệu đầu vào từ một bộ dữ liệu, sau đó xây dựng các độc lập mô hình có cùng thuật toán nhưng với từng bộ dữ liệu đầu vào khác nhau. Các mô hình này sẽ được kết hợp bằng phương pháp biểu quyết đa số (majority voting).

1.3.1.2 Khái niệm: Random Forest

Random Forest [5], hay còn gọi là Random Decision Forest, là một phương pháp ensemble, kết hợp nhiều mô hình cây quyết định phân lớp/hồi quy (Classification and regression trees - CART), cải tiến của phương pháp Bagging. RF được phát triển bởi Leo Breiman. Ông đồng thời cũng là tác giả của CART [4]



Hình 1.1: Minh họa: Bagging

1.3.1.3 Mô tả thuật toán xây dựng Random Forest

- Bước 1: Chọn ngẫu nhiên k từ n thuộc tính ($k < n$) có trong bộ dữ liệu huấn luyện.
- Bước 2: Sử dụng phương pháp bootstrapping, chọn có hoán lại từ bộ dữ liệu huấn luyện để tạo ra một bộ dữ liệu mới.
- Bước 3: Xây dựng cây quyết định với bộ dữ liệu ở bước 2, chỉ sử dụng k thuộc tính đã chọn ở bước 1
- Bước 4: Lặp lại Bước 2-3 để tạo nhiều cây quyết định khác.
- Bước 5: Kết hợp các cây quyết định thành phần bằng phương pháp bỏ phiếu.



Hình 1.2: Minh họa: Random Forest [22]

1.3.1.4 Dự đoán bằng Random Forest

Với x là một điểm dữ liệu mới, mỗi cây thành phần dự đoán hồi quy $T_i(x)$, dự đoán hồi quy của mô hình là:

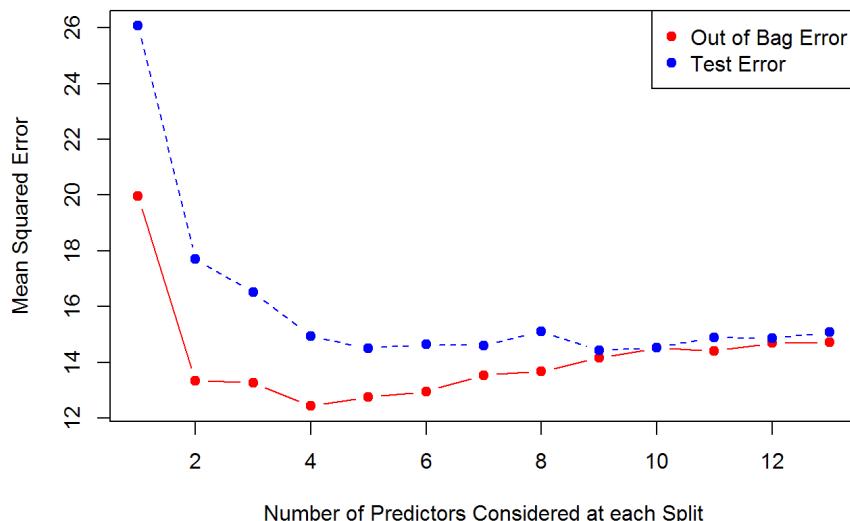
$$\hat{f}_{rf}^N(x) = \frac{1}{N} \sum_{i=1}^N T_i(x) \quad (1.14)$$

Với $\hat{T}_i(x)$ là kết quả phân lớp của cây thành phần, dự đoán phân lớp của mô hình là:

$$\hat{C}_{rf}^N(x) = \text{majority vote } \{\hat{T}_i(x)\}_1^N \quad (1.15)$$

1.3.1.5 Đánh giá Random Forest

Do sử dụng phương pháp bootstrapping để tạo các mẫu huấn luyện cho các CART thành phần, chỉ có khoảng $\frac{2}{3}$ dữ liệu không trùng lặp trong bộ huấn luyện ban đầu được sử dụng cho việc huấn luyện. $\frac{1}{3}$ dữ liệu còn lại không tham gia huấn luyện được gọi là 'Out-of-bag dataset' (OOB). Phần dữ liệu này có thể xem như là một tập kiểm thử (validation dataset) dùng để đánh giá và tính toán độ quan trọng các thuộc tính của các CART trong rừng. Độ lỗi của RF trên tập dữ liệu OOB được gọi là 'Out-of-bag Error' (E_{OOB}).



Hình 1.3: Tương quan Out-of-bag error và test error

Có thể so sánh E_{OOB} giữa các RF có số thuộc tính được chọn k (ở bước 1) khác nhau và số lượng cây trong rừng khác nhau để chọn được mô hình RF có độ lỗi thấp nhất.

1.3.1.6 Điểm mạnh của Random Forest

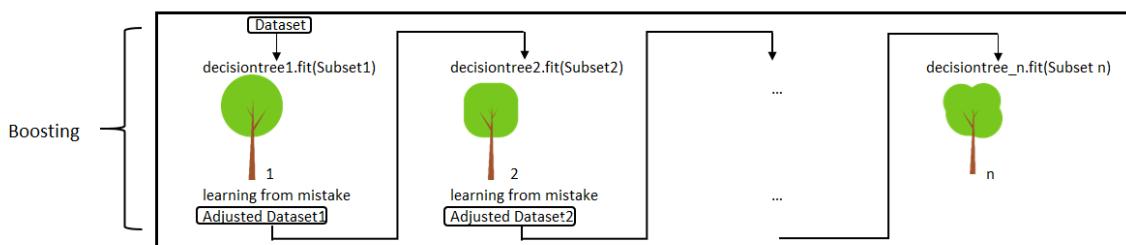
Thuật toán xây dựng có sự ngẫu nhiên trong mẫu dữ liệu (dùng phương pháp bootstrapping) và ngẫu nhiên trong số thuộc tính của mẫu so với tập thuộc tính ban đầu. Do đó, các tập huấn luyện con được tạo ra có tính đa dạng, ít liên quan. Mỗi CART xây dựng từ những tập dữ liệu con này không dùng tất cả dữ liệu training, cũng như không dùng tất cả các thuộc tính của dữ liệu để xây dựng nên mỗi cây có bias cao. Tuy nhiên, kết quả cuối của RF là kết hợp của nhiều cây thành phần, thông tin từ các cây sẽ bổ sung thông tin cho nhau, dẫn đến mô hình có low bias và low variance, tức là mô hình có kết quả dự đoán tốt.

Trong rừng, mỗi cây thành phần chỉ được huấn luyện trên một tập nhỏ các thuộc tính thay vì toàn bộ (bước 1), cơ chế này giúp RF thực thi nhanh khi áp dụng trên tập dữ liệu có số lượng lớn thuộc tính. Hơn nữa, việc xây dựng từng cây thành phần là độc lập nên có thể dễ dàng thực hiện song song.

1.3.2 Gradient Boosting (Tăng cường độ dốc)

1.3.2.1 Khái niệm: Boosting

Boosting là một kỹ thuật ensemble với mục đích từ một số mô hình học yếu (weak learner) tạo ra mô hình học mạnh hơn (strong learner) bằng cách cho những mô hình sau sửa lỗi (học từ lỗi) của các mô hình trước. Các mô hình được thêm vào theo cách này cho đến khi đạt số lượng mô hình tối đa cho phép hoặc dự đoán hoàn toàn trùng khớp với tập huấn luyện.



Hình 1.4: Minh họa: Boosting

Thuật toán boosting đơn giản nhất là Ada Boost, được giới thiệu vào năm 1995 bởi Freund và Schapire [12]. Thuật toán này sử dụng các CART có độ sâu nhỏ, hay còn gọi là một gốc (stump), làm các weak learner thành phần.

Ada Boost thực hiện việc học tăng cường bằng cách gán cho mỗi điểm dữ liệu trong bộ huấn luyện bộ huấn luyện một trọng số mẫu (sample weight). Ban đầu, tất cả điểm dữ liệu đều được khởi tạo trọng số này bằng $\frac{1}{n}$, với n là tổng số điểm dữ liệu. Sau đó, với mỗi gốc được tạo, tổng lỗi của gốc đó sẽ quyết định trọng số của gốc trong việc bỏ phiếu (weighted voting) cho kết quả dự đoán cuối. Những điểm dữ liệu cũng được cập nhật lại sample weight dựa vào kết quả gốc hiện tại dự đoán đúng (sample weight

giảm) hay dự đoán sai (sample wieght tăng) thể hiện gốc sau đó cần dự thiết dự đoán đúng các điểm dữ liệu đang được dự đoán sai (gốc tiếp theo cố gắng sửa lỗi của gốc trước đó)

Cụ thể thuật toán Ada Boost như sau [11]:

- Khởi tạo $w_1 = \{\frac{1}{N}\}_1^N$
- Với M là số gốc tối đa, lặp lại m=1,...M:

- Huấn luyện gốc sử dụng sample weight w_m
- Tính độ lỗi:

$$\epsilon_t = Pr_{i \in D_t}[h_t \neq y_i] \quad (1.16)$$

- Thiết lập trọng số cho gốc này:

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t} \quad (1.17)$$

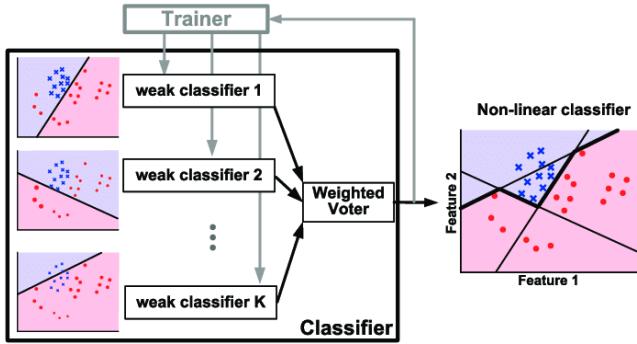
- Cập nhật sample wieght cho việc huấn luyện gốc tiếp theo:

$$D_{t+1} = \frac{D_t(i)e^{-\alpha_t}}{Z_t} \quad (1.18)$$

(với Z_t là hệ số chuẩn hóa để tổng sample wieght bằng 1)

- Cuối cùng, kết hợp các gốc bằng bỏ phiếu có trọng số:

$$H(x) = sign(\sum_{t=1}^T \alpha_t h_t(x)) \quad (1.19)$$



Hình 1.5: Minh họa thuật toán AdaBoost, quá trình kết hợp các mô hình yếu [30]

1.3.2.2 Gradient Boosting

Ý tưởng của thuật toán Gradient Boosting (GB) bắt nguồn từ Leo Breiman. Ông cho rằng phương pháp boosting có thể xem như là một thuật toán tối ưu hóa trên một hàm mất mát (Loss function) phù hợp [3]. Thuật toán Gradient boosting sau đó được phát triển bởi Jerome H. Friedman [13] [14].

Khác với Ada Boost, ở Gradient Boosting:

- Sử dụng cây phân lớp, hồi quy làm các weaker learner thành phần thay vì gốc
- Những CART thành phần không huấn luyện dựa trên trọng số mẫu (mô hình tiếp theo cố gắng dự đoán đúng những điểm lỗi của mô hình trước), mà dựa trên 'độ lệch (lỗi) giả'(pseudo-residual) của CART tiền nhiệm trước đó (mô hình tiếp theo cố gắng giảm độ lỗi có được từ mô hình trước).
- Những mô hình thành phần không có trọng số riêng, thay vào đó chúng có chung một tỉ lệ tốc độ học (learning rate) trong khoảng 0-1 để điều chỉnh tốc độ học của từng mô hình mới.

Thuật toán kết thúc khi đạt số lượng cây tối đa hoặc việc thêm những cây mới không giảm pseudo-residual.

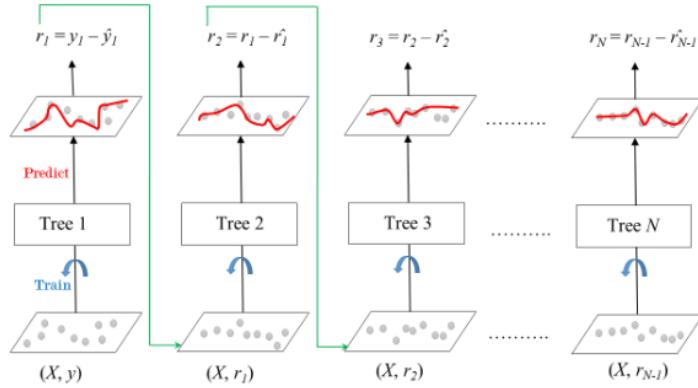
1.3.2.3 Thuật toán Gradient Boosting [26] [27]:

Input: Dữ liệu huấn luyện $\{(x_i, y_i)\}_{i=1}^n$, một hàm Loss $L(y, F(x))$ và số lượng cây tối đa M

- Bước 1: Khởi tạo dự đoán ban đầu với hằng số

$$F_0(x) = \operatorname{argmin}_\gamma \sum_{i=1}^n L(y_i, \gamma) \quad (1.20)$$

- Bước 2: Lặp m=1 đến M để tạo M cây thành phần:



Hình 1.6: Minh họa: Gradient boosting [20]

(A) Tính 'độ lệch giả' (pseudo-residual) của các điểm dữ liệu trong tập huấn luyện: Với mỗi $i = 1, \dots, n$

$$r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)} \quad (1.21)$$

(B) Xây dựng cây dự đoán các giá trị r_{im} , gọi R_{jm} là các lá của cây, với $j=1, \dots, J_m$ (cây có J_m lá)

(C) Tính giá trị output ở từng lá của cây vừa tạo :

$$\gamma_{jm} = \operatorname{argmin}_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma) \quad (1.22)$$

(D) Cập nhật dự đoán mới:

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \quad (1.23)$$

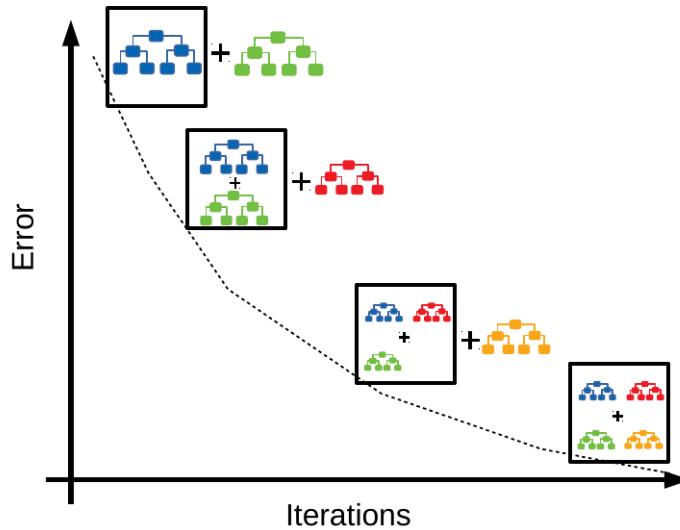
với ν là tỷ lệ tốc độ học (learning rate)

Output: Kết quả dự đoán là kết quả của cây cuối cùng: $F_M(x)$

1.3.2.4 Điểm mạnh của Gradient Boosting

Qua mỗi cây mới được thêm vào, độ lỗi của mô hình sẽ ngày càng được giảm. Với các siêu tham số (hyperparameter) hợp lý, mô hình có thể cho được độ chính xác cao.

Gradient Boosting có thể được tối ưu với nhiều hàm Loss khác nhau và cho phép hiệu chỉnh (fine tuning) nhiều siêu tham số (hyperparameter) giúp mô hình có độ linh hoạt cao giữa nhiều bộ dữ liệu khác nhau.



Hình 1.7: Độ lỗi qua các iteration - GB [15]

1.3.3 XGBoost (Extreme Gradient Boosting)

XGBoost, viết tắt cho eXtreme Gradient Boosting, là một phiên bản Gradient Boosting được tối ưu hóa bởi Tianqi Chen [7].

Để cải tiến GB, mục tiêu của XGBoost không chỉ cố gắng tối thiểu hóa hàm Loss mà còn kèm theo một hàm chính quy hóa (regularization). Ta có thể gọi hàm mục tiêu (objective function) của XGBoost là: [8]

$$obj(\theta) = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (1.24)$$

với L là hàm Loss, Ω là hàm regularization, f_k là mô hình thành phần thứ k. Việc thêm hàm regularization giúp cải tiến vấn đề overfit của mô hình Gradient Boosting, do sau mỗi lần thêm cây thành phần, độ lỗi của mô hình GB luôn được giảm cho đến tối thiểu hoặc mô hình đạt tối đa số cây thành phần.

1.3.3.1 Xây dựng CART trong mô hình XGBoost

Một trong những cải tiến của XGBoost so với Gradient Boost là ở việc tạo các cây hồi quy/phân lớp thành phần.

Ở iteration thứ t, cây được tạo cho dự đoán

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) = \sum_{k=1}^T \Omega(f_k(x_i)) \quad (1.25)$$

Độ phức tạp của cây được định nghĩa là [8]

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (1.26)$$

với γ và λ là các hệ số regularization, T là số lá của cây. Ta thấy $\sum_{j=1}^T w_j^2$ là tổng bình phương các lá, hay chính là tổng bình phương output của cây. Ta có thể đặt:

$$O_{value}^2 = f_t(x_i)^2 = \sum_{j=1}^T w_j^2 \quad (1.27)$$

khi đó độ phức tạp của cây là:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda O_{value}^2 \quad (1.28)$$

Lúc này, hàm mục tiêu của XGBoost là [8] [28]

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \\ &= \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + O_{value}) + \gamma T + \frac{1}{2} \lambda O_{value}^2 + constant \end{aligned} \quad (1.29)$$

Vậy, XGBoost chỉ thêm cây có output làm tối thiểu hóa được hàm mục tiêu này. Các hằng số sẽ không ảnh hưởng đến bài toán tối thiểu hóa nên ta có thể bỏ chúng đi

$$obj^{(t)} = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + O_{value}) + \frac{1}{2} \lambda O_{value}^2 \quad (1.30)$$

Áp dụng xấp xỉ Taylor bậc 2 (second order Taylor Approximation) [29]:

$$\begin{aligned} L(y_i, \hat{y}_i^{(t-1)} + O_{value}) &\approx L(y_i, \hat{y}_i^{(t-1)}) + \left[\frac{d}{d\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}) \right] O_{value} \\ &\quad + \frac{1}{2} \left[\frac{d^2}{d\hat{y}_i^{(t-1)} d\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}) \right] O_{value}^2 \end{aligned} \quad (1.31)$$

Đặt

$$\begin{aligned} g &= \left[\frac{d}{d\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}) \right] \\ h &= \left[\frac{d^2}{d\hat{y}_i^{(t-1)} d\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}) \right] \end{aligned} \quad (1.32)$$

Khi đó, hàm mục tiêu có thể được viết thành

$$obj^{(t)} = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)}) + \sum_{i=1}^n g_i O_{value} + \frac{1}{2} \sum_{i=1}^n h_i O_{value}^2 + \frac{1}{2} \lambda O_{value}^2 \quad (1.33)$$

Ta thấy được $\sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)})$ không chứa O_{value} , nên nó sẽ không ảnh hưởng đến quá trình tối thiểu hóa và có thể được bỏ đi. Hàm mục tiêu bây giờ là

$$obj^{(t)} = O_{value} \sum_{i=1}^n g_i + \frac{1}{2} O_{value}^2 \left(\sum_{i=1}^n h_i + \lambda \right) \quad (1.34)$$

Để tìm O_{value} làm tối thiểu hàm mục tiêu, ta lấy đạo hàm hàm mục tiêu theo O_{value} và tìm nghiệm bằng 0:

$$\begin{aligned} & \sum_{i=1}^n g_i + O_{value} \left(\sum_{i=1}^n h_i + \lambda \right) = 0 \\ \Leftrightarrow O_{value} &= -\frac{\sum_{i=1}^n g_i}{\sum_{i=1}^n h_i + \lambda} \end{aligned} \quad (1.35)$$

Ở bài toán hồi quy, giả sử ta sử dụng hàm Loss MSE $L(y_i, \hat{y}_i^{(t-1)}) = \frac{1}{2}(y_i - \hat{y}_i^{(t-1)})^2$, ta có thể tính g_i và h_i

$$\begin{aligned} g &= -(y_i - \hat{y}_i^{(t-1)}) \\ h &= 1 \end{aligned} \quad (1.36)$$

thì hàm mục tiêu là:

$$O_{value} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})}{\sum_{i=1}^n 1 + \lambda} \quad (1.37)$$

Ta thấy được ở phân số trên, tử số chính là tổng độ lỗi (sum of residuals) và mẫu số chính là tổng số phần tử trong kết quả (number of residuals) trong output + λ

Vậy hàm mục tiêu của mô hình hồi quy này có thể hiểu là [28]

$$O_{value} = \frac{\text{Sum of Residuals}}{\text{Number of Residuals} + \lambda} \quad (1.38)$$

Đây chính là công thức kết quả của một lá trên cây hồi quy.

Ở bài toán phân lớp, giả sử ta sử dụng hàm Loss

$$L(y_i, \hat{y}_i^{(t-1)}) = -[y_i \log(\hat{y}_i^{(t-1)}) + (1 - y_i) \log(1 - \hat{y}_i^{(t-1)})] \quad (1.39)$$

ta có thể tính g_i và h_i [28]

$$\begin{aligned} g &= -(y_i - \hat{y}_i^{(t-1)}) \\ h &= \hat{y}_i^{(t-1)}(1 - \hat{y}_i^{(t-1)}) \end{aligned} \quad (1.40)$$

thì hàm mục tiêu là:

$$O_{value} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})}{\sum_{i=1}^n (\hat{y}_i^{(t-1)}(1 - \hat{y}_i^{(t-1)})) + \lambda} \quad (1.41)$$

hay dễ hiểu chính là: [28]

$$O_{value} = \frac{\text{Sum of Residuals}}{\Sigma[\text{Previous Probability} \times (1 - \text{Previous Probability})] + \lambda} \quad (1.42)$$

Đây chính là công thức kết quả của một lá trên cây phân lớp.

Tách lá ở cây

Ở XGBoost, khi tách một lá, ta dùng hàm Gain để xác định xem các cây con mới này phân cụm tốt hơn lá ban đầu hay không.

$$\begin{aligned} \text{Gain} &= \text{Similarity Score(Lá trái)} + \text{Similarity Score(Lá phải)} \\ &\quad - \text{Similarity Score(Gốc)} - \gamma \end{aligned} \quad (1.43)$$

và tất nhiên cây con có Gain lớn nhất sẽ được chọn để tách lá.

Công thức của Similarity Score (được miêu tả trong bản thảo XGBoost ban đầu [7]) ở hàm Gain trong XGBoost chính là âm của hàm mục tiêu đã được tối thiểu hóa. Tức là, để tính công thức Similarity Score, ta thế đáp án O_{value} ta tính được ở (1.35) vào (1.34)

$$\begin{aligned} \text{Similarity Score} &= -\sum_{i=1}^n g_i \left(-\frac{\sum_{i=1}^n g_i}{\sum_{i=1}^n h_i + \lambda} \right) - \frac{1}{2} \left(\sum_{i=1}^n h_i + \lambda \right) \left(-\frac{\sum_{i=1}^n g_i}{\sum_{i=1}^n h_i + \lambda} \right)^2 \\ &= \frac{1}{2} \frac{\left(\sum_{i=1}^n g_i \right)^2}{\sum_{i=1}^n h_i + \lambda} \end{aligned} \quad (1.44)$$

Tuy nhiên, do Similarity Score là tương đối giữa các cây con và để giảm số tính toán cần thiết (tối ưu hóa), Similarity Score được áp dụng thực tế được bỏ đi phần $\frac{1}{2}$

$$\text{Similarity Score} = \frac{\left(\sum_{i=1}^n g_i \right)^2}{\sum_{i=1}^n h_i + \lambda} \quad (1.45)$$

Tương tự như cách tính O_{value} cho bài toán hồi quy và phân lớp đã nêu ở trên, ta có thể tính được hàm Similarity Score với các hàm Loss khác nhau. Giả sử bài toán hồi quy và phân lớp sử dụng hàm Loss nêu trên, ta có thể tính hàm Similarity Score:

$$\text{Similarity Score} = \frac{\left(\sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})\right)^2}{\sum_{i=1}^n 1 + \lambda} \quad (1.46)$$

hay dễ hiểu chính là: [28]

$$\text{Similarity Score} = \frac{(\text{Sum of Residuals})^2}{\text{Number of Residuals} + \lambda} \quad (1.47)$$

cho hồi quy, và

$$\text{Similarity Score} = \frac{\left(\sum_{i=1}^n (y_i - \hat{y}_i^{(t-1)})\right)^2}{\sum_{i=1}^n (\hat{y}_i^{(t-1)}(1 - \hat{y}_i^{(t-1)})) + \lambda} \quad (1.48)$$

hay dễ hiểu chính là: [28]

$$\text{Similarity Score} = \frac{(\text{Sum of Residuals})^2}{\Sigma[\text{Previous Probability} \times (1 - \text{Previous Probability})] + \lambda} \quad (1.49)$$

cho phân lớp.

Tỉa cành cây

Quay lại với hàm Gain (1.43) sau khi ta đã tính được công thức của Similarity Score (1.45)

$$\text{Gain} = \frac{\left(\sum_{i=1}^n g_i^{\text{Lá trái}}\right)^2}{\sum_{i=1}^n h_i^{\text{Lá trái}} + \lambda} + \frac{\left(\sum_{i=1}^n g_i^{\text{Lá phải}}\right)^2}{\sum_{i=1}^n h_i^{\text{Lá phải}} + \lambda} - \frac{\left(\sum_{i=1}^n g_i^{\text{Ngọn}}\right)^2}{\sum_{i=1}^n h_i^{\text{Ngọn}} + \lambda} - \gamma \quad (1.50)$$

Khi đã xây dựng xong cây, từ dưới lên trên, những nút đã tách lá nhưng cho Gain nhỏ hơn 0 sẽ bị 'tỉa cành' (tree pruning), gộp lại hai lá mà nút đó đã tách thành trở về một nút ban đầu. Tỉa cành ở đây giúp giảm độ phức tạp của các cây thành phần (regularization), giúp XGBoost giảm overfit

Dựa vào công thức hàm Gain (1.50), ta thấy được hệ số regularization λ và α càng cao thì việc tỉa cành sẽ càng mạnh.

1.3.4 Logistic Classification (Hồi quy Logistic)

Hồi quy Logistic (Logistic Regression) là một mô hình học máy tuyến tính dùng để giải quyết các bài toán **phân loại nhị phân**, trong đó đầu ra là một biến phân loại có hai trạng thái (ví dụ: 0/1, âm tính/dương tính).

Ý tưởng chính

Thay vì dự đoán một giá trị liên tục như hồi quy tuyến tính, Logistic Regression dự đoán xác suất một điểm dữ liệu thuộc về lớp dương (lớp 1), bằng cách sử dụng hàm sigmoid để biến đổi đầu ra thành giá trị trong khoảng $[0, 1]$.

Công thức mô hình

Cho đầu vào $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, và vector trọng số $\mathbf{w} = (w_0, w_1, \dots, w_n)^T$:

- **Hàm dự đoán (sigmoid):**

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{với } z = \mathbf{w}^T \mathbf{x} + b$$

- **Quy tắc phân loại:**

$$\hat{y} \geq 0.5 \Rightarrow \text{lớp 1}; \quad \hat{y} < 0.5 \Rightarrow \text{lớp 0}$$

Hàm mất mát (Binary Cross-Entropy)

Hàm mất mát dùng để huấn luyện mô hình là:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

Trong đó:

- m : số mẫu trong tập huấn luyện
- $y^{(i)}$: nhãn thực tế của mẫu thứ i
- $\hat{y}^{(i)}$: xác suất dự đoán của mẫu thứ i

Huấn luyện mô hình

Trọng số \mathbf{w} được cập nhật qua thuật toán tối ưu như:

- Gradient Descent
- Stochastic Gradient Descent
- Hoặc các trình tối ưu hóa hiện đại (Adam, RMSprop,...)

Ưu điểm

- Mô hình đơn giản, dễ hiểu và dễ triển khai.

- Dự đoán xác suất thay vì chỉ nhãn.
- Hoạt động tốt với dữ liệu tuyến tính.

Nhược điểm

- Không xử lý tốt các quan hệ phi tuyến (non-linear).
- Nhạy cảm với dữ liệu mất cân bằng giữa các lớp.

Ứng dụng

- Phân loại email spam / không spam.
- Dự đoán khả năng khách hàng rời đi (churn prediction).
- Chẩn đoán y tế (ví dụ: bệnh có/không).
- Phân tích rủi ro tài chính (nợ xấu, vỡ nợ).

1.3.5 K-Nearest Neighbors - KNN (Làng giềng gần nhất)

K-Nearest Neighbors (KNN) là một thuật toán học máy không tham số (non-parametric), sử dụng khoảng cách để phân loại hoặc dự đoán một điểm dữ liệu mới dựa trên các điểm gần nhất trong tập huấn luyện.

Nguyên lý hoạt động

1. Tính khoảng cách từ điểm cần phân loại đến tất cả các điểm trong tập huấn luyện.
2. Chọn ra k điểm gần nhất (nearest neighbors).
3. Dự đoán nhãn của điểm mới dựa trên đa số nhãn (phân loại) hoặc trung bình (hồi quy) của k điểm gần nhất.

Công thức tính khoảng cách

Phổ biến nhất là khoảng cách Euclid:

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \sqrt{\sum_{j=1}^n (x_j - x_j^{(i)})^2}$$

Các lựa chọn khác:

-
- Khoảng cách Manhattan:

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \sum_{j=1}^n |x_j - x_j^{(i)}|$$

- Khoảng cách Minkowski (tổng quát):

$$d(\mathbf{x}, \mathbf{x}^{(i)}) = \left(\sum_{j=1}^n |x_j - x_j^{(i)}|^p \right)^{1/p}$$

Quy tắc phân loại

$$\hat{y} = \text{mode} \left(y^{(i_1)}, y^{(i_2)}, \dots, y^{(i_k)} \right)$$

Trong đó:

- \hat{y} : nhãn dự đoán
- $y^{(i)}$: nhãn của điểm lân cận thứ i

Ưu điểm

- Dễ cài đặt và trực quan.
- Không cần huấn luyện mô hình (lazy learning).
- Phù hợp với bài toán phi tuyến.

Nhược điểm

- Tính toán chậm với dữ liệu lớn (phải tính khoảng cách với tất cả điểm huấn luyện).
- Nhạy cảm với nhiễu và thang đo dữ liệu (cần chuẩn hóa).
- Chọn k không phù hợp có thể gây quá khớp hoặc dưới khớp.

Ứng dụng

- Nhận diện chữ viết tay (như MNIST).
- Dự đoán người dùng giống nhau trong hệ thống gợi ý.
- Phân loại bệnh, dữ liệu gen, khách hàng,...

1.3.6 K-Means (Thuật toán phân cụm K trung bình)

K-Means là một thuật toán học không giám sát phổ biến [2], được sử dụng rộng rãi trong các bài toán phân cụm dữ liệu. Mục tiêu của K-Means là chia tập dữ liệu thành **K cụm** sao cho các điểm dữ liệu trong cùng một cụm có độ tương đồng cao với nhau và khác biệt rõ ràng với các cụm còn lại.

Nguyên lý hoạt động

Thuật toán hoạt động theo các bước chính sau:

1. **Khởi tạo:** Chọn ngẫu nhiên K tâm cụm ban đầu (centroids).
2. **Phân cụm:** Gán mỗi điểm dữ liệu vào cụm có tâm gần nhất (theo khoảng cách Euclidean).
3. **Cập nhật:** Tính lại vị trí tâm cụm mới bằng trung bình các điểm trong cụm.
4. **Lặp lại:** Quay lại bước 2 và 3 cho đến khi các tâm cụm hội tụ (không thay đổi đáng kể) hoặc đạt số vòng lặp tối đa.

Công thức khoảng cách Euclidean

Khoảng cách giữa điểm dữ liệu x và tâm cụm c được tính bằng công thức:

$$d(x, c) = \sqrt{\sum_{i=1}^n (x_i - c_i)^2}$$

Trong đó:

- x : điểm dữ liệu.
- c : tâm cụm.
- n : số chiều của không gian dữ liệu.

Tham số quan trọng

- **K:** Số lượng cụm cần phân chia (phải được xác định trước).
- **Max iterations:** Số vòng lặp tối đa để thuật toán hội tụ.
- **Tolerance:** Ngưỡng thay đổi của tâm cụm để quyết định dừng thuật toán.

Ưu điểm

- Dễ hiểu và dễ triển khai.
- Tính toán nhanh, đặc biệt với dữ liệu lớn.

Hạn chế

- Nhạy cảm với vị trí khởi tạo tâm cụm.
- Phải biết trước số cụm K .
- Không hoạt động tốt với dữ liệu có hình dạng phức tạp hoặc phân bố chồng lấn.

Ứng dụng

- Phân đoạn khách hàng (customer segmentation).
- Nhóm ảnh trong xử lý ảnh (image clustering).
- Nhận dạng mẫu (pattern recognition).
- Nén ảnh (image compression).

1.3.7 Spectral Clustering (Phân cụm phổ)

Spectral Clustering (phân cụm phổ) [25] là một thuật toán phân cụm tiên tiến dựa trên lý thuyết đồ thị và đại số tuyến tính. Thay vì phân cụm trực tiếp trên không gian dữ liệu, thuật toán này chuyển dữ liệu sang một không gian mới thông qua việc phân tích các giá trị riêng (eigenvalues) và vector riêng (eigenvectors) của một ma trận biểu diễn quan hệ giữa các điểm dữ liệu.

Nguyên lý hoạt động

Spectral Clustering hoạt động thông qua các bước chính sau:

1. **Xây dựng ma trận kè** (**Adjacency Matrix**): Xác định mức độ liên kết giữa các điểm dữ liệu (thường dùng khoảng cách Gaussian hoặc k-NN).
2. **Tính ma trận Laplacian**: Từ ma trận kè, tính ma trận Laplacian $L = D - A$, trong đó D là ma trận bậc (degree matrix) và A là ma trận kè.
3. **Phân tích giá trị riêng**: Tính các vector riêng tương ứng với k giá trị riêng nhỏ nhất (hoặc lớn nhất tuỳ loại Laplacian).

4. **Ánh xạ không gian mới:** Biểu diễn mỗi điểm dữ liệu theo các vector riêng.
5. **Áp dụng phân cụm K-Means:** Áp dụng thuật toán K-Means trong không gian mới này để phân chia cụm.

Ưu điểm

- Phân cụm tốt cho dữ liệu có hình dạng phức tạp, không lồi.
- Không yêu cầu giả định phân phối dữ liệu.
- Có thể áp dụng cho dữ liệu biểu diễn dưới dạng đồ thị.

Hạn chế

- Hiệu suất kém với dữ liệu lớn (do tính toán ma trận và phân tích trị riêng).
- Cần chọn tham số phù hợp: số cụm k , hàm tương đồng, và tham số Gaussian σ .

Ứng dụng

- Phân tích mạng xã hội.
- Phân đoạn ảnh trong thị giác máy tính.
- Nhận dạng cộng đồng (community detection) trong đồ thị.

1.3.8 Gaussian Mixture Model - GMM (Mô hình hỗn hợp Gauss)

Gaussian Mixture Model (GMM) [2] là một mô hình xác suất dùng để biểu diễn sự phân bố của dữ liệu như là sự kết hợp của nhiều phân phối Gaussian (chuẩn) khác nhau. GMM là một phương pháp phân cụm dựa trên mô hình (model-based clustering), cho phép mô hình hóa các cụm có hình dạng ellipsoid và phân bố chồng lấn.

Nguyên lý hoạt động

GMM giả định rằng dữ liệu được tạo ra từ sự kết hợp của K phân phối Gaussian. Mỗi điểm dữ liệu có xác suất thuộc về mỗi Gaussian khác nhau, thay vì gán cứng vào một cụm như K-Means.

Xác suất tổng hợp của một điểm dữ liệu x là:

$$p(x) = \sum_{k=1}^K \pi_k \cdot \mathcal{N}(x | \mu_k, \Sigma_k)$$

Trong đó:

- π_k : trọng số (mixing coefficient) của thành phần thứ k ($\sum \pi_k = 1$).
- μ_k : vector trung bình của phân phối Gaussian thứ k .
- Σ_k : ma trận hiệp phương sai của phân phối Gaussian thứ k .
- $\mathcal{N}(x | \mu_k, \Sigma_k)$: phân phối chuẩn đa biến.

Thuật toán ước lượng tham số: EM (Expectation-Maximization)

1. **E-step (Expectation):** Tính xác suất mỗi điểm thuộc về từng thành phần Gaussian (trách nhiệm).
2. **M-step (Maximization):** Cập nhật các tham số π_k , μ_k , và Σ_k để cực đại hóa log-likelihood.
3. **Lặp lại** cho đến khi hội tụ.

Ưu điểm

- Mô hình được các cụm phức tạp hơn hình cầu (so với K-Means).
- Gán mềm (soft clustering), mỗi điểm có thể thuộc nhiều cụm với các xác suất khác nhau.

Hạn chế

- Nhạy cảm với khởi tạo ban đầu.
- Dễ bị rơi vào cực trị cục bộ.
- Hiệu suất giảm với dữ liệu có chiều cao.

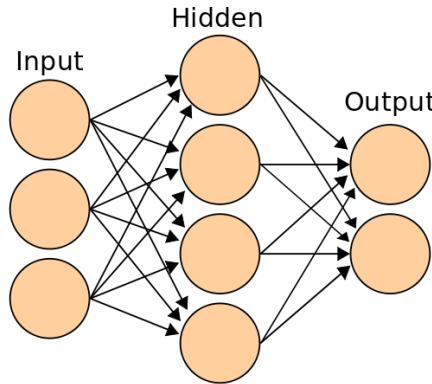
Ứng dụng

- Nhận dạng giọng nói và âm thanh.
- Phân đoạn ảnh (image segmentation).
- Hệ thống khuyến nghị.
- Phát hiện dị thường (anomaly detection).

1.3.9 LSTM (Long Short-Term Memory)

1.3.9.1 Artificial Neural Network (ANN)

Artificial Neural Network hay mạng thần kinh nhân tạo, gọi tắt là mạng thần kinh hoặc mạng neural, là một mô hình xử lý thông tin lấy cảm hứng từ mạng neural sinh học. Kiến trúc của một mạng neural gồm 3 thành phần đó là lớp vào - input layer, lớp ẩn - hidden layer và lớp ra - output layer. Lưu ý, một mạng neural có thể chứa nhiều lớp ẩn ở giữa lớp vào và lớp ra.



Hình 1.8: Minh họa cấu trúc của một mạng neural. [6]

Ta có thể mô tả hoạt động mạng neural bằng công thức

$$y_i^{(l)} = f_i^{(l)} \left(\sum_{j=1}^{n^{(l-1)}} w_{ij}^{(l)} y_j^{(l-1)} + b_i^{(l)} \right). \quad (1.51)$$

Trong đó:

- $l = 1 \dots L$, với L là số lớp của mạng và n^l là số neural ở lớp thứ l .
- $y_i^{(l)}$ là đầu ra của neural thứ i ở lớp thứ l , với $i = 1 \dots n^l$.
- $f_i^{(l)}$ là hàm truyền của neural thứ i ở lớp thứ l . Hàm truyền tồn tại nhiều lựa chọn và mỗi lựa chọn có ảnh hưởng lớn đến kết quả đầu ra của mạng.
- $w_{ij}^{(l)}$ là trọng số của đầu vào thứ j ở lớp thứ l thể hiện độ mạnh của từng đầu vào đối với quá trình xử lý thông tin để chuyển đổi dữ liệu từ lớp này sang lớp khác.
- Đầu vào của mạng là $y^{(0)} = \mathbf{x}$ và đầu ra cuối là $\mathbf{y} = y^{(L)}$.
- $b_i^{(l)}$ là *ngưỡng* (bias) của neural thứ i ở lớp thứ l .

Như vậy, quá trình học của mạng neural là quá trình tìm bộ trọng số \mathbf{w} sao cho phù hợp nhất. Quá trình này sẽ lặp liên tục và có thể không dừng đến khi tìm ra kết

quả như ý. Vì vậy, trong thực tế, khi huấn luyện một mô hình mạng neural, một tiêu chuẩn dựa trên một giá trị sai số nào giữa đầu ra của mạng và đầu ra mong muốn cần được thiết lập, hoặc một số lần lặp tối đa xác định nào đó. Từ đây, ta tiếp cận thuật toán lan truyền ngược (Backpropagation), được sử dụng để điều chỉnh các trọng số liên kết sao cho tổng sai số nhỏ nhất. Với các mạng neural hiện đại, giải thuật được sử dụng kết hợp với một phương pháp tối ưu hóa như gradient descent để rút ngắn thời gian chạy của mạng.

1.3.9.2 Backpropagation

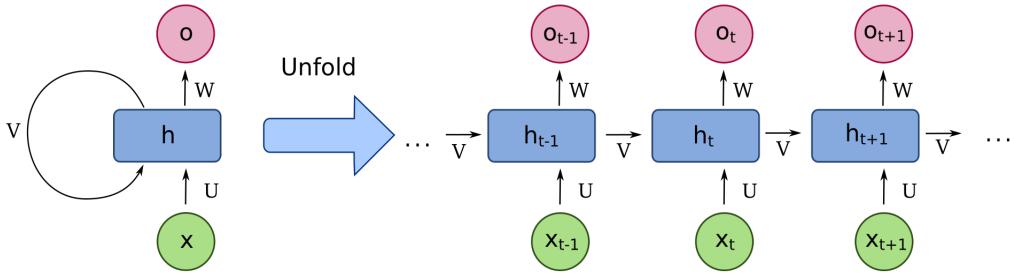
Trước hết, Gradient descent là một thuật toán tìm giá trị nhỏ nhất của hàm số $f(x)$ dựa trên đạo hàm của nó. Thuật toán gồm 3 bước chính: (1) Khởi tạo giá trị $x = x_0$ tùy ý; (2) Tính $x_1 = x_0 - \eta f'(x_0)$, với η là *learning rate* (tạm dịch là tốc độ học); (3) Tính lại $f(x)$ với các giá trị x mới, sao cho nếu $f(x)$ đủ nhỏ thì dừng lại, còn không, lặp lại bước (2) với x_2, x_3, \dots .

Ta có thể nói, thuật toán lan truyền ngược áp dụng Stochastic Gradient Descent (gradient descent ngẫu nhiên - SGD) cho mạng neural có mục tiêu là tìm giá trị nhỏ nhất cho hàm $e(\mathbf{w})$ qua đạo hàm $\nabla e(\mathbf{w}) : \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}}$ với toàn bộ i, j, l . Điểm khác biệt lớn của SGD là bước khởi tạo toàn bộ các trọng số sẽ khởi tạo ngẫu nhiên để tránh *bias*.

1.3.9.3 Recurrent Neural Network (RNN)

Hình 1.8 ở phần trước là một ví dụ cho *mạng neural nhân tạo truyền thẳng* (Feed-forward Neural Network - FNN) gồm hai lớp ẩn kết nối hoàn toàn với nhau ở mỗi nút. Kiến trúc mạng truyền thẳng như trên không có các kết nối ngược trở lại từ các neural đầu ra về các neural đầu vào và mạng không lưu lại các giá trị output trước cũng như các trạng thái kích hoạt của neural.

Một kiểu kiến trúc khác là kiến trúc phản hồi (feedback), hay còn có tên là *mạng neural hồi quy* (Recurrent Neural Network - RNN) cho phép đưa tín hiệu theo cả hai hướng thẳng và ngược lại bằng các vòng lặp. Mạng lưu lại các trạng thái trước đó, và hơn nữa, các trạng thái tiếp theo không chỉ phụ thuộc vào các tín hiệu đầu vào, mà còn phụ thuộc vào các trạng thái trước đó của mạng. Ý tưởng đằng sau RNN là RNN giống như trí nhớ ngắn hạn (Short Term Memory - STM) trong não người, tức là, RNN sẽ "học" và "ghi nhớ" lại thông tin ở chu kỳ trước và áp dụng ở bước "học" tiếp theo.



Hình 1.9: Mô hình mạng neural hồi quy. [16]

Dựa theo hình 1.9, trong đó, x_t và o_t lần lượt là đầu vào và đầu ra tại thời điểm thứ t , h_t là trạng thái ẩn tại thời điểm thứ t và đóng vai trò là bộ nhớ của mạng. h_t sẽ được tính dựa trên sự kết hợp giữa các trạng thái ẩn trước và đầu vào. Ta có thể viết:

$$h_t = f(Ux_t + VS_{t-1}).$$

Sau đó, o_t sẽ được tính dựa theo Wh_t , với (U, V, W) là ba tham số của mạng. Hai loại hàm truyền f phổ biến nhất là hàm tanh hoặc hàm ReLU.

Với khả năng “nhớ” được, mạng neural hồi quy được sử dụng để xử lý thông tin dạng chuỗi và các dữ liệu thời gian. Một số ứng dụng tiêu biểu là: mô hình ngôn ngữ và phát sinh văn bản, dịch máy (Machine Translation) và phát sinh mô tả cho ảnh (Generating Image Descriptions).

1.3.9.4 Hạn chế của RNN

Việc huấn luyện một RNN, cũng tương tự như mạng bình thường, sẽ sử dụng lan truyền ngược - Backpropagation như trên. Tuy nhiên, do bản chất có bộ nhớ của RNN, quá trình lan truyền ngược này sẽ được thực hiện qua từng thời điểm trong mạng. Quá trình này được gọi là *lan truyền ngược liên hồi* (backpropagation through time).

Mục tiêu là tính đạo hàm của lỗi với tham số (U, V, W) tương ứng, sau đó, học các tham số này bằng cách sử dụng gradient descent. Ta có, ở từng trạng thái h , đạo hàm của hàm lỗi được khai triển như sau:

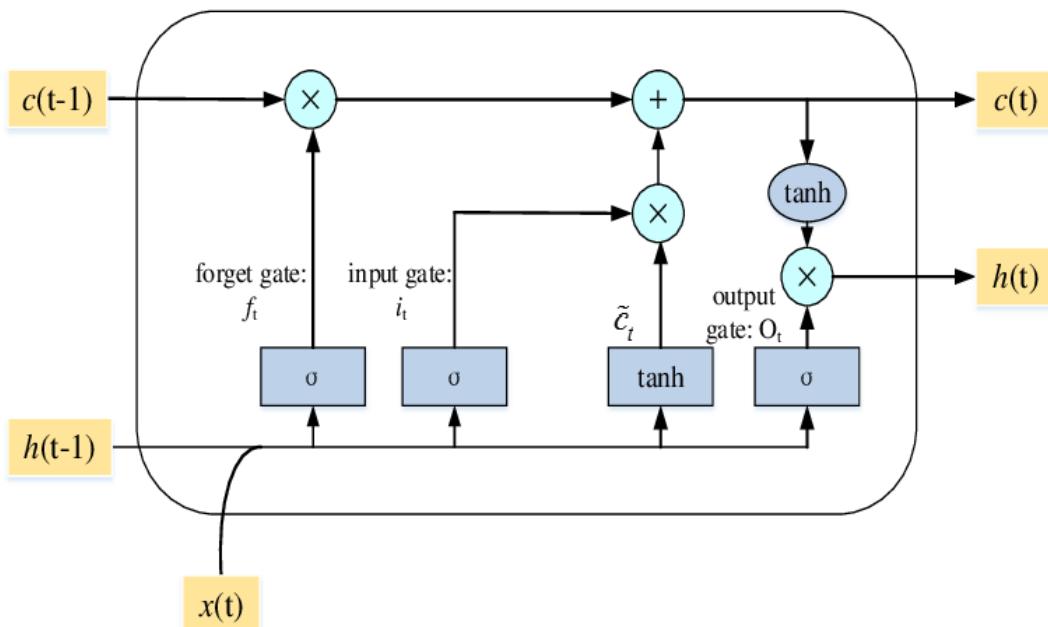
$$\sum_t \frac{\partial e_t(\mathbf{w})}{\partial \mathbf{w}} \propto \sum_t \left(\prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial \mathbf{w}}.$$

Giả sử, trong trường hợp giá trị $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1$, giá trị cập nhật sẽ bị giảm về gần 0 rất nhanh. Ví dụ, nếu lượng cập nhật chỉ là 0.01, nhưng qua 100 thời điểm, thì giá trị trên sẽ là $0.01^{100} \approx 0$. Mà khi đạo hàm bằng 0 thì có nghĩa là xảy ra hiện tượng bão hòa dẫn đến các nút phía trước cũng sẽ bị bão hòa theo. Vấn đề này không chỉ xảy

ra với mạng RNN mà ngay cả mạng neural thường khá sâu cũng có hiện tượng này. Người ta gọi đây là hiện tượng **mất mát đạo hàm** (vanishing gradient).

Một phương pháp phổ biến để giải quyết tình trạng mất mát đạo hàm là sử dụng kiến trúc mạng nhớ dài-ngắn hạn (Long Short-Term Memory (LSTM)).

1.3.9.5 Long Short-Term Memory (LSTM)



Hình 1.10: Cấu trúc của LSTM. [32]

LSTM là một mô hình cải tiến của RNN và có cấu trúc tương tự nhưng khác ở cách tính toán đối với các trạng thái ẩn. Mấu chốt của khả năng “nhớ lâu” của LSTM là cấu trúc “trạng thái nhớ”. Ngoài ra, quá trình thêm hoặc loại bỏ thông tin sẽ dựa trên qui định của các cổng. Tóm lại, cốt lõi của mạng LSTM bao gồm trạng thái nhớ và các cổng.

Các cổng của LSTM là một phương pháp định nghĩa thông tin băng qua và chúng được tạo bằng hàm sigmoid σ . Cụ thể, hàm sigmoid có giá trị thuộc khoảng $(0, 1)$ mang ý nghĩa độ lớn thông tin được phép truyền qua tại mỗi lớp mạng. Nếu kết quả là 0, điều này có nghĩa là không thông tin nào được phép đi qua, và ngược lại, 1 nghĩa là toàn bộ thông tin được đi qua.

Một mạng LSTM có 3 loại cổng, đều có đầu vào là trạng thái trước, đặt tên là $S \in \{h, c\}$, và đầu vào, nhân với trọng số tương ứng. Lưu ý, các cổng sẽ nhận một loại trọng số khác nhau:

- **Cổng quên (Forget gate)** $f_t = \sigma(W_f S_{t-1} + W_f X_t)$.
- **Cổng vào (Input gate)** $i_t = \sigma(W_i S_{t-1} + W_i X_t)$

- **Cổng ra (Output gate)** $o_t = \sigma(W_o S_{t-1} + W_o X_t)$

Sau đó, trạng thái nhớ trung gian có thể được tính bằng hàm tanh qua công thức $\tilde{C}_t = \tanh(W_c S_{t-1} + W_c X_t)$. Từ đây, **trạng thái nhớ** sẽ nhận trạng thái trung gian và trạng thái nhớ trước, kết hợp với cổng vào và cổng quên như sau:

$$c_t = (i_t \times \tilde{C}_t) + (f_t \times c_{t-1}). \quad (1.52)$$

Cuối cùng, trạng thái ẩn sẽ được tính bằng hàm tanh của trạng thái nhớ nhân với cổng ra:

$$h_t = o_t \times \tanh(c_t). \quad (1.53)$$

Lưu ý, phép nhân trong công thức 1.52 và 1.53 phụ thuộc vào S , tức là, các cổng sẽ thực hiện tính cho cả h và c , sau đó, thực hiện phép nhân tương ứng cho mỗi loại trạng thái.

1.4 Giới thiệu các bộ chuẩn hóa dữ liệu

1.4.1 Chuẩn hóa dữ liệu với Min-Max Scaler

Trong học máy, việc chuẩn hóa dữ liệu đầu vào giúp mô hình học hiệu quả hơn, đặc biệt với các mô hình nhạy cảm với thang đo (như KNN, SVM, hồi quy Logistic, mạng nơ-ron,...).

Một trong các phương pháp chuẩn hóa phổ biến là **Min-Max Scaling**, hay còn gọi là *feature normalization*.

Công thức Min-Max Scaling

$$x_i^{\text{scaled}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

Trong đó:

- x_i : giá trị gốc của đặc trưng.
- x_{\min} : giá trị nhỏ nhất của đặc trưng trong tập dữ liệu.
- x_{\max} : giá trị lớn nhất của đặc trưng.
- x_i^{scaled} : giá trị sau khi chuẩn hóa.

Ý nghĩa

Min-Max Scaler biến đổi các đặc trưng về khoảng giá trị $[0, 1]$, giúp:

-
- Tăng tốc độ hội tụ của thuật toán tối ưu.
 - Tránh hiện tượng một vài đặc trưng chi phối quá trình học do thang đo lớn.
 - Dữ liệu có cùng thang đo, phù hợp với các mô hình tính toán khoảng cách.

Ví dụ

Cho tập dữ liệu: $x = [20, 40, 60, 80, 100]$

- $x_{\min} = 20, x_{\max} = 100$
- $x_3 = 60 \Rightarrow x_3^{\text{scaled}} = \frac{60-20}{100-20} = \frac{40}{80} = 0.5$

Lưu ý

- Cần áp dụng giá trị x_{\min}, x_{\max} từ tập huấn luyện cho cả tập kiểm tra.
- Không nên dùng Min-Max Scaling nếu dữ liệu chứa nhiều ngoại lệ (nên dùng Robust Scaler).

1.4.2 Chuẩn hóa dữ liệu với Standard Scaler

Trong học máy, việc chuẩn hóa dữ liệu giúp các mô hình hoạt động hiệu quả hơn, đặc biệt với các thuật toán nhạy cảm với độ lớn của đặc trưng như: hồi quy Logistic, KNN, SVM, mạng nơ-ron,...

Một kỹ thuật chuẩn hóa phổ biến là **Standard Scaler** – biến đổi dữ liệu sao cho có trung bình bằng 0 và độ lệch chuẩn bằng 1.

Công thức chuẩn hóa Standard Scaler

$$x_i^{\text{scaled}} = \frac{x_i - \mu}{\sigma}$$

Trong đó:

- x_i : giá trị gốc của đặc trưng.
- μ : trung bình (mean) của đặc trưng.
- σ : độ lệch chuẩn (standard deviation).
- x_i^{scaled} : giá trị sau khi chuẩn hóa.

Ý nghĩa

- Dữ liệu sau chuẩn hóa có phân phối chuẩn hóa chuẩn: $\mathcal{N}(0, 1)$.
- Trung bình bằng 0 và phương sai bằng 1 giúp mô hình học ổn định hơn.

Ví dụ

Cho tập dữ liệu: $x = [10, 12, 14, 16, 18]$

- Trung bình $\mu = 14$, độ lệch chuẩn $\sigma = \sqrt{\frac{(4^2+2^2+0^2+2^2+4^2)}{5}} = \sqrt{8} \approx 2.828$
- $x_1^{\text{scaled}} = \frac{10-14}{\sigma} \approx \frac{-4}{2.828} \approx -1.414$

Lưu ý

- Nên tính μ và σ trên tập huấn luyện, rồi áp dụng lên cả tập kiểm tra.
- Phù hợp nếu dữ liệu có phân phối gần chuẩn.
- Không phù hợp khi có nhiều ngoại lệ \rightarrow khi đó dùng RobustScaler.

1.4.3 Chuẩn hóa dữ liệu với Robust Scaler

Robust Scaler là một kỹ thuật chuẩn hóa dữ liệu giúp giảm ảnh hưởng của các giá trị ngoại lệ (outliers). Khác với Min-Max Scaler (dựa trên max/min) hay Standard Scaler (dựa trên trung bình/độ lệch chuẩn), Robust Scaler sử dụng các số liệu thống kê mạnh như **trung vị** (median) và **IQR – khoảng tứ phân vị**.

Công thức chuẩn hóa

$$x_i^{\text{scaled}} = \frac{x_i - \text{Median}(x)}{\text{IQR}(x)}$$

Trong đó:

- $\text{Median}(x)$: trung vị của đặc trưng.
- $\text{IQR}(x) = Q_3 - Q_1$: khoảng tứ phân vị (Interquartile Range), với:
 - Q_1 : phân vị thứ 25% (lower quartile)
 - Q_3 : phân vị thứ 75% (upper quartile)
- x_i^{scaled} : giá trị sau chuẩn hóa.

Ưu điểm

-
- Không bị ảnh hưởng mạnh bởi ngoại lệ.
 - Giữ nguyên phân phối cơ bản của dữ liệu phần lớn.
 - Phù hợp với các mô hình yêu cầu dữ liệu chuẩn hóa như KNN, SVM, Logistic Regression.

Ví dụ

Giả sử đặc trưng $x = [10, 12, 14, 16, 100]$

- $\text{Median}(x) = 14$
- $Q_1 = 12, Q_3 = 16 \Rightarrow \text{IQR} = 4$
- $x_5^{\text{scaled}} = \frac{100-14}{4} = 21.5$
- Trong khi Min-Max sẽ nén dữ liệu về $[0,1]$, nhưng sẽ bị kéo lệch vì 100 là ngoại lệ.

Lưu ý

- Nên tính Median và IQR từ tập huấn luyện và áp dụng cho tập kiểm tra.
- Không thích hợp nếu dữ liệu đã phân bố chuẩn và không có ngoại lệ \rightarrow dùng StandardScaler.

1.4.4 Chuẩn hóa dữ liệu với MaxAbsScaler

MaxAbsScaler là một kỹ thuật chuẩn hóa dữ liệu theo phương pháp co dãn tuyến tính (linear scaling), sử dụng giá trị tuyệt đối lớn nhất trong từng đặc trưng để đưa dữ liệu về khoảng $[-1, 1]$, mà vẫn giữ nguyên dấu gốc của dữ liệu.

Công thức chuẩn hóa

$$x_i^{\text{scaled}} = \frac{x_i}{\max(|x|)}$$

Trong đó:

- x_i : giá trị ban đầu của đặc trưng.
- $\max(|x|)$: giá trị tuyệt đối lớn nhất trong đặc trưng đó.
- x_i^{scaled} : giá trị sau chuẩn hóa, nằm trong khoảng $[-1, 1]$.

Đặc điểm

- Dữ liệu được chia cho giá trị tuyệt đối lớn nhất, nên không làm thay đổi sparsity (độ thừa).
- Giữ nguyên dấu ban đầu của dữ liệu.
- Thích hợp với dữ liệu có giá trị cả âm và dương, đặc biệt là dữ liệu sparse (rất nhiều giá trị 0).

Ví dụ

Cho đặc trưng $x = [-3, -1, 0, 2, 4]$:

$$\max(|x|) = 4, \quad x^{\text{scaled}} = \left[\frac{-3}{4}, \frac{-1}{4}, 0, \frac{2}{4}, \frac{4}{4} \right] = [-0.75, -0.25, 0, 0.5, 1.0]$$

Lưu ý

- Phù hợp với dữ liệu sparse (tránh tạo thêm giá trị khác 0).
- Không xử lý outlier – giá trị cực trị vẫn ảnh hưởng đến việc scale.
- Không trung tâm hóa dữ liệu (không đưa trung bình về 0).

1.4.5 Chuẩn hóa dữ liệu với Quantile Transformer

Quantile Transformer là một phương pháp chuẩn hóa dữ liệu phi tuyến, chuyển đổi phân phối gốc của dữ liệu về một phân phối mục tiêu (thường là phân phối chuẩn hoặc phân phối đều) bằng cách sử dụng **phân vị (quantiles)**.

Ý tưởng chính

Quantile Transformer thực hiện hai bước:

1. Ánh xạ mỗi giá trị dữ liệu sang giá trị phân vị tương ứng trong tập huấn luyện.
2. Biến đổi các phân vị đó sang giá trị trong phân phối mục tiêu: *uniform* hoặc *normal*.

Biến đổi phân vị

Giả sử dữ liệu ban đầu là $x \in \mathbb{R}$, ta có:

$$x_i^{\text{scaled}} = F_{\text{target}}^{-1}(F_{\text{empirical}}(x_i))$$

Trong đó:

-
- $F_{\text{empirical}}(x_i)$: phân phối tích lũy thực nghiệm (ECDF) – giá trị phân vị của x_i
 - F_{target}^{-1} : hàm nghịch đảo của phân phối mục tiêu (chuẩn hoặc đều)
 - x_i^{scaled} : giá trị sau khi chuẩn hóa

Phân phối đầu ra

- **Uniform** (phân phối đều): đưa dữ liệu về khoảng $[0, 1]$
- **Normal** (phân phối chuẩn): đưa dữ liệu về phân phối chuẩn $\mathcal{N}(0, 1)$

Ưu điểm

- Biến đổi phân phối lệch về phân phối chuẩn/đều → cải thiện hiệu suất mô hình.
- Giảm tác động của ngoại lệ (outliers).
- Phù hợp với các mô hình tuyến tính hoặc giả định phân phối chuẩn.

Nhược điểm

- Là phép biến đổi phi tuyến → có thể làm mất quan hệ tuyến tính gốc.
- Dễ bị overfit nếu tập huấn luyện nhỏ.

Ví dụ đơn giản

Cho dữ liệu $x = [10, 100, 1000]$

- $10 \rightarrow \text{quantile} = 0.0$
- $100 \rightarrow \text{quantile} = 0.5$
- $1000 \rightarrow \text{quantile} = 1.0$
- Nếu dùng phân phối chuẩn: giá trị tương ứng sẽ là $[-\infty, 0, +\infty]$

1.5 Giới thiệu các chỉ số đánh giá hiệu suất mô hình

1.5.1 Accuracy (Độ chính xác)

Accuracy là độ đơn giản nhất dùng để đánh giá hiệu suất của mô hình phân loại, được tính bằng tỷ lệ giữa số lượng dự đoán đúng và tổng số dự đoán:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.54)$$

Ví dụ với bài toán phân loại *Cat/Non-cat*, giả sử có 90 ảnh *Cat* được phân loại đúng, 10 ảnh *Cat* bị phân loại sai, 940 ảnh *Non-cat* phân loại đúng và 60 ảnh *Non-cat* bị phân loại sai. Khi đó:

$$\text{Accuracy} = \frac{90 + 940}{1000 + 100} = 93.6\% \quad (1.55)$$

Tuy nhiên, Accuracy không phản ánh cụ thể mô hình xử lý tốt nhóm nào, dễ gây hiểu nhầm nếu dữ liệu mất cân bằng.

1.5.2 F1-score

Khi cả Precision và Recall đều quan trọng, ta sử dụng F1-score, là trung bình điều hòa của Precision và Recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1.56)$$

F1-score giúp cân bằng giữa Precision và Recall, đặc biệt hữu ích khi dữ liệu bị mất cân bằng.

1.5.3 Precision

Precision đo lường tỷ lệ dự đoán dương tính đúng trong tất cả các dự đoán dương tính:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1.57)$$

Áp dụng cho bài toán *Cat/Non-cat*:

$$\begin{aligned} \text{Precision}_{\text{Cat}} &= \frac{90}{90 + 60} = 60\% \\ \text{Precision}_{\text{Non-cat}} &= \frac{940}{940 + 10} = 98.9\% \end{aligned}$$

Precision giúp đánh giá mô hình có dự đoán nhầm quá nhiều hay không trong nhóm dương tính.

1.5.4 Recall

Recall (Sensitivity, True Positive Rate) là tỷ lệ phát hiện đúng các mẫu dương tính thực sự:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (1.58)$$

Ví dụ:

$$\text{Recall}_{\text{Cat}} = \frac{90}{90 + 10} = 90\%$$
$$\text{Recall}_{\text{Non-cat}} = \frac{940}{940 + 60} = 94\%$$

Recall cao thể hiện mô hình ít bỏ sót mẫu dương tính thực sự.

1.5.5 ROC AUC

ROC Curve (Receiver Operating Characteristic Curve) là biểu đồ thể hiện mối quan hệ giữa TPR và FPR:

$$\text{TPR} = \frac{TP}{TP + FN} \quad (\text{Recall}) \quad (1.59)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (1.60)$$

AUC (Area Under the Curve) là diện tích dưới đường cong ROC, biểu thị tổng quát khả năng phân loại của mô hình. AUC có giá trị từ 0 đến 1:

- AUC gần 1: mô hình phân loại tốt.
- AUC gần 0.5: mô hình phân loại ngẫu nhiên.

Ví dụ: Với đầu ra xác suất $[0.45, 0.6, 0.7, 0.3]$, ta thay đổi ngưỡng để xác định nhãn phân loại. Khi vẽ TPR và FPR ứng với nhiều ngưỡng, ta có thể vẽ được ROC Curve. Diện tích dưới đường cong chính là AUC.

Giải thích chi tiết các tham số

Các độ đo như Accuracy, Precision, Recall, F1-score,... đều dựa trên **Ma trận nhầm lẫn (Confusion Matrix)**. Ma trận này được xây dựng dựa trên so sánh giữa nhãn dự đoán của mô hình và nhãn thực tế. Đối với bài toán phân loại nhị phân (binary classification), ma trận nhầm lẫn có 4 phần tử chính:

	Dự đoán: Positive	Dự đoán: Negative
Thực tế: Positive	TP (True Positive)	FN (False Negative)
Thực tế: Negative	FP (False Positive)	TN (True Negative)

- **TP (True Positive)**: Số mẫu dương tính được mô hình dự đoán đúng là dương tính.
- **TN (True Negative)**: Số mẫu âm tính được mô hình dự đoán đúng là âm tính.
- **FP (False Positive)**: Số mẫu âm tính bị mô hình dự đoán nhầm là dương tính (hay còn gọi là lỗi loại I).
- **FN (False Negative)**: Số mẫu dương tính bị mô hình dự đoán nhầm là âm tính (hay còn gọi là lỗi loại II).

1.5.6 Mean Squared Error (MAE)

MSE (Mean Squared Error) là một trong những độ đo phổ biến nhất dùng để đánh giá hiệu suất của các mô hình hồi quy. Nó đo lường sai số trung bình bình phương giữa giá trị dự đoán và giá trị thực tế.

Giả sử với một bộ dữ liệu gồm n mẫu, giá trị thực tế được ký hiệu là y_i , giá trị dự đoán là \hat{y}_i , thì MSE được tính theo công thức:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1.61)$$

MSE phạt nặng các sai số lớn do sai số được bình phương. Do đó, MSE nhạy cảm với các điểm ngoại lai (*outliers*).

Ví dụ: Trong bài toán dự đoán giá nhà, y_i là giá trị thực tế của căn nhà thứ i , và \hat{y}_i là giá trị dự đoán. MSE sẽ đo lường khoảng cách trung bình bình phương giữa các giá trị này.

1.5.7 Mean Absolute Error (MAE)

MAE (Mean Absolute Error) là một độ đo khác, đánh giá sai số trung bình tuyệt đối giữa giá trị thực tế và giá trị dự đoán:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1.62)$$

Không giống như MSE, MAE sử dụng trị tuyệt đối thay vì bình phương, do đó nó ít bị ảnh hưởng bởi các điểm ngoại lai. MAE được coi là một metric *mạnh hơn* khi mô hình cần khả năng chống chịu với dữ liệu nhiễu.

So sánh MSE và MAE:

- **MSE** nhẫn mạnh các lỗi lớn (do bình phương sai số), phù hợp khi muốn phạt mạnh các dự đoán sai nhiều.
- **MAE** phản ánh mức sai lệch trung bình thực sự, bền vững hơn với các ngoại lai.

1.5.8 Root Mean Squared Error (RMSE)

RMSE (Root Mean Squared Error) là một metric đánh giá độ lệch giữa giá trị dự đoán và giá trị thực tế trong các bài toán hồi quy. RMSE chính là căn bậc hai của MSE (Mean Squared Error), do đó vẫn giữ đặc tính "phạt nặng" các sai số lớn, nhưng giá trị của nó có cùng đơn vị với biến đầu ra.

Công thức tính RMSE

Giả sử có n mẫu dữ liệu, y_i là giá trị thực tế, và \hat{y}_i là giá trị dự đoán của mô hình. Khi đó, RMSE được tính như sau:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1.63)$$

Ý nghĩa

- RMSE biểu thị độ lệch chuẩn của sai số dự đoán.
- Đơn vị của RMSE giống với đơn vị của biến đầu ra, giúp dễ diễn giải trong thực tế.
- RMSE nhạy cảm với các điểm ngoại lai (outliers) hơn so với MAE, do bình phương sai số.

So sánh RMSE và MSE

- **MSE**: thường được dùng cho mục đích tối ưu trong huấn luyện mô hình (ví dụ trong đạo hàm của hàm mất mát).
- **RMSE**: thường được dùng để báo cáo kết quả mô hình vì đơn vị dễ hiểu hơn.

Ví dụ

Nếu mô hình dự đoán giá nhà, trong đó giá trị thực tế và dự đoán tính bằng triệu đồng, thì RMSE = 50 nghĩa là sai số trung bình của mô hình là khoảng **50 triệu đồng**.

1.5.9 Mean Absolute Percentage Error (MAPE)

MAPE (Mean Absolute Percentage Error) là một độ đo thường được sử dụng trong các bài toán hồi quy để đánh giá sai số trung bình phần trăm giữa giá trị dự đoán và giá trị thực tế.

Công thức tính

Giả sử ta có n mẫu dữ liệu, với y_i là giá trị thực tế và \hat{y}_i là giá trị dự đoán tương ứng. Công thức tính MAPE như sau:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (1.64)$$

Ý nghĩa

- MAPE biểu thị sai số trung bình theo phần trăm so với giá trị thực tế.
- MAPE dễ giải và thường được sử dụng trong các bài toán dự báo thời gian (*forecasting*).
- Ví dụ: MAPE = 8.5% nghĩa là trung bình mô hình dự đoán sai lệch 8.5% so với giá trị thực tế.

Ưu và nhược điểm

- **Ưu điểm:**
 - Dễ giải trực quan dưới dạng phần trăm.
 - Không phụ thuộc vào đơn vị đo.
- **Nhược điểm:**
 - Không xác định được nếu có $y_i = 0$ (chia cho 0).
 - Nhạy cảm với các giá trị thực nhỏ, dễ làm tăng sai số phần trăm.

Lưu ý khi sử dụng

MAPE không phù hợp nếu tập dữ liệu chứa các giá trị thực tế bằng 0 hoặc gần 0. Trong các trường hợp này, nên dùng SMAPE (Symmetric MAPE) hoặc các metric khác như MAE, RMSE.

1.5.10 Adjusted Rand Index (ARI)

Khái niệm

Adjusted Rand Index (ARI) là một độ đo thống kê dùng để đánh giá mức độ tương đồng giữa hai tập phân cụm: một là kết quả phân cụm của mô hình, và một là nhãn thực (ground truth), nếu có. ARI điều chỉnh theo sự ngẫu nhiên, giúp khắc phục nhược điểm của chỉ số Rand gốc (Rand Index - RI), vốn có thể cho giá trị cao ngay cả khi hai phân cụm gần như ngẫu nhiên.

Chỉ số ARI được sử dụng phổ biến trong các bài toán phân cụm khi muốn so sánh mức độ chính xác của mô hình phân cụm không giám sát với phân nhóm thực tế.

Công thức tính ARI

Giả sử ta có:

- n : Tổng số điểm dữ liệu.
- $X = \{X_1, X_2, \dots, X_r\}$: Tập phân cụm thực tế gồm r cụm.
- $Y = \{Y_1, Y_2, \dots, Y_s\}$: Tập phân cụm dự đoán gồm s cụm.

Gọi n_{ij} là số phần tử nằm đồng thời trong cụm X_i và Y_j .

Gọi:

$$a_i = \sum_j n_{ij} \quad (\text{tổng số điểm trong cụm thực tế } X_i)$$
$$b_j = \sum_i n_{ij} \quad (\text{tổng số điểm trong cụm dự đoán } Y_j)$$

Khi đó, công thức ARI được tính như sau:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{n}{2}]}{\frac{1}{2} [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{n}{2}]} \quad (1.65)$$

Giải thích ý nghĩa

- ARI có giá trị nằm trong khoảng $[-1, 1]$.
- ARI = 1 khi hai phân cụm hoàn toàn khớp nhau.
- ARI ≈ 0 khi phân cụm gần như ngẫu nhiên.
- ARI < 0 có thể xảy ra nếu phân cụm kém hơn cả ngẫu nhiên.

Ví dụ sử dụng

Chẳng hạn trong bài toán phân cụm ảnh khuôn mặt hoặc nhóm khách hàng, ARI cho phép đánh giá mô hình phân cụm khi có nhãn thật để so sánh. Đây là công cụ hữu ích khi so sánh các thuật toán phân cụm như K-Means, DBSCAN, Agglomerative Clustering, v.v.

CHƯƠNG

2

PHÂN TÍCH DỮ LIỆU

2.1 Dữ liệu dạng bảng

2.1.1 Predict Student's Dropout and Academic Success

2.1.1.1 Giới thiệu bộ dữ liệu

2.1.1.1.1 Nguồn dữ liệu

Bộ dữ liệu được thu thập bởi Martins và cộng sự [18], hỗ trợ bởi chương trình SATDAP - Capacitação da Administração Pública POCI-05-5762-FSE-000191, Bồ Đào Nha.

2.1.1.1.2 Mô tả dữ liệu

Đây là bộ dữ liệu thu thập từ nhiều cơ sở giáo dục đại học hoặc tương đương, từ các ngành nông học, thiết kế, giáo dục, điều dưỡng, báo chí, quản lý, dịch vụ xã hội và công nghệ. Mỗi dòng dữ liệu bao gồm các thông tin yếu tố kinh tế xã hội tại thời điểm vào thời điểm sinh viên nhập học, các thông tin về cá nhân sinh viên, kèm kết quả học tập của sinh viên vào cuối học kỳ 1 và học kỳ 2 (hai học kỳ đầu tiên). Bộ dữ liệu bị thiếu dữ liệu ở bất kỳ cột nào.

Ví dụ một phần dữ liệu:

Bảng 2.1: Một phần bảng dữ liệu student dropout

Tuition fees up to date	Scholarship holder	Curricular units 1st sem (approved)	Curricular units 1st sem (grade)	...
1	0	0	0.000000	...
0	0	6	14.000000	...
0	0	0	0.000000	...
1	0	6	13.428571	...
1	0	5	12.333333	...
1	0	5	11.857143	...
...

Bộ dữ liệu có 4424 dòng, bao gồm 37 cột như sau:

- Kiểu dữ liệu: **Quantitative**

1. **Marital Status:** Tình trạng hôn nhân

Các giá trị:

- **1:** Độc thân
- **2:** Kết hôn
- **3:** Góa chồng/vợ
- **4:** Ly dị
- **5:** Sống chung (không pháp lý)
- **6:** Ly thân

2. **Application mode:** Trạng thái xét tuyển

Các giá trị:

- **1:** Giai đoạn 1 – Điện tuyển sinh chung
- **2:** Theo Quy định số 612/93
- **5:** Giai đoạn 1 – điện đặc biệt (Quần đảo Azores)
- **7:** Người đã có văn bằng giáo dục đại học khác
- **10:** Theo Quy định số 854-B/99
- **15:** Sinh viên quốc tế (bậc cử nhân)
- **16:** Giai đoạn 1 – điện đặc biệt (Quần đảo Madeira)
- **17:** Giai đoạn 2 – điện tuyển sinh chung
- **18:** Giai đoạn 3 – điện tuyển sinh chung
- **26:** Theo Quy định số 533-A/99, mục b2 (Chương trình khác)
- **27:** Theo Quy định số 533-A/99, mục b3 (Trường khác)
- **39:** Trên 23 tuổi
- **42:** Chuyển trường

-
- **43**: Chuyển ngành học
 - **44**: Người có văn bằng chuyên môn công nghệ
 - **51**: Chuyển trường/ngành học
 - **53**: Người có văn bằng chương trình ngắn hạn
 - **57**: Chuyển trường/ngành học (Sinh viên quốc tế)

3. **Course:** Mã ngành

Các giá trị:

- **33**: Công nghệ sản xuất nhiên liệu sinh học
- **171**: Thiết kế hoạt hình và đa phương tiện
- **8014**: Công tác xã hội (hệ học buổi tối)
- **9003**: Nông học
- **9070**: Thiết kế truyền thông
- **9085**: Điều dưỡng thú y
- **9119**: Kỹ thuật tin học
- **9130**: Kỹ thuật nuôi ngựa
- **9147**: Quản trị
- **9238**: Công tác xã hội
- **9254**: Du lịch
- **9500**: Điều dưỡng
- **9556**: Vệ sinh răng miệng
- **9670**: Quản trị quảng cáo và tiếp thị
- **9773**: Báo chí và truyền thông
- **9853**: Giáo dục tiểu học
- **9991**: Quản trị (hệ học buổi tối)

4. **Daytime/evening attendance:** Buổi học

Các giá trị:

- **0**: Tối
- **1**: Sáng

5. **Previous qualification:** Trình độ khi xét tuyển

Các giá trị:

- **1**: Giáo dục trung học phổ thông
- **2**: Giáo dục đại học – bằng cử nhân
- **3**: Giáo dục đại học – bằng không rõ
- **4**: Giáo dục đại học – bằng thạc sĩ
- **5**: Giáo dục đại học – bằng tiến sĩ
- **6**: Đang theo học đại học
- **9**: Chưa hoàn thành Lớp 12
- **10**: Chưa hoàn thành Lớp 11
- **12**: Khác – trình độ tương đương lớp 11

- **14:** Lớp 10
- **15:** Chưa hoàn thành Lớp 10
- **19:** Lớp 9/10/11 hoặc tương đương
- **38:** Lớp 6/7/8 hoặc tương đương
- **39:** Đã học khóa chuyên môn công nghệ
- **40:** Trình độ cử nhân (EU)
- **42:** Đã học khóa kỹ thuật cao đẳng chuyên nghiệp
- **43:** Trình độ thạc sĩ (EU)

6. **Nationality:** Quốc tịch

Các giá trị:

- **1:** Bồ Đào Nha
- **2:** Đức
- **6:** Tây Ban Nha
- **11:** Ý
- **13:** Hà Lan
- **14:** Anh
- **17:** Litva
- **21:** Angola
- **22:** Cabo Verde
- **24:** Guinée-Bissau
- **25:** Mozambique
- **26:** Santomean
- **32:** Thổ Nhĩ Kỳ
- **41:** Brazil
- **62:** Romania
- **100:** Moldova
- **101:** Mexico
- **103:** Ukraina
- **105:** Nga
- **108:** Cuba
- **109:** Colombia

7. **Mother's qualification:** Trình độ học vấn mẹ

Các giá trị:

- **1:** Trung học phổ thông – lớp 12 hoặc tương đương
- **2:** Giáo dục đại học – Cử nhân
- **3:** Giáo dục đại học – Bằng tốt nghiệp đại học
- **4:** Giáo dục đại học – Thạc sĩ
- **5:** Giáo dục đại học – Tiến sĩ
- **6:** Đang theo học giáo dục đại học

-
- **9:** Lớp 12 – chưa hoàn thành
 - **10:** Lớp 11 – chưa hoàn thành
 - **11:** Lớp 7 (theo hệ thống cũ)
 - **12:** Khác – trình độ tương đương lớp 11
 - **13:** Năm 2 bổ túc trung học phổ thông
 - **14:** Lớp 10
 - **18:** Đã học khóa thương mại tổng quát
 - **19:** Lớp 9/10/11 hoặc tương đương
 - **20:** Đã học khóa bổ túc trung học phổ thông
 - **22:** Đã học khóa kỹ thuật – nghề nghiệp
 - **25:** Khóa bổ túc trung học phổ thông - chưa hoàn thành
 - **26:** Lớp 7
 - **27:** Lớp 6/7/8 hoặc tương đương
 - **29:** Lớp 9 – chưa hoàn thành
 - **30:** Lớp 8
 - **31:** Đã học khóa Quản trị và Thương mại
 - **33:** Đã học khóa bổ túc Kế toán và Quản trị
 - **34:** Không rõ
 - **35:** Không biết đọc hoặc viết
 - **36:** Biết đọc nhưng chưa học đến lớp 4
 - **37:** Lớp 4/5 hoặc tương đương
 - **38:** Lớp 6/7/8 hoặc tương đương
 - **39:** Đã học khóa chuyên môn công nghệ
 - **40:** Trình độ cử nhân (chương trình chuẩn EU)
 - **41:** Đã học khóa chuyên ngành cao học
 - **42:** Đã học khóa cao đẳng kỹ thuật chuyên nghiệp
 - **43:** Thạc sĩ (chương trình chuẩn EU)
 - **44:** Tiến sĩ (chương trình chuẩn EU)

8. **Father's qualification:** Trình độ học vấn cha

Các giá trị: (tương tự **Mother's qualification**)

9. **Mother's occupation:** Nghề nghiệp mẹ

Các giá trị:

- **0:** Đang trong quá trình học tập
- **1:** Đại biểu cơ quan lập pháp, cơ quan hành pháp, giám đốc hoặc quản lý cấp cao
- **2:** Chuyên gia trong các lĩnh vực trí tuệ và khoa học
- **3:** Kỹ thuật viên và chuyên viên trình độ trung cấp
- **4:** Nhân viên hành chính
- **5:** Nhân viên dịch vụ cá nhân, an ninh và bán hàng

- **6:** Nông dân và lao động lành nghề trong nông nghiệp, ngư nghiệp, lâm nghiệp
- **7:** Lao động lành nghề trong công nghiệp, xây dựng và thủ công
- **8:** Công nhân lắp đặt, vận hành máy móc và lắp ráp
- **9:** Lao động phổ thông
- **10:** Quân nhân
- **90:** Khác
- **99:** Không rõ
- **101:** Sĩ quan quân đội
- **102:** Hạ sĩ quan quân đội
- **103:** Nhân sự lực lượng vũ trang
- **112:** Giám đốc dịch vụ hành chính và thương mại
- **114:** Giám đốc khách sạn, ăn uống, thương mại và các dịch vụ tương tự khác
- **121:** Chuyên gia khoa học tự nhiên, toán học, kỹ thuật và các lĩnh vực liên quan
- **122:** Nhân viên y tế
- **123:** Giáo viên
- **124:** Chuyên gia tài chính, kế toán, tổ chức hành chính, quan hệ công – thương
- **125:** Chuyên gia công nghệ thông tin và truyền thông
- **131:** Kỹ thuật viên trung cấp trong khoa học và kỹ thuật
- **132:** Kỹ thuật viên trung cấp trong lĩnh vực y tế
- **134:** Kỹ thuật viên trung cấp trong lĩnh vực pháp luật, xã hội, thể thao, văn hóa
- **135:** Kỹ thuật viên công nghệ thông tin và truyền thông
- **141:** Nhân viên văn phòng, thư ký, và nhân viên xử lý dữ liệu
- **143:** Nhân viên dữ liệu, kế toán, thống kê, tài chính và lưu trữ
- **144:** Nhân viên hỗ trợ hành chính
- **151:** Giúp việc
- **152:** Nhân viên bán hàng
- **153:** Nhân viên chăm sóc cá nhân và tương tự
- **154:** Nhân viên bảo vệ và dịch vụ an ninh
- **161:** Nông dân định hướng thị trường và lao động lành nghề trong nông nghiệp, chăn nuôi
- **163:** Nông dân tự cung tự cấp, người chăn nuôi, ngư dân, thợ săn và hái lượm
- **171:** Lao động lành nghề trong xây dựng (trừ thợ điện)
- **172:** Lao động lành nghề trong luyện kim, gia công kim loại và tương tự
- **173:** Lao động lành nghề trong in ấn, sản xuất thiết bị chính xác, kim hoàn, thủ công

-
- **174:** Lao động lành nghề trong điện và điện tử
 - **175:** Công nhân chế biến thực phẩm, gỗ, may mặc và các ngành thủ công khác
 - **181:** Công nhân vận hành thiết bị và máy móc cố định
 - **182:** Công nhân dây chuyền
 - **183:** Lái xe/Công nhân vận hành thiết bị di động
 - **191:** Nhân viên vệ sinh
 - **192:** Lao động phổ thông trong nông nghiệp, chăn nuôi, thủy sản và lâm nghiệp
 - **193:** Lao động phổ thông trong khai khoáng, xây dựng, sản xuất và vận tải
 - **194:** Nhân viên phụ bếp
 - **195:** Bán hàng rong hoặc cung cấp dịch vụ đường phố (trừ liên quan đến thực phẩm)
10. **Father's occupation:** Nghề nghiệp cha
Các giá trị: (tương tự **Mother's occupation**)
11. **Displaced:** Người tị nạn
Các giá trị:
 - **0:** Không
 - **1:** Đúng
12. **Educational special needs:** Cần giáo dục đặc biệt
Các giá trị:
 - **0:** Không
 - **1:** Có
13. **Debtor:** Mắc nợ
Các giá trị:
 - **0:** Không
 - **1:** Có
14. **Tuition fees up to date:** Đóng học phí đầy đủ
Các giá trị:
 - **0:** Không
 - **1:** Đúng
15. **Gender:** Giới tính
Các giá trị:
 - **0:** Nữ
 - **1:** Nam
16. **Scholarship holder:** Học bổng
Các giá trị:
 - **0:** Không

- **1:** Có
17. **International:** Học sinh quốc tế
Các giá trị:
 - **0:** Không
 - **1:** Đúng
18. **Target:** Tình trạng sinh viên
Các giá trị:
 - **Graduate:** Đã tốt nghiệp
 - **Enrolled:** Đăng ký học
 - **Dropout:** Đã bỏ học
- Kiểu dữ liệu: **Qualitative**
 - **Discrete:**
 - 19. **Application order:** Xét tuyển vào trường là nguyện vọng thứ mấy (0 - nguyện vọng 1, đếm lên)
 - 20. **Age at enrollment:** Tuổi nhập học
 - 21. **Curricular units 1st sem (credited):** Số học phần được miễn học kỳ 1
 - 22. **Curricular units 1st sem (enrolled):** Số học phần đăng ký học kỳ 1
 - 23. **Curricular units 1st sem (evaluations):** Số học phần được đánh giá học kỳ 1
 - 24. **Curricular units 1st sem (approved):** Số học phần qua môn trong học kỳ 1
 - 25. **Curricular units 1st sem (without evaluations):** Số học phần không đánh giá học kỳ 1
 - 26. **Curricular units 2nd sem (credited):** Số học phần được miễn học kỳ 2
 - 27. **Curricular units 2nd sem (enrolled):** Số học phần đăng ký học kỳ 2
 - 28. **Curricular units 2nd sem (evaluations):** Số học phần được đánh giá học kỳ 2
 - 29. **Curricular units 2nd sem (approved):** Số học phần qua môn trong học kỳ 2
 - 30. **Curricular units 2nd sem (without evaluations):** Số học phần không đánh giá học kỳ 2
 - **Continuous:**
 - 31. **Previous qualification (grade):** Điểm bằng cấp trước đó
 - 32. **Admission grade:** Điểm đầu vào
 - 33. **Curricular units 1st sem (grade):** Điểm trung bình học kỳ 1 (trong khoảng 0-20*)
 - 34. **Curricular units 2nd sem (grade):** Điểm trung bình học kỳ 2 (trong khoảng 0-20)
 - 35. **Unemployment rate:** Tỉ lệ thất nghiệp khi người đó nhập học
 - 36. **Inflation rate:** Tỉ lệ lạm phát khi người đó nhập học
 - 37. **GDP:** GDP khi người đó nhập học

2.1.1.2 Các nghiên cứu liên quan

Tác giả của bài báo [19] sử dụng các mô hình học máy (Machine Learning) để dự đoán tình trạng sinh viên tốt nghiệp đúng hạn. Nghiên cứu này nhằm lựa chọn mô hình học máy tối ưu để dự đoán tình trạng sinh viên tốt nghiệp đúng hạn, đồng thời xác định các thuộc tính thông tin có tác động mạnh đến khả năng tốt nghiệp đúng hạn hoặc quá hạn của sinh viên, từ đó đề xuất khuyến nghị giúp nhà trường nâng cao tỷ lệ tốt nghiệp. Để thực hiện mục tiêu này, bài báo sử dụng tập dữ liệu của 6.696 sinh viên chuyên ngành Ngân hàng thuộc Học viện Ngân hàng Hà Nội từ năm 2010 đến 2020. Dữ liệu đã được tiền xử lý bằng cách làm sạch, chuyển đổi, mã hóa, chia thành tập huấn luyện và kiểm tra theo tỷ lệ 80:20, và đặc biệt là áp dụng kỹ thuật SMOTE để xử lý tình trạng mất cân bằng dữ liệu (86,96% "Đúng hạn" và 13,04% "Quá hạn"). Nghiên cứu đã thử nghiệm 7 mô hình học máy: Logistic Regression (LR), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), XGBoost và CatBoost, và đánh giá hiệu suất của chúng dựa trên các chỉ số như Precision, Recall, Accuracy, và đặc biệt là F1 Score, cùng với kỹ thuật LIME để giải thích sự đóng góp của các thuộc tính. Kết quả nghiên cứu cho thấy Random Forest (RF) là mô hình tối ưu nhất, đạt độ chính xác 92% và điểm F1 là 92%, vượt trội so với các mô hình khác (XGBoost và CatBoost đạt 89% F1, LR và SVM đạt 80% F1).

Nghiên cứu tiếp theo này [23] tập trung vào việc dự đoán chất lượng không khí ở Jakarta, cụ thể là nồng độ PM2.5, nhằm phát triển một mô hình dự đoán chính xác và có khả năng giải thích được thông qua việc sử dụng khuôn khổ XGBoost-SHAP. Mục tiêu chính là không chỉ dự đoán nồng độ PM2.5 một cách đáng tin cậy mà còn cung cấp những hiểu biết sâu sắc về các đặc điểm ảnh hưởng đáng kể nhất đến các dự đoán đó, từ đó cung cấp thông tin thực tế cho cộng đồng và cơ sở dữ liệu cho các nhà hoạch định chính sách để xây dựng các chiến lược giảm thiểu ô nhiễm. Để đạt được điều này, nghiên cứu đã sử dụng dữ liệu thứ cấp về Chỉ số Tiêu chuẩn Ô nhiễm Không khí (ISPU) của tỉnh DKI Jakarta từ tháng 1 năm 2021 đến tháng 5 năm 2024, bao gồm 4774 điểm dữ liệu hàng ngày từ 5 trạm quan trắc. Dữ liệu này bao gồm các thuộc tính như PM10, PM2.5, SO2, CO, O3, NO2, giá trị "max" (giá trị cao nhất của các tham số được theo dõi), "critical" (tham số có giá trị ISPU cao nhất), và "category" (loại chất lượng không khí). Quá trình tiền xử lý dữ liệu bao gồm xử lý các giá trị thiếu (41 hàng thiếu trong cột "critical" được điền bằng mode, các giá trị '0' trong các cột số được xử lý bằng KNN Imputer với k=5), mã hóa các biến phân loại sử dụng Label Encoder, và xử lý dữ liệu ngoại lai bằng phương pháp Winsorizing. Mô hình học máy chính được áp dụng là XGBoost, được lựa chọn vì khả năng xử lý dữ liệu phức tạp và ngăn ngừa overfitting. Điều chỉnh siêu tham số (hyperparameter tuning) cho XGBoost được thực hiện bằng Grid Search với cơ chế kiểm định chéo 10 lần (10-fold cross-validation). Dữ liệu được chia thành tập huấn luyện và kiểm tra với ba tỷ lệ (90:10, 80:20, 70:30), và hiệu suất mô hình được đánh giá bằng MSE, RMSE, R2, và đặc biệt là MAPE (Mean Absolute Percentage Error), với MAPE dưới 10% được coi là "chính xác". Kết quả cho thấy mô hình XGBoost đạt hiệu suất vượt trội, với MAPE thấp nhất là 4.44% khi sử dụng tỷ lệ dữ liệu kiểm thử 10%, giá trị này dưới 10% cho thấy khả năng dự đoán chính xác. So với các nghiên cứu trước đây về dự đoán PM2.5, mô hình XGBoost này cho thấy hiệu suất tốt hơn so với Hybrid ARIMA-ANN (MAPE 13.6%), LSTM

(MAPE 8.07%), và Ordinary Kriging (MAPE 19.5%). Phân tích SHAP đã cung cấp những hiểu biết quan trọng, chỉ ra rằng đặc điểm "max" có ảnh hưởng lớn nhất đến dự đoán, tiếp theo là PM10 (có xu hướng làm tăng nồng độ PM2.5). Nhìn chung, nghiên cứu kết luận rằng mô hình XGBoost có thể được sử dụng hiệu quả để dự đoán nồng độ PM2.5 với độ chính xác cao và các phát hiện này đóng góp đáng kể vào việc hiểu các tương tác giữa các thông số chất lượng không khí, cung cấp cơ sở cho việc ra quyết định trong các nỗ lực giảm thiểu ô nhiễm không khí tại khu vực DKI Jakarta.

2.1.1.3 Phân tích dữ liệu

2.1.1.3.1 Thống kê dữ liệu

Tính các thông số thống kê:

Lấy cột 'Curricular units 1st sem (grade)', ta thực hiện tính các thông số thống kê

```
sem1_grade = df_raw[ 'Curricular_units_1st_sem_(grade)' ]
```

- Mean: Áp dụng công thức 1.1

```
mean_sem1_grade = sem1_grade.sum() / sem1_grade.count()
```

$\rightarrow 10.640821575154185$

- Q1: Áp dụng công thức 1.6

```
pos = 0.25 * (n + 1)
k = int(np.floor(pos))
d = pos - k
```

```
if k <= 0:
    q1_sem1_grade = sem1_grade.iloc[0]
else:
    lower = sem1_grade.iloc[k - 1]
    upper = sem1_grade.iloc[k]
    q1_sem1_grade = lower + d * (upper - lower)
```

$\rightarrow 11.0$

- Median: Dữ liệu có $n=4424$, theo công thức 1.3

```
median_sem1_grade = (
    sem1_grade.iloc[n//2 - 1] + sem1_grade.iloc[n//2]
) / 2
```

$\rightarrow 12.285714285714286$

-
- Q3: Áp dụng công thức 1.6

```

pos = 0.75 * (n + 1)
k = int(np.floor(pos))
d = pos - k

if k >= n:
    q3_sem1_grade = sem1_grade.iloc[-1]
else:
    lower = sem1_grade.iloc[k - 1]
    upper = sem1_grade.iloc[k]
    q3_sem1_grade = lower + d * (upper - lower)

-> 13.4

```

- Mode: Áp dụng công thức 1.4

```

counts = sem1_grade.value_counts()
max_count = counts.max()
modes = counts[counts == max_count].index.tolist()

-> 0.0

```

- IQR: Từ Q3 và Q1 ta có 1.13 $IQR = 13.4 - 11.0 = 2.4$

- Variance: Áp dụng 1.8

```

squared_diffs = [(x - mean_sem1_grade) ** 2 for x in sem1_grade]

var_sem1_grade = sum(squared_diffs) / n

-> 23.455771808893843

```

- Standard Deviation: Áp dụng 1.10

```

sd_sem1_grade = math.sqrt(var_sem1_grade)

-> 4.843115919415293

```

- Coefficient of Variation: Áp dụng 1.12

```

cv_sem1_grade = sd_sem1_grade / mean_sem1_grade

-> 0.4551449232758249

```

Thực hiện tương tự, ta thu được bảng 2.2 thể hiện các thông số thống kê trên một số đặc trưng của bộ dữ liệu.

Từ đây, ta có thể rút ra một số nhận xét:

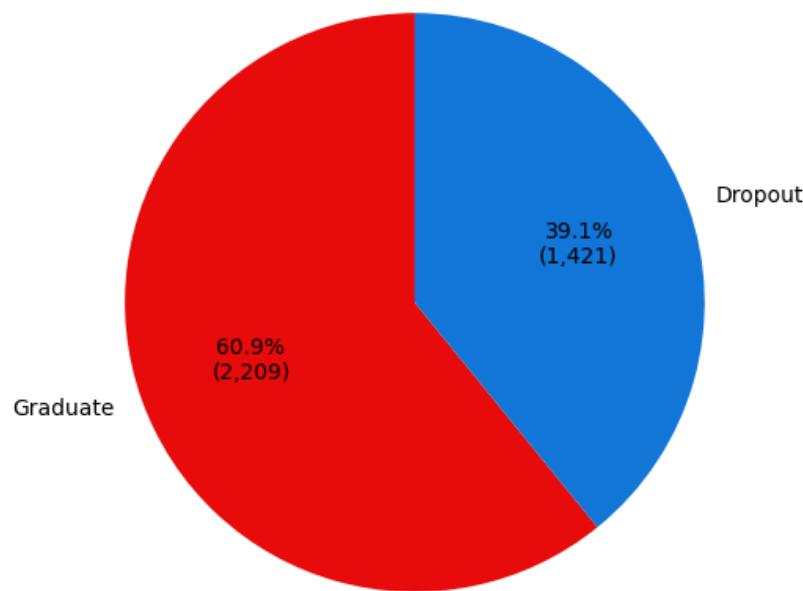
Bảng 2.2: Thống kê dữ liệu một số đặc trưng dữ liệu sinh viên

Thông số	Application order	Age at enrollment	Curricular units 1st sem (approved)	Curricular units 1st sem (grade)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)
Mean	1.7278	23.2651	4.7066	10.6408	4.4358	10.2302
Min	0.0000	17.0000	0.0000	0.0000	0.0000	0.0000
Q1	1.0000	19.0000	3.0000	11.0000	2.0000	10.7500
Median	1.0000	20.0000	5.0000	12.2857	5.0000	12.2000
Q3	2.0000	25.0000	6.0000	13.4000	6.0000	13.3333
Max	9.0000	70.0000	26.0000	18.8750	20.0000	18.5714
Mode	1.0000	18.0000	6.0000	0.0000	6.0000	0.0000
Variance	1.7261	57.5749	9.5743	23.4611	9.0888	27.1525
SD	1.3138	7.5878	3.0942	4.8437	3.0148	5.2108
CV	0.7604	0.3261	0.6574	0.4552	0.6796	0.5094
IQR	1.0000	6.0000	3.0000	2.4000	4.0000	2.5833

- Application order: Có trung vị 1, trung bình ≈ 1.73 và mode 1, cho thấy đa số sinh viên đậu vào trường ở nguyện vọng 2 của họ. Cho thấy đa số sinh viên thường chọn nguyện vọng 1 là mục tiêu cao hơn, và nguyện vọng 2 là vị trí an toàn cho khả năng của họ.
- Age at enrollment: Mode = 18 cho thấy tuổi sinh viên nhập học nhiều nhất là ngay vừa tốt nghiệp xong trung học. Trung bình 23.265, trung vị 20 cho thấy phân phối tuổi lệch về phải.
- Curricular units 1st sem (approved): Có trung vị 5 và trung bình 4.7, cho thấy đa số sinh viên có số môn đậu trong kỳ thấp. Mode = 6 cho thấy đa số sinh viên chỉ tích lũy 6 môn. Tuy nhiên max đạt đến 26, không chỉ việc học 26 môn là không hợp lý, đó cũng là một giá trị quá lệch cho thấy có thể có vấn đề trong thu thập dữ liệu (ví dụ như có thể tính nhầm môn được miễn vào trong này...), nên loại bỏ ngoại lệ lớn này khi trực quan, mô hình hóa dữ liệu.
- Curricular units 1st sem (grade): Với thang điểm 0-20(*), điểm trung bình học kỳ 1 có trung vị ≈ 12 và trung bình 10.64, Q1 = 11 cho thấy đa số sinh viên đạt điểm trên trung bình. Phân phối điểm lệch về trái chứng tỏ có một số trường hợp điểm thấp kéo trung bình xuống. Khi sinh viên bỏ thi thì tự động 0 điểm, cho nên mode = 0 là bình thường không đáng lo ngại. Khoảng điểm min-max 0-18.75 nằm trong thang điểm thực tế, cho thấy không vấn đề về thu thập dữ liệu.
- Curricular units 2nd sem (approved): Số môn đạt được trong kỳ 2 cũng gần giống với học kỳ 1. Đa số sinh viên tích lũy số môn thấp với trung bình 4.4358, trung vị 5. Có cùng mode = 6 với học kỳ 1 cho thấy đa số sinh viên chỉ đạt 6 môn mỗi kỳ. Ta cũng thấy max = 20 là ngoại lệ lớn, cho thấy phần nào nhất quán với số môn học kỳ 1, cần chú ý trong trực quan và mô hình hóa.

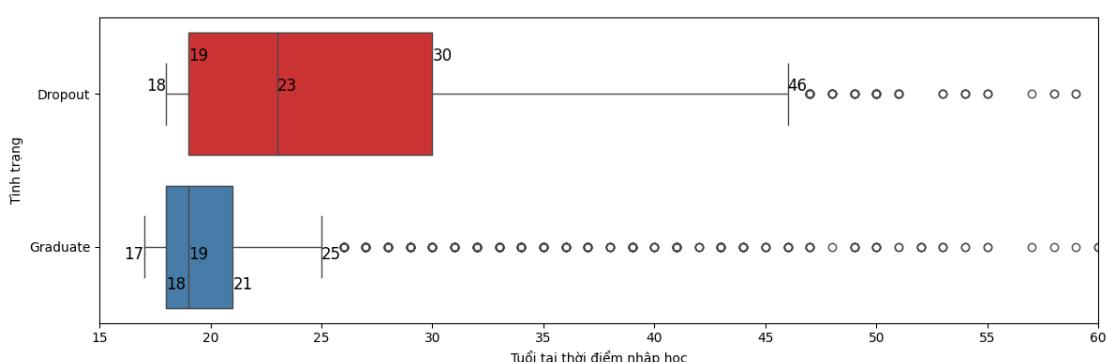
- Curricular units 2nd sem (grade): Điểm trung bình học kỳ 2 của các sinh viên cũng có đặc điểm tương tự như điểm trung bình học kỳ 1. Đa phần sinh viên qua được mức trung bình, với trung bình ≈ 10.23 , trung vị 12.2, dữ liệu lệch về trái hơn do một lượng sinh viên bỏ thi, cũng là lý do cho mode = 0.

2.1.1.3.2 Trực quan hóa dữ liệu



Hình 2.1: Số lượng sinh viên theo tình trạng

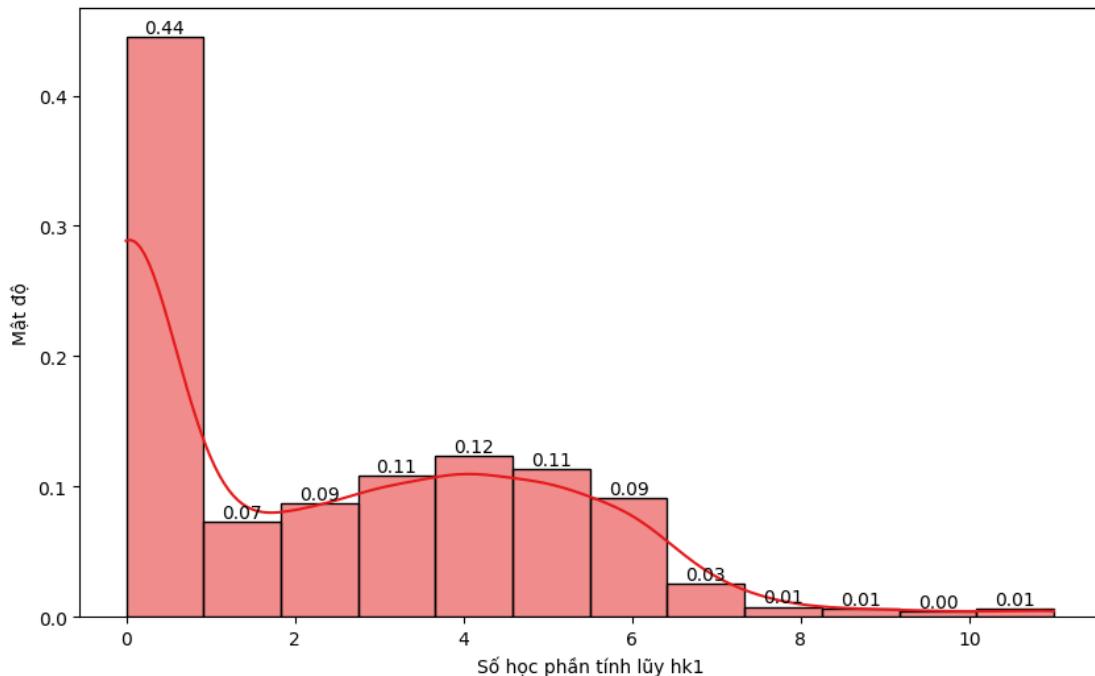
Biểu đồ pie(1.2.5) 2.1 cho thấy sự phân phối dữ liệu giữa nhãn Dropout (đã nghỉ học) và Graduate (đã tốt nghiệp). Dữ liệu 2 nhãn có phân phối gần 40:60, có mốc cân bằng nhẹ.



Hình 2.2: Tuổi khi nhập học theo tình trạng

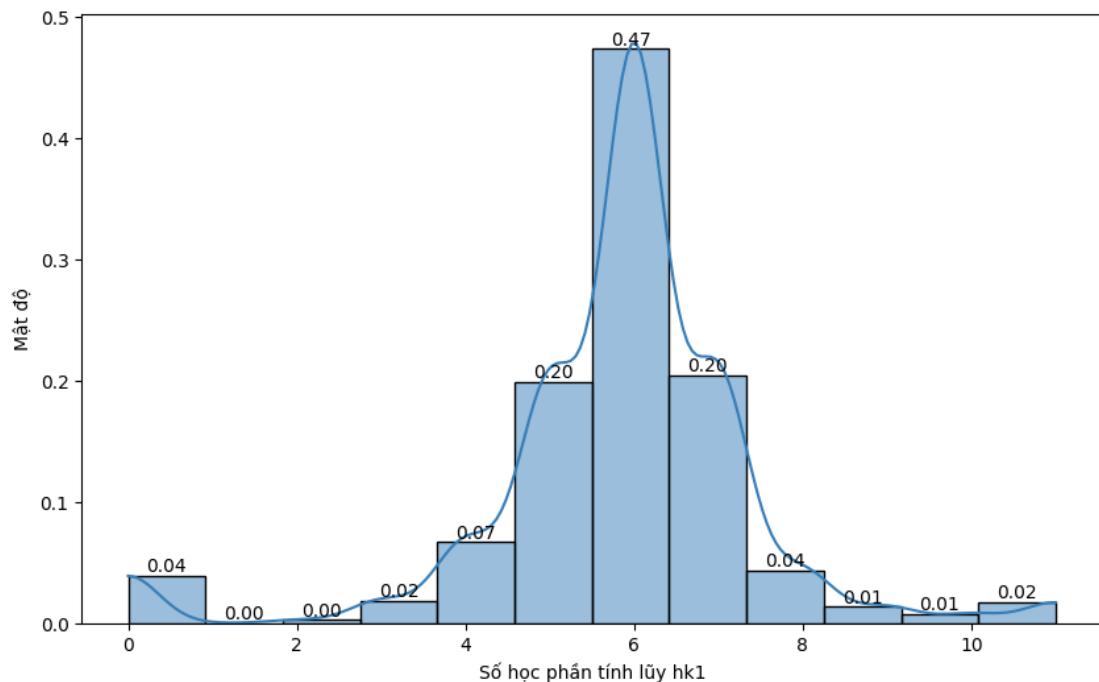
Biểu bô boxplot(1.2.2) 2.2 cho thấy tồn tại các sinh viên tốt nghiệp dù độ tuổi nhập học cao, nhưng đa phần sinh viên tốt nghiệp là người nhập học trong độ tuổi 18-21. Qua 25 tuổi nhập học mà tốt nghiệp đã có thể xem là ngoại lệ. Nhóm bỏ học cao hơn rõ ràng, cho thấy độ tuổi nhập học cao có liên hệ với nguy cơ bỏ học.

Xét số môn tích lũy hk 1 có trường hợp ngoại lệ cao, áp dụng $UpperFence = Q_3 + 1.5 * IQR = 6 + 1.5 * 3 = 10.5$ (bảng 2.2), ta chọn số tín chỉ = 11 là tối đa để loại bỏ các điểm ngoại lệ. Từ đó xây dựng biểu đồ hình 2.3 và hình 2.4



Hình 2.3: Phân phối số môn tích lũy hk 1 nhóm Dropout

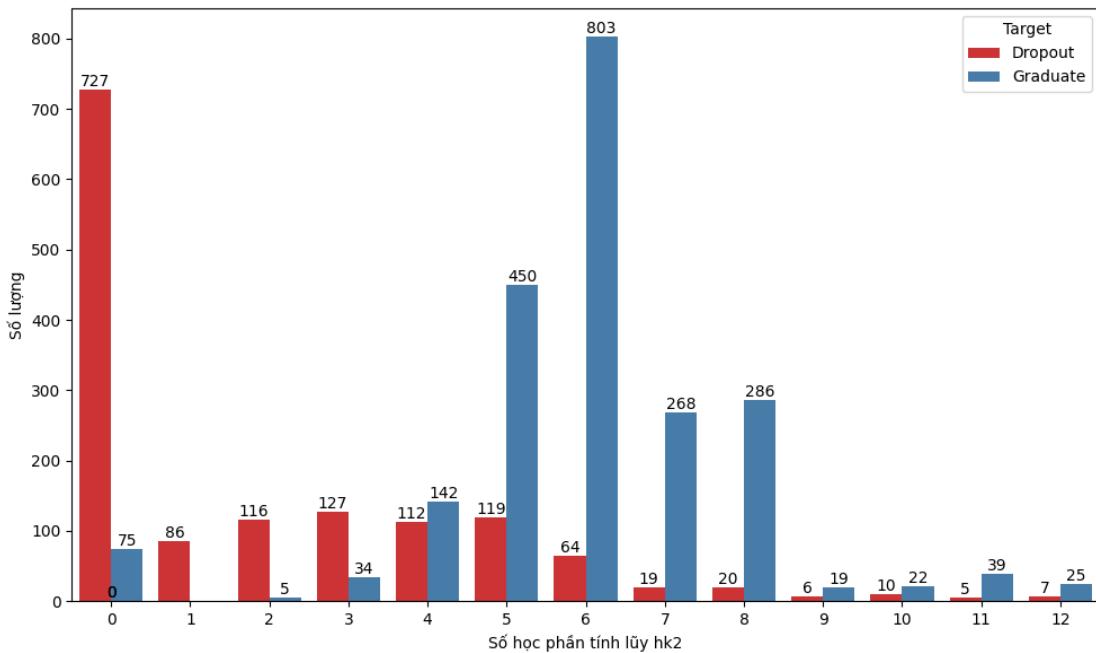
Biểu đồ histogram(1.2.1) 2.3 thể hiện phân phối số môn tích lũy học kỳ một nhóm Dropout lệch về phải, phần lớn sinh viên bỏ học tích lũy ít học phần, số lượng tích lũy nhiều học phần giảm dần. Cột 0 học phần chiếm phân phối cao nhất ở 0.44, cho thấy sinh viên không tích lũy được học phần nào có nguy cơ bỏ học cao nhất. Đường KDE với đỉnh ở cột 0, cong giảm dần về bên phải cũng phản ánh điều tương tự, nhóm sinh viên bỏ học sẽ có xu hướng tích lũy được ít hoặc không môn nào.



Hình 2.4: Phân phối số môn tích lũy hk 1 nhóm Graduate

Biểu đồ histogram 2.4 thể hiện phân phối số môn tích lũy học kỳ một nhóm Graduate là dạng histogram chuẩn, đỉnh phân bố tập trung tại 6 môn. Đường KDE có hình chuông, đối xứng quanh trung tâm, cho thấy phần lớn sinh viên tốt nghiệp đều đạt được mức học phần tích lũy tương đối, với một số rất ít trường hợp đặc biệt tích lũy rất ít hoặc nhiều học phần.

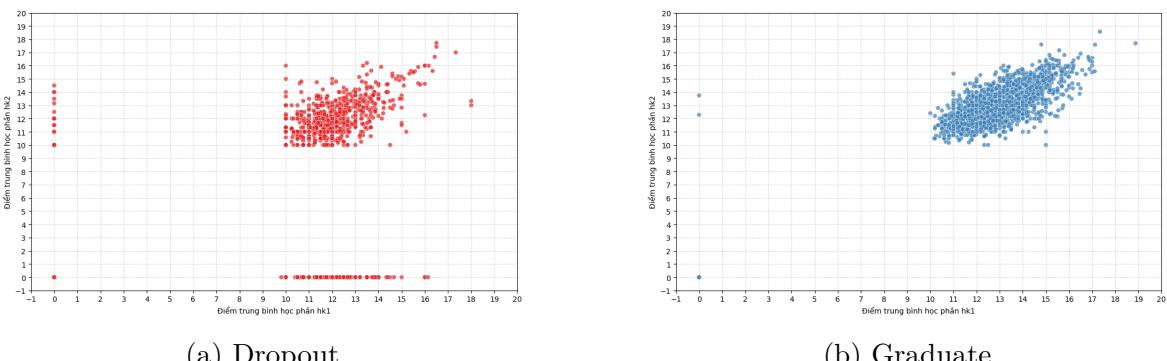
Số môn tích lũy hk 2 cũng tồn tại trường hợp ngoại lệ cao, lần nữa áp dụng $UpperFence = Q_3 + 1.5 * IQR = 6 + 1.5 * 4 = 12$ (bảng 2.2), ta chọn số tín chỉ = 12 là tối đa để loại bỏ các điểm ngoại lệ.



Hình 2.5: Phân phối số môn tích lũy hk 2

Biểu đồ cột(1.2.6) 2.5 cho thấy khác biệt rõ rệt số môn tích lũy hk 2 giữa nhóm bỏ học và tốt nghiệp. Nhóm sinh viên bỏ học thường có số môn đạt được trong kỳ thấp có số lượng áp đảo, nhiều nhất ở cột 0. Ngược lại, những sinh viên tích lũy được từ 5 học phần trở lên thuộc nhóm tốt nghiệp cao hơn đáng kể, số lượng vượt xa số bỏ học.

Từ các biểu đồ phân tích số học phần tích lũy hai kỳ, có thể kết luận số môn đạt được mỗi kỳ liên quan chặt chẽ với tình trạng bỏ học, với sinh viên có số tích lũy ít có xu hướng bỏ học cao.

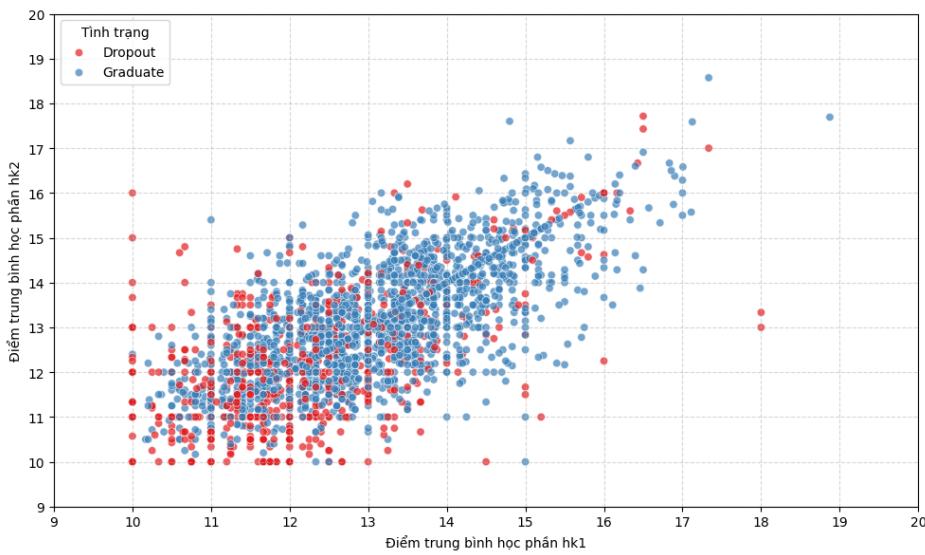


Hình 2.6: Phân bố điểm trung bình sinh viên

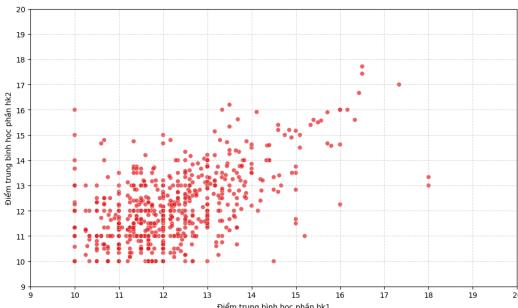
Các biểu đồ Scatter(1.2.3) 2.6 thể hiện quan hệ điểm trung bình hk1, hk2 và tình trạng của sinh viên. Hình 2.6 cho thấy nhóm tốt nghiệp toàn bộ (trừ một vài rất ít điểm ngoại lệ) có điểm trung bình cả hai kỳ hoàn toàn trên trung bình và tập trung ở vùng có điểm cao. Trong khi đó, từ hình 2.6a thấy được ở nhóm bỏ học, có nhiều trường hợp bỏ học/thi/rớt môn một hoặc cả hai kỳ, tạo thành các điểm trên đường

$x=0$, $y=0$ và điểm $(0,0)$. Nhiều điểm tập trung chính xác trên đường $x=10$ và $y=10$, có thể suy đoán đây là các trường hợp được vớt điểm đủ trung bình.

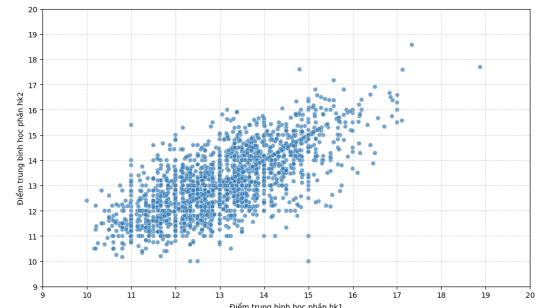
Dữ liệu điểm hai học kỳ không có giá trị trong khoảng $(0, 10)$, có thể thấy với bộ dữ liệu các trường hợp có điểm dưới trung bình, rớt môn, sẽ không được tính điểm tích lũy. Từ đó, để nhìn rõ hơn 2 cụm điểm nhóm tốt nghiệp và bỏ học, ta vẽ hình 2.7 với trục x , y giới hạn để phóng to đoạn 10-20.



(a) Kết hợp



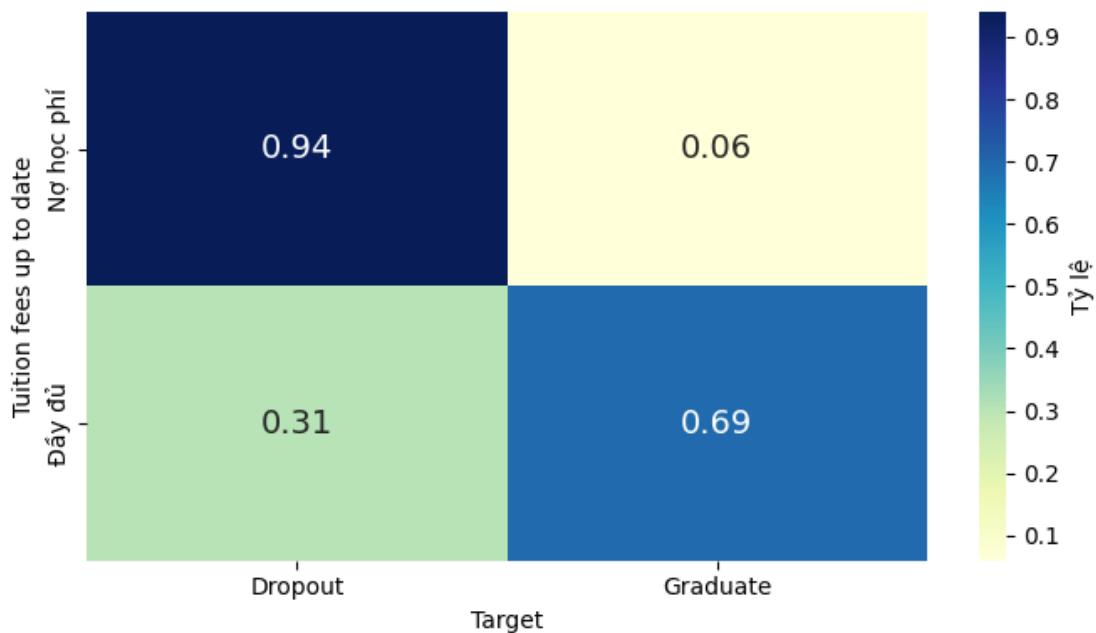
(b) Dropout



(c) Graduate

Hình 2.7: Phân bố điểm trung bình sinh viên (to phóng đoạn 9-20)

Hình 2.7, cho thấy rõ điểm trung bình hai kỳ nhóm tốt nghiệp cao, tạo thành một cụm từ khoảng 12-16 điểm cho cả 2 kỳ. Còn nhóm rớt môn có cụm điểm phân tán hơn, hướng về dưới và bên trái, tức điểm của hai kỳ có xu hướng nhỏ hơn so với nhóm tốt nghiệp. Từ hai hình 2.6, 2.7, có thể kết luận điểm số trung bình có mối quan hệ với tình trạng bỏ học của sinh viên, đặc biệt khi điểm gần sát mức trung bình hoặc dưới trung bình ($0\bar{d}$).



Hình 2.8: Liên hệ giữa tình trạng nợ học phí và tình trạng sinh viên

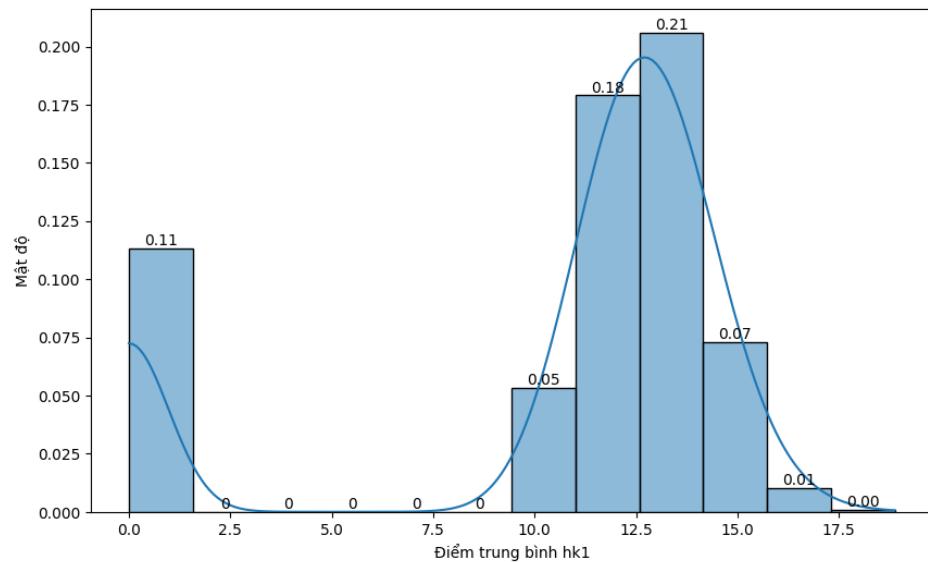
Biểu đồ heatmap(1.2.4) hình 2.8 thể hiện phần lớn sinh viên bỏ học có xu hướng nợ học phí, chiếm tỉ lệ đến 0.94. Còn các sinh viên đóng học phí đầy đủ, chỉ 0.31 bỏ học. Từ đây cho thấy tình trạng nợ học phí có mối quan hệ mạnh với nguy cơ bỏ học.

2.1.1.4 Data transformation

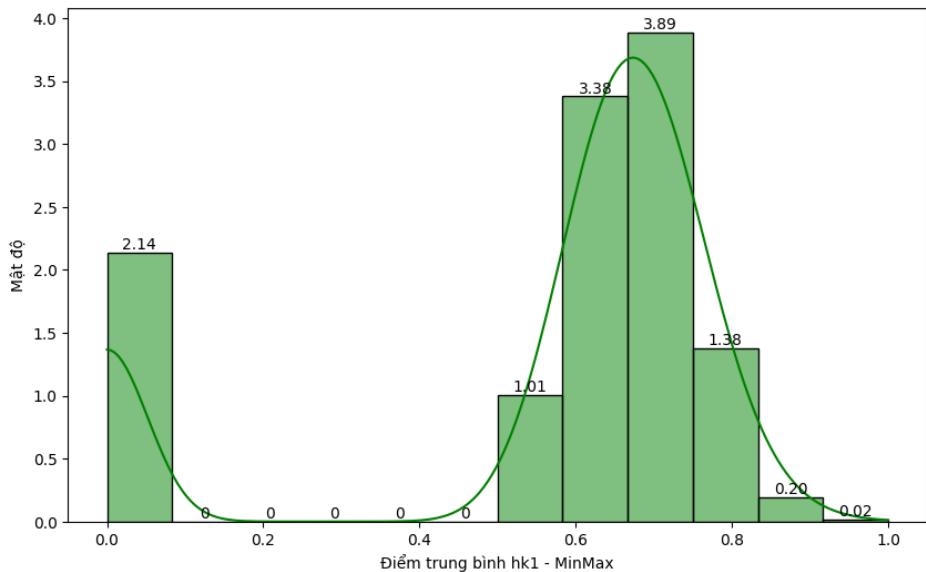
Biểu diễn các thuật toán data transformation: min-max scaler(1.4.1), standard scaler(1.4.2), robust scaler(1.4.3), Max ABS scaler(1.4.4), Quantile scaler(1.4.5) trên cột 'Curricular units 1st sem (grade)' ta thu được bảng thống kê sau:

Bảng 2.3: Biểu diễn thống kê cột 'Curricular units 1st sem (grade)' qua các scalar

	Dữ liệu gốc	Min-Max	Standard	Robust	MaxAbs	Quantile
Mean	10.535	0.558	0	-0.723	0.558	0.484
Min	0	0	-2.083	-4.937	0	0
Q1	11	0.583	0.092	-0.537	0.583	0.249
Median	12.341	0.654	0.357	0	0.654	0.5
Q3	13.5	0.715	0.586	0.463	0.715	0.754
Max	18.875	1	1.649	2.613	1	1
Mode	0	0	-2.083	-4.937	0	0
Var	25.58	0.072	1	4.093	0.072	0.097
SD	5.058	0.268	1	2.023	0.268	0.312
CV	0.48	0.48		-2.8	0.48	0.644
IQR	2.5	0.132	0.494	1	0.132	0.505

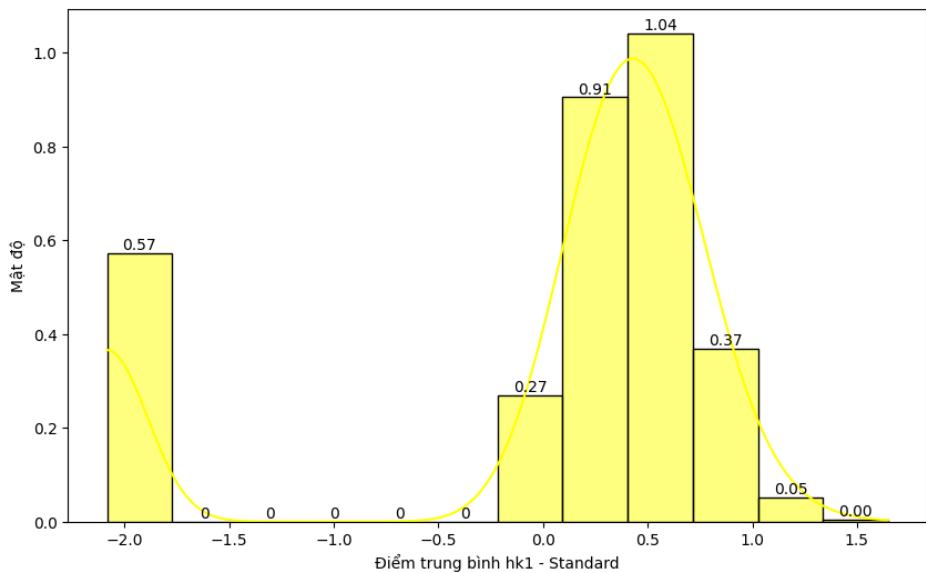


Hình 2.9: Histogram 'Curricular units 1st sem (grade)' gốc



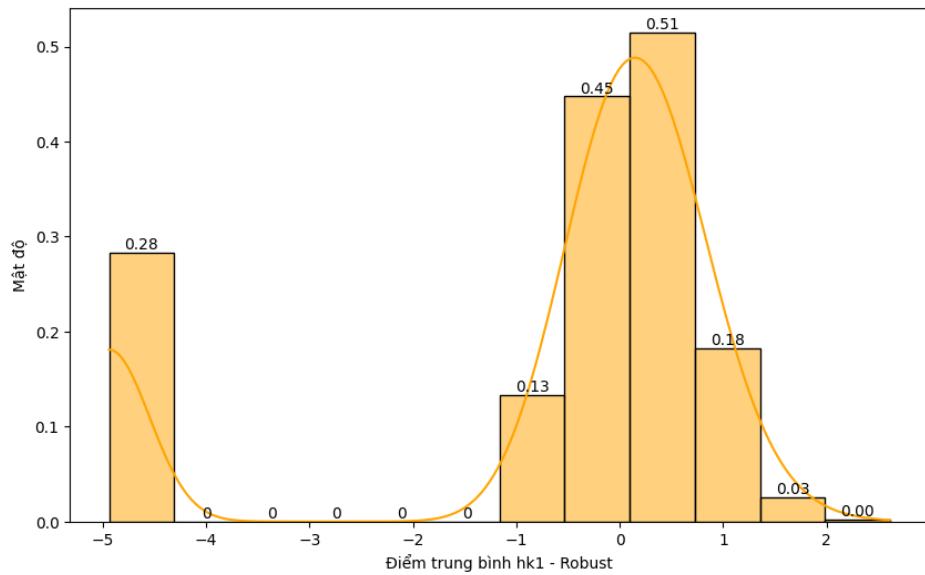
Hình 2.10: Histogram 'Curricular units 1st sem (grade)' qua Min-Max Scaler

Min-Max scaler biến đổi dữ liệu về khoảng $[0, 1]$. Min, max của đặc trưng 'Curricular units 1st sem (grade)' trở thành 0, 1. Trung bình trở thành 0.558,... (bảng 2.3). Khoảng giá trị dữ liệu thay đổi nhưng hình dáng phân phối được giữ nguyên (hình 2.10)



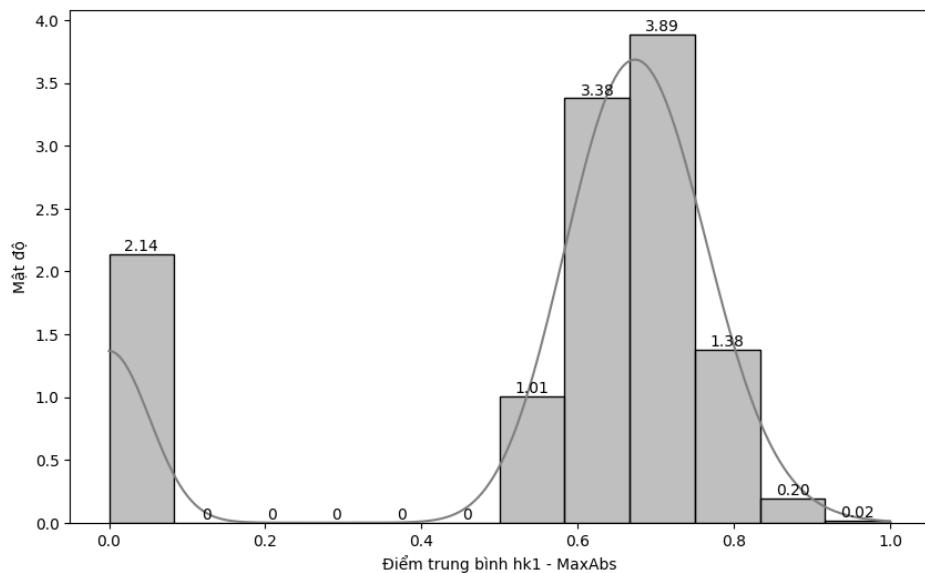
Hình 2.11: Histogram 'Curricular units 1st sem (grade)' qua Standard Scaler

Standard scaler chuyển đổi dữ liệu về mean = 0 và sd = 1. Dữ liệu mới có min, max = -2.083, 1.649 (bảng 2.3). CV không thể tính được vì mean = 0. Hình dáng dữ liệu được giữ nguyên (hình 2.11)



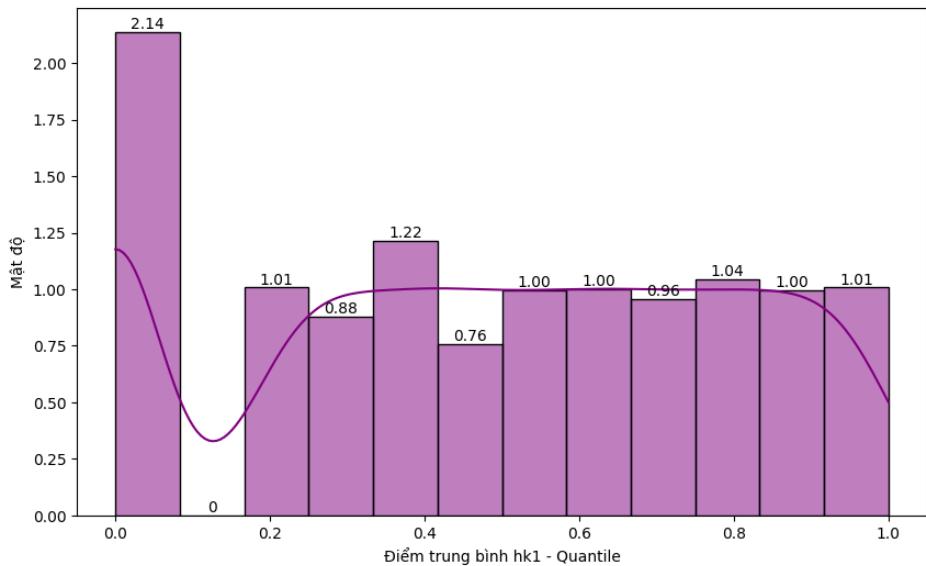
Hình 2.12: Histogram 'Curricular units 1st sem (grade)' qua Robust Scaler

Robust scaler sử dụng trung vị thay vì trung bình và IQR thay vì độ lệch chuẩn (1.4.3), nên kết quả sau chuẩn hóa cho trung vị bằng 0 và IQR bằng 1, tuy nhiên các giá trị lớn và nhỏ vẫn giữ nguyên độ lệch lớn so với phần giữa: min, max có giá trị -4.937, 2.613, lớn hơn khoảng so với sử dụng standard scaler (bảng 2.3). Hình dáng dữ liệu được giữ nguyên (hình 2.12)



Hình 2.13: Histogram 'Curricular units 1st sem (grade)' qua Max ABS Scaler

Max ABS scaler chia từng điểm dữ liệu cho giá trị tuyệt đối lớn nhất (1.4.4) để đưa dữ liệu về đoạn $[-1, 1]$. Tuy nhiên dữ liệu cột điểm trung bình hk 1 ở đây không có dữ liệu âm, nên kết quả thu được là tương tự với Min-Max Scaler (bảng 2.3, hình 2.13)



Hình 2.14: Histogram 'Curricular units 1st sem (grade)' qua Quantile Scaler

Quantile Scaler chuyển dữ liệu về phân phối đều trong khoảng [0,1] (1.4.5), nên hình dánh phân phối trở nên khá phẳng (hình 2.14); Q1, trung vị, Q3 trở thành gần chính xác 0.25, 0.5, 0.75 (bảng 2.3)

2.1.1.5 Mô hình hóa dữ liệu

2.1.1.5.1 Cấu hình cài đặt

Các mô hình sử dụng:

- **Logistic Classification:**

```
LogisticRegressionCV (
    max_iter=max_iter ,
    Cs=cs ,
)
```

khoảng hypertune tham số

```
list_max_iter = [10, 20, 30, 40, 50, 100, 200, 300, 500]
list_cs = [1, 2, 3, 4, 5, 8, 10]
```

- **Random Forest Classifier:**

```
RandomForestClassifier (
    max_depth=max_depth ,
    n_estimators=n_estimators ,
    min_samples_leaf=min_samples_leaf ,
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 5, 10]
list_n_estimators = [50, 100, 150, 200]
list_min_samples_leaf = [3, 5, 10, 20]
```

- **XGBoost Classifier:**

```
XGBClassifier(
    max_depth=max_depth,
    n_estimators=n_estimators,
    reg_lambda=reg_lambda,
    learning_rate=lr,
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 4, 5]
list_lambda = [3, 2, 1, 0.5]
list_learning_rate = [0.5, 0.25, 0.1, 0.05]
list_n_estimators = [25, 50, 100, 150]
```

Thuật toán data transform sử dụng: Do dữ liệu có nhiều điểm ngoại lệ, nhóm chọn Robust Scaler để chuẩn hóa dữ liệu.

Chia tập dữ liệu huấn luyện: Cross-validation 5 folds.

2.1.1.5.2 Phân lớp tình trạng sinh viên

Chọn các đặc trưng huấn luyện: 'Tuition fees up to date', 'Scholarship holder', 'Debtor', 'Application order', 'Age at enrollment', 'Curricular units 1st sem (approved)', 'Curricular units 1st sem (grade)', 'Curricular units 2nd sem (approved)', 'Curricular units 2nd sem (grade)'

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 10
- cs: 2

Bảng 2.4: Kết quả Logistic Classification - Phân lớp tình trạng sinh viên

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.889935	0.889082	0.818310	0.894695	0.854619	0.930804
std	0.001182	0.005009	0.019150	0.010877	0.007953	0.008678
min	0.888097	0.881766	0.803571	0.881481	0.844278	0.921640
max	0.891266	0.894437	0.850000	0.908000	0.865455	0.942839

- **Random Forest Classifier:**

Mô hình tốt nhất:

- max_depth: 10
- n_estimators: 100
- min_samples_leaf: 3

Bảng 2.5: Kết quả Random Forest Classifier - Phân lớp tình trạng sinh viên

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.921657	0.895071	0.824762	0.903870	0.862393	0.940038
std	0.002541	0.008536	0.014551	0.016729	0.011015	0.006931
min	0.918746	0.886040	0.800000	0.888889	0.848485	0.933446
max	0.925517	0.907275	0.838710	0.931452	0.876660	0.951630

- **XGBoost Classifier:**

Mô hình tốt nhất:

- max_depth: 4
- n_estimators: 100
- reg_lambda: 3
- learning_rate: 0.1

Bảng 2.6: Kết quả XGBoost Classifier - Phân lớp tình trạng sinh viên

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.917023	0.895355	0.823323	0.905815	0.862492	0.937652
std	0.003497	0.009013	0.016033	0.015954	0.011883	0.006926
min	0.913400	0.883191	0.796429	0.885496	0.844697	0.928940
max	0.921953	0.904422	0.839286	0.924000	0.873346	0.947808

So sánh các mô hình:

Bảng 2.7: So sánh kết quả các mô hình - Dự đoán tình trạng sinh viên

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision	Mean ROC AUC
Logistic Regression	0.889082	0.854619	0.818310	0.894695	0.930804
Random Forest	0.895071	0.862393	0.824762	0.903870	0.940038
XGBoost	0.895355	0.862492	0.823323	0.905815	0.937652

XGBoost, là mô hình phức tạp nhất, cho kết quả tốt nhất, dẫn đầu ở Accuracy, F1 và Precision. Random Forest có riêng Recall và ROC AUC cao hơn, nếu phát hiện sinh viên bỏ học (recall cao) được xem qua quan trọng hơn kết quả toàn diện thì có thể ưu tiên mô hình Random Forest.

2.1.1.6 Kết luận

Trong phần này, nhóm đã tiến hành phân tích và xây dựng mô hình dự đoán tình trạng sinh viên. Qua mô hình hóa cho thấy các đặc trưng liên quan đến kết quả học tập các học kì đầu, tuổi nhập học và tình trạng tài chính có ảnh hưởng rõ rệt đến khả năng sinh viên bỏ học. Kết quả thu được mô hình tốt nhất XGBoost Classifier với độ chính xác 0.895, F1 0.862, Recall 0.823, Precision 0.905. Đây là cơ sở quan trọng để các cơ sở giáo dục đại học có thể sớm nhận diện và hỗ trợ các sinh viên có nguy cơ bỏ học, từ đó nâng cao hiệu quả đào tạo và giảm thiểu tỷ lệ bỏ học.

2.1.2 MAL show Dataset

2.1.2.1 Giới thiệu bộ dữ liệu

2.1.2.1.1 Nguồn dữ liệu

Bộ dữ liệu được thu thập bằng Jikan api, nguồn cơ sở dữ liệu website myanimelist.net

2.1.2.1.2 Mô tả dữ liệu

Đây là bộ dữ liệu chức thông tin các bộ phim và sản phẩm phương tiện truyền thông liên quan đến hoạt hình Nhật Bản, được cập nhật cho đến tháng 6 năm 2025 từ cơ sở dữ liệu của website My Anime List (MAL). Mỗi dòng dữ liệu chứa thông tin về bộ phim kèo theo thông tin đánh giá, nhận xét của người dùng.

Ví dụ một phần dữ liệu:

Bảng 2.8: Một phần bảng dữ liệu MAL dataset

mal_id	title_english	title_japanese	type	status	score	rank	members	...
33352	Violet Evergar-den	ヴァイオレット エヴァーガーデン	TV	Finished Airing	8.68	67	1,904,842	...
35677	Liz and the Blue Bird	リズと青い鳥	Movie	Finished Airing	8.22	375	154,915	...
41457	86 Eighty-Six	86—エイティシックス—	TV	Finished Airing	8.33	255	890,163	...
52991	Frieren: Beyond Journey's End	葬送のフリレン	TV	Finished Airing	9.3	1	1,140,504	...
...

Bộ dữ liệu có 28,707 dòng, bao gồm 20 cột như sau:

- Kiểu dữ liệu: **Qualitative**

1. **title**: Tên (chung, có thể là dịch hoặc phiên âm)

2. **title_english**: Tên dịch tiếng Anh

3. **title_japanese**: Tên gốc tiếng Nhật

4. **type**: Thể loại

Có các giá trị:

- TV: Phim chiếu TV
- Movie: Phim chiếu rạp
- Special: Phần đặc biệt
- OVA: Phần đặc biệt (tặng kèm DVD, Blu-ray...)
- TV Special: Phần đặc biệt (chiếu TV)
- ONA: Phát trực tuyến
- Music: Âm nhạc
- CM: Quảng cáo (TV, ...)
- PV: Quảng cáo (trailer, teaser...)

5. **status**: Tình trạng chiếu

Có các giá trị:

- Finished Airing: Đã chiếu hết
- Currently Airing: Đang chiếu
- Not yet aired: Sắp chiếu

6. **synopsis**: Tóm tắt nội dung

7. **genres**: Thể loại, chuỗi danh sách ngăn cách bởi dấu ','

8. **themes**: Chủ đề chính, chuỗi danh sách ngăn cách bởi dấu ','

-
- 9. **demographics**: Đối tượng khán giả, chuỗi danh sách ngắn cách bởi dấu ‘,’
 - 10. **studios**: Tên xưởng phim thực hiện
 - 11. **url**: Đường dẫn trang web của dòng hiện tại
- Kiểu dữ liệu: **Quantitative**
 - **Discrete**:
 - 12. **mal_id**: ID phim trong cơ sở dữ liệu
 - 13. **episodes**: Số tập
 - 14. **members**: Số người theo dõi (đang xem, đã xem)
 - 15. **favorites**: Số người đánh giá yêu thích (Mỗi tài khoản chỉ có thể chọn yêu thích tối đa 10, 20 nếu là tài khoản trả phí)
 - 16. **rank**: Xếp hạng (theo đánh giá)
 - 17. **popularity**: Xếp hạng (theo số người theo dõi)
 - 18. **year**: Năm bắt đầu chiếu
 - **Continuous**:
 - 19. **score**: Điểm đánh giá trung bình tất cả đánh giá. Thang 0-10
 - 20. **aired_date**: Ngày bắt đầu chiếu

2.1.2.2 Các nghiên cứu liên quan

Jesús Armenta-Segura và Grigori Sidorov [1], các phương pháp học máy truyền thống đã được ứng dụng nhằm dự đoán mức độ thành công của các bộ anime ngay từ giai đoạn phát triển ban đầu. Tác giả đã xây dựng AniSyn7 – một tập dữ liệu gồm 6.928 tóm tắt nội dung phim hoạt hình, được gán nhãn nhị phân (thành công/không thành công) dựa trên điểm số từ MyAnimeList. Nghiên cứu đã triển khai ba bộ phân loại học máy phổ biến gồm Máy vector hỗ trợ (SVM), Naive Bayes và Hồi quy Logistic, kết hợp với các phương pháp vector hóa như Bag of Words, n-grams và cây phụ thuộc cú pháp. Kết quả cho thấy, mô hình BBTC (kết hợp BoW, bigram, trigram và trigram ký tự) sử dụng SVM hoặc Logistic Regression đạt F1-score cao nhất là 0.55. Điều này chứng minh rằng các đặc trưng ngôn ngữ trong phần tóm tắt nội dung có thể mang lại giá trị đáng kể trong việc dự đoán thành công của anime. Nghiên cứu không chỉ cung cấp một tập dữ liệu nền tảng cho các nghiên cứu sau này mà còn mở ra tiềm năng mở rộng theo hướng đa nhiệm hoặc đa phương thức, bao gồm cả phân tích hình ảnh và kỹ thuật học sâu hiện đại nhằm nâng cao độ chính xác và khả năng ứng dụng trong thực tiễn.

Trong nghiên cứu của Nathanael Setiawan và cộng sự [24], mô hình chuỗi thời gian Prophet được áp dụng nhằm dự đoán xu hướng phổ biến của các thể loại anime trong tương lai. Dựa trên dữ liệu từ MyAnimeList, nhóm nghiên cứu sử dụng Prophet để phân tích sự thay đổi mức độ phổ biến theo thời gian, đạt RMSE 1.102 và MAPE 13.551%. Kết quả cho thấy các thể loại như Super Power, Demons và Supernatural sẽ tiếp tục tăng trưởng đến năm 2025, trong khi Josei, Cars và Kids giảm mạnh. Nghiên

cứu này là một trong số ít ứng dụng chuỗi thời gian trong lĩnh vực anime, hỗ trợ các nhà sản xuất và nhóm nội địa hóa định hướng phát triển thể loại phù hợp với thị hiếu toàn cầu.

2.1.2.3 Phân tích dữ liệu

2.1.2.3.1 Tiền xử lý

Trước tiên, nhóm thực hiện một số bước tiền xử lý bộ dữ liệu thô như sau:

- Bộ dữ liệu không chứa thông tin bình luận đánh giá. Nhóm sử dụng cùng api nguồn dữ liệu để thu thập thêm thông tin bình luận, đánh giá như sau:

Thu thập dữ liệu tạo thành cột review_tags_count chứa Json chứa dữ liệu phê bình, sau đó tách dữ liệu thành các cột

- **Recommended:** Số đánh giá tích cực
- **NotRecommended:** Số đánh giá tiêu cực
- **MixedFeelings:** Số đánh giá trung lập
- **Creative:** Số bài đánh giá được đánh giá là sáng tạo
- **Funny:** Số bài đánh giá được đánh giá là hài hước
- **Informative:** Số bài đánh giá được đánh giá là cung cấp nhiều thông tin
- **Well-written:** Số bài đánh giá được đánh giá là viết hay

- Thực hiện kiểm tra dữ liệu thiếu:

Đặc trưng quan trọng 'score' hiện có 5,369 dòng không có dữ liệu. Việc này có thể xảy ra cho những trường hợp không đủ người đánh giá để có điểm đánh giá, những phim chưa chiếu,... Lọc bỏ các dòng này khỏi dữ liệu. Dữ liệu còn lại có 7,870 dòng.

- Dataset có quá nhiều thể loại media có thể gây nhiễu dữ liệu, cho nên đầu tiên nhóm lọc lại 'type' = ('TV', 'Movie'), chỉ quan tâm những dòng thực sự là phim chiếu
- Các cột đang là danh sách chuỗi phân tách bởi dấu ',', cần phải xử lý. Ở đây nhóm thực hiện áp dụng one-hot encoding.

2.1.2.3.2 Thông kê dữ liệu

Bảng 2.9 thể hiện các thông số thống kê trên một số đặc trưng của bộ dữ liệu.

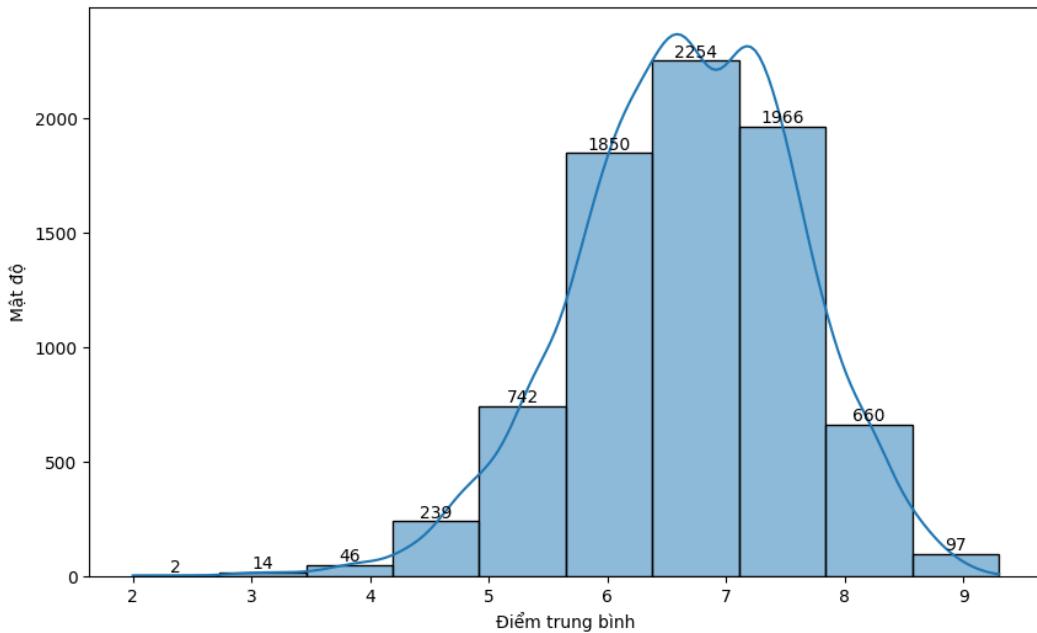
Từ đây, ta có thể rút ra một số nhận xét:

Bảng 2.9: Thống kê dữ liệu một số đặc trưng dữ liệu MAL dataset

	score	favorites	members	Recommended	Mixed Feelings	Not Recommended
Mean	6.669	1,485.5222	120,803.3152	5.0469	1.5848	1.3471
Min	2	0	206	0	0	0
Q1	6.07	3	2,162.5	0	0	0
Median	6.7	32	15,593.5	2	1	0
Q3	7.33	346	100,486	9	3	2
Max	9.3	238,872	4,173,914	20	13	17
Mode	6.42	0	423	0	0	0
Var	0.8738	$7.1 * 10^7$	$8.8 * 10^{10}$	34.0366	4.2754	5.4627
SD	0.9348	8,452.6298	297,372.0879	5.8341	2.0677	2.3372
CV	0.1402	5.69	2.4616	1.156	1.3047	1.735
IQR	1.26	343	98,323.5	9	3	2

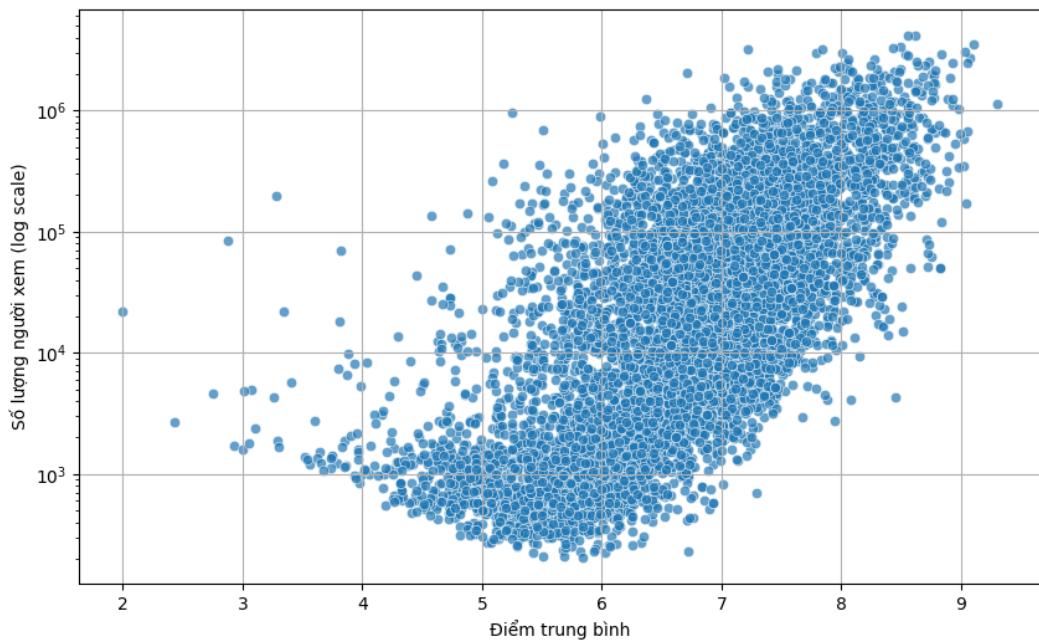
- Score: Điểm trung bình là 6.669, trung vị 6.7, cho thấy phân phối điểm khá đồng đều và tập trung quanh mức trung bình. Độ lệch chuẩn thấp (0.9348) thể hiện mức độ biến thiên điểm số thấp. Khoảng điểm từ 2 đến 9.3 nằm trong thang điểm hợp lý, không có dữ liệu lỗi.
- Favorites: Số lượng phim được đánh dấu yêu thích có mức trung bình 1,485, nhưng trung vị chỉ là 32 và mode = 0, cho thấy đa phần phim có ít lượt yêu thích, một số rất ít phim nổi bật có số yêu thích cực cao (max = 238,872). Sự chênh lệch lớn được phản ánh rõ qua độ lệch chuẩn rất cao (SD = 8,452.63), cho thấy dữ liệu lệch phai rất mạnh, cần lưu ý trong quá trình mô hình hóa.
- Members: Số lượng người xem có giá trị trung bình rất cao (120,803), nhưng trung vị chỉ 15,593 và mode là 423, cũng như Favorite cho thấy đa số người dùng chỉ xem một lượng ít phim nổi bật, số đó kéo trung bình lên cao (max lên đến hơn 4 triệu), trong khi đa số phim có ít người xem.
- Các lượt đánh giá có trung bình thấp, trung vị thấp và mode = 0, giá trị tối đa nhiều nhất cũng chỉ đạt 20 ở lượt đánh giá tích cực. So với members, cho thấy đa số người dùng không viết đánh giá. Vậy khi một người chọn viết đánh giá, ý kiến đó có thể có ảnh hưởng lớn, cần phân tích thêm

2.1.2.3.3 Trực quan hóa dữ liệu



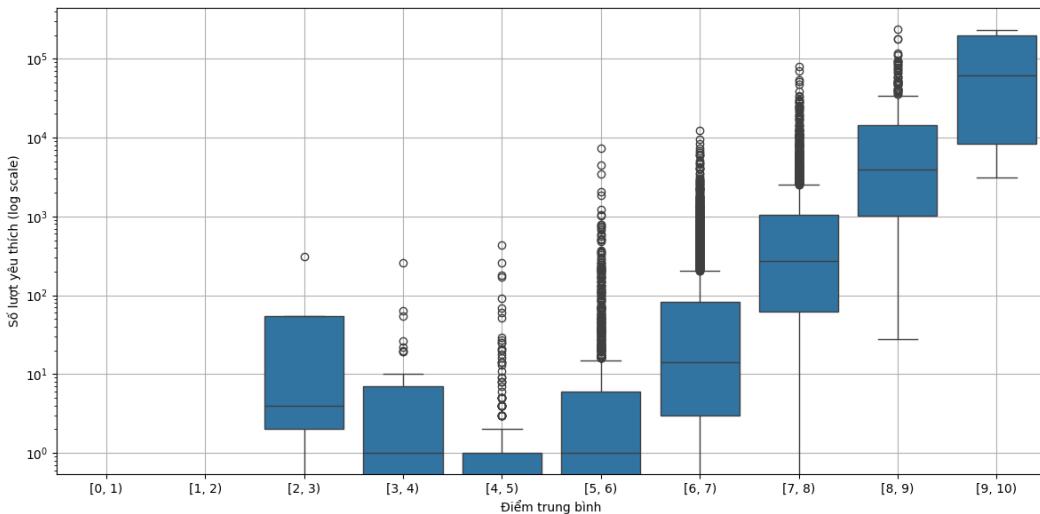
Hình 2.15: Phân phối điểm đánh giá trung bình

Biểu đồ 2.15 cho thấy phân phối điểm đánh giá trung bình có dạng chuẩn, đỉnh phân bố tập trung tại mức 6.5-7 điểm. Đường KDE gần như là hình chuông, đối xứng quanh trung tâm, cho thấy phần lớn phim đạt mức gần 7. Thực tế, người dùng thường cho điểm 6-7 cho phim mức trung bình thay vì 5, lý giải cho đỉnh phân phối lệch sang phải trên thang 0-10.



Hình 2.16: Số lượng người xem theo điểm

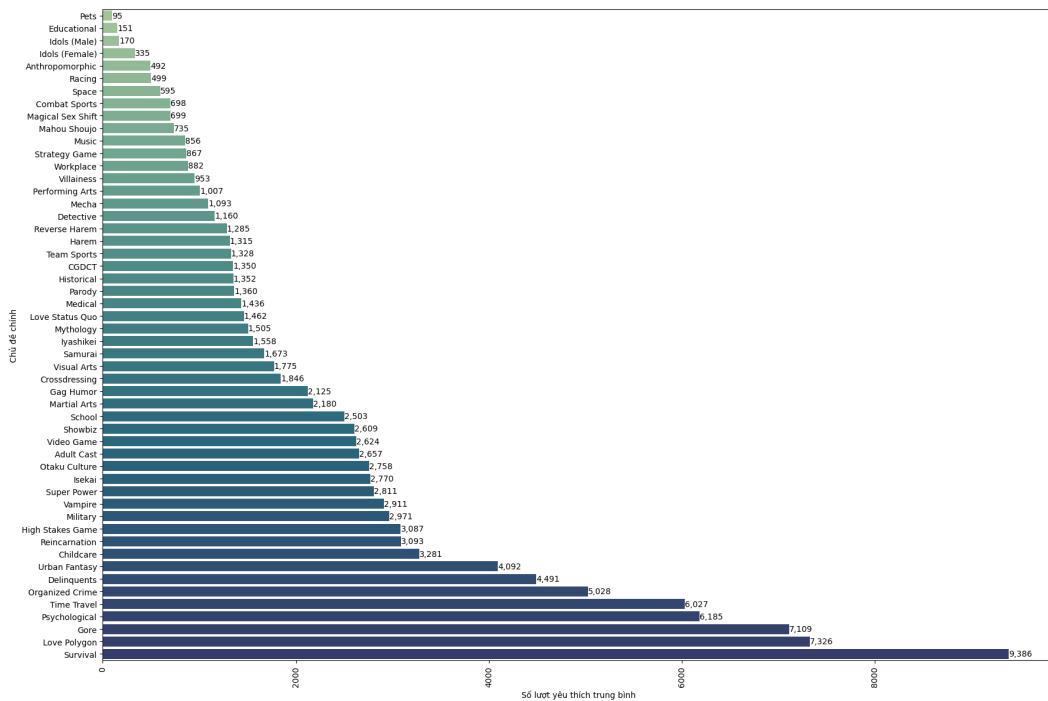
Biểu đồ 2.16 cho thấy số lượng người xem quan hệ chặt chẽ với đánh giá trung bình của phim. Đánh giá phim càng tốt thì càng nhiều người xem.



Hình 2.17: Số lượt yêu thích theo điểm

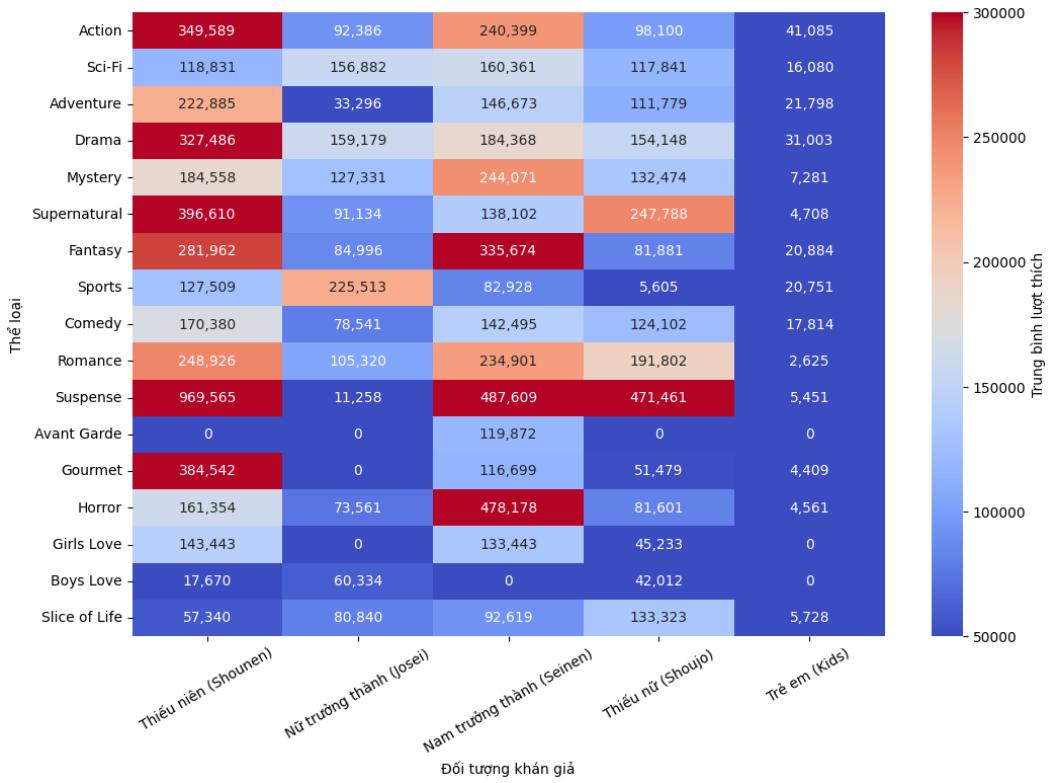
Biểu đồ 2.17 cho thấy số lượng liên hệ tích cực với điểm trung bình. Các boxplot lượt yêu thích ở mức điểm cao có trung vị cao hơn và khoảng tứ vị phân位 cao hơn hẳn phần điểm thấp hơn. Tuy nhiên, có trường hợp ngoại lệ ở phần điểm thấp nhất lại có lượt yêu thích tương tự với phần điểm cho thấy phim ở mức trung bình 6-7. Điều này có thể giải thích bằng hiện tượng một số người sử dụng 'xem vì ghét' (hate-watching), hoặc phim 'dở quá đến mức hay' đối với một số người dùng, nên các phim được đánh giá rất thấp này có được một lượng yêu thích nhất định. Còn các

phim chỉ dở bình thường, dù có đánh giá cao hơn nhưng không bắt được thị hiếu.



Hình 2.18: Số lượt yêu thích trung bình các chủ đề chính

Biểu đồ 2.18 thể hiện số lượt yêu thích trung bình giữa các chủ đề chính. Các chủ đề như 'Survival', 'Psychological' hay 'Time Travel' được yêu thích nhiều, cho thấy người xem chuộng nội dung căng thẳng, kịch tính. Trong khi đó, các chủ đề như 'Pets', 'Educational' hay 'Idols' ít được quan tâm, phản ánh xu hướng đa số không hứng thú với nội dung nhẹ nhàng.



Hình 2.19: Số lượt yêu thích trung bình các thể loại/đối tượng khán giả

Biểu đồ 2.19 thể hiện lượt yêu thích trung bình các thể loại phim theo từng nhóm đối tượng khán giả. Có thể thấy với từng nhóm khán giả yêu thích một số thể loại cụ thể, cho nên thể loại phim hợp với đối tượng khán giả hướng đến có ảnh hưởng lớn đến sự thành công của phim. Ở biểu đồ ta cũng thấy được, các phim hướng đối tượng trẻ em có lượt yêu thích rất thấp ở mọi thể loại.

2.1.2.4 Mô hình hóa dữ liệu

2.1.2.4.1 Cấu hình cài đặt

Các mô hình sử dụng:

- **Logistic Classification:**

```
LogisticRegressionCV (
    max_iter=max_iter ,
    Cs=cs ,
)
```

khoảng hypertune tham số

```
list_max_iter = [10, 20, 30, 40, 50, 100, 200, 300, 500]
list_cs = [1, 2, 3, 4, 5, 8, 10]
```

- **Random Forest Classifier:**

```
RandomForestClassifier(
    max_depth=max_depth,
    n_estimators=n_estimators,
    min_samples_leaf=min_samples_leaf,
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 5, 10]
list_n_estimators = [50, 100, 150, 200]
list_min_samples_leaf = [3, 5, 10, 20]
```

- **XGBoost Classifier:**

```
XGBClassifier(
    max_depth=max_depth,
    n_estimators=n_estimators,
    reg_lambda=reg_lambda,
    learning_rate=lr,
    reg_alpha=alpha,
)
```

khoảng hypertune tham số

```
list_max_depth = [ 6, 7, 8, 9]
list_lambda = [0.5, 1, 2]
list_learning_rate = [1, 0.8, 0.5, 0.25]
list_n_estimators = [50, 100, 150, 200]
list_alpha = [0.5, 1, 2]
```

Thuật toán data transform sử dụng: Do dữ liệu có nhiều điểm ngoại lệ, nhóm chọn Robust Scaler để chuẩn hóa dữ liệu.

Chia tập dữ liệu huấn luyện: Cross-validation 5 folds.

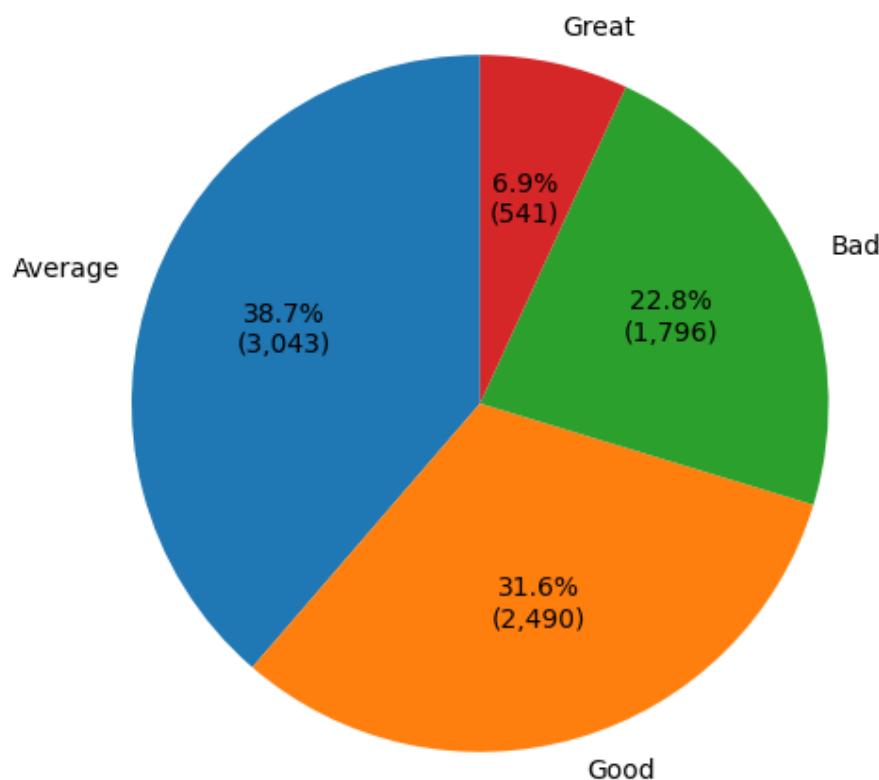
2.1.2.4.2 Phân lớp đánh giá phim

Chuyển cột điểm trung bình giá về dạng nhãn. Dựa vào thống kê dữ liệu bảng 2.9, phân phối điểm đánh giá hình 2.15, lượt yêu thích theo điểm 2.17, có thể chia nhãn điểm như sau

- Dưới 6: Bad - Tệ

- Từ 6-7: Ok - Ôn
- Từ 7-8: Good - Hay
- Trên 8: Great - Tuyệt vời

Phân phối nhãn điểm trên tập dữ liệu như hình 2.20



Hình 2.20: Phân phối nhãn điểm đánh giá

Chọn các đặc trưng huấn luyện: 'genres' (one hot encoding), 'themes' (one hot encoding), 'studios' (category encoding)

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 20
- cs: 5

Bảng 2.10: Kết quả Logistic Classification - Phân lớp đánh giá

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.576740	0.553721	0.452583	0.556177	0.475009
std	0.004452	0.009356	0.014631	0.026914	0.020463
min	0.571998	0.541833	0.434268	0.530762	0.451901
max	0.582357	0.566301	0.473644	0.597646	0.505563

- **Random Forest Classifier:**

Mô hình tốt nhất:

- max_depth: 10
- n_estimators: 150
- min_samples_leaf: 3

Bảng 2.11: Kết quả Random Forest Classifier - Phân lớp đánh giá

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.612567	0.574050	0.428677	0.648379	0.433171
std	0.003059	0.013495	0.016348	0.144203	0.019790
min	0.609367	0.553785	0.407844	0.481355	0.412235
max	0.616343	0.588235	0.451760	0.769657	0.463677

- **XGBoost Classifier:**

Mô hình tốt nhất:

- max_depth: 7
- n_estimators: 200
- reg_lambda: 2
- reg_alpha: 0.5
- learning_rate: 0.5

Bảng 2.12: Kết quả XGBoost Classifier - Phân lớp đánh giá

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.982310	0.707992	0.658002	0.696674	0.673979
std	0.002217	0.013157	0.020672	0.016154	0.019047
min	0.979566	0.691924	0.636567	0.675056	0.652368
max	0.985052	0.722112	0.684336	0.713385	0.697402

So sánh các mô hình:

Bảng 2.13: So sánh kết quả các mô hình (và nghiên cứu liên quan) - Phân lớp đánh giá

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision
Logistic Regression	0.553721	0.475009	0.452583	0.556177
Random Forest	0.574050	0.433171	0.428677	0.648379
XGBoost	0.707992	0.673979	0.658002	0.696674
BBTC, LogReg* [1]	0.60	0.55	0.64	0.48
BoW, NB** [1]	0.5	0.66	0.5	1

(*): *Bag of Words, Bigrams, Trigrams, Character Trigrams + Logistic Regression*

(**): *Bag of Words + Naive Bayes*

Trong bài toán phân lớp đánh giá, Logistic Regression và Random Forest cho kết quả thấp hơn đáng kể, độ chính xác chỉ tầm 0.55, 0.57. Đặc biệt F1 và Recall đều dưới mức 0.5 ở cả 2 mô hình. Cho thấy đây là bài toán phân lớp phức tạp khiến hai mô hình đơn giản hơn này cho kết quả không tốt. Mô hình XGBoost thể hiện rõ rệt sự vượt trội so với các mô hình khác, đạt độ chính xác ≈ 0.708 , f1 ≈ 0.674 , recall 0.658 và precision ≈ 0.697 . Đây là mô hình duy nhất đạt đồng thời kết quả cao ở cả bốn chỉ số, cho thấy một phần hiệu quả trong việc phân lớp. Tuy có độ chính xác trên tập test gần 0.708, bảng 2.12 cho thấy độ chính xác huấn luyện mô hình đến 0.98, cho thấy dữ liệu nhiễu cao và mô hình đang bị overfit. Mô hình XGBoost thu được cũng có các kết quả đa phần cao hơn hai mô hình từ nghiên cứu của Armenta và cộng sự [1].

2.1.2.4.3 Phân lớp đánh giá độ phổ biến

Để xác định độ phổ biến của phim, ta có thể dựa vào cột 'members', số lượng người xem. Chuyển cột 'members' về nhãn, đầu tiên áp dụng hàm logarithm để giảm độ lớn hiện có của dữ liệu, sau đó ta có thể sử dụng các phân vị để tạo 4 nhãn có phân phối bằng nhau.:

```
df_processed[ 'log_members' ] = p.log1p( df_processed[ 'members' ] )

df_processed[ 'members_label_as_int' ] = pd.qcut(
    df_processed[ 'log_members' ],
    q=4,
    labels=[0, 1, 2, 3]
).astype( int )
```

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 100
- cs: 5

Bảng 2.14: Kết quả Logistic Classification - Phân lớp độ phô biến

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.575792	0.546939	0.537928	0.551536	0.541903
std	0.004049	0.008926	0.009028	0.011581	0.008809
min	0.570752	0.535394	0.528976	0.534441	0.533827
max	0.581216	0.558765	0.547981	0.567014	0.554310

- **Random Forest Classifier:**

Mô hình tốt nhất:

- max_depth: 10
- n_estimators: 200
- min_samples_leaf: 3

Bảng 2.15: Kết quả Random Forest Classifier - Phân lớp phô biến

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.615706	0.565880	0.545500	0.595757	0.558836
std	0.002279	0.011968	0.015212	0.009591	0.013481
min	0.612759	0.546813	0.524157	0.582396	0.541538
max	0.618087	0.579262	0.565344	0.608816	0.575794

- **XGBoost Classifier:**

Mô hình tốt nhất:

- max_depth: 8
- n_estimators: 50
- reg_lambda: 2
- reg_alpha: 0.5
- learning_rate: 0.5

Bảng 2.16: Kết quả XGBoost Classifier - Phân lớp độ phô biến

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.886237	0.613711	0.616620	0.626318	0.620354
std	0.002355	0.009508	0.005633	0.014513	0.008927
min	0.882382	0.599202	0.609210	0.604851	0.606661
max	0.888391	0.625498	0.622128	0.642860	0.629061

So sánh các mô hình:

Bảng 2.17: So sánh kết quả các mô hình - Phân lớp độ phô biến

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision
Logistic Regression	0.546939	0.541903	0.537928	0.551536
Random Forest	0.565880	0.558836	0.545500	0.595757
XGBoost	0.613711	0.620354	0.616620	0.626318

Mô hình Logistic Regression và Random Forest tiếp tục cho kết quả thấp hơn đáng kể, độ chính xác chỉ tầm 0.54, 0.56. Tuy nhiên không có chỉ số nào có kết quả dưới 0.5 như bài phân lớp đánh giá trước. Mô hình XGBoost tiếp tục thể hiện rõ rệt sự vượt trội so với các mô hình khác, đạt độ chính xác ≈ 0.61 , f1 0.62 , recall ≈ 0.62 và precision ≈ 0.62 .

2.1.2.4.4 Phân lớp đánh giá độ yêu thích

Để xác định độ yêu thích của phim, ta có thể dựa vào cột 'favorites', số lượng người xem. Chuyển cột 'favorites' về nhãn, đầu tiên áp dụng hàm logarithm để giảm độ lớn hiện có của dữ liệu, sau đó ta có thể sử dụng các phân vị để tạo 4 nhãn có phân phối bằng nhau.:

```
df_processed [ 'favorites' ] = p . log1p ( df_processed [ 'favorites' ] )

df_processed [ 'favorites_label_as_int' ] = pd . qcut (
    df_processed [ 'logFavorites' ] ,
    q=4,
    labels=[0, 1, 2, 3]
) . astype ( int )
```

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 10
- cs: 4

Bảng 2.18: Kết quả Logistic Classification - Phân lớp độ yêu thích

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.541758	0.516247	0.502026	0.506926	0.502540
std	0.004543	0.015478	0.014786	0.014137	0.014253
min	0.536506	0.491036	0.481737	0.484314	0.482091
max	0.546723	0.529880	0.515479	0.519512	0.515302

- **Random Forest Classifier:**

Mô hình tốt nhất:

- max_depth: 10
- n_estimators: 50
- min_samples_leaf: 5

Bảng 2.19: Kết quả Random Forest Classifier - Phân lớp độ yêu thích

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.541061	0.503890	0.464455	0.534843	0.471668
std	0.008845	0.023235	0.021512	0.021478	0.022017
min	0.526408	0.468127	0.434337	0.497847	0.441626
max	0.549327	0.524900	0.485608	0.549807	0.493303

- **XGBoost Classifier:**

Mô hình tốt nhất:

- max_depth: 7
- n_estimators: 200
- reg_lambda: 2
- reg_alpha: 0.5
- learning_rate: 0.25

Bảng 2.20: Kết quả XGBoost Classifier - Phân lớp độ yêu thích

	train_acc	test_acc	test_recall	test_precision	test_f1
mean	0.898395	0.561888	0.560000	0.558003	0.558385
std	0.004875	0.007222	0.012267	0.008956	0.010239
min	0.894121	0.554337	0.544279	0.546286	0.544556
max	0.905331	0.569721	0.577044	0.567922	0.570822

So sánh các mô hình:

Bảng 2.21: So sánh kết quả các mô hình - Phân lớp độ yêu thích

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision
Logistic Regression	0.516247	0.502540	0.502026	0.506926
Random Forest	0.503890	0.471668	0.464455	0.534843
XGBoost	0.561888	0.558385	0.560000	0.558003

Các mô hình đều cho kết quả khá thấp. Mô hình Logistic Regression và Random Forest thấp hơn đáng kể, độ chính xác chỉ tầm 0.51, 0.5. Mô hình Random Forest có F1 và Recal dưới 0.5. Mô hình XGBoost cho kết quả tốt hơn chút ít so với các mô hình khác, đạt độ chính xác 0.56, f1 0.55, recall 0.56 và precision 0.558.

2.1.2.5 Kết luận

Trong nghiên cứu này, nhóm đã phân tích và mô hình hóa phân lớp đánh giá, độ phổ biến và độ yêu thích của các bộ phim bộ dữ liệu MAL dataset. Kết quả cho thấy XGBoost luôn cho kết quả cao nhất, đạt độ chính xác cao nhất ở cả ba bài toán, đặc biệt là phân loại đánh giá với accuracy 0.708 và F1 0.674. So với nghiên cứu liên quan, mô hình thu được là cải tiến. Tuy nhiên, hiện tượng overfitting và kết quả thấp ở bài toán độ yêu thích cho thấy cần cải thiện thêm về tiền xử lý dữ liệu và mô hình hóa, cho thấy các đặc trưng hành vi được lựa chọn có mối quan hệ rõ ràng và có khả năng phân biệt tốt giữa các lớp tính cách.

2.1.3 Extrovert vs. Introvert Behavior Data

2.1.3.1 Giới thiệu bộ dữ liệu

2.1.3.1.1 Nguồn dữ liệu

Bộ dữ liệu được thu thập bởi Rakesh Kapilavayi, viện công nghệ thông tin Vishnu, Ấn Độ, thông qua Google Forms trong dự án nghiên cứu các đặc điểm tính cách và xu hướng hành vi ở sinh viên của anh ta.

2.1.3.1.2 Mô tả dữ liệu

Đây là bộ dữ liệu chứa thông tin khảo sát các sinh viên, mỗi dòng là thông tin liên quan đến đặc điểm hành vi, tính cách do người tham gia khảo sát trả lời.

Ví dụ một phần dữ liệu:

Bộ dữ liệu có 2,900 dòng, bao gồm 8 cột như sau:

Bảng 2.22: Một phần bảng dữ liệu Behavior dataset

Time spent Alone	Stage fear	Social event attendance	Going outside	Drained after socializing	Friends circle size	Post frequency	Personality
4	No	4	6	No	13	5	Extrovert
9	Yes	0	0	Yes	0	3	Introvert
9	Yes	1	2	Yes	5	2	Introvert
0	No	6	7	No	14	8	Extrovert
...

- Kiểu dữ liệu: **Qualitative**
 1. **Stage_fear**: Sợ sân khấu/đứng trước đám đông (Yes/No)
 2. **Drained_after_socializing**: Cảm thấy mệt mỏi sau khi xã giao (Yes/No)
 3. **Personality**: Tính cách (Extrovert, Introvert)
- Kiểu dữ liệu: **Quantitative**
 - **Discrete**:
 4. **Time_spent_Alone**: Thời gian một mình mỗi ngày (0–11)
 5. **Social_event_attendance**: Mức độ tham gia sự kiện xã hội (0–10, người tham gia tự đánh giá từ 0, rất ít, đến 10, rất nhiều)
 6. **Going_outside**: Mức độ ra ngoài (0–7, người tham gia tự đánh giá từ 0, ít khi, đến 7, thường xuyên)
 7. **Friends_circle_size**: Số lượng bạn gần gũi (0-15)
 8. **Post_frequency**: Mức độ đăng bài lên mạng xã hội (ít khi 0–10 thường xuyên)

2.1.3.2 Các nghiên cứu liên quan

Tác giả Rakesh Kapilavayi [17] áp dụng quy trình xử lý dữ liệu làm sạch dữ liệu, phân tích mối quan hệ giữa các đặc trưng đến lựa chọn mô hình học máy phù hợp như Random Forest, XGBoost và SVM. Kết quả thu được không chỉ chứng minh hiệu quả của các thuật toán học máy trong lĩnh vực phân tích nhân cách mà còn mở ra hướng tiếp cận mới trong phát triển các hệ thống hỗ trợ đánh giá tâm lý - nhân sự, giáo dục cá nhân hóa hoặc các nền tảng tương tác người dùng thông minh.

Nghiên cứu [21] tập trung vào việc đánh giá độ tin cậy của bộ cơ sở dữ liệu sàng lọc rối loạn phổ tự kỷ (ASD) trẻ em trên kho dữ liệu UCI bằng cách áp dụng các thuật toán học máy. Các nhà nghiên cứu đã sử dụng bộ dữ liệu sàng lọc ASD trẻ em từ kho dữ liệu UCI, bao gồm 292 trường hợp, và một bộ dữ liệu kiểm nghiệm thực tế gồm 18 trường hợp do các chuyên gia thu thập. Quá trình tiền xử lý dữ liệu bao gồm xóa các trường hợp thiếu dữ liệu và lựa chọn 10 đặc trưng liên quan nhất bằng phương

pháp Chi Square. Nghiên cứu đã xây dựng mô hình dự đoán bằng cách khảo sát bảy thuật toán học máy: SVM, Random Forest, Decision Trees, Logistic Regression, K-Nearest-Neighbors, Naïve Bayes, và Multi Layer Perceptron (MLP), với kỹ thuật xác thực chéo 10 lần (10-fold cross-validation) để nâng cao chất lượng mô hình. Kết quả thử nghiệm cho thấy tất cả bảy thuật toán đều cho kết quả phân loại cao, phù hợp với các nghiên cứu trước đó. Đặc biệt, các thuật toán Random Forest, SVM, Logistic Regression, KNN và MLP đã đạt tỷ lệ dự đoán đúng 100% trên bộ dữ liệu thực tế. Từ những kết quả này, nghiên cứu kết luận rằng bộ dữ liệu phân loại rối loạn phổ tự kỷ trẻ em trên kho dữ liệu UCI là đáng tin cậy và đề xuất sử dụng mô hình thuật toán SVM để phát triển ứng dụng sàng lọc ASD trẻ em.

2.1.3.3 Phân tích dữ liệu

2.1.3.3.1 Thống kê dữ liệu

Bảng 2.23 thể hiện các thông số thống kê trên một số đặc trưng của bộ dữ liệu.

Bảng 2.23: Thống kê dữ liệu một số đặc trưng dữ liệu Behavior dataset

	Post frequency	Friends circle size	Going outside	Social event attendance	Time spent Alone
Mean	3.5647	6.2689	3	3.9634	4.5058
Min	0	0	0	0	0
Q1	1	3	1	2	2
Median	3	5	3	3.9634	4
Q3	6	10	5	6	7
Max	10	15	7	10	11
Mode	2	5	0	2	0
Var	8.3728	17.9127	4.9355	8.2519	11.8417
SD	2.8936	4.2323	2.2216	2.8726	3.4412
CV	0.8117	0.6751	0.7405	0.7248	0.7637
IQR	5	7	4	4	5

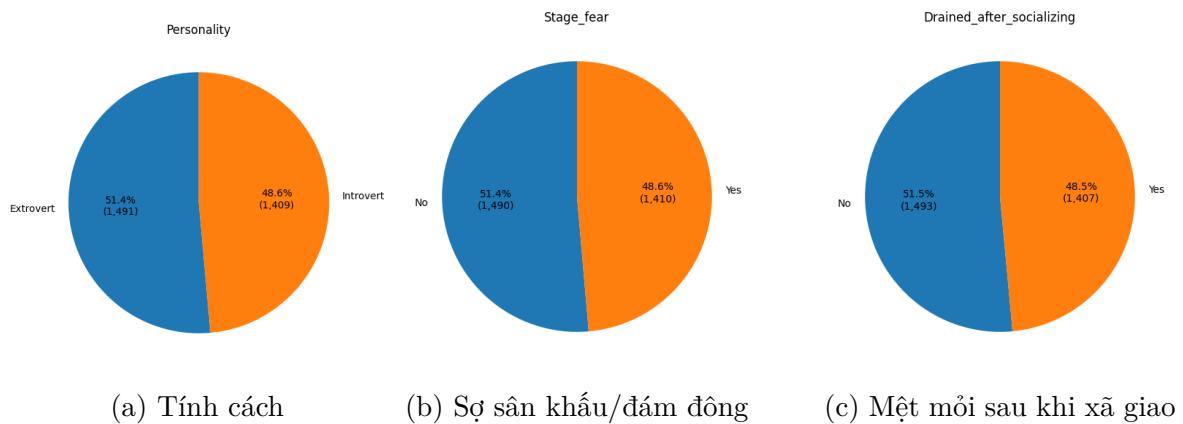
Từ Bảng 2.23, ta có thể rút ra một số nhận xét:

- Post frequency: Trung bình là 3.56 bài, trung vị là 3 và mode là 2, cho thấy tần suất đăng bài khá đồng đều, phần lớn người dùng đăng bài ở mức thấp đến trung bình. Độ lệch chuẩn 2.89 tương đối cao so với trung bình, thể hiện sự phân tán khá lớn, có thể do một số ít người đăng bài rất thường xuyên ($\text{max} = 10$), trong khi một số khác không đăng gì ($\text{min} = 0$)
- Friends circle size: Giá trị trung bình 6.27 và trung vị là 5. Tuy nhiên, giá trị $\text{max} = 15$ và $\text{SD} = 4.23$ phản ánh sự phân tán cao, có thể do một số người có mạng

lưới bạn rất rộng trong khi số khác thì không. Mode = 5 phù hợp với median, cho thấy có tính tập trung cao quanh trung vị.

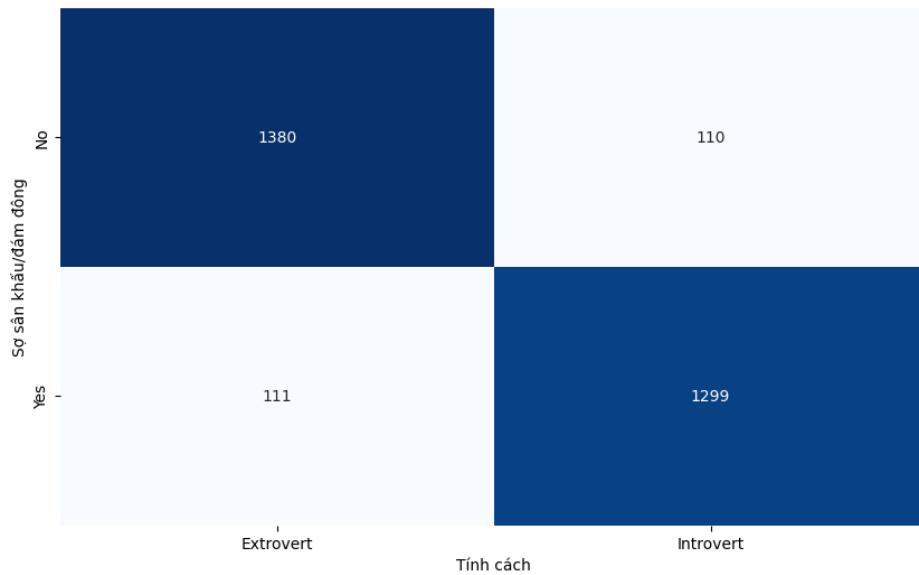
- Going outside: Trung bình và trung vị đều là 3, cho thấy mức độ ra ngoài vừa phải là phổ biến. Mode = 0 cho thấy đa số không tích ra ngoài. Độ lệch chuẩn 2.22 tương đối lớn so với trung bình.
- Social event attendance: Trung bình, trung vị bằng 3.96 và mode = 2. SD = 2.87 và IQR = 4 cho thấy dữ liệu không phân bố rộng.
- Time spent Alone: Trung bình là 4.51, trung vị là 4, và mode = 0, cho thấy có nhiều người dành thời gian một mình mỗi ở mức trung bình đến cao.

2.1.3.3.2 Trực quan hóa dữ liệu



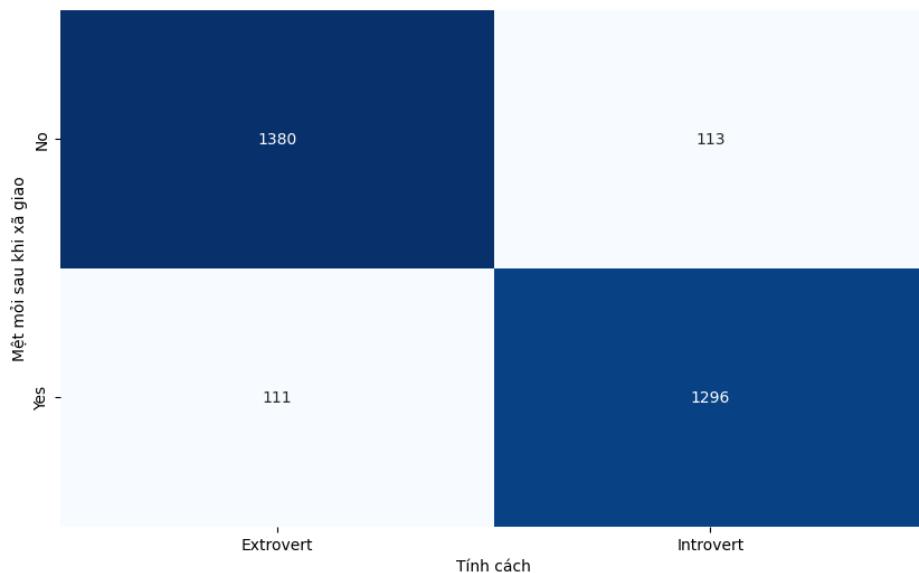
Hình 2.21: Phân phối nhẫn các cột qualitative

Các biểu đồ pie2.21 cho thấy sự phân phối dữ liệu giữa các nhẫn thuộc các cột 'Personality', 'Stage_fear', 'Drained_after_socializing' là gần như cân bằng.



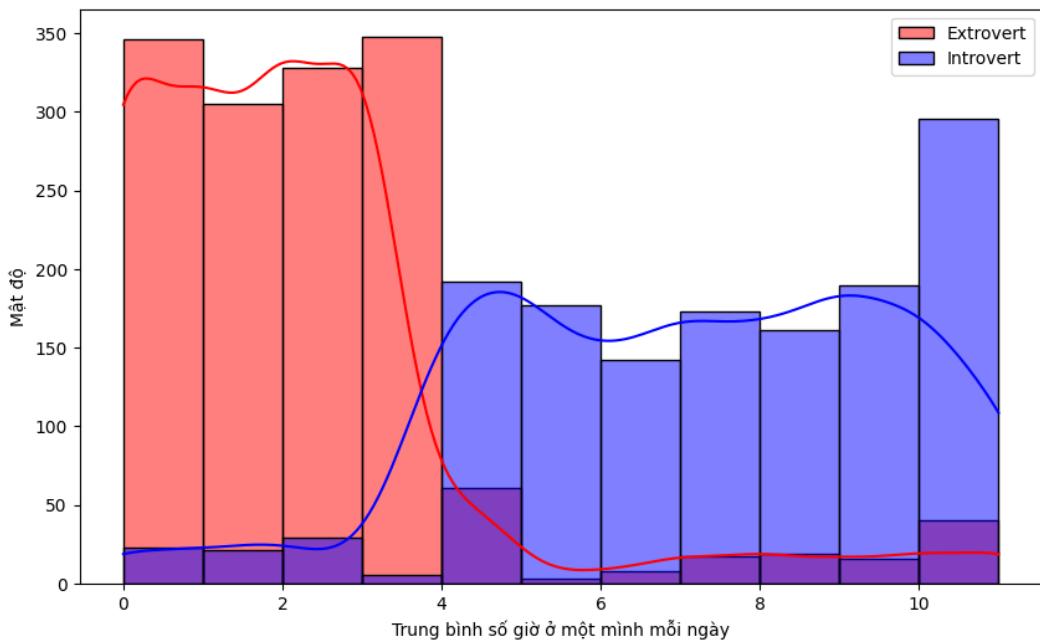
Hình 2.22: Tương quan 'Stage_fear' và 'Personality'

Biểu đồ heatmap 2.22 cho thấy tương quan mạnh giữa 'Stage_fear' và 'Personality'. Đa phần những người không có tính 'Sợ đám đông' đa phần thuộc nhóm tính cách 'Extrovert' và ngược lại.



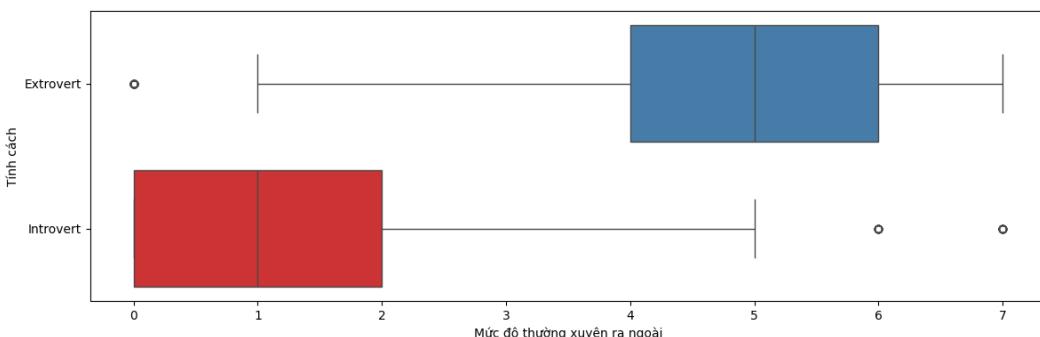
Hình 2.23: Tương quan 'Drained_after_socializing' và 'Personality'

Biểu đồ heatmap 2.23 cho thấy tương quan mạnh giữa 'Drained_after_socializing' và 'Personality'. Đa phần những người không bị 'Mệt mỏi sau khi xã giao' đa phần thuộc nhóm tính cách 'Extrovert' và ngược lại.



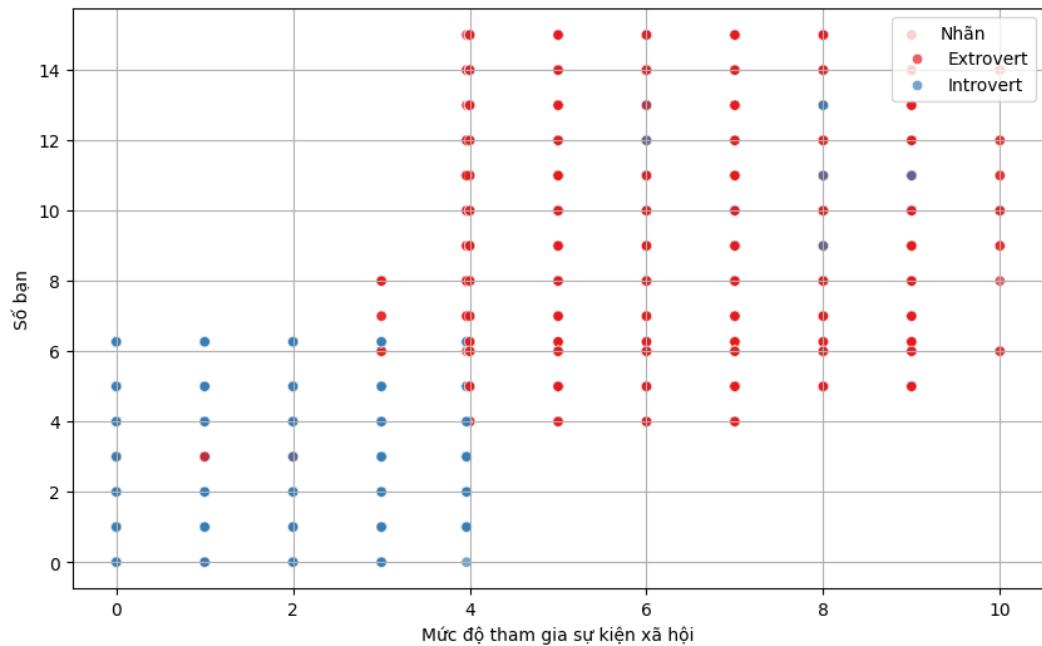
Hình 2.24: Phân phối trung bình số giờ ở một mình theo tính cách

Biểu đồ 2.24 thể hiện sự phân bố thời gian trung bình mỗi ngày một người ở một mình phân loại theo tính cách. Biểu đồ này phản ánh sự khác biệt rõ rệt trong hành vi giữa hai nhóm tính cách. Người hướng ngoại (Extrovert) chủ yếu dành từ 0 đến 4 giờ mỗi ngày để ở một mình. Đường cong mật độ màu đỏ cho thấy xu hướng giảm mạnh sau mốc 4 giờ, cho thấy rất ít người hướng ngoại dành quá nhiều thời gian ở một mình. Ngược lại, người hướng nội (Introvert) lại có xu hướng dành nhiều thời gian hơn ở một mình. Phần lớn nằm trong khoảng từ 5 giờ trở lên mỗi ngày.



Hình 2.25: Phân phối trung bình số giờ ở một mình theo tính cách

Biểu đồ 2.25 phản ánh hai tính cách có hướng hành vi đặc trưng riêng. Người hướng ngoại có mức độ ra ngoài cao hơn rõ rệt, với phần lớn giá trị nằm ở khoản trên 4. Trung vị của nhóm này là 5, cho thấy họ thường xuyên tham gia vào các hoạt động bên ngoài. Trong khi đó, người hướng nội có mức độ ra ngoài thấp hơn đáng kể, với trung vị 1 và phần lớn giá trị tập trung dưới 2 kèm một vài ngoại lệ.



Hình 2.26: Tương quan 'Social_event_attendance' và 'Friends_circle_size' theo nhãn 'Personality'

Biểu đồ 2.26 cho thấy 'Social_event_attendance' và 'Friends_circle_size' có tương quan thuận. Người có 'Số bạn' cao thì 'Mức độ tham gia sự kiện xã hội' cũng tăng theo và ngược lại. Và biểu đồ cũng cho thấy các điểm có thể phân ra thành 2 vùng rõ rệt cho hai nhãn tính cách. Với 'Số bạn' trên 6 và 'Mức độ tham gia sự kiện' trên 5 tạo thành vùng tính cách hướng ngoại, còn lại là tính cách hướng nội, với một số điểm ngoại lệ.

2.1.3.4 Mô hình hóa dữ liệu

2.1.3.4.1 Cấu hình cài đặt

Các mô hình sử dụng:

- **Logistic Classification:**

```
LogisticRegressionCV(
    max_iter=max_iter,
    Cs=cs,
)
```

khoảng hypertune tham số

```
list_max_iter = [10, 20, 30, 40, 50, 100, 200, 300, 500]
list_cs = [1, 2, 3, 4, 5, 8, 10]
```

- **Random Forest Classifier:**

```
RandomForestClassifier(
    max_depth=max_depth,
    n_estimators=n_estimators,
    min_samples_leaf=min_samples_leaf,
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 5, 10]
list_n_estimators = [50, 100, 150, 200]
list_min_samples_leaf = [3, 5, 10, 20]
```

- **XGBoost Classifier:**

```
XGBClassifier(
    max_depth=max_depth,
    n_estimators=n_estimators,
    reg_lambda=reg_lambda,
    learning_rate=lr,
    reg_alpha=alpha,
)
```

khoảng hypertune tham số

```
list_max_depth = [6, 7, 8, 9]
list_lambda = [0.5, 1, 2]
list_learning_rate = [1, 0.8, 0.5, 0.25]
list_n_estimators = [50, 100, 150, 200]
list_alpha = [0.5, 1, 2]
```

Thuật toán data transform sử dụng: Standard Scaler.

Chia tập dữ liệu huấn luyện: Cross-validation 5 folds.

2.1.3.4.2 Phân lớp tính cách 'Personality'

Chọn các đặc trưng huấn luyện: 'Stage_fear', 'Drained_after_socializing', 'Time_spent_Alone', 'Social_event_attendance', 'Going_outside', 'Friends_circle_size', 'Post_frequency'

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 10
- cs: 1

Bảng 2.24: Kết quả Logistic Classification - Phân lớp tính cách

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.934483	0.934483	0.925555	0.945973	0.935549	0.909680
std	0.002859	0.011437	0.018584	0.012504	0.011448	0.021048
min	0.930172	0.922414	0.909396	0.929293	0.923339	0.888679
max	0.937500	0.951724	0.956376	0.961131	0.953177	0.938883

- **Random Forest Classifier:**

Mô hình tốt nhất:

- max_depth: 2
- n_estimators: 50
- min_samples_leaf: 3

Bảng 2.25: Kết quả Random Forest Classifier - Phân lớp tính cách

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.934483	0.934483	0.925555	0.945973	0.935549	0.959291
std	0.002859	0.011437	0.018584	0.012504	0.011448	0.008782
min	0.930172	0.922414	0.909396	0.929293	0.923339	0.945732
max	0.937500	0.951724	0.956376	0.961131	0.953177	0.968442

- **XGBoost Classifier:**

Mô hình tốt nhất:

- max_depth: 6
- n_estimators: 100
- reg_lambda: 0.5
- reg_alpha: 2
- learning_rate: 0.25

Bảng 2.26: Kết quả XGBoost Classifier - Phân lớp tính cách

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.936897	0.933103	0.923539	0.945228	0.934151	0.962129
std	0.001869	0.011200	0.018143	0.013075	0.011241	0.009516
min	0.934914	0.920690	0.906040	0.928814	0.921502	0.947868
max	0.939224	0.946552	0.953020	0.961131	0.948247	0.972851

So sánh các mô hình:

Bảng 2.27: So sánh kết quả các mô hình (và nghiên cứu liên quan) - Phân lớp tính cách

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision	Mean ROC AUC
Logistic Regression	0.934483	0.935549	0.925555	0.945973	0.909680
Random Forest	0.934483	0.935549	0.925555	0.945973	0.959291
XGBoost	0.933103	0.934151	0.923539	0.945228	0.962129
Rakesh Kapilavayi [17]		0.92	0.94	0.94	0.95

Xét tổng thể, cả ba mô hình đều cho kết quả rất tương đồng, với độ chính xác, F1, Recall và Precision gần như giống hệt nhau. Logistic Regression và Random Forest có cùng kết quả trên toàn bộ các chỉ số chính ngoại trừ ROC AUC thì Random Forest cao hơn. XGBoost có chỉ thấp hơn rất ít so với hai mô hình còn lại, nhưng lại đạt giá trị ROC AUC cao nhất (0.962129), cho thấy khả năng phân biệt giữa các lớp tốt hơn.

So với thực hiện trước đó của tác giả [17], các mô hình đều cho kết quả tốt hơn.

2.1.3.4.3 Phân lớp 'Drained after socializing'

Chọn các đặc trưng huấn luyện: 'Stage_fear', 'Time_spent_Alone', 'Social_event_attendance', 'Going_outside', 'Friends_circle_size', 'Post_frequency'

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 10
- cs: 1

Bảng 2.28: Kết quả Logistic Classification - Phân lớp 'Drained after socializing'

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.988276	0.988276	1.000000	0.976432	0.988068	0.989695
std	0.000771	0.003084	0.000000	0.006101	0.003116	0.003183
min	0.987069	0.986207	1.000000	0.972318	0.985965	0.985479
max	0.988793	0.993103	1.000000	0.986014	0.992958	0.993882

- **Random Forest Classifier:**

Mô hình tốt nhất:

-
- max_depth: 2
 - n_estimators: 50
 - min_samples_leaf: 3

Bảng 2.29: Kết quả Random Forest Classifier - Phân lớp 'Drained after socializing'

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.988276	0.988276	1.000000	0.976432	0.988068	0.985857
std	0.000771	0.003084	0.000000	0.006101	0.003116	0.005108
min	0.987069	0.986207	1.000000	0.972318	0.985965	0.979223
max	0.988793	0.993103	1.000000	0.986014	0.992958	0.992890

• **XGBoost Classifier:**

Mô hình tốt nhất:

- max_depth: 6
- n_estimators: 50
- reg_lambda: 1
- reg_alpha: 2
- learning_rate: 1

Bảng 2.30: Kết quả XGBoost Classifier - Phân lớp 'Drained after socializing'

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.988276	0.988276	1.000000	0.976432	0.988068	0.988963
std	0.000771	0.003084	0.000000	0.006101	0.003116	0.002343
min	0.987069	0.986207	1.000000	0.972318	0.985965	0.985783
max	0.988793	0.993103	1.000000	0.986014	0.992958	0.992111

So sánh các mô hình:

Bảng 2.31: So sánh kết quả các mô hình - Phân lớp 'Drained after socializing'

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision	Mean ROC AUC
Logistic Regression	0.988276	0.988068	1.000000	0.976432	0.989695
Random Forest	0.988276	0.988068	1.000000	0.976432	0.985857
XGBoost	0.988276	0.988068	1.000000	0.976432	0.988963

Ba mô hình Logistic Regression, Random Forest và XGBoost cho kết quả giống hệt nhau trên hầu hết các chỉ số và đạt chỉ số cao, cho thấy với các thuộc tính đã chọn, nhãn 'Drained after socializing' Yes và No gần như phân tách nhau. Với Recall 1 ở cả 3, cho thấy tất cả các 'Drained after socializing' = Yes đều được phát hiện chính xác.

2.1.3.4.4 Phân lớp 'Stage fear'

Chọn các đặc trưng huấn luyện: 'Drained_after_socializing', 'Time_spent_Alone', 'Social_event_attendance', 'Going_outside', 'Friends_circle_size', 'Post_frequency'

Kết quả:

- **Logistic Classification:**

Mô hình tốt nhất:

- max_iter: 10
- cs: 1

Bảng 2.32: Kết quả Logistic Classification - Phân lớp 'Stage fear'

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.989310	0.989310	1.000000	0.978531	0.989138	0.991670
std	0.000934	0.003738	0.000000	0.007342	0.003756	0.001367
min	0.988362	0.984483	1.000000	0.969072	0.984293	0.989528
max	0.990517	0.993103	1.000000	0.986014	0.992958	0.993086

- **Random Forest Classifier:**

Mô hình tốt nhất:

- max_depth: 2
- n_estimators: 50
- min_samples_leaf: 3

Bảng 2.33: Kết quả Random Forest Classifier - Phân lớp 'Stage fear'

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.989310	0.989310	1.000000	0.978531	0.989138	0.991813
std	0.000934	0.003738	0.000000	0.007342	0.003756	0.002298
min	0.988362	0.984483	1.000000	0.969072	0.984293	0.988196
max	0.990517	0.993103	1.000000	0.986014	0.992958	0.993675

- **XGBoost Classifier:**

Mô hình tốt nhất:

-
- max_depth: 6
 - n_estimators: 50
 - reg_lambda: 0.5
 - reg_alpha: 2
 - learning_rate: 0.5

Bảng 2.34: Kết quả XGBoost Classifier - Phân lớp 'Stage fear'

	train_acc	test_acc	test_recall	test_precision	test_f1	test_roc_auc
mean	0.989310	0.988621	0.998582	0.978489	0.988427	0.989972
std	0.000934	0.004967	0.003172	0.007412	0.005026	0.002688
min	0.988362	0.981034	0.992908	0.968858	0.980736	0.986292
max	0.990517	0.993103	1.000000	0.986014	0.992958	0.993681

So sánh các mô hình:

Bảng 2.35: So sánh kết quả các mô hình - Phân lớp 'Stage fear'

Model	Mean Test Accuracy	Mean F1	Mean Recall	Mean Precision	Mean ROC AUC
Logistic Regression	0.989310	0.989138	1.000000	0.978531	0.991670
Random Forest	0.989310	0.989138	1.000000	0.978531	0.991813
XGBoost	0.988621	0.988427	0.998582	0.978489	0.989972

Kết quả cả 3 mô hình đều rất cao. Mô hình XGBoost kém hơn một phần rất ít về các chỉ số. Logistic Regression và Random Forest cho kết quả giống hệt nhau trên tất cả các chỉ số, trong đó Recall đạt 1, cho thấy toàn bộ các mẫu 'Stage fear' = Yes đều được phát hiện chính xác. Các mô hình đều đạt chỉ số cao, cho thấy với các thuộc tính đã chọn, nhãn 'Stage fear' Yes và No gần như phân tách nhau.

2.1.3.5 Kết luận

Trong phần này, nhóm đã tiến hành phân tích và xây dựng mô hình phân loại trên bộ dữ liệu khảo sát hành vi và đặc điểm tính cách. Qua mô hình hóa các nhãn tính cách và đặc điểm hành vi ('personality', 'Drained after socializing', 'stage fear'), các mô hình sau huấn luyện đều cho kết quả rất tốt trên cả ba bài toán phân loại, với độ chính xác và các chỉ số đánh giá cao và ổn định.

2.2 Dữ liệu dạng chuỗi thời gian

2.2.1 Daily Climate Delhi

2.2.1.1 Giới thiệu bộ dữ liệu

2.2.1.1.1 Nguồn dữ liệu

Bộ dữ liệu là bộ dữ liệu dự báo thời tiết Ấn Độ, công khai cho khóa học phân tích dữ liệu đại học PES, Bangalore, nguồn từ Weather Undergroud API.

2.2.1.1.2 Mô tả dữ liệu Mỗi dòng dữ liệu chứa thông tin thời tiết một ngày tại thành phố Delhi, Ấn Độ. Khoảng dữ liệu từ ngày 01/01/2013 đến ngày 01/01/2017.

Ví dụ một phần dữ liệu:

Bảng 2.36: Một phần bảng dữ liệu Daily Climate Delhi

date	meantemp	humidity	wind_speed	meanpressure
2013-01-01	10.000000	84.500000	0.000000	1015.666667
2013-01-02	7.400000	92.000000	2.980000	1017.800000
2013-01-03	7.166667	87.000000	4.633333	1018.666667
2013-01-04	8.666667	71.333333	1.233333	1017.166667
...

Bộ dữ liệu có 1462 dòng, bao gồm 5 cột như sau:

- Kiểu dữ liệu: **Quantitative**
 - **Continuous:**
 3. **date:** Ngày
 4. **meantemp:** Nhiệt độ trung bình ($^{\circ}\text{C}$)
 5. **humidity:** Độ ẩm (%)
 6. **wind_speed:** Tốc độ gió (km/h)
 7. **meanpressure:** Áp suất khí (mbar)

2.2.1.2 Các nghiên cứu liên quan

Yunsu Xiaozi [31] áp dụng mô hình học sâu CNN và LSTM cho việc dự đoán nhiệt độ trung bình. Trong notebook này, mô hình CNN-LSTM được xây dựng để dự báo nhiệt độ trung bình hàng ngày dựa trên dữ liệu quá khứ. Dữ liệu được xử lý bằng kỹ thuật cửa sổ trượt (sliding window), sau đó chuẩn hóa để phục vụ huấn luyện mô hình.

Kiến trúc kết hợp CNN để trích xuất đặc trưng và LSTM để học quan hệ thời gian cho phép mô hình đạt hiệu quả dự báo tốt, được đánh giá qua chỉ số RMSE.

Các tác giả [9] tập trung vào việc dự báo nhiệt độ và lượng mưa hàng tháng tại Việt Nam, một vấn đề quan trọng trong lĩnh vực nông nghiệp nhằm hỗ trợ người dân lập kế hoạch gieo trồng phù hợp và ứng phó với biến đổi khí hậu. Để giải quyết bài toán này, các tác giả đã đề xuất một mô hình học sâu đa biến bộ nhớ dài-ngắn hạn (Multivariate Long Short-Term Memory - MLSTM), cải tiến từ mạng nơ-ron thông thường để xử lý hiệu quả hơn dữ liệu chuỗi thời gian có nhiều thuộc tính đầu vào. Dữ liệu được sử dụng bao gồm nhiệt độ và lượng mưa trung bình hàng tháng tại Việt Nam từ năm 1901 đến 2015, cùng với dữ liệu thời tiết của ICRISAT từ 1978 đến 2018. Các dữ liệu này trải qua quá trình tiền xử lý như chuyển đổi về dạng chuỗi thời gian theo tuần/tháng và biến đổi thành dữ liệu đa biến đầu vào. Mô hình MLSTM được đánh giá và so sánh hiệu quả với các mô hình dự báo khác như LSTM, MLP (Mạng nơ-ron đa tầng), và SVR (Hồi quy Vector Hỗ trợ) thông qua các độ đo lỗi RMSE (Root Mean Square Error) và MAE (Mean Absolute Error). Kết quả thực nghiệm cho thấy mô hình MLSTM đạt hiệu quả khá tốt, với độ lỗi RMSE trên tập dữ liệu nhiệt độ là 1.311 và MAE là 1.051, tương ứng trên tập dữ liệu lượng mưa là 2.299 và 2.450. Mô hình MLSTM cũng cho kết quả dự báo tốt hơn và có độ lỗi thấp nhất so với các mô hình LSTM, SVR, MLP trên các tập dữ liệu thử nghiệm. Nghiên cứu kết luận rằng phương pháp dự báo nhiệt độ và lượng mưa bằng kỹ thuật học sâu sử dụng mô hình MLSTM là khá chính xác và có thể áp dụng vào hệ thống thực tế để hỗ trợ ngành nông nghiệp. Hướng phát triển trong tương lai bao gồm cải thiện độ chính xác và phát triển công cụ tiện ích cho người dùng cuối.

2.2.1.3 Phân tích dữ liệu

2.2.1.3.1 Thông kê dữ liệu

Bảng 2.37 thể hiện các thông số thống kê trên một số đặc trưng của bộ dữ liệu.

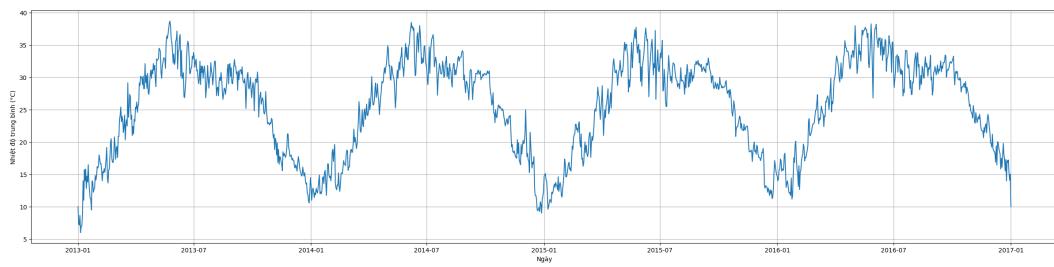
Bảng 2.37: Thống kê dữ liệu một số đặc trưng dữ liệu Daily Climate Delhi

	meantemp	humidity	wind_speed	meanpressure
Mean	25.4955	60.7717	6.8022	1,011.1045
Min	6	13.4286	0	-3.0417
Q1	18.8571	50.375	3.475	1,001.5804
Median	27.7143	62.625	6.2217	1,008.5635
Q3	31.3058	72.2188	9.2382	1,014.9449
Max	38.7143	100	42.22	7,679.3333
Mode	31	65.5	0	1,016
Var	53.9946	281.2212	20.8082	32,483.4543
SD	7.3481	16.7697	4.5616	180.2317
CV	0.2882	0.2759	0.6706	0.1783
IQR	12.4487	21.8438	5.7632	13.3645

Từ đây, ta có thể rút ra một số nhận xét:

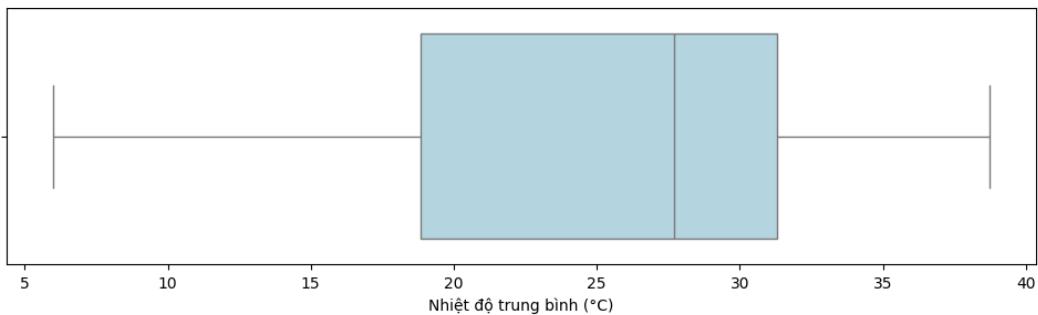
- Meantemp: Trung bình nhiệt độ là 25.5°C , trung vị là 27.7°C và mode là 31°C – cho thấy thời tiết ở Delhi nhìn chung khá nóng. $\text{IQR} = 12.45$ và $\text{SD} = 7.35$ phản ánh mức độ biến thiên tương đối cao giữa các ngày.
- Humidity: Trung bình 60.77 và trung vị 62.63 cho thấy Delhi có độ ẩm tương đối cao. Mode = 65.5 cũng gần với trung vị, cho thấy phân phối khá cân đối. Tuy nhiên, độ lệch chuẩn lên tới 16.77 và $\text{IQR} = 21.84$ cho thấy sự biến động theo mùa đáng kể. Độ ẩm dao động từ mức thấp 13.4 đến tối đa 100, phản ánh thời tiết từ khô nóng và ẩm ướt
- Wind speed: Trung bình 6.80 và trung vị 6.22, với mode = 0 cho thấy có nhiều ngày gió rất nhẹ hoặc gần như không có gió. $\text{Q3}=9.238$ nhưng Max = 42.22 có thể là ngoại lệ, cần chú ý. $\text{SD} = 4.56$ cũng là mức cao so với trung bình, thể hiện sự phân tán lớn.
- Meanpressure: Giá trị trung bình là 1011.1 mbar, trung vị là 1008.56 mbar, đều nằm gần mức áp suất khí quyển tiêu chuẩn (≈ 1013 mbar), cho thấy đa phần dữ liệu hợp lý. Tuy nhiên, giá trị min = -3.04 mbar và max = 7679.33 mbar là bất thường và không thể xảy ra trong thực tế. Áp suất khí quyển trên Trái Đất thường dao động trong khoảng 870–1083 mbar, nên các giá trị này có khả năng là lỗi dữ liệu.

2.2.1.3.2 Trực quan hóa dữ liệu



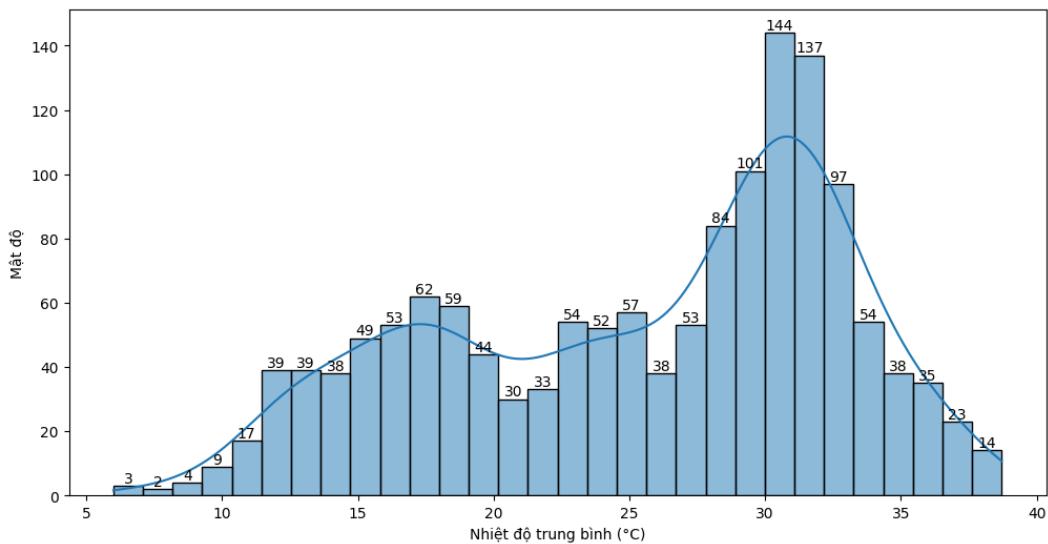
Hình 2.27: Nhiệt độ trung bình từng ngày

Biểu đồ đường 2.27 thể hiện sự thay đổi nhiệt độ trung bình ($^{\circ}\text{C}$) theo thời gian. Từ biểu đồ, thời gian từ đầu năm 2013 đến cuối năm 2016, ta thấy mỗi năm đều có mô hình biến động tương tự, cho thấy tính chất thời vụ mạnh mẽ.



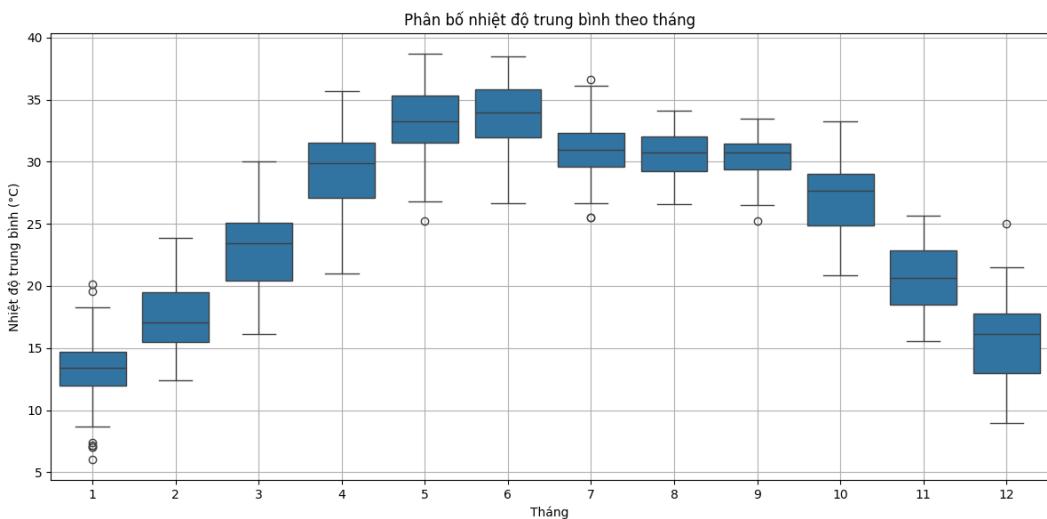
Hình 2.28: Nhiệt độ trung bình

Biểu đồ 2.28 cho thấy phần lớn nhiệt độ nằm trong khoảng từ 19°C đến 32°C , với trung vị khoảng 28°C . Điều này cho thấy khí hậu nhìn chung khá ấm, với nhiệt độ trung bình thường nghiêng về mức cao. Không có điểm ngoại lệ rõ rệt, cho thấy nhiệt độ biến động trong phạm vi tương đối đều và ổn định.



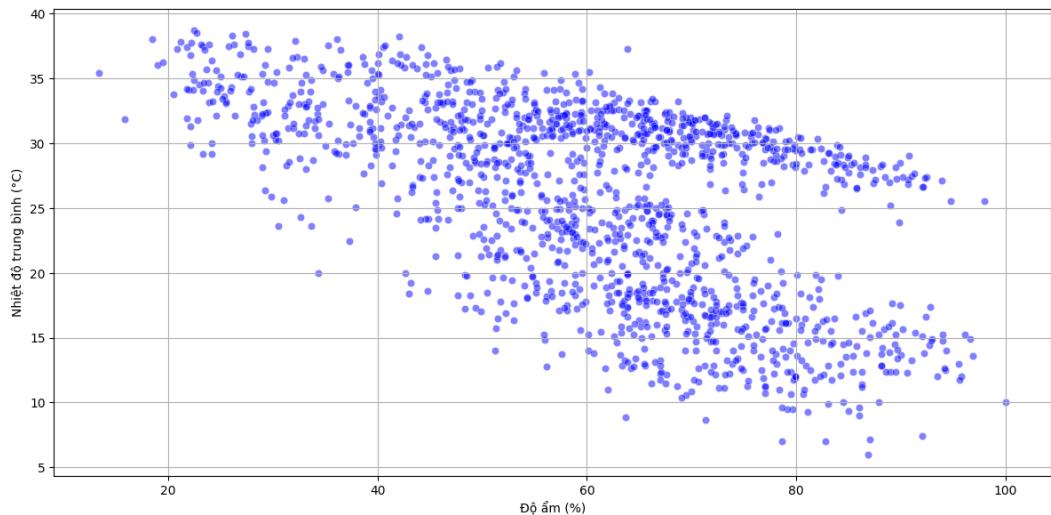
Hình 2.29: Phân phối nhiệt độ trung bình

Biểu đồ 2.29 cho thấy phân phối nhiệt độ trung bình có dạng chuông, lệch trái, cho thấy khí hậu trong dữ liệu thiên về nhiệt độ nóng.



Hình 2.30: Nhiệt độ trung bình các tháng

Biểu đồ 2.30 cho thấy sự phân bố nhiệt độ trung bình theo từng tháng trong năm, phản ánh rõ rệt tính chu kỳ của thời tiết. Từ tháng 1 đến tháng 6, nhiệt độ tăng dần, với đỉnh nằm ở tháng 5 và 6. Sau đó, từ tháng 7 trở đi, nhiệt độ bắt đầu giảm dần, đặc biệt rõ rệt từ tháng 10 đến tháng 12. Sự thay đổi này tạo nên dạng hình vòng cung điển hình cho chu kỳ mùa. Biên độ nhiệt trong các tháng mùa hè (tháng 4 đến 6) và mùa đông (tháng 1, 12) cũng lớn hơn, cho thấy thời tiết trong giai đoạn này biến động mạnh hơn. Ngược lại, các tháng giữa như tháng 8 và 9 ổn định hơn, với hộp hẹp và ít ngoại lệ.



Hình 2.31: Tương quan nhiệt độ - độ ẩm

Biểu đồ 2.31 thể hiện mối tương quan âm giữa hai biến. Mối quan hệ này phản ánh đặc điểm khí hậu phổ biến: khi trời nắng nóng khô thì độ ẩm thấp, còn những ngày mưa hoặc lạnh thường đi kèm độ ẩm cao

2.2.1.4 Mô hình hóa dữ liệu

2.2.1.4.1 Cấu hình cài đặt

Các mô hình sử dụng:

- **Linear Regression:**

```
LinearRegression()
```

- **Random Forest Regressor:**

```
RandomForestRegressor(
    n_estimators=n_estimators,
    max_depth=max_depth,
    min_samples_leaf=min_samples_leaf
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 5, 10]
list_n_estimators = [50, 100, 150, 200]
list_min_samples_leaf = [1, 3, 5, 10]
```

- **XGBoost Classifier:**

```
XGBRegressor(
    n_estimators=n_estimators ,
    max_depth=max_depth ,
    reg_lambda=reg_lambda ,
    learning_rate=learning_rate ,
    reg_alpha=reg_alpha ,
)
```

khoảng hypertune tham số

```
list_max_depth_xgb = [ 6, 7, 8]
list_lambda = [0.5, 1, 2]
list_learning_rate = [0.05, 0.1, 0.5]
list_n_estimators_xgb = [100, 150, 200]
list_alpha = [0.5, 1]
```

Thuật toán data transform sử dụng: Standard Scaler.

Chia tập dữ liệu huấn luyện: Train-Test: 80-20.

2.2.1.4.2 Hồi quy nhiệt độ trung bình ngày 'meantemp'

Chuyển đặc trưng 'meantemp' về dạng sliding window để dự đoán hồi quy, thử nghiệm trên các khoảng: 3,7,14,30

Kết quả:

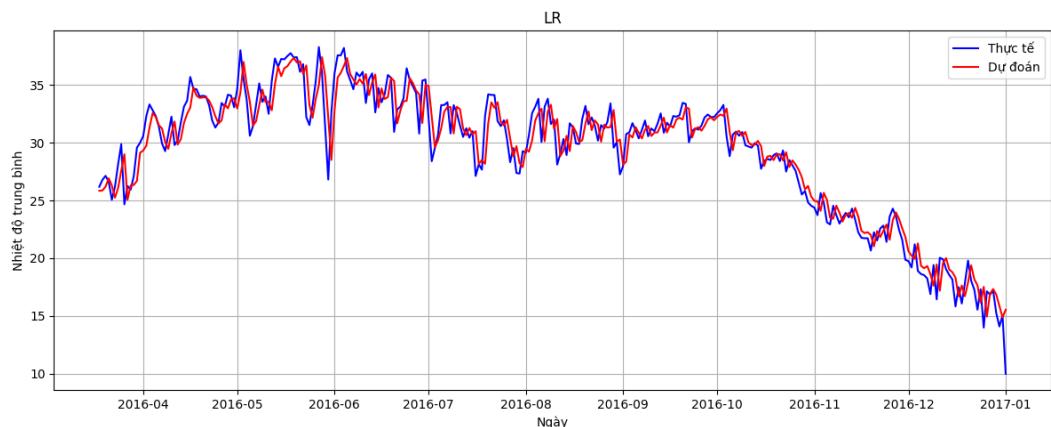
- **Linear Regression:**

Mô hình tốt nhất:

- Khoảng sliding window: 14

Bảng 2.38: Kết quả Linear Regression

	Dataset	MAE	RMSE	MAPE
0	Train	1.206136	1.582341	5.355188
1	Test	1.225678	1.609280	4.539541



Hình 2.32: Kết quả trên tập test Linear Regression

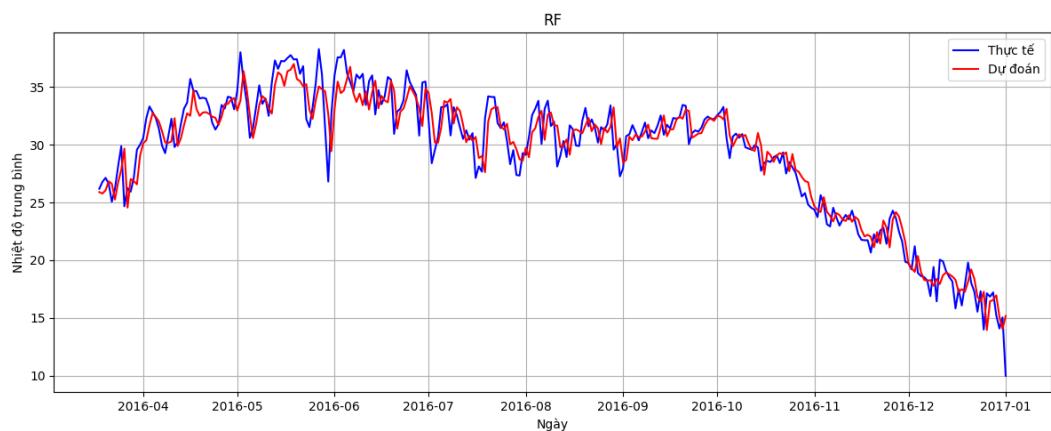
- **Random Forest Regressor:**

Mô hình tốt nhất:

- Khoảng sliding window: 14
- max_depth: 10
- n_estimators: 200
- min_samples_leaf: 5

Bảng 2.39: Kết quả Random Forest Regressor

	Dataset	MAE	RMSE	MAPE
0	Train	0.797399	1.072293	3.490075
1	Test	1.240903	1.602772	4.509175



Hình 2.33: Kết quả trên tập test RF Regressor

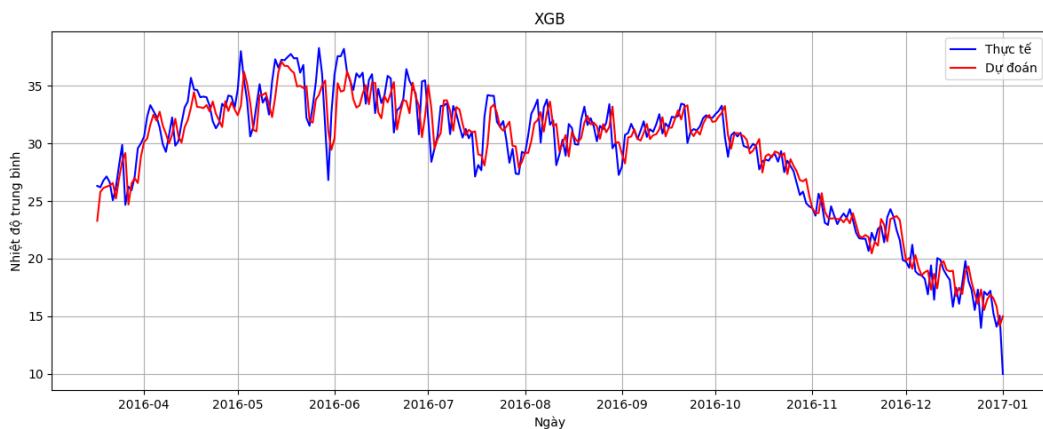
- **XGBoost Regressor:**

Mô hình tốt nhất:

- Khoảng sliding window: 7
- max_depth: 6
- n_estimators: 100
- reg_lambda: 2
- reg_alpha: 0.5
- learning_rate: 0.05

Bảng 2.40: Kết quả XGBoost Regressor

	Dataset	MAE	RMSE	MAPE
0	Train	0.851096	1.107114	3.817686
1	Test	1.301333	1.712462	4.686959



Hình 2.34: Kết quả trên tập test XGB Regressor

So sánh các mô hình:

Bảng 2.41: So sánh kết quả các mô hình (và nghiên cứu liên quan)

Model	MAE	RMSE	MAPE
Linear Regression	1.225678	1.609280	4.539541
Random Forest	1.240903	1.602772	4.509175
XGBoost	1.301333	1.712462	4.686959
CNN LSTM [31]		5.53722	

Random Forest đạt hiệu suất tổng thể tốt nhất với RMSE và MAPE thấp nhất, trong khi Linear Regression cho MAE thấp nhất. XGBoost xếp sau hai mô hình trên nhưng vẫn cho kết quả chấp nhận được. So với nghiên cứu trước sử dụng CNN LSTM [31], các mô hình hiện tại đều vượt trội rõ rệt về độ chính xác, đặc biệt là RMSE, cho thấy mô hình truyền thống có thể phù hợp hơn trong bài toán dự báo nhiệt độ trung bình với dữ liệu này. Mô hình đơn giản cho kết quả tốt hơn có thể do nhiệt độ trung

bình các ngày đã đủ rõ ràng để các mô hình tuyến tính hoặc cây quyết định khai thác hiệu quả mà không cần kiến trúc học sâu. Tương tự đó cũng có thể lý do mô hình XGBoost phức tạp cho kết quả thấp hơn

2.2.1.5 Kết luận

Trong phần này, nhóm đã tiến hành phân tích và xây dựng mô hình hồi quy trên dữ liệu thời tiết thành phố Delhi. Kết quả cho thấy các mô hình truyền thống như Linear Regression và Random Forest hoạt động hiệu quả hơn đáng kể so với mô hình học sâu CNN LSTM từ nghiên cứu trước. Random Forest là mô hình thể hiện tốt nhất tổng thể, đạt RMSE và MAPE thấp nhất trên tập kiểm tra, trong khi Linear Regression cho MAE thấp nhất.

2.2.2 Dữ liệu lượt khám chữa bệnh tỉnh Sóc Trăng

2.2.2.1 Giới thiệu bộ dữ liệu

2.2.2.1.1 Nguồn dữ liệu

Đây là dữ liệu lượt khám chữa bệnh tại các cơ sở y tế tỉnh Sóc Trăng (ST), nhóm tự thu thập từ cơ sở dữ liệu chuyên ngành y tế HOC tỉnh Sóc Trăng

2.2.2.1.2 Mô tả dữ liệu

Mỗi dòng dữ liệu chứa thông tin số lượng lượt khám chữa bệnh mỗi ngày. Khoảng dữ liệu từ ngày 01/01/2021 đến ngày 21/06/2025.

Ví dụ một phần dữ liệu:

Bảng 2.42: Một phần bảng dữ liệu Khám chữa bệnh ST

NGAYVAO	SO_LUONG	E11	E11_9	I10	...
2022-01-01	1201	21	145	246	...
2022-01-02	819	37	74	122	...
2022-01-03	1651	42	147	291	...
2022-01-04	6268	369	186	1741	...
...

Bộ dữ liệu có 1268 dòng, bao gồm 15 cột như sau:

- Kiểu dữ liệu: **Quantitative**

- **Continuous:**

3. **NGAYVAO:** Ngày khám
4. **SO_LUONG:** Tổng số lượt khám
5. **E11:** Lượt khám bị bệnh đái tháo đường không phụ thuộc insuline
6. **E11_9:** Bệnh đái tháo đường không phụ thuộc insuline (Chưa có biến chứng)
7. **I10:** Lượt khám bị bệnh lý tăng huyết áp
8. **J01:** Lượt khám bị bệnh viêm xoang cấp
9. **J02:** Lượt khám bị bệnh viêm họng cấp
10. **J06_9:** Lượt khám bị bệnh nhiễm trùng đường hô hấp trên cấp, không phân loại
11. **K21:** Lượt khám bị bệnh trào ngược dạ dày - thực quản
12. **K21_0:** Lượt khám bị bệnh trào ngược dạ dày - thực quản với viêm thực quản
13. **J20:** Lượt khám bị bệnh viêm phế quản cấp
14. **M25_5:** Lượt khám bị bệnh đau khớp
15. **M54_5:** Lượt khám bị bệnh đau cột sống thắt lưng
16. **R10_4:** Lượt khám bị bệnh đau bụng không xác định và đau bụng khác

2.2.2.2 Phân tích dữ liệu

2.2.2.2.1 Thống kê dữ liệu

Bảng 2.43 thể hiện các thông số thống kê trên một số đặc trưng của bộ dữ liệu.

Bảng 2.43: Thống kê dữ liệu một số đặc trưng dữ liệu Lượt khám chữa bệnh ST

	SO_LUONG	E11	E11_9	I10	J01
Mean	5,637.0662	199.8052	223.5213	883.0639	75.2634
Min	293	0	1	17	0
Q1	2,869	38	171	397	13
Median	6,572.5	239	209	1,014	92.5
Q3	7,384.25	292.25	265	1,187.25	113
Max	11,929	644	686	2,362	174
Mode	6,623	28	195	1,066	5
Var	6,138,540.7312	17,225.6739	6,730.8812	192,124.2856	2,359.326
SD	2,477.6079	131.2466	82.0419	438.3198	48.5729
CV	0.4395	0.6569	0.367	0.4964	0.6454
IQR	4,515.25	254.25	94	790.25	100

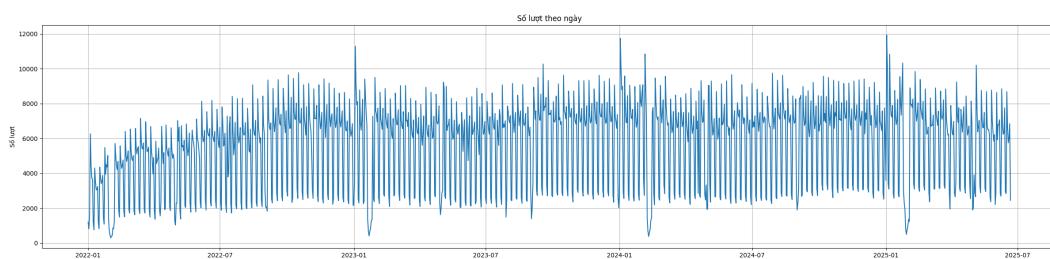
Từ đây, ta có thể rút ra một số nhận xét:

- SO_LUONG: Trung bình là 5637 lượt, trung vị 6572.5 cao hơn cho thấy phân phối hơi lệch trái, tức là có nhiều cơ sở y tế có lượt khám cao. Tuy nhiên, độ lệch

chuẩn khá lớn (2477.6) và $IQR = 4515.25$ phản ánh mức độ chênh lệch lớn giữa các địa phương.

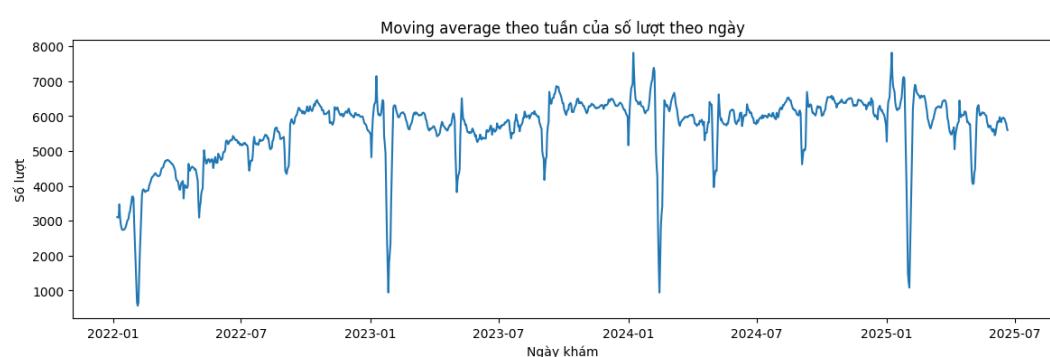
- E11: Trung bình 199.8 ca mỗi đơn vị, trung vị cao hơn một chút (239), nhưng mode lại khá thấp (28) cho thấy một số nơi có số ca rất ít. Độ lệch chuẩn cao, 131.25, cho thấy dữ liệu có biến thiên lớn.
- E11_9: Trung bình là 223.52, median = 209, và mode = 195 cho thấy phân phối tập trung quanh trung bình. Độ lệch chuẩn ở mức trung bình (82.04), và CV thấp hơn (0.367), cho thấy bệnh này được thống kê khá ổn định và đồng đều giữa các đơn vị.
- I10: trung bình 883 lượt mỗi đơn vị. Trung vị (1014) cao hơn trung bình và mode là 1066 cho thấy phân phối hơi lệch trái.
- J01: Trung bình 75.26 thấp. Trung vị = 92.5 và mode = 5 cho thấy phân phối lệch phải. Độ lệch chuẩn khá lớn so với trung bình ($SD = 48.57$, $CV = 0.6454$).

2.2.2.2.2 Trực quan hóa dữ liệu



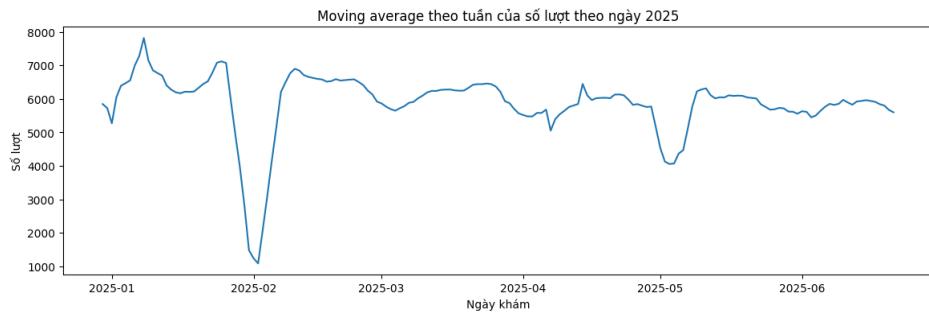
Hình 2.35: Số lượt khám ngày

Biểu đồ 2.35 cho thấy dao động mạnh giữa các ngày, khiến biểu đồ khó đọc. Ta có thể chuyển về trung bình lượt khám mỗi tuần để có biểu đồ dễ xem hơn.

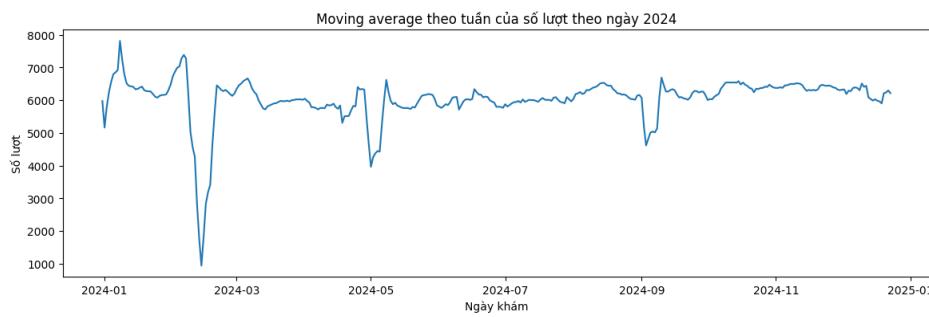


Hình 2.36: Số lượt khám trung bình tuần

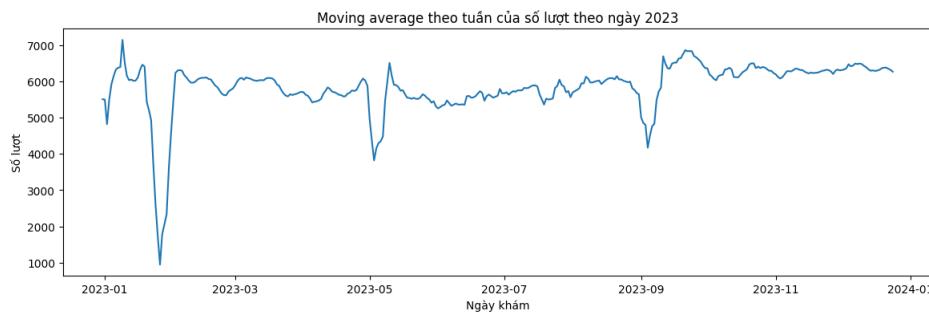
Biểu đồ 2.36 cho thấy lượt khám trung bình theo tuần, đường lượt khám rõ ràng hơn so với ngày. Ta thấy biểu đồ có tính chu kỳ, mỗi năm số lượt khám ổn định khoảng 6000 và có vài mốc giảm đột ngột.



Hình 2.37: Số lượt khám trung bình tuần 2025

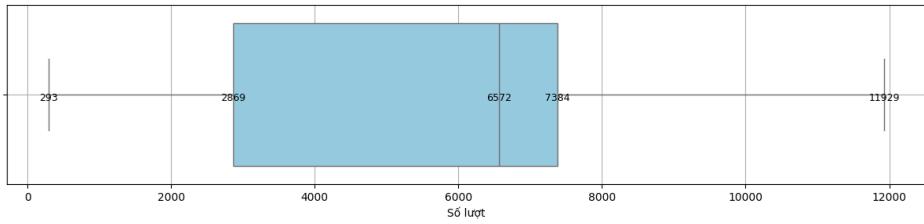


Hình 2.38: Số lượt khám trung bình tuần 2024



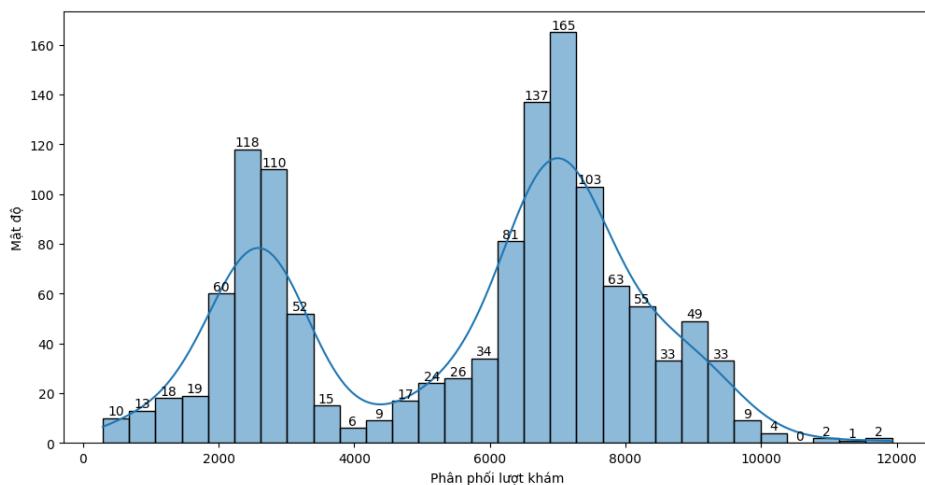
Hình 2.39: Số lượt khám trung bình tuần 2023

Các biểu đồ 2.37, 2.38, 2.39 phóng to số lượt khám trung bình tuần các năm 2025, 2024, 2023. Ta thấy được các đợt giảm đột ngột thường trùng vào dịp lễ có ngày nghỉ dài, cụ thể là Tết, 30/4, 2/9.



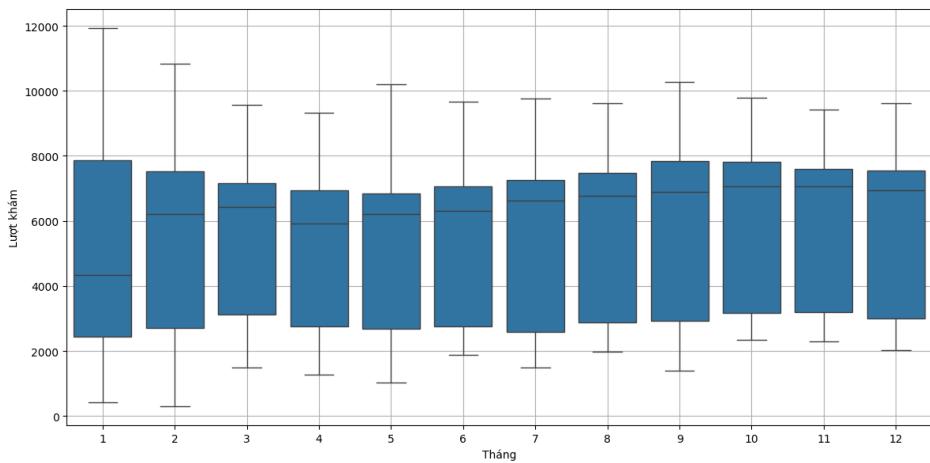
Hình 2.40: Số lượt khám

Biểu đồ 2.40 cho thấy số lượt khám mỗi ngày có sự dao động lớn, nhưng phần lớn tập trung trong khoảng từ gần 3000 đến hơn 7000 lượt. Trung vị đạt khoảng 6572 lượt, phản ánh mức độ hoạt động phổ biến của hệ thống y tế ở ngưỡng cao và ổn định. Tuy nhiên, vẫn tồn tại những ngày có số lượt tăng vọt gần 12000 hoặc giảm sâu xuống dưới 300. Những giá trị này có thể liên quan đến các dịp nghỉ lễ dài



Hình 2.41: Phân phối số lượt khám

Biểu đồ 2.41 thể hiện mật độ hai đỉnh, phân phối số lượt khám mỗi ngày không đồng nhất mà chia thành hai nhóm. Sự xuất hiện của hai đỉnh thể hiện đặc điểm hoạt động không liên tục.



Hình 2.42: Số lượt khám qua các tháng

Biểu đồ 2.42 thể hiện phân bố số lượt khám bệnh theo từng tháng trong năm, phản ánh rõ sự biến động có tính mùa vụ. Tháng 1 có sự dao động mạnh nhất với nhiều giá trị thấp bất thường, có thể do tác động từ kỳ nghỉ Tết kéo dài. Các tháng còn lại ổn định hơn.

2.2.2.3 Mô hình hóa dữ liệu

2.2.2.3.1 Cấu hình cài đặt

Các mô hình sử dụng:

- **Linear Regression:**

```
LinearRegression()
```

- **Random Forest Regressor:**

```
RandomForestRegressor(
    n_estimators=n_estimators,
    max_depth=max_depth,
    min_samples_leaf=min_samples_leaf
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 5, 10]
list_n_estimators = [50, 100, 150, 200]
list_min_samples_leaf = [1, 3, 5, 10]
```

- **XGBoost Classifier:**

```

XGBRegressor(
    n_estimators=n_estimators ,
    max_depth=max_depth ,
    reg_lambda=reg_lambda ,
    learning_rate=learning_rate ,
    reg_alpha=reg_alpha ,
)

```

khoảng hypertune tham số

```

list_max_depth_xgb = [ 6 , 7 , 8]
list_lambda = [0.5 , 1 , 2]
list_learning_rate = [0.05 , 0.1 , 0.5]
list_n_estimators_xgb = [100 , 150 , 200]
list_alpha = [0.5 , 1]

```

Thuật toán data transform sử dụng: Standard Scaler.

Chia tập dữ liệu huấn luyện: Train-Test: 80-20.

2.2.2.3.2 Hồi quy số lượng khám chữa bệnh ngày 'SO_LUONG'

Chuyển đặc trưng 'SO_LUONG' về dạng sliding window để dự đoán hồi quy, thử nghiệm trên các khoảng: 30,90,180,365. Do số lượt khám có thể ảnh hưởng bởi mùa, điều kiện thời tiết,... nên chọn khoảng tháng, quý, nửa năm, năm.

Kết quả:

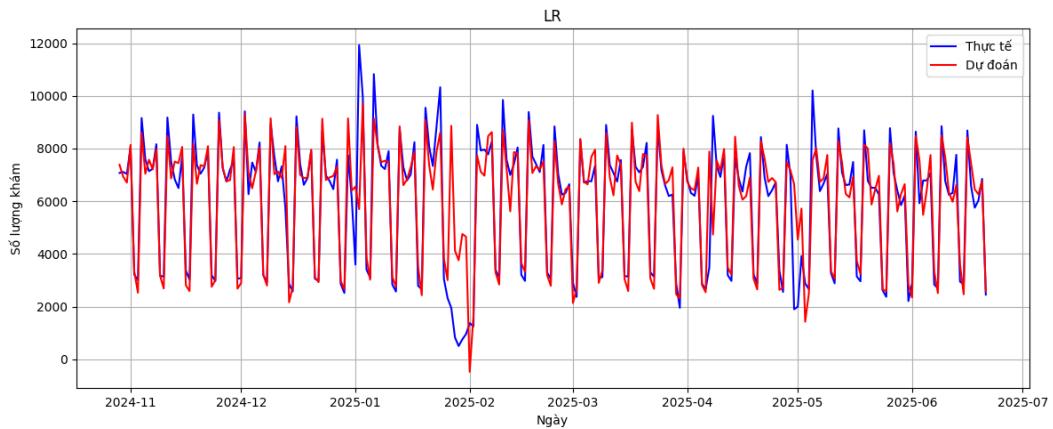
- **Linear Regression:**

Mô hình tốt nhất:

- Khoảng sliding window: 90

Bảng 2.44: Kết quả Linear Regression

	Dataset	MAE	RMSE	MAPE
0	Train	516.283210	978.315505	15.271054
1	Test	634.900687	1128.755081	21.188820



Hình 2.43: Kết quả trên tập test Linear Regression

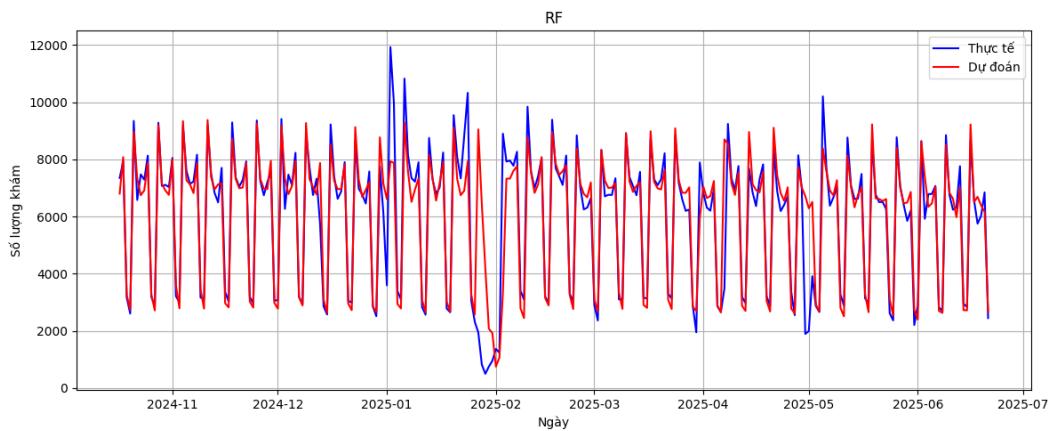
- **Random Forest Regressor:**

Mô hình tốt nhất:

- Khoảng sliding window: 30
- max_depth: 10
- n_estimators: 50
- min_samples_leaf: 3

Bảng 2.45: Kết quả Random Forest Regressor

	Dataset	MAE	RMSE	MAPE
0	Train	262.970338	520.311427	7.967650
1	Test	543.096956	1090.594231	17.977288



Hình 2.44: Kết quả trên tập test RF Regressor

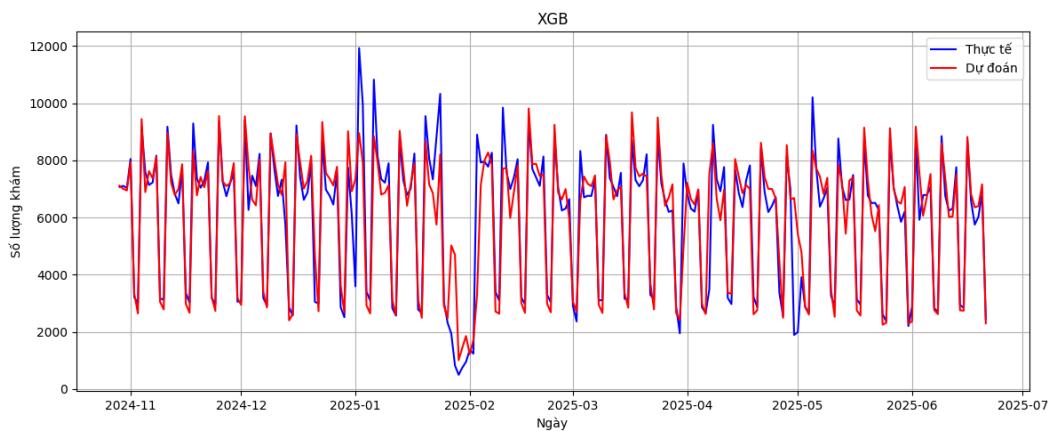
- **XGBoost Regressor:**

Mô hình tốt nhất:

- Khoảng sliding window: 90
- max_depth: 8
- n_estimators: 100
- reg_lambda: 1
- reg_alpha: 0.5
- learning_rate: 0.5

Bảng 2.46: Kết quả XGBoost Regressor

	Dataset	MAE	RMSE	MAPE
0	Train	31.960930	52.874545	0.864103
1	Test	590.553101	989.477861	14.796663



Hình 2.45: Kết quả trên tập test XGB Regressor

So sánh các mô hình:

Bảng 2.47: So sánh kết quả các mô hình

Model	MAE	RMSE	MAPE
Linear Regression	634.900687	1128.755081	21.188820
Random Forest	543.096956	1090.594231	17.977288
XGBoost	590.553101	989.477861	14.796663

Trong ba mô hình được đánh giá, Linear Regression cho thấy hiệu suất kém nhất với các sai số khá lớn trên tập kiểm tra ($MAE = 634.90$, $RMSE = 1128.76$, $MAPE = 21.19\%$), cho thấy mô hình tuyến tính đơn giản này không đủ khả năng nắm bắt các mối quan hệ phức tạp trong dữ liệu chuỗi thời gian. Random Forest Regressor cải thiện đáng kể độ chính xác so với Linear Regression, đặc biệt là ở chỉ số MAE và MAPE (lần lượt là 543.10 và 17.98%). XGBoost Regressor là mô hình cho kết quả tốt nhất về tổng thể, với RMSE thấp nhất (989.48) và MAPE nhỏ nhất (14.80%), mặc dù MAE cao hơn một chút so với Random Forest. Ở cả 3 mô hình (bảng 2.44, 2.45, 2.46), chỉ số lỗi train thấp nhưng test cao, có thể thấy các mô hình đang bị overfit.

2.2.2.4 Kết luận

Trong phần này, nhóm đã tiến hành phân tích và xây dựng mô hình hồi quy dự đoán số lượt khám chữa bệnh các cơ sở y tế tỉnh Sóc Trăng. Kết quả thử nghiệm cho thấy Linear Regression có sai số cao và không phù hợp với dữ liệu chuỗi thời gian. Random Forest cải thiện độ chính xác đáng kể, trong khi XGBoost Regressor đạt hiệu suất tốt nhất với sai số thấp nhất. Tuy nhiên, cả ba mô hình đều cho thấy dấu hiệu overfitting, với sai số tập huấn luyện thấp hơn rõ rệt so với tập test. Nhóm nhận thấy cần cải thiện thêm về đặc trưng đầu vào và kỹ thuật regularization.

2.2.3 Dữ liệu giá chứng khoán Hòa Phát Group

2.2.3.1 Giới thiệu bộ dữ liệu

2.2.3.1.1 Nguồn dữ liệu

Đây là bộ dữ liệu giá cổ phiếu Hòa Phát Group (HPG) nhóm tự thu thập từ api cafef.vn.

2.2.3.1.2 Mô tả dữ liệu Mỗi dòng dữ liệu chứa thông tin giá cổ phiếu HPG. Khoảng dữ liệu từ ngày 15/11/2007 đến ngày 20/06/2025.

Ví dụ một phần dữ liệu:

Bảng 2.48: Một phần bảng dữ liệu giá cổ phiếu HPG

Ngay	GiaDieuChinh	GiaDongCua	GiaMoCua	GiaCaoNhat	GiaThapNhat	...
15/11/2007	2.40	127.00	130.00	130.00	109.00	...
16/11/2007	2.29	121.00	121.00	121.00	121.00	...
19/11/2007	2.17	115.00	115.00	115.00	115.00	...
20/11/2007	2.08	110.00	110.00	110.00	110.00	...
...

Bộ dữ liệu có 4384 dòng, bao gồm 14 cột như sau:

- Kiểu dữ liệu: **Qualitative**
 1. **Stock**: Mã cổ phiếu, HPG
- Kiểu dữ liệu: **Quantitative**
 - **Continuous**:
 2. **Ngay**: Ngày
 3. **GiaDieuChinh**: Giá điều chỉnh

-
4. **GiaDongCua:** Giá đóng cửa
 5. **GiaMoCua:** Giá mở cửa
 6. **GiaCaoNhat:** Giá cao nhất
 7. **GiaCaoNhat:** Giá cao nhất
 8. **GiaThapNhat:** Giá thấp nhất
 9. **ThayDoi:** Giá thay đổi và tỉ lệ thay đổi giữa giá mở - đóng
 10. **GiaThayDoi:** Giá thay đổi (Thay đổi giữa giá mở - đóng)
 11. **ThayDoiPhanTram:** Tỉ lệ thay đổi giữa giá mở - đóng
 12. **KhoiLuongKhopLenh:** Khối lượng cổ phiếu giao dịch
 13. **GiaTriKhopLenh:** Giá tổng số cổ phiếu giao dịch
 14. **KLThoaThuan:** Khối lượng thỏa thuận (khối lượng dự kiến trước giao dịch)
 15. **GtThoaThuan:** Giá trị thỏa thuận (giá trị dự kiến trước giao dịch)

2.2.3.2 Phân tích dữ liệu

2.2.3.2.1 Thống kê dữ liệu

Bảng 2.49 thể hiện các thông số thống kê trên một số đặc trưng của bộ dữ liệu.

Bảng 2.49: Thống kê dữ liệu một số đặc trưng dữ liệu giá chứng khoán HPG

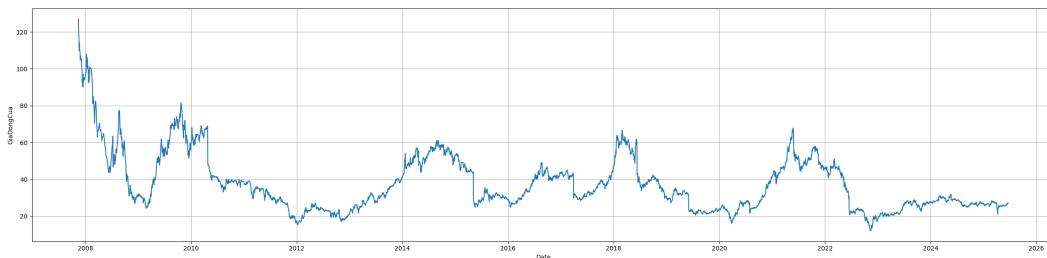
	GiaDieuChinh	GiaDongCua	ThayDoiPhanTram	KhoiLuongKhopLenh
Mean	10.1627	37.4874	0.0599	$8.1 * 10^6$
Min	0.68	12.1	-22.86	0
Q1	1.79	26.45	-1.11	472,235
Median	5.73	32.7	0	2,542,830
Q3	17.035	45.95	1.2425	12,830,875
Max	39.9	127	7	99,658,800
Mode	1.17	23	0	0
Var	104.3475	238.1792	5.284	$137 * 10^{12}$
SD	10.2151	15.4331	2.2987	$11.7 * 10^6$
CV	1.0051	0.4117	38.3627	1.4354
IQR	15.245	19.5	2.3525	12,358,640

Từ đây, ta có thể rút ra một số nhận xét:

- GiaDieuChinh: Giá điều chỉnh có trung bình 10.16 nhưng trung vị chỉ 5.73 và mode là 1.17, cho thấy phần lớn phiên giao dịch có mức giá thấp, trong khi một số ít phiên có giá rất cao kéo trung bình lên. Biến thiên lớn được thể hiện qua độ lệch chuẩn 10.21 và hệ số biến thiên > 1 ($CV = 1.0051$), phản ánh sự phân tán dữ liệu mạnh và phân phối lệch phải rõ rệt.

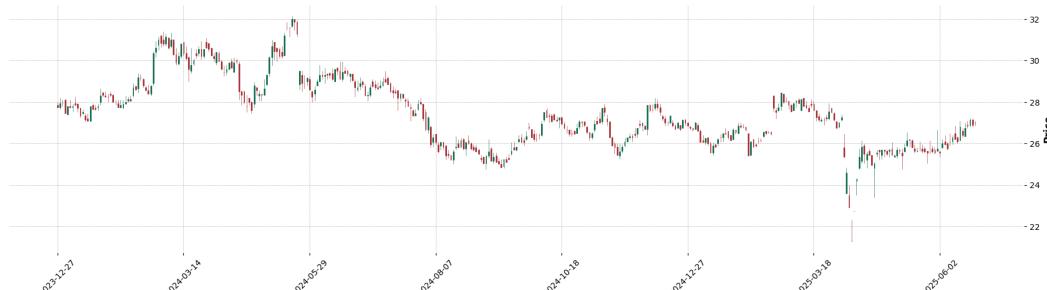
- GiaDongCua: Mức giá đóng cửa có trung bình 37.49, trung vị 32.7 và mode là 23, phản ánh sự chênh lệch giá giữa các giai đoạn. Dữ liệu dao động trong khoảng 12.1 đến 127, với độ lệch chuẩn khá lớn (15.43). Tuy nhiên, hệ số biến thiên ($CV = 0.41$) ở mức vừa phải, cho thấy giá biến động nhưng không quá cực đoan nếu so với giá trị trung bình.
- ThayDoiPhanTram: Mức thay đổi phần trăm theo ngày có trung bình gần 0 (≈ 0.06), cho thấy giá thường đi ngang hoặc biến động nhỏ. Tuy nhiên, giá trị min = -22.86% và max = 7% cho thấy vẫn có các phiên biến động rất mạnh. Mode = 0 càng củng cố việc thị trường thường ít thay đổi trong ngắn hạn.
- KhoiLuongKhopLenh: Trung bình khối lượng khớp lệnh là 8.1 triệu cổ phiếu, nhưng trung vị chỉ khoảng 2.54 triệu và mode bằng 0 cho thấy có rất nhiều phiên giao dịch với thanh khoản thấp, bị kéo lên bởi một số phiên có khối lượng cực lớn (max ≈ 100 triệu). Độ lệch chuẩn cao (≈ 11.7 triệu) và $CV = 1.43$ phản ánh tính chất cực kỳ không ổn định của biến này.

2.2.3.2.2 Trực quan hóa dữ liệu



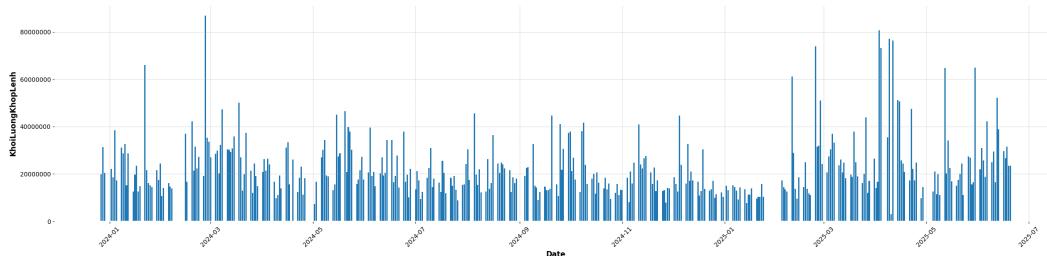
Hình 2.46: Giá đóng cửa theo thời gian

Biểu đồ 2.46 thể hiện biến động giá đóng cửa của cổ phiếu trong giai đoạn dài từ 2007 đến 2025. Giá có xu hướng giảm mạnh ở giai đoạn đầu, sau đó dao động lên xuống theo chu kỳ nhưng không duy trì được xu hướng tăng dài hạn. Trong những năm gần đây, giá có dấu hiệu ổn định hơn và biến động nhẹ quanh một mức nhất định. Nhìn chung, biểu đồ cho thấy cổ phiếu trải qua nhiều biến động lớn, nhưng hiện tại đang trong trạng thái bình ổn.



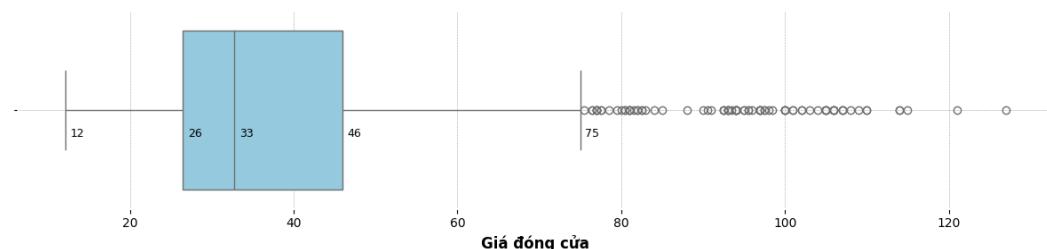
Hình 2.47: Giá hình nến theo thời gian - 365 ngày gần nhất

Biểu đồ 2.47 thể hiện biến động giá cổ phiếu HPG theo cây nến, trong một năm gần đây nhất.



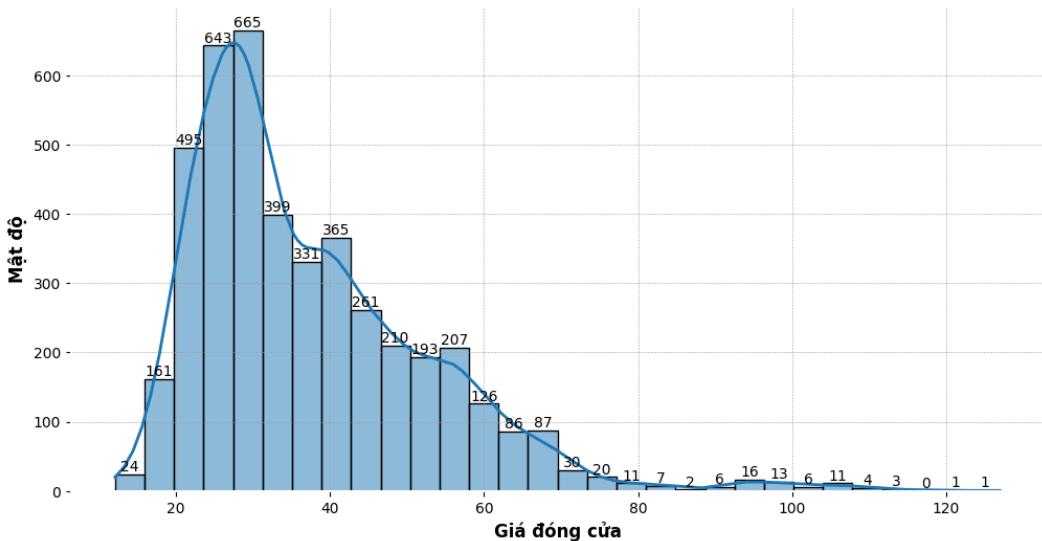
Hình 2.48: Khối lượng khớp lệnh theo thời gian - 365 ngày gần nhất

Biểu đồ 2.48 thể hiện khối lượng khớp lệnh cổ phiếu HPG trong một năm gần đây nhất. Khối lượng giao dịch biến động mạnh, với nhiều đợt tăng vọt xuất hiện rải rác trong toàn bộ giai đoạn. Một số phiên có khối lượng cao, cho thấy những thời điểm giao dịch sôi động bất thường. Có những đoạn không có khối lượng giao dịch là ngày thị trường không giao dịch (ngày nghỉ).



Hình 2.49: Giá đóng cửa

Biểu đồ 2.49 thể hiện phân bố giá đóng cửa của cổ phiếu có trung vị khoảng 33, với phần lớn giá trị nằm trong khoảng từ 26 đến 46. Điều này cho thấy giai đoạn phổ biến nhất của giá cổ phiếu dao động quanh vùng trung bình thấp. Tuy nhiên, bên phải có rất nhiều điểm ngoại lệ, kéo dài đến hơn 120, cho thấy đã từng có nhiều phiên giao dịch với giá cao bất thường. Phân bố lệch phải rõ rệt, tức là phần lớn thời gian cổ phiếu giao dịch ở mức thấp đến trung bình, nhưng có không ít lần tăng vọt lên mức rất cao.



Hình 2.50: Phân phối giá đóng cửa

Biểu đồ 2.50 cho thấy phần lớn cổ phiếu được giao dịch ở mức giá từ khoảng 25 đến 45, với đỉnh cao nhất rơi vào khoảng 28. Phía bên phải của phân phối kéo dài và thưa dần, cho thấy có một số phiên có giá đóng cửa rất cao, thậm chí vượt 100. Tuy nhiên, tần suất những mức giá này rất thấp. Điều này cho thấy phân phối có dạng lệch phải, phản ánh cổ phiếu chủ yếu được giao dịch ở mức giá thấp đến trung bình, và chỉ hiếm khi tăng mạnh lên vùng giá cao.



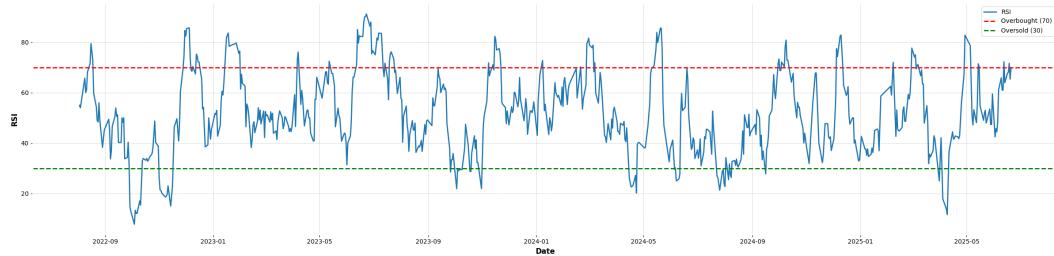
Hình 2.51: Giá đóng cửa 2 năm gần nhất cùng các đường MA

Biểu đồ 2.51 thể hiện vai trò của các đường trung bình động (MA) trong việc phản ánh xu hướng giá theo thời gian. Đường MA20 đại diện cho xu hướng ngắn hạn, MA50 cho trung hạn và MA100, MA200 dài hạn. Giai đoạn đầu, cả MA20, MA50, MA100 và MA200 đều có xu hướng giảm mạnh, đặc biệt là MA100 và MA200 thể hiện rõ xu hướng giảm dài hạn của thị trường. Khi giá bắt đầu phục hồi từ đầu năm 2023, các đường MA ngắn hạn như MA20 và MA50 nhanh chóng điều chỉnh theo, lần lượt cắt lên các đường MA dài hạn, đây là dấu hiệu đảo chiều tích cực trong ngắn hạn. Trong khi đó, MA100 và MA200 mất độ dốc giảm và dần đi ngang, cho thấy đà giảm dài hạn đã chững lại. Từ giữa 2023 đến giữa 2025, các đường MA hội tụ gần nhau và đi ngang, phản ánh giai đoạn tích lũy và thiếu xu hướng rõ ràng. Điều này cho thấy thị trường bước vào trạng thái ổn định, không có lực tăng hay giảm mạnh rõ rệt trong trung và dài hạn.



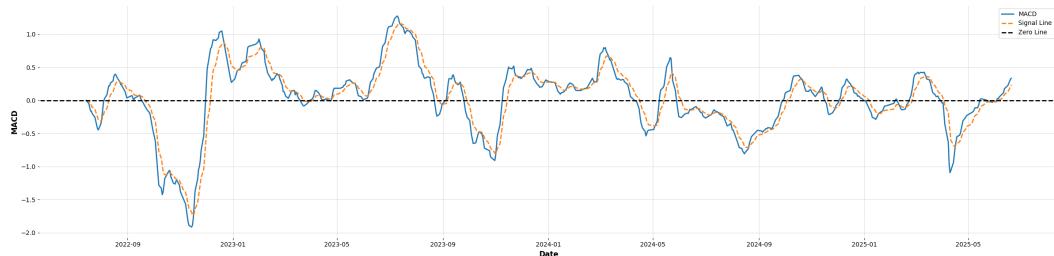
Hình 2.52: Giá đóng cửa 2 năm gần nhất cùng dải Bollinger bands

Biểu đồ 2.52 cho thấy phần lớn thời gian giá dao động bên trong dải, tức là biến động vẫn nằm trong vùng bình thường. Có một số thời điểm vượt ra ngoài dải, như ở cuối năm 2022 và đầu năm 2025. Những lúc này, giá rơi mạnh xuống dưới dải dưới, phản ánh trạng thái quá bán trong ngắn hạn. Ngược lại, một vài giai đoạn như giữa năm 2023 và đầu 2024, giá vượt nhẹ lên trên dải trên, cho thấy thị trường tăng mạnh và có dấu hiệu quá mua. Ở các thời điểm đó, giá đều nhanh chóng quay lại vùng bên trong dải, cho thấy các pha vượt dải chỉ mang tính tạm thời và thường là dấu hiệu cho điều chỉnh hoặc đảo chiều ngắn hạn.



Hình 2.53: RSI giá đóng cửa 2 năm gần nhất

Biểu đồ 2.53 cho thấy đa số thời gian RSI dao động trong khoảng từ 30 đến 70, cho thấy thị trường chủ yếu ở trạng thái cân bằng, không quá nghiêng về một chiều tăng hay giảm. Những lần RSI vượt ngưỡng đều chỉ diễn ra ngắn hạn, sau đó nhanh chóng quay lại vùng trung lập. Điều này cho thấy xu hướng giá mang tính dao động ngắn hạn, chưa có dấu hiệu duy trì lực mua hoặc bán mạnh trong thời gian dài.



Hình 2.54: MACD giá đóng cửa 2 năm gần nhất

Biểu đồ 2.54 thể hiện nhiều biến động của xu hướng giá đóng cửa. Giai đoạn cuối năm 2022 đến đầu 2023, MACD tăng mạnh và vượt ngưỡng 0, cho thấy tín hiệu tích cực từ thị trường. Tuy nhiên, sau đó đường này bắt đầu suy yếu và dao động quanh

mức 0, phản ánh trạng thái thiếu xu hướng rõ ràng. Trong thời gian này, MACD liên tục cắt lên và cắt xuống đường tín hiệu (Signal Line), nhưng phần lớn các giao cắt đều diễn ra trong vùng trung tính, nên không tạo ra tín hiệu mua bán mạnh. Đặc biệt, trong nửa cuối giai đoạn, MACD chủ yếu nằm dưới Signal Line và dưới ngưỡng 0, cho thấy lực mua suy yếu và áp lực bán chiếm ưu thế hơn.

2.2.3.3 Mô hình hóa dữ liệu

2.2.3.3.1 Cấu hình cài đặt

Các mô hình sử dụng:

- **Linear Regression:**

```
LinearRegression()
```

- **Random Forest Regressor:**

```
RandomForestRegressor(
    n_estimators=n_estimators,
    max_depth=max_depth,
    min_samples_leaf=min_samples_leaf
)
```

khoảng hypertune tham số

```
list_max_depth = [2, 3, 5, 10]
list_n_estimators = [50, 100, 150, 200]
list_min_samples_leaf = [1, 3, 5, 10]
```

- **XGBoost Classifier:**

```
XGBRegressor(
    n_estimators=n_estimators,
    max_depth=max_depth,
    reg_lambda=reg_lambda,
    learning_rate=learning_rate,
    reg_alpha=reg_alpha,
)
```

khoảng hypertune tham số

```
list_max_depth_xgb = [ 6, 7, 8]
list_lambda = [0.5, 1, 2]
list_learning_rate = [0.05, 0.1, 0.5]
list_n_estimators_xgb = [100, 150, 200]
list_alpha = [0.5, 1]
```

Thuật toán data transform sử dụng: Standard Scaler.

Chia tập dữ liệu huấn luyện: Train-Test: 80-20.

2.2.3.3.2 Hồi quy giá đóng cửa 'GiaDongCua'

Chuyển đặc trưng 'GiaDongCua' và 'ThayDoiPhanTram' về dạng sliding window để dự đoán hồi quy, thử nghiệm trên các khoảng: 3, 7, 14, 21, 30, 90, 180, 365.

Kết quả:

- **Linear Regression:**

Mô hình tốt nhất:

- Khoảng sliding window: 14

Bảng 2.50: Kết quả Linear Regression

	Dataset	MAE	RMSE	MAPE
0	Train	0.715833	1.224575	1.779761
1	Test	0.413751	0.638391	1.579713



Hình 2.55: Kết quả trên tập test Linear Regression

- **Random Forest Regressor:**

Mô hình tốt nhất:

- Khoảng sliding window: 3
- max_depth: 10
- n_estimators: 50
- min_samples_leaf: 3

Bảng 2.51: Kết quả Random Forest Regressor

	Dataset	MAE	RMSE	MAPE
0	Train	0.525237	0.906095	1.304823
1	Test	0.473422	0.728184	1.908067



Hình 2.56: Kết quả trên tập test RF Regressor

- **XGBoost Regressor:**

Mô hình tốt nhất:

- Khoảng sliding window: 3
- max_depth: 6
- n_estimators: 200
- reg_lambda: 1
- reg_alpha: 0.5
- learning_rate: 0.05

Bảng 2.52: Kết quả XGBoost Regressor

	Dataset	MAE	RMSE	MAPE
0	Train	0.592337	0.918549	1.511546
1	Test	0.491972	0.833300	2.085304



Hình 2.57: Kết quả trên tập test XGB Regressor

So sánh các mô hình:

Bảng 2.53: So sánh kết quả các mô hình

Linear Regression	0.419264	0.641979	1.610565
Random Forest	0.481274	0.745688	1.955963
XGBoost	0.544149	0.915723	2.370569

Trong ba mô hình được đánh giá, Linear Regression cho thấy hiệu suất tốt nhất với các sai số thấp nhất trên tập kiểm tra ($MAE = 0.419$, $RMSE = 0.642$, $MAPE = 1.61$), cho thấy mô hình tuyến tính đơn giản này vẫn có thể nắm bắt tốt xu hướng trong dữ liệu chuỗi thời gian. Random Forest Regressor có độ chính xác thấp hơn so với Linear Regression, đặc biệt là ở chỉ số MAPE cao hơn đáng kể (1.96), mặc dù MAE và RMSE chỉ nhỉnh hơn một chút (lần lượt là 0.481 và 0.746). XGBoost Regressor là mô hình cho hiệu suất kém nhất trong ba phương pháp, với các sai số đều cao hơn, đặc biệt là MAPE (2.37), cho thấy chưa tận dụng được độ phức tạp của mô hình.

2.2.3.4 Kết luận

Trong phần này, nhóm đã thực hiện phân tích và xây dựng các mô hình hồi quy nhằm dự đoán giá đóng cửa cổ phiếu HPG từ dữ liệu lịch sử. Sau khi xử lý dữ liệu và áp dụng kỹ thuật sliding window, ba mô hình Linear Regression, Random Forest Regressor và XGBoost Regressor đã được huấn luyện và so sánh. Kết quả cho thấy Linear Regression đạt hiệu suất tốt nhất trên tập kiểm tra, với sai số thấp hơn rõ rệt so với hai mô hình còn lại, phản ánh mối quan hệ tuyến tính giữa các phiên giao dịch có thể nắm bắt tốt xu hướng giá. Ngược lại, Random Forest và XGBoost tuy phức tạp hơn nhưng không mang lại độ chính xác cao hơn, đặc biệt MAPE lớn hơn đáng kể. Điều này cho thấy trong bối cảnh dự đoán ngắn hạn, mô hình đơn giản và cấu hình hợp lý vẫn có thể mang lại kết quả hiệu quả.

2.3 Dữ liệu dạng ảnh

2.3.1 SIIM-FISABIO-RSNA COVID-19

2.3.1.1 Giới thiệu bộ dữ liệu

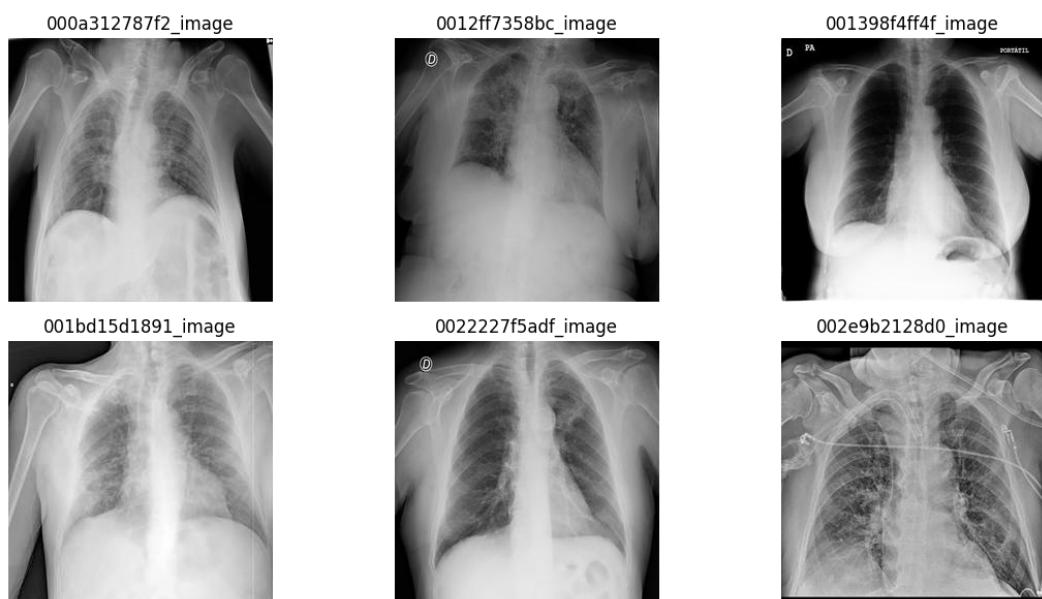
2.3.1.1.1 Nguồn dữ liệu

Bộ dữ liệu là bộ dữ liệu hình ảnh X-quang ngực được gán nhãn nhằm hỗ trợ phát hiện COVID-19, bởi Hội Tin học Hình ảnh trong Y học (Society for Imaging Informatics in Medicine - SIIM) hợp tác với Quỹ thúc đẩy nghiên cứu y sinh và sức khỏe của khu vực Valencia (Foundation for the Promotion of Health and Biomedical Research of Valencia Region - FISABIO) và Hiệp hội X quang Bắc Mỹ (Radiological Society of North America - RSNA)

2.3.1.1.2 Mô tả dữ liệu

Bộ dữ liệu bao gồm 6,334 hình ảnh X-quang kích thước 256x256 kèm file gắn nhãn. Các hình ảnh được gán nhãn khung vùng lạ, hoặc không có nếu không có phát hiện lạ.

Ví dụ một phần dữ liệu:

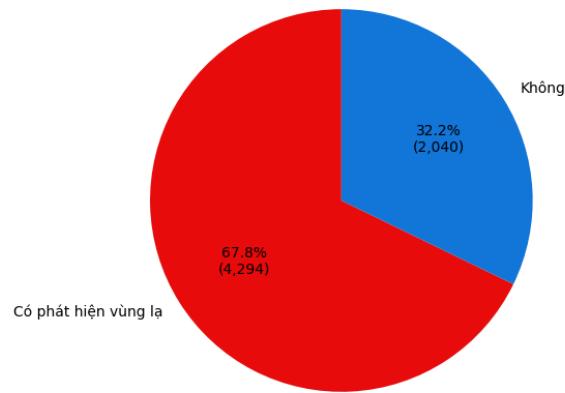


Hình 2.58: Hình ảnh X-quang có phát hiện vùng lạ



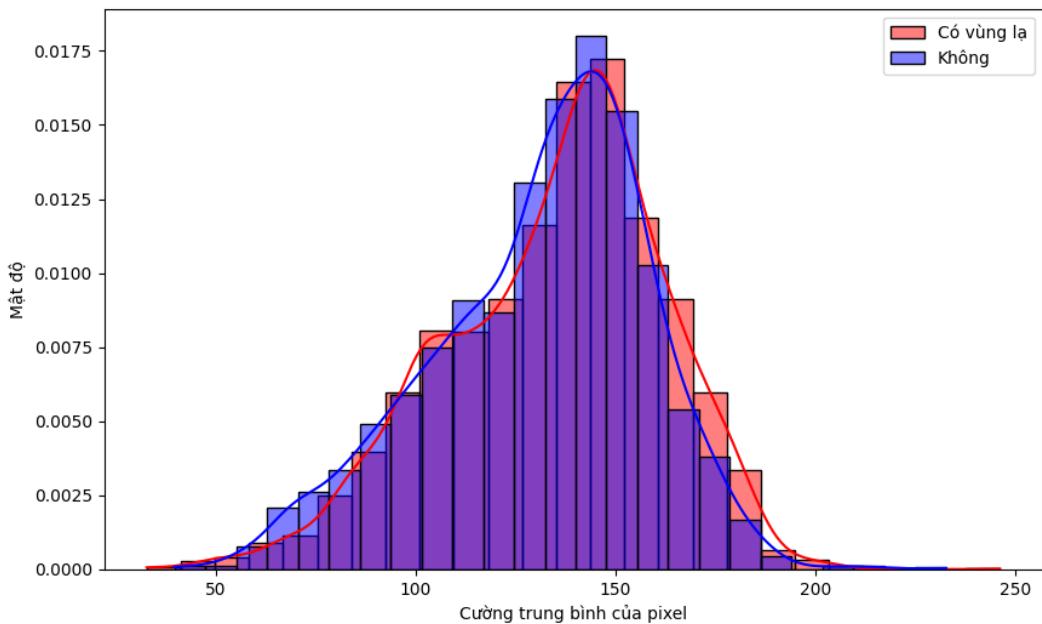
Hình 2.59: Hình ảnh X-quang không phát hiện vùng la

2.3.1.2 Phân tích, trực quan hóa dữ liệu



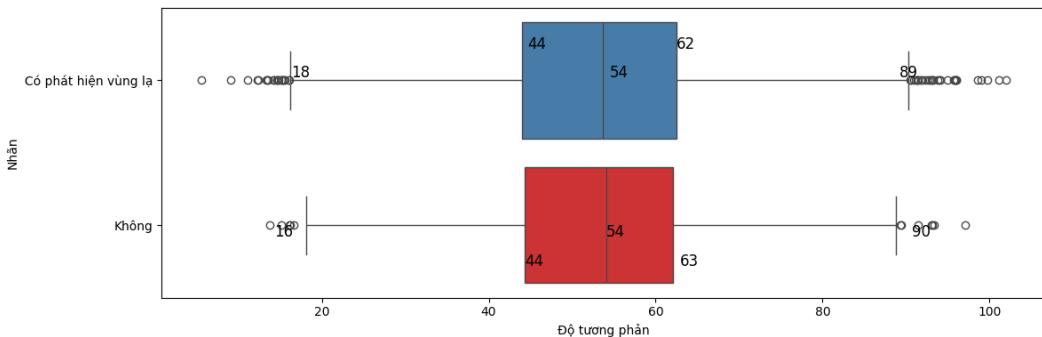
Hình 2.60: Số lượng hình ảnh mỗi nhãn

Biểu đồ pie 2.60 cho thấy sự phân phối dữ liệu giữa hai nhãn. Dữ liệu mất cân bằng với số lượng hình có phát hiện vùng lạ chiếm đến 67.8% tập dữ liệu.



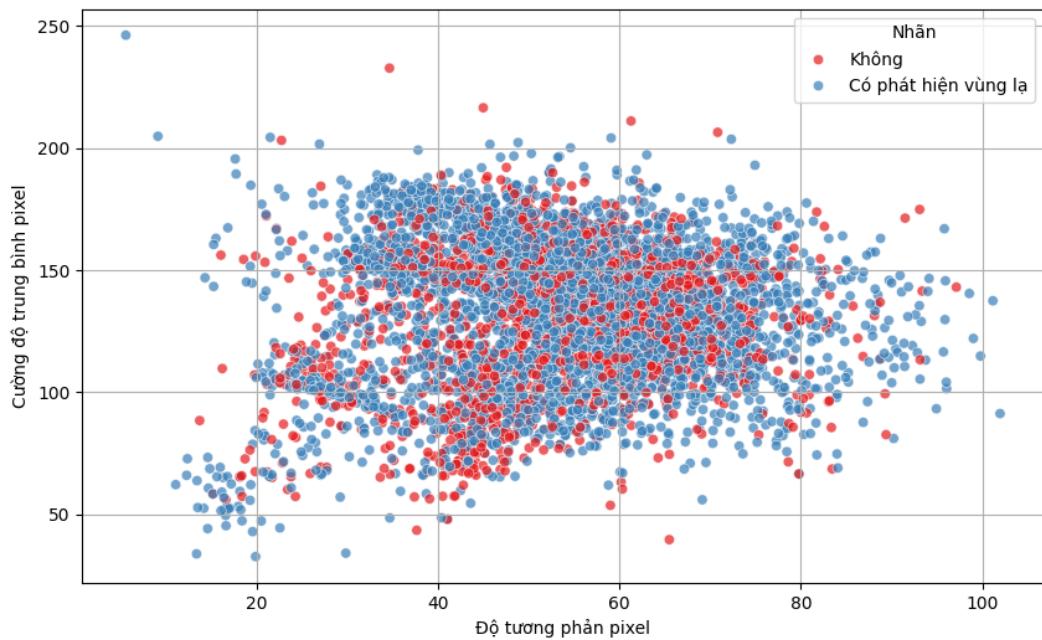
Hình 2.61: Phân phối cường độ pixel trung bình

Biểu đồ 2.61 cho thấy sự phân phối cường độ pixel trung bình giữa hai nhóm ảnh. Cả hai phân phối đều có hình chuông và khá giống nhau. Nhóm ảnh có vùng lạ hơi lệch về cường độ cao hơn một ít, tuy nhiên đó là không đáng kể.



Hình 2.62: Độ tương phản theo nhãn

Biểu đồ 2.62 cho thấy độ tương phản giữa hai loại hình là tương đương nhau, với một số ngoại lệ.



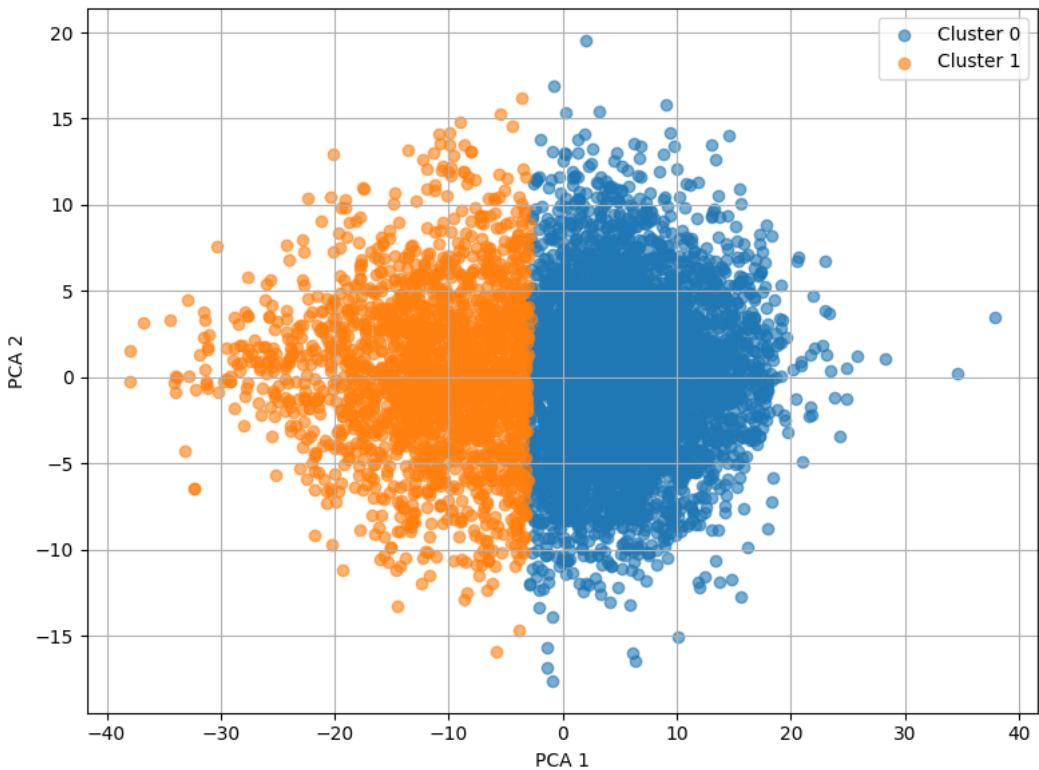
Hình 2.63: Tương quan cường độ và tương phản pixel với loại nhãn

Biểu đồ 2.63 thể hiện mối quan hệ giữa độ tương phản pixel và cường độ trung bình pixel trên hai nhãn dữ liệu. Các điểm giữa hai nhãn không tạo ra phân vùng rõ rệt.

2.3.1.3 Mô hình hóa dữ liệu

Để thực hiện bài toán phân cụm, nhóm sử dụng đặc trưng được tính bằng cách chia ảnh thành các ô vuông nhỏ (patch) có kích thước cố định, sử dụng trung bình độ sáng của mỗi ô để tạo thành một vector đặc trưng.

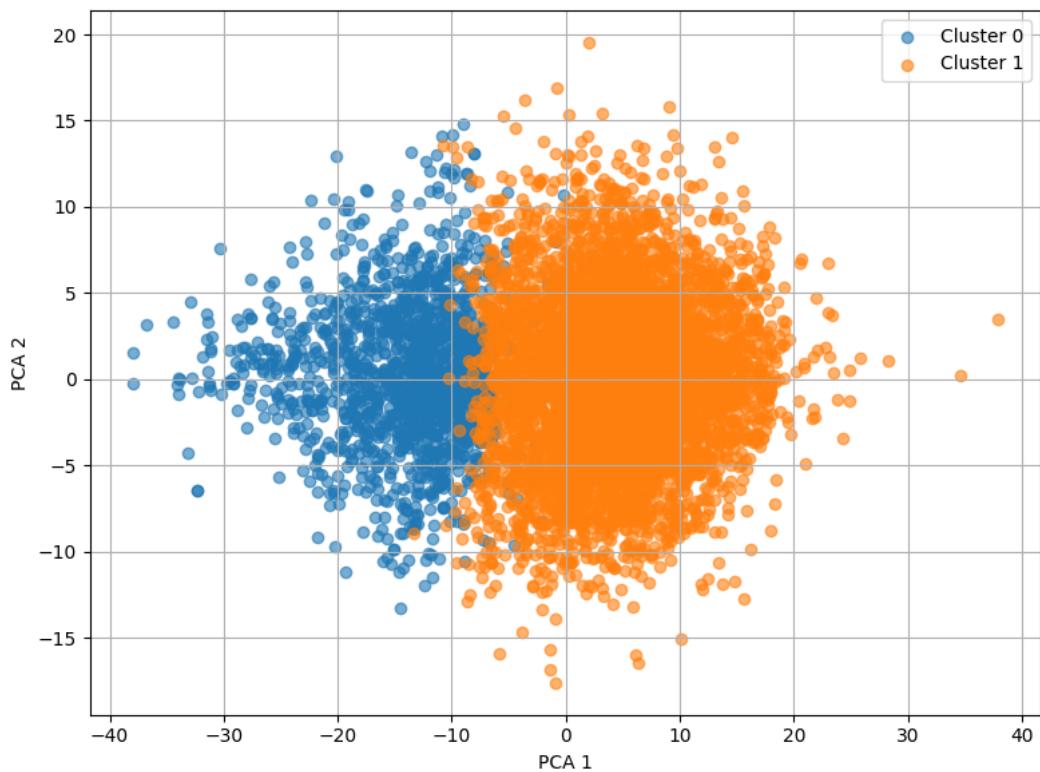
2.3.1.3.1 Phân cụm sử dụng K-means



Hình 2.64: K-Means. ARI = 0.0039

Biểu đồ 2.64 cho thấy kết quả phân cụm ảnh X-quang ngực bằng thuật toán K-means, sử dụng giảm chiều dữ liệu bằng PCA để vẽ trên 2 chiều. Hai cụm được phân tách khá rõ ràng theo chiều ngang. Các điểm dữ liệu trong mỗi cụm khá tập trung và ít bị lẫn với cụm còn lại, cho thấy mô hình đã chia cụm ổn định về mặt hình ảnh. Giá trị ARI chỉ đạt 0.0039, cho thấy kết quả phân cụm gần như không tương quan với nhãn thực tế và hiệu quả phân cụm rất thấp.

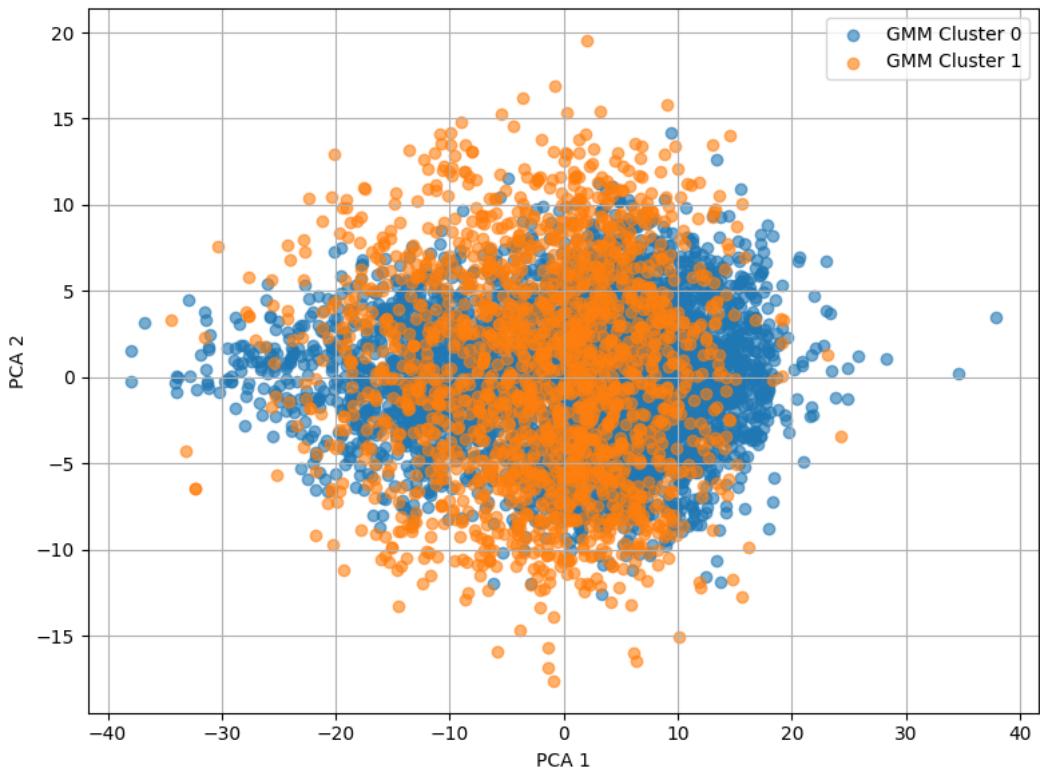
2.3.1.3.2 Phân cụm sử dụng Spectral Clustering



Hình 2.65: Spectral Clustering. ARI = 0.0046

Biểu đồ 2.65 thể hiện kết quả phân cụm ảnh X-quang ngực bằng thuật toán Spectral Clustering sau khi giảm chiều dữ liệu bằng PCA. Các điểm dữ liệu phân bố hai cụm khá đều, nhưng có một số vùng chồng lấn nhẹ ở khu vực trung tâm. Mặc dù hình ảnh cho thấy một sự phân cụm tương đối rõ. ARI chỉ đạt 0.0046, cho thấy kết quả phân cụm chưa tốt, gần như ngẫu nhiên.

2.3.1.3.3 Phân cụm sử dụng Gaussian Mixture



Hình 2.66: Gaussian Mixture. ARI = -0.0095

Biểu đồ thể hiện kết quả phân cụm ảnh X-quang ngực bằng mô hình Gaussian Mixture (GMM) sau khi giảm chiều dữ liệu bằng PCA. Mô hình chia dữ liệu thành 2 cụm đè lên nhau. Chỉ số ARI = -0.0095 cho thấy kết quả phân cụm gần như hoàn toàn không tương quan với nhãn thực tế, thậm chí kém hơn cả phân cụm ngẫu nhiên.

So sánh các mô hình:

Bảng 2.54: So sánh kết quả các mô hình

Model	ARI
K-Means	0.0039
Spectral Clustering	0.0046
Gaussian Mixture	-0.0095

Bảng 2.55 so sánh cho thấy các mô hình phân cụm đều cho kết quả rất thấp, phản ánh khả năng phân tách các cụm gần như không đáng kể. Trong đó, Spectral Clustering đạt chỉ số ARI cao nhất với giá trị 0.0046, nhưng chênh lệch so với K-Means (0.0039) là không đáng kể. Mô hình Gaussian Mixture thậm chí còn cho kết quả âm (-0.0095), cho thấy việc phân cụm của nó còn tệ hơn ngẫu nhiên. Nhìn chung, cả ba mô hình đều không phù hợp để phân tách dữ liệu trong trường hợp này.

2.3.2 Outdoor-fire dataset

2.3.2.1 Giới thiệu bộ dữ liệu

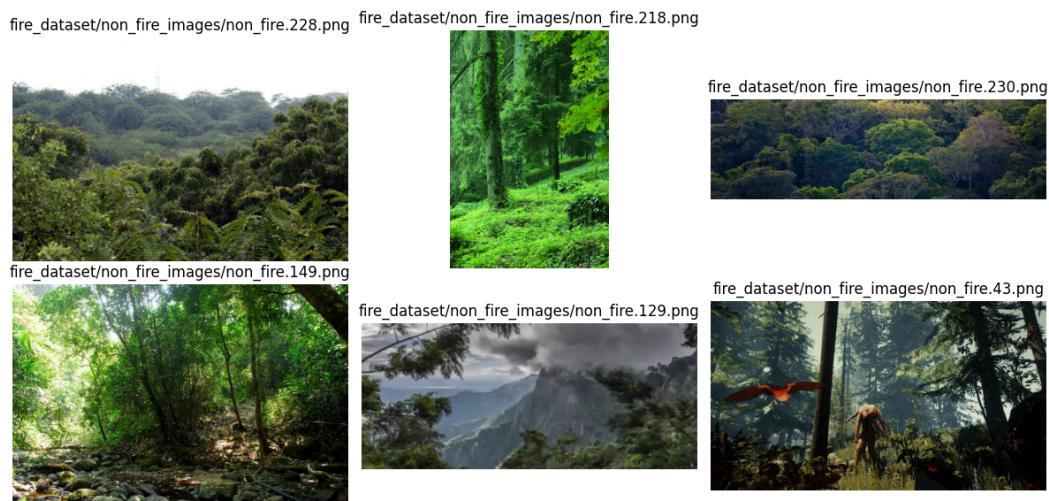
2.3.2.1.1 Nguồn dữ liệu

Bộ dữ liệu là bộ dữ liệu hình ảnh do tác giả Ahmed Saied thu thập cho cuộc thi NASA Space Apps.

2.3.2.1.2 Mô tả dữ liệu

Bộ dữ liệu chứa các hình ảnh có hoặc không có lửa cháy trong hình. Dữ liệu có 999 hình ảnh tổng cộng.

Ví dụ một phần dữ liệu:

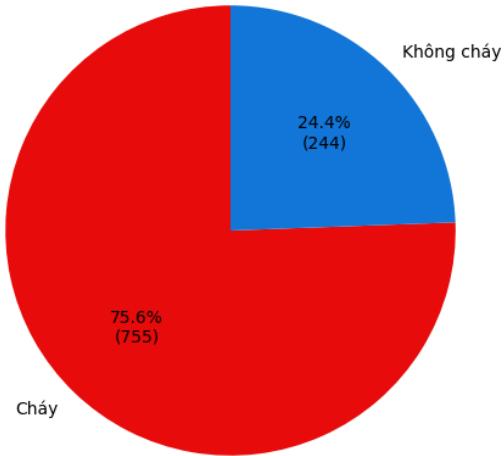


Hình 2.67: Hình ảnh không lửa



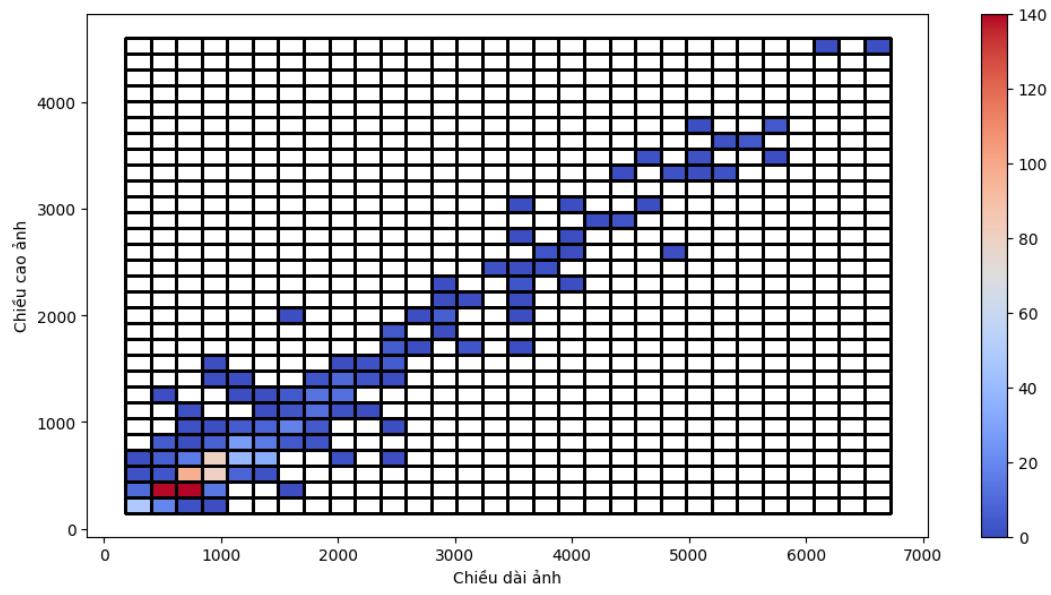
Hình 2.68: Hình ảnh có đám cháy

2.3.2.2 Phân tích, trực quan hóa dữ liệu



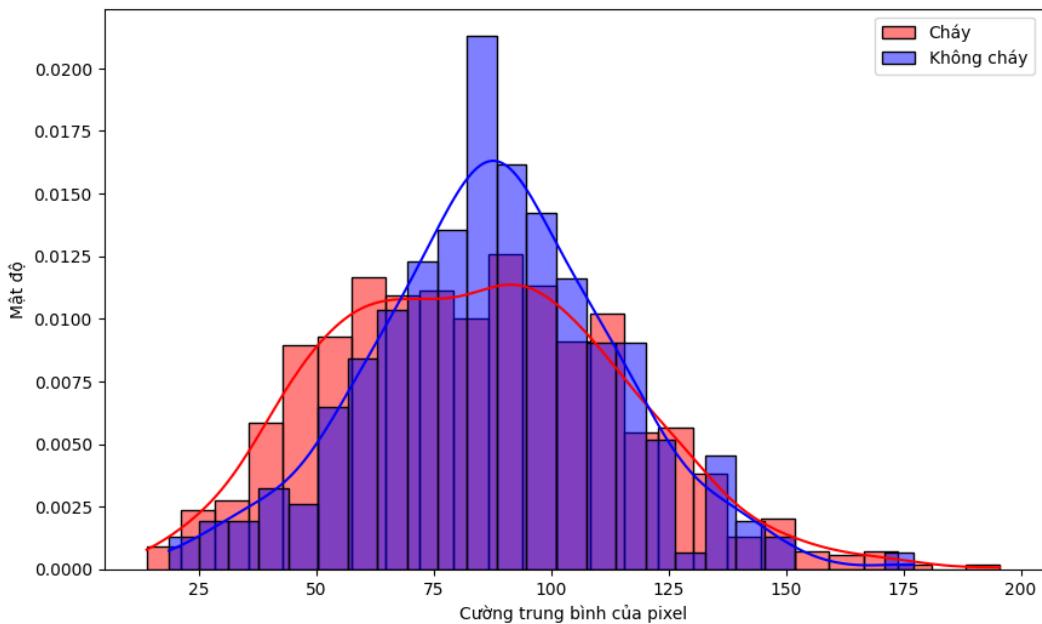
Hình 2.69: Số lượng hình ảnh mỗi nhãn

Biểu đồ pie 2.60 cho thấy sự phân phối dữ liệu giữa hai nhãn. Ta thấy dữ liệu chứa nhiều hình ảnh có đám cháy hơn so với không, hình ảnh chứa đám cháy chiếm 75.6% trong bộ dữ liệu



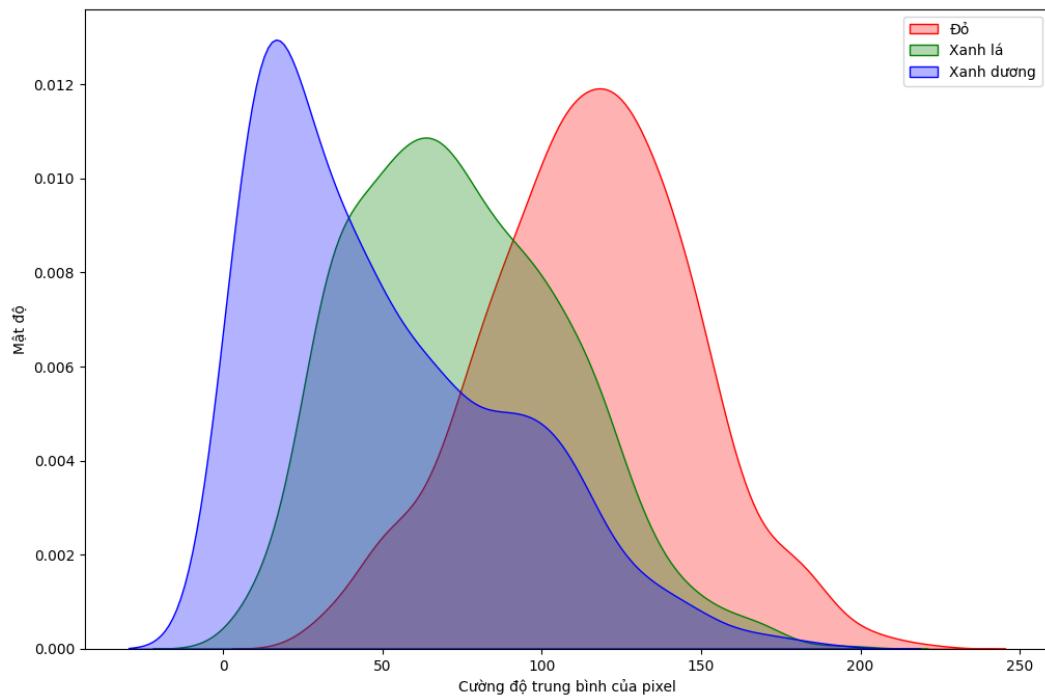
Hình 2.70: Phân phối độ phân giải ảnh

Biểu đồ 2.70 thể hiện phân bố chiều dài và chiều cao ảnh của tập dữ liệu. Dễ thấy rằng phần lớn ảnh có kích thước phân giải nhỏ, kích thước phân giải càng lớn càng ít hình. Có một vài hình rất lớn.

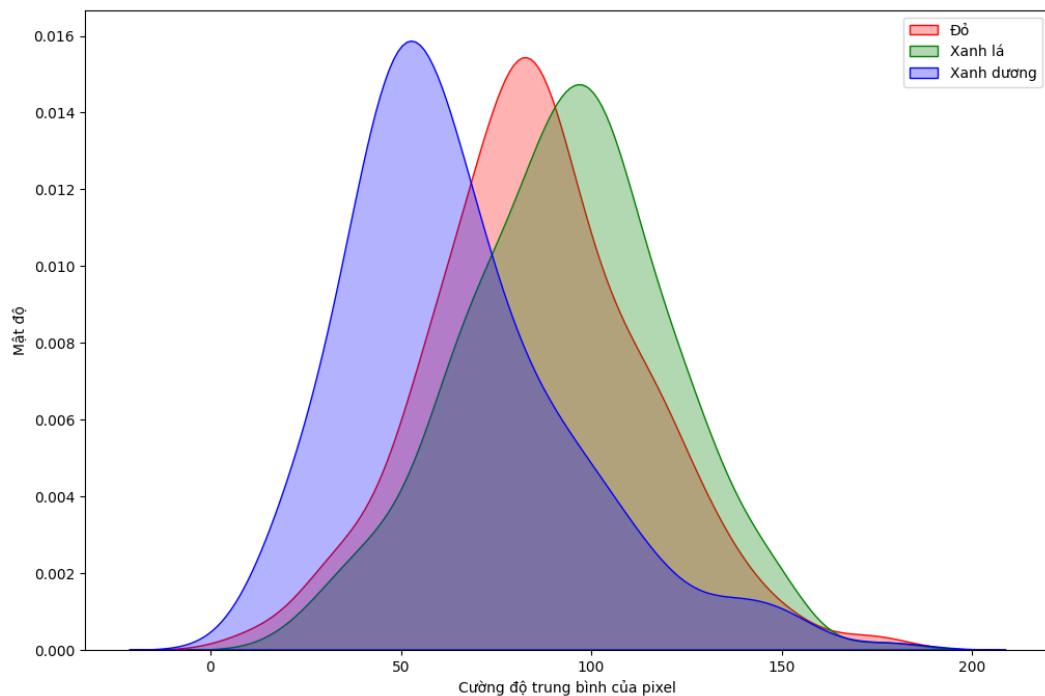


Hình 2.71: Phân phối cường độ trung bình ảnh

Biểu đồ 2.71 thể hiện sự phân phối cường độ pixel trung bình giữa hai nhóm ảnh. Cả hai phân phối đều có dáng hình chuông, tuy nhiên nhóm không cháy có đỉnh cao hơn và có dốc như phân phối chuẩn. Còn nhóm ảnh có đám cháy đỉnh thấp và trải rộng, cho thấy cường độ sáng thiếu đồng nhất, có thể là do sự xuất hiện của nguồn sáng (đám cháy) trong ảnh.



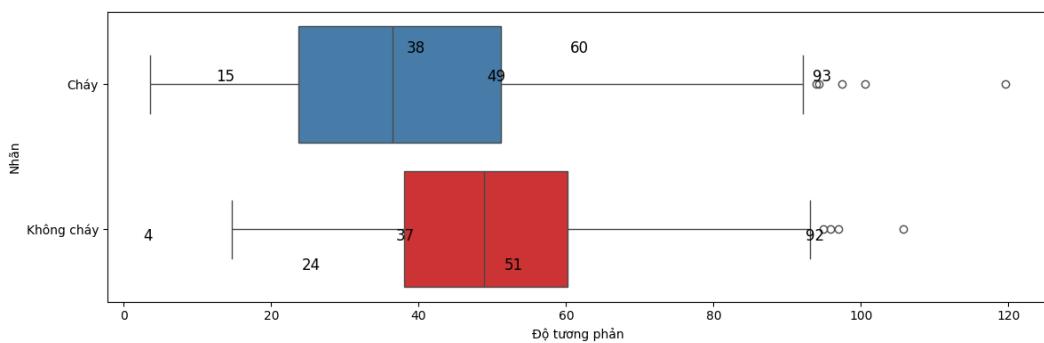
Hình 2.72: Phân phối cường độ trung bình các màu trong ảnh có đám cháy



Hình 2.73: Phân phối cường độ trung bình các màu trong ảnh không có cháy

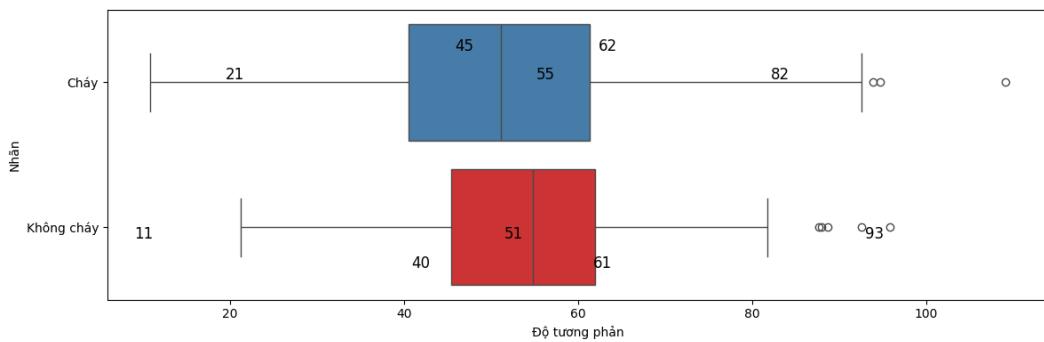
Biểu đồ 2.72 cho thấy kênh đỏ nổi bật hơn hẳn, với đỉnh phân bố cao và lệch phái, cho thấy cường độ đỏ trong các vùng cháy thường rất mạnh. Điều này phản ánh đặc tính quang phổ của lửa, vốn phát sáng chủ yếu trong vùng đỏ, cam. Kênh xanh dương lại có phân bố thấp và lệch trái rõ rệt, tập trung nhiều ở vùng cường độ thấp, cho thấy sự suy giảm đáng kể của màu xanh trong các khu vực bị ảnh hưởng bởi cháy, có thể do ánh sáng đỏ lấn át và khói làm tối các vùng ảnh.

Biểu đồ 2.73 cho thấy trong ảnh không có cháy sự phân bố cường độ giữa ba kênh khá đồng đều và đối xứng, gần với phân phối chuẩn, thể hiện sự cân bằng tự nhiên của ánh sáng trong môi trường bình thường. Không có kênh màu nào quá trội, và ba đường phân bố chồng lấn nhau nhiều.



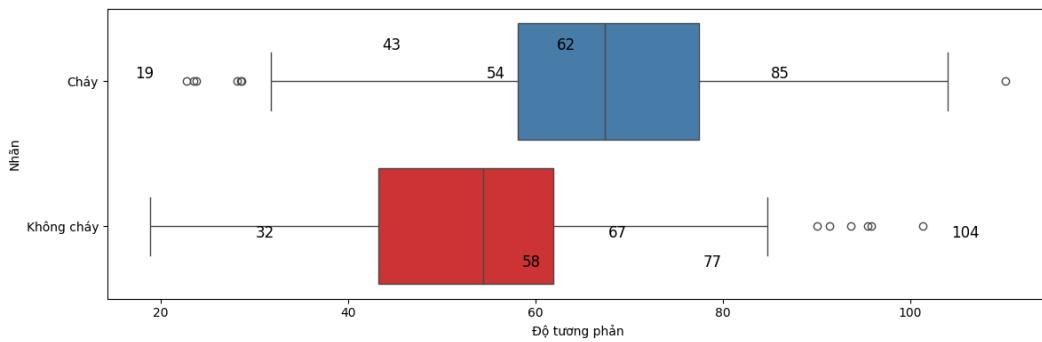
Hình 2.74: Độ tương phản kênh xanh dương theo nhãn

Biểu đồ 2.74 cho thấy rằng trong ảnh cháy, độ tương phản của kênh xanh dương có xu hướng thấp và ít biến động hơn, có thể do sự mất cân bằng màu khi ngọn lửa và khói làm giảm độ phân biệt sáng-tối ở vùng xanh dương. Trong khi đó, ở ảnh không cháy, kênh xanh dương giữ được sự phân tán rộng và mức độ tương phản cao hơn



Hình 2.75: Độ tương phản kênh xanh lá theo nhãn

Biểu đồ 2.75 cho thấy độ tương phản kênh xanh lá của hai loại hình ảnh khá tương đồng, với khá biệt duy nhất là khoảng từ vị phân và hai đầu whisker của nhóm ảnh có đám cháy phân rộng hơn.



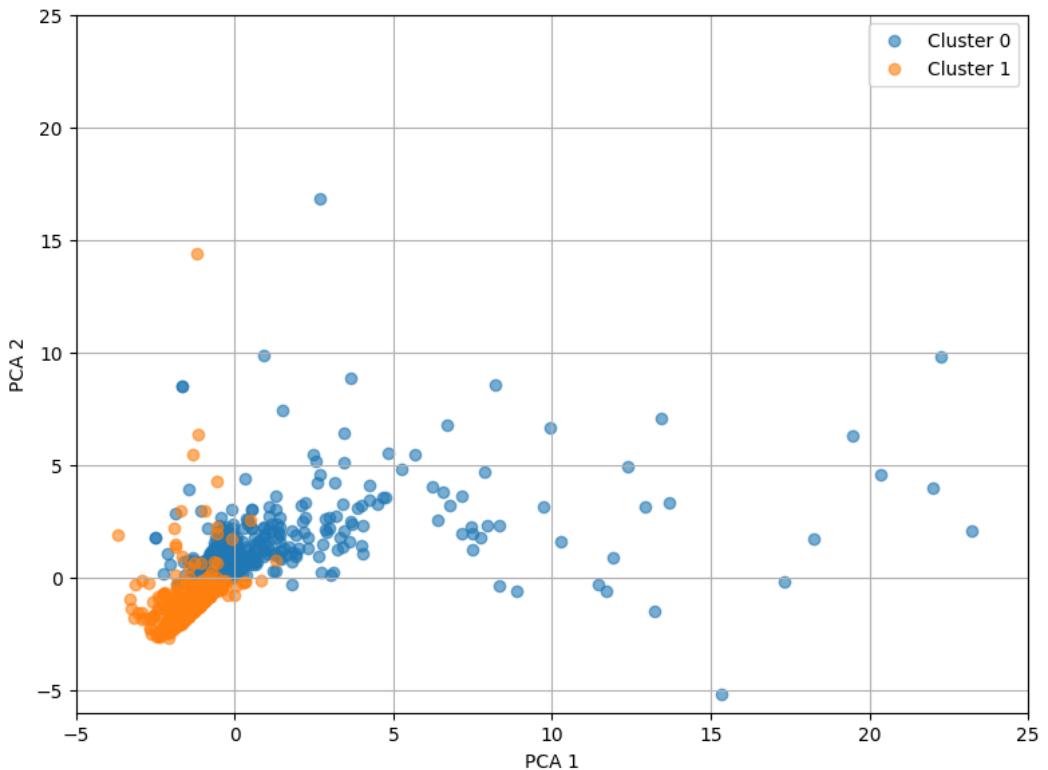
Hình 2.76: Độ tương phản kênh đỏ theo nhãn

Biểu đồ 2.76 thể hiện rõ sự khác biệt về độ tương phản của kênh đỏ giữa hai nhóm ảnh cháy và không cháy. Ở ảnh cháy, độ tương phản tập trung nhiều hơn ở các giá trị cao, với phân bố nghiêng về phía phải và trung vị đạt mức 62, cho thấy lửa làm nổi bật các vùng màu đỏ. Ngược lại, ảnh không cháy có độ tương phản thấp hơn một chút, với trung vị 58 và phân bố hẹp hơn, tuy vẫn tồn tại một số ngoại lệ cao nhưng không đủ để đẩy toàn bộ phân bố lên. Sự khác biệt này cho thấy rằng trong điều kiện có cháy, vùng màu đỏ không chỉ trở nên sáng hơn mà còn có sự biến động cục bộ rõ rệt hơn, góp phần tạo ra tương phản cao và có thể dùng làm chỉ báo quan trọng trong việc nhận diện vùng cháy trong ảnh.

2.3.2.3 Mô hình hóa dữ liệu

Để phục vụ bài toán phân loại ảnh, nhóm sử dụng đặc trưng màu được trích xuất bằng cách tính histogram màu ba chiều trong không gian RGB. Phương pháp này dựa trên việc đếm tần suất xuất hiện của các tổ hợp giá trị màu trong ảnh, sau khi phân chia từng kênh màu thành các khoảng giá trị rời rạc. Kết quả là một vector đặc trưng phản ánh phân bố tổng quát của màu sắc trong ảnh, được chuẩn hóa để đảm bảo khả năng so sánh giữa các mẫu và làm đầu vào cho mô hình học máy.

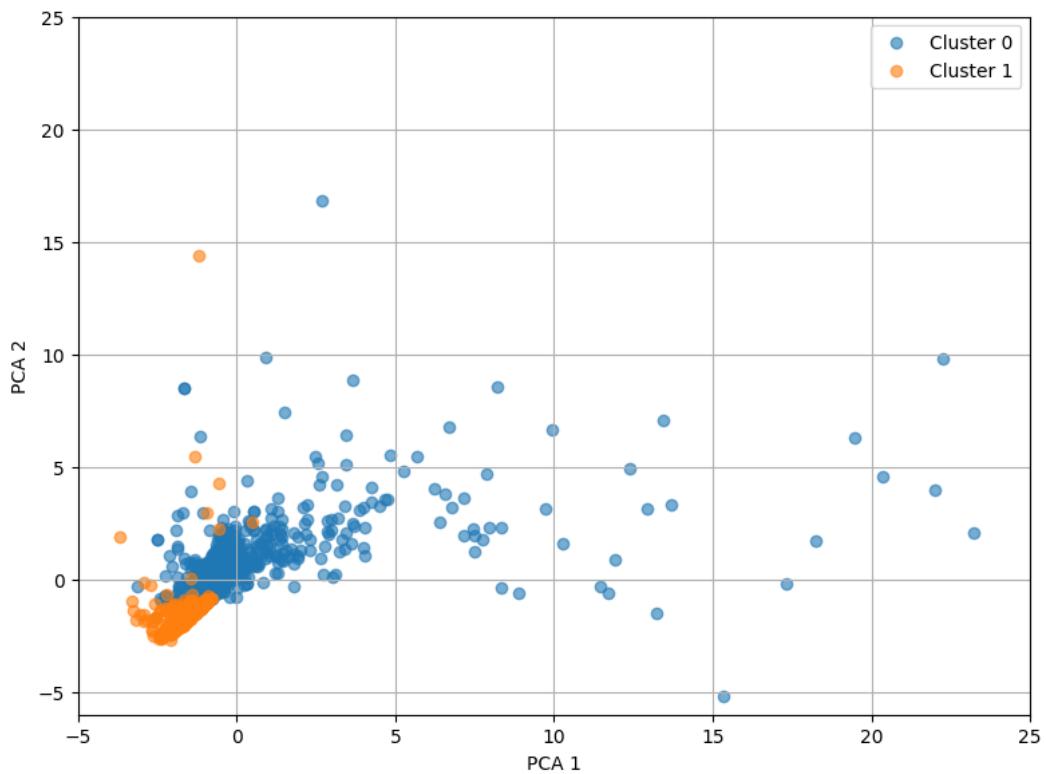
2.3.2.3.1 Phân cụm sử dụng K-means



Hình 2.77: K-Means. ARI = 0.2899

Biểu đồ 2.77 thể hiện kết quả phân cụm ảnh K-Means trên dữ liệu ảnh đám cháy sau khi giảm chiều bằng PCA. Hai cụm không tách biệt rõ ràng. Các điểm dữ liệu của hai cụm có xu hướng chồng lấn lên nhau, đặc biệt tập trung gần gốc tọa độ. Chỉ số ARI đạt 0.2899, cho thấy mô hình phân cụm tương đối nhưng chưa sát với thực tế.

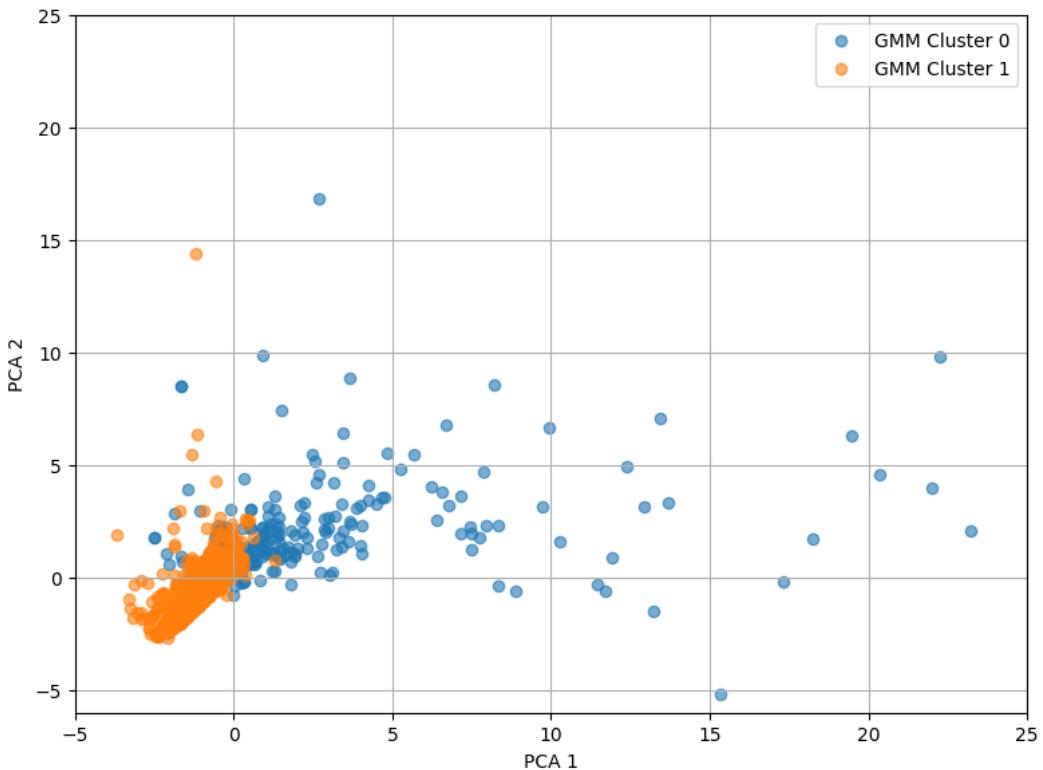
2.3.2.3.2 Phân cụm sử dụng Spectral Clustering



Hình 2.78: Spectral Clustering. ARI = 0.0838

Biểu đồ 2.78 cho thấy kết quả phân cụm của thuật toán Spectral Clustering sau khi dữ liệu được giảm chiều bằng PCA. Các điểm dữ liệu của hai cụm chồng lấn nhiều. Giá trị ARI chỉ đạt 0.0838, cho thấy kết quả phân cụm gần như không tương quan với nhãn thực tế và hiệu quả phân cụm rất thấp.

2.3.2.3.3 Phân cụm sử dụng Gaussian Mixture



Hình 2.79: Gaussian Mixture. ARI = 0.4966

Biểu đồ 2.79 cho thấy kết quả phân cụm của mô hình Gaussian Mixture sau khi dữ liệu được giảm chiều bằng PCA. Hai cụm được xác định rõ ràng, cụm màu cam tập trung chủ yếu ở vùng gần gốc tọa độ, trong khi cụm màu xanh lan rộng hơn phía ngoài phải. Mặc dù vẫn có một số điểm bị chồng lấn giữa hai cụm, nhưng sự tách biệt đã phần nào rõ ràng hơn. Chỉ số ARI đạt 0.4966, cao hơn đáng kể so với KMeans và Spectral Clustering, cho thấy GMM phản ánh cấu trúc dữ liệu tốt hơn và phân cụm có mức độ tương quan khá với nhãn thực tế.

Bảng 2.55: So sánh kết quả các mô hình

Model	ARI
K-Means	0.2899
Spectral Clustering	0.0838
Gaussian Mixture	0.4966

Bảng kết quả cho thấy mô hình Gaussian Mixture đạt hiệu suất cao nhất với chỉ số ARI là 0.4966, thể hiện khả năng phân cụm gần đúng với nhãn thực tế. K-Means đứng thứ hai với giá trị ARI 0.2899, cho thấy mức độ phân biệt cụm ở mức trung bình. Trong khi đó, Spectral Clustering có kết quả thấp nhất với ARI chỉ 0.0838, phản ánh khả năng phân cụm yếu và gần như không tương quan nhiều với cấu trúc thực tế của dữ liệu. Điều này cho thấy Gaussian Mixture là lựa chọn phù hợp hơn cả trong bài toán phân cụm này.

TÀI LIỆU THAM KHẢO

- [1] Jesús Armenta-Segura and Grigori Sidorov. “Anime Success Prediction Based on Synopsis Using Traditional Classifiers”. In: *Proceedings of Congreso Mexicano de Inteligencia Artificial, COMIA*. 2023.
- [2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [3] Leo Breiman. *Arcing the edge*. Tech. rep. Technical Report 486, Statistics Department, University of California at ..., 1997.
- [4] Leo Breiman. *Classification and regression trees*. Wadsworth International Group, 1984.
- [5] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001).
- [6] Colin M.L. Burnett. *Artificial neural network*. 2006. URL: https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg.
- [7] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.
- [8] xgboost developers. *Introduction to Boosted Trees*. URL: <https://xgboost.readthedocs.io/en/latest/tutorials/model.html> (visited on 07/07/2025).
- [9] Thị Hà Dương and Thái Nghe Nguyễn. “Ứng dụng mô hình đa biến bộ nhớ dài - ngắn hạn trong dự báo nhiệt độ và lượng mưa”. In: *Can Tho University Journal of Science* 58.4 (Aug. 2022), 8–16. ISSN: 1859-2333. DOI: [10.22144/ctu.jvn.2022.158](https://doi.org/10.22144/ctu.jvn.2022.158). URL: <http://dx.doi.org/10.22144/ctu.jvn.2022.158>.
- [10] Bradley Efron. “Bootstrap methods: another look at the jackknife”. In: *Breakthroughs in statistics*. Springer, 1992.
- [11] Yoav Freund, Robert Schapire, and Naoki Abe. “A short introduction to boosting”. In: *Journal-Japanese Society For Artificial Intelligence* 14.771-780 (1999), p. 1612.
- [12] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.
- [13] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.
- [14] Jerome H Friedman. “Stochastic gradient boosting”. In: *Computational statistics & data analysis* 38.4 (2002), pp. 367–378.
- [15] Wanda Huel. *Gradient Boosting Trees for Classification: A Beginner’s Guide*. URL: <https://morioh.com/p/e108a4521555>.

- [16] Ixnay. *Recurrent neural network unfold*. 2017. URL: https://en.wikipedia.org/wiki/File:Recurrent_neural_network_unfold.svg.
- [17] CRakesh Kapilavayi. *Extrovert vs. Introvert Behavior Data*. <https://www.kaggle.com/datasets/rakeshkapilavai/extrovert-vs-introvert-behavior-data/data>. Kaggle. 2025.
- [18] Mónica V Martins et al. “Early prediction of student’s performance in higher education: A case study”. In: *Trends and Applications in Information Systems and Technologies: Volume 1 9*. Springer. 2021, pp. 166–175.
- [19] Văn Thủy Nguyễn. “Using Machine Learning models to predict the on-time graduation status of students”. In: *Tạp chí Khoa học và Đào tạo Ngân hàng* 255 (Aug. 2023), 52–64. ISSN: 1859-011X. DOI: [10.59276/TCKHDT.2023.08.2506](https://doi.org/10.59276/TCKHDT.2023.08.2506). URL: <http://dx.doi.org/10.59276/TCKHDT.2023.08.2506>.
- [20] nikki2398@nikki2398. *ML - Gradient Boosting*. 2020. URL: <https://www.geeksforgeeks.org/ml-gradient-boosting/> (visited on 07/07/2025).
- [21] Thuan P.Q and Nguyễn Thuân. “ÚNG DỤNG CÁC THUẬT TOÁN HỌC MÁY ĐỂ ĐÁNH GIÁ BỘ CƠ SỞ DỮ LIỆU TRONG PHÂN LOẠI RỒI LOẠN PHỎ TỰ KÝ”. In: *Tạp chí Khoa học Đại học Đà Lạt* 10 (Sept. 2020), p. 39. DOI: [10.37569/DalatUniversity.10.3.649\(2020\)](https://doi.org/10.37569/DalatUniversity.10.3.649(2020)).
- [22] Juan Quiroz et al. “Fault detection of broken rotor bar in LS-PMSM using random forests”. In: *Measurement* 116 (Nov. 2017), pp. 273–280. DOI: [10.1016/j.measurement.2017.11.004](https://doi.org/10.1016/j.measurement.2017.11.004).
- [23] Mayda Rohmani, Dwi Hartanti, and Anindhiasti Asri. “KNOWLEDGE-BASED HIJAB PRODUCT SELECTION RECOMMENDATION SYSTEM AT CANDY SCARVES”. In: *Jurnal Riset Informatika* 7 (June 2025), pp. 219–227. DOI: [10.34288/jri.v7i3.377](https://doi.org/10.34288/jri.v7i3.377).
- [24] Nathanael Setiawan et al. “Time Series Model to Predict Future Popular Animes Genres in 2025”. In: *E3S Web of Conferences*. Vol. 388. EDP Sciences. 2023, p. 02002.
- [25] Jianbo Shi and J. Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905. DOI: [10.1109/34.868688](https://doi.org/10.1109/34.868688).
- [26] Josh Starmer. *Gradient Boost Part 2 (of 4): Regression Details*. 2019. URL: <https://www.youtube.com/watch?v=2xudP0Bz-vs&>; (visited on 07/07/2025).
- [27] Josh Starmer. *Gradient Boost Part 4 (of 4): Classification Details*. 2019. URL: <https://www.youtube.com/watch?v=StWY5QWMXCw> (visited on 07/07/2025).
- [28] Josh Starmer. *XGBoost Part 3 (of 4): Mathematical Details*. 2020. URL: <https://www.youtube.com/watch?v=ZVFeW798-2I> (visited on 07/07/2025).
- [29] Taylor's theorem. 2021. URL: https://en.wikipedia.org/wiki/Taylor%27s_theorem (visited on 07/07/2025).

-
- [30] Zhuo Wang, Jintao Zhang, and Naveen Verma. “Realizing Low-Energy Classification Systems by Implementing Matrix Multiplication Directly Within an ADC”. In: *IEEE Transactions on Biomedical Circuits and Systems* 9 (Dec. 2015), pp. 1–1. DOI: [10.1109/TBCAS.2015.2500101](https://doi.org/10.1109/TBCAS.2015.2500101).
 - [31] Yunsu Xiaozi. *Daily Climate time series data*. <https://www.kaggle.com/code/yunsuxiaozi/cnn-lstm>. Kaggle. 2023.
 - [32] Xiaofeng Yuan, Lin Li, and Yalin Wang. “Nonlinear Dynamic Soft Sensor Modeling With Supervised Long Short-Term Memory Network”. In: *IEEE Transactions on Industrial Informatics* PP (Feb. 2019), pp. 1–1. DOI: [10.1109/TII.2019.2902129](https://doi.org/10.1109/TII.2019.2902129).