

ĐỒ ÁN MÔN KHOA HỌC DỮ LIỆU NÂNG CAO

ĐỀ TÀI:

**ỨNG DỤNG MÔ HÌNH PHÂN LỚP
TRONG BỘ DỮ LIỆU HORSE COLIC**

Nhóm 8
GVHD: TS. Đỗ Như Tài
Ngày 29/12/2023

1

GIỚI THIỆU

- **Bộ dữ liệu Horse Colic Dataset:**

- “Colic” là một thuật ngữ mô tả rối loạn tiêu hóa ở ngựa, là nguyên nhân dẫn đến tử vong của khoảng 64.000 con ngựa tại Hoa Kỳ mỗi năm.
- Nó có thể do rất nhiều nguyên nhân khác nhau gây ra và thường bắt đầu đột ngột.



- **Our problem:**

- Xây dựng mô hình phân loại gồm 10 phương pháp để hỗ trợ nhân viên chăm sóc ngựa và các bác sĩ xác định nguy cơ tử vong của ngựa nếu mắc bệnh. Từ đó giúp doanh nghiệp tiết kiệm chi phí cũng như có các kế hoạch giảm thiểu tổn thất tốt nhất.
- Sử dụng Accuracy, Precision, Recall, F1-score và confusion matrix để so sánh hiệu suất của các phương pháp phân loại.

Mục lục

1 GIỚI THIỆU

2 CÁC KIẾN THỨC LIÊN QUAN VÀ
NHỮNG PHƯƠNG PHÁP CHÍNH

3 NGHIÊN CỨU VẤN ĐỀ

4 THỬ NGHIỆM VÀ KẾT QUẢ

5 KẾT LUẬN

2

CÁC KIẾN THỨC LIÊN QUAN VÀ NHỮNG PHƯƠNG PHÁP CHÍNH

Các kiến thức liên quan và những phương pháp chính

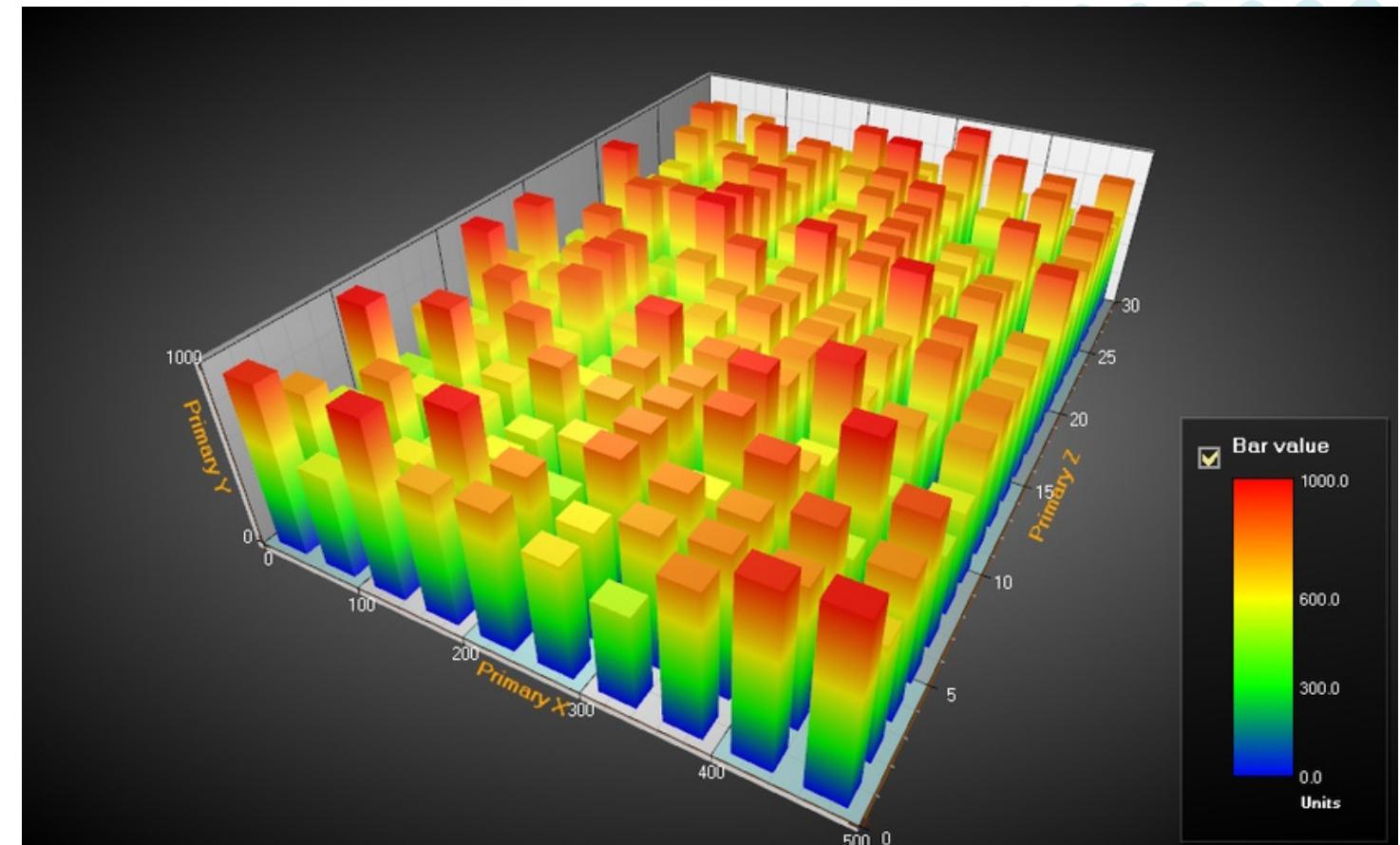
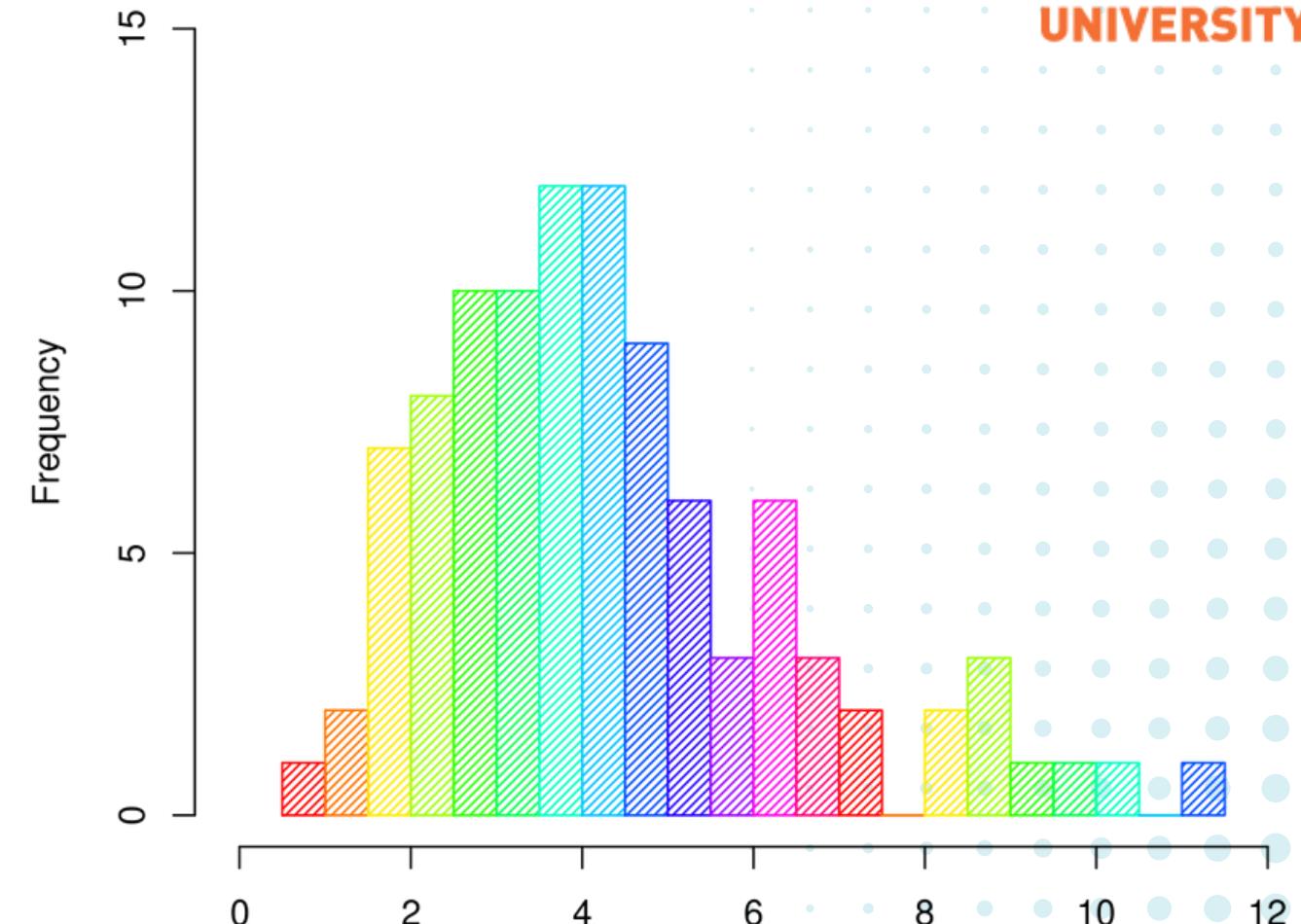
- **EDA**

Countplot: Biểu đồ phổ biến để so sánh tần suất xuất hiện của các nhóm trong dữ liệu.

Histogram: Biểu đồ thống kê biểu diễn phân phối tần suất của dữ liệu.

Correlogram: Biểu diễn mối quan hệ tuyến tính giữa các biến số dữ liệu.

Heatmap: Minh họa mức độ tương quan giữa các biến số trong một tập dữ liệu lớn.



Các kiến thức liên quan và những phương pháp chính

• Tiền xử lý dữ liệu

One Hot Encoder: Biến đổi dữ liệu phân loại thành dạng số liệu.

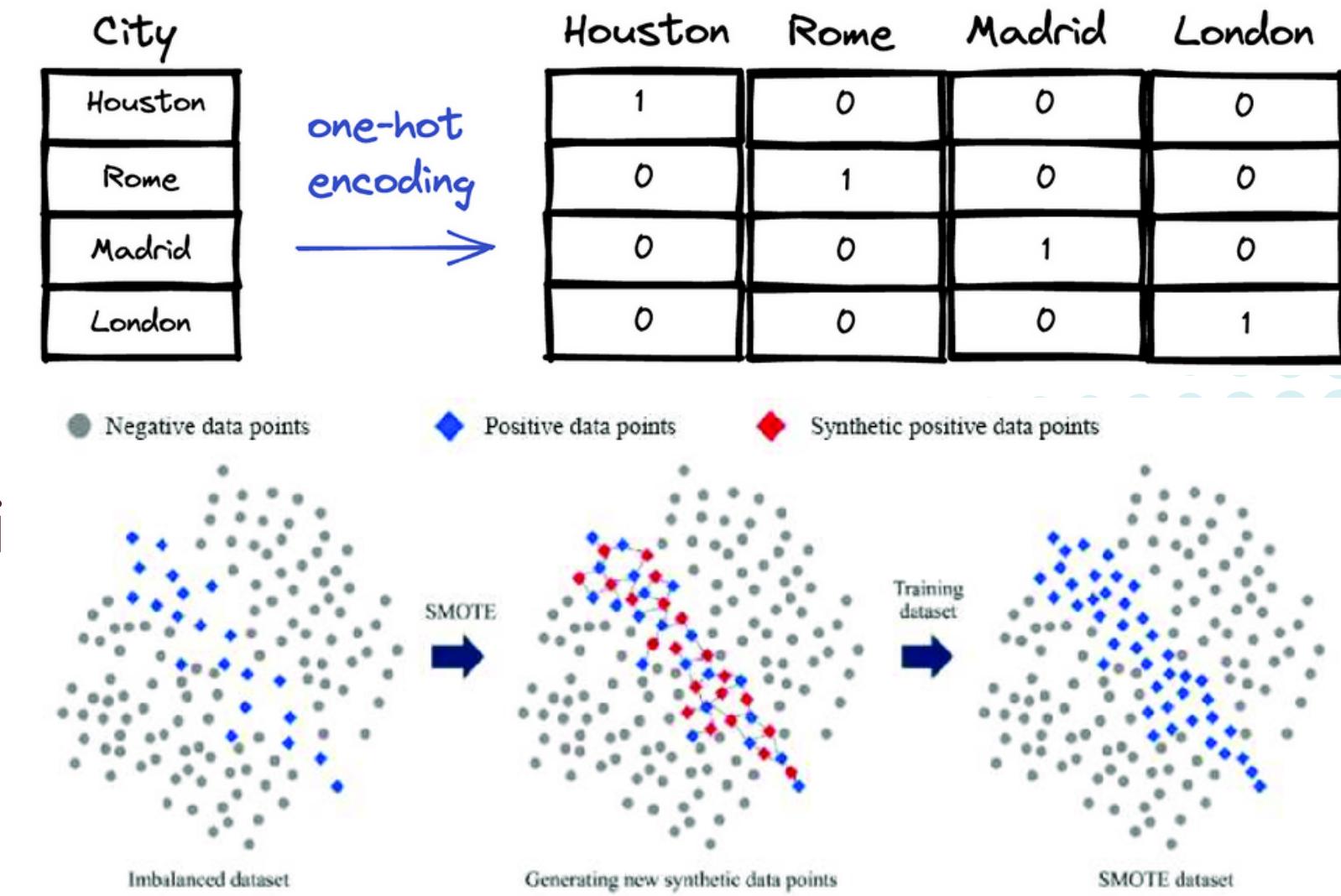
Label Encoder: Chuyển đổi các biến phân loại thành số nguyên tăng dần.

Ordinal Encoder: Chuyển đổi các biến phân loại thành dạng số nguyên

Min Max Scaler: Đồng nhất dữ liệu đầu vào.

Standard Scaler: Phương pháp chuẩn hóa dữ liệu.

SMOTE: Kỹ thuật tăng cường dữ liệu được sử dụng trong bài toán xử lý mẫu không cân bằng.



$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Các kiến thức liên quan và những phương pháp chính

- Model

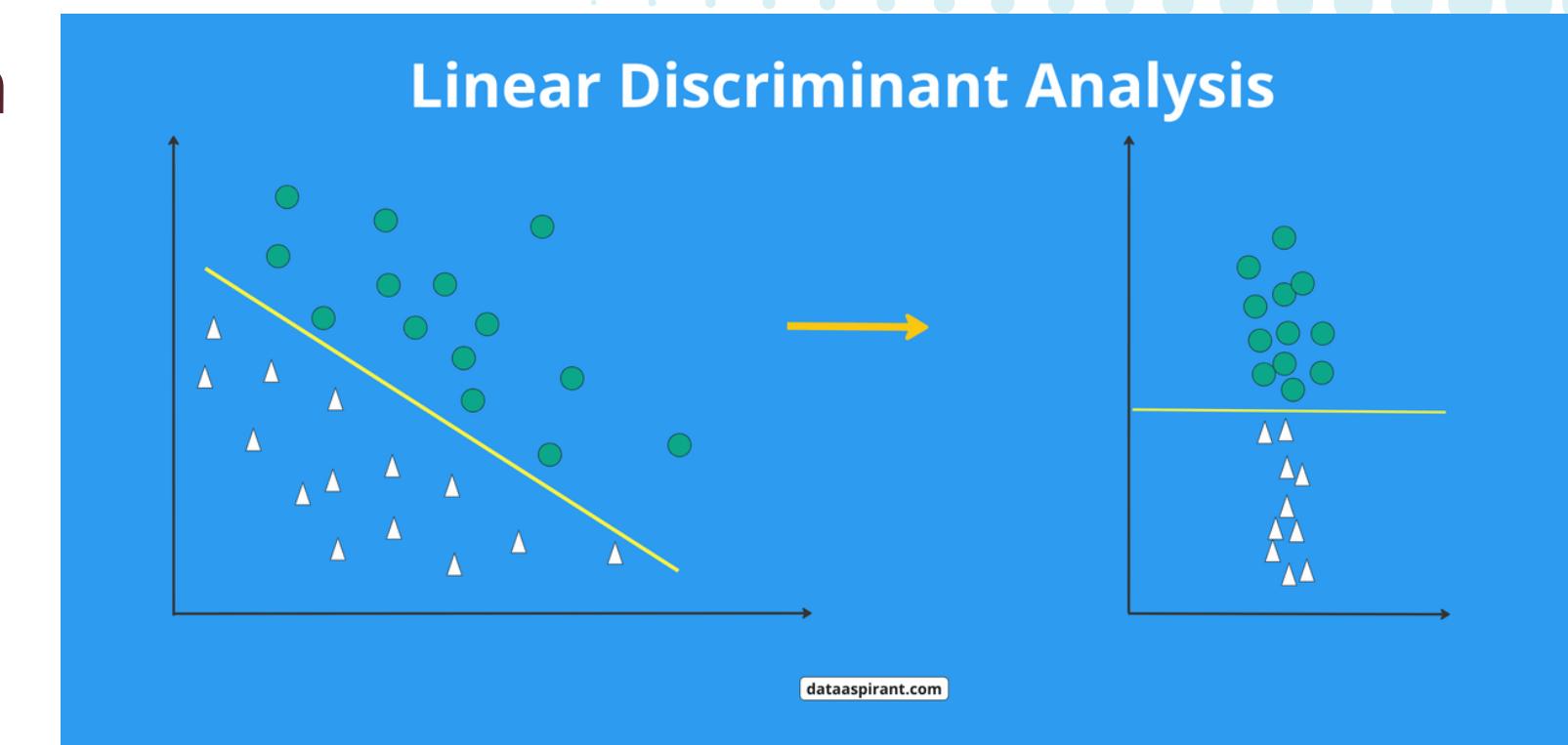
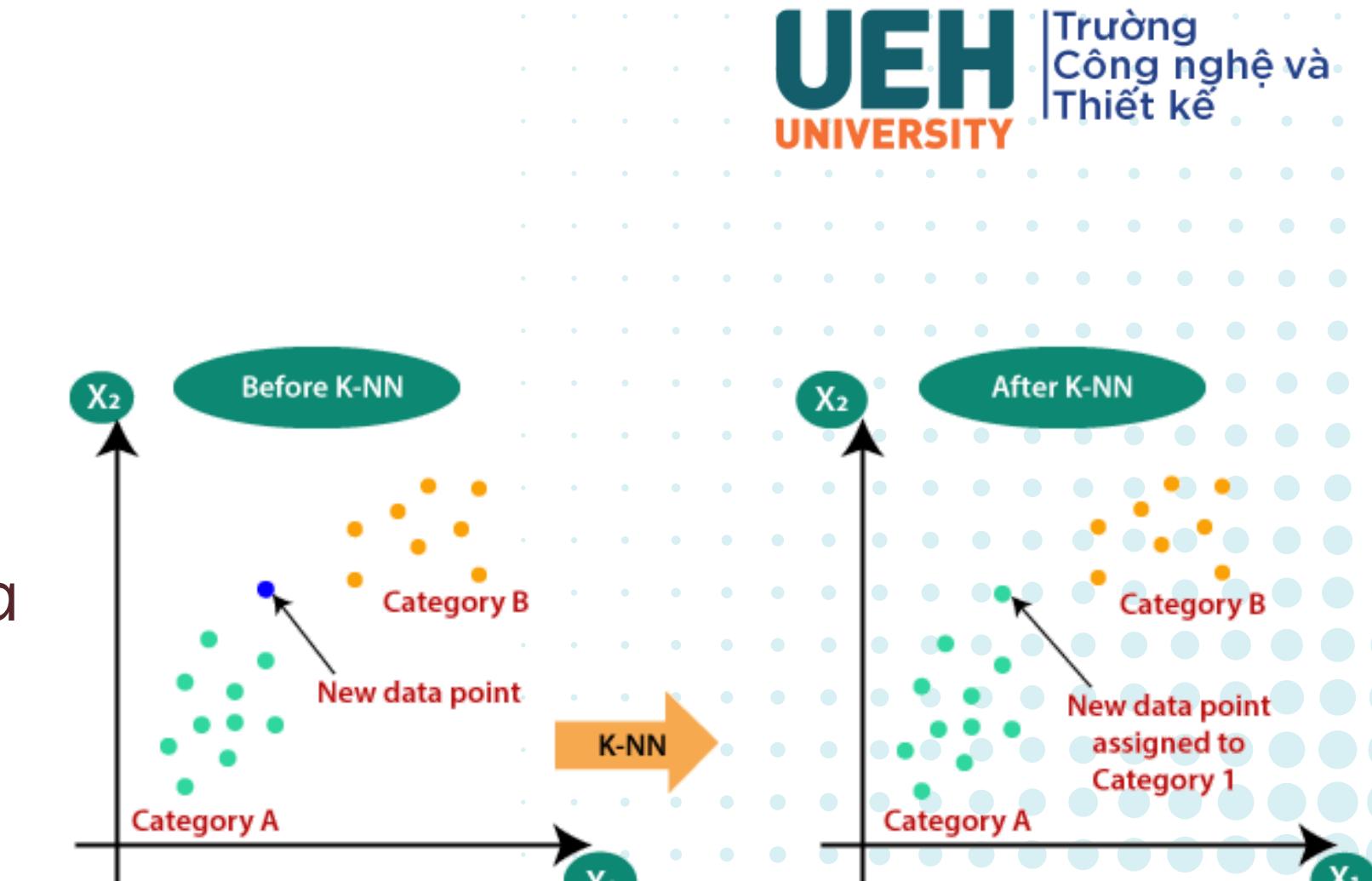
Decision Tree: Dự đoán bằng cách tạo ra một chuỗi các quyết định dựa trên các đặc trưng của dữ liệu đầu vào.

Linear Discriminant Analysis: Tối ưu hóa sự phân biệt giữa các lớp.

K-Neighbors: Phân loại dựa trên nguyên lý gần nhất hạn chế.

SVC: Phân loại dựa trên máy vector hỗ trợ.

GaussianNB: Phân loại văn bản và dữ liệu có đặc trưng liên tục.

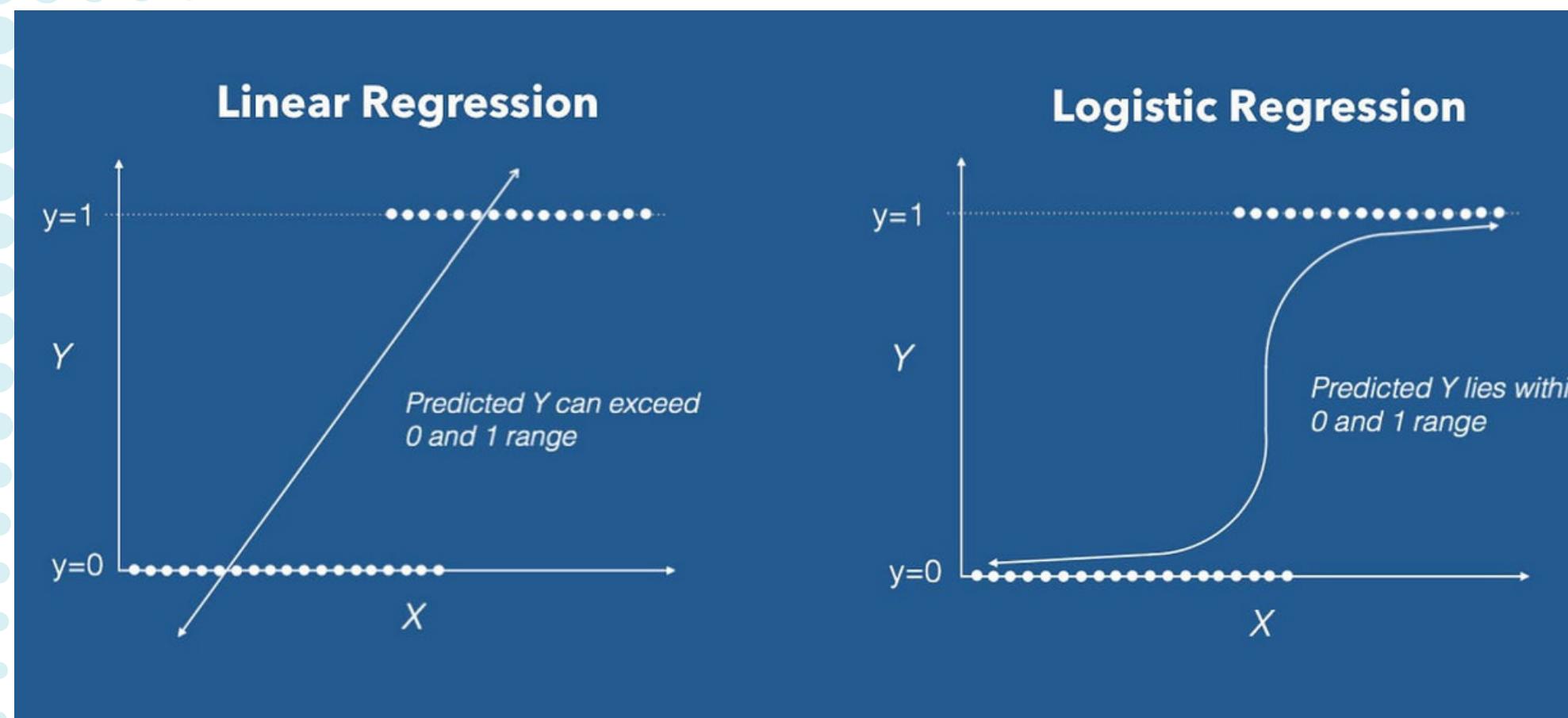


Các kiến thức liên quan và những phương pháp chính

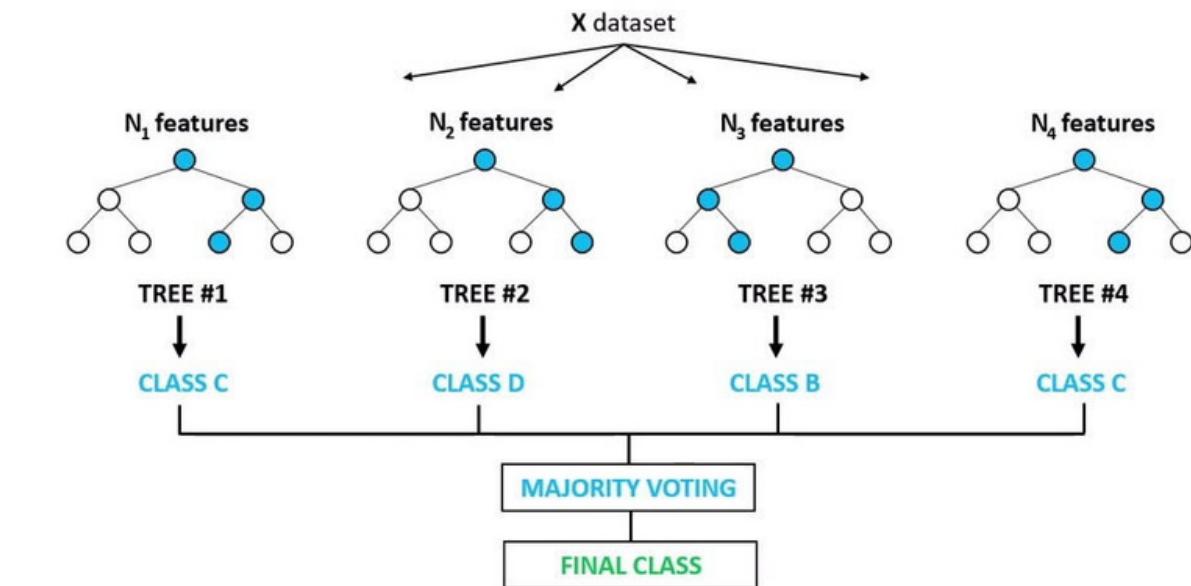
- Model

Logistic Regression: Dự đoán xác suất của một sự kiện xảy ra dựa trên các biến đầu vào.

Random Forest: Kết hợp nhiều cây quyết định để tạo ra dự đoán cuối cùng



Random Forest Classifier

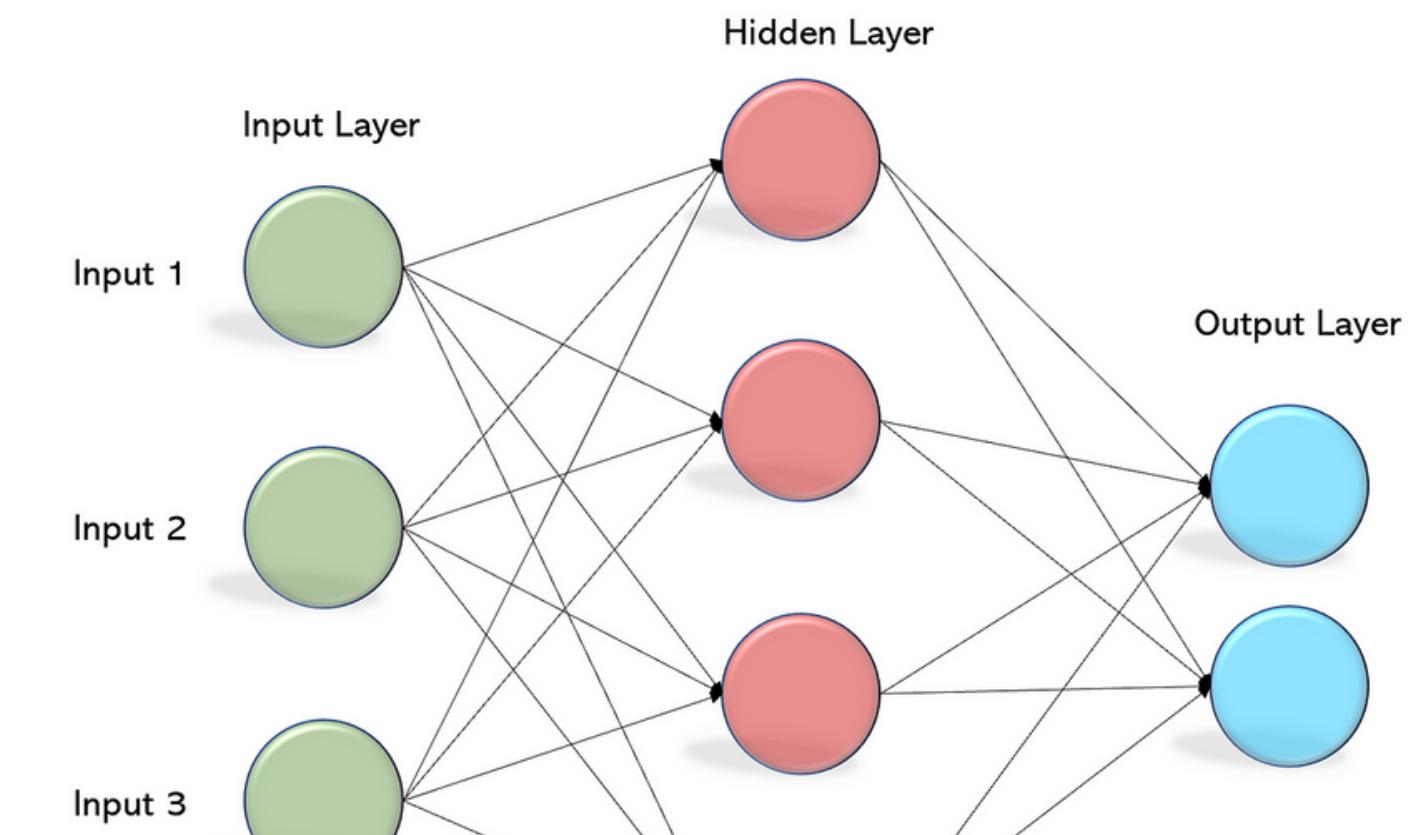
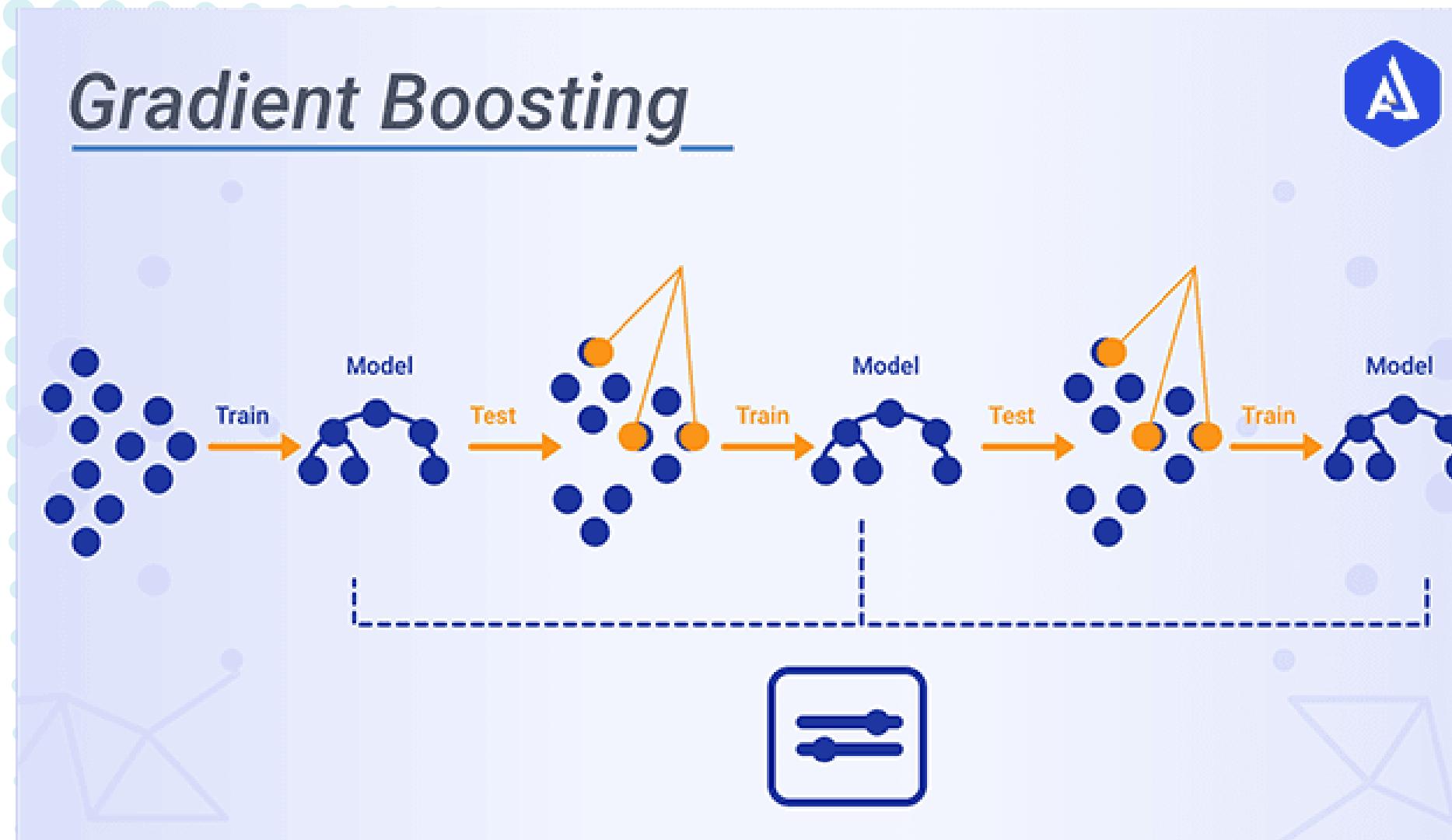


Các kiến thức liên quan và những phương pháp chính

- Model

AdaBoost, Gradient Boost: Thuộc loại "boosting". Dự đoán tiếp theo dựa trên sai số của bước trước đó.

MLP: Học các quy luật phức tạp từ dữ liệu.



• Đánh giá mô hình

-**Accuracy, Precision, Recall, F1-score**

-**Confusion matrix:** Bao gồm các thông tin về số lượng các điểm dữ liệu được phân loại đúng và sai cho từng lớp.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

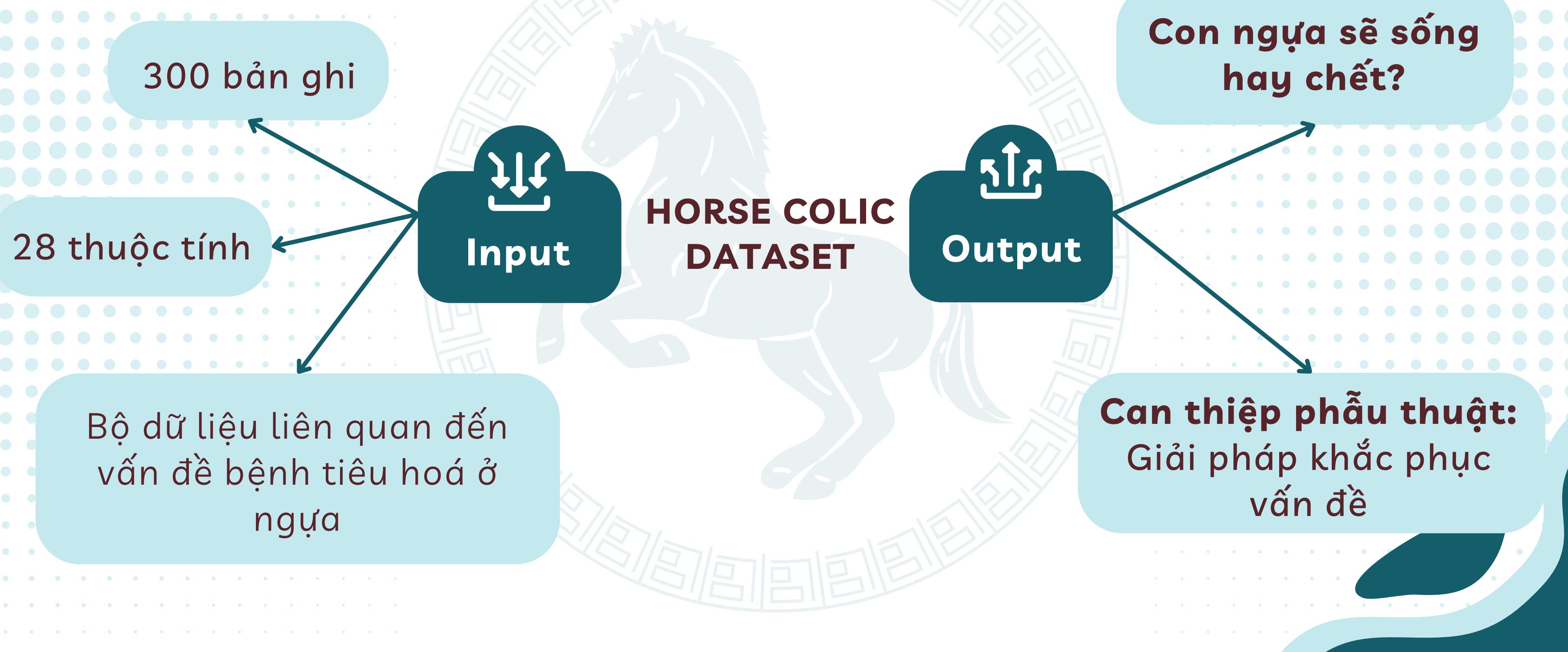
3

NGHIÊN CỨU VẤN ĐỀ

3

NGHIÊN CỨU VẤN ĐỀ

3.1. XÁC ĐỊNH VẤN ĐỀ



3

NGHIÊN CỨU VẤN ĐỀ

3.1. XÁC ĐỊNH VẤN ĐỀ



Mục tiêu chính

Tạo mô hình dự đoán kết quả việc con
ngựa sẽ như thế nào?

Hỗ trợ bác sĩ thú y xác định trường hợp ngựa có
nguy cơ tử vong cao và can thiệp phẫu thuật

Xác thực phán đoán, điều trị chính xác
tình trạng ngựa bị đau bụng

3

NGHIÊN CỨU VẤN ĐỀ

3.2. KHÁM PHÁ DỮ LIỆU (EDA)

Dạng dữ liệu: Dữ liệu này được thu thập từ 300 trường hợp bệnh lý của ngựa. Bao gồm 3 file chính:

- **horse-colic.names**
- **horse-colic.data**
- **horse-colic.csv**

Bộ dữ liệu 28 biến bao gồm 27 cột và 300 dòng

Mục tiêu chính: Dự đoán kết quả cuối cùng của các trường hợp viêm ruột của ngựa là sống, chết hay trợ tử.

Biến phân loại

'Surgery', 'Age', 'mucous membranes', 'pain',
'peristalsis', 'abdominal distension', 'nasogastric tube', 'nasogastric reflux',
'rectal examination - feces',
'abdomen',
'abdominocentesis appearance', 'outcome',
'surgical lesion?', 'cp_data'

Biến số

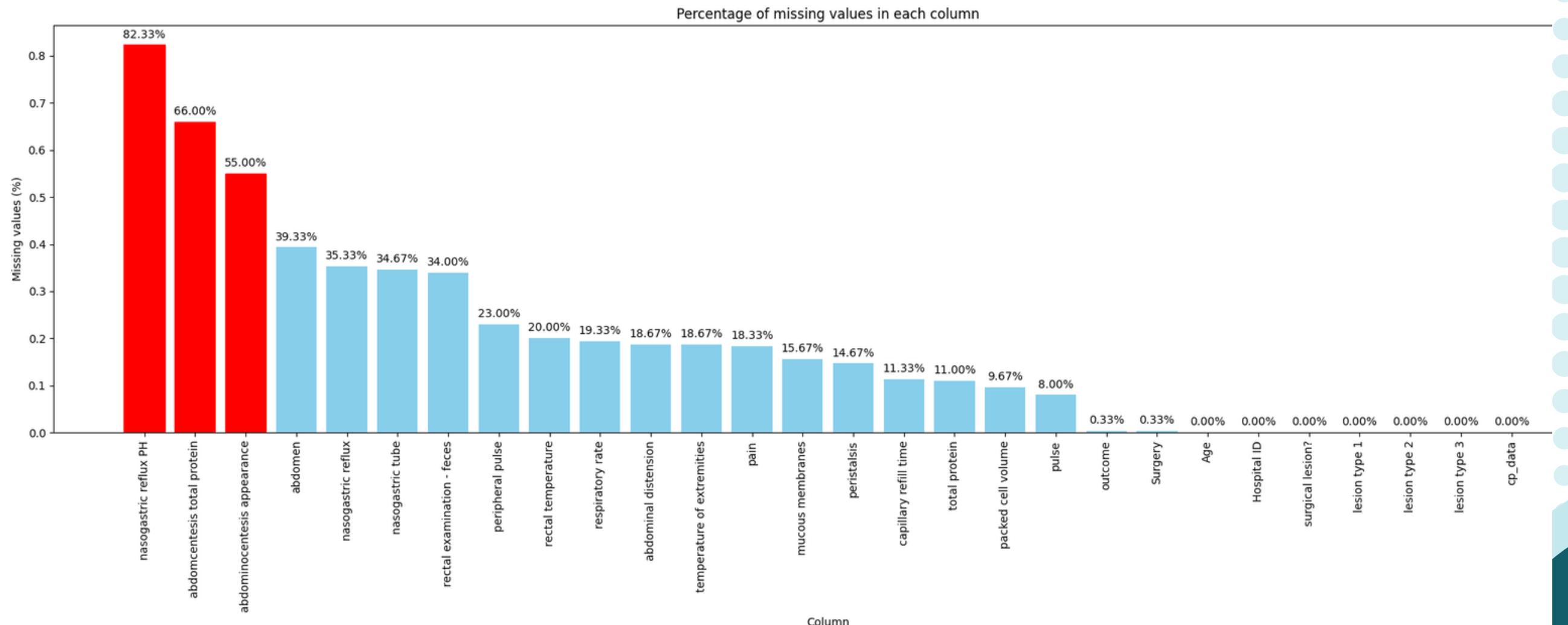
'Hospital ID', 'rectal temperature', 'pulse',
'respiratory rate',
'temperature of extremities',
'peripheral pulse', 'capillary refill time', 'nasogastric reflux PH',
'packed cell volume', 'total protein',
'abdomcentesis total protein', 'lesion type 1',
'lesion type 2', 'lesion type 3'

3

NGHIÊN CỨU VẤN ĐỀ

3.2. KHÁM PHÁ DỮ LIỆU (EDA)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype  
--- 
 0   Surgery          299 non-null    float64
 1   Age              300 non-null    int64   
 2   Hospital ID      300 non-null    int64   
 3   rectal temperature 240 non-null    float64
 4   pulse             276 non-null    float64
 5   respiratory rate 242 non-null    float64
 6   temperature of extremities 244 non-null    float64
 7   peripheral pulse 231 non-null    float64
 8   mucous membranes 253 non-null    float64
 9   capillary refill time 268 non-null    float64
 10  pain              245 non-null    float64
 11  peristalsis       256 non-null    float64
 12  abdominal distension 244 non-null    float64
 13  nasogastric tube 196 non-null    float64
 14  nasogastric reflux 194 non-null    float64
 15  nasogastric reflux PH 53 non-null    float64
 16  rectal examination - feces 198 non-null    float64
 17  abdomen           182 non-null    float64
 18  packed cell volume 271 non-null    float64
 19  total protein     267 non-null    float64
 20  abdominocentesis appearance 135 non-null    float64
 21  abdominocentesis total protein 102 non-null    float64
 22  outcome            299 non-null    float64
 23  surgical lesion? 300 non-null    int64  
 24  lesion type 1     300 non-null    object  
 25  lesion type 2     300 non-null    object  
 26  lesion type 3     300 non-null    object  
 27  cp_data            300 non-null    int64  
dtypes: float64(21), int64(4), object(3)
memory usage: 65.8+ KB
```



BIỂU ĐỒ MISSING VALUE

THÔNG TIN VỀ DỮ LIỆU

3

NGHIÊN CỨU VẤN ĐỀ

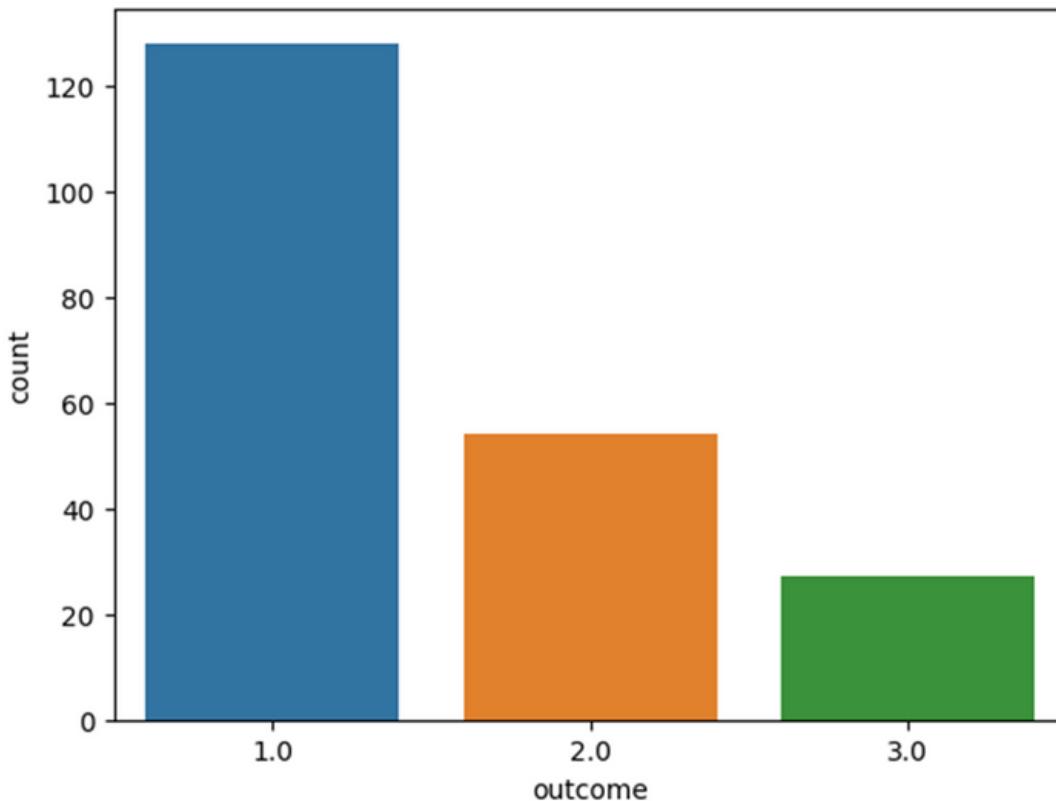
3.2. KHÁM PHÁ DỮ LIỆU (EDA)

```
[ ] _lesions = df_dataset.loc[:, ['lesion type 1','lesion type 2','lesion type 3']]
_lesions_counts = [
    _lesions[_lesions.iloc[:, 0] == '00000'].count()[0],
    _lesions[_lesions.iloc[:,0] != '00000'] & (_lesions.iloc[:,1] == '00000').count()[0],
    _lesions[_lesions.iloc[:,0] != '00000'] & (_lesions.iloc[:,1] != '00000') & (_lesions.iloc[:,2] == '00000').count()[0],
    _lesions[_lesions.iloc[:,0] != '00000'] & (_lesions.iloc[:,1] != '00000') & (_lesions.iloc[:,2] != '00000').count()[0]
]

for i, count in zip(range(4), _lesions_counts):
    print(f"{count} horse(s) with {i} lesion(s)")
print(f"-----\n-----\ntotal={sum(_lesions_counts)}")
```

```
56 horse(s) with 0 lesion(s)
237 horse(s) with 1 lesion(s)
6 horse(s) with 2 lesion(s)
1 horse(s) with 3 lesion(s)

-----
total=300
```



- Đề xuất sử dụng phương pháp SMOTE xử lý mất cân bằng dữ liệu của biến “outcome”
- Mã hóa để kiểm tra tình trạng sau bệnh của ngựa là chỉ sống '1' hoặc chết '0' từ 3 giá trị ban đầu là '1.0', '2.0', '3.0'

Dữ liệu của phần này tập trung vào type of lesion 1, dữ liệu trong cột type of lesion 2, và type of lesion 3 không có nhiều

=> Loại bỏ type of lesion 2, và type of lesion 3, type of lesion 1 chuyển thành 5 cột mới “site of lesion”, “type”, “subtype”, “specific code”, “none”

3

NGHIÊN CỨU VẤN ĐỀ

3.2. KHÁM PHÁ DỮ LIỆU (EDA)



Các biến liên tục 'rectal temperature', 'pulse', 'respiratory rate', 'nasogastric reflux PH', 'packed cell volume', 'total protein', và 'abdomcentesis total protein'

🔍 Xu hướng lệch phải, phân phối không đều và có nhiều trường hợp đặc biệt.

=> Thay thế các giá trị thiếu, ngoại lệ và nhiễu bằng giá trị trung bình, trung vị hoặc chuẩn hóa dữ liệu, kỹ thuật như KNN

Các biến rời rạc 'Surgery', 'Age', 'pain'

🔍 Giá trị rỗng và không có thứ tự tự nhiên, quá nhiều hạng mục

=> Cần chuẩn hóa dữ liệu trước khi áp dụng vào mô hình, sử dụng mã hóa one-hot hoặc giảm số lượng hạng mục

3

NGHIÊN CỨU VẤN ĐỀ

3.3. TIỀN XỬ LÝ DỮ LIỆU

Các bước tiền xử lý dữ liệu

- Chia dữ liệu thành 2 tập train/test.
- Xử lý các dữ liệu bị nhiễu.
- Loại bỏ các cột/hàng.
- Điền dữ liệu bị thiếu.
- Cân bằng dữ liệu.
- Chuẩn hoá dữ liệu.

3

NGHIÊN CỨU VẤN ĐỀ

3.3. TIỀN XỬ LÝ DỮ LIỆU

- Chia dữ liệu thành 2 tập train/test.

```
[36] df_train = df_dataset.iloc[index_train]
     df_test  = df_dataset.iloc[index_test]
     print(f'train: {len(df_train)} / test: {len(df_test)} - Tỷ lệ: {len(df_train) / len(df_dataset)}, {len(df_test) / len(df_dataset)}')
train: 210 / test: 90 - Tỷ lệ: 0.7, 0.3
```

- Xử lí dữ liệu bị nhiễu

```
#replace the value 9 to 2 in the age column
df_train["Age"] = df_train["Age"].replace(to_replace=9,value=2)
```

3

NGHIÊN CỨU VẤN ĐỀ

3.3. TIỀN XỬ LÝ DỮ LIỆU

Loại bỏ cột/hàng:

- “***nasogastric reflux PH***”, “***abdominocentesis appearance***”, “***abdomcentesis total protein***” do dữ liệu thiếu quá nhiều.
- Hospital ID” và “cp_data” vì không ảnh hưởng lớn đến kết quả.
- “Surgical lesion” loại bỏ vì sử dụng cho mục đích phân loại khác.
- Loại “***lesion type 2***” và “***lesion type 3***” sau khi phân tích ở phần khám phá dữ liệu.
- Chia cột “***lesion type 1***” thành 5 cột mới: “***site of lesion***”, “***type***”, “***subtype***”, “***specific code***”, “***none***” và sau đó loại bỏ cột gốc “***lesion type 1***”.

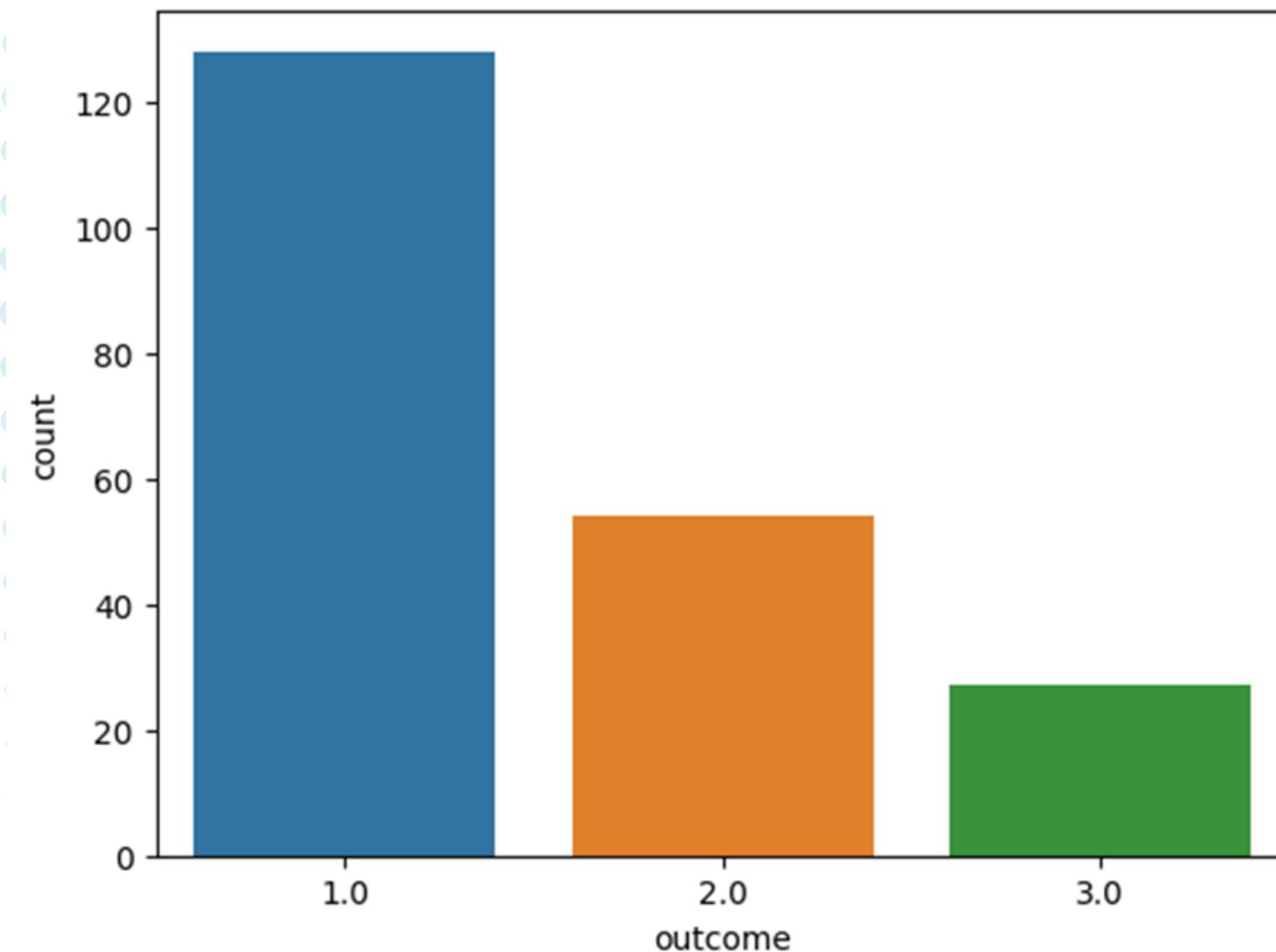
#	Column	Non-Null Count	Dtype
0	Surgery	209 non-null	float64
1	Age	210 non-null	int64
2	rectal temperature	165 non-null	float64
3	pulse	194 non-null	float64
4	respiratory rate	175 non-null	float64
5	temperature of extremities	169 non-null	float64
6	peripheral pulse	164 non-null	float64
7	mucous membranes	178 non-null	float64
8	capillary refill time	190 non-null	float64
9	pain	174 non-null	float64
10	peristalsis	178 non-null	float64
11	abdominal distension	171 non-null	float64
12	nasogastric tube	139 non-null	float64
13	nasogastric reflux	133 non-null	float64
14	rectal examination - feces	145 non-null	float64
15	abdomen	134 non-null	float64
16	packed cell volume	190 non-null	float64
17	total protein	188 non-null	float64
18	outcome	209 non-null	float64
19	site of lesion	210 non-null	float64
20	type	210 non-null	float64
21	subtype	210 non-null	float64
22	specific code	210 non-null	float64
23	none	210 non-null	float64

3

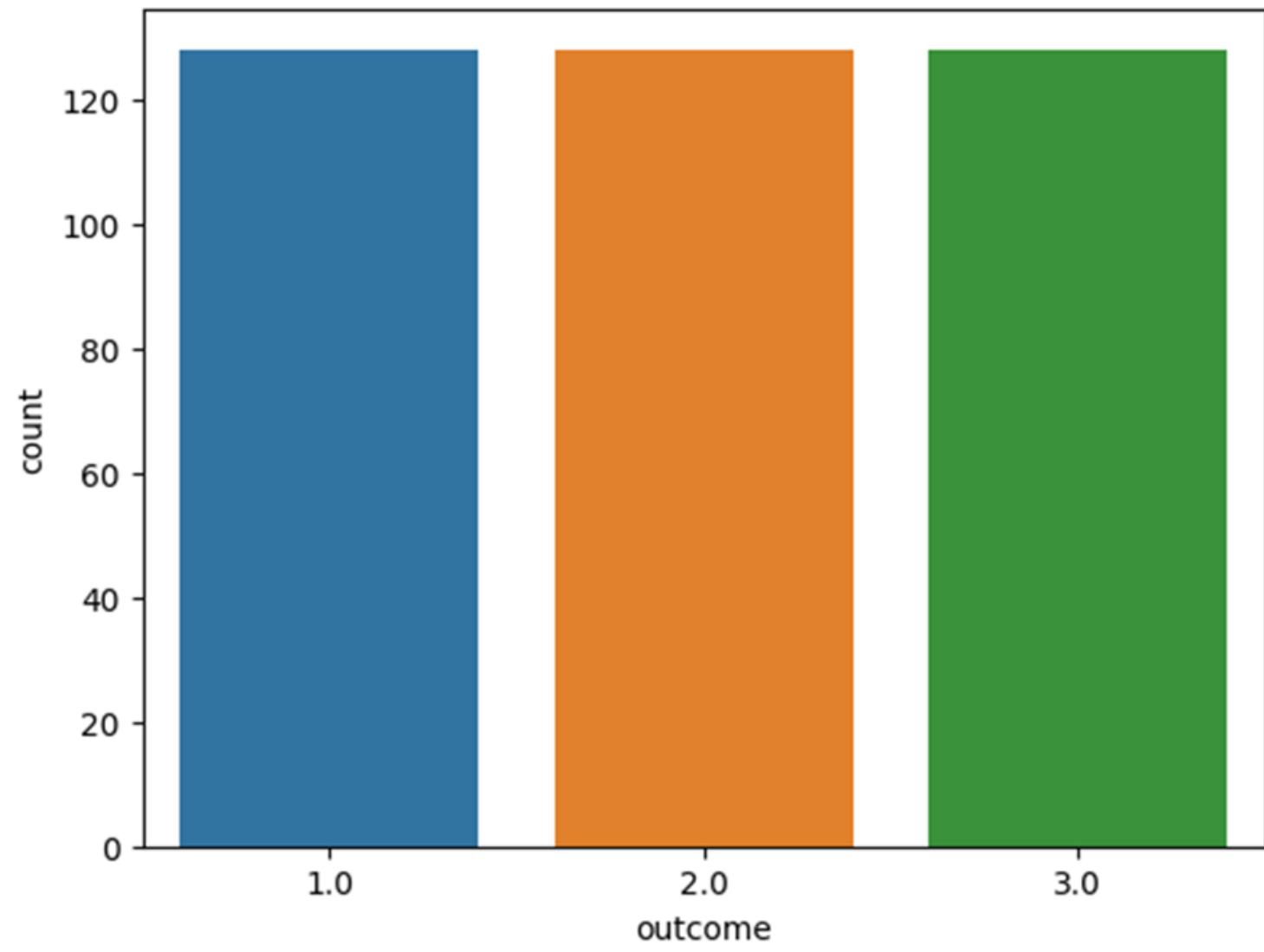
NGHIÊN CỨU VẤN ĐỀ

3.3. TIỀN XỬ LÝ DỮ LIỆU

- Cân bằng dữ liệu:



a. Dữ liệu trước khi xử lí



b. Dữ liệu sau khi xử lí

4

THỬ NGHIỆM VÀ KẾT QUẢ

4 THỬ NGHIỆM VÀ KẾT QUẢ

MÔI TRƯỜNG
THIẾT LẬP

THIẾT LẬP
HUẤN LUYỆN

- Chương trình trên máy ảo Linux cung cấp bởi Colaboratory của Google, với Python 3.10.
- Thư viện chính: sklearn
- Thủ nghiệm trên 10 mô hình:
 - (1) KNN
 - (2) Naive Bayes
 - (3) SVM
 - (4) Decision Tree
 - (5) Random Forest
 - (6) AdaBoost
 - (7) Gradient Boosting
 - (8) Linear Discriminant Analysis
 - (9) Multi-layer Perceptron
 - (10) Logistic Regression
- Sử dụng accuracy, confusion matrix so sánh hiệu suất phương pháp phân loại

4 THỬ NGHIỆM VÀ KẾT QUẢ

THIẾT LẬP HUẤN LUYỆN

Tập dữ liệu chia 2 phần train/test tỉ lệ 7/3 theo phương pháp hold-out

- Tập train:** Huấn luyện, điều chỉnh tham số với chiến lược:
- Hold-out (tiếp tục chia 7/3 với train/valid)
 - k-fold : $k=5$ (chia k phần đều nhau với $k-1$ phần cho train/ 1 phần cho valid)
 - Random_state = 42: siêu tham số ngẫu nhiên trong học máy
 - Trong đó, train: huấn luyện và valid: điều chỉnh tham số

Tập test: Kiểm nghiệm lại độ hiệu quả của thuật toán

4

THỬ NGHIỆM VÀ KẾT QUẢ

ĐỀ XUẤT CHIẾN LƯỢC

STT	Mô hình	Accuracy	F1	Precision	Recall
1	Gradient Boosting				
2	Random Forest				
3	SVM				
4	Multi-layer perceptron				
5	AdaBoost				
6	KNN				
7	Decision tree				
8	Linear discriminant analysis				
9	Logistic regression				
10	Naive Bayes				

Bảng: Mô hình đề xuất để dự báo

4

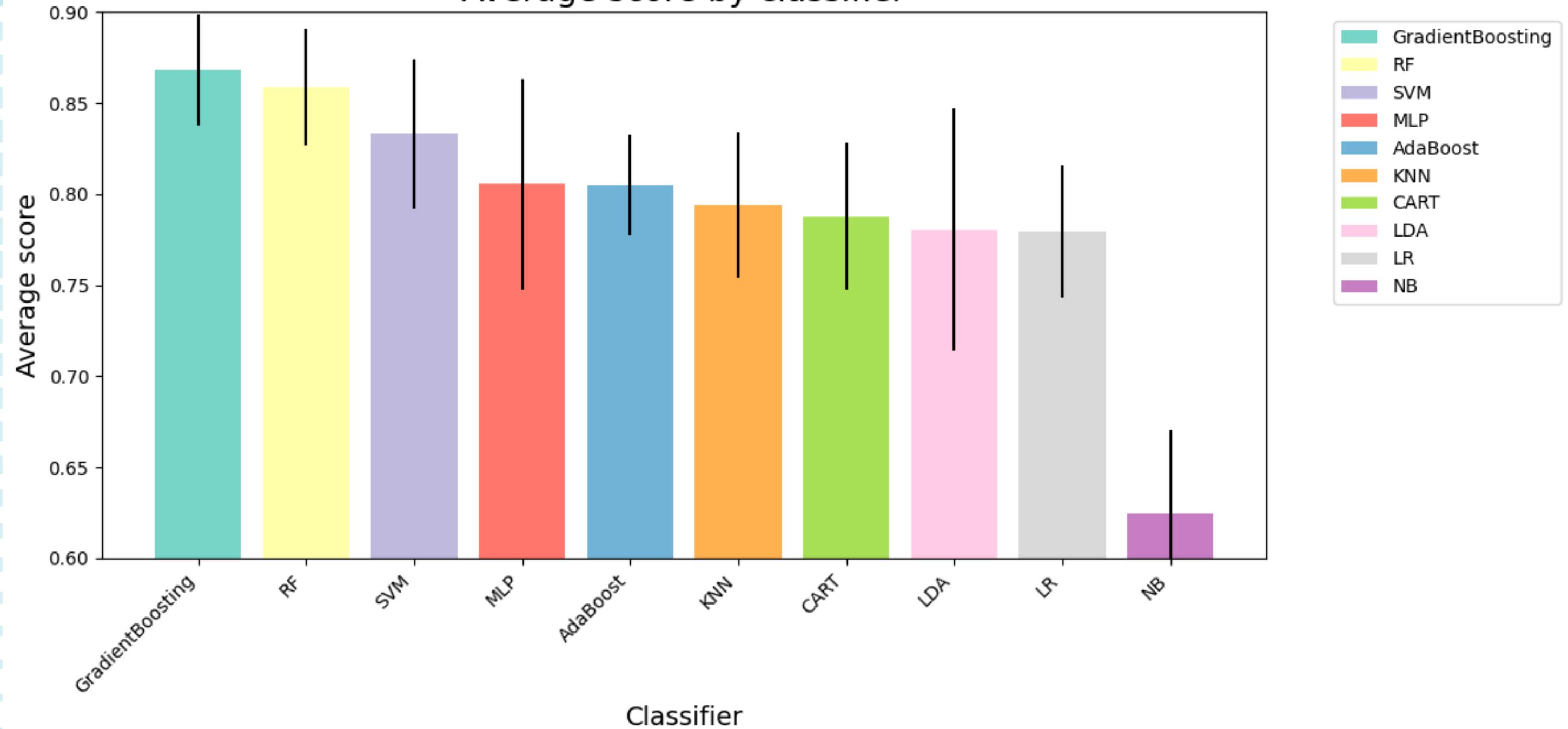
THỬ NGHIỆM VÀ KẾT QUẢ

	Classifier	Accuracy	F1	Precision	Recall	Average score	Average std
6	GradientBoosting	0.867191	0.867241	0.871693	0.867191	0.868329	0.030650
4	RF	0.856938	0.857007	0.863324	0.856938	0.858552	0.031945
2	SVM	0.830930	0.829375	0.841369	0.830930	0.833151	0.041107
8	MLP	0.804921	0.802154	0.810035	0.804921	0.805508	0.057841
5	AdaBoost	0.802016	0.803014	0.812928	0.802016	0.804994	0.027500
0	KNN	0.791866	0.783509	0.809169	0.791866	0.794103	0.039806
3	CART	0.786535	0.786931	0.791414	0.786535	0.787853	0.040238
7	LDA	0.778947	0.777254	0.787365	0.778947	0.780629	0.066744
9	LR	0.778811	0.777759	0.783188	0.778811	0.779642	0.036423
1	NB	0.612133	0.580150	0.694553	0.612133	0.624742	0.045436

4

THỬ NGHIỆM VÀ KẾT QUẢ

Average score by classifier



4 THỬ NGHIỆM VÀ KẾT QUẢ

```
tunning_models = {}
tunning_params = {}
# khởi tạo các tham số mặc định
tunning_models['RandomForest'] = RandomForestClassifier(random_state=params["random_state"])
tunning_params['RandomForest'] = {
    'n_estimators': [100, 300, 500],
    'max_depth': [None, 5, 10, 15],
    'min_samples_split': [2, 5, 10],
}

tunning_models['GradientBoosting'] = GradientBoostingClassifier(random_state=params["random_state"])
tunning_params['GradientBoosting'] = {
    'n_estimators': [100, 300, 500],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
```

4

THỬ NGHIỆM VÀ KẾT QUẢ

Kết quả:

1. RandomForest:

Kết quả tốt nhất: 0.8621326042378673

Số cây quyết định (n_estimators): 500

Độ sâu tối đa của cây (max_depth): Không giới hạn (None)

Số mẫu tối thiểu để chia một nút (min_samples_split): 2

2. Gradient Boosting:

Kết quả tốt nhất: 0.8671907040328092

Số cây quyết định (n_estimators): 100

Tốc độ học (learning_rate): 0.1

Độ sâu tối đa của cây (max_depth): 3

4

THỬ NGHIỆM VÀ KẾT QUẢ

Tunning [RandomForest]

Baseline [RF]

+ acc = 0.700 => Sau khi tinh chỉnh 0.733

+ precision = 0.699

+ recall = 0.700

+ F1-score = 0.697

Tunning [GradientBoosting]

Baseline [GradientBoosting]

+ acc = 0.700

+ precision = 0.709

+ recall = 0.700

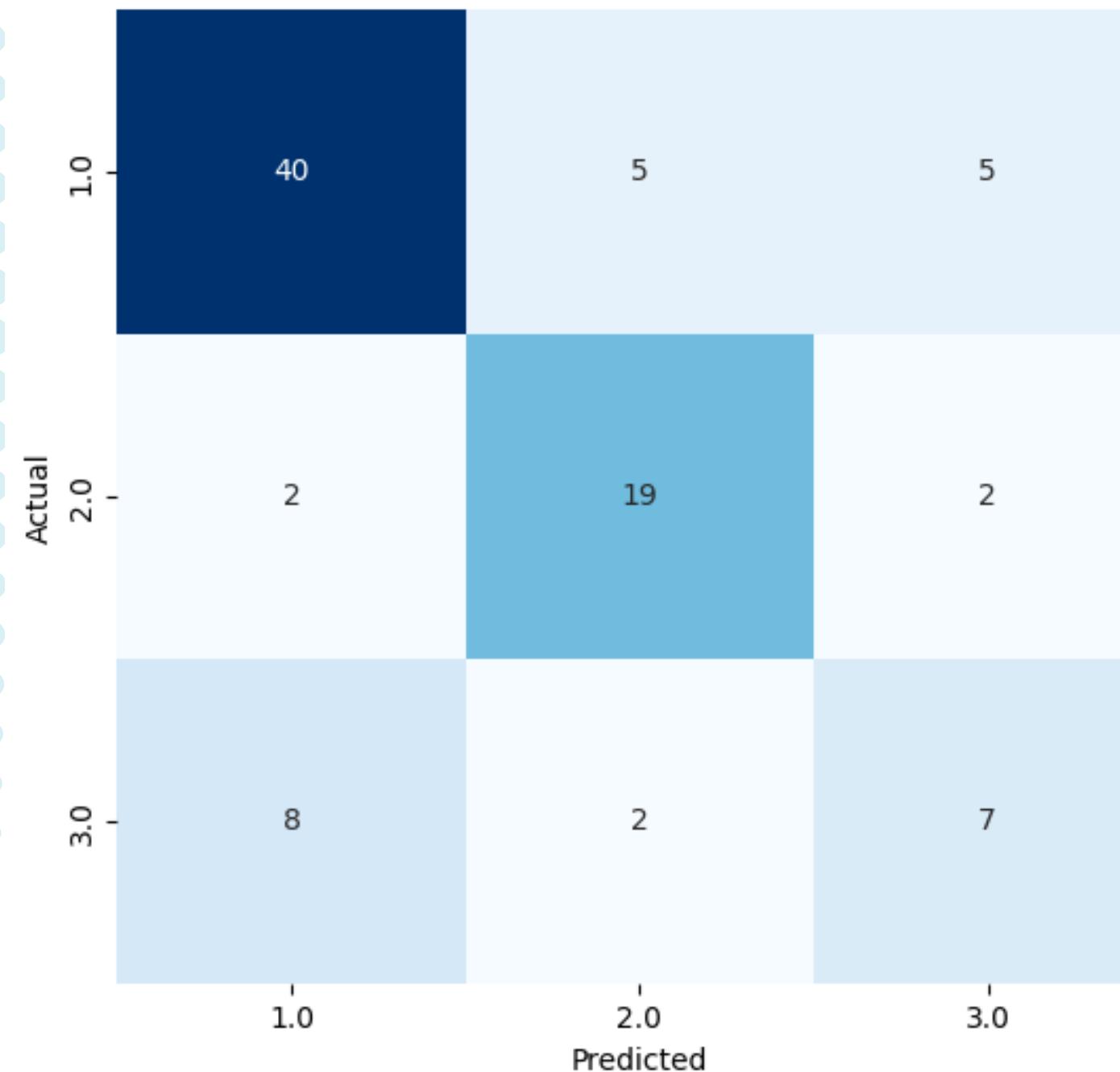
+ F1-score = 0.704



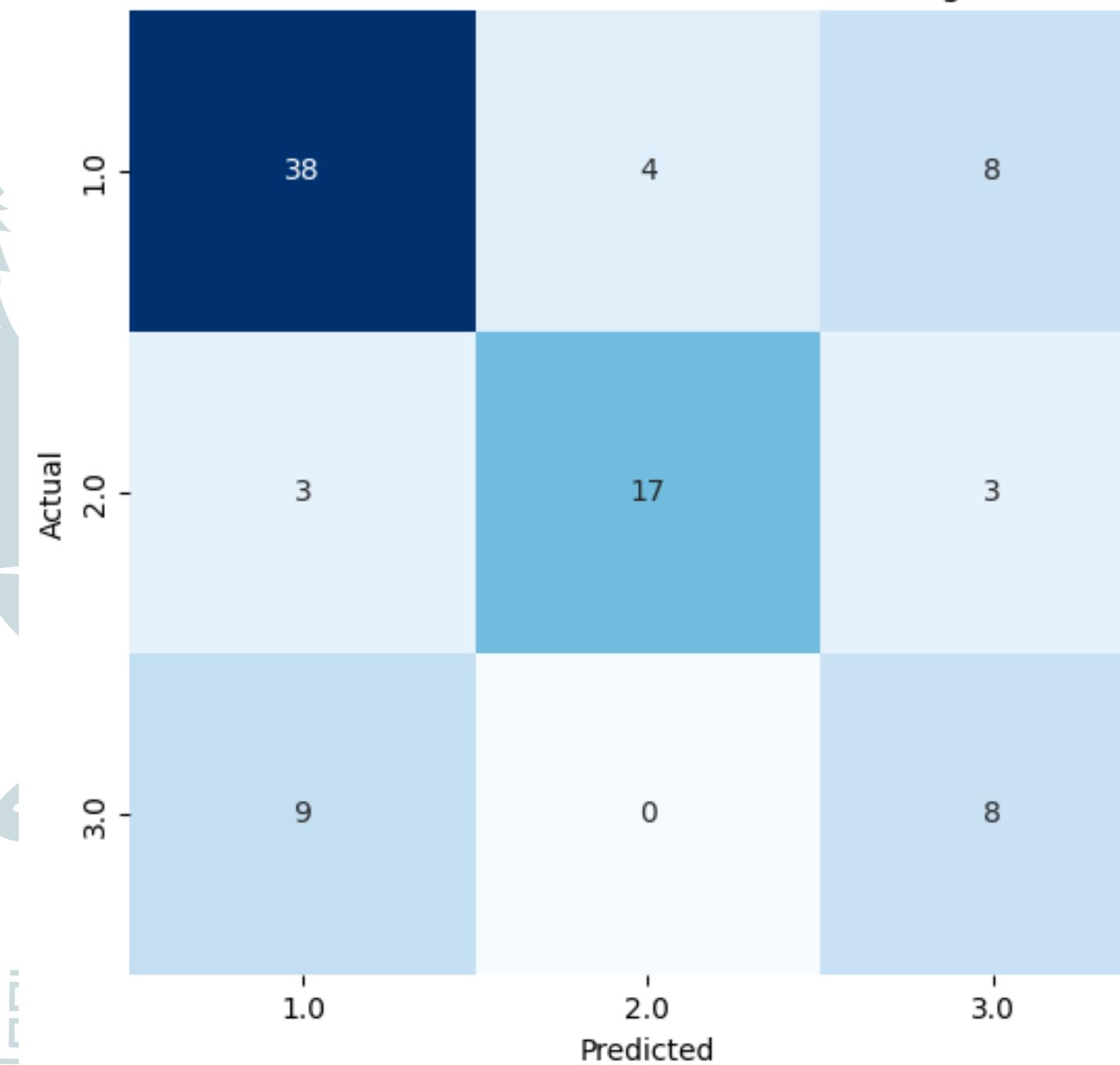
4

THỬ NGHIỆM VÀ KẾT QUẢ

Confusion Matrix - RandomForest



Confusion Matrix - GradientBoosting



5

KẾT LUẬN

Phương hướng sau khi thực hiện:

- Tinh chỉnh Tham số: Tiếp tục tinh chỉnh tham số cho cả hai mô hình để xem liệu có thể cải thiện hiệu suất hay không.
- Tổ Hợp Mô Hình (Ensemble): Xem xét việc sử dụng ensemble của RF và GB để kết hợp ưu điểm của cả hai mô hình.
- Kiểm Soát Kích Thước Mô Hình: Nếu kích thước mô hình là một vấn đề, có thể xem xét giảm số cây hoặc thử nghiệm các biện pháp kiểm soát kích thước.



Thank you