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

Nguyễn Khánh Toàn, Trần Phương Nghi, Hứa Tuấn Vĩ, Trần Gia Huy, Phạm Mai Chi
Trường Công nghệ và Thiết kế UEH

GIỚI THIỆU

"Colic" là bệnh rối loạn tiêu hóa ở ngựa. nó thường bắt đầu đột ngột nên tính nghiêm trọng của của bệnh thường tăng rất nhanh và có thể khiến ngựa tử vong.

Xây dựng **mô hình phân loại gồm 10 phương pháp** (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 và (10) logistic regression để dự đoán tình trạng bệnh của ngựa.

PHƯƠNG PHÁP

- **EDA** Countplot, Histogram, Correlogram, Heatmap
- **Tiền xử lý dữ liệu** One Hot, Label, Ordinal, Min Max Scaler, Standard Scaler

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

- **Các phương pháp đánh giá mô hình**
  accuracy, precision, recall, F1-Score và confusion matrix

XÁC ĐỊNH VẤN ĐỀ

Bộ dữ liệu : "Horse Colic", về vấn đề bệnh tiêu hóa ở ngựa
Đầu vào : 300 bản ghi, 28 thuộc tính
Đầu ra : - Kết quả xảy ra: sống/ chết
- Can thiệp phẫu thuật: giải pháp khắc phục
Mục tiêu : - Tạo mô hình dự đoán tình trạng của ngựa
- Hỗ trợ bác sĩ thú y xác định trường hợp ngựa có nguy cơ tử vong cao và can thiệp phẫu thuật

ĐÁNH GIÁ DỮ LIỆU

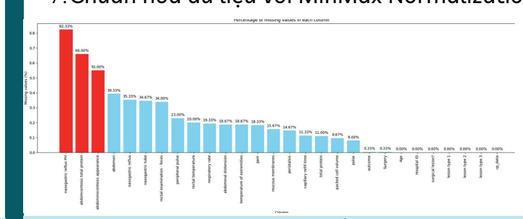
KHÁM PHÁ DỮ LIỆU

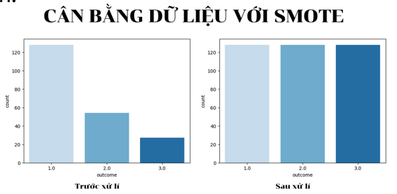
- **Hiển thị một số thông tin về dữ liệu :** 300 dòng , 28 cột, bao gồm các thuộc tính số (numeric) và thuộc tính rời rạc (nominal).
- **Kiểm tra tính toàn vẹn của dữ liệu :** bộ dữ liệu không có giá trị trùng lặp. Có rất nhiều giá trị bị thiếu ở một số cột (trên 50%).
- **Thể hiện các tính chất thống kê trên dữ liệu số :** Count, Mean, Std, Deviation, Min, 25th Percentile, 50th Percentile , 75th Percentile, Max.
- **Hiển thị dữ liệu (Visualize Data) :** hiển thị trên từng tính chất đơn, nhiều tính chất.

Đề xuất loại bỏ các cột thiếu bà không quan trọng như: nasogastric reflux PH, abdominocentesis appearance, abdomcentesis total, "Hospital ID", "cp_data" protein, "Surgical lesion", "lesion type 2"và "lesion type 3"

TIỀN XỬ LÍ DỮ LIỆU

1. Chia dữ liệu thành 70-30 cho huấn luyện và đánh giá mô hình.
2. Điều chỉnh cột "Age" từ "9" thành "2" để sửa lỗi dữ liệu.
3. Loại bỏ các cột thiếu và không quan trọng.
4. Chia cột "lesion type 1" thành 5 cột mới, sau đó loại bỏ cột gốc.
5. Xử lý dữ liệu thiếu bằng nhiều phương pháp khác nhau.
6. Áp dụng phương pháp SMOTE để cân bằng dữ liệu.
7. Chuẩn hoá dữ liệu với MinMax Normalization.





THIẾT LẬP THỬ NGHIỆM

- Thử nghiệm trên **10 mô hình:** KNN, Naive Bayes, SVM, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, Linear Discriminant Analysis, Multi-layer Perceptron, Logistic Regression
- So sánh hiệu suất phương pháp phân loại: accuracy, precision, recall, F1-Score và confusion matrix
- Bộ dữ liệu chia 2 phần train/test (7/3) theo phương pháp hold-out
  - Tập train: Chiến lược: Hold-out (chia 7/3 với train/valid); k-fold: k=5; Random_state = 42
  - Tập test: Kiểm nghiệm lại độ hiệu quả của thuật toán

KẾT QUẢ

STT	Mô hình	Accuracy	F1	Precision	Recall	Avg score	Avg Std
6	GradientBoosting	0.867	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

1. **Mô hình Random Forest và Gradient Boosting đạt hiệu suất tốt (0.85,0.867)**
2. Sau khi tinh chỉnh: đều cho ra kết quả khá tốt ở tập test: 0.7 và 0.733
3. Sử dụng GridSearchCV để tinh chỉnh dữ liệu và tìm ra kết quả tốt nhất cho Gradient Boosting và Random Forest
4. Đối với ma trận nhầm lẫn thì ở cả 2 mô hình đều cho kết quả phân loại 1.0: sống, tốt hơn 2 lớp còn lại là chết và trợ tử (2.0 và 3.0)

KẾT LUẬN

Cả 2 mô hình đều đạt được kết quả tốt, nhưng không có sự chênh lệch đáng kể giữa RF và GB. Sự chọn lựa giữa RF và GB có thể phụ thuộc vào các yếu tố khác nhau như tốc độ huấn luyện, khả năng giải thích, và sự phức tạp của mô hình.

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 1 vấn đề, có thể xem xét giảm số cây, thử nghiệm các biện pháp kiểm soát kích thước.

TÀI LIỆU THAM KHẢO

[1] Mary McLeish", "Horse Colic - UCI Machine Learning Repository".
[2] Jason Brownlee, "Machine Learning Mastery with Python: Understand Your Data Create Accurate Models and Work Projects End-to-End".
[3] Jason Brownlee, "Data Preparation for Machine Learning".