ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

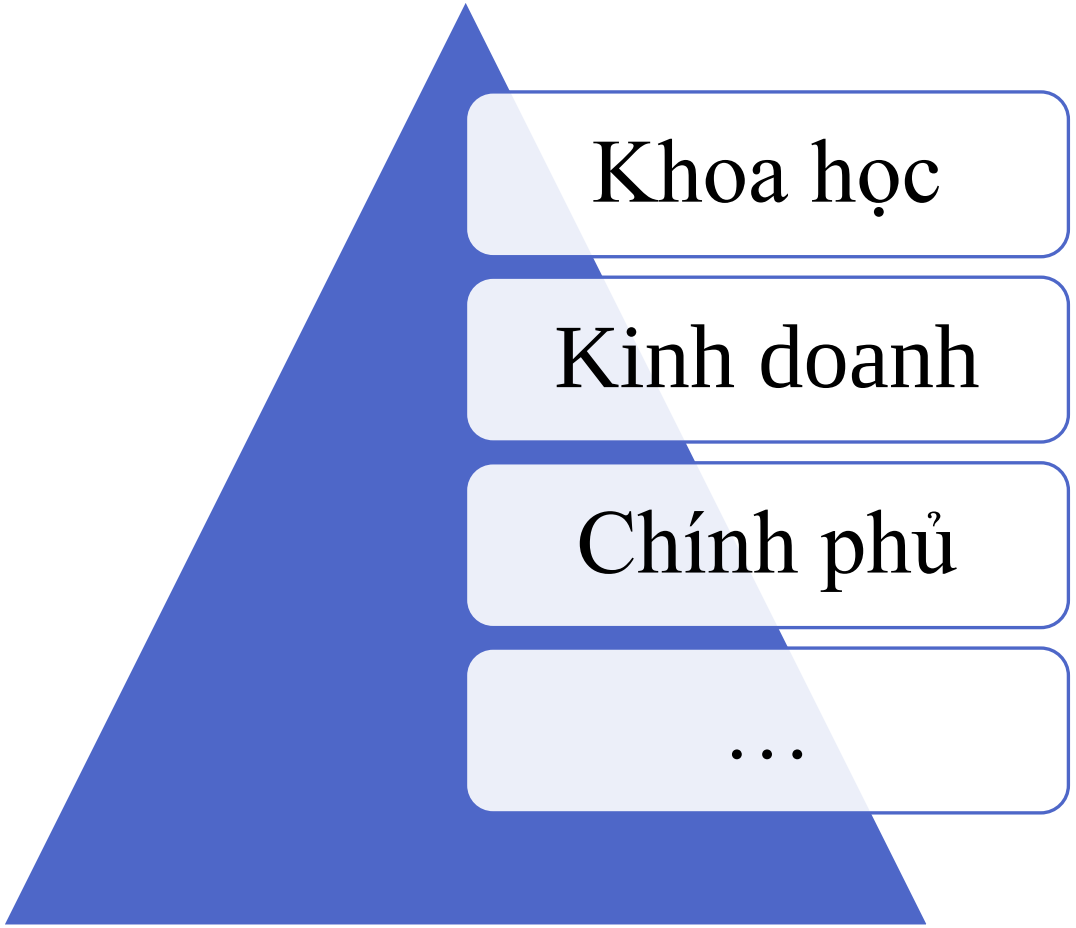
CHƯƠNG 7
Hệ thống phân loại cho doanh nghiệp

Biên soạn: ThS. Nguyễn Thị Anh Thư

Nội dung

1. Giới thiệu
2. Quy trình triển khai
3. Lựa chọn thuật toán
4. Đánh giá hiệu quả
5. Triển khai hệ thống
6. Giám sát và bảo trì
7. Bài tập
8. Tổng kết

1. Giới thiệu



Hệ thống phân loại là một hệ thống được sử dụng để sắp xếp các đối tượng vào các nhóm dựa trên các đặc điểm chung.

Hệ thống phân loại có thể được áp dụng trong nhiều lĩnh vực khác nhau.

Phân loại (Classification)

Học có giám sát (Supervised Learning)

Phân loại (Classification)

Được sử dụng để dự đoán nhãn (label) cho các dữ liệu mới dựa trên các dữ liệu đã được dán nhãn sẵn (training data).

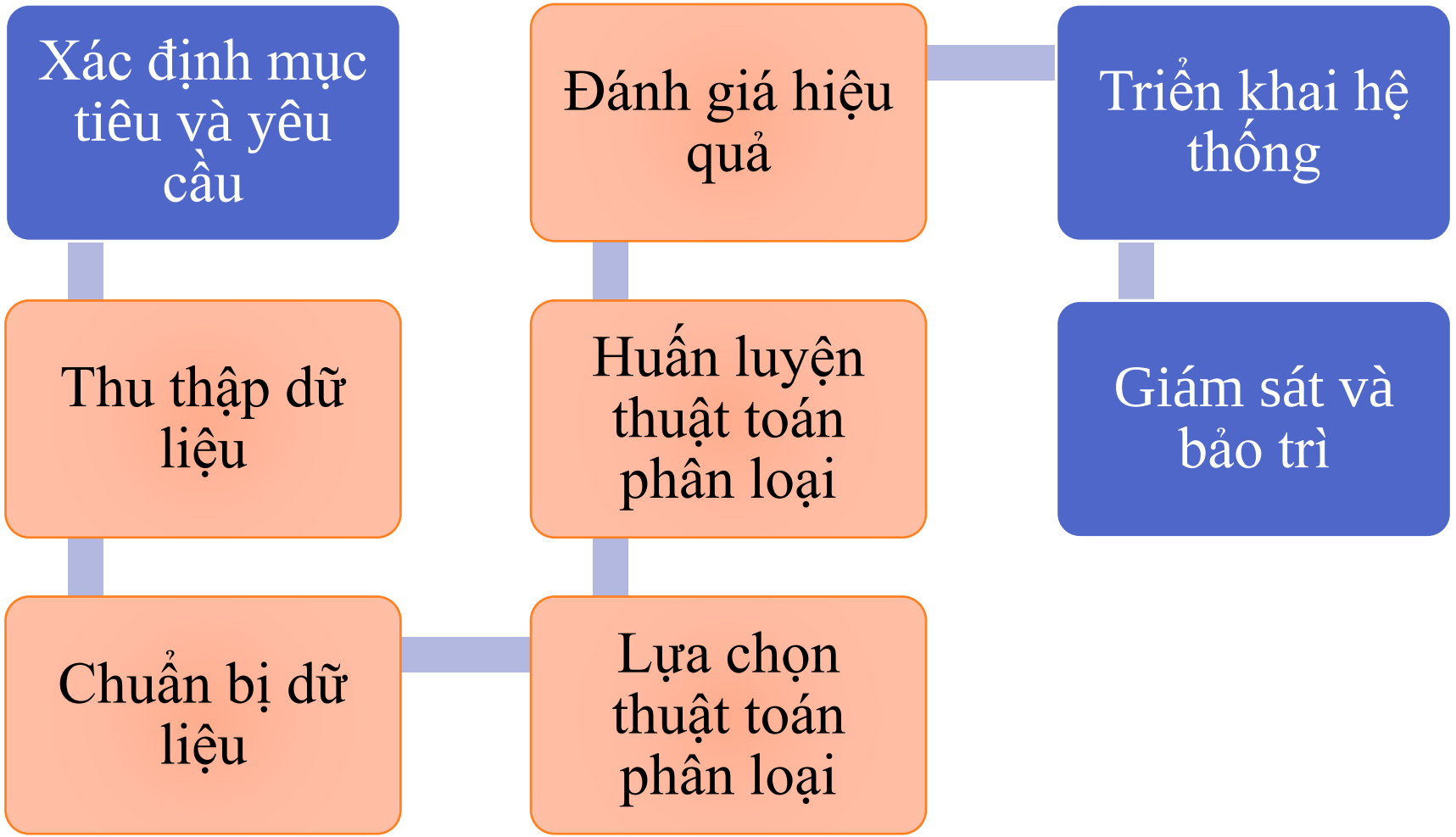
Có hai loại chính:

- **Phân loại nhị phân**: Dự đoán dữ liệu thuộc một trong hai lớp (ví dụ: email spam hay không spam, khách hàng tiềm năng hay không tiềm năng).
- **Phân loại đa lớp**: Dự đoán dữ liệu thuộc một trong nhiều lớp (ví dụ: loại hoa, loại bệnh ung thư).

1. Giới thiệu

Hệ thống phân loại email	• Hệ thống phân loại email sử dụng thuật toán phân loại để phân loại email vào các thư mục khác nhau, chẳng hạn như Hộp thư đến, Spam và Thùng rác.
Hệ thống phân loại hình ảnh	• Hệ thống phân loại hình ảnh sử dụng thuật toán phân loại để phân loại hình ảnh vào các lớp khác nhau, chẳng hạn như chó, mèo và xe cộ.
Hệ thống phân loại tín dụng	• Hệ thống phân loại tín dụng sử dụng thuật toán phân loại để đánh giá khả năng trả nợ của người vay.

2. Quy trình triển khai



2. Quy trình triển khai

1. Xác định mục tiêu và yêu cầu

Bước đầu tiên là xác định mục tiêu và yêu cầu của hệ thống phân loại. Cần xác định rõ ràng loại dữ liệu muốn phân loại, các lớp phân loại và độ chính xác mong muốn.

2. Thu thập dữ liệu

Bước tiếp theo là thu thập dữ liệu để huấn luyện thuật toán phân loại. Dữ liệu thu thập cần phải đa dạng và đại diện cho các lớp phân loại mà hệ thống phân loại có thể phân biệt được.

3. Chuẩn bị dữ liệu

Dữ liệu thu thập được cần phải được chuẩn bị trước khi huấn luyện thuật toán phân loại.

2. Quy trình triển khai

4. Lựa chọn thuật toán phân loại

Có nhiều thuật toán phân loại khác nhau có thể được sử dụng. Lựa chọn thuật toán phân loại phù hợp phụ thuộc vào nhiều yếu tố, chẳng hạn như loại dữ liệu, kích thước dữ liệu và độ chính xác mong muốn.

5. Huấn luyện thuật toán phân loại

Thuật toán phân loại được huấn luyện trên tập huấn luyện. Quá trình huấn luyện có thể mất nhiều thời gian, tùy thuộc vào kích thước dữ liệu và độ phức tạp của thuật toán.

6. Đánh giá hiệu quả

Hiệu quả của thuật toán phân loại được đánh giá trên tập kiểm tra và tập xác thực. Nếu hiệu quả không đạt yêu cầu, cần điều chỉnh các tham số của thuật toán hoặc thử một thuật toán khác.

2. Quy trình triển khai

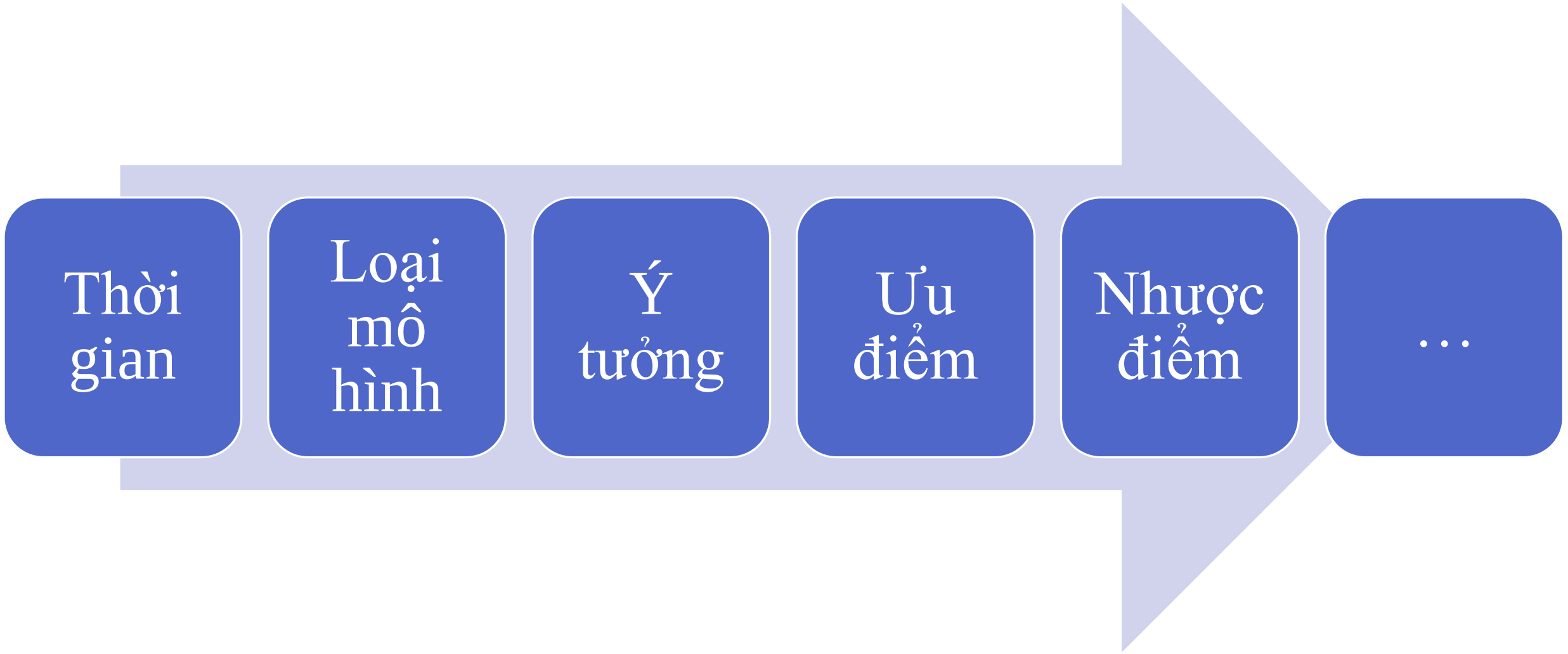
7. Triển khai hệ thống

Hệ thống phân loại được triển khai sau khi thuật toán phân loại được huấn luyện và đánh giá hiệu quả. Hệ thống phân loại có thể được triển khai dưới dạng một ứng dụng web, một ứng dụng di động, hoặc một dịch vụ API.

8. Giám sát và bảo trì

Hệ thống phân loại cần được giám sát và bảo trì để đảm bảo hiệu quả hoạt động. Cần cập nhật dữ liệu huấn luyện và đánh giá hiệu quả của hệ thống phân loại định kỳ.

3. Lựa chọn thuật toán



3. Lựa chọn thuật toán

Đặc điểm	K-Nearest Neighbors (KNN)
Thời gian	**Ra mắt:** 1950s **Tốc độ huấn luyện:** Nhanh **Tốc độ dự đoán:** Nhanh với K nhỏ, chậm với K lớn
Loại mô hình	Học máy dựa trên ví dụ. Không học mô hình từ dữ liệu. Dự đoán dựa trên K điểm gần nhất.
Ý tưởng	Dựa vào "kẻ láng giềng gần nhất" để dự đoán. Tìm K điểm dữ liệu gần nhất với điểm dữ liệu mới. Gán nhãn cho điểm dữ liệu mới dựa trên nhãn của K điểm gần nhất
Ưu điểm	Đơn giản, dễ hiểu. Hiệu quả với tập dữ liệu nhỏ. Ít yêu cầu về dữ liệu. Linh hoạt với nhiều loại dữ liệu. Khả năng giải thích tốt.
Nhược điểm	Tốc độ dự đoán chậm với tập dữ liệu lớn. Nhạy cảm với nhiễu. Khó lựa chọn giá trị K. Khả năng khái quát hạn chế. Kích thước dữ liệu ảnh hưởng đến hiệu quả.

3. Lựa chọn thuật toán

Đặc điểm	Support Vector Machines (SVM)
Thời gian	**Ra mắt:** 1998 **Tốc độ huấn luyện:** Chậm, đặc biệt với tập dữ liệu lớn **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên mô hình. Học mô hình hyperplane phân chia các lớp dữ liệu. Tìm hyperplane với margin lớn nhất.
Ý tưởng	Tìm đường phân chia (hyperplane) tốt nhất để phân chia các lớp dữ liệu. Margin là khoảng cách từ hyperplane đến các điểm dữ liệu gần nhất. Hyperplane với margin lớn nhất giúp phân chia các lớp dữ liệu rõ ràng nhất.
Ưu điểm	Hiệu quả cao với tập dữ liệu có nhiều thuộc tính. Khả năng khái quát tốt. Ít yêu cầu về dữ liệu. Ít bị ảnh hưởng bởi nhiễu. Có thể xử lý dữ liệu không tuyến tính bằng cách sử dụng kernel.
Nhược điểm	Tốc độ huấn luyện chậm, đặc biệt với tập dữ liệu lớn. Khó lựa chọn tham số. Khả năng giải thích hạn chế. Không hiệu quả với dữ liệu có nhiều lớp.

3. Lựa chọn thuật toán

Đặc điểm	Naive Bayes
Thời gian	**Ra mắt:** 1953 **Tốc độ huấn luyện:** Nhanh **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên xác suất. Sử dụng định lý Bayes để dự đoán. Giả định các thuộc tính độc lập.
Ý tưởng	Dựa trên định lý Bayes để tính toán xác suất một mẫu dữ liệu thuộc về một lớp. Giả định các thuộc tính của dữ liệu độc lập với nhau. Lớp có xác suất cao nhất được chọn.
Ưu điểm	Đơn giản, dễ hiểu. Hiệu quả với tập dữ liệu nhỏ. Ít yêu cầu về dữ liệu. Tốc độ huấn luyện và dự đoán nhanh. Khả năng giải thích tốt.
Nhược điểm	Giả định độc lập giữa các thuộc tính thường không đúng. Khả năng khái quát hạn chế. Không hiệu quả với dữ liệu có nhiễu.

3. Lựa chọn thuật toán

Đặc điểm	Decision Trees
Thời gian	**Ra mắt:** 1957 **Tốc độ huấn luyện:** Nhanh **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên quy tắc. Học mô hình dạng cây để phân chia dữ liệu. Dựa trên các câu hỏi để đưa ra dự đoán.
Ý tưởng	Xây dựng một cây quyết định để phân chia dữ liệu thành các lớp khác nhau. Cây bao gồm các nút (node) và nhánh (branch). Mỗi nút đại diện cho một câu hỏi về một thuộc tính của dữ liệu. Mỗi nhánh đại diện cho một câu trả lời cho câu hỏi. Dữ liệu được phân chia theo các nhánh cho đến khi đến lá (leaf). Lớp của lá được chọn là dự đoán cho dữ liệu.
Ưu điểm	Đơn giản, dễ hiểu. Dễ giải thích. Hiệu quả với tập dữ liệu nhỏ. Ít yêu cầu về dữ liệu.
Nhược điểm	Khả năng khái quát hạn chế. Dễ bị quá khớp (overfitting). Khó lựa chọn điểm cắt (threshold). Kích thước cây có thể lớn

3. Lựa chọn thuật toán

Đặc điểm	Random Forest
Thời gian	**Ra mắt:** 2001 **Tốc độ huấn luyện:** Chậm hơn Decision Trees **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên tập hợp. Kết hợp nhiều Decision Trees để tạo ra một mô hình mạnh mẽ hơn. Sử dụng phương pháp bootstrap để tạo ra các tập dữ liệu con.
Ý tưởng	Tạo ra nhiều Decision Trees (cây quyết định) khác nhau. Mỗi cây được huấn luyện trên một tập dữ liệu con được tạo ra bằng phương pháp bootstrap. Dự đoán của mô hình là kết quả đa số (majority vote) của các cây.
Ưu điểm	Hiệu quả cao với nhiều loại dữ liệu. Khả năng khái quát tốt. Chống quá khớp (overfitting). Dễ dàng điều chỉnh tham số. Khả năng giải thích tốt.
Nhược điểm	Tốc độ huấn luyện chậm hơn Decision Trees. Khó lựa chọn số lượng cây. Kích thước mô hình có thể lớn.

3. Lựa chọn thuật toán

Đặc điểm	Gradient Boosting
Thời gian	**Ra mắt:** 1999 **Tốc độ huấn luyện:** Chậm hơn Decision Trees **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên tập hợp. Kết hợp nhiều Decision Trees (Cây Quyết Định) để tạo ra một mô hình mạnh mẽ hơn. Sử dụng thuật toán Gradient Descent để tối ưu hóa mô hình.
Ý tưởng	Tạo ra nhiều Decision Trees (Cây Quyết Định) theo trình tự. Mỗi cây được huấn luyện để sửa lỗi của các cây trước đó. Dự đoán của mô hình là tổng hợp của các dự đoán của các cây
Ưu điểm	Hiệu quả cao với nhiều loại dữ liệu. Khả năng khái quát tốt. Chống quá khớp (overfitting). Có thể xử lý dữ liệu không tuyến tính. Khả năng giải thích tốt.
Nhược điểm	Tốc độ huấn luyện chậm hơn Decision Trees. Khó lựa chọn số lượng cây và tham số. Kích thước mô hình có thể lớn.

3. Lựa chọn thuật toán

Đặc điểm	XGBoost
Thời gian	**Ra mắt:** 2014 **Tốc độ huấn luyện:** Nhanh hơn Gradient Boosting **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên tập hợp. Là một triển khai hiệu quả của thuật toán Gradient Boosting. Sử dụng các kỹ thuật tối ưu hóa để tăng tốc độ huấn luyện và hiệu suất.
Ý tưởng	Tương tự như Gradient Boosting, XGBoost tạo ra nhiều Decision Trees (Cây Quyết Định) theo trình tự. Sử dụng thuật toán Regularization để chống quá khớp (overfitting). Sử dụng thuật toán Parallel Computing để tăng tốc độ huấn luyện.
Ưu điểm	Hiệu quả cao với nhiều loại dữ liệu. Khả năng khái quát tốt. Chống quá khớp (overfitting). Tốc độ huấn luyện nhanh hơn Gradient Boosting. Có thể xử lý dữ liệu không tuyến tính. Khả năng giải thích tốt.
Nhược điểm	Khó lựa chọn tham số. Kích thước mô hình có thể lớn.

3. Lựa chọn thuật toán

Đặc điểm	LightGBM
Thời gian	**Ra mắt:** 2017 **Tốc độ huấn luyện:** Nhanh hơn XGBoost **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên tập hợp. Là một triển khai hiệu quả của thuật toán Gradient Boosting. Sử dụng các kỹ thuật tối ưu hóa để tăng tốc độ huấn luyện và hiệu suất.
Ý tưởng	Tương tự như XGBoost, LightGBM tạo ra nhiều Decision Trees (Cây Quyết Định) theo trình tự. Sử dụng thuật toán Gradient-based Decision Tree (GDBT) để tối ưu hóa mô hình. Sử dụng thuật toán Parallel Computing để tăng tốc độ huấn luyện.
Ưu điểm	Hiệu quả cao với nhiều loại dữ liệu. Khả năng khái quát tốt. Chống quá khớp (overfitting). Tốc độ huấn luyện nhanh hơn XGBoost. Có thể xử lý dữ liệu không tuyến tính. Khả năng giải thích tốt.
Nhược điểm	Khó lựa chọn tham số. Kích thước mô hình có thể lớn.

3. Lựa chọn thuật toán

Đặc điểm	CatBoost
Thời gian	**Ra mắt:** 2017 **Tốc độ huấn luyện:** Nhanh hơn XGBoost và LightGBM **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên tập hợp. Là một triển khai hiệu quả của thuật toán Gradient Boosting. Sử dụng các kỹ thuật tối ưu hóa để tăng tốc độ huấn luyện và hiệu suất. Sử dụng Category-aware features.
Ý tưởng	Tương tự như XGBoost và LightGBM, CatBoost tạo ra nhiều Decision Trees (Cây Quyết Định) theo trình tự. Sử dụng thuật toán Gradient-based Decision Tree (GDBT) để tối ưu hóa mô hình. Sử dụng thuật toán Parallel Computing để tăng tốc độ huấn luyện. Sử dụng Category-aware features để xử lý dữ liệu phi số.
Ưu điểm	Hiệu quả cao với nhiều loại dữ liệu. Khả năng giải thích tốt. Khả năng khái quát tốt. Chống quá khớp (overfitting). Tốc độ huấn luyện nhanh hơn XGBoost và LightGBM. Có thể xử lý dữ liệu không tuyến tính và dữ liệu phi số.
Nhược điểm	Khó lựa chọn tham số. Kích thước mô hình có thể lớn.

3. Lựa chọn thuật toán

Đặc điểm	CNN (Convolutional Neural Network)
Thời gian	**Ra mắt:** 1980s **Tốc độ huấn luyện:** Nhanh hơn so với các mô hình học máy truyền thống **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên mô hình thống kê. Lấy cảm hứng từ cấu trúc của não bộ con người. Gồm nhiều lớp nơ-ron convolutional kết nối với nhau. Mỗi nơ-ron convolutional thực hiện một phép biến đổi phi tuyến.
Ý tưởng	Học mối quan hệ phi tuyến giữa dữ liệu đầu vào (hình ảnh) và đầu ra (nhãn). Tự động trích xuất đặc trưng từ hình ảnh.
Ưu điểm	Hiệu quả cao với dữ liệu hình ảnh. Khả năng khái quát tốt. Chống quá khớp (overfitting). Có thể xử lý dữ liệu không tuyến tính. Khả năng giải thích tốt.
Nhược điểm	Yêu cầu lượng dữ liệu lớn để huấn luyện hiệu quả. Khó lựa chọn tham số. Kích thước mô hình có thể lớn. Tốn nhiều tài nguyên tính toán.

3. Lựa chọn thuật toán

Đặc điểm	RNN (Recurrent Neural Network)
Thời gian	**Ra mắt:** 1980s **Tốc độ huấn luyện:** Chậm hơn so với các mô hình học máy truyền thống **Tốc độ dự đoán:** Nhanh
Loại mô hình	Học máy dựa trên mô hình thống kê. Lấy cảm hứng từ cấu trúc của não bộ con người. Gồm nhiều lớp nơ-ron recurrent kết nối với nhau. Mỗi nơ-ron recurrent thực hiện một phép biến đổi phi tuyến.
Ý tưởng	Học mối quan hệ tuần tự giữa các phần tử trong dữ liệu. Khả năng ghi nhớ thông tin từ quá khứ. Khả năng dự đoán tương lai.
Ưu điểm	Hiệu quả cao với dữ liệu tuần tự. Khả năng khái quát tốt. Chống quá khớp (overfitting). Có thể xử lý dữ liệu không tuyến tính. Khả năng giải thích tốt.
Nhược điểm	Yêu cầu lượng dữ liệu lớn để huấn luyện hiệu quả. Khó lựa chọn tham số. Kích thước mô hình có thể lớn. Tốn nhiều tài nguyên tính toán.

4. Đánh giá hiệu quả

Phân loại nhị phân

Phân loại đa lớp

Phân loại nhị phân

Lớp tích cực (Positive) và tiêu cực (Negative) trong phân loại nhị phân là hai loại phân loại mà mô hình học máy sẽ dự đoán cho mỗi mẫu dữ liệu đầu vào.

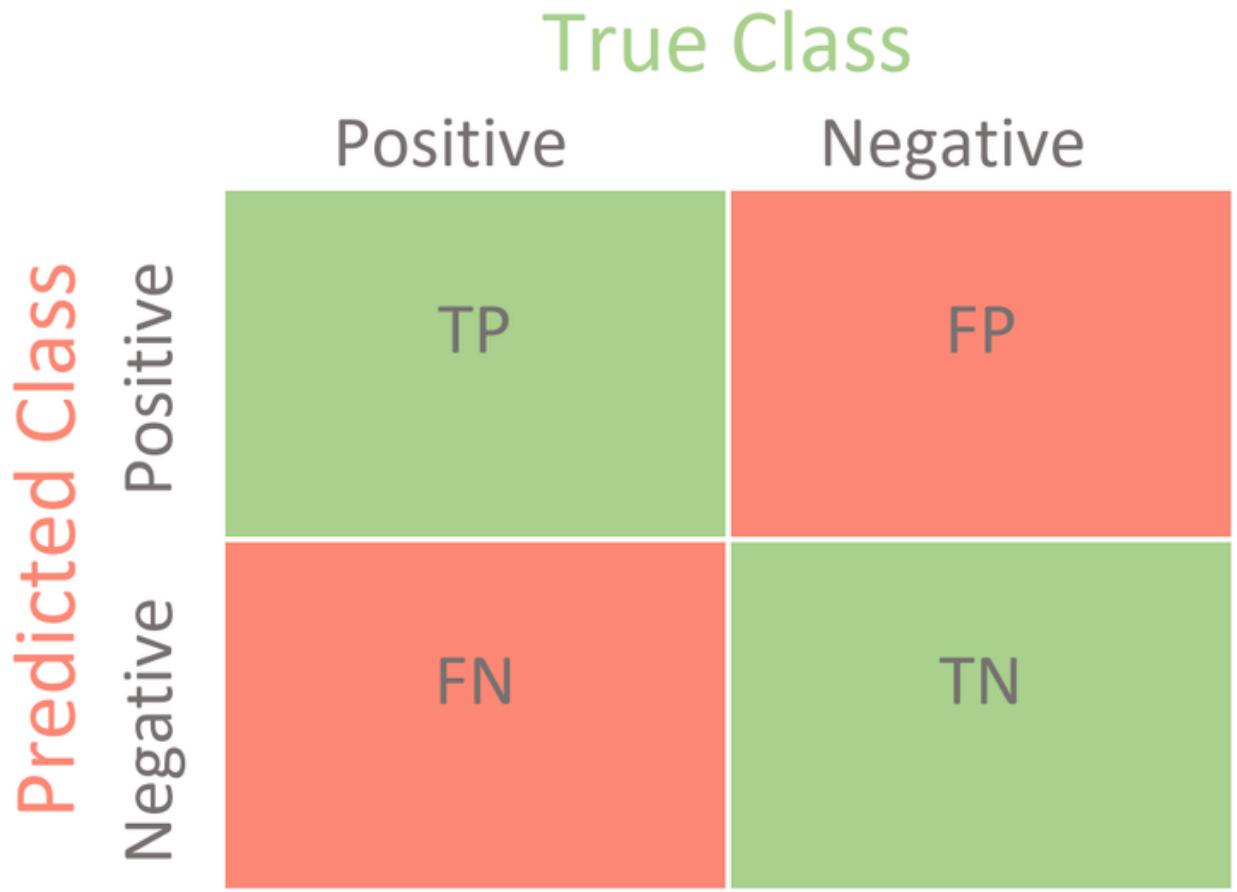
Lớp tích cực: Thể hiện **kết quả mong muốn** hoặc **sự kiện quan trọng** mà mô hình muốn dự đoán.

- **Mắc bệnh** trong mô hình dự đoán bệnh.
- **Có khả năng thanh toán** trong mô hình đánh giá tín dụng.
- **Là thư rác** trong mô hình lọc thư rác.

Lớp tiêu cực: Thể hiện **kết quả ngược lại** với lớp tích cực.

- **Không mắc bệnh** trong mô hình dự đoán bệnh.
- **Không có khả năng thanh toán** trong mô hình đánh giá tín dụng.
- **Là thư hợp lệ** trong mô hình lọc thư rác.

Việc xác định lớp tích cực và tiêu cực phụ thuộc vào mục tiêu cụ thể của bài toán phân loại.



Phân loại nhị phân

Ma trận nhầm lẫn (Confusion Matrix)

Thể hiện số lượng các dự đoán chính xác và không chính xác cho từng lớp, giúp ta hình dung hiệu quả mô hình một cách trực quan.

Phân loại nhị phân

Độ chính xác (Accuracy): (TP + TN) / (TP + FN + FP + TN)

Ý nghĩa:

- Giá trị độ chính xác cho biết tỷ lệ phần trăm các mẫu được mô hình dự đoán chính xác.
- Giá trị càng cao thể hiện mô hình càng hiệu quả trong việc phân loại các mẫu.

Lưu ý:

- **Độ chính xác có thể bị ảnh hưởng bởi sự mất cân bằng dữ liệu (data imbalance).** Ví dụ, nếu tập dữ liệu có 90% mẫu thuộc lớp A và 10% mẫu thuộc lớp B, mô hình luôn dự đoán tất cả các mẫu là A sẽ đạt độ chính xác 90%, nhưng thực tế mô hình không hề phân loại được gì.
- **Độ chính xác không cung cấp thông tin về khả năng phân biệt các lớp của mô hình.** Ví dụ, mô hình có thể dự đoán chính xác 80% các mẫu, nhưng trong số đó 70% là dự đoán "đúng" do may mắn (ngẫu nhiên dự đoán đúng lớp đa số) và chỉ 10% là dự đoán "đúng" do mô hình thực sự phân biệt được các lớp.

Phân loại nhị phân

Độ nhạy (Sensitivity) / Tỷ lệ thu hồi (Recall): TP / (TP + FN)

Ý nghĩa:

- Giá trị độ nhạy cho biết tỷ lệ phần trăm các mẫu thuộc lớp "tích cực" được mô hình dự đoán chính xác.
- Giá trị càng cao thể hiện mô hình càng hiệu quả trong việc phát hiện các mẫu thuộc lớp "tích cực".

Ví dụ:

Giả sử ta có mô hình phân loại để dự đoán bệnh nhân có mắc bệnh ung thư hay không.

- TP: Số lượng bệnh nhân thực sự mắc ung thư và được dự đoán mắc ung thư.
- FN: Số lượng bệnh nhân thực sự mắc ung thư nhưng được dự đoán không mắc ung thư.

Độ nhạy cao cho biết mô hình ít bỏ sót các trường hợp ung thư (FN thấp), giúp đảm bảo không có bệnh nhân nào bị bỏ sót việc điều trị.

Phân loại nhị phân

Độ đặc hiệu (Specificity): TN / (TN + FP)

Ý nghĩa:

- Giá trị độ đặc hiệu cho biết tỷ lệ phần trăm các mẫu thuộc lớp "tiêu cực" được mô hình dự đoán chính xác.
- Giá trị càng cao thể hiện mô hình càng hiệu quả trong việc loại trừ các mẫu thuộc lớp "tiêu cực".

Ví dụ:

Giả sử ta có mô hình phân loại để dự đoán bệnh nhân có mắc bệnh ung thư hay không.

- TN: Số lượng người thực sự không mắc ung thư và được dự đoán không mắc ung thư.
- FP: Số lượng người thực sự không mắc ung thư nhưng được dự đoán mắc ung thư.

Độ đặc hiệu cao cho biết mô hình ít dự đoán sai các trường hợp không mắc bệnh là ung thư (FP thấp), giúp đảm bảo không có người khỏe mạnh nào bị chẩn đoán sai và điều trị không cần thiết.

Phân loại nhị phân

Giá trị dự đoán dương (Positive Predictive Value) / Độ chính xác (Precision):

TP / (TP + FP)

Ý nghĩa:

- **Giá trị PPV cao** cho biết mô hình dự đoán chính xác các trường hợp dương tính.
- **Giá trị PPV thấp** cho biết mô hình dự đoán nhiều trường hợp âm tính giả là dương tính.

PPV/Precision là một thước đo quan trọng trong các trường hợp:

- **Chi phí của việc dự đoán sai cao**. Ví dụ, trong chẩn đoán y tế, việc dự đoán một người khỏe mạnh mắc bệnh có thể dẫn đến các xét nghiệm và điều trị không cần thiết, gây tốn kém và ảnh hưởng tâm lý.
- **Có nhiều trường hợp âm tính hơn trường hợp dương tính**. Ví dụ, trong việc phát hiện gian lận, số lượng giao dịch hợp pháp (âm tính) thường cao hơn nhiều so với số lượng giao dịch gian lận (dương tính).

Phân loại nhị phân

Giá trị dự đoán âm (Negative Predictive Value): TN / (FN + TN)

Ý nghĩa:

- NPV thể hiện tỷ lệ **mẫu được dự đoán là "tiêu cực" thực sự là "tiêu cực"**.
- Giá trị NPV càng **cao** thể hiện mô hình càng **ít dự đoán sai** các trường hợp thuộc lớp "tích cực" là "tiêu cực".

Lưu ý:

- NPV **có thể bị ảnh hưởng bởi sự mất cân bằng dữ liệu (data imbalance)**. Ví dụ, nếu tập dữ liệu có 90% mẫu thuộc lớp A và 10% mẫu thuộc lớp B, mô hình luôn dự đoán tất cả các mẫu là A sẽ đạt NPV 90%, nhưng thực tế mô hình không hề phân biệt được gì.
- NPV cần được xem xét **cùng với Độ nhạy (Sensitivity)** và **Giá trị dự đoán dương (PPV)** để đánh giá toàn diện hiệu quả mô hình

Phân loại nhị phân

Điểm F1 (F1 Score): 2 * (Precision * Recall) / (Precision + Recall)

Ý nghĩa:

- F1 Score **tích hợp** thông tin từ cả Precision và Recall, giúp đánh giá **khả năng tổng thể** của mô hình trong việc phân loại các mẫu.
- Giá trị F1 càng **cao** thể hiện mô hình càng **hiệu quả** trong việc **cân bằng** giữa việc **dự đoán chính xác** các mẫu thuộc lớp "tích cực" và **giảm thiểu tỷ lệ bỏ sót**.

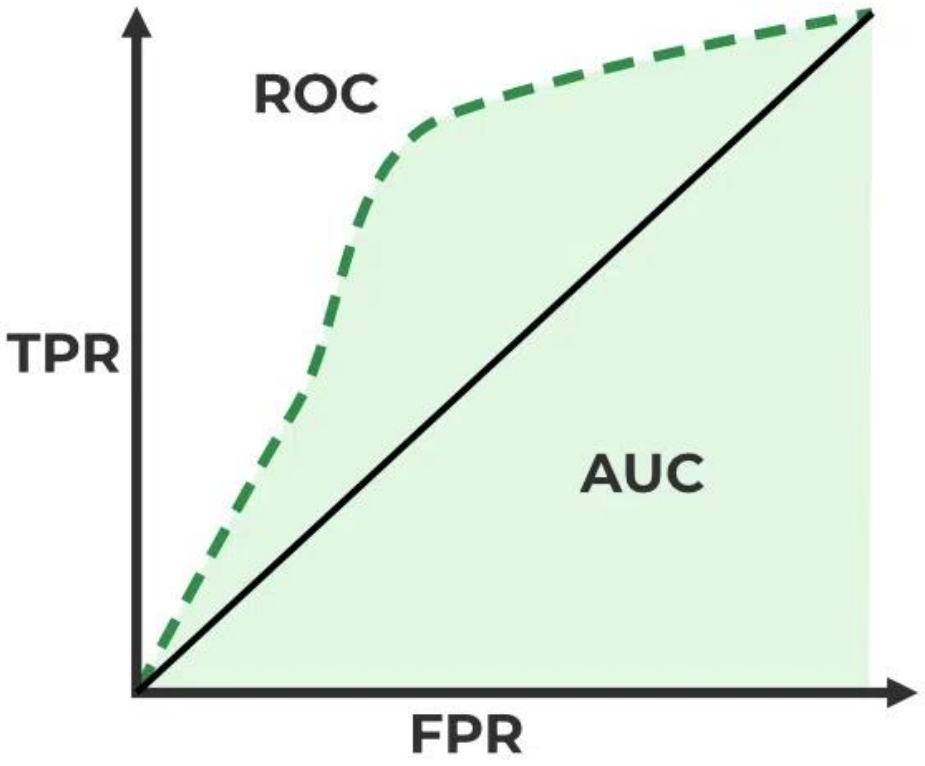
Lưu ý:

- F1 Score giúp **giải quyết** vấn đề **thiên vị** (bias) trong việc đánh giá mô hình, thường xảy ra khi sử dụng chỉ một trong hai thước đo Precision hoặc Recall.
- F1 Score **nhạy cảm** với sự **mất cân bằng dữ liệu (data imbalance)**.
- F1 Score **không** cung cấp thông tin chi tiết về **khả năng dự đoán** của mô hình đối với từng lớp riêng lẻ.

Mô hình phân loại để dự đoán bệnh nhân có mắc bệnh ung thư hay không.

- **F1 Score cao:** Cho biết mô hình có khả năng tốt trong việc **cân bằng** giữa việc **phát hiện chính xác** những người mắc bệnh ung thư và **giảm thiểu** việc **bỏ sót** những người mắc bệnh.

Phân loại nhị phân



Đường cong ROC (ROC Curve)

- **Trục hoành:** Tỷ lệ dương giả (FPR)
- **Trục tung:** Tỷ lệ dương thực (TPR)
- **Đường chéo:** Đường phân chia giữa mô hình phân loại tốt (trên đường chéo) và mô hình phân loại tệ (dưới đường chéo)

Điểm AUC (AUC Score): Diện tích dưới đường cong ROC

- Giá trị AUC **càng cao** thể hiện mô hình **càng hiệu quả** trong việc phân loại các mẫu.
- **AUC = 0.5** tương đương với việc **mô hình dự đoán ngẫu nhiên**.
- **AUC = 1** thể hiện mô hình **phân loại chính xác** tất cả các mẫu.

Phân loại nhị phân

Đường cong ROC (ROC Curve) và Điểm AUC (AUC Score)

Ý nghĩa:

- **ROC Curve** thể hiện mối quan hệ giữa **Tỷ lệ dương thực (True Positive Rate - TPR)** và **Tỷ lệ dương giả (False Positive Rate - FPR)** khi thay đổi ngưỡng phân loại (threshold) của mô hình.
- **Diện tích dưới đường cong ROC (AUC)** là thước đo tổng thể hiệu quả của mô hình phân loại, với giá trị AUC càng **cao** thể hiện mô hình càng **hiệu quả**.

Mô hình phân loại để dự đoán bệnh nhân có mắc bệnh ung thư hay không.

- **ROC Curve cao:** Cho biết mô hình có khả năng **phân biệt tốt** giữa người mắc bệnh ung thư và người không mắc bệnh.
- **AUC cao:** Cho biết mô hình có hiệu quả **cao** trong việc dự đoán bệnh ung thư.

Phân loại nhị phân

Ngưỡng phân loại chia các mẫu dữ liệu thành hai nhóm:

- Nhóm tích cực nếu giá trị dự đoán lớn hơn hoặc bằng ngưỡng phân loại.
- Nhóm tiêu cực nếu giá trị dự đoán nhỏ hơn ngưỡng phân loại.

Cách thủ công:

- Lựa chọn ngưỡng phân loại dựa trên kinh nghiệm hoặc kiến thức chuyên môn.
- Thử nghiệm với các giá trị ngưỡng khác nhau để tìm ra giá trị phù hợp nhất.

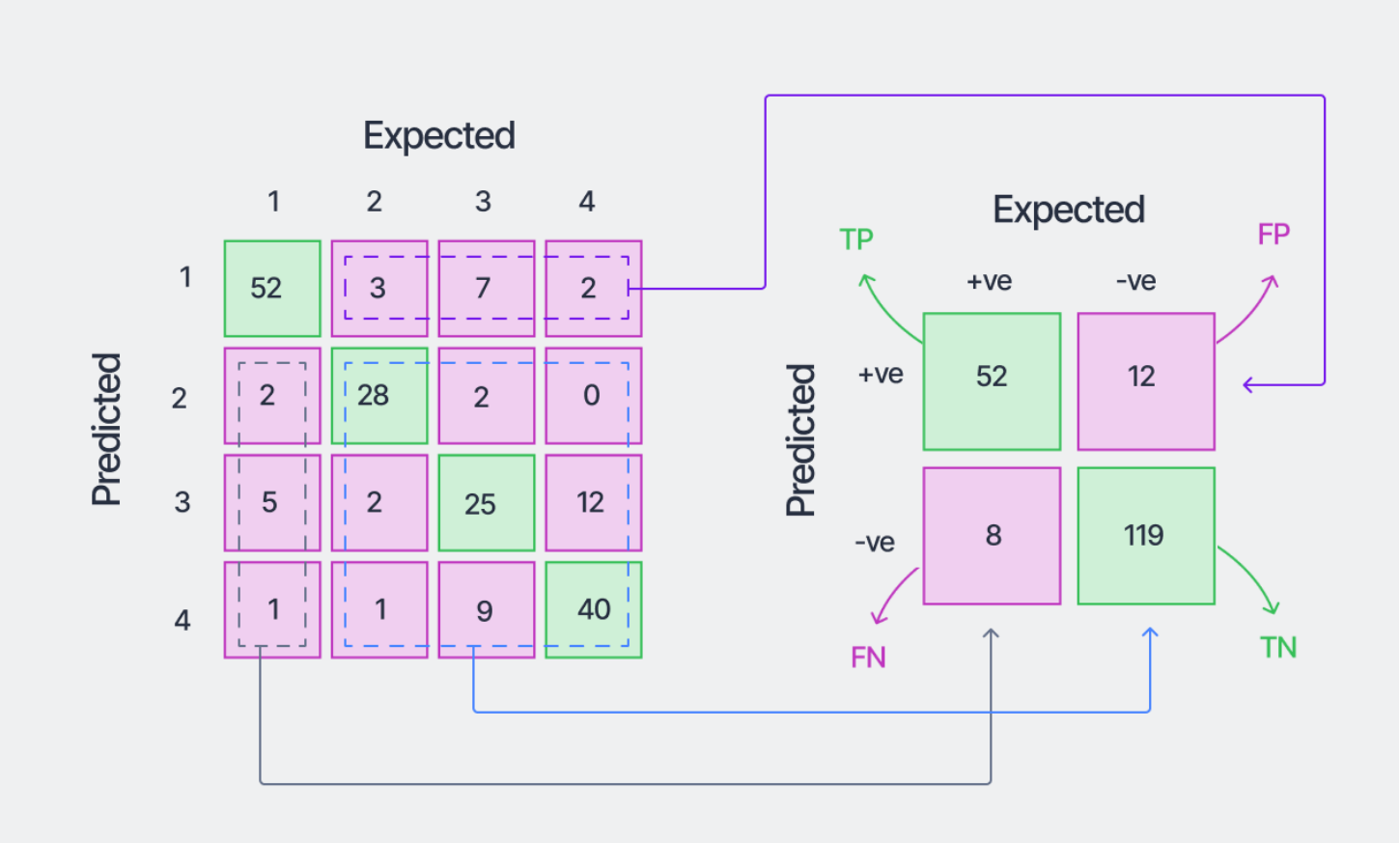
Cách tự động:

- Sử dụng các thuật toán để lựa chọn ngưỡng phân loại tối ưu, ví dụ như: **Grid Search**, **Cross-Validation**.

Lưu ý:

- Việc lựa chọn ngưỡng phân loại phụ thuộc vào mục đích sử dụng và mức độ chấp nhận rủi ro của mô hình.
- Có thể sử dụng ***đường cong ROC (ROC Curve)*** để đánh giá hiệu quả của mô hình phân loại với các giá trị ngưỡng khác nhau.

Phân loại đa lớp



Ma trận nhầm lẫn (Confusion Matrix)

Chuyển đổi ma trận nhầm lẫn nhiều lớp thành ma trận 1-vs-all (đối với lớp 1 – positive).

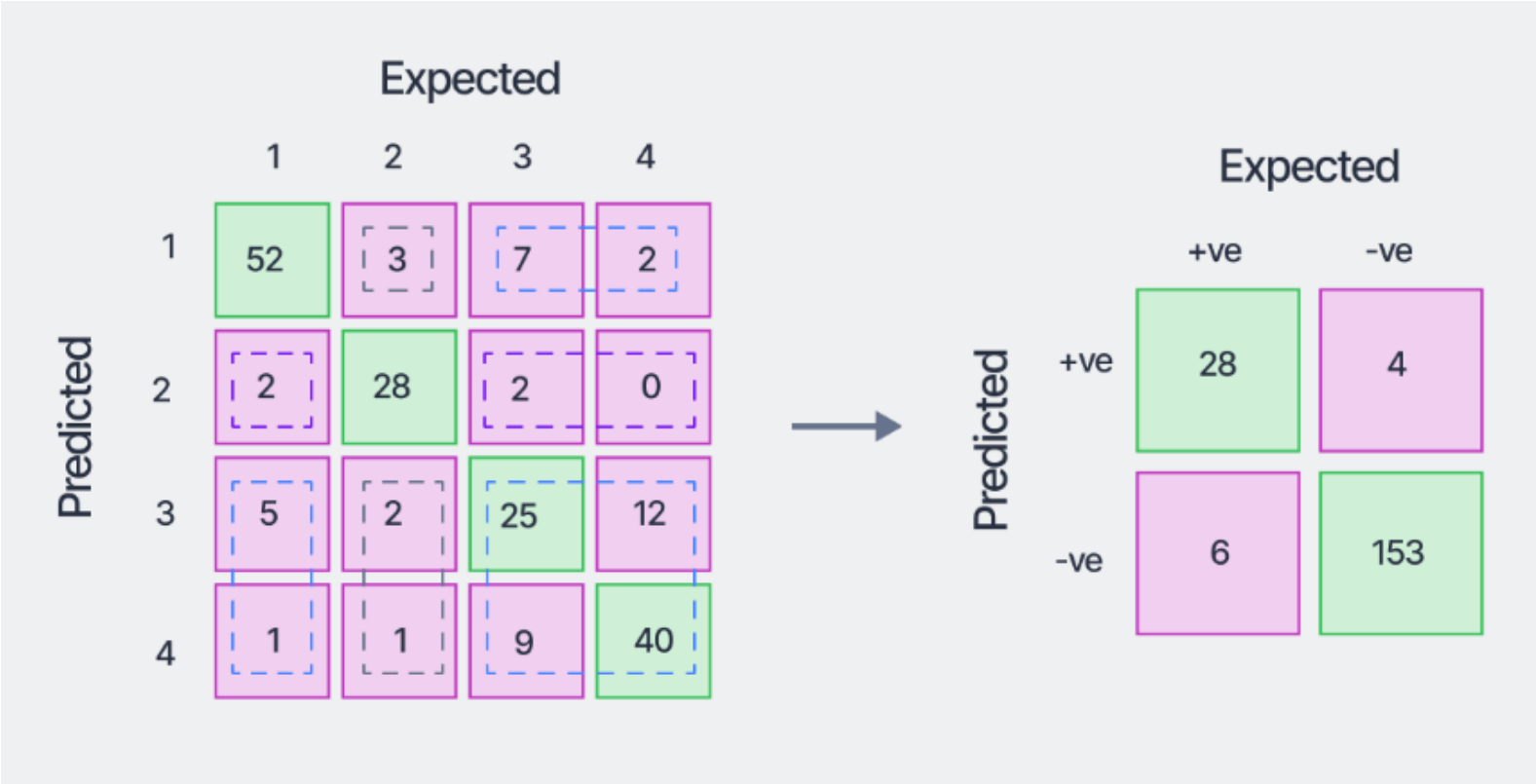
Lớp tích cực: Thể hiện **kết quả mong muốn** hoặc **sự kiện quan trọng** mà mô hình muốn dự đoán.

- Lớp được chọn để tính toán độ đo.

Lớp tiêu cực: Thể hiện **kết quả ngược lại** với lớp tích cực.

- Tất cả các lớp còn lại.

Phân loại đa lớp



Ma trận nhầm lẫn (Confusion Matrix)

Chuyển đổi ma trận nhầm lẫn nhiều lớp thành ma trận 1-vs-all (đối với lớp 2 – positive).

Lớp tích cực: Thể hiện **kết quả mong muốn** hoặc **sự kiện quan trọng** mà mô hình muốn dự đoán.

- Lớp được chọn để tính toán độ đo.

Lớp tiêu cực: Thể hiện **kết quả ngược lại** với lớp tích cực.

- Tất cả các lớp còn lại.

Phân loại đa lớp

Dựa vào ma trận chuyển đổi của từng lớp, tính các độ đo đánh giá của từng lớp theo công thức sau:

- **Độ chính xác (Accuracy):** (TP + TN) / (TP + FN + FP + TN)
- **Độ nhạy (Sensitivity) / Tỷ lệ thu hồi (Recall):** TP / (TP + FN)
- **Độ đặc hiệu (Specificity):** TN / (TN + FP)
- **Giá trị dự đoán dương (PPV) / Độ chính xác (Precision):** TP / (TP + FP)
- **Giá trị dự đoán âm (Negative Predictive Value):** TN / (FN + TN)
- **Điểm F1 (F1 Score):** 2 * (Precision * Recall) / (Precision + Recall)

Phân loại đa lớp

Độ chính xác trung bình mAP (mean Average Precision)

Đo lường khả năng của mô hình phân biệt giữa các lớp khác nhau, đặc biệt hữu ích trong các trường hợp có nhiều lớp và dữ liệu không cân bằng.

Ý nghĩa:

- Giá trị mAP cao (thường từ 0.7 đến 1.0) thể hiện mô hình có hiệu suất tốt trong việc phân loại các lớp.
- Giá trị mAP thấp (thường dưới 0.5) cho thấy mô hình cần cải thiện khả năng phân biệt các lớp.

1. Tính toán AP cho từng lớp:

- Sử dụng đường cong ROC (Receiver Operating Characteristic) để vẽ mối quan hệ giữa độ chính xác và độ thu hồi.
- Diện tích dưới đường cong ROC (AUC) được sử dụng để tính toán AP.

2. Tính toán mAP: mAP là trung bình cộng của AP cho tất cả các lớp.

3. So sánh mAP của các mô hình khác nhau: Mô hình có mAP cao hơn là mô hình có hiệu suất tốt hơn trong việc phân loại các lớp.

4. Phân tích kết quả mAP: Xác định các lớp có AP thấp để cải thiện hiệu suất mô hình. Điều chỉnh các siêu tham số của mô hình để tăng AP cho các lớp có hiệu suất thấp.

Phân loại đa lớp

Macro-F1

Là trung bình cộng của F1-score cho từng lớp, giúp đánh giá khả năng của mô hình phân loại tất cả các lớp một cách công bằng.

Ý nghĩa:

- Giá trị Macro-F1 cao (thường từ 0.7 đến 1.0) thể hiện mô hình có hiệu suất tốt trong việc phân loại tất cả các lớp.
- Giá trị Macro-F1 thấp (thường dưới 0.5) cho thấy mô hình cần cải thiện khả năng phân loại một hoặc nhiều lớp.

Cách đánh giá mô hình phân loại đa lớp dựa trên Macro-F1:

- Tính toán F1-score cho từng lớp.
- Tính trung bình cộng của F1-score cho tất cả các lớp để tính Macro-F1.
- So sánh Macro-F1 của các mô hình khác nhau.
- Phân tích F1-score của từng lớp để xác định các lớp cần cải thiện.

Phân loại đa lớp

Micro-F1

Là F1-score được tính toán trên toàn bộ tập dữ liệu, không phân biệt các lớp. Hoạt động hiệu quả với dữ liệu không cân bằng.

Ý nghĩa:

- Giá trị Micro-F1 cao (thường từ 0.7 đến 1.0) thể hiện mô hình có hiệu suất tốt trong việc phân loại tất cả các lớp.
- Giá trị Micro-F1 thấp (thường dưới 0.5) cho thấy mô hình cần cải thiện khả năng phân loại một hoặc nhiều lớp.

Cách đánh giá mô hình phân loại đa lớp dựa trên Micro-F1:

- Tính toán F1-score trên toàn bộ tập dữ liệu, không phân biệt các lớp.
- So sánh Micro-F1 của các mô hình khác nhau.
- Phân tích F1-score của từng lớp để xác định các lớp cần cải thiện.

Phân loại đa lớp

Micro-F1

- **Lớp A:**
  - Độ chính xác (Precision): 100 / (100 + 20) = 0.833
  - Độ thu hồi (Recall): 100 / (100 + 40) = 0.714
- **Lớp B:**
  - Độ chính xác: 80 / (80 + 40) = 0.667
  - Độ thu hồi: 80 / (80 + 60) = 0.571
- **Lớp C:**
  - Độ chính xác: 60 / (60 + 60) = 0.5
  - Độ thu hồi: 60 / (60 + 40) = 0.6

Lớp	Dự đoán đúng	Dự đoán sai	Số lượng mẫu
A	100	20	140 (100 + 40)
B	80	40	140 (80 + 60)
C	60	60	100 (60 + 40)

- Micro-F1 được tính toán trên toàn bộ tập dữ liệu, không phân biệt các lớp:
  - **Độ chính xác:** (100 + 80 + 60) / (100 + 20 + 80 + 40 + 60 + 60) = 0.692
  - **Độ thu hồi:** (100 + 80 + 60) / (100 + 40 + 80 + 60 + 40 + 60) = 0.636
  - **Micro-F1:** 2 * 0.692 * 0.636 / (0.692 + 0.636) = 0.664

Vậy, Micro-F1 của mô hình là 0.664, cho thấy mô hình có hiệu suất tương đối tốt trong việc phân loại cả 3 lớp.

Phân loại đa lớp

Weighted-F1

Là một chỉ số đánh giá hiệu suất của mô hình phân loại đa lớp, đặc biệt hữu ích trong các trường hợp có nhiều lớp và dữ liệu không cân bằng.

Weighted-F1 là F1-score được tính toán có trọng số, dựa trên số lượng mẫu của mỗi lớp. Chú trọng đến các lớp.

Ý nghĩa:

- Giá trị Weighted-F1 cao (thường từ 0.7 đến 1.0) thể hiện mô hình có hiệu suất tốt trong việc phân loại tất cả các lớp, đặc biệt chú trọng đến các lớp có số lượng mẫu lớn.
- Giá trị Weighted-F1 thấp (thường dưới 0.5) cho thấy mô hình cần cải thiện khả năng phân loại một hoặc nhiều lớp, đặc biệt là các lớp có số lượng mẫu lớn.

Cách đánh giá mô hình phân loại đa lớp dựa trên Weighted-F1:

1. Tính F1-score cho từng lớp.
2. Nhân F1-score của mỗi lớp với số lượng mẫu của lớp đó.
3. Cộng các giá trị F1-score đã được nhân trọng số và chia cho tổng số lượng mẫu để được Weighted-F1.

Phân loại đa lớp

Weighted-F1

Lớp	Dự đoán đúng	Dự đoán sai	Số lượng mẫu
A	100	20	120
B	80	40	80
C	60	60	60

1. Tính F1-score cho từng lớp:

- Lớp A: F1-score = 0.769
- Lớp B: F1-score = 0.615
- Lớp C: F1-score = 0.545

2. Tính Weighted-F1:

- Weighted-F1 = (120 * 0.769 + 80 * 0.615 + 60 * 0.545) / (120 + 80 + 60) = 0.67

Vậy, Weighted-F1 của mô hình là 0.681, cho thấy mô hình có hiệu suất tương đối tốt trong việc phân loại cả 3 lớp, đặc biệt chú trọng đến lớp A có số lượng mẫu lớn nhất.

➢Có thể điều chỉnh trọng số của các lớp để phù hợp với mục đích đánh giá.

5. Triển khai hệ thống

1. Chuẩn bị:

- **Mô hình phân loại:** Mô hình phân loại đã được huấn luyện và đánh giá hiệu quả.
- **Ứng dụng web / Ứng dụng di động:** Đã được phát triển và sẵn sàng để tích hợp mô hình.

2. Triển khai mô hình trên server:

- **Môi trường:** Cloud Platforms / On-Premise Servers
- Mô hình phân loại được triển khai trên server và ứng dụng sẽ gửi yêu cầu đến server để thực hiện phân loại.
  - **Lưu trữ mô hình:** Mô hình phân loại được lưu trữ trên server.
    - File format: .h5 (HDF5), .pb (Protocol Buffer), .pkl (Pickle), v.v.
  - **Tạo API:** API được tạo ra để nhận yêu cầu từ ứng dụng và gửi kết quả phân loại.
    - REST API: Tạo RESTful API bằng các frameworks như Flask, FastAPI, v.v.
  - **Gọi API:** Ứng dụng gửi yêu cầu đến API và nhận kết quả phân loại.

6. Giám sát và bảo trì

Giám sát:

- **Theo dõi các chỉ số chính:** Theo dõi thường xuyên các chỉ số phản ánh hiệu suất và tình trạng của mô hình.
- **Phát hiện dữ liệu bất thường:** Phân phối dữ liệu thực tế được cung cấp cho mô hình bắt đầu khác với dữ liệu mà nó được đào tạo.
- **Cảnh báo và ghi nhật ký:** Theo dõi hành vi của mô hình và chẩn đoán các vấn đề dễ dàng hơn.

Bảo trì:

- **Đào tạo lại:** Thường xuyên đào tạo lại mô hình với dữ liệu mới, đặc biệt nếu phát hiện dữ liệu bất thường. Điều này giúp mô hình thích ứng với những thay đổi trong thế giới thực và duy trì hiệu suất tối ưu.

Công cụ và tài nguyên: TensorBoard, Mlflow, Cloud Monitoring Services.

7. Bài tập

Tìm hiểu công cụ và thư viện hỗ trợ (Python) để áp dụng thuật toán phân lớp nhị phân hoặc đa lớp. Đánh giá hiệu quả của mô hình phân lớp.

- Công dụng, cú pháp sử dụng, ví dụ demo cách áp dụng.
- Vận dụng vào tập dữ liệu khai thác của nhóm.

8. Tổng kết

Phân loại nhị phân

Phân loại đa lớp

Question & Answer