Bài 46 - Đánh giá mô hình phân loại trong ML

13 Aug 2020 - phamdinhkhanh

Menu

1. Đánh giá mô hình

Trong quá trình xây dựng một mô hình machine learning, một phần không thể thiếu để biết được chất lượng của mô hình như thế nào đó chính là đánh giá mô hình.

Đánh giá mô hình giúp chúng ta lựa chọn được mô hình phù hợp nhất đối với bài toán của mình. Tuy nhiên để tìm được thước đo đánh giá mô hình phù hợp thì chúng ta cần phải hiểu về ý nghĩa, bản chất và trường hợp áp dụng của từng thước đo.

Chính vì vậy bài viết này sẽ cung cấp cho các bạn kiến thức về các thước đo cơ bản nhất, thường được áp dụng trong các mô hình phân loại trong machine learning nhưng chúng ta đôi khi còn chưa nắm vững hoặc chưa biết cách áp dụng những thước đo này sao cho phù hợp với từng bộ dữ liệu cụ thể.

Hãy cùng phân tích và tìm hiểu các thước đo này qua các ví dụ bên dưới.

2. Bộ dữ liệu

Giả định rằng chúng ta đang xây dựng một mô hình phân loại nợ xấu. Nhãn của các quan sát sẽ bao gồm GOOD (thông thường) và BAD (nợ xấu). Kích thước của các tập dữ liệu như sau:

- Tập train: 1000 hồ sơ bao gồm 900 hồ sơ GOOD và 100 hồ sơ BAD.
- tập test: 100 hồ sơ bao gồm 85 hồ sơ GOOD và 15 hồ sơ BAD.

Để thuận tiện cho diễn giải và đồng nhất với những tài liệu tham khảo khác về ký hiệu thì biến mục tiêu y nhãn BAD có giá trị 1 và GOOD giá trị 0. Đồng thời trong các công thức diễn giải và bảng thống kê, nhãn BAD là positive và GOOD là negative. Positive và Negative ở đây chỉ là qui ước tương ứng với giá trị 1 và 0 chứ không nên hiểu theo nghĩa đen là `tích cực` và `tiêu cực`.

Top

Một mô hình phân loại f đưa ra kết quả dự báo trên tập train được thống kê trên bảng chéo như sau:

Predict/Actual		Actual	
		BAD (Positive)	GOOD (Negative)
Predict	BAD (Positive)	55 (TP - True Positive)	50 (FP - False Positive)
	GOOD (Negative)	45 (FN - False Negative)	850 (TN - True Negative)
Total		100	900

Các chỉ số TP, FP, TN, FN lần lượt có ý nghĩa là :

- TP (True Positive): Tổng số trường hợp dự báo khớp Positive.
- TN (True Negative): Tổng số trường hợp dự báo khớp Negative.
- FP (False Positive): Tổng số trường hợp dự báo các quan sát thuộc nhãn Negative thành Positive.
- FN (False Negative): Tổng số trường hợp dự báo các quan sát thuộc nhãn Positive thành Negative.

Những chỉ số trên sẽ là cơ sở để tính toán những metric như accuracy, precision, recall, f1 score mà ta sẽ tìm hiểu bên dưới.

3. Độ chính xác (accuracy)

Khi xây dựng mô hình phân loại chúng ta sẽ muốn biết một cách khái quát tỷ lệ các trường hợp được dự báo đúng trên tổng số các trường hợp là bao nhiêu. Tỷ lệ đó được gọi là độ chính xác. Độ chính xác giúp ta đánh giá hiệu quả dự báo của mô hình trên một bộ dữ liệu. Độ chính xác càng cao thì mô hình của chúng ta càng chuẩn xác. Khi một ai đó nói mô hình của họ dự báo chính xác 90.5% thì chúng ta hiểu rằng họ đang đề cập tới độ chính xác được tính theo công thức :

$$\text{Accuracy} = \frac{TP + TN}{\text{total sample}} = \frac{55 + 850}{1000} = 90.5\%$$

Tính toán accuracy trên sklearn :

```
1    from sklearn.metrics import accuracy_score
2    accuracy_score(y_true, y_pred)
```

Trong đó `y_label` là nhãn của dữ liệu và `y_pred` là nhãn dự báo.

Trong các metrics đánh giá mô hình phân loại thì độ chính xác là metric khá được ưa chuộng vì nó có công thức tường minh và dễ diễn giải ý nghĩa. Tuy nhiên hạn chế của nó là đo lường trên *tất cả* các nhãn mà không quan tâm đến độ chính xác trên từng nhãn. Do đó nó không phù hợp để đánh giá những tác vụ mà *tầm quan trọng* của việc dự báo các nhãn không còn như nhau. Hay nói cách khác, như trong ví dụ phân loại nợ xấu, việc chúng ta phát hiện đúng một hồ sơ nợ xấu quan trọng hơn việc chúng ta phát hiện đúng một hồ sơ thông thường.

Khi đó chúng ta sẽ quan tâm hơn tới độ chính xác được đo lường chỉ **trên nhãn BAD** hơn và sẽ cần những metrics như precision, recall đánh giá chuyên biệt trên nhóm này. Cùng tìm hiểu về các metrics này bên dưới.

Top

4. Precision

Precision trả lời cho câu hỏi trong các trường hợp được dự báo là positive thì có bao nhiêu trường hợp là đúng ? Và tất nhiên precision càng cao thì mô hình của chúng ta càng tốt trong việc phân loại hồ sơ BAD (BAD chính là nhóm positive). Công thức của precision như sau:

$$\text{Precision} = \frac{TP}{\text{total predicted positive}} = \frac{TP}{TP+FP} = \frac{55}{55+50} = 52.4\%$$

Precision sẽ cho chúng ta biết mức độ chuẩn xác của mô hình đối với các hồ sơ được dự báo là BAD. Ví dụ khi precision = 52.4%, chúng ta tin rằng trong các hồ sơ được dự báo là BAD thì có 52.4% tỷ lệ các hồ sơ được phân loại đúng.

Cũng có ý nghĩa gần tương tự như precision, có cùng tử số nhưng có một chút khác biệt về mẫu số trong công thức tính toán, và cũng là một chỉ số giúp đo lường hiệu suất dự báo trên nhóm positive, đó là recall.

5. Recall

Recall đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive. Công thức của recall như sau:

$$\text{Recall} = \frac{TP}{\text{total actual positive}} = \frac{TP}{TP+FN} = \frac{55}{55+45} = 55\%$$

Để tính được recall thì chúng ta phải biết trước nhãn của dữ liệu. Do đó recall có thể được dùng để đánh gía trên tập train và validation vì chúng ta đã biết trước nhãn. Trên tập test khi dữ liệu được coi như mới hoàn toàn và **chưa biết nhãn** thì chúng ta sẽ sử dụng precision.

Tính toán precision và recall trên sklearn chúng ta sẽ dựa trên ground truth `y_label` và xác suất dự báo `y_prob` :

```
1    from sklearn.metrics import precision_recall_curve
2    prec, rec, thres = precision_recall_curve(y_label, y_prob)
```

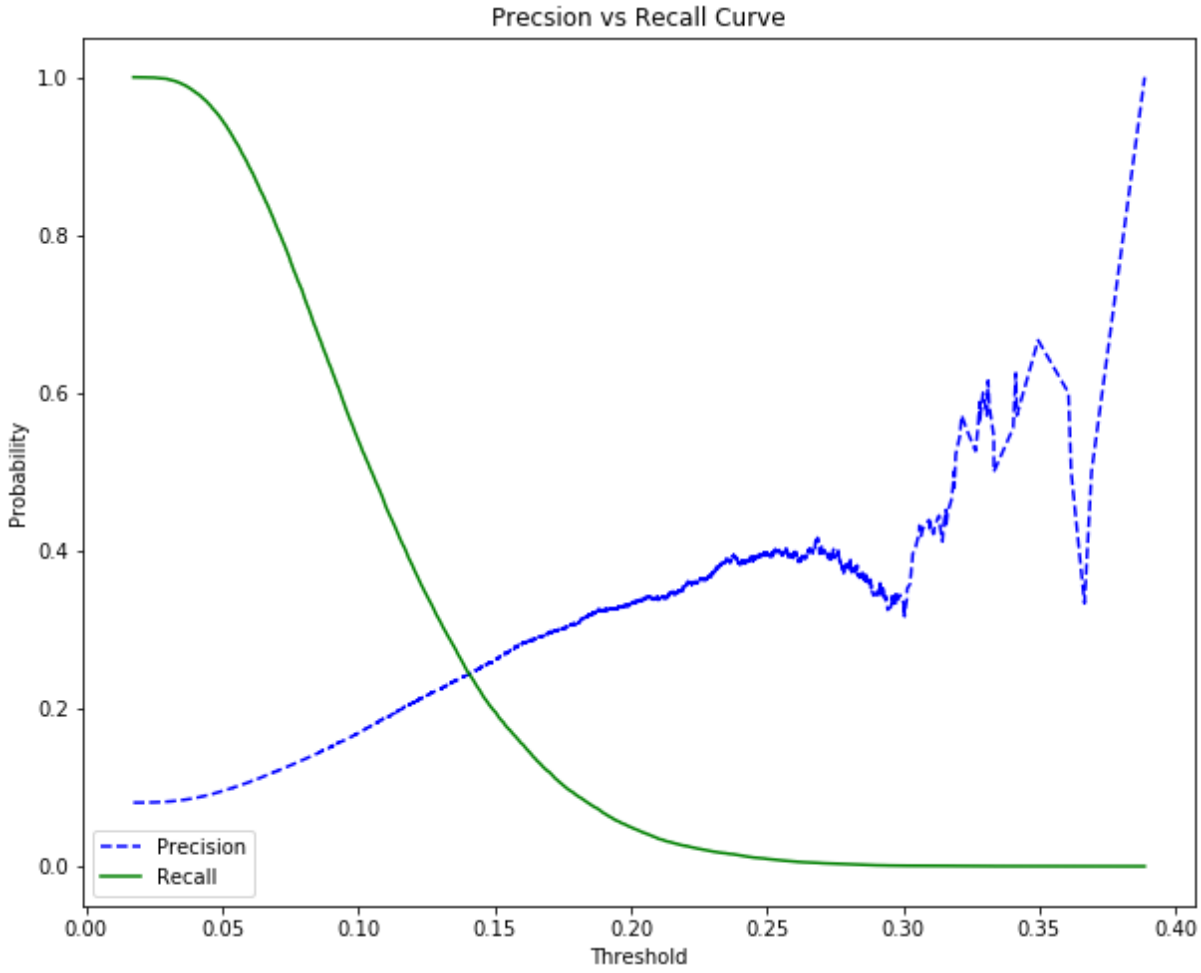
6. Trade off giữa precision và recall

Thông thường các model sẽ lựa chọn một ngưỡng mặc định là 0.5 để quyết định nhãn. Tức là nếu ta có một hàm phân loại $f_\theta()$ thì nhãn dự báo sẽ dựa trên độ lớn của xác suất dự báo như sau:

$$\begin{cases} f_\theta(x) \geq 0.5, \text{label} = 1 \\ f_\theta(x) < 0.5, \text{label} = 0 \end{cases}$$

Do đó precision và recall sẽ không cố định mà chịu sự biến đổi theo ngưỡng xác suất được lựa chọn. Bên dưới là một đồ thị minh họa cho sự biến đổi này. Đồ thị được trích từ home credit kaggle kernel - phamdinhkhanh (https://www.kaggle.com/phamdinhkhanh/home-credit-default-risk).

Top



Thậm chí bằng một chút suy luận logic, ta còn có thể chứng minh được mối quan hệ giữa precision và recall khi biến đổi theo threshold là mối quan hệ đánh đổi (*trade off*). Khi precision cao thì recall thấp và ngược lại. Thật vậy :

- Giả sử trong ví dụ về phân loại nợ xấu, chúng ta muốn khi mô hình dự báo một hồ sơ là BAD thật chắc chắn nên lựa chọn một ngưỡng threshold cao hơn, chẳng hạn như 0.9. Khi đó một hồ sơ rơi vào BAD thì khả năng rất rất cao là hồ sơ đó sẽ đúng là BAD bởi xác suất 90% là một mức tin cậy khá cao. Mặt khác xin nhắc lại precision bằng **số lượng được dự báo là BAD đúng** chia cho tổng số **được dự báo là BAD** nên nó có xu hướng cao khi threshold được thiết lập cao. Đồng thời do số lượng các quan sát được dự báo là BAD sẽ giảm xuống khi threshold cao hơn và số lượng hồ sơ BAD không đổi nên recall thấp hơn.
- Trong trường hợp chúng ta muốn nới lỏng kết quả phân loại hồ sơ BAD một chút bằng cách giảm threshold và chấp nhận một số hợp đồng bị dự báo sai từ GOOD sang BAD. Khi đó số lượng hồ sơ được dự báo là BAD tăng lên trong khi số lượng hồ sơ BAD được dự báo đúng tăng không đáng kể. Điều đó dẫn tới precision giảm và recall tăng.

Sự đánh đổi giữa precision và recall khiến cho kết quả của mô hình thường l : precision cao, recall thấp hoặc precision thấp, recall cao. Khi đó rất khó để lựa chọn đâu là một mô hình tốt vì không biết rằng đánh giá trên precision hay recall sẽ phù hợp hơn. Chính vì vậy chúng ta sẽ tìm cách kết hợp cả precision và recall trong một chỉ số mới, đó chính là f1 score.

7. F1 Score

F_1 Score là trung bình điều hòa giữa precision và recall. Do đó nó đại diện hơn trong việc đánh gía độ chính xác trên đồng thời precision và recall.

Top

$$\mathrm{F}_1 = \frac{2}{\mathrm{precision}^{-1} + \mathrm{recall}^{-1}} = \frac{2}{0.524^{-1} + 0.55^{-1}} = 53.7\%$$

Trong trường hợp $\mathrm{precision} = 0$ hoặc $\mathrm{recall} = 0$ ta qui ước $\mathrm{F}_1 = 0$.

Ta chứng minh được rằng giá trị của F_1 score luôn nằm trong khoảng của precision và recall. Thật vậy :

$$\begin{aligned}\mathrm{F}_1 &= \frac{2\ \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \\ &\leq \frac{2\ \mathrm{precision} \times \mathrm{recall}}{2\ \min(\mathrm{precision}, \mathrm{recall})} = \max(\mathrm{precision}, \mathrm{recall})\end{aligned}$$

Tương tự:

$$\begin{aligned}\mathrm{F}_1 &= \frac{2\ \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \\ &\geq \frac{2\ \mathrm{precision} \times \mathrm{recall}}{2\ \max(\mathrm{precision}, \mathrm{recall})} = \min(\mathrm{precision}, \mathrm{recall})\end{aligned}$$

Do đó đối với những trường hợp mà precision và recall quá chênh lệch thì F_1 score sẽ cân bằng được cả hai độ lớn này và giúp ta đưa ra một đánh giá khách quan hơn. Ví dụ như kết quả bảng bên dưới :

Predict/Actual		Actual: BAD (Positive)	Actual: GOOD (Negative)	
Predict	**BAD (Positive)**	55 (TP - True Positive)	5 (FP - False Positive)	**precision=55/(55+5)=91.6%**
	GOOD (Negative)	45 (FN - False Negative)	895 (TN - True Negative)	
Total		100	900	
		recall=55/(55+45)=55%		

Nếu dựa trên precision thì giá trị precision=91.6% cho thấy đây là một model *khá tốt*. Tuy nhiên trong 100 trường hợp positive thì mô hình chỉ nhận diện được đúng 55 trường hợp nên xét theo recall=55% thì đây không phải là một mô hình tốt. Trong trường hợp này F_1 sẽ được sử dụng như một chỉ số đại diện cho cả precision và recall. Điểm F_1 bằng 69% cho thấy đây là một mô hình có sức mạnh ở mức trung bình và đánh giá của chúng ta sẽ xác thực hơn so với việc quá lạc quan vào mô hình khi chỉ nhìn vào precision và quá bi quan nếu chỉ dựa trên recall.

Trên sklearn, f1 score được tính như sau :

```
1     from sklearn.metrics import f1_score
2     f1_score(y_label, y_pred)
```

Trong đó y_label là nhãn của dữ liệu và y_pred là nhãn dự báo.

8. Tại sao F1 score không là trung bình cộng precision và recall

Có một học viên thắc mắc mình rằng tại sao F_1 score không được lấy bằng trung bình cộng giữa precision và recall? Lấy ví dụ trực quan trong trường hợp mô hình của bạn có precision quá thấp và recall quá cao, chẳng hạn precision=0.01 và recall=1.0.

Top

Nhìn vào biểu đồ trade off giữa precision và recall thì đây có thể được xem như một mô hình thiết lập threshold thấp. Nó tương đương với việc dự đoán ngẫu nhiên toàn bộ là positive. Do đó không thể xem đó là một mô hình tốt.

Nếu sử dụng công thức trung bình thì

$$\mathrm{F}_1 = \frac{\mathrm{precision}+\mathrm{recall}}{2} = 0.5005$$

giá trị này cho thấy đây là một mô hình ở mức trung bình. Trong khi sử dụng công thức trung bình điều hòa thì

$$\mathrm{F}_1 = \frac{2\ \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \approx 0$$

giá trị này giúp nhận diện được mô hình không tốt.

Tóm lại sử dụng trung bình điều hòa sẽ phạt nặng hơn những trường hợp mô hình có precision thấp, recall cao hoặc precision cao, recall thấp. Đây là những trường hợp tương đương với dự báo thiên về một nhóm là positive hoặc negative nên không phải là mô hình tốt. Điểm số từ trung bình điều hòa sẽ giúp ta nhận biết được những trường hợp không tốt như vậy.

9. Accuracy và F1 score

Accuracy và F1 score đều được sử dụng để đánh giá hiệu suất của mô hình phân loại. Vậy trong tình huống nào chúng ta nên sử dụng chỉ số nào là phù hợp ? Điều đó phụ thuộc vào bộ dữ liệu của bạn có xảy ra hiện tượng mất cân bằng hay không ? Hãy cùng quay trở lại phân tích bảng kết quả đầu tiên. Ta gọi trường hợp này là dự báo theo *mô hình* :

Predict/Actual		Actual	
		BAD (Positive)	**GOOD (Negative)**
Predict	**BAD (Positive)**	55 (TP - True Positive)	50 (FP - False Positive)
	GOOD (Negative)	45 (FN - False Negative)	850 (TN - True Negative)
Total		100	900

Khi dự báo theo *mô hình* dễ dàng tính được accuracy=90.5%, đây là một kết quả cũng khá cao và chúng ta nhận định rằng mô hình phân loại tốt.

Tuy nhiên xét tình huống chúng ta dự báo *ngẫu nhiên* toàn bộ mẫu là các hồ sơ GOOD. Như vậy độ chính xác đạt được thậm chí đã lên tới 90%. Lúc này chúng ta nghi ngờ sự phù hợp của accuracy trong việc đánh giá mô hình vì không cần tới mô hình cũng tạo ra một kết quả gần như tương đương với có mô hình.

Mặt khác, khi sử dụng F_1 score làm chỉ số đánh giá ta thu được điểm số khi dự báo *ngẫu nhiên* là 0% và khi dự báo theo *mô hình* là 69% (bạn đọc hãy tự tính). Các bạn đã thấy sự chênh lệch điểm số F_1 score giữa hai mô hình chưa ? Đồng thời F_1 score cũng không khiến chúng ta lạc quan vào những mô hình có chất lượng thấp nhưng do sử dụng accuracy nên chúng có kết qủa đánh giá cao. Ngoài ra F_1 score chỉ tính toán độ chính xác trên nhóm mẫu thiểu (positive) là nhóm mà chúng ta mong muốn đánh giá hơn trong trường hợp mất cân bằng nên nó sẽ phù hợp hơn accuracy được tính toán trên cả mẫu positive và negative.

Top

10. AUC

ROC là đường cong biểu diễn khả năng phân loại của một mô hình phân loại tại các ngưỡng threshold. Đường cong này dựa trên hai chỉ số :

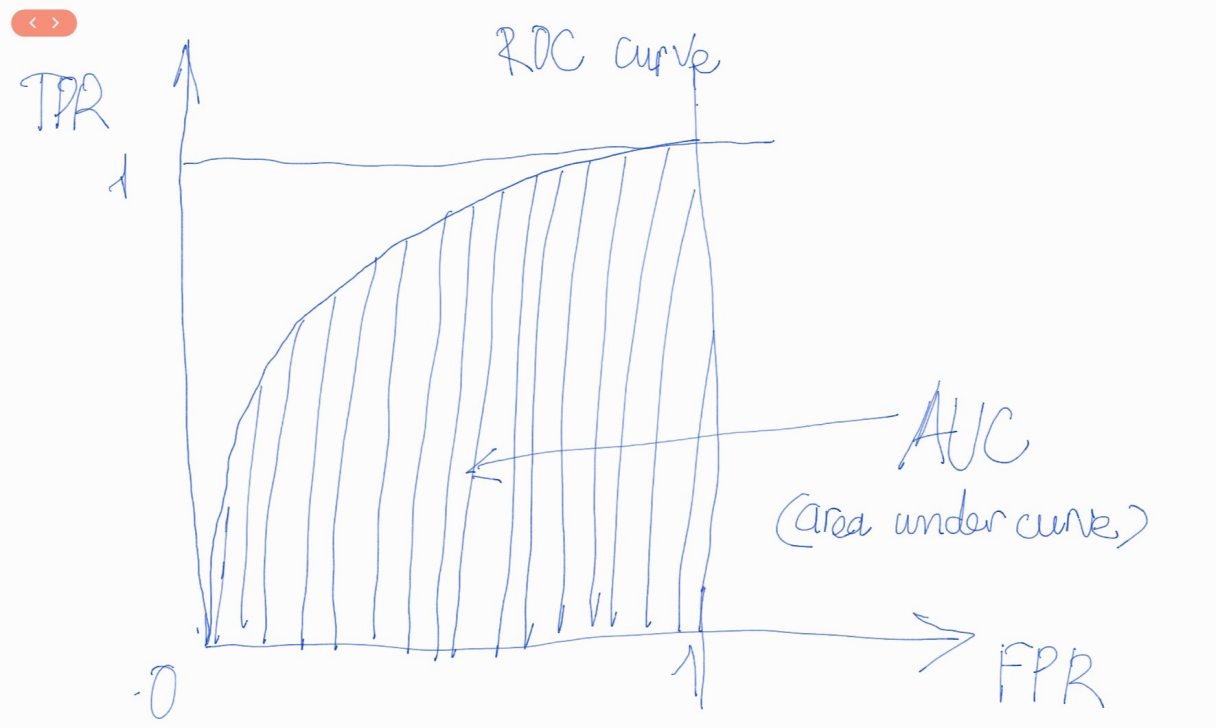
- TPR (true positive rate): Hay còn gọi là recall hoặc sensitivity. Là tỷ lệ các trường hợp phân loại đúng positive trên tổng số các trường hợp thực tế là positive. Chỉ số này sẽ đánh giá mức độ dự báo chính xác của mô hình trên positive. Khi giá trị của nó càng cao, mô hình dự báo càng tốt **trên nhóm positive**. Nếu $\mathrm{TPR} = 0.9$, chúng ta tin rằng 90% các mẫu thuộc nhóm positive đã được mô hình phân loại đúng.

$$\mathrm{TPR/recall/sensitivity} = \frac{TP}{\text{total positive}}$$

- FPR (false positive rate): Tỷ lệ dự báo sai các trường hợp thực tế là negative thành thành positive trên tổng số các trường hợp thực tế là negative. Nếu giá trị của $\mathrm{FPR} = 0.1$, mô hình đã dự báo sai 10% trên tổng số các trường hợp là negative. Một mô hình có FPR càng thấp thì mô hình càng chuẩn xác vì sai số của nó **trên nhóm negative** càng thấp. Phần bù của FPR là specificity đo lường tỷ lệ dự báo đúng các trường hợp negative trên tổng số các trường hợp thực tế là negative.

$$\mathrm{FPR} = 1 - \mathrm{specificity} = \frac{FP}{\text{total negative}}$$

Đồ thị ROC là một đường cong cầu lồi dựa trên TPR và FPR có hình dạng như bên dưới:



AUC là chỉ số được tính toán dựa trên đường cong ROC (receiving operating curve) nhằm **đánh giá khả năng phân loại** của mô hình tốt như thế nào ? Phần diện tích gạch chéo nằm dưới đường cong ROC và trên trục hoành là AUC (area under curve) có giá trị nằm trong khoảng [0, 1]. Khi diện tích này càng lớn thì đường cong ROC có xu hướng tiệm cận đường thẳng $y = 1$ và khả năng phân loại của mô hình càng tốt. Khi đường cong ROC nằm sát với đường chéo đi qua hai điểm (0, 0) và (1, 1), mô hình sẽ tương đương với một phân loại ngẫu nhiên.

AUC được tính toán như sau:

Top

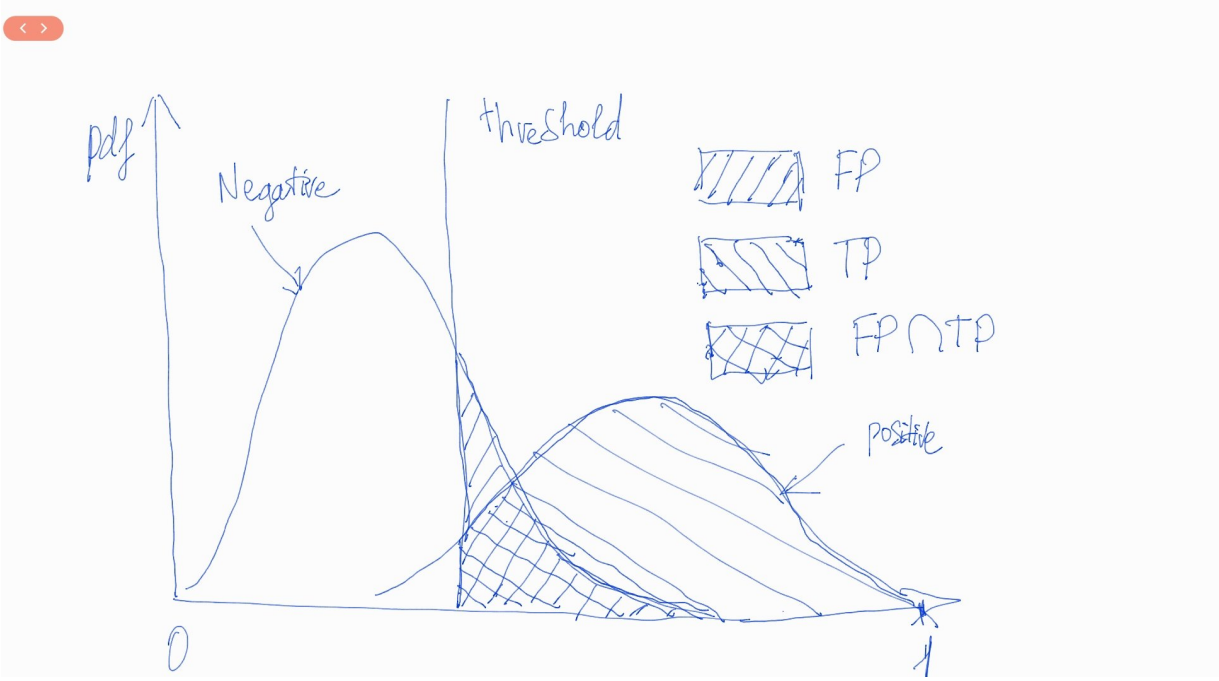
```
from sklearn.metrics import auc, roc_curve
fpr, tpr, thres = metrics.roc_curve(y_label, y_pred)
# Tính toán auc
auc(fpr, tpr)
```

Biểu diễn đường cong ROC:

```
def _plot_roc_curve(fpr, tpr, thres):
    roc = plt.figure(figsize = (10, 8))
    plt.plot(fpr, tpr, 'b-', label = 'ROC')
    plt.plot([0, 1], [0, 1], '--')
    plt.axis([0, 1, 0, 1])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC Curve')

_plot_roc_curve(fpr, tpr, thres)
```

11. Mối quan hệ giữa TPR và FPR

TPR và FPR sẽ có mối quan hệ cùng chiều. Thật vậy, chúng ta sẽ cùng diễn giải điều này qua hình vẽ bên dưới.



Hình 1: Đồ thị phân phối của mật độ xác suất (probability densitiy function - pdf) của điểm số nhóm negative bên trái và nhóm positive bên phải. Mô hình sẽ căn cứ vào đường thẳng threshold vuông góc với trục hoành (y) để đưa ra dự báo là positive hay negative. Nếu điểm số nằm bên trái threshold thì sẽ được dự báo là negative và nằm bên phải được dự báo là positive. Như vậy trên hình vẽ, phần diện tích FP sẽ là false positive rate phần diện tích TP sẽ là true positive rate. Khi ta dịch chuyển ngưỡng threshold từ trái sang phải thì các phần diện tích FP và TP sẽ cùng tăng dần. Điều này tương ứng với mối quan hệ giữa TPR (true positive rate) và FPR (false positive rate) là đồng biến theo sự thay đổi của threshold.

Bây giờ bạn đã hiểu tại sao đường cong ROC lại là một đường đồng biến rồi chứ ?

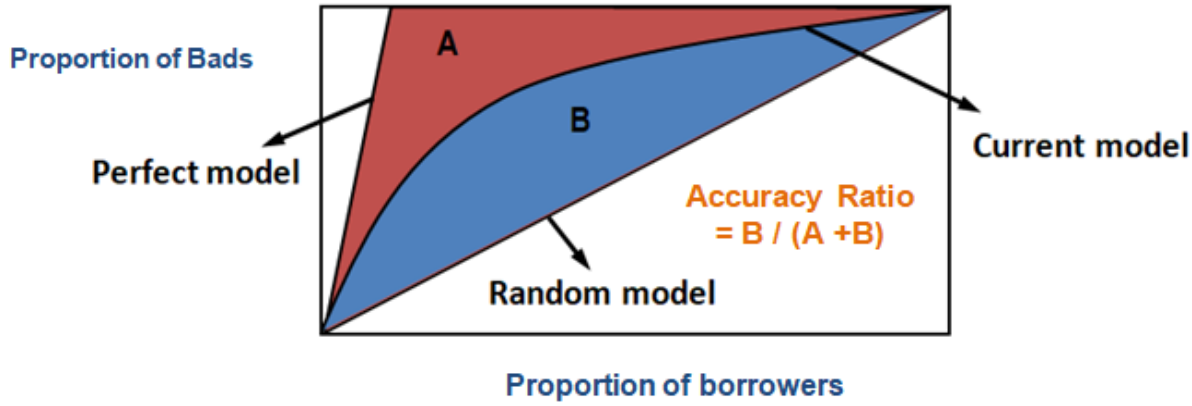
Top

Ngoài ra mô hình dự báo xác suất của chúng ta sẽ càng tốt nếu đồ thị phân phối xác suất của negative và positive có sự tách biệt càng lớn. Khi đó phần diện tích chồng lấn giữa hai phân phối càng nhỏ và mô hình giảm thiểu tỷ lệ dự báo nhầm. Đồng thời các phân phối xác suất giữa negative và positive càng cách xa nhau thì đồ thị ROC càng lồi. Tính chất lồi của ROC được thể hiện qua độ lớn của phần diện tích AUC.

12. gini và CAP

Trong lĩnh vực credit risk, các mô hình scorecard sử dụng hệ số gini làm thước đo đánh giá sức mạnh phân loại của các mô hình. Hệ số này cho thấy khả năng một hồ sơ sẽ vỡ nợ trong tương lai được nhận biết từ mô hình là bao nhiêu phần trăm. Một mô hình scorecard càng mạnh thì hệ số gini càng cao và phân phối điểm số của hai nhóm GOOD và BAD sẽ càng khác biệt. Giá trị của gini nằm giao động trong khoảng [0, 1].

Một hệ số khác tương tự như gini đó là **CAP** (*Cumulative Accuracy Profile*). Hệ số này được tính toán dựa trên đường cong CAP có biểu diễn như hình bên dưới:



Hình 2 Hệ số CAP và đường cong CAP của mô hình scorecard. Trên đồ thị, trục hoành biểu diễn tỷ lệ phần trăm tích lũy của số lượng hồ sơ vay và trục tung biểu diễn phần trăm tích lũy của số lượng hồ sơ vay của nhóm BAD được thống kê từ phần trăm mẫu được rút ra tương ứng trên trục hoành. Các hồ sơ sẽ được sắp xếp theo điểm số giảm dần. Đầu tiên chúng ta sẽ lấy ra một tỷ lệ $x\%$ hồ sơ có điểm số cao nhất tương ứng với điểm x trên trục hoành. Từ mẫu $x\%$ này, chúng ta thống kê được $y\%$ tỷ lệ các hồ sơ BAD được phát hiện. Sau đó gia tăng dần kích thước mẫu tích lũy ta sẽ thu được đường CAP như đường *curent model* trên hình vẽ.

Trên hình vẽ chúng ta có 3 đường cong CAP đó là *perfect model, current model, random model* lần lượt tương ứng với các model hoàn hảo (*perfect model*), model hiện tại và model ngẫu nhiên. Model hoàn hảo là mô hình phân loại một cách hoàn hảo các hồ sơ nợ xấu. Đường CAP của mô hình hoàn hảo sẽ tiệm cận với đường thẳng $y = 1$ cho thấy rằng chúng ta có thể lựa chọn một ngưỡng điểm nào đó nằm giữa (0, 1) sao cho mô hình phân loại được 100% các trường hợp vỡ nợ. Mô hình hoàn hảo rất ít khi đạt được trên thực tế và nếu có một mô hình gần tiệm cận với đường thẳng $y = 1$ thì đó là một mô hình rất rất tốt.

Đối lập với đường CAP hoàn hảo là đường CAP ngẫu nhiên. Đường CAP này biểu diễn kết quả của một sự phân loại ngẫu nhiên các nhãn BAD nên tỷ lệ hồ sơ BAD phân phối đều trên toàn miền điểm số. Do đó hình dạng của đường CAP ngẫu nhiên sẽ tiệm cận với đường chéo chính đi qua (0, 0) và (1, 1).

Tại sao phân phối xác suất tích lũy của BAD lại là một đường cong lồi ?

- Giả sử chúng ta lựa chọn tập mẫu S gồm $x\%$ quan sát có điểm *cao nhất* (lưu ý là các quan sát đã được sắp xếp theo điểm số giảm dần). Do BAD có phân phối chủ yếu tập trung vào nhóm có điểm số cao nên tỷ lệ các hồ sơ được dự báo BAD trên tổng số hồ sơ nhãn BAD

top

trong S sẽ lớn hơn tỷ lệ tích lũy các quan sát $x\%$. Tỷ lệ này đồng thời cũng chính là TPR (true positive rate) trên S.

- Ở những $x\%$ cao thì các quan sát được thêm vào có điểm số nhỏ dần và do đó tốc độ tăng của TPR giảm dần. Do đó đường CAP của mô hình hiện tại có hình dạng là một đường cong lồi.

Công thức CAP:

Hầu hết các mô hình có hình dạng của đường cong CAP tương tự như đường current model. Tức là nằm giữa đường CAP hoàn hảo và CAP ngẫu nhiên. Một mô hình càng tốt nếu đường CAP của nó càng gần đường hoàn hảo và khi đường CAP càng gần đường ngẫu nhiên thì kết quả dự báo của mô hình càng kém. Chỉ số CAP sẽ được tính toán dựa trên phần diện tích A, B nằm giữa các đường CAP hoàn hảo, hiện tại và ngẫu nhiên như trên hình vẽ theo công thức:

$$\text{CAP} = \frac{A}{A+B}$$

Visualize đường cong CAP như thế nào ?

Để vẽ đường cong CAP chúng ta lần lượt thực hiện các bước sau:

- B1: Sắp xếp xác suất vỡ nợ được dự báo theo thứ tự *giảm dần* và chia nó thành 10 phần (decile) với số lượng quan sát đều nhau. Bạn cũng có thể lựa chọn chia thành 15, 20 phần, tùy theo kích thước tập huấn luyện lớn hay nhỏ. Cách phân chia này sẽ xếp hạng những người vay rủi ro nhất có nhóm xếp hạng (*rating grade*) thấp nhất và những người vay an toàn nhất nên có nhóm xếp hạng cao nhất.
- B2: Tính số người vay trong mỗi nhóm (cột *number of borrowers*).
- B3: Tính số lượng khách hàng nợ xấu trong mỗi nhóm (cột *number of bads*).
- B4: Tính số lượng khách hàng nợ xấu tích lũy trong mỗi nhóm (cột *cumulative bads*). Nợ xấu tích lũy của một nhóm xếp hạng thứ i sẽ bằng tổng nợ xấu của các nhóm xếp hạng trước đó từ $1, 2, \ldots$ cho tới i.
- B5: Tính tỷ lệ phần trăm khách hàng nợ xấu trong mỗi nhóm (cột *% of bads*) có giá trị bằng cột *number of bads* chia cho tổng số lượng hồ sơ BAD.
- B6: Tính tỷ lệ phần trăm tích lũy của khách hàng nợ xấu trong mỗi phần (cột *cumulative % of bads*) được tính dựa trên tổng tích lũy của cột *% of bads*.

	Rating Grade	Number of Borrowers	Number of Bads	Cumulative Bads	% of Bads	Cumulative % of Bads
Worst Score	1	2500	2179	2179	44.71	44.71
	2	2500	1753	3932	35.97	80.67
	3	2500	396	4328	8.12	88.80
	4	2500	111	4439	2.28	91.08
	5	2500	110	4549	2.26	93.33
	6	2500	85	4634	1.74	95.08
	7	2500	67	4701	1.37	96.45
	8	2500	69	4770	1.42	97.87
	9	2500	49	4819	1.01	98.87
Best Score	10	2500	55	4874	1.13	100.00
		25000	4874			

Khi đó chúng ta sẽ thu được cột cuối cùng tương ứng với giá trị trục tung của đường cong CAP tại các điểm giá trị 10% liên tiếp của trục hoành.

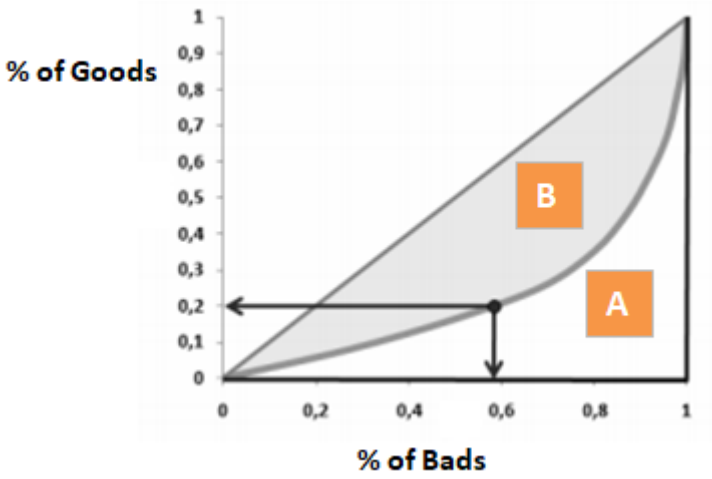
Top

```
# 1. Đường cong perfect model
# Số lượng quan sát nhãn positive
no_positive = np.sum(y_train)
# Số lượng quan sát
total = len(y_train)

plt.plot([0, no_positive, total],
         [0, no_positive, no_positive],
         c = 'grey',
         linewidth = 2,
         label = 'Perfect Model')

# 2. Đường cong ngẫu nhiên

plt.plot([0, total],
         [0, no_positive],
         c = 'r', linestyle = '--', label = 'Random Model')

# 3. Đường cong CAP của mô hình hiện tại
# Sắp xếp nhãn y_train theo thứ tự xác suất giảm dần
y_label_sorted = [y for _, y in sorted(zip(y_prob, y_train))]
# Tổng lũy kế số lượng các quan sát positive theo xác suất giảm dần
y_values = np.append([0], np.cumsum(y_label_sorted))
# Tổng lũy kế số lượng các quan sát
x_values = np.arange(0, total + 1)
# Đường CAP của current model
plt.plot(x_values,
         y_values,
         c = 'b',
         label = 'Current CAP model',
         linewidth = 4)

# Plot information
plt.xlabel('Total observations', fontsize = 16)
plt.ylabel('Positive observations', fontsize = 16)
plt.title('Cumulative Accuracy Profile', fontsize = 16)
plt.legend(loc = 'lower right', fontsize = 16)

```

Đường cong lorenz và hệ số gini

Đường cong lorenz được sử dụng để mô tả sự bất bình đẳng trong phân phối giữa GOOD và BAD. Ý nghĩa của nó tương tự như đường cong CAP. Nhưng chúng ta sẽ thay phân phối tích lũy của số lượng mẫu bằng phân phối tích lũy của GOOD. Đồ thị của đường cong lorenz có hình dạng như bên dưới :

Top



Hệ số gini thể hiện mức độ cải thiện của mô hình trong khả năng phân loại GOOD và BAD so với mô hình ngẫu nhiên. Giá trị của hệ số gini được tính bằng diện tích :

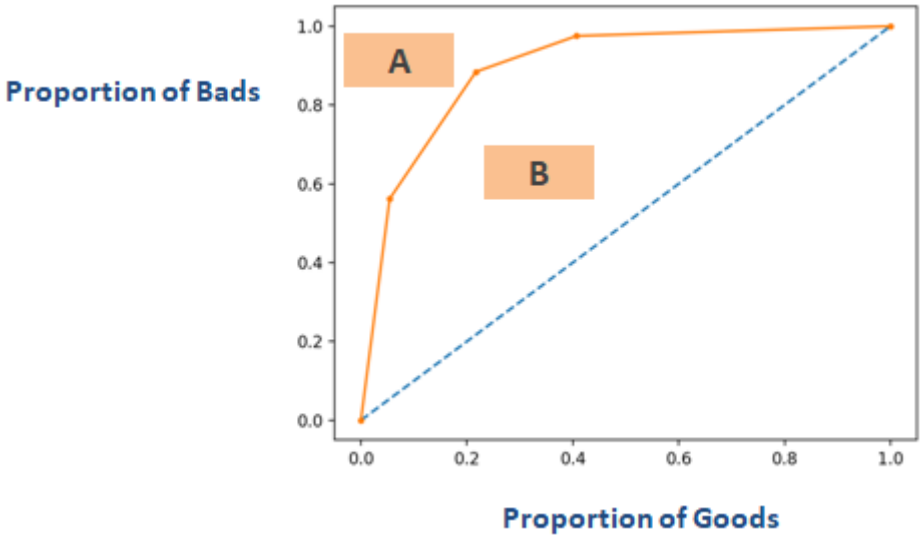
$$\text{gini} = \frac{B}{A+B} = 2B$$

khai triển đẳng thức sau cùng được suy ra từ diện tích $A + B = 0.5$

Mối liên hệ giữa gini và AUC

Ngoài ra chúng ta còn có mối liên hệ giữa hệ số gini và AUC theo phương trình sau:

$$\text{gini} = 2 \times \text{AUC} - 1$$



Thật vậy, nếu chúng ta thể hiện trên đồ thị đồng thời đường cong ROC và lorenz thì hai đường này sẽ trùng nhau. Giả sử A là phần diện tích nằm dưới đường thẳng $y = 1$ và nằm trên đường cong ROC, B là phần diện tích nằm trên đường chéo chính và dưới đường cong ROC. Khi đó ta sẽ nhận thấy rằng:

$$\text{gini} = 2 * B$$

$$\text{AUC} = B + 0.5$$

Do đó

$$\text{gini} = 2 \times \text{AUC} - 1$$

Top

13. Tổng kết

Như vậy qua bài viết này các bạn đã nắm trong tay khá nhiều các chỉ số để đánh giá mô hình phân loại trong machine learning. Đây là những kiến thức cơ bản nhưng lại rất quan trọng mà chúng ta cần phải nắm vững để lựa chọn được mô hình tốt nhất. Đồng thời chúng ta không chỉ biết cách áp dụng mà còn hướng tới hiểu sâu về công thức và ý nghĩa thực tiễn của từng chỉ số.

Để viết được tài liệu này, một phần không thể thiếu là những tài liệu tham khảo bên dưới.

14. Tài liệu tham khảo

1. Understanding AUC - ROC Curve (https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)
2. Trade off precision and recall - Andrew Ng (https://www.youtube.com/watch?v=W5meQnGACGo)
3. Receiver Operating Characteristic (ROC) with cross validation (https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc_crossval.html#sphx-glr-auto-examples-model-selection-plot-roc-crossval-py)
4. Accuracy, Precision, Recall or F1? (https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9)
5. gini cumulative accuracy profile auc (https://www.listendata.com/2019/09/gini-cumulative-accuracy-profile-auc.html)
6. Classification Accuracy is Not Enough: More Performance Measures You Can Use (https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/)

Top