Chương 3: Phân lớp và cách đánh giá bộ phân lớp

(Tài liệu nội bộ)

Tháng 3 năm 2020

Nội dung trình bày

1. Bài toán phân lớp nhị phân
2. Đánh giá hiệu quả phân lớp
3. Bài toán Phân lớp đa lớp
4. Bài toán Phân lớp đa nhãn
5. Bài toán Phân lớp đa đầu vào
6. Thực hành

Bộ dữ liệu MNIST

- Sử dụng cho bài toán nhận diện chữ viết tay, cụ thể là bài toán Nhận diện 10 chữ số (Hình 3.1).
- 70000 ảnh chữ số viết tay của học sinh phổ thông và nhân viên của Cục điều tra dân số Mỹ.
- Ảnh có kích thước nhỏ và được gán nhãn là chữ số được nhận diện.
- "Hello world" của máy học.
- Được trộn và chia sẵn thành tập huấn luyện (60000 ảnh đầu) và tập kiểm thử (10000 ảnh sau).
- Scikit-Learn có hàm giúp tải những bộ dữ liệu nổi tiếng.*

Bộ dữ liệu MNIST

Nội dung trình bày

1. Bài toán phân lớp nhị phân

Bài toán phân lớp nhị phân

- Đơn giản hóa bài toán nhận diện 10 chữ số thành bài toán phân lớp nhị phân.
- VD: **Bài toán nhận dạng chữ số 5**: là một bài toán phân lớp nhị phân
  - Đầu vào: Một ảnh chữ số viết tay
  - Đầu ra: **True** = "là chữ số 5" (dương tính) hoặc **False** = "không phải là chữ số 5" (âm tính).
- VD: **Bài toán thử nCov?**
  - Đầu vào: Mẫu dịch họng
  - Đầu ra: **True** = "dương tính" hoặc **False** = "âm tính".

Nội dung trình bày

❷ Đánh giá hiệu quả phân lớp
- Ma trận nhầm lẫn
- Accuracy
- Đường cong ROC

Ma trận nhầm lẫn

- Confusion matrix
- Là một cách tốt hơn để đánh giá hiệu quả của một bộ phân lớp.
- Đếm số điểm dữ liệu đáng lẽ có nhãn A nhưng bị dự đoán nhầm sang nhãn B, điền vào ma trận với dòng là lớp đúng và cột là lớp dự đoán.
- VD: Ma trận nhầm lẫn 2 lớp: Muốn biết số lần số khác-5 bị phân lớp thành số 5 thì tra ô ở dòng khác-5 (Non-5), cột 5 của ma trận nhầm lẫn.

	True\Predicted	Non-5	5
Negative class	Non-5	53057 True negatives	1522 False positives
Positive class	5	1325 False negativs	4096 True positives

Ma trận nhầm lẫn đã chuẩn hóa

Trước chuẩn hóa:

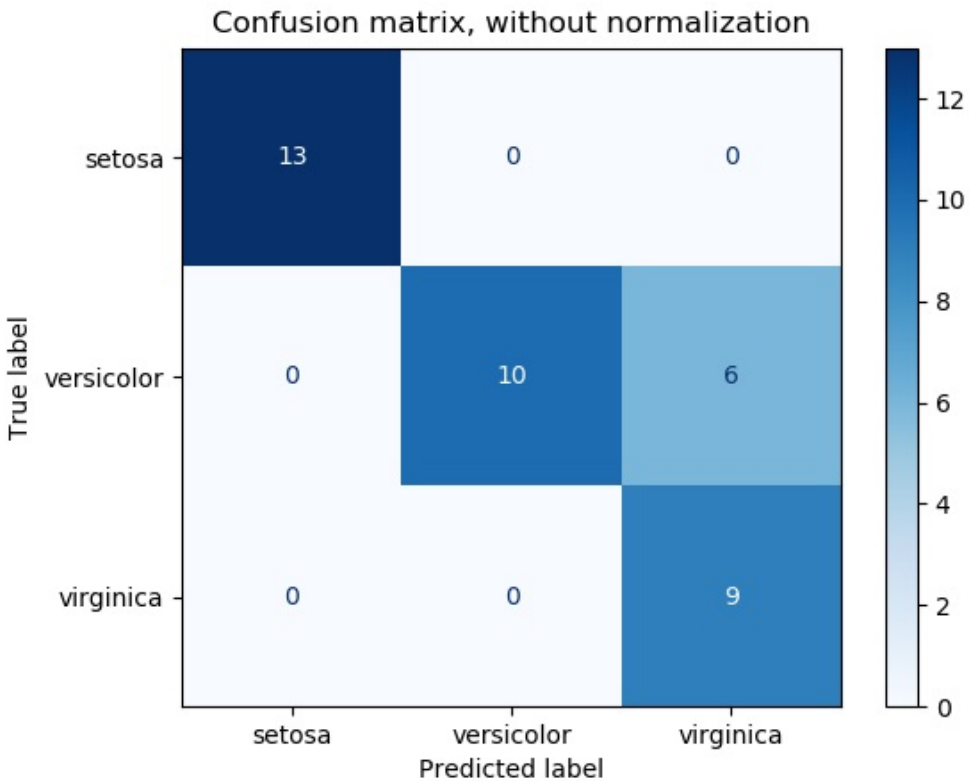
	True\Predicted	Non-5	5
Negative class	Non-5	53057 True negatives	1522 False positives
Positive class	5	1325 False negativs	4096 True positives

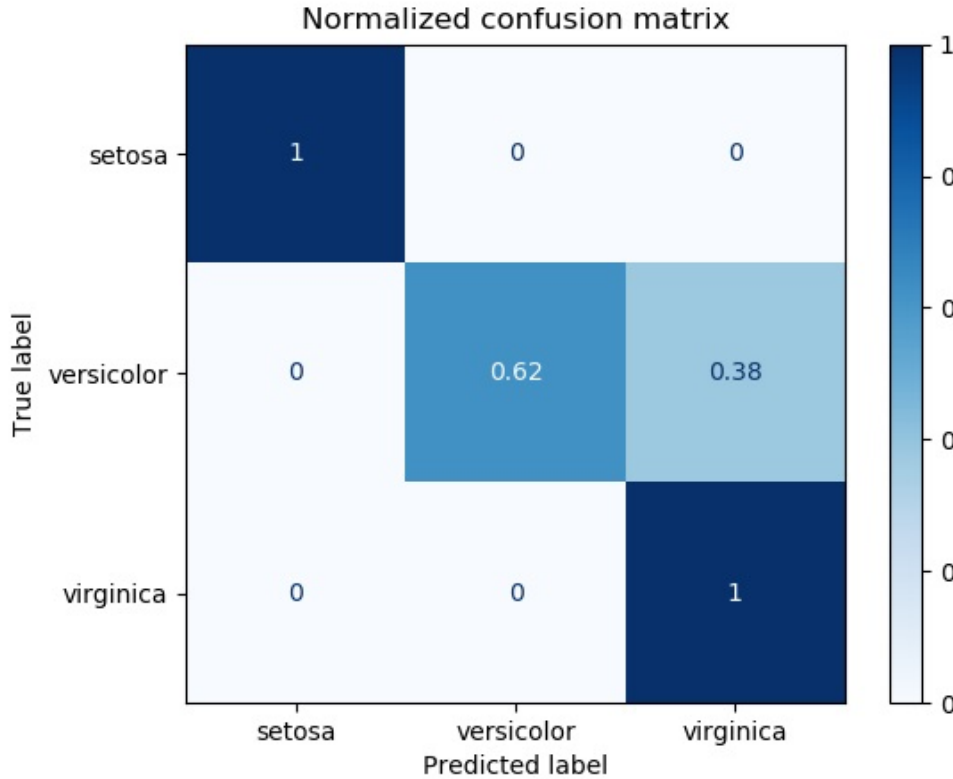
Sau chuẩn hóa:

True\Predicted	Non-5	5
Non-5	97% TNR = TN/(FP + TN)	3% FPR = FP/(FP + TN)
5	24% FNR = FN/(TP + FN)	76% TPR = TP/(TP + FN)

Bộ phân lớp lý tưởng có tất cả các số **không nằm** trên đường chéo chính đều bằng 0.

Biển diễn hình ảnh của Ma trận nhầm lẫn đã chuẩn hóa





Ma trận nhầm lẫn đa lớp

- VD: Muốn biết số lần số 5 bị phân lớp thành số 3 thì tra ô ở dòng 5, cột 3 của ma trận nhầm lẫn.

	0	1	2	3	4	5	6	7	8	9
0	967	0	2	1	0	2	7	0	1	0
1	0	1126	4	2	0	0	2	0	0	1
2	5	2	1001	2	1	0	1	5	15	0
3	0	0	8	953	1	17	0	5	21	5
4	1	1	3	0	948	0	3	4	2	20
5	2	0	1	8	1	853	4	3	13	7
6	4	2	0	0	1	2	947	0	2	0
7	1	8	11	2	5	0	0	976	2	23
8	3	0	4	4	0	8	0	3	948	4
9	2	3	3	5	21	6	1	19	5	944

- **Nếu số lớp nhiều, không tiện vẽ ma trận nhầm lẫn thì làm thế nào?**

Độ chính xác accuracy

- Độ chính xác Accuracy = tỉ lệ điểm dữ liệu được dự đoán đúng trên tổng số điểm dữ liệu trong tập kiểm thử = (TN+TP)/(P+N) = (TN+TP)/(TP+TN+FP+FN)

	True\Predicted	Non-5	5
Negative class	Non-5	53057 True negatives	1522 False positives
Positive class	5	1325 False negativs	4096 True positives

- Điều gì xảy ra nếu chỉ 10% dữ liệu là số 5, 90% dữ liệu là không phải 5 và bộ phân lớp dự đoán mọi điểm dữ liệu là không phải 5?
  - Accuracy chắc chắn lớn hơn 90% nhưng liệu đây là bộ phân lớp tốt?
  - Bộ dữ liệu gọi là **lệch/không cân bằng** khi phân bố các nhãn không đồng đều (imbalanced/skewed dataset).
  - Accuracy không phải là độ đo tốt khi dữ liệu không cân bằng.

Độ chính xác, độ phủ và độ đo F1

	True\Predicted	Non-5	5
Negative class	Non-5	53057 True negatives	1522 False positives
Positive class	5	1325 False negativs	4096 True positives

- **Độ chính xác**: $precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$
  - Là độ chính xác accuracy của dự đoán dương tính (tỉ lệ dự đoán dương tính đúng).
- **Độ phủ**: $recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$
  - Tỉ lệ những điểm dữ liệu dương tính (positive) được dự đoán đúng.
  - Còn gọi là độ nhạy (sensitivity) hoặc (true positive rate).
- **Độ đo F1**: $F1 = \frac{2 \times precision \times recall}{precision + recall}$
  - Trung bình điều hòa của precision và recall, là độ đo kết hợp hai độ đo.
  - Dùng để so sánh 2 bộ phân lớp.

Độ chính xác, độ phủ và độ đo F1

	True\Predicted	Non-5	5
Negative class	Non-5	53057 True negatives	1522 False positives
Positive class	5	1325 False negativs	4096 True positives

- **Độ chính xác**: $precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$
- **Độ phủ**: $recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$
- **Độ đo F1**: $F1 = \frac{2 \times precision \times recall}{precision + recall}$
- F1 dùng khi precision và recall có vai trò quan trọng như nhau.
- Một số bài toán precision có thể quan trọng hơn recall hoặc ngược lại.
  - Precision cao: "Dự đoán dương tính là phải thật sự dương tính càng nhiều càng tốt".
  - Recall cao: "Có bao nhiêu điểm thật sự dương tính là phải dự đoán được hết".
- Ví dụ:
  - Bài toán phát hiện video clip an toàn cho trẻ em?
  - Bài toán phát hiện trộm cắp trong siêu thị?

Độ chính xác, độ phủ và độ đo F1

	True\Predicted	Non-5	5
Negative class	Non-5	53057 True negatives	1522 False positives
Positive class	5	1325 False negativs	4096 True positives

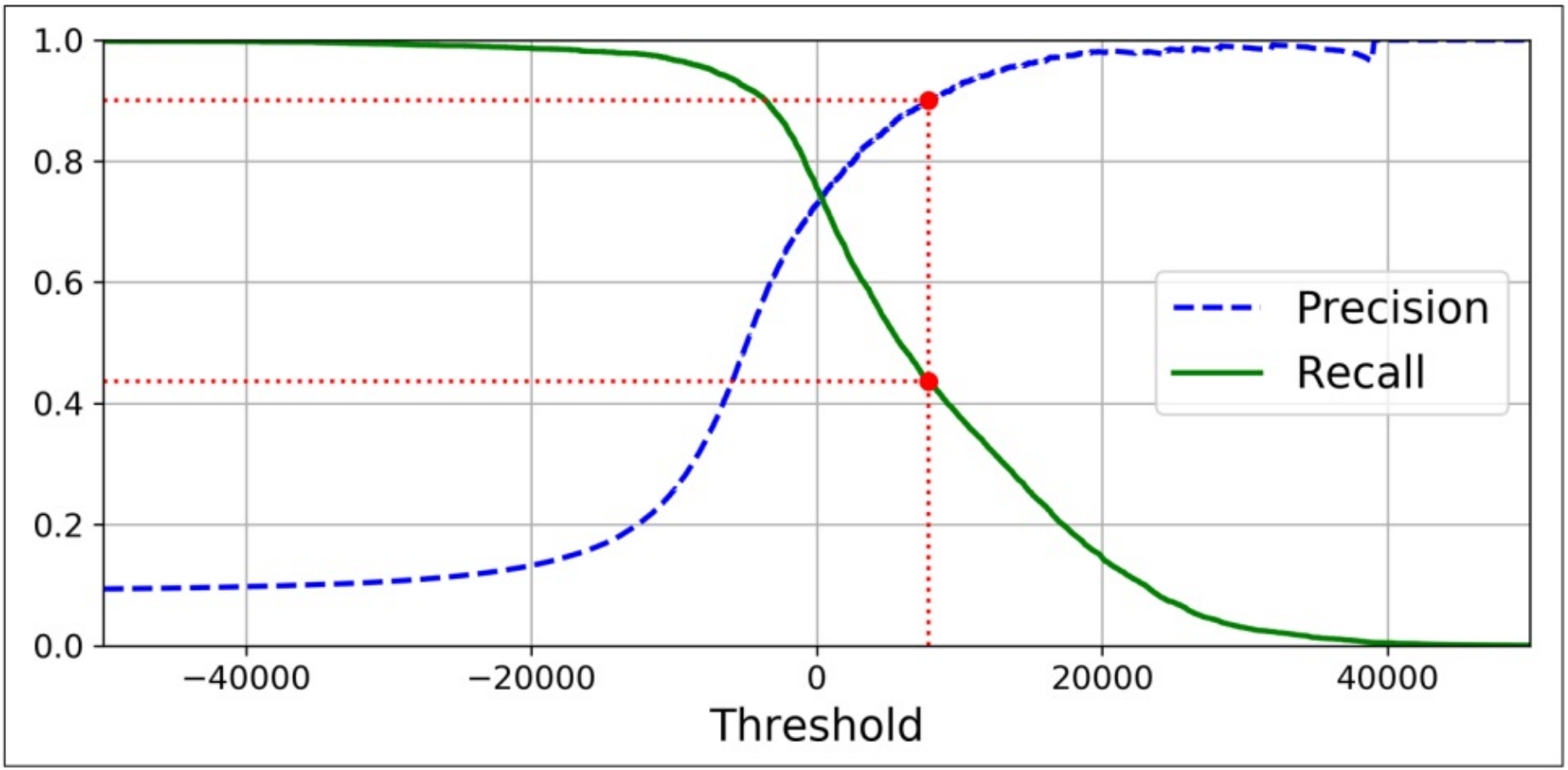
- **Độ chính xác**: $precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$
- **Độ phủ**: $recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$
- **Độ đo F1**: $F1 = \frac{2 \times precision \times recall}{precision + recall}$
- F1 dùng khi precision và recall có vai trò quan trọng như nhau.
- Một số bài toán precision có thể quan trọng hơn recall hoặc ngược lại. Nôm na:
  - Precision cao: "Đã dự đoán **dương tính** là phải thật sự **dương tính**".
  - Recall cao: "Có bao nhiêu **điểm** thật sự là **dương tính** là phải dự đoán được hết".
- Ví dụ:
  - Bài toán phát hiện video clip an toàn cho trẻ em? (Precision cao).
  - Bài toán phát hiện trộm cắp trong siêu thị? (Recall cao).

Thẩm định chéo với K phần

- K-fold cross validation
- Khi tập dữ liệu huấn luyện nhỏ thì nên dùng phương pháp này.
- Chia tập dữ liệu huấn luyện thành K phần bằng nhau, dùng K-1 phần để huấn luyện mô hình rồi đánh giá trên 1 phần dữ liệu còn lại.
- Lặp lại K lần đánh giá như thế, accuracy (/precision/recall/f1) của mô hình được tính bằng trung bình cộng của K kết quả đo.

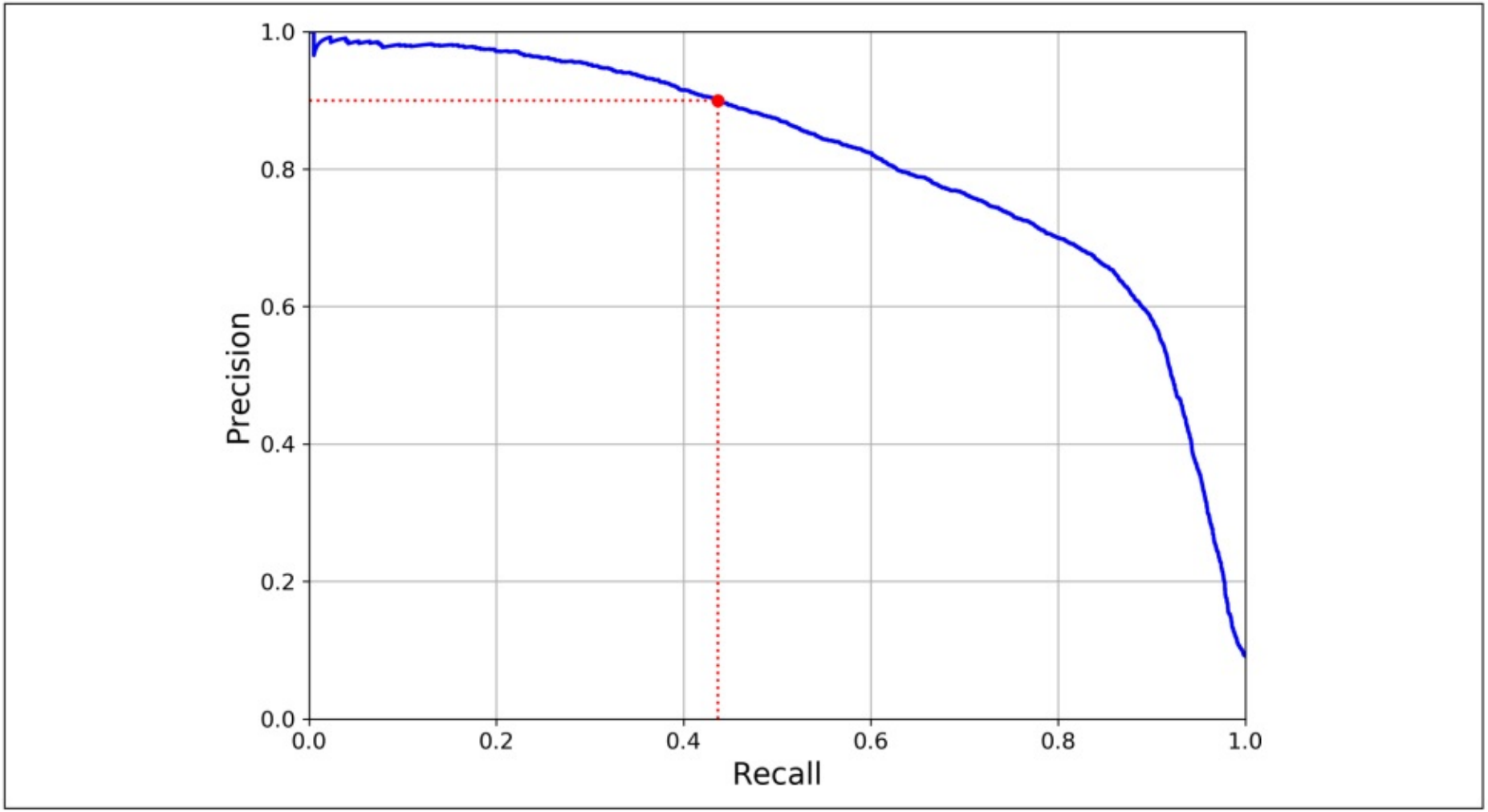
Sự được mất giữa độ chính xác và độ phủ

- Đường cong precision, recall theo các ngưỡng quyết định khác nhau. (Ngưỡng quyết định của thuật toán học SGD sử dụng)
- Scikit-Learn cung cấp sẵn công cụ để vẽ các đồ thị, chọn ngưỡng.



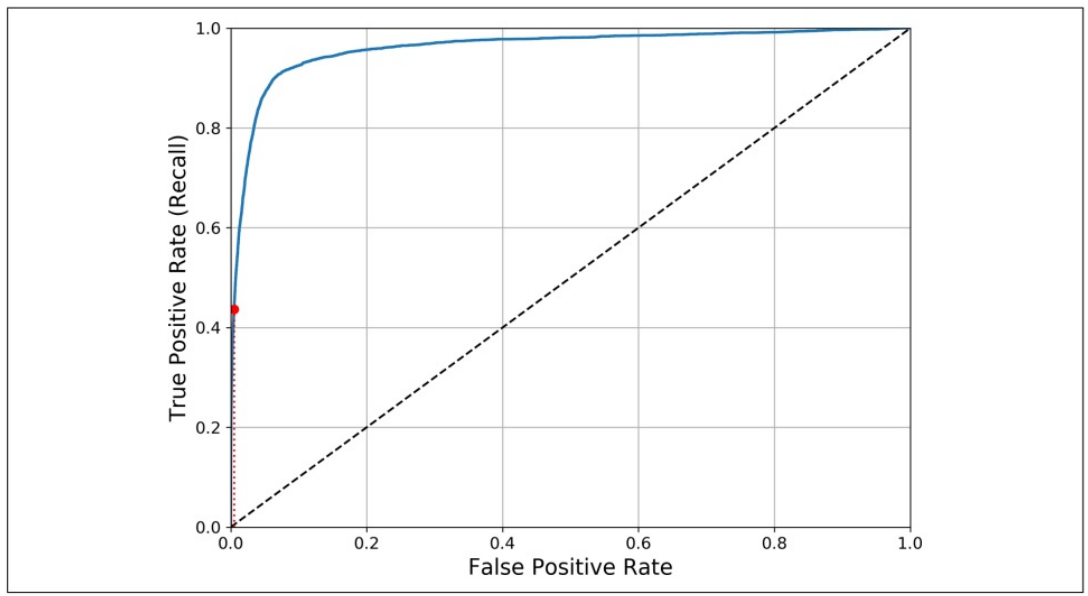
Sự được mất giữa độ chính xác và độ phủ

- Đường cong precision theo recall



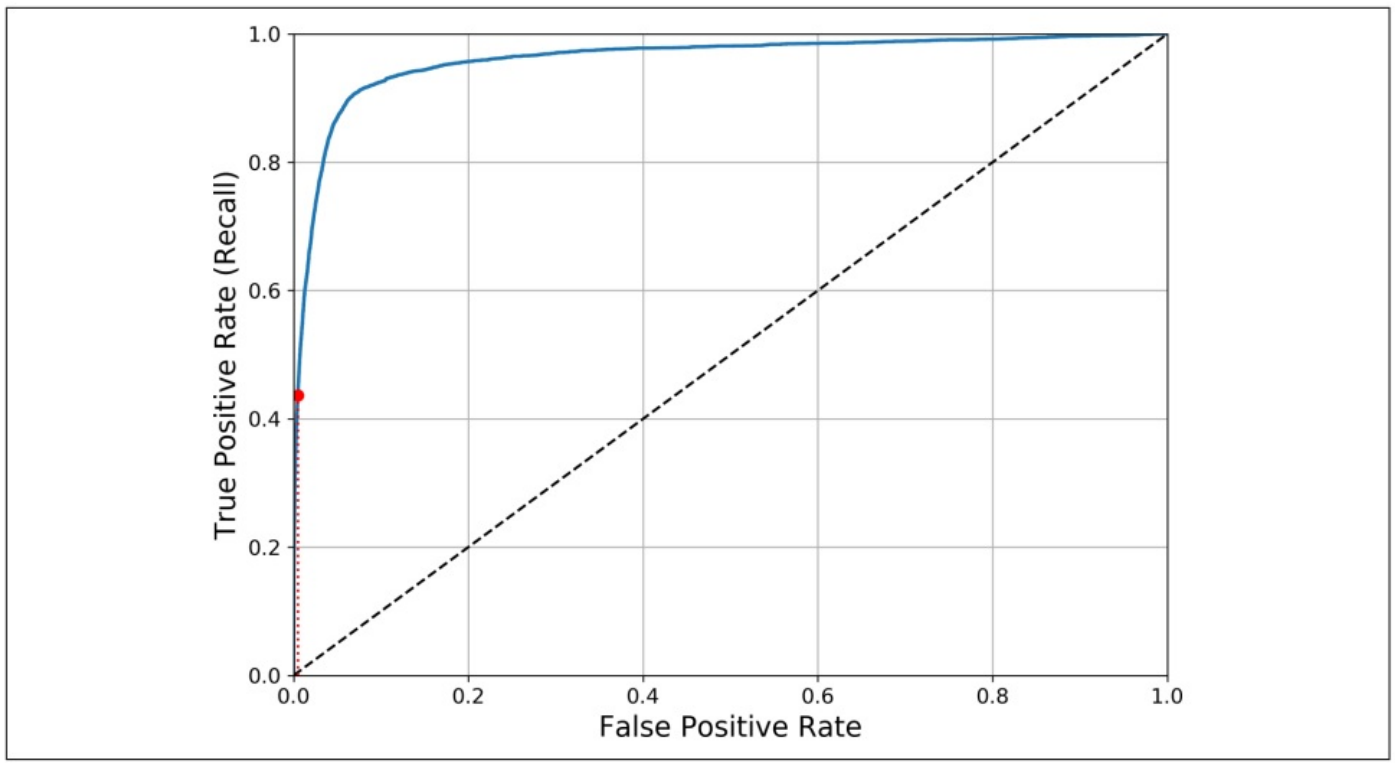
Đường cong ROC

- = Receiver operating characteristic curve
- Tương tự như đường cong precision theo recall, nhưng cho TPR theo FPR.
  - TPR (true positive rate): **độ nhạy** (recall, sensitivity), là tỉ lệ điểm dữ liệu dương tính được phân lớp đúng.
  - FPR (false positive rate): tỉ lệ điểm dữ liệu âm tính bị phân lớp, = 1-TNR.
  - TNR (true negative rate): **độ đặc hiệu** (specificity), là tỉ lệ điểm dữ liệu âm tính được phân lớp đúng.
- Nói cách khác, đường cong ROC vẽ độ nhạy theo **1-độ đặc hiệu**.



Đường cong ROC

- Đường gạch đứt là đường cong ROC của bộ phân lớp ngẫu nhiên. Bộ phân lớp tốt phải càng xa về góc trên trái càng tốt.
- **Diện tích bên dưới đường cong ROC gọi là AUC** (Area Under the Curve) cũng cho biết hiệu quả của bộ phân lớp.
  - Bộ phân lớp lý tưởng có AUC bằng 1.
  - Bộ phân lớp ngẫu nhiên có AUC bằng 0.5.
- Một cách để so sánh hiệu quả của 2 bộ phân lớp là **so sánh AUC** của chúng. (Scikit-Learn có hàm để tính AUC của ROC)



Nội dung trình bày

3. Bài toán Phân lớp đa lớp

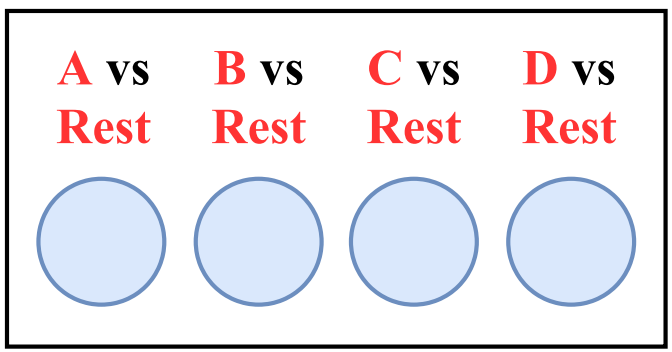
Giới thiệu về Phân lớp đa lớp

- Phân lớp đa lớp (multiclass classification/multinominal classification)
- **Phân lớp đa lớp** phân loại điểm dữ liệu vào một trong nhiều lớp (nhiều hơn 2 lớp): MNIST.
- Một số phương pháp phân lớp **có bản chất là phân lớp đa lớp (SGD, Naive Bayes, Random Forest)** , một số khác **thuần là phân lớp nhị phân (Logistic Regression, SVM)**.
- Cách tạo bộ phân lớp đa lớp từ bộ phân lớp nhị phân?

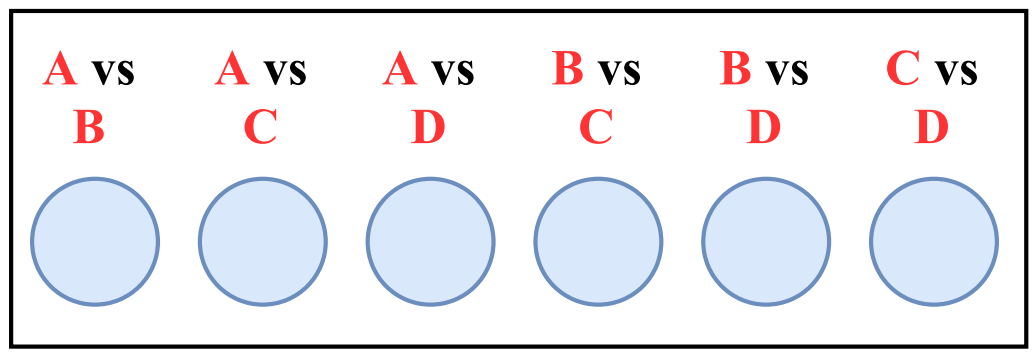
Tạo bộ phân lớp đa lớp từ bộ phân lớp nhị phân

Có thể tạo ra bộ phân lớp đa lớp từ bộ phân lớp nhị phân theo một trong 2 chiến lược.

- Một với tất cả nhãn còn lại (OvR): sử dụng n bộ phân lớp nhị phân.
  - MNIST: 10 bộ phân lớp nhị phân cho 10 chữ số 0, 1, 2, ...
  - Chọn chữ số có điểm số dự đoán cao nhất.
- Một với một (OvO): sử dụng $n(n-1)/2$ bộ phân lớp nhị phân.
  - MNIST: 45 bộ phân lớp nhị phân 0-1, 0-2, 1-2, 1-3,...
  - Chọn chữ số được dự đoán bởi nhiều bộ nhận diện nhất.

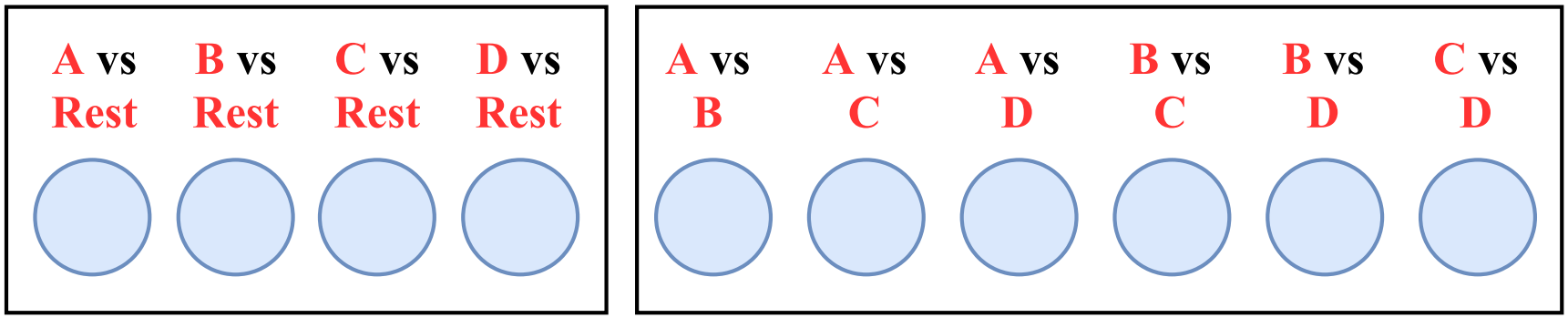


One vs Rest



One vs One

Phân lớp đa lớp với Scikit-Learn



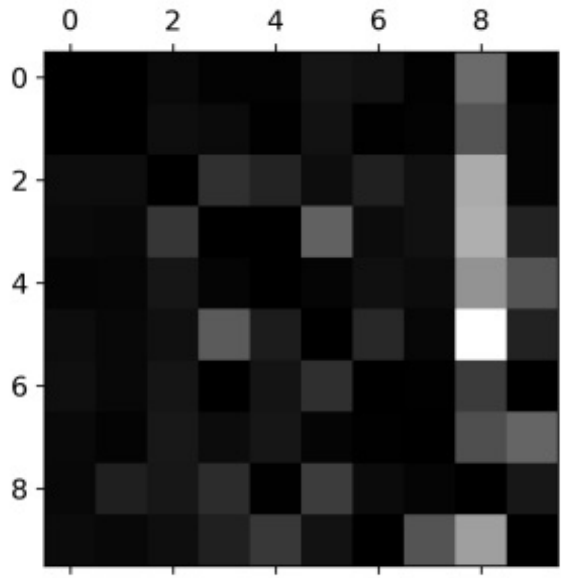
One vs Rest **One vs One**

- Đối với hầu hết thuật toán, OvR tốt hơn.
- Một số thuật toán huấn luyện rất chậm khi kích thước tập dữ liệu huấn luyện lớn (như SVM) thì chiến lược OvO tốt hơn OvR.
- Scikit-Learn tự động lựa chọn chiến lược phù hợp cho từng thuật toán được sử dụng.
- Ta có thể điều chỉnh tham số để buộc Scikit-Learn chọn chiến lược nào (TH-101).
- Nếu dùng các thuật toán bản chất là phân lớp đa lớp như SGD thì không cần chọn chiến lược (TH-102).

Phân tích lỗi với Ma trận nhầm lẫn

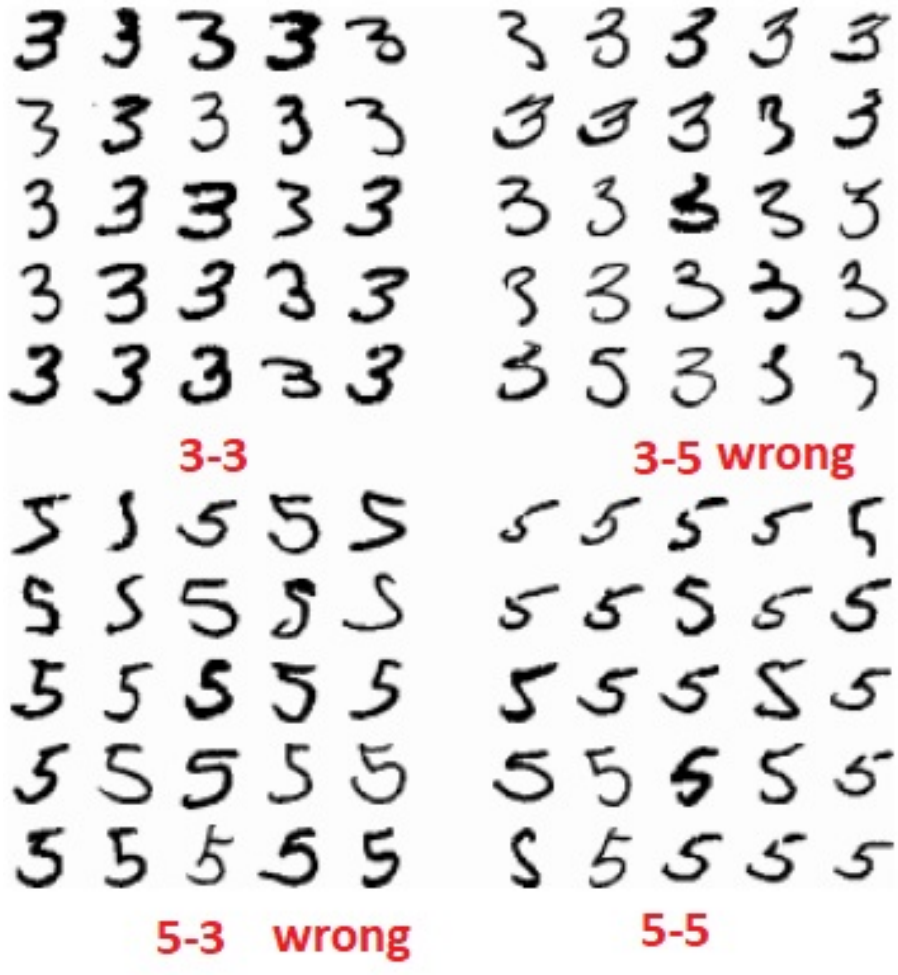
```
>>> y_train_pred = cross_val_predict(sgd_clf, X_train_scaled, y_train, cv=3)
>>> conf_mx = confusion_matrix(y_train, y_train_pred)
>>> conf_mx
array([[5578,    0,   22,    7,    8,   45,   35,    5,  222,    1],
       [   0, 6410,   35,   26,    4,   44,    4,    8,  198,   13],
       [  28,   27, 5232,  100,   74,   27,   68,   37,  354,   11],
       [  23,   18,  115, 5254,    2,  209,   26,   38,  373,   73],
       [  11,   14,   45,   12, 5219,   11,   33,   26,  299,  172],
       [  26,   16,   31,  173,   54, 4484,   76,   14,  482,   65],
       [  31,   17,   45,    2,   42,   98, 5556,    3,  123,    1],
       [  20,   10,   53,   27,   50,   13,    3, 5696,  173,  220],
       [  17,   64,   47,   91,    3,  125,   24,   11, 5421,   48],
       [  24,   18,   29,   67,  116,   39,    1,  174,  329, 5152]])
```

- Hình ảnh của ma trận nhầm lẫn (Đường chéo được fill bởi số 0 để dễ nhìn lỗi; Ô trắng là lỗi).



- Cột số 8 có màu sáng nghĩa là nhiều lớp bị gán nhầm thành 8.

Phân tích lỗi bằng cách xem xét từng lỗi sai



- Lý giải từng lỗi xem tại sao mô hình dự đoán sai, làm sao để hết sai.
- Chẳng hạn như nên quay ảnh cho bớt nghiêng.

Nội dung trình bày

4 Bài toán Phân lớp đa nhãn

Phân lớp đa nhãn

- Phân lớp đa nhãn (Multilabel classification)
- Bài toán ví dụ:
  - Đầu vào: Ảnh của 1 chữ số
  - Đầu ra: 2 nhãn nhị phân. Nhãn thứ nhất: chữ số đó có lớn hay không (lớn: 7,8,9); Nhãn thứ hai: chữ số đó là chữ số lẻ.
- Tạo tập dữ liệu

```
from sklearn.neighbors import KNeighborsClassifier

y_train_large = (y_train >= 7)
y_train_odd = (y_train % 2 == 1)
y_multilabel = np.c_[y_train_large, y_train_odd]

knn_clf = KNeighborsClassifier()
knn_clf.fit(X_train, y_multilabel)
```

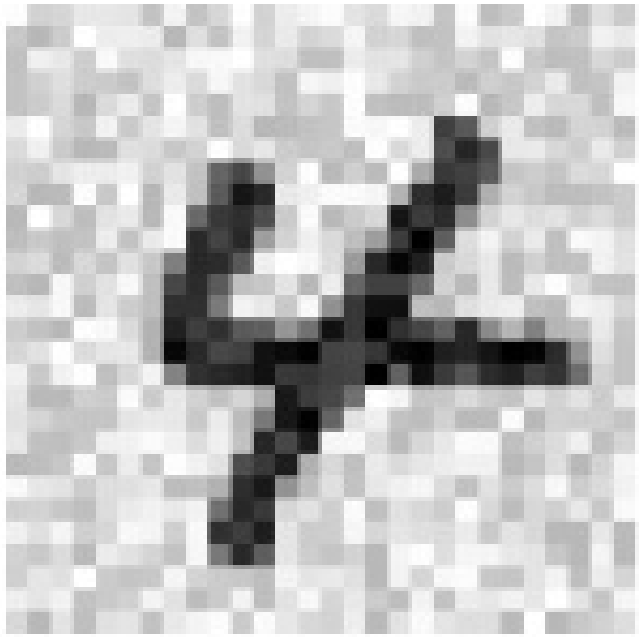
- Đánh giá: Có nhiều cách, đơn giản nhất là tính F1-score cho từng nhãn và lấy trung bình. Có thể coi các nhãn quan trọng như nhau hoặc đặt một số nhãn ưu tiên hơn.

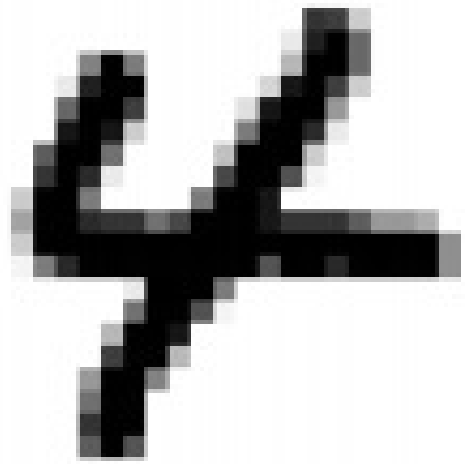
Nội dung trình bày

5 Bài toán Phân lớp đa đầu vào

Phân lớp đa đầu vào

- Phân lớp đa đầu vào (Multioutput multiclass classification/Multioutput classification)
- Là bài toán tổng quát của Phân lớp đa nhãn. Trong đó, mỗi nhãn có thể có nhiều hơn 2 giá trị.
- Bài toán ví dụ: Khử nhiễu ảnh.
  - Đầu vào: Ảnh của 1 chữ số có nhiễu
  - Đầu ra: Ảnh đã khử nhiễu. Giá trị của mỗi pixel được dự đoán lại (để bỏ nhiễu). Có 784 pixel, mỗi pixel nhận giá trị trong [0,255].





Nội dung trình bày

6 Thực hành

Thực hành chương 3

Viết chương trình:

- Tải bộ dữ liệu MNIST. (tr85).
  - Hiển thị ảnh của 1 chữ số (tr86).
  - Xem nhãn của chữ số đó.
- Huấn luyện bộ phân lớp nhị phân để nhận diện chữ số 5 dùng thuật toán SGD (tr88).
- Đánh giá bộ phân lớp
  - Tính accuracy bằng phương pháp thẩm định chéo. (tr89)
  - Tính ma trận nhầm lẫn (tr90, 91).
  - Tính precision, recall, F1 (tr92, 93)
  - Vẽ đường cong precision theo recall
  - Vẽ đường cong ROC (tr 97)
  - Tính AUC của bộ phân lớp (tr 98)
  - Huấn luyện thêm bộ phân lớp sử dụng thuật toán Random Forest. So sánh 2 bộ phân lớp SGD và Random Forest bằng cách vẽ 2 đường cong ROC trên cùng một biểu đồ, so sánh 2 AUC của chúng (tr 99, 100).

Thực hành chương 3 (tt)

Viết chương trình:

- Phân lớp đa lớp trên bộ dữ liệu MNIST.
  - Sử dụng SVM với hai chiến lược OvO và OvR (tr101).
  - Sử dụng thêm StandardScaler (tr102).
  - Đánh giá bộ phân lớp (tr102).
- Phân tích lỗi cho bộ phân lớp đa lớp.
  - Sử dụng ma trận nhầm lẫn (tr103)
  - Phân tích từng lỗi sai.
- Phân lớp đa lớp: tạo bộ phân lớp, đánh giá (tr106).
- Phân lớp đa nhãn: tạo bộ phân lớp cho bài toán khử nhiễu (tr108).

Tài liệu tham khảo

- Sách tham khảo “Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd Edition” của tác giả “Aurélien Géron”.