Classification

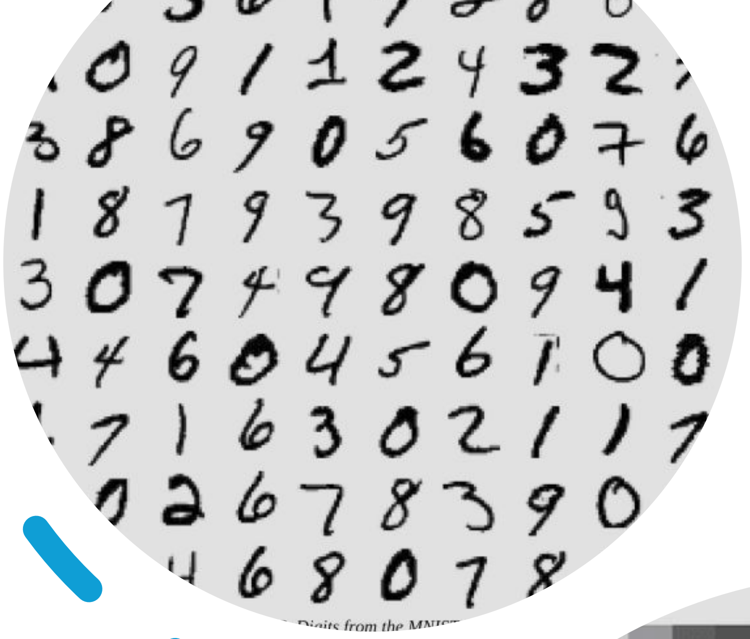
From Binary Classifiers to Performance Metrics

Introduction to Classification

- **Definition:** Predicting discrete class labels (e.g., spam/not spam).
- **Examples:**
  - Email filtering (binary classification).
  - Handwritten digit recognition (multiclass).
  - Medical diagnosis (probabilistic output).
- **Key Algorithms:**
  - Logistic Regression, SVM, Decision Trees, Random Forests.

MNIST Dataset Example

- **Dataset:** 70,000 handwritten digits (0–9).
- **Features:** 28×28-pixel grayscale images.
- **Goal:** Train a classifier to recognize digits.



Binary Classifier (5-detector)

- **Task:** Distinguish "5" vs. "not 5."
- **Example:** https://colab.research.google.com/drive/1thDxZkECSKTjVhQQsGgyjbsFkalp3NCk#scrollTo=tMhCVWUquT-8
- **Performance Measures:** Accuracy, confusion matrix, ROC curves.

Performance Metrics

- Accuracy using cross-validation
  - **Issue:** Useless for skewed datasets (e.g., 90% "not 5").
  - https://colab.research.google.com/drive/1thDxZkECSKTjVhQQsGgyjbsFkalp3NCk#scrollTo=tMhCVWUquT-8

Performance Metrics

- Confusion Matrix
  - It shows how well your classification model is performing — by comparing predicted labels vs actual labels.
  - True Positive (TP): Model correctly predicts the positive class.
  - True Negative (TN): Model correctly predicts the negative class.
  - False Positive (FP): Model incorrectly predicts positive when it's actually negative. (Type I Error)
  - False Negative (FN): Model incorrectly predicts negative when it's actually positive. (Type II Error)

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Performance Metrics

- Type I and Type II Errors
  - Type I = False Alarm (raising the alert when it's safe)
  - Type II = Missed Detection (not raising the alarm when there's danger)

Error Type	Also Known As	Meaning	Example
Type I Error	**False Positive**	Rejecting a true null hypothesis (detecting something that isn't there)	Healthy person diagnosed as sick
Type II Error	**False Negative**	Failing to reject a false null hypothesis (missing something real)	Sick person diagnosed as healthy

Performance Metrics

- Metrics from Confusion Matrix
  - **Accuracy:** Overall correctness
    - $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
  - **Precision (Positive Predictive Value):** How many predicted positives are correct
    - $Precision = \frac{TP}{TP+FP}$
  - **Recall (Sensitivity or True Positive Rate):** How many actual positives were identified correctly
    - $Recall = \frac{TP}{TP+FN}$
  - **Specificity (True Negative Rate):** How many actual negatives were identified correctly
    - $Specificity = \frac{TN}{TN+FP}$
  - **Fall-out (False Positive Rate)**
    - $FPR = \frac{FP}{TN+FP}$
  - **F1 Score:** Harmonic mean of Precision and Recall
    - Metric to compare two classifiers.
    - $F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$

Performance Metrics

- The Precision/Recall Trade-off

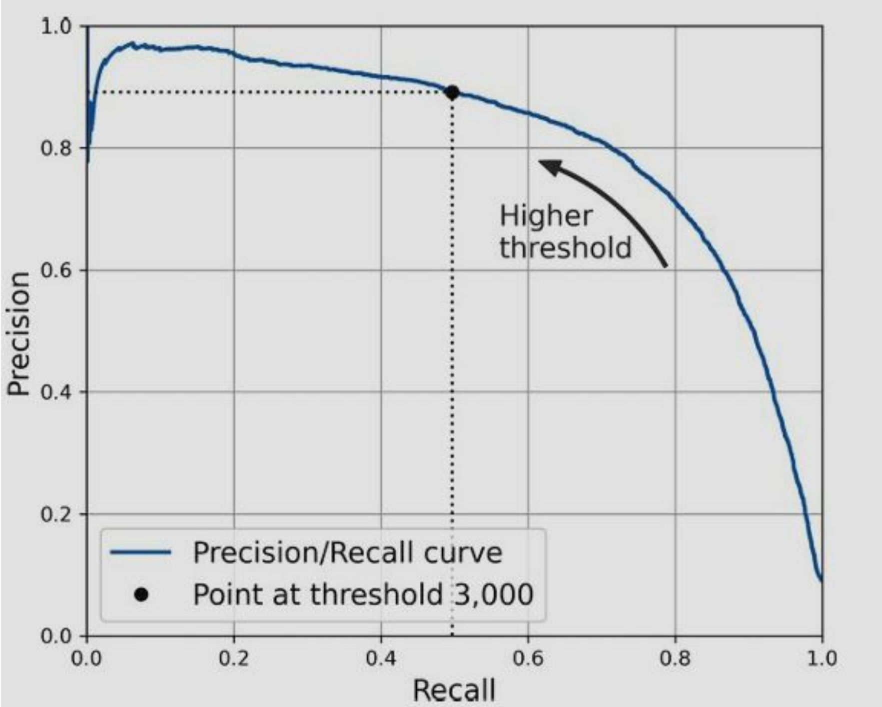
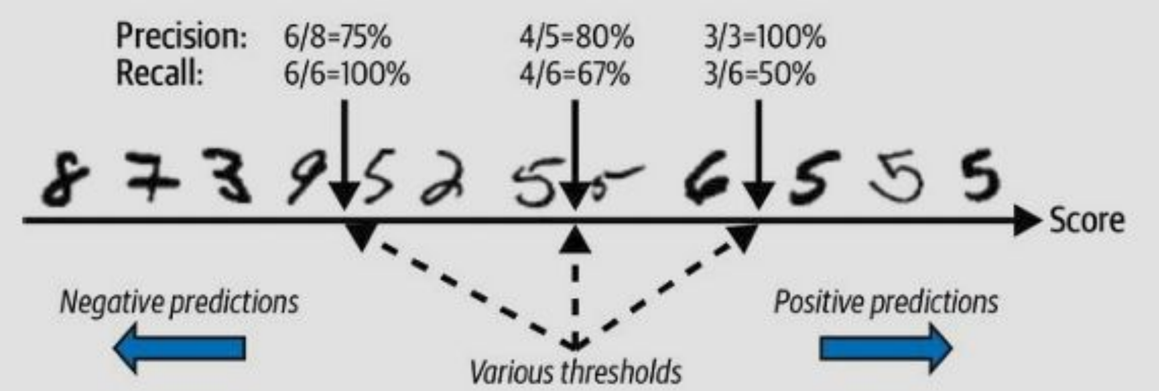


Figure 3-6. Precision versus recall



Performance Metrics

- The Receiver Operating Characteristic (ROC) Curve
  - **Plot:** TPR (Recall) vs. FPR (1 - Specificity).
  - Area Under the Curve (AUC) is used to compare classifiers.
  - A perfect classifier will have a ROC AUC equal to 1, whereas a purely random classifier will have a ROC AUC equal to 0.5.
- Rule of Thumb
  - Prefer the PR curve whenever the positive class is rare or when you care more about the false positives (Type I error) than the false negatives (Type II error).
  - The PR curve makes it clear that the classifier has room for improvement: the curve could really be closer to the top-right corner.

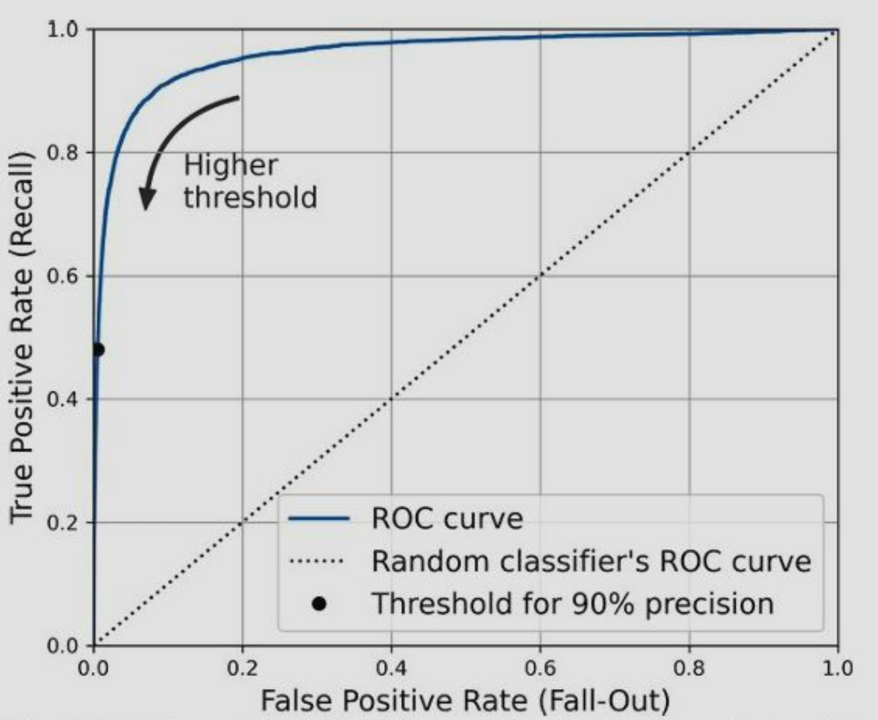


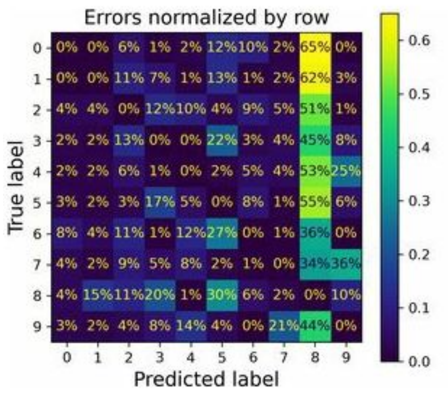
Figure 3-7. A ROC curve plotting the false positive rate against the true positive rate for all possible thresholds; the black circle highlights the chosen ratio (at 90% precision and 48% recall)

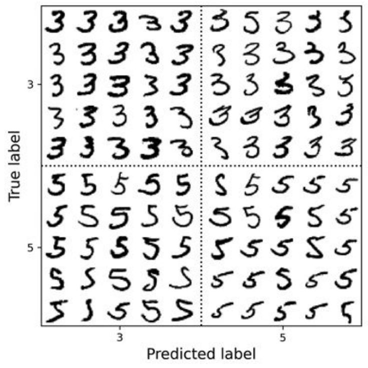
Multiclass or Multinomial Classification

- **OvR (One-vs-Rest) or OvA (One-vs-All):** Train 10 binary classifiers (one per digit).
  - When you want to classify an image, you get the decision score from each classifier for that image, and you select the class whose classifier outputs the highest score.
- **OvO (One-vs-One):** Train 45 classifiers (all pairs).
  - If there are N classes, you need to train $\frac{N \times (N-1)}{2}$ classifiers.
  - When you want to classify an image, you have to run the image through all 45 classifiers and see which class wins the most duels.
  - **Advantage:** Each classifier only needs to be trained on the part of the training set containing the two classes that it must distinguish.

Error Analysis

- **Confusion Matrix:** Identify misclassified digits (e.g., 3s vs. 5s).
- **Improvement:**
  - Preprocess images (e.g., center digits, augmenting the training set with slightly shifted and rotated variants of the training images).
  - Try advanced models (e.g., Random Forests, CNNs).





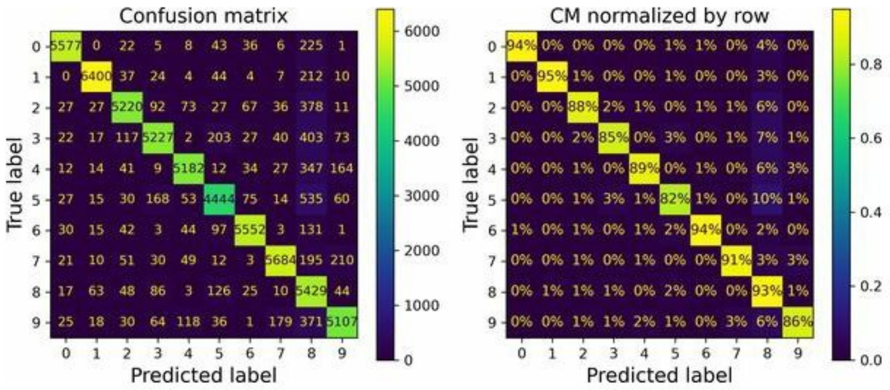
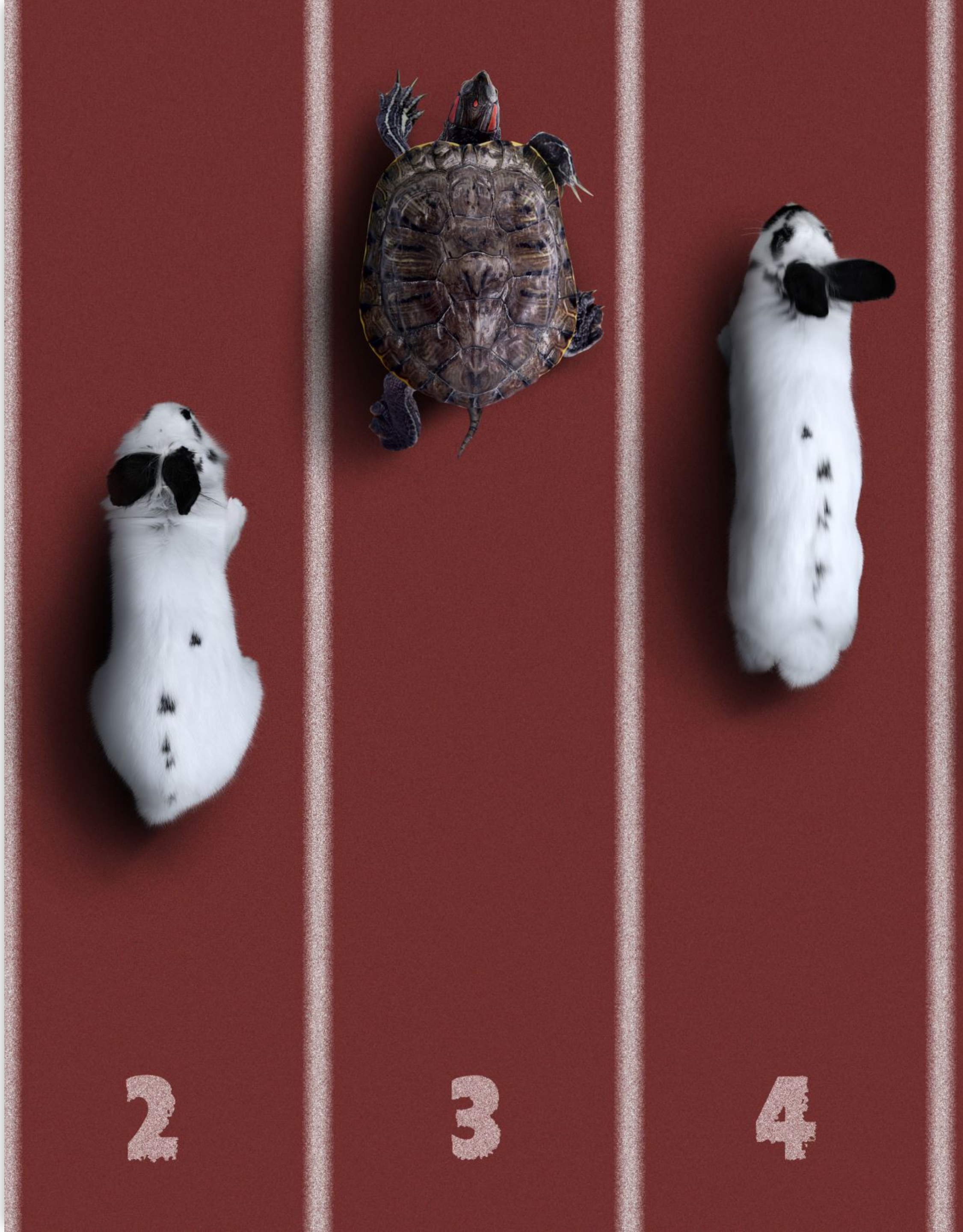


Figure 3-9. Confusion matrix (left) and the same CM normalized by row (right)

Multilabel Classification

-

- **Definition:** Assign multiple labels per instance (e.g., "cat" + "dog").
- **Example:** Creates a $y\_multilabel$ array containing two target labels for each digit image: the first indicates whether or not the digit is large (7, 8, or 9), and the second indicates whether or not it is odd.



Multioutput Classification

- **Generalization of multilabel:** Each label can be multiclass.
- **Example:** Image denoising (pixel-level classification).
- The classifier's output is multilabel (one label per pixel) and each label can have multiple values (pixel intensity ranges from 0 to 255).

Figure 3-12. A noisy image (left) and the target clean image (right)

Summary of Classification Types

Type	Output Shape	Label Type	Example
Binary	Single 0/1	Binary	Spam detection
Multiclass	Single class (0–9)	Multiclass	Digit recognition
Multilabel	Multiple 0/1s	Binary per label	Image tags (cat + sunny)
Multioutput	Multiple classes	Multiclass per label	Denoising (pixel intensities)