Machine Learning for Economists

Class 8: Classification and Risk Quant

葛雷

中国人民大学经济院

2025 年 4 月 10 日

RENMIN UNIVERSITY OF CHINA

What is Classification?

- Classification is the task of predicting a label (class).
- Binary classification: two classes (e.g., spam vs. ham).
- Multiclass classification: more than two classes.
- Multilabel and multioutput classifications

Applications of Classification Models

Finance:

- Credit scoring: Predict if a customer will default on a loan.
- Fraud detection: Classify whether a transaction is fraudulent.

Healthcare:

- Disease diagnosis: Predict presence of conditions (e.g., cancer, diabetes).
- Risk stratification: Identify high-risk patients.

Marketing:

- Customer segmentation.
- Predict likelihood to respond to a campaign.

LLM is also a Classification Model

- Why ?
- Word (token) prediction from the huge dictionary

Related Job: Risk Quants

- Credit Risk Quant: Models borrower defaults
- Market Risk Quant: Focuses on pricing derivatives and hedging strategies risk
- Operational Risk Quant: Analyzes tail-risk events (e.g., fraud, system failures)
- Model Risk Quant: model validation to prevent risks from modeling

Why Linear Regression not for classification

- Why Linear Regression not for classification?
- Let me give you a example

Sigmoid function for the non-linearity

Equation 4-14. Logistic function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

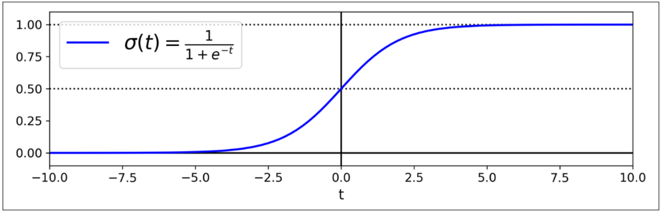

Figure 4-21. Logistic function

What is Logistic Regression?

- A statistical model for binary classification
- Predicts the probability of an event occurring
- Output ranges between 0 and 1
- Uses the logistic (sigmoid) function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression

- Traditional Logistic Regression:

$$\hat{y} = \sigma(x\theta^T) = \frac{1}{1 + e^{-x\theta^T}}$$

- Other machine learning models:

$$\hat{y} = \sigma(h_\theta(x)) = \frac{1}{1 + e^{-h_\theta(x)}}$$

- $h_\theta(x)$ is also called score function, it can be ANN, Xgboost, random forest ...

Cross-entropy Loss Function (Cost function)

- Binary cross-entropy loss (log loss):

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\hat{Y}^{(i)}) + (1-y^{(i)})\log(1-\hat{Y}^{(i)})]$$

- Where:
  - m = number of training examples
  - $y^{(i)}$ = true label (0 or 1)
  - $\hat{Y}^{(i)}$= model predicted the probability

Cross-entropy Loss?

- Why we use Cross-entropy Loss?
- Bernoulli Distribution + MLE
- Pencil and paper time!!!

Gradient Descent

- Iterative optimization algorithm
- Update rule:

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial L(\theta)}{\partial \theta_t}$$

- Partial derivative:

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial L(\theta)}{\partial \hat{Y}^{(i)}} \frac{\partial h_\theta(x^{(i)})}{\partial \theta}$$

Example: Training a Binary Classifier

- Example: Classify whether a digit is 5 or not.
- Model: `SGDClassifier` from Scikit-Learn.
- Uses stochastic gradient descent.

Measuring Performance

- Accuracy is not reliable for imbalanced datasets. (Why? Let me give a example)
- Confusion matrix is more informative.
- Precision and recall are better metrics.

Confusion Matrix

Predicted Class

Actual Class		**Positive**	**Negative**	
	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Confusion Matrix on textbook (labels different)

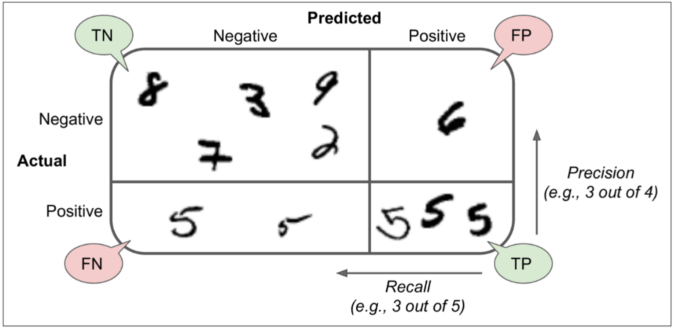

Figure 3-2. An illustrated confusion matrix shows examples of true negatives (top left), false positives (top right), false negatives (lower left), and true positives (lower right)

Precision/Recall Trade-off

- You can adjust the decision threshold.
- Higher precision => lower recall and vice versa.

Precision/Recall Trade-off

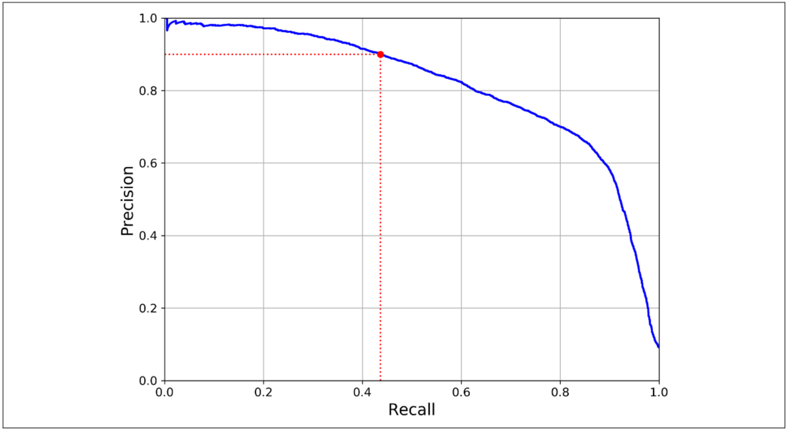

Figure 3-5. Precision versus recall

F1 Score

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

- Precision $= \frac{TP}{TP+FP}$
- Recall $= \frac{TP}{TP+FN}$
- F1 Score $= F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$ s.t. Harmonic mean of precision and recall

ROC Curve

- ROC = Receiver Operating Characteristic curve.
- Plots True Positive Rate (Recall) against False Positive Rate.
- TPR $= \frac{TP}{TP+FN}$
- FPR $= \frac{FP}{FP+TN}$
- Helps evaluate classifier performance at all thresholds.

ROC Curve

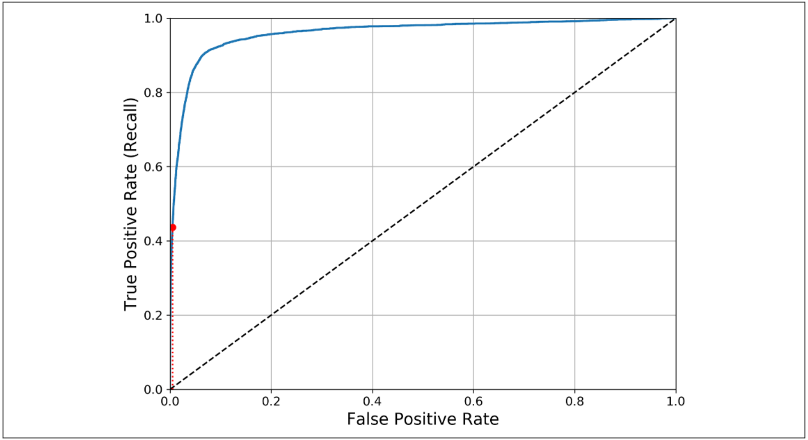

Figure 3-6. This ROC curve plots the false positive rate against the true positive rate for all possible thresholds; the red circle highlights the chosen ratio (at 43.68% recall)

ROC Curve: Compare models

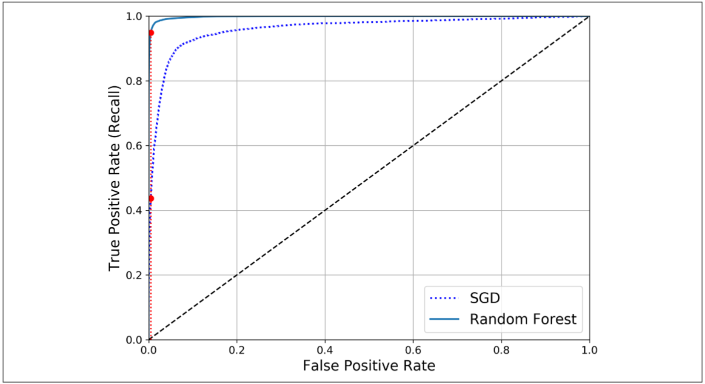

Figure 3-7. Comparing ROC curves: the Random Forest classifier is superior to the SGD classifier because its ROC curve is much closer to the top-left corner, and it has a greater AUC

Understanding AUC (Area Under Curve)

- AUC is the area under the ROC curve.
- AUC $= 1$: perfect classifier.
- AUC $= 0.5$: random guessing.
- The higher the AUC, the better the model distinguishes between classes.
- Useful when comparing multiple classifiers.

Introduction to Classification

Logistic Regression

Performance Measures

Multiclass Classification

Multiclass Classification: Softmax Function

Sigmoid vs Softmax

From Sigmoid $\rightarrow$ Softmax

Softmax Function

For a K-class classification problem:

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{where } k = 1, \ldots, K \tag{1}$$

Cross-Entropy Loss (Softmax Loss)

Given the softmax output $\hat{y}$ and the true label y , the loss for one observation is:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k \log(\hat{y}_k)$$

- **K is the number of classes**
- Penalizes incorrect predictions more when the predicted probability is low
- Encourages the correct class to have a high predicted probability

Reference

1. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow (3rd edition)
2. https://encord.com/glossary/confusion-matrix/
3. Kaggle
4. Wikipedia
5. ChatGPT
6. DeepSeek