DDA3020 Tutorial 10
Performance Evaluation

Rongxiao Qu
School of Data Science
Email: rongxiaoqu@link.cuhk.edu.cn
Office hour: Tue 10:30 - 11:30, by appointment

Date: 2022.11.22

Contents

- Evaluation Metrics
  - Regression
  - Classification
    - Confusion Matrix
    - ROC Curve
- Cross Validation
  - LOOCV
  - K-fold CV
- Exercise

Evaluation Metrics

- Measures the quality of the statistical or machine learning model

(a good evaluation metric may not necessarily be a loss function)

- Regression:

  Mean Squared Error: $\frac{1}{n}\sum_{i=1}^{n}(y_i \ - \widehat{y_i})^2$

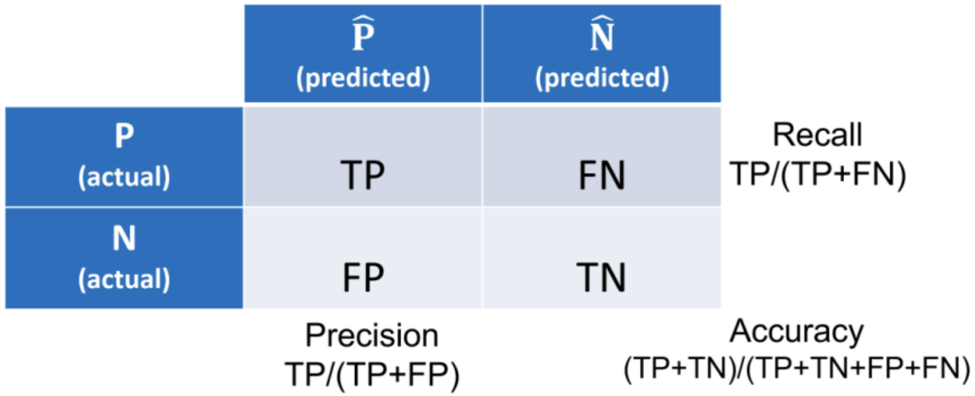
  Mean Absolute Error: $\frac{1}{n}\sum_{i=1}^{n}|y_i \ - \hat{y}_i|$

Evaluation Metrics

- Classification:

Confusion Matrix:

2 classes: Positive and Negative class



	$\hat{P}$ (predicted)	$\hat{N}$ (predicted)	
P (actual)	TP	FN	Recall TP/(TP+FN)
N (actual)	FP	TN	
	Precision TP/(TP+FP)		Accuracy (TP+TN)/(TP+TN+FP+FN)

- Accuracy = $\dfrac{TP+TN}{TP + \mathrm{TN} + FP + FN}$

- Precision = $\dfrac{TP}{TP+FP}$

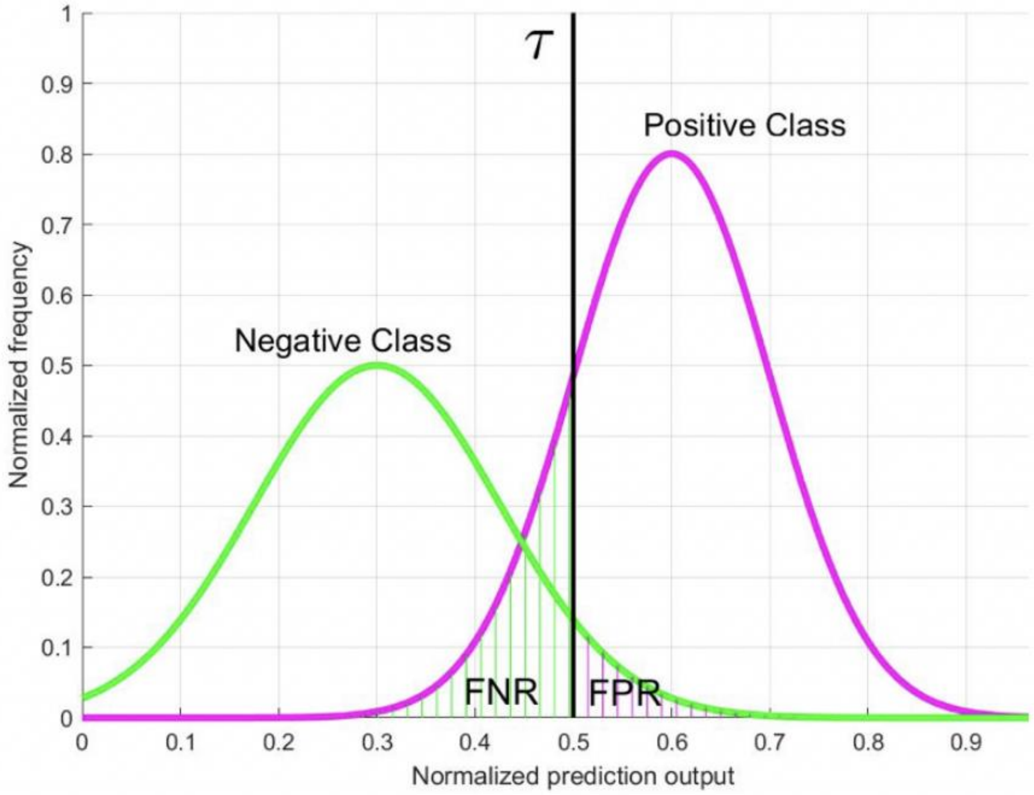
(also called Positive Predictive Value)

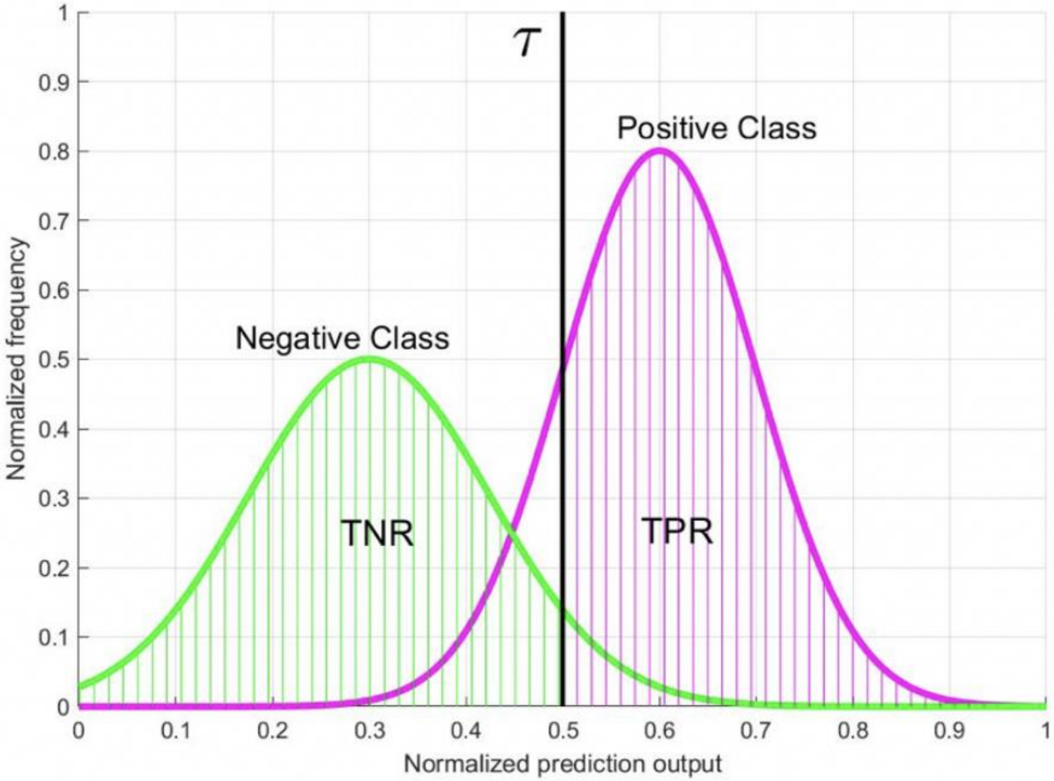
- Recall = $\dfrac{TP}{TP+FN} = \mathrm{TPR}$

(also called sensitivity)

TPR = TP/(TP+FN)
FNR = FN/(TP+FN) (Type Ⅱ error)
TNR = TN/(FP+TN)
FPR = FP/(FP+TN) (Type Ⅰ error)

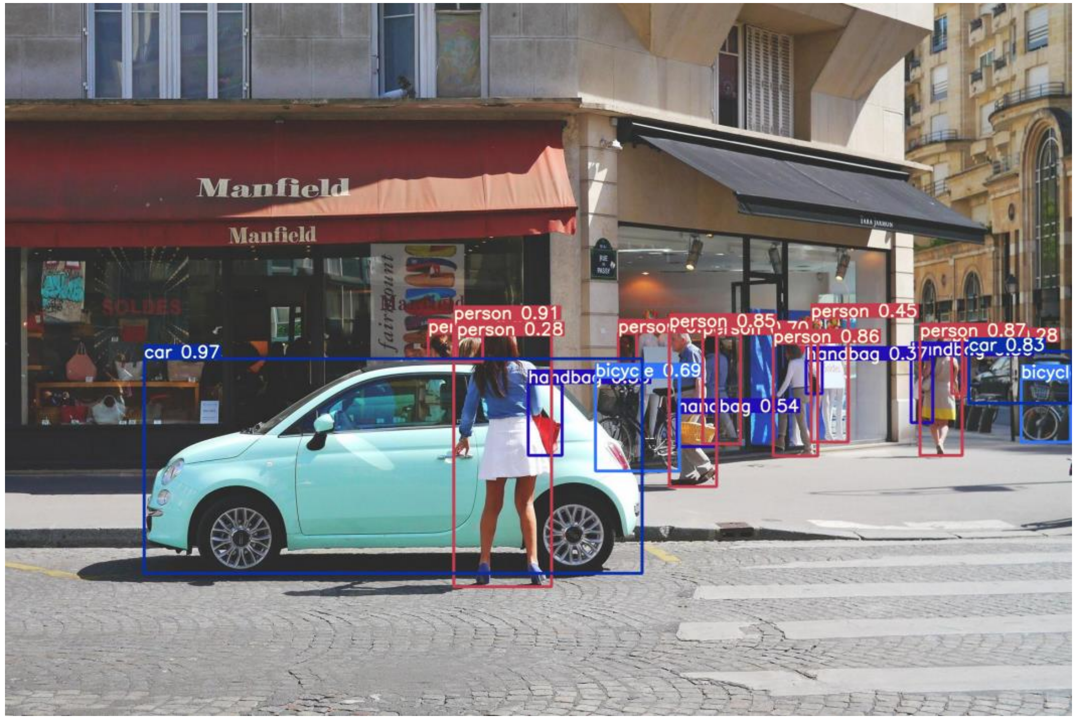
Evaluation Metrics

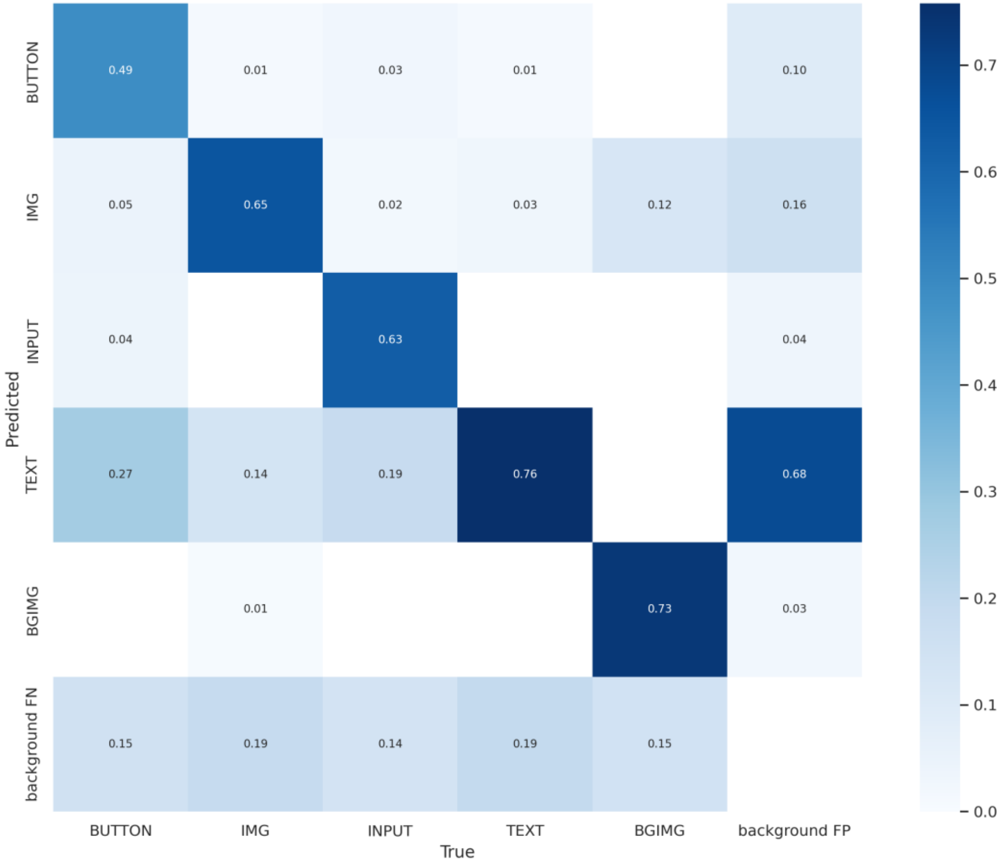




Evaluation Metrics

- In reality, there may be more than two classes to classify.
- For example, in object detection:

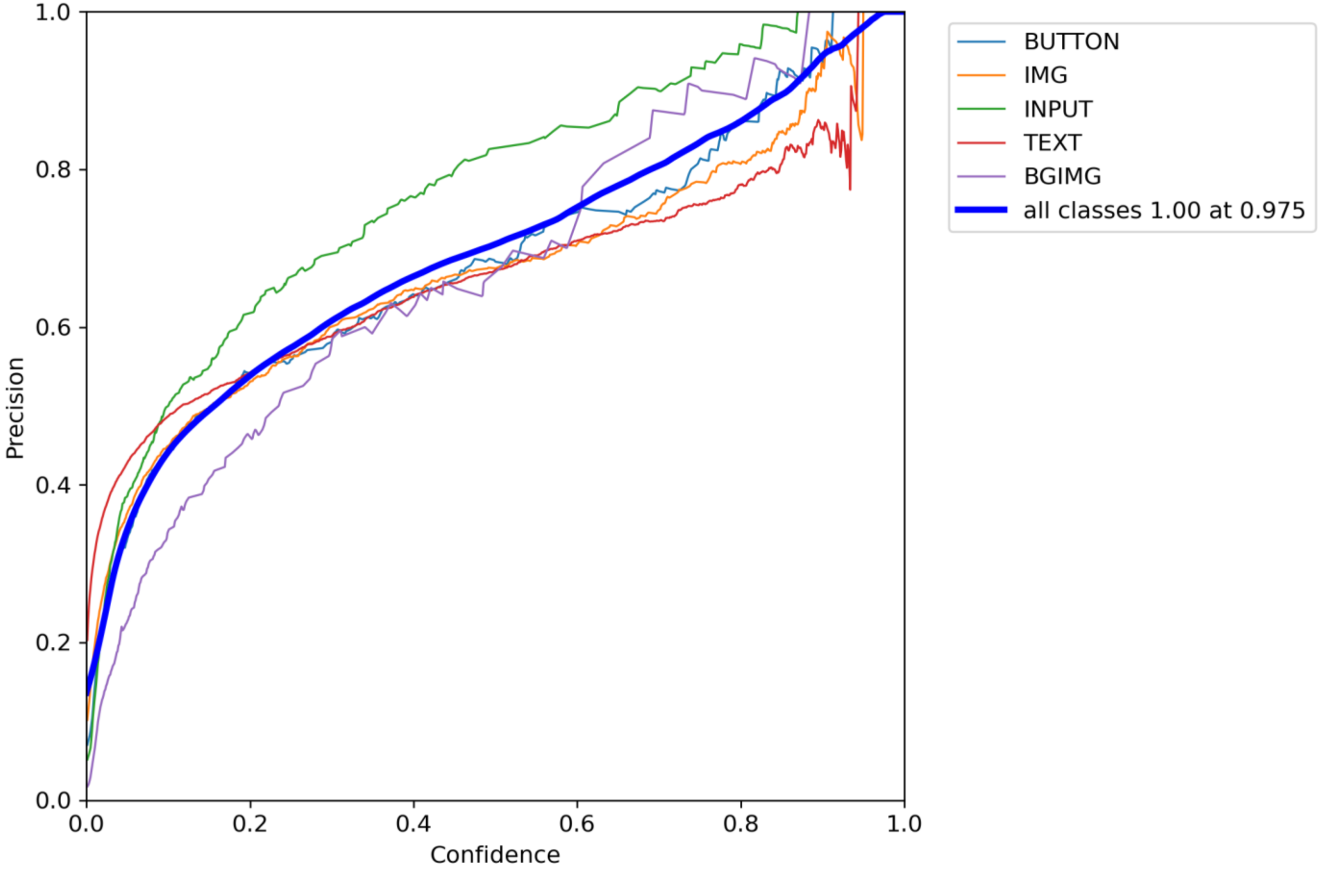




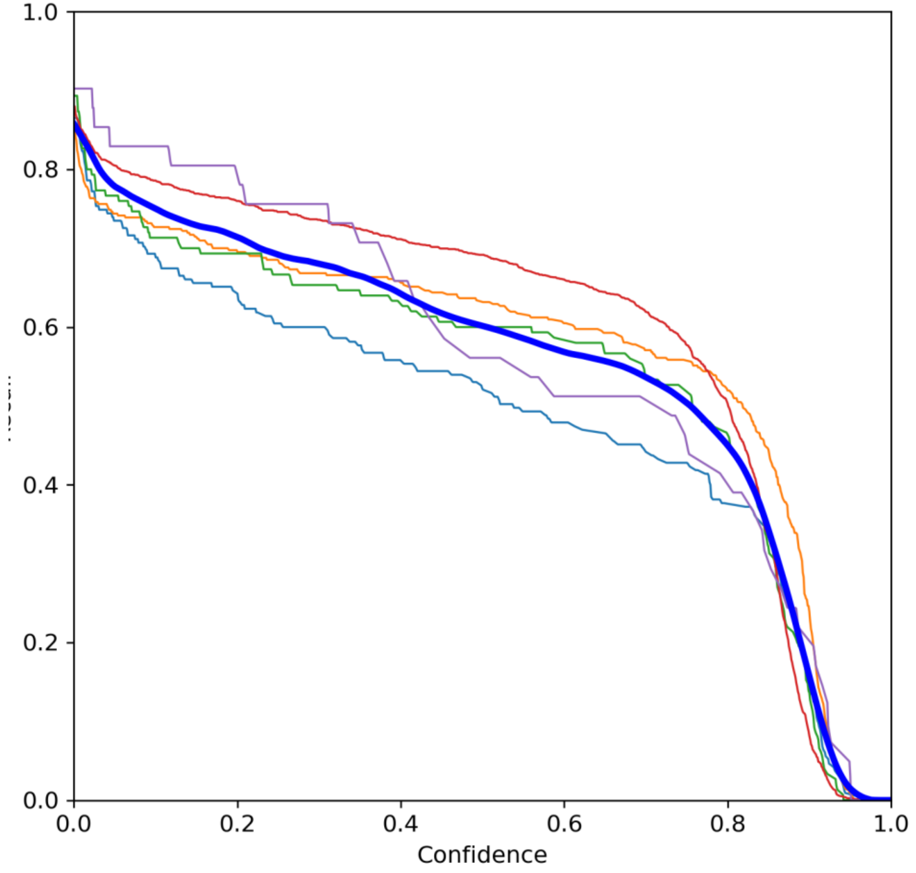
Predicted \ True	BUTTON	IMG	INPUT	TEXT	BGIMG	background FP
BUTTON	0.49	0.01	0.03	0.01		0.10
IMG	0.05	0.65	0.02	0.03	0.12	0.16
INPUT	0.04		0.63			0.04
TEXT	0.27	0.14	0.19	0.76		0.68
BGIMG		0.01			0.73	0.03
background FN	0.15	0.19	0.14	0.19	0.15	

Evaluation Metrics

P-curve

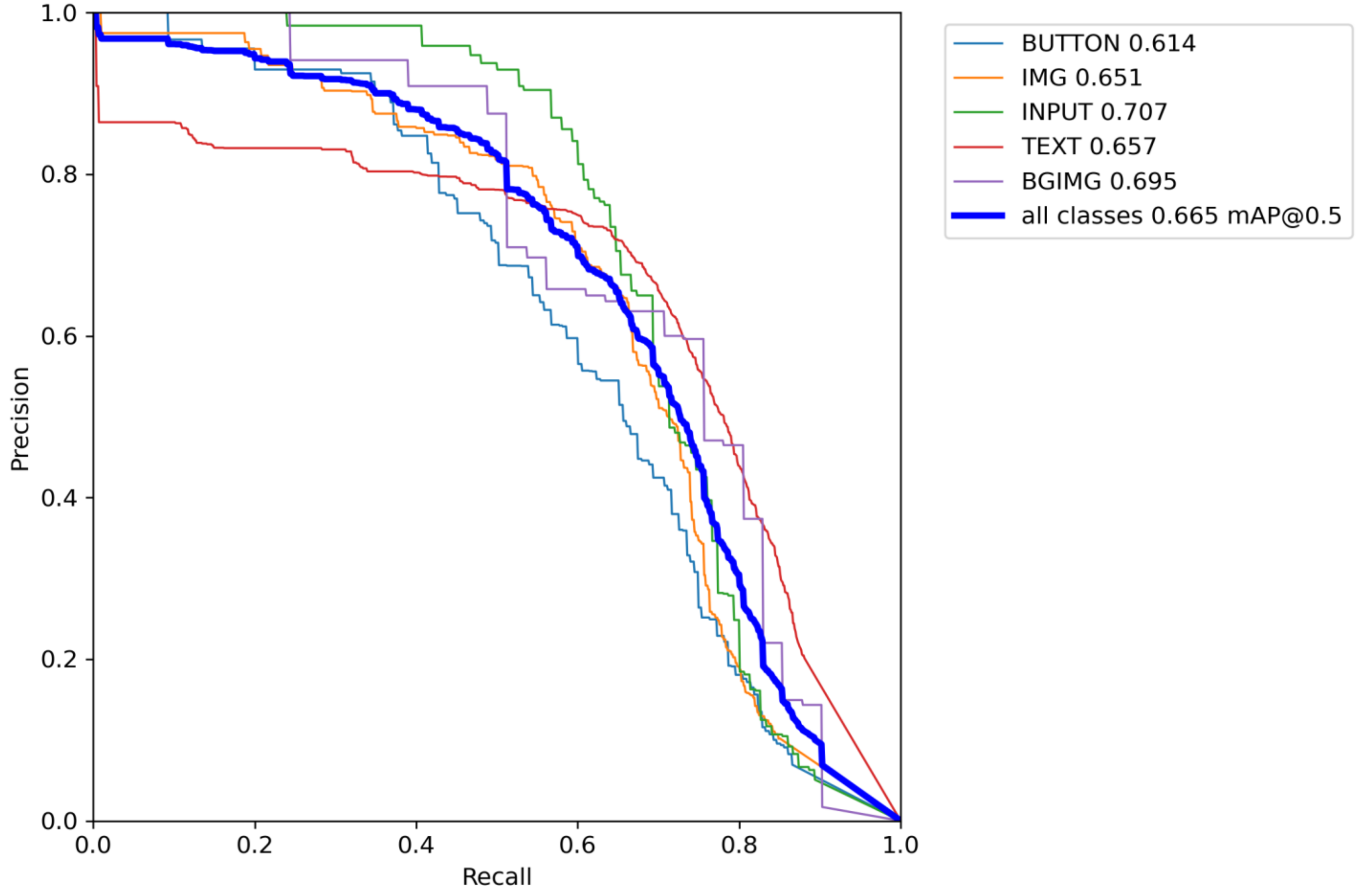


R-curve

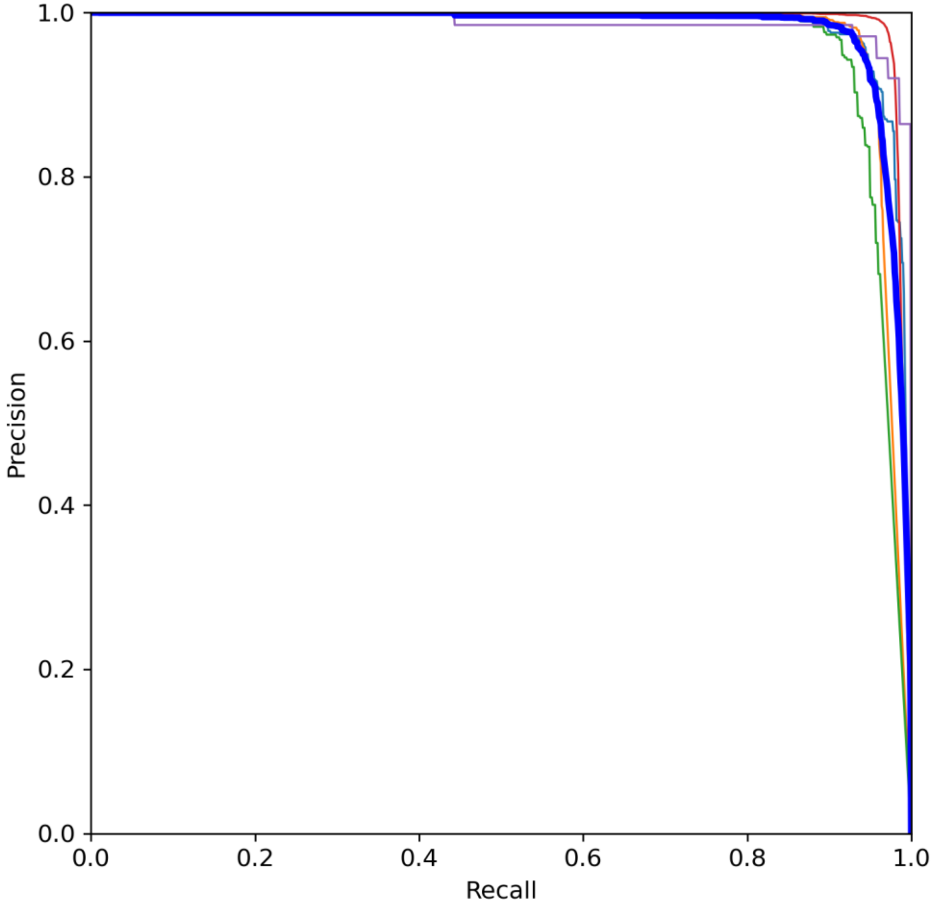


Evaluation Metrics

PR-curve

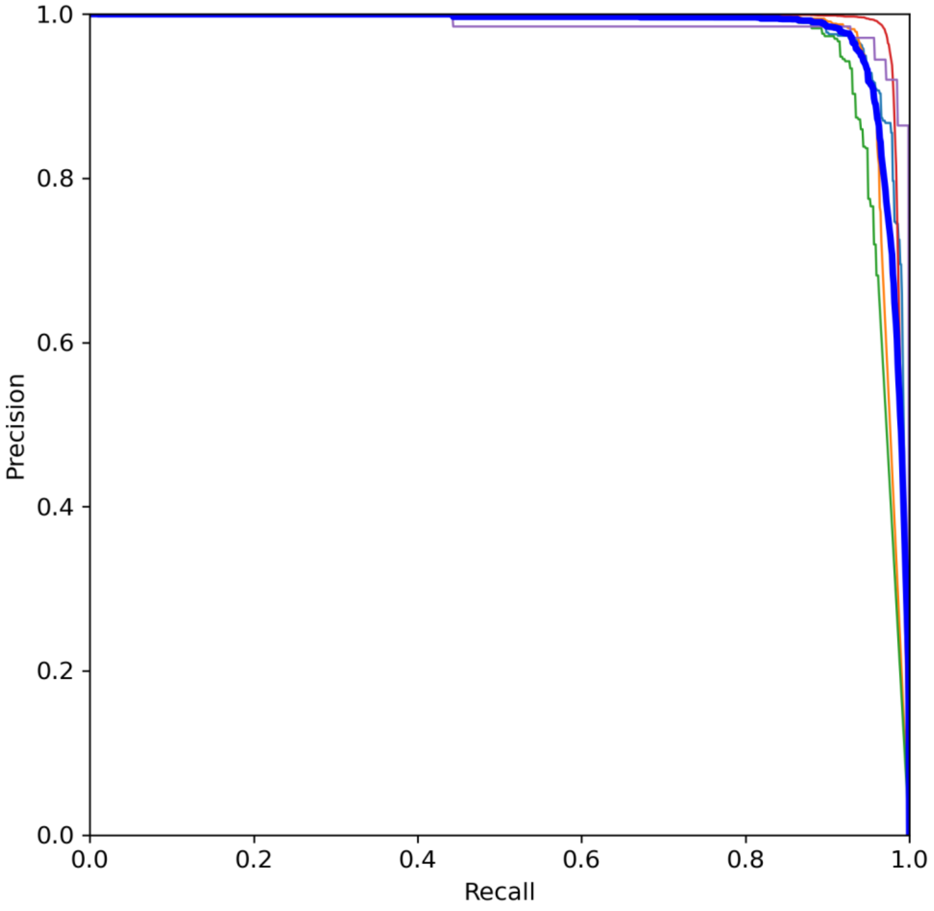


PR-curve



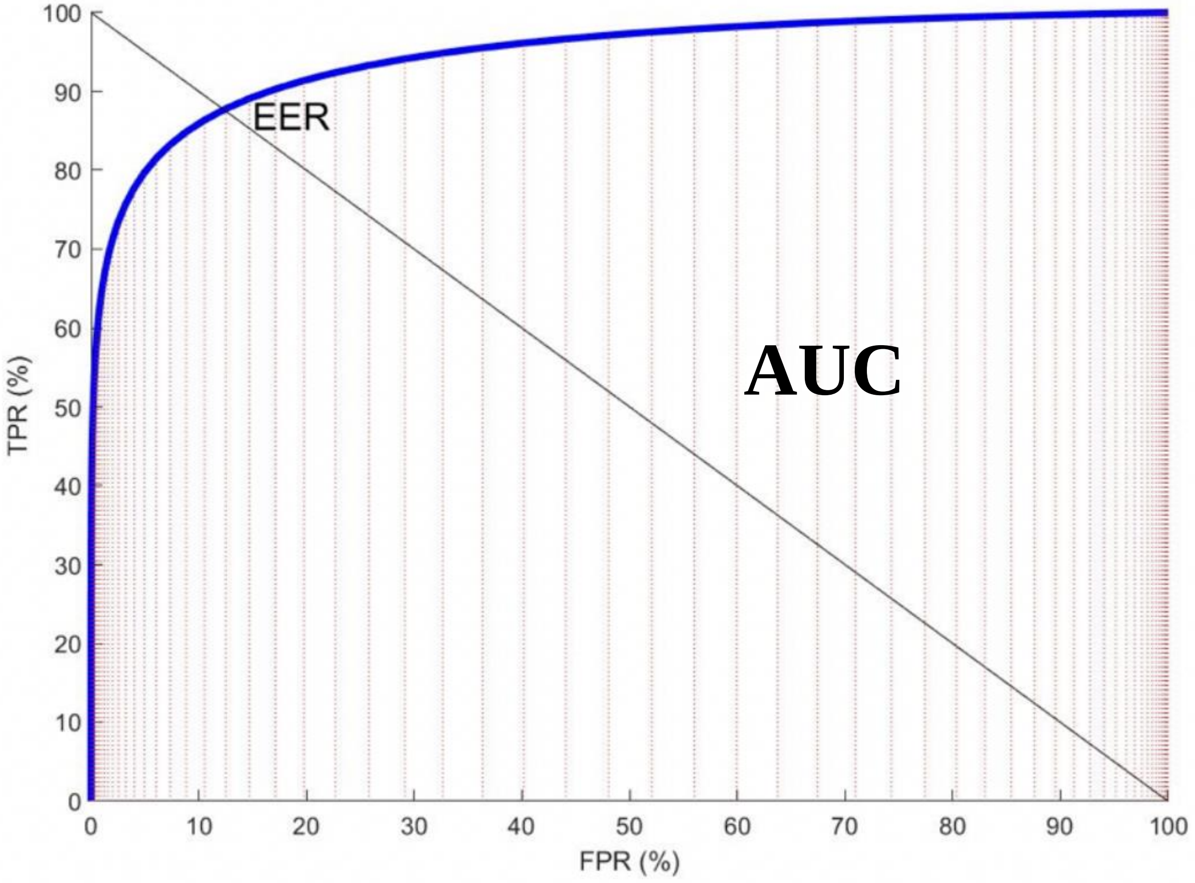
Evaluation Metrics

PR-curve



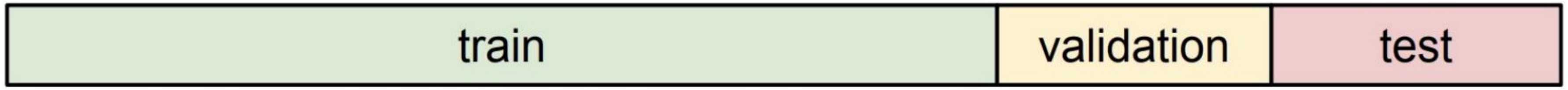
ROC-curve

(Receiver Operating Characteristic Curve)



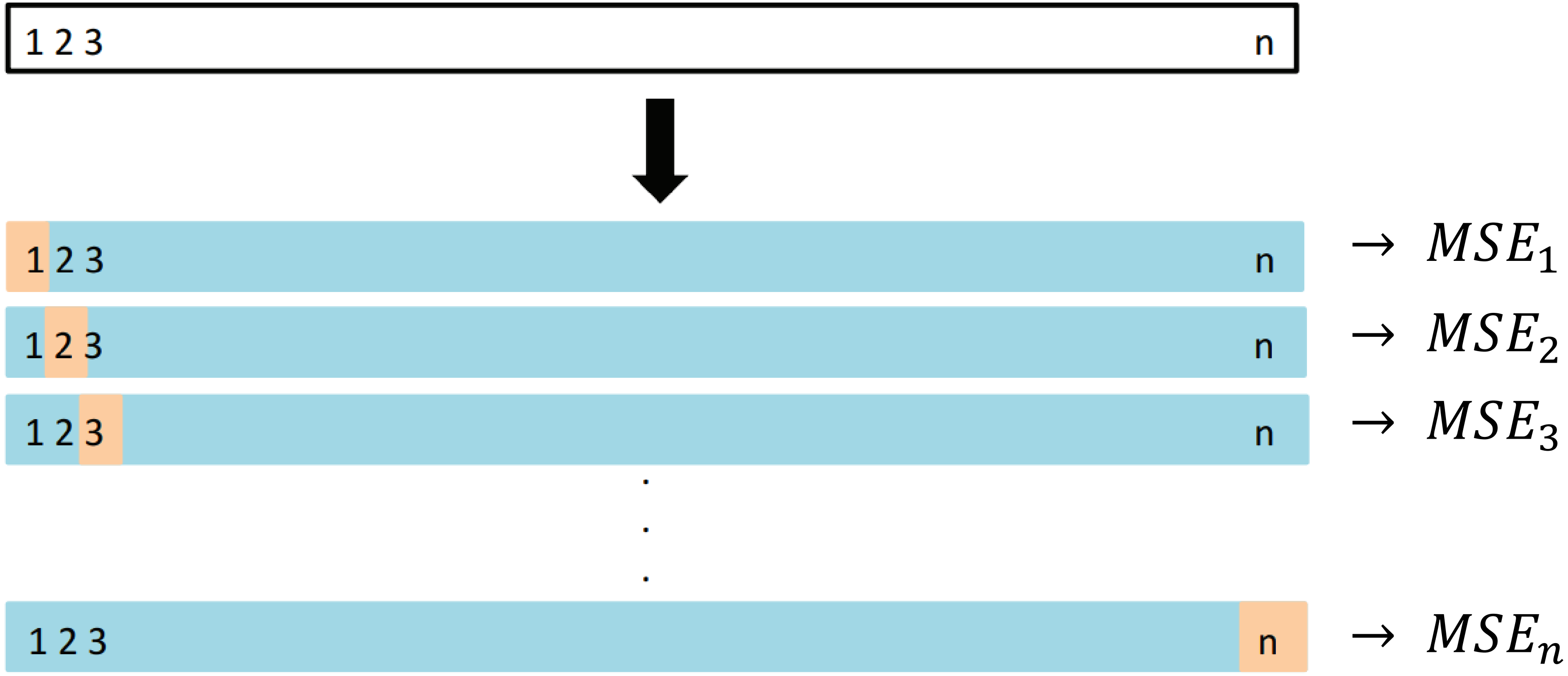
Cross Validation

- Validation: estimate the performance (or accuracy) of the model, protect against overfitting in a predictive model.

- Validation set: 

- When data is limited, we want to reuse the training data
- → Cross validation

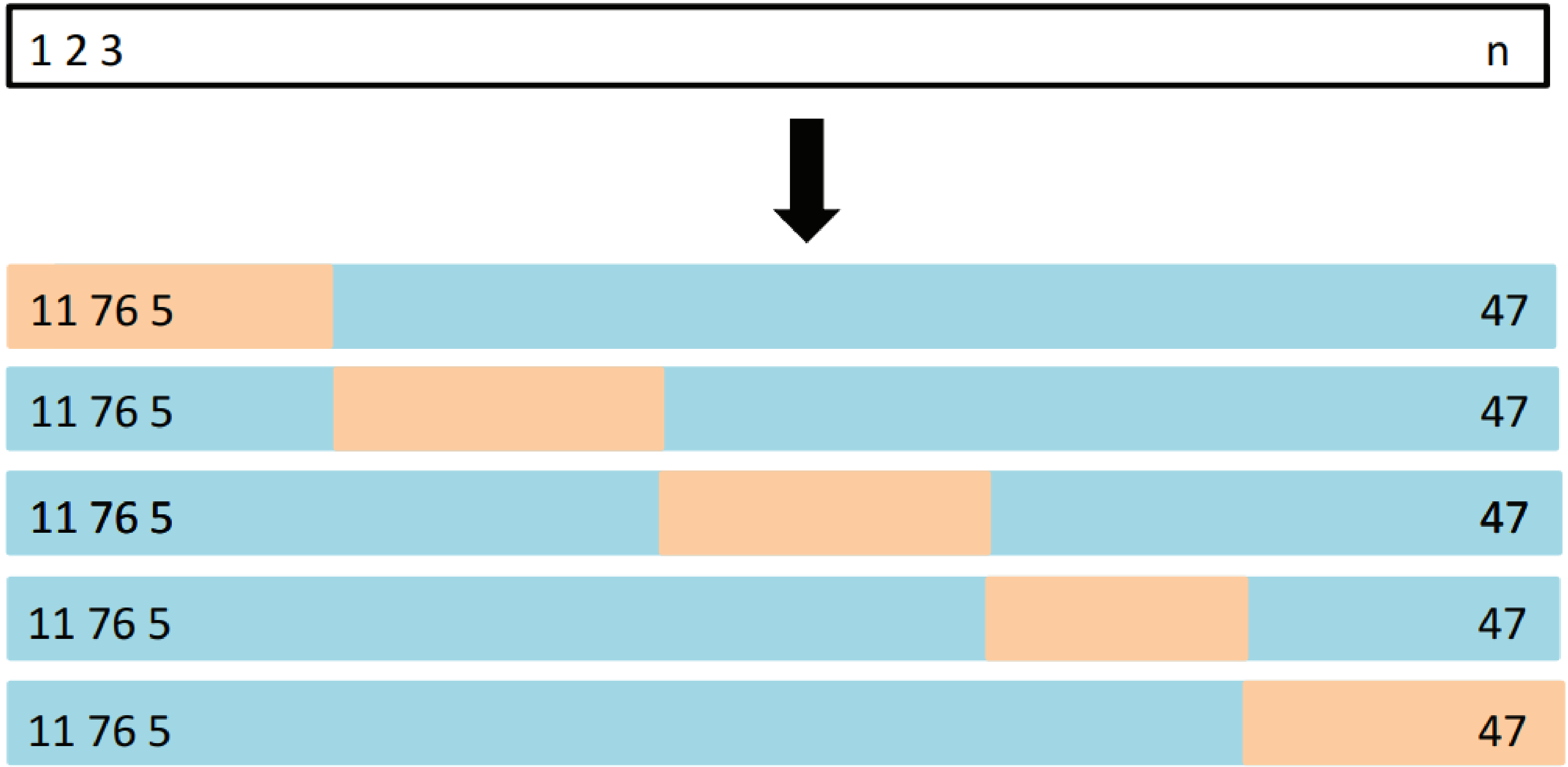
Leave One Out Cross Validation (LOOCV)



$$CV = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

Here, the MSE can be any metrics that you want to measure.

k-fold Cross Validation



$\rightarrow MSE_1$

$\rightarrow MSE_2$

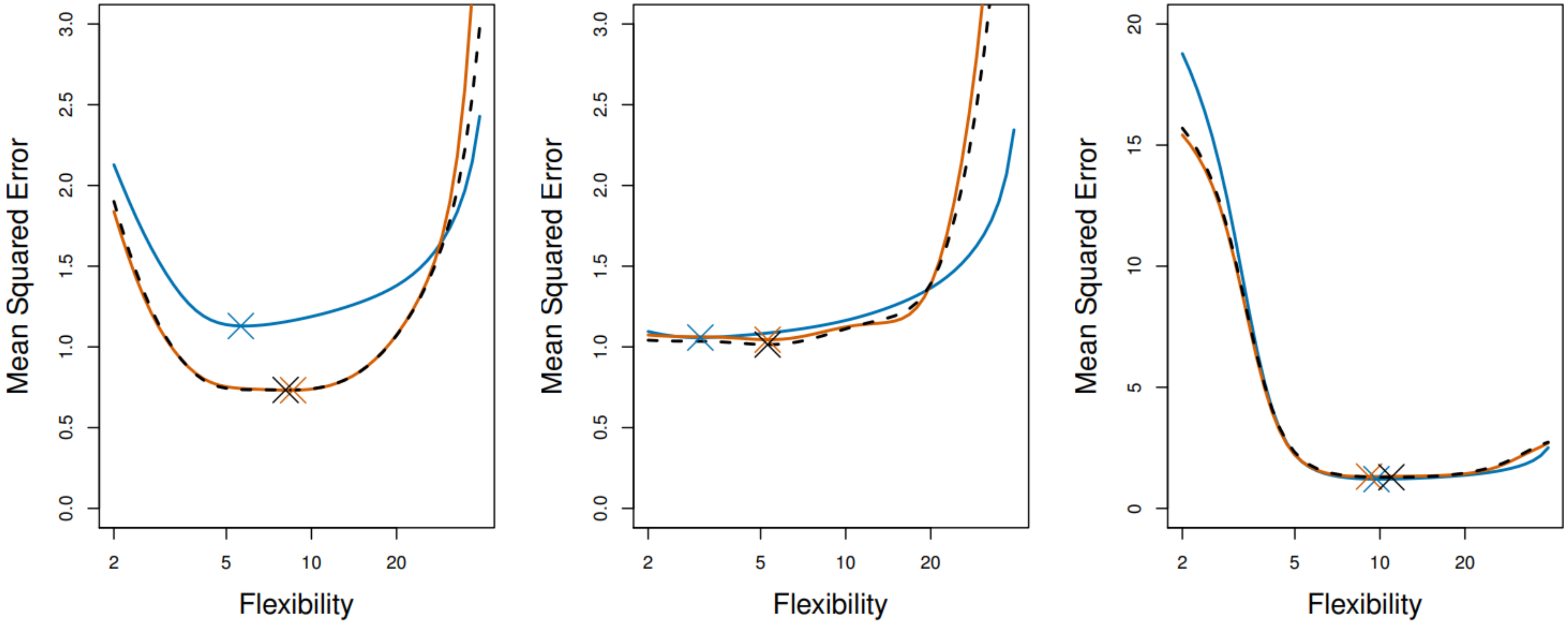
$\rightarrow MSE_3$

$\rightarrow MSE_4$

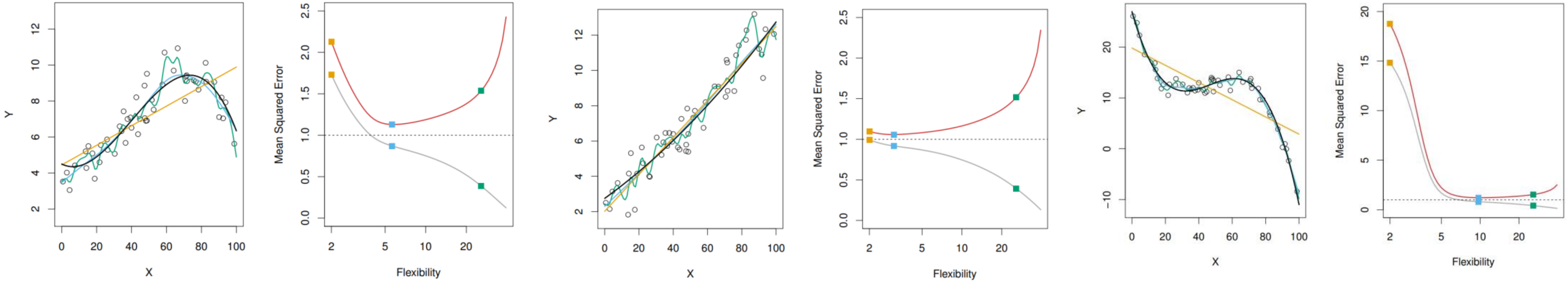
$\rightarrow MSE_5$

$$CV = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

Cross Validation: Comparisons



Blue: Test Error
Black: LOOCV
Orange: 10-fold



Left: Black: true function
Right: Red: test MSE
Grey: training MSE

Exercise 1:

Behavior of training set error with increasing sample size

The error on the test set will always decrease as we get more training data, since the model will be better estimated. However, as shown in Figure 1, for sufficiently complex models, the error on the training set can increase as we get more training data, until we reach some plateau. Explain why.

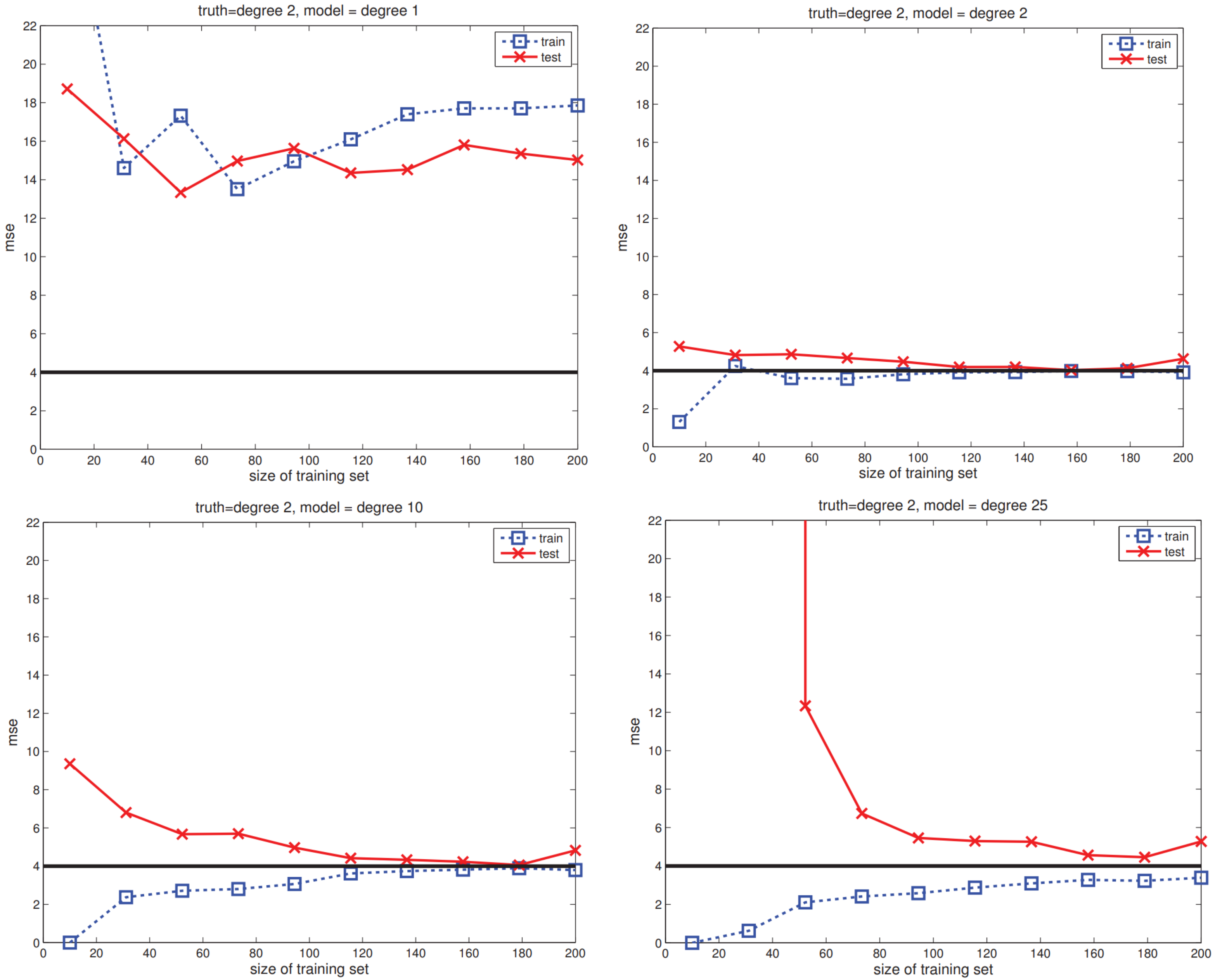

truth=degree 2, model = degree 1
truth=degree 2, model = degree 2
truth=degree 2, model = degree 10
truth=degree 2, model = degree 25
train
test
mse
size of training set
0
2
4
6
8
10
12
14
16
18
20
22
0
20
40
60
80
100
120
140
160
180
200

Figure 1

Exercise 1:

Behavior of training set error with increasing sample size

The statement “**the error on the test will always decrease as we get more training data since the model will be better estimated”** is not a gold standard. The prerequisites behind this statement are at least:

- The test data shares an identical distribution with the training data.
- The model is complex enough to learn the knowledge embedded in the data.

When the training set is small, the trained model is usually over-fitted to the current data set (since the model is very complex), so the accuracy can be relatively high. As the training set increases, the model has to learn to adapt to more general-purpose parameters, thus reducing the overfitting effect laterally, resulting in lower accuracy.

Exercise 2:
Optimal threshold on classification probability

Consider a case where we have learned a conditional probability distribution $P(y|x)$. Suppose there are only two classes, and let $p_0 = P(Y = 0|x)$ and $p_1 = P(Y = 1|x)$. Consider the loss matrix below:

Predicted label $\hat{y}$	True label y	
	0	1
0	0	λ_{01}
1	λ_{10}	0

a. Show that the decision $\hat{y}$ that minimizes the expected loss is equivalent to setting a probability threshold θ and predicting $\hat{y} = 0$ if $p_1 < \theta$ and $\hat{y} = 1$ if $p_1 \geq \theta$. What is θ as a function of λ_{01} and λ_{10}? (Show your work.)

b. Show a loss matrix where the threshold is 0.1. (Show your work.)

Exercise 2:
Optimal threshold on classification probability

a. The posterior loss expectation for choosing action $\hat{y}$, given x, is:

$$\rho(\hat{y}|x) = E_{p(y|x)}[L(\hat{y}, y)]$$

$$= p_o * L(\hat{y}, 0) + p_1 * L(\hat{y}, 1)$$

$$= p_o * L(\hat{y}, 0) + (1 - p_0) * L(\hat{y}, 1)$$

$$= L(\hat{y}, 1) + p_0(L(\hat{y}, 0) \ - L(\hat{y}, 1))$$

Thus:

$$\rho(0|x) = \lambda_{01} \ - p_0 * \lambda_{01}$$

$$\rho(1|x) = p_0 * \lambda_{10}$$

Both are linear functions of p_0, hence, the unique optimal threshold is where $\rho(0|x) = \rho(1|x)$, i.e.,

$$p_0 = \frac{\lambda_{01}}{\lambda_{01}+\lambda_{10}}, \qquad p_1 = 1 - p_0 = \frac{\lambda_{10}}{\lambda_{01}+\lambda_{10}}$$

b. Let $\lambda_{10} = 1, \lambda_{01} = 9$ will do.

Thanks!

2022.11.22