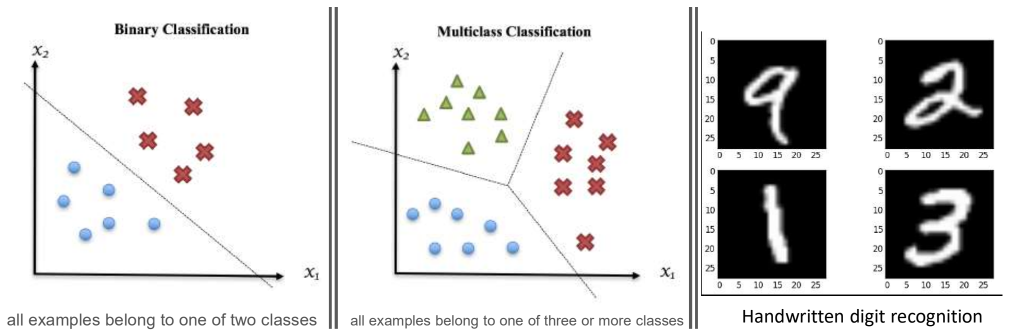
Chapter 5 Classification:

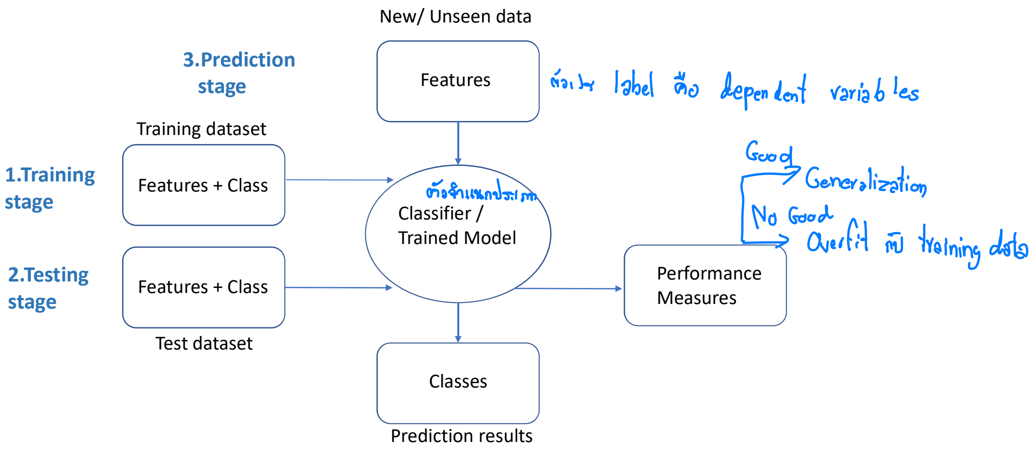
A predictive modeling problem that involves assigning a class label to each observation

Associate Professor Yachai Limpiyakorn, Ph.D.



Types of Classification

Classification (Supervised Learning)



Dataset

- ชุดข้อมูลสอน (training dataset) ใช้ในขั้นตอนการเรียนรู้เพื่อสร้างโมเดลผลลัพธ์
- ชุดข้อมูลทดสอบ (test dataset) ใช้ทดสอบโมเดลผลลัพธ์เพื่อวัดสมรรถนะ (performance) หรือ ความเป็นทั่วไป (generalization) ในการใช้โมเดลนั้นกับข้อมูลใหม่หรือข้อมูลทั่วไป
- ชุดข้อมูลตรวจความสมเหตุสมผล (validation dataset) ใช้ปรับแต่งสมรรถนะ (performance) ของโมเดล เช่น ค่าพารามิเตอร์ที่ใช้ในการเรียนรู้ หรือนำทาง (gauge) เพื่อหลีกเลี่ยง overfit

Dataset 80%

20%

Hold out ใช้ตอนเน้นชุดข้อมูลขนาดใหญ่

Model Assessment

- **Accuracy** = $\dfrac{\sum_{i=1}^{|test\_set|} \delta_i}{|test\_set|}$

where $\delta_i = 1$ (hit)

$= 0$ (miss)

- **Accuracy** = $\dfrac{\sum_{i=1}^{k} \sum_{j=1}^{|test\_fold_i|} \delta_{ij}}{|D|}$ ขนาด data set

where $\delta_{ij} = 1$ hit

$: 0$ miss

Model Assessment : k-fold Cross-validation

each sample เป็น Test data 1 ครั้ง

Training k-1 ครั้ง

stratified k-fold แบ่งสัดส่วน

บอกว่า train k รอบ

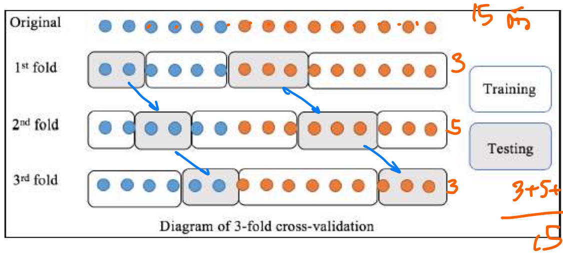


Diagram of 3-fold cross-validation

$\dfrac{3+5+3}{15}$

$\dfrac{11}{15}$

Confusion Matrix: Performance of Classifier

มี 2 class (+,-)

FN ผิดสี่ยงกว่า

FP ผิด (serious)

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

Sensitivity: ทั้งหมดของตัวจริง + ทายถูกกี่ตัว

Specificity: ทั้งหมดของตัวจริง - ทายถูกว่าเป็น - กี่ตัว

Negative Predictive Value: ทั้งหมดที่ทาย - ถูกกี่ตัว

TP = positive class correctly identified as positive

FN = positive class incorrectly identified as negative

FP = negative class incorrectly identified as positive

TN = negative class correctly identified as negative

Precision คือทั้งหมดที่ทายมาถูกกี่ครั้ง

Accuracy

- How good the model is at guessing the correct labels or ground truths.

		Predicted Class		
		Positive	**Negative**	
Actual Class	**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** $\frac{TP+TN}{(TP+TN+FP+FN)}$

ACCURACY = $\dfrac{\text{WHAT THE MODEL PREDICTED CORRECTLY}}{\text{EVERYTHING}}$

Precision

		Predicted Class		
		Positive	**Negative**	
Actual Class	**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** $\frac{TP+TN}{(TP+TN+FP+FN)}$

- Precision is the ratio of what the model predicted correctly to what the model predicted.
- There is one precision value for each category/ class.

PRECISION FOR 'WIN' = WHAT THE MODEL PREDICTED <u>CORRECTLY</u> AS 'WIN' / WHAT THE MODEL PREDICTED AS 'WIN'

PRECISION FOR 'LOSE' = WHAT THE MODEL PREDICTED <u>CORRECTLY</u> AS 'LOSE' / WHAT THE MODEL PREDICTED AS 'LOSE'

Confusion matrix of email classification (2)

		Predicted Class	
		Spam	**Non-Spam**
Actual Class	**Spam**	TP=45	FN=20
	Non-Spam	FP=5	TN=30

- **Precision** shows correctness achieved in positive prediction.

  Precision = 45/(45+5)= 90%

  The 90% of examples are classified as spam are actually spam.

- **Accuracy** is proportion of the total number of predictions that are correct.

  Accuracy = (45+30)/(45+20+5+30)

  = 75%

  The 75% of examples are correctly classified by the classifier.

https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

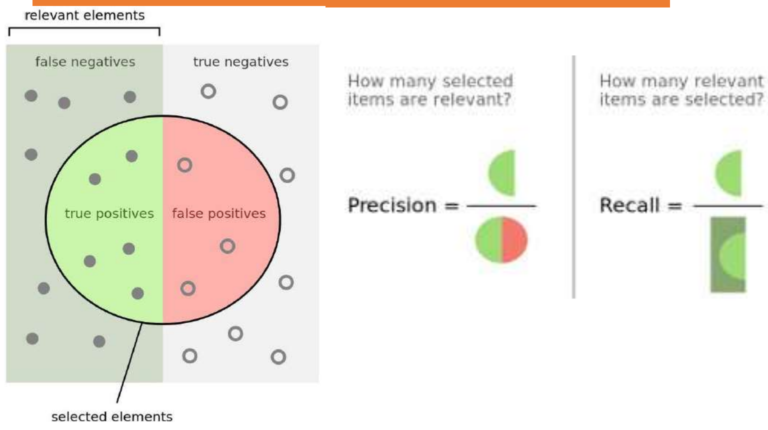
		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative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

RECALL FOR 'BOMB' = WHAT THE MODEL PREDICTED <u>CORRECTLY</u> AS 'BOMB' / WHAT IS ACTUALLY 'BOMB'

Recall (Sensitivity)

- index of diagnostic accuracy, also called *true positive rate* → a highly sensitive test rarely overlooks an actual positive
- A 90 percent sensitivity means that 90 percent of the diseased people screened by the test will give a "true-positive" result and the remaining 10 percent a "false-negative" result
- the cost of missing a prediction is much higher than a wrong prediction.

Precision vs. Recall



F1 score is a weighted average of the recall (sensitivity) and precision. F1 score might be good choice when you seek to balance between Precision and Recall.

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

It helps to compute recall and precision in one equation so that the problem to distinguish the models with low recall and high precision or vice versa could be solved.

F1 มาจากสูตร Fβ

$$F\beta \text{ Score} = \frac{1+\beta^2}{\frac{1}{\text{Precision}} + \frac{\beta^2}{\text{Recall}}} = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}}$$

ถ้า $\beta = 1$ จะเป็น F1

F1 Score

- F1 score ranges from 0-100%
- Higher F1 score → better classifier
- β denotes user-defined hyperparameter
- β > 0 always ***
- β > 1 favors Recall ถ้าให้ความสำคัญกับ recall ก็ให้ $\beta > 1$
- β < 1 favors Precision
- If consider recall as twice important as precision, set β=2.
- Standard F-score equivalent set β=1

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	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 $\frac{TP+TN}{(TP+TN+FP+FN)}$

Specificity

- ability of a test to identify correctly those without disease, also called *true negative rate*
- A 90 percent specificity means that 90 percent of the non-diseased persons will give a "true-negative" result, 10 percent of non-diseased people screened by the test will be wrongly classified as "diseased" when they are not.
- A highly specific test rarely registers a positive classification for anything that is not the target of testing.

Confusion matrix of email classification (1)

		Predicted Class: Spam	Predicted Class: Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

- **Sensitivity (Recall):** *True Positive Rate;* proportion of emails which are spam among all spam emails

Recall = 45/(45+20) = 69.23%

The 69.23% spam emails are correctly classified and excluded from all non-spam emails.

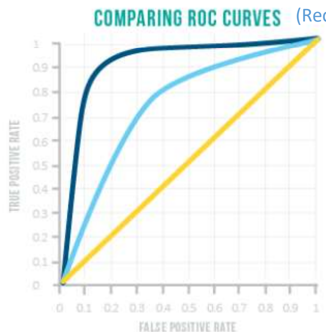
- **Specificity:** *True Negative Rate;* proportion of emails which are non-spam among all non-spam emails

Specificity = 30/(30+5) = 85.71%

The 85.71% non-spam emails are accurately classified and excluded from all spam emails.

The TPR (sensitivity) is plotted against the FPR (1 - specificity) for given cut-off values to give a plot similar to the one below. Ideally a point around the shoulder of the curve is picked which both limits false positives whilst maximizing true positives.

balance model ที่ให้ TP เยอะ FP น้อยๆ



(Receiver Operating Characteristics Curve)

Area Under Curve

AUC=1 (ideal model) ability to identify all TP while avoiding FP

AUC=0.5 (random guess) TPR and FPR would increase in parallel to one another

$$FPR = \frac{FP}{FP + TN}$$ Specificity สูง

A test that gave a ROC curve such as the yellow line would be no better than random guessing, pale blue is good, but a test represented by the dark blue line would be excellent. It would make cutoff determination relatively simple and yield a high true positive rate at very low false positives rate – sensitive and specific.

Scaling Data: Before or After Train-Test Split?

DATA LEAKAGE ข้อมูลรั่ว

HAPPENS WHEN INFORMATION FROM OUTSIDE THE TRAINING DATASET IS USED TO CREATE A MODEL.

NEVER FIT YOUR SCALER TO THE TEST DATA ห้ามทำ preprocessing ใน test_data

TREAT THE TEST DATA AS FUTURE, UNSEEN DATA ทำใน train

ALWAYS REMEMBER: SCALE BASED ON THE TRAINING SET, AND THEN APPLY THOSE TRANSFORMATIONS TO THE TEST SET TO MAINTAIN THE PURITY OF THE TEST ENVIRONMENT, I.E. THE SCALING OPERATION DOES NOT HAVE INFORMATION ABOUT THE DISTRIBUTION OF THE TEST SET.

https://medium.com/@megha.natarajan/scaling-data-before-or-after-train-test-split-35e9a9a7453f

Step by Step:

- **Split Your Data:** Divide your dataset into training and test sets, typically using a 70:30 or 80:20 ratio, ensuring each set is representative of the overall distribution.
- **Compute Scaling Parameters on the Training Set:** This includes the mean and standard deviation in standard scaling and the min/max values in min-max scaling.
- **Scale the Training Data:** Apply the scaling transformation to the training data.
- **Fit the Model:** Use the scaled training data to train your model.
- **Scale the Test Data:** Before making predictions, scale the test data using the same parameters computed from the training data.
- **Evaluate the Model:** Assess model performance using the scaled test data.

IMBALANCED CLASSIFICATION

Classification problem where there is an unequal distribution of classes in the training dataset.

The number of samples that belong to each class may be referred to as the class distribution.

Most imbalanced classification problems involve two classes: a negative case with the majority of examples and a positive case with a minority of examples.

For example, found every 1 fraud per 10,000 authentic transactions, i.e. the class distribution is highly unbalanced. เรื่องปกติที่เป็น imbalance dataset ข้อมูลคนป่วย น้อยกว่าข้อมูลคนปกติ

Many real-world classification problems have an imbalanced class distribution: fraud detection, spam detection, and churn prediction.

Imbalanced classifications pose a challenge for predictive modeling.

Poor predictive performance, specifically for the minority class considered more important, but more sensitive to classification errors.

TECHNIQUES TO DEAL WITH A CLASS IMBALANCE

Oversampling

oversampling minority class boost up ให้สูงขึ้น

increase number of minority observations until reaching a balanced dataset

- Random oversampling- simply duplicate minority class observations, reducing variance of dataset, though duplicate minority แต่ various ต่ำลง ไม่ตรง
- Synthetic Minority Over-sampling TEchnique (SMOTE)
- Adaptive Synthetic (ADASYN)

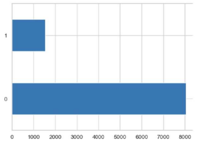
Undersampling

undersampling majority class ลด class ที่เยอะเกิน

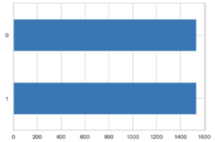
This could potentially result in removing key characteristics of the majority class. สุ่ม class 0 เท่าจำนวน minority class

- Random undersampling

Class 0 (~~8043~~ 1533 instances) คนปกติ
Class 1 (1533 instances) คนป่วย



Overall $\boxed{3066}$ instances
Each class 1533 instances



- Near miss
- Tomeks links