Classical Machine Learning: Classification and Regression (IV)

Learning Objectives

- Learn how to choose the *best* machine learning model.
- Learn how to deal with class imbalance problems (with some useful performance metrics).

Compare and choose machine learning model

C-S David Chen, Department of Civil Engineering, National Taiwan University

# Choose the "best" machine learning model


Model_compare.ipynb

- **Learning algorithm $\mathcal{A}$ depends on training examples $\mathcal{D}$ and hypothesis set $\mathcal{H}$** and there is no single learning algorithm dominated for every kind of data. (No free lunch!)
- **You cannot know which algorithms are best suited to your problem beforehand.**
- **When comparing performance of different algorithms, make sure each algorithm is evaluated in the same way on the same data.**

Algorithm Comparison
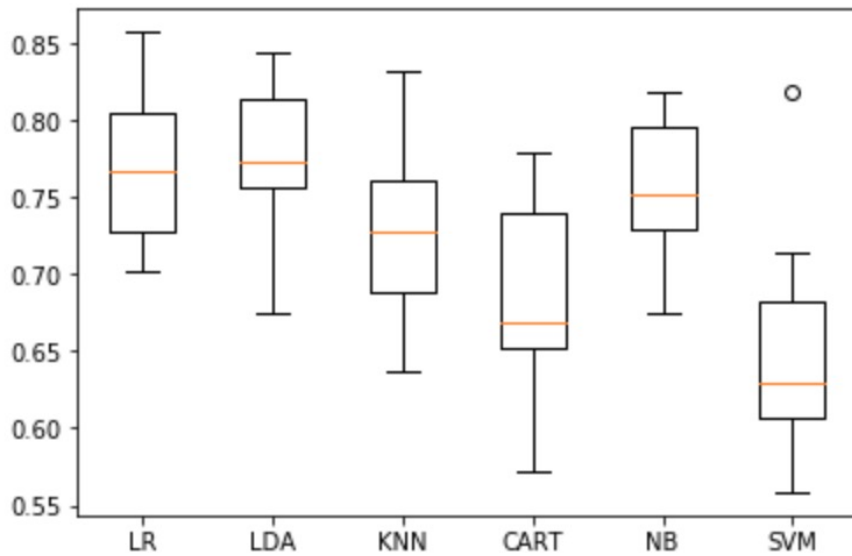
Algorithm Comparison
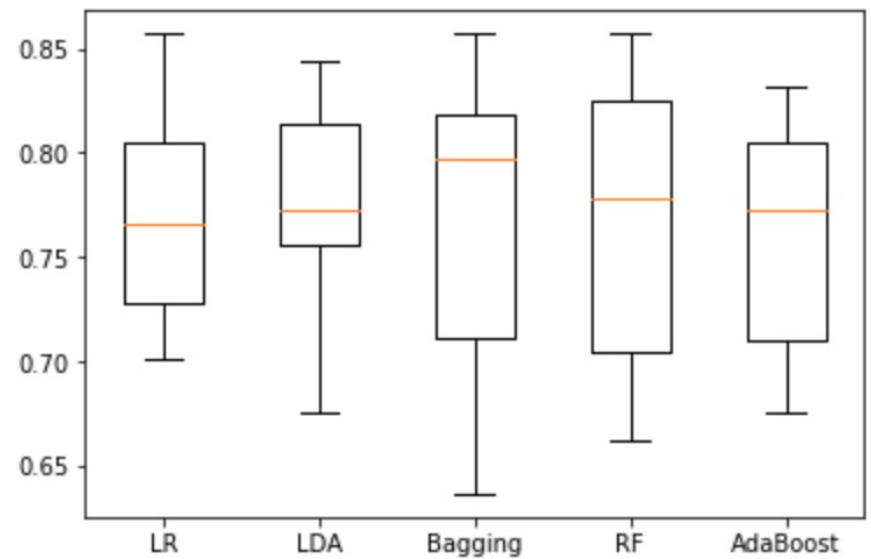
Learning Principle: Occam's Razor

Occam's Razor: The simplest model that fits the data is also the most plausible (合理).

- Occam's razor in machine learning means simpler model has a better chance of being right. However, if a complex explanation of the data performs better, we will take it.
- The argument that simpler has a better chance of being right goes as follows. With complex hypotheses, there would be enough of them to fit the data set regardless of what the labels are, even if these are completely random. Therefore, fitting the data does not mean much.
- If, instead, we have a simple model with few hypotheses and we still found one that perfectly fits. This is surprising, and therefore it means some thing.

Class imbalance problems and some alternative performance metrics

C-S David Chen, Department of Civil Engineering, National Taiwan University

Class Imbalance Problem

- Lots of classification problems where the classes are skewed (more records from one class than another)
  - Credit card fraud
  - Intrusion detection
  - Defective products in manufacturing assembly line
  - COVID-19 test results on a random sample

- **Key Challenge**:
  - Evaluation measures such as accuracy are not well-suited for imbalanced class

Class Imbalance Techniques

- **Choose better performance metrics**
- **Balance skewed classes**
- **Cost-sensitive learning**

Confusion Matrix

● Confusion Matrix:

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Accuracy

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a ✔ (TP)	b (FN)
Class=No	c (FP)	d ✔ (TN)

● Most widely-used metric:

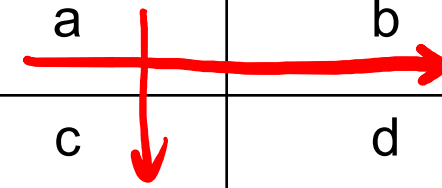$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Problem with Accuracy

- Consider a 2-class problem
  - Number of Class NO examples = 990
  - Number of Class YES examples = 10

- If a model predicts everything to be class NO, accuracy is 990/1000 = 99 %
  - This is misleading because this trivial model does not detect any class YES example
  - Detecting the rare class is usually more interesting (e.g., frauds, intrusions, defects, etc)

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	0	10
	Class=No	0	990

Alternative Measures

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F-measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Alternative Measures

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS — Class=Yes	10	0
ACTUAL CLASS — Class=No	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F-measure (F)} = \frac{2*1*0.5}{1+0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS — Class=Yes	1	9
ACTUAL CLASS — Class=No	0	990

$$\text{Precision (p)} = \frac{1}{1+0} = 1$$

$$\text{Recall (r)} = \frac{1}{1+9} = 0.1$$

$$\text{F-measure (F)} = \frac{2*0.1*1}{1+0.1} = 0.18$$

$$\text{Accuracy} = \frac{991}{1000} = 0.991$$

Recall (查全率) and Precision (精確率)

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a	b
Class=No	c	d

$$\text{Recall} = \frac{a}{a+b}$$

$$\text{Precision} = \frac{a}{a+c}$$

A: 所有的肯定解

S: AI 輸出的肯定解

H: AI 回答正確

0	10
0	990
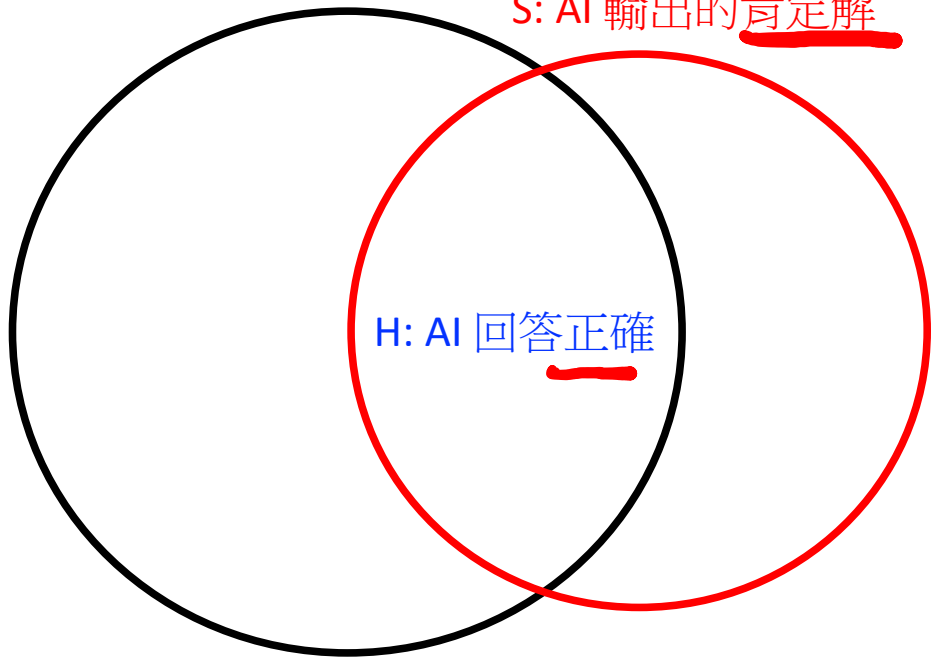
$$\text{Recall (查全率)} = \frac{H}{A}$$

$$\text{Precision (精確率)} = \frac{H}{S}$$

Learning from Movie: 要重視查全率還是精確率?

当着陛下的面行凶

Fun Time:

要重視查全率
還是精確率？

「寧可錯殺，不可錯放」重視的是 (1)查全率 **Recall (2)** 精確率**Precision**

A: 所有的肯定解

S: AI 輸出的肯定解

H: AI 回答正確
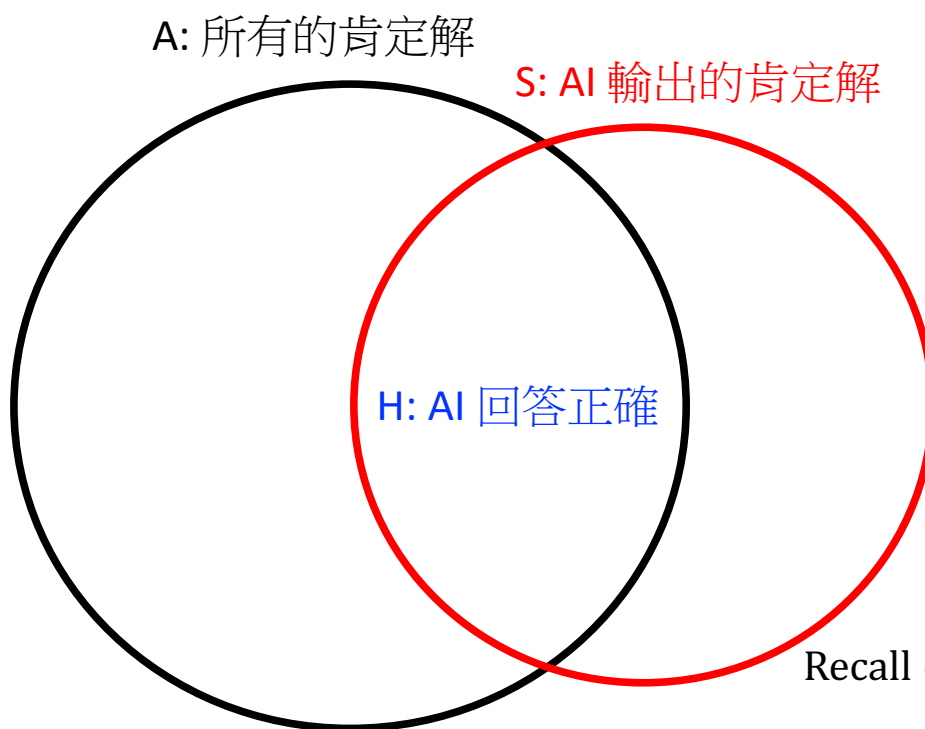
$Recall\ (查全率) = \frac{H}{A}$

$Precision\ (精確率) = \frac{H}{S}$

Fun Time:

要重視查全率
還是精確率？

開發人臉辨識系統我們應該要重視查全率
Recall 還是精確率 **Precision? (1)查全率**
Recall (2) 精確率Precision

A: 所有的肯定解

S: AI 輸出的肯定解

H: AI 回答正確
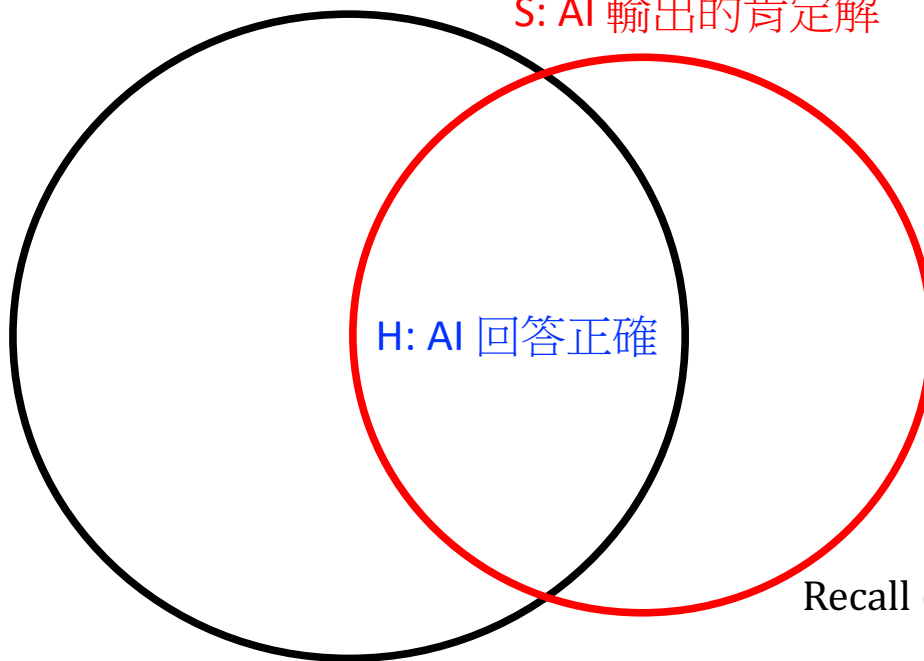
$$Recall\ (查全率) = \frac{H}{A}$$
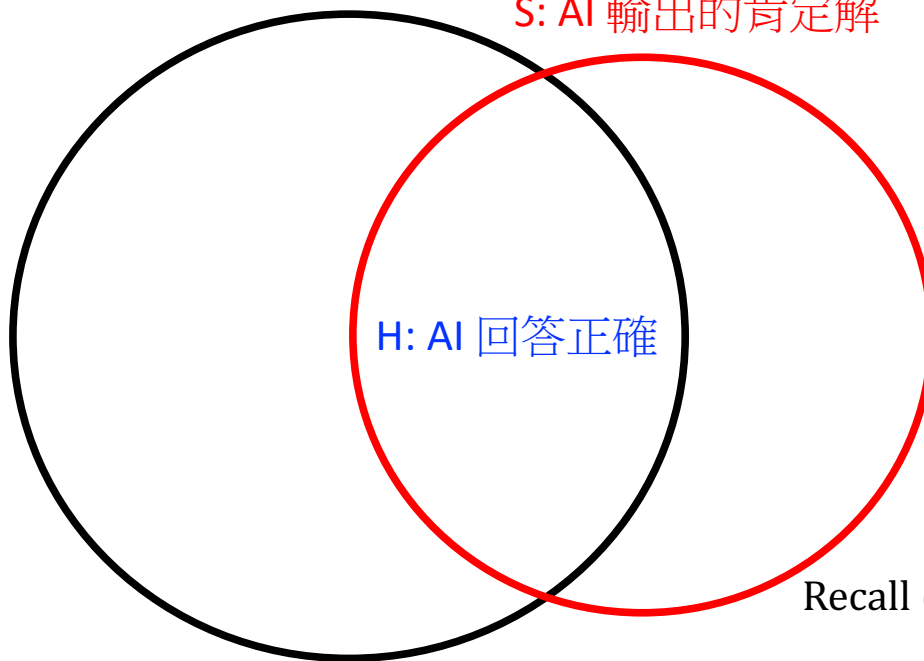
$$Precision\ (精確率) = \frac{H}{S}$$

Fun Time:

要重視查全率
還是精確率?

開發癌症檢測系統我們應該要重視查全率
Recall 還是精確率 **Precision? (1)查全率
Recall (2) 精確率Precision**

slido #872333
Polls

A: 所有的肯定解

S: AI 輸出的肯定解

H: AI 回答正確

$$Recall\ (查全率) = \frac{H}{A}$$

$$Precision\ (精確率) = \frac{H}{S}$$

Measures of Classification Performance

	PREDICTED CLASS		
		Yes	No
ACTUAL CLASS	Yes	TP	FN
	No	FP	TN

α **is a Type I error or a false positive (FP).**

β **is a Type II error or a false negative (FN).**

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$ErrorRate = 1 - accuracy$$

$$Precision = Positive\ Predictive\ Value = \frac{TP}{TP + FP}$$

$$Recall = Sensitivity = TP\ Rate = \frac{TP}{TP + FN}$$

$$Specificity = TN\ Rate = \frac{TN}{TN + FP}$$

$$FP\ Rate = \alpha = \frac{FP}{TN + FP} = 1 - specificity$$

$$FN\ Rate = \beta = \frac{FN}{FN + TP} = 1 - sensitivity$$

$$Power = sensitivity = 1 - \beta$$

F_α

F_1 measure

ROC (Receiver Operating Characteristic)

- A graphical approach for displaying trade-off between detection rate and false alarm rate

- Developed in 1950s for signal detection theory to analyze noisy signals

- ROC curve plots TPR against FPR

  – Performance of a model represented as a point in an ROC curve

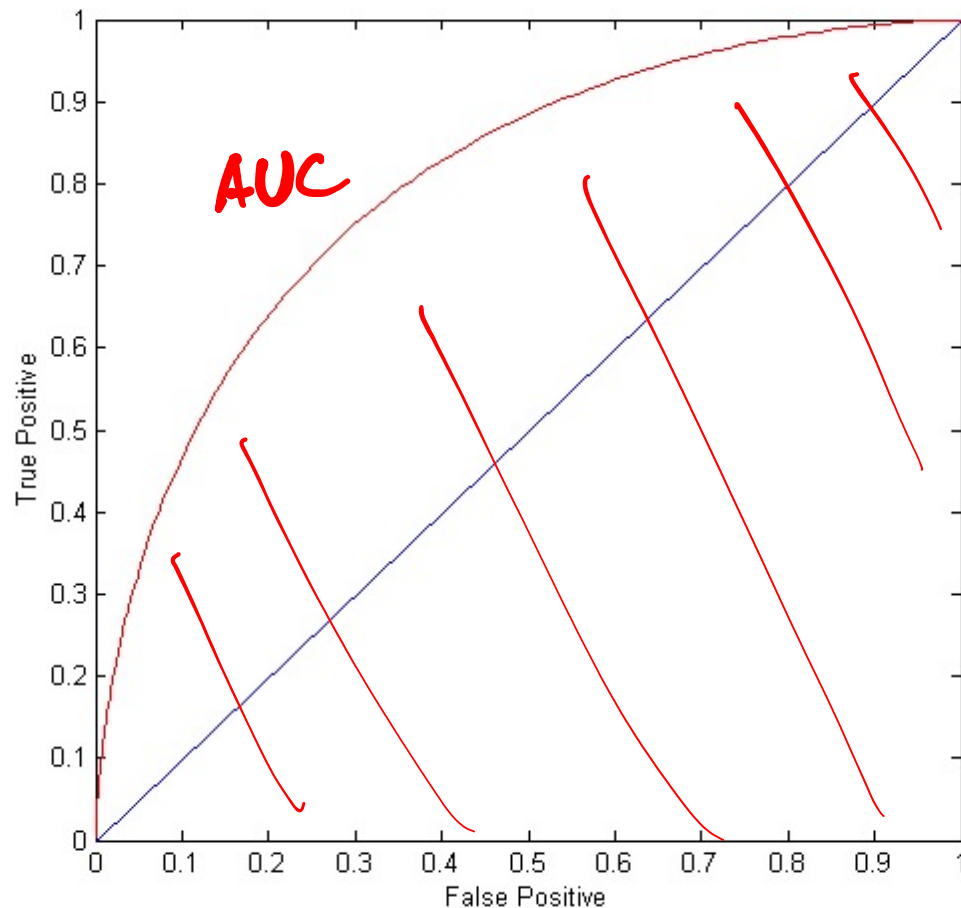| ACTUAL CLASS		PREDICTED CLASS	
		Yes	No
	Yes	TP	FN
	No	FP	TN

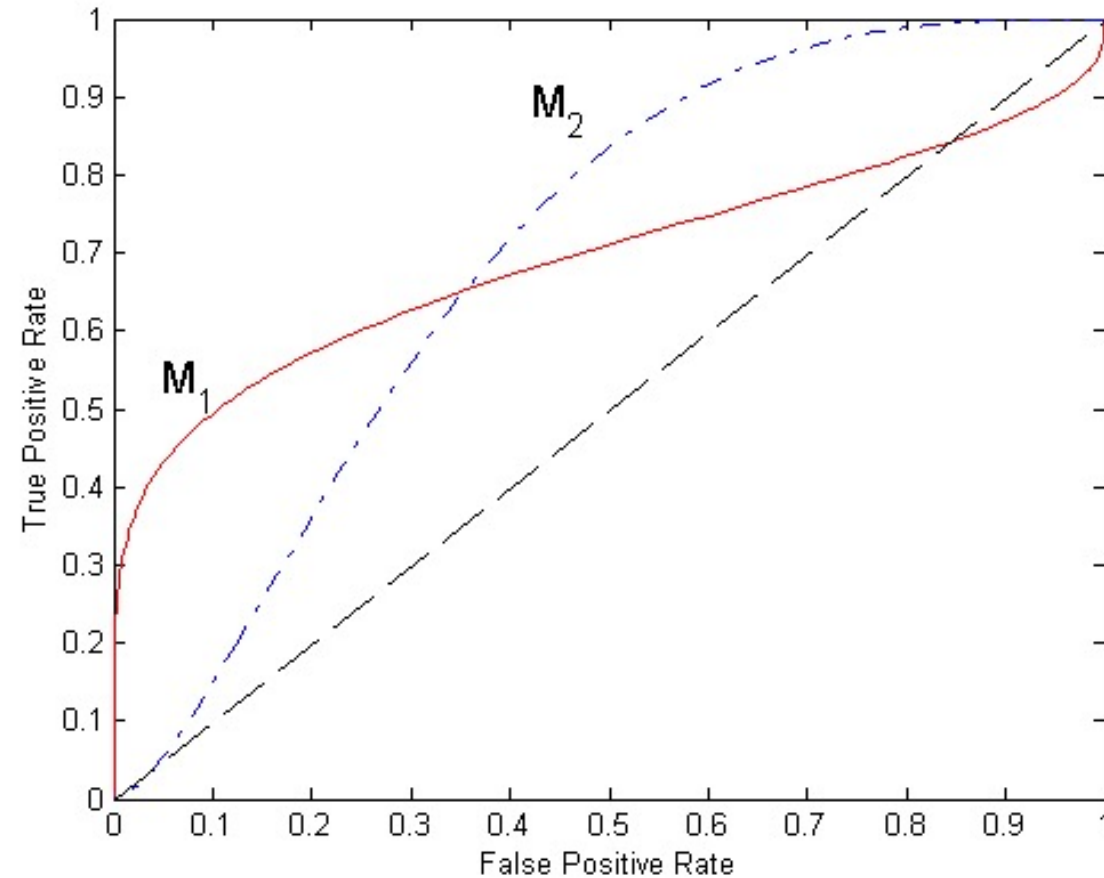$$TPR = recall = \frac{TP}{TP + FN}$$

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

ROC Curve

(TPR,FPR):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal

- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class

Using ROC for Model Comparison

- No model consistently outperforms the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR

- Area Under the ROC curve (AUC)
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

Common Performance Metrics for Classification and Regression

分類問題	迴歸問題
混淆矩陣（Confusion Matrix）	均方誤差（Mean Square Error）
正確率（Accuracy）	決定係數（Coefficient of Determination）
精確率（Precision）	
召回率（Recall）	
F1 值（F1-Score）	
AUC（Area Under the Curve，曲線下面積）	

Performance_metrics.ipynb

Class Imbalance

- **Choose better performance metrics**
  - **Accuracy is often not a good metric for class imbalance problems.**
  - **Use precision, recall, ROC (AUC), F-score etc.**
- ✓ **Balance skewed classes**
  - **Data oversampling and undersampling**
- ✓ **Cost-sensitive learning**

Class Imbalance: Balance Skewed Classes

- **Data Oversampling**
  - **Random Oversampling**
  - **SMOTE**
  - **...**

- **Data Undersampling**
  - **Random Undersampling**
  - **Tomek Links**
  - **Edited Nearest Neighbors**
  - **...**

- **Combined Oversampling and Undersampling**

IM_data_sampling.ipynb

Class Imbalance: Cost Sensitive Algorithms

- **Modified version of machine learning algorithms designed to take the differing costs of misclassification into account when fitting the model on training records.**

- **Many cost-sensitive algorithms to choose**
  - **Logistic regression**
  - **Support vector machines**
  - **Decision trees**
  - **Random forest**
  - **Gradient boosting**
  - **…**

jupyterlab

IM_cost_sensitive_algo.ipynb

Summary: Class Imbalance Problems and Techniques

- Many practical classification problems are **class imbalance problems** where the classes are skewed (more records from one class than another)
- Sometimes you need to choose better performance metrics to train your model.
  - Accuracy is often not a good metric for class imbalance problems.
  - Use precision, recall, ROC (AUC), F-score etc.
- You can use techniques from data oversampling and undersampling (or combination) to balance the skewed class.
- Cost-sensitive learning algorithms are also very useful to attack class imbalance problems.