Metrics & Methods for Performance Evaluation

CS 6316 ~ Machine Learning

Fall 2017

Material adapted from Tan, Steinbach, Kumar ~ Model Evaluation ~ Introduction to Data Mining

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Focus on the **predictive capability** of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- **Confusion Matrix**: Summary of prediction results on a classification problem

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Confusion Matrix

Predictive Model: Evaluation

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

predictive result / classification		actual result / classification: yes	actual result / classification: no
	yes	**tp** (true positive)	**fp** (false positive) ← Type 1 error
	no	**fn** (false negative)	**tn** (true negative)

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{True Negative Rate} = \frac{tn}{tn + fp}$$

Terminology

- **True Positive (TP):** actually true; predicted true
- **True Negative (TN):** actually false; predicted false
- **False Positive (FP):** actually false; predicted true [Type I error]
  - "False alarm"
- **False Negative (FN):** actually true; predicted false [Type II error]

Accuracy

- The most obvious and most widely-used metric:

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

- **Accuracy** is the ratio of correct predictions to total predictions made
- Often presented as a percentage

Limitation of Accuracy

- Let's consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9%
  - Accuracy is misleading because model does not detect any class 1 examples!

Misclassification Rate / Error Rate

- Classification accuracy can be easily turned into a misclassification rate (or **error rate**) by inverting the value:

- **Error rate = (1 – (correct predictions / total predictions)) * 100**

Cost Matrix

- **C(i|j):** Cost of misclassifying class j example as class i

		PREDICTED CLASS	
ACTUAL CLASS	C(i|j)	Class=Yes	Class=No
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M_2	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

Cost vs. Accuracy

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

Cost	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	p	q
	Class=No	q	p

Accuracy is proportional to cost if
1. C(Yes|No)=C(No|Yes) = q
2. C(Yes|Yes)=C(No|No) = p

$N = a + b + c + d$

$Accuracy = (a + d)/N$

$$\begin{aligned} Cost &= p\,(a + d) + q\,(b + c) \\ &= p\,(a + d) + q\,(N - a - d) \\ &= q\,N - (q - p)(a + d) \\ &= N\,[q - (q-p) \times Accuracy] \end{aligned}$$

Cost-Sensitive Measures

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

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

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

Cost-Sensitive Measures

$$Precision\ (p) = \frac{TP}{TP + FP}$$

- **Precision** (positive predictive value) is the fraction of relevant instances among the retrieved instances)
- In other words: the ability of the classifier not to label as positive a sample that is negative

Cost-Sensitive Measures

$$Recall\ (r) = \frac{TP}{TP + FN}$$

- **Recall** (**sensitivity** or true positive rate) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances
- In other words: the ability of the classifier to find all the positive samples
- *E.g. percentage of sick people who are correctly identified as having the condition*

Cost-Sensitive Measures

$$F - measure\ (F) = \frac{2\,TP}{2\,TP + FP + FN}$$

- **F-measure** (F1 score) is the harmonic mean of precision and sensitivity
- In other words: it's a measure of a test's accuracy. Considering both precision and recall.
- F1 score = 1: (best case) *perfect precision and recall*
- F1 score = 0: (worst case)

Others (Specificity)

- **Specificity** (true negative rate): measures the proportion of negatives that are correctly identified as such
  - *E.g. percentage of healthy people who are correctly identified as not having the condition*

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

Others (Matthews Correlation Coefficient)

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

- **Matthews Correlation Coefficient (MCC):** is used as a measure of the quality of binary classifications. It is regarded as a ***balanced measure which can be used even if the classes are of very different sizes***. It's a correlation coefficient between the observed and predicted binary classifications
- +1: prefect prediction; 0: no better than random;
- -1: total disagreement between prediction and observation

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?

- Methods for Performance Evaluation
  - How to obtain reliable estimates?

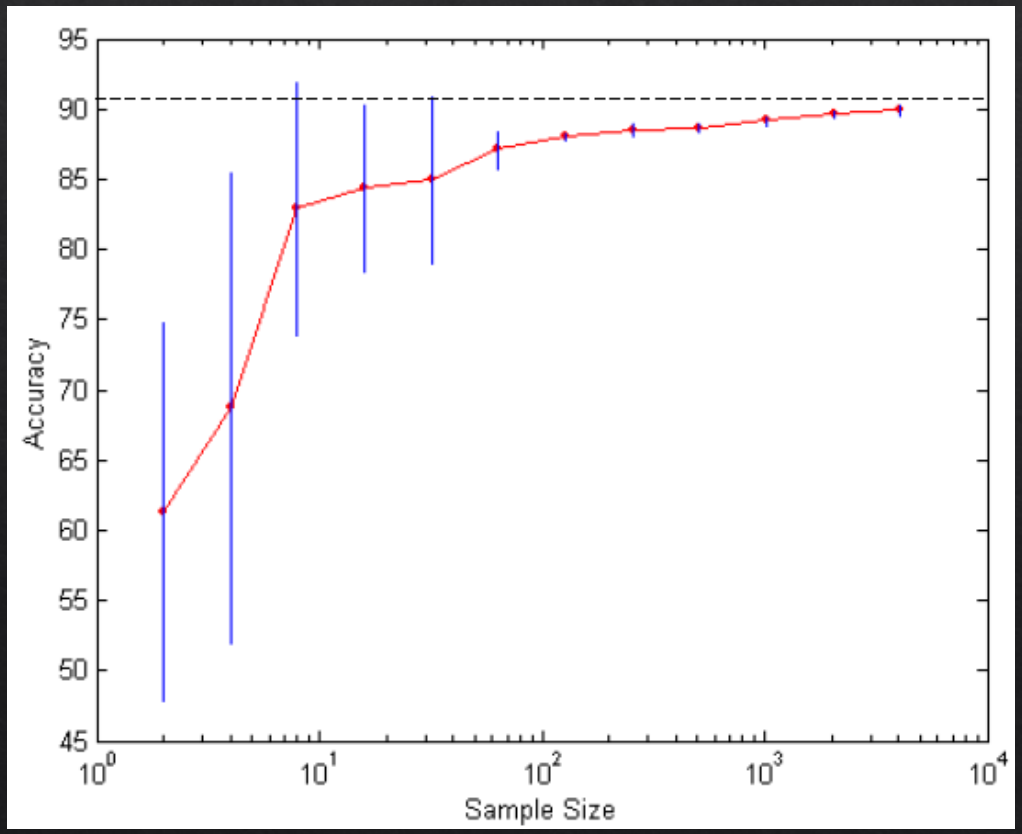
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?

- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve:
  - Arithmetic sampling (Langley, et al.)
  - Geometric sampling (Provost et al.)
- Effect of small sample size:
  - Bias in the estimate
  - Variance of estimate

Methods of Estimation

- **Holdout**
  - Reserve 2/3 for training and 1/3 for testing
- **Random subsampling**
  - Repeated holdout
- **Stratified sampling**
  - oversampling vs undersampling
- **Bootstrap**
  - Sampling with replacement
- **Cross validation**
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

ROC (Receiver Operating Characteristic)

- Developed in the 1950s for **signal detection theory** to analyze noisy signals
  - Characterize the trade-off between positive hits (**TP**) and false alarms (**FP**)

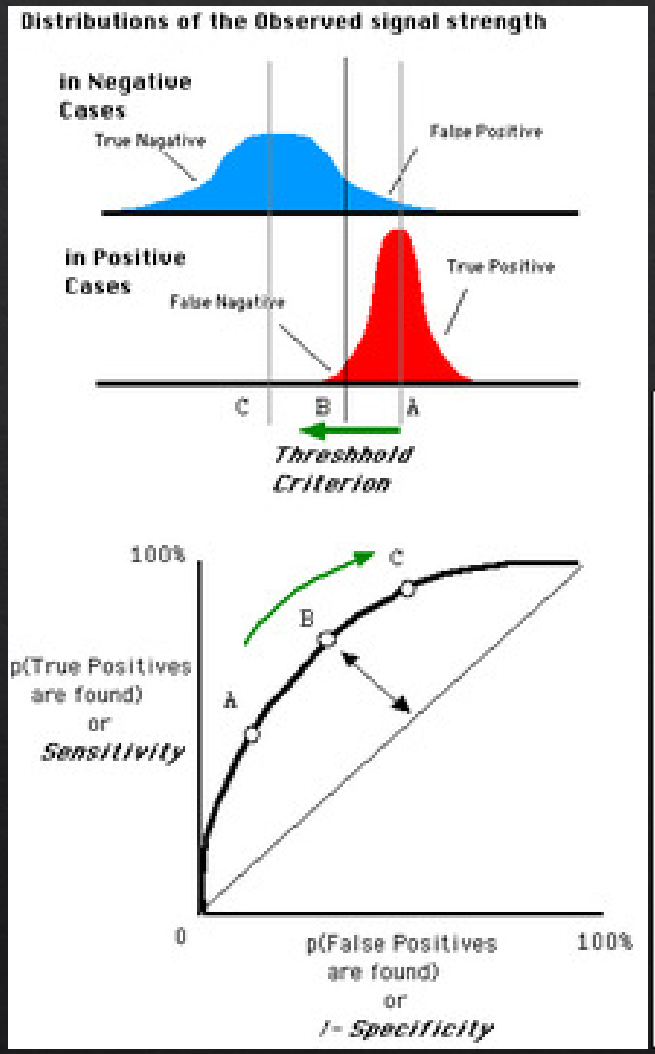
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)

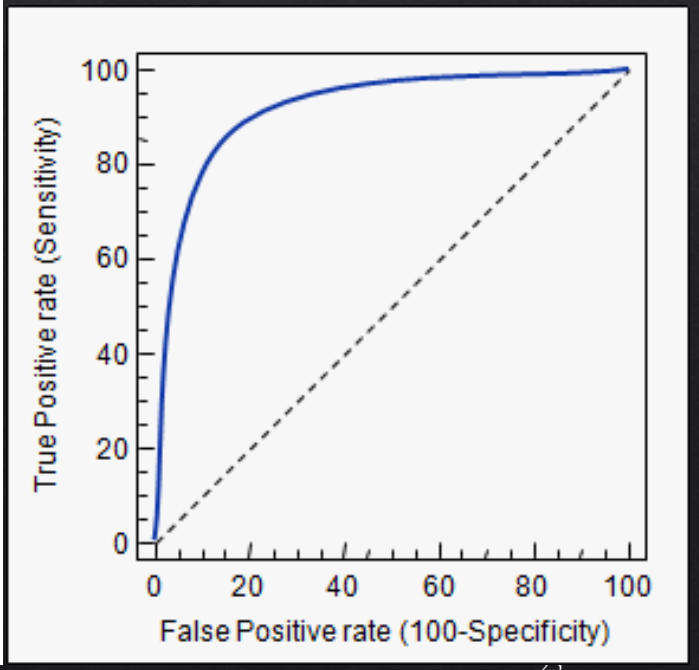
ROC (Receiver Operating Characteristic)

- ROC curve plots TP (on the y-axis) against FP (on the x-axis)

- Performance of each classifier represented as a point on the ROC curve
  - Changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

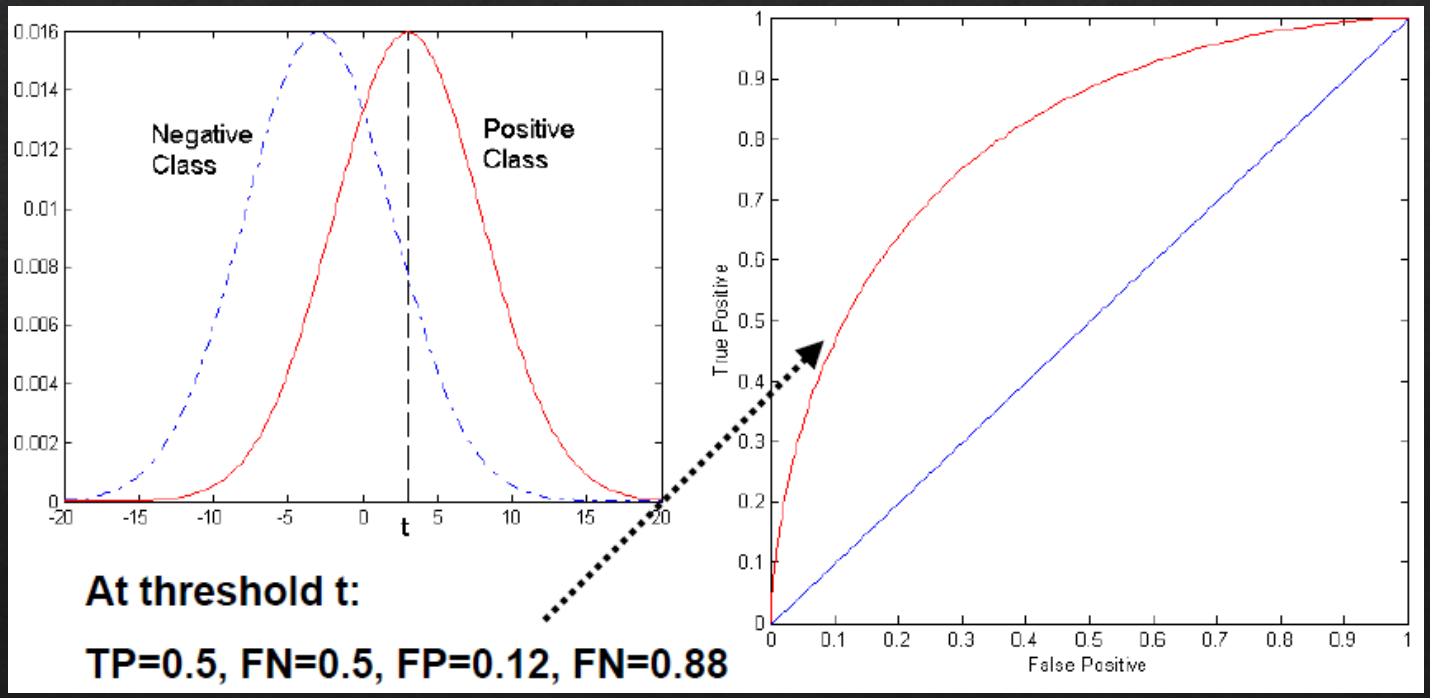
ROC Curve


Distributions of the Observed signal strength
in Negative Cases
True Negative
False Positive
in Positive Cases
False Negative
True Positive
C
B
A
Threshhold Criterion
100%
C
B
A
p(True Positives are found) or Sensitivity
0
p(False Positives are found) or 1 - Specificity
100%


True Positive rate (Sensitivity)
100
80
60
40
20
0
0
20
40
60
80
100
False Positive rate (100-Specificity)

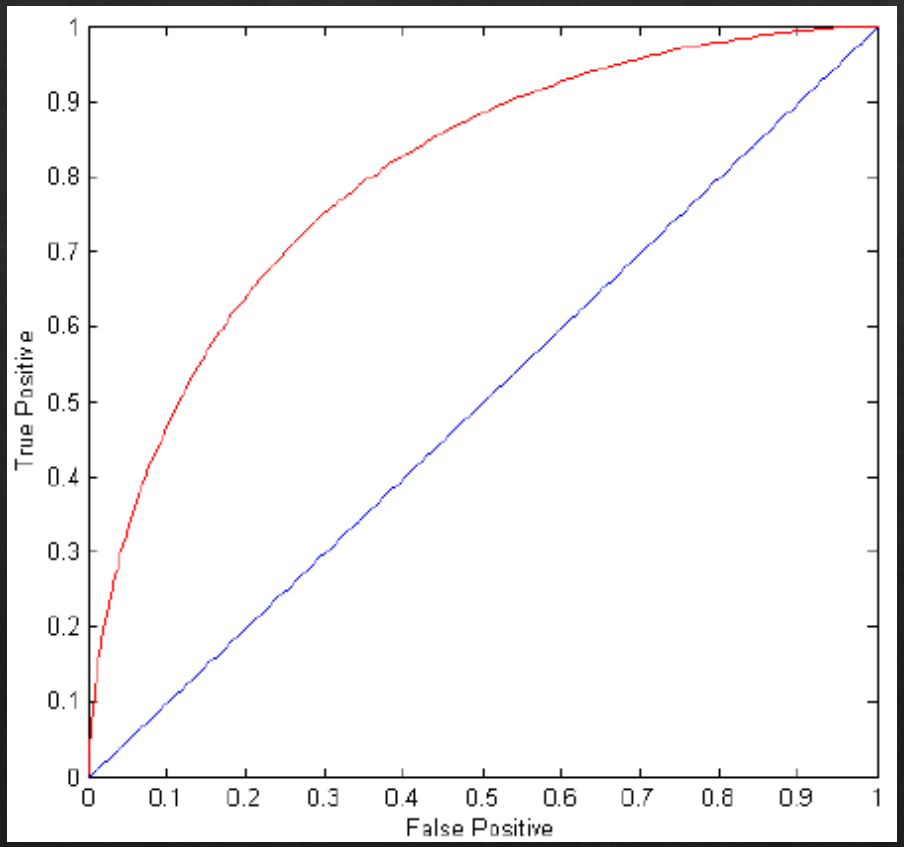
ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)
- Any points located at x > t is classified positive



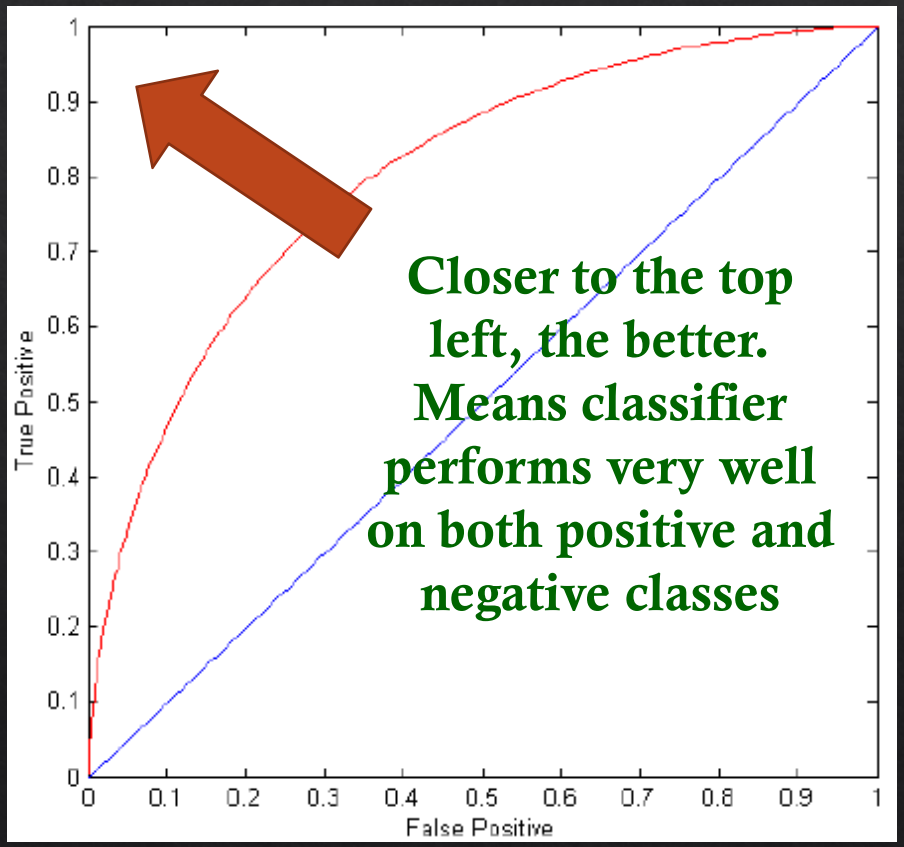
ROC Curve

- **(TP, FP):**
  - **(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
    - Prediction is opposite of the true class

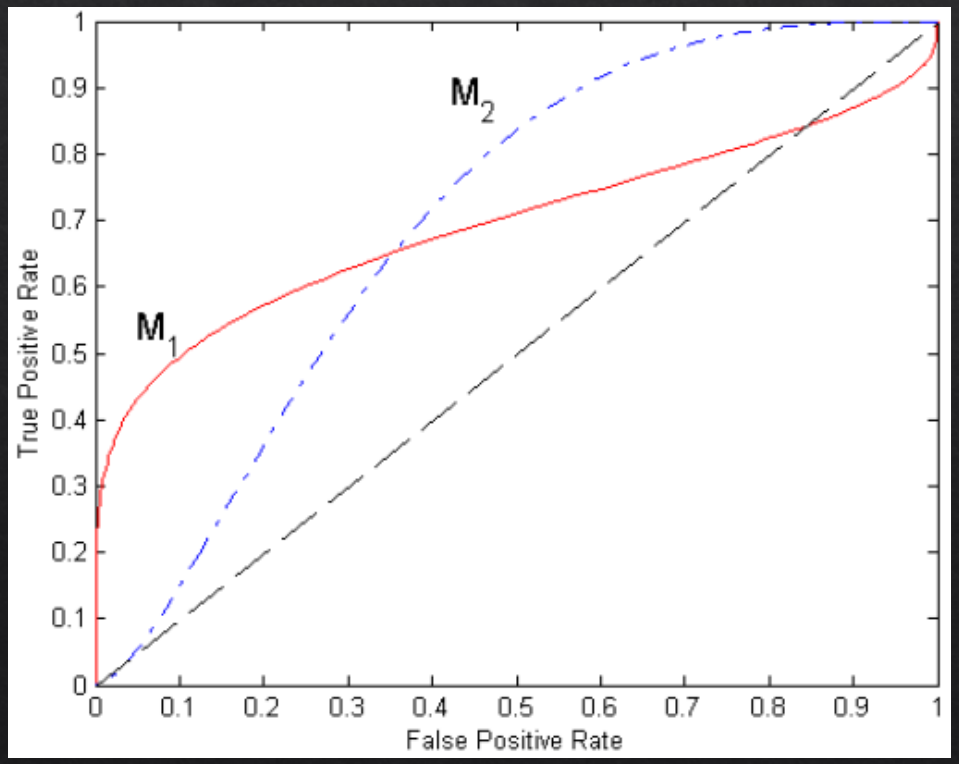


ROC Curve

- **(TP, FP):**
  - **(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
    - Prediction is opposite of the true class



Using ROC for Model Comparison



- No model consistently outperform the other
  - M_1 is better for small False Positive Rate (FPR)
  - M_2 is better for large FPR

- **Area Under the ROC curve:**
  - **Ideal**: Area = 1
  - Random guess: Area = 0.5

How to Construct a ROC curve

Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each rest instance **P(+|A)**
- **Sort** the instances according to P(+|A) in decreasing order
- **Apply threshold** at each unique value of P(+|A)
- **Count** the number of TP, FP, TN, FN at each threshold
- TP rate, **TPR** = TP / (TP + FN)
- FP rate, **FPR** = FP / (FP + TN)

How to Construct a ROC curve

	Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=		0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
	TP	5	4	4	3	3	3	3	2	2	1	0
	FP	5	5	4	4	3	2	1	1	0	0	0
	TN	0	0	1	1	2	3	4	4	5	5	5
	FN	0	1	1	2	2	2	2	3	3	4	5
→	TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→	FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve: