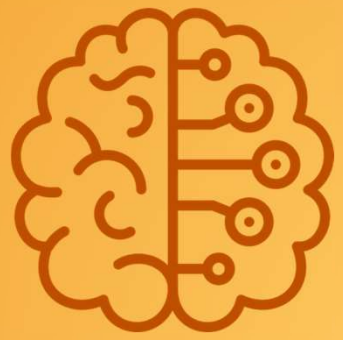
CS 4375 – Introduction to Machine Learning

Evaluating Machine Learning Models

Erick Parolin

THE UNIVERSITY OF TEXAS AT DALLAS



Model Evaluation

- **So far, we covered a plenty of learners for classification task…**
  - Learning algorithm
  - How to use them for classification
  - How they explore the feature space
  - Etc.

- **Now we will learn how to evaluate the *performance* of these models**
  - **Focus on their predictive capability (NOT time/space complexity)**
  - How to evaluate the performance of a model.
  - How to obtain reliable estimates.
  - How to compare performance of the different models.

Metrics for Performance Evaluation

Confusion Matrix

Confusion Matrix		Predicted Class	
		Positive (Class=1)	Negative (Class=0)
Actual Class	Positive (Class=1)	TP	FN
	Negative (Class=0)	FP	TN

Each record from the dataset used for model evaluation will fall into one of the cells of the confusion matrix:

TP: True Positive
FP: False Positive
FN: False Negative
TN: True Negative

Metrics for Performance Evaluation

Accuracy

Confusion Matrix		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	TP	FN
	Negative (Class=0)	FP	TN

TP: True Positive
FP: False Positive
FN: False Negative
TN: True Negative

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

Metrics for Performance Evaluation

Accuracy

- **Limitation:** what if I have *unbalanced dataset*?

- **Example:**
  - Number of Class 1 instances is 99,900
  - Number of Class 0 instances is 100
  - If the model is a constant that predicts everything as Class 1, we have accuracy 99.9%

  - Accuracy may be misleading: the model does not detect any Class 0!!!

Metrics for Performance Evaluation

Cost Matrix

Cost Matrix		Predicted Class	
		Positive (Class=1)	Negative (Class=0)
Actual Class	Positive (Class=1)	C(1|1)	C(0|1)
	Negative (Class=0)	C(1|0)	C(0|0)

C(P | A): cost of predicting class P when actual class is A

Metrics for Performance Evaluation

Cost of Classification – Numerical Example

Dataset
Class 1: 357 examples
Class 0: 5,303 examples

Cost Matrix

Cost Matrix		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	-1	200
	Negative (Class=0)	20	0

Model 1		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	282	75
	Negative (Class=0)	462	4,841

Model 1
Accuracy: **90.5%**
Cost: 23,958

Model 2		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	332	25
	Negative (Class=0)	607	4,696

Model 2
Accuracy: 88.8%
Cost: 16,808

Metrics for Performance Evaluation

Precision: measures how well a model can correctly predict positive instances

Confusion Matrix		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	TP	FN
	Negative (Class=0)	FP	TN

TP: True Positive
FP: False Positive
FN: False Negative
TN: True Negative

$$Precision = \frac{TP}{TP + FP}$$

Metrics for Performance Evaluation

Recall: measures how well a model can identify positive instances

Confusion Matrix		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	TP	FN
	Negative (Class=0)	FP	TN

TP: True Positive
FP: False Positive
FN: False Negative
TN: True Negative

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

Metrics for Performance Evaluation

F-score: a measure of the harmonic mean of precision and recall.

Confusion Matrix		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)
Actual Class	Positive (Class=1)	TP	FN
	Negative (Class=0)	FP	TN

TP: True Positive
FP: False Positive
FN: False Negative
TN: True Negative

$$F_\beta = \frac{(\beta^2 + 1) \cdot Prec \cdot Rec}{\beta^2 \cdot Prec + Rec}$$

Most common ($\beta = 1$):

$$\mathbf{F1} = 2\frac{\mathbf{Prec \cdot Rec}}{\mathbf{Prec + Rec}} = \frac{\mathbf{2TP}}{\mathbf{2TP + FP + FN}}$$

What if we have more than two classes?

Multiclass (classes are mutually exclusive)

- $\boldsymbol{Prec_k}$**:** Measures how many of the predicted positive instances for class *k* are correct.
- $\boldsymbol{Rec_k}$**:** Measures how many actual positive instances for class *k* were correctly predicted.
- $\boldsymbol{F1_k}$**:** The harmonic mean of precision and recall for class *k*.
- ***Accuracy*** is computed in the same manner.

$$Prec_k = \frac{TP_k}{TP_k + FP_k} \qquad Rec_k = \frac{TP_k}{TP_k + FN_k} \qquad F1_k = 2\frac{Prec_k \cdot Rec_k}{Prec_k + Rec_k}$$

Model level performance can be computed by averaging the measurements across all classes $\mathbf{k} \in \mathbf{K}$.

- **Micro average:** Calculates metrics globally by counting the total TP, FP and FN.
- **Macro average:** Calculates metrics for each label and find the unweighted mean (regardless the proportion of each class).
- **Weighted average:** Calculate metrics for each label and find the average weighted by support (# of true examples for each class).

What if we have more than two classes?

Multiclass Example

Confusion Matrix		Predicted Class			Support
		Class=0	Class=1	Class=2	
Actual Class	Class=0	512	12	22	546
	Class=1	2	77	13	92
	Class=2	36	59	831	926

$$\text{Prec}_0 = \frac{512}{550} = 93.1\%$$

$$\text{Prec}_1 = \frac{77}{148} = 52.0\%$$

$$\text{Prec}_2 = \frac{831}{866} = 96.0\%$$

$$\text{Prec}_{\text{macro}} = \frac{93.1\% + 52.0\% + 96.0\%}{3} = 80.4\%$$

$$\text{Prec}_{\text{micro}} = \frac{512 + 77 + 831}{550 + 148 + 866} = 90.8\% = \text{Acc}$$

$$\text{Prec}_{\text{weighted}} = \frac{93.1\% * 546 + 52\% * 92 + 96\% * 926}{546 + 92 + 926} = 92.4\%$$

$$\text{Acc} = \frac{512 + 77 + 831}{546 + 92 + 926} = 90.8\%$$

$$\text{Rec}_0 = \frac{512}{546} = 93.8\%$$

$$\text{Rec}_1 = \frac{77}{92} = 83.7\%$$

$$\text{Rec}_2 = \frac{831}{926} = 89.7\%$$

$$\mathbf{Rec_{macro}} = \frac{\mathbf{1}}{\boldsymbol{K}} \sum_{k \in K} \boldsymbol{Rec}_{\text{k}} = \frac{93.8\% + 83.7\% + 89.7\%}{3} = 89.1\%$$

$$\mathbf{Rec_{micro}} = \frac{\sum_{k \in K} \boldsymbol{TP}_k}{\sum_{k \in K} \boldsymbol{TP}_k + \boldsymbol{FN}_k} = \frac{512 + 77 + 831}{546 + 92 + 926} = 90.8\% = \text{Acc}$$

$$\mathbf{Rec_{weighted}} = \frac{\sum_{k \in K} \boldsymbol{Rec}_k * \boldsymbol{Sup}_k}{\boldsymbol{N}} = \frac{93.8\% * 546 + 83.7\% * 92 + 89.7 * 926}{546 + 92 + 926} = 90.8\% = \text{Acc}$$

What if we have more than two classes?

Multi-label (each example can be assigned more than one label)

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}$$

Where $Y_i \in \{0,1\}^{|K|}$ is the true label set (the set of true labels) and $\hat{Y}_i \in \{0,1\}^{|K|}$ is the set of predicted labels for instance i. It measures how much the predicted and true labels overlap.

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \hat{Y}_i|}{|Y_i|}$$

Measures the fraction of relevant labels that were predicted correctly.

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \hat{Y}_i|}{|\hat{Y}_i|}$$

Measures the fraction of relevant labels among the predicted labels.

What if we have more than two classes?

Multi-label (each example can be assigned more than one label)

$$\mathrm{F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \times |Y_i \cap \hat{Y}_i|}{|Y_i| + |\hat{Y}_i|}$$

Harmonic mean of precision and recall over all instances.

Example:

$$\mathbf{Y} = \{\{0,1,1\},\{1,0,1\},\{1,1,1\},\{1,0,1\}\}$$
$$\hat{\mathbf{Y}} = \{\{0,0,1\},\{1,0,1\},\{1,1,0\},\{1,1,0\}\}$$

$$\mathbf{F1} = \frac{1}{4}\cdot\left[\frac{2\cdot1}{3}+\frac{2\cdot2}{4}+\frac{2\cdot2}{5}+\frac{2\cdot1}{4}\right] = \mathbf{74.2\%}$$

Evaluation Metrics – Summary

Other Evaluation Metrics

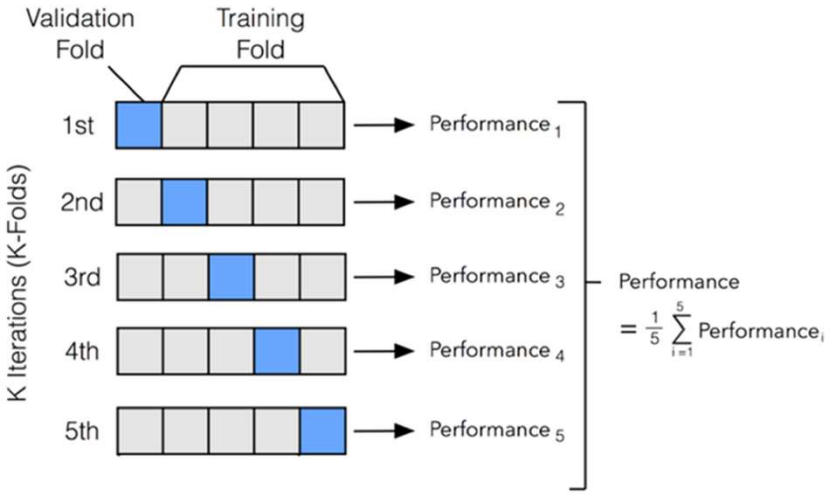
Confusion Matrix		Predicted Class: Positive (Class=1)	Predicted Class: Negative (Class=0)		
Actual Class	Positive (Class=1)	TP	FN	*Recall* or ***Sensitivity or True Positive Rate*** $\frac{TP}{TP + FN}$	***False Negative Rate*** $\frac{FN}{TP + FN}$
	Negative (Class=0)	FP	TN	***Specificity or True Negative Rate*** $\frac{TN}{FP + TN}$	***False Positive Rate*** $\frac{FP}{FP + TN}$
		Precision $\frac{TP}{TP + FP}$	***Negative Predictive Value*** $\frac{TN}{FN + TN}$	***Accuracy*** $\frac{TP + TN}{TP + FP + FN + TN}$	

Model Validation

Model Validation and Resampling Methods

- **Cross-validation:** A resampling technique used to evaluate model performance by partitioning the data into multiple subsets (folds) and rotating the training and testing phases.

  - Cross-validation divides the data into k folds.
  - The model is trained on k-1 folds and tested on the remaining fold.
  - This process is repeated k times, and the final performance is the average across all folds.



Model Validation

Model Validation and Resampling Methods

- **Hold-out:** A simple validation method where the dataset is split into two or more distinct sets (typically training and test sets) to evaluate the model's performance.

- **Bootstrap:** A statistical resampling method used to estimate the accuracy of a model by generating multiple training datasets through sampling with replacement from the original dataset.

ROC

Receiver Operating Characteristics (ROC)

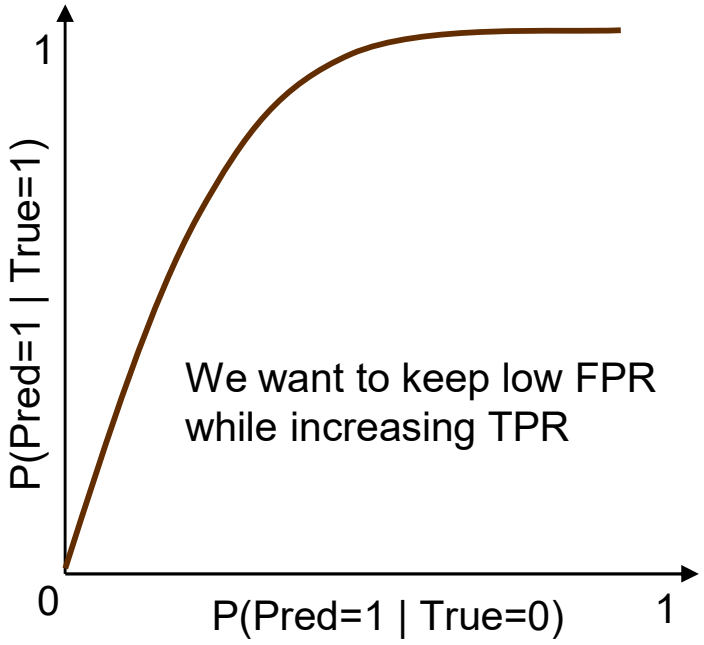
- **Threshold Independence: ROC** shows how the model performs across all possible decision thresholds (accuracy, precision and recall depend on a specific threshold).

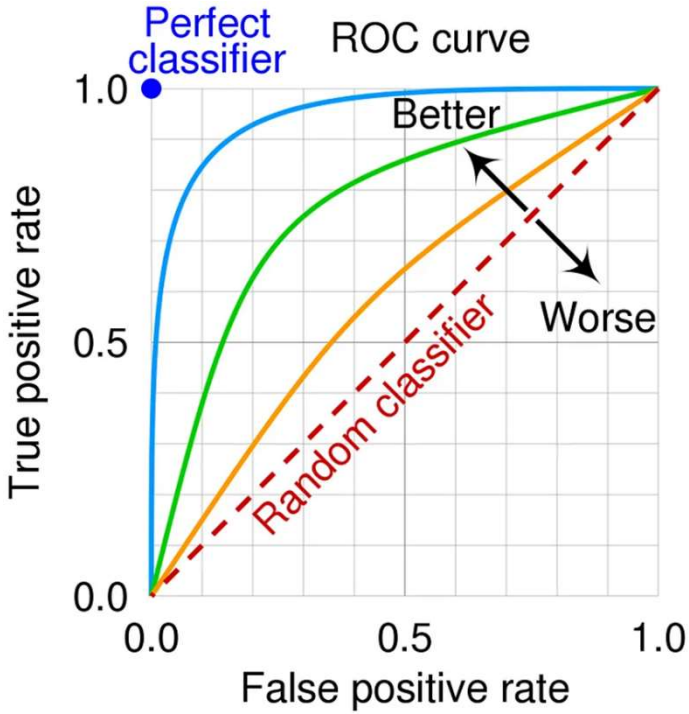
- **Trade-offs:** ROC curve highlights the trade-off between True Positive Rate (Recall) and False Positive Rate. This helps understand how the model balances detecting true positives while avoiding false positives.
  - Useful when getting a positive wrong costs more than getting a negative wrong (or vice versa).

- **Class Imbalance:** Useful in cases of class imbalance, where accuracy can be misleading. It provides a clearer picture of how well the model discriminates between classes, even when one class is much more frequent.

ROC

Receiver Operating Characteristics (ROC)





ROC

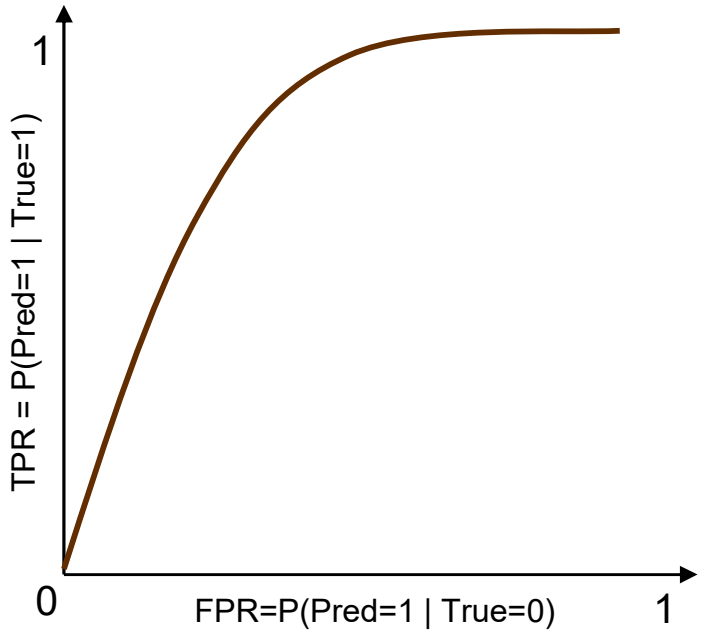
Algorithm for Creating ROC Curve

Step 1: Sort test set by predictions P(Class=1) in decreasing order.

Step 2: Set thresholds along these examples (it can be based on equal area/width bins or example by example)

Step 3: Compute TPR & FPR for each threshold of Step 2

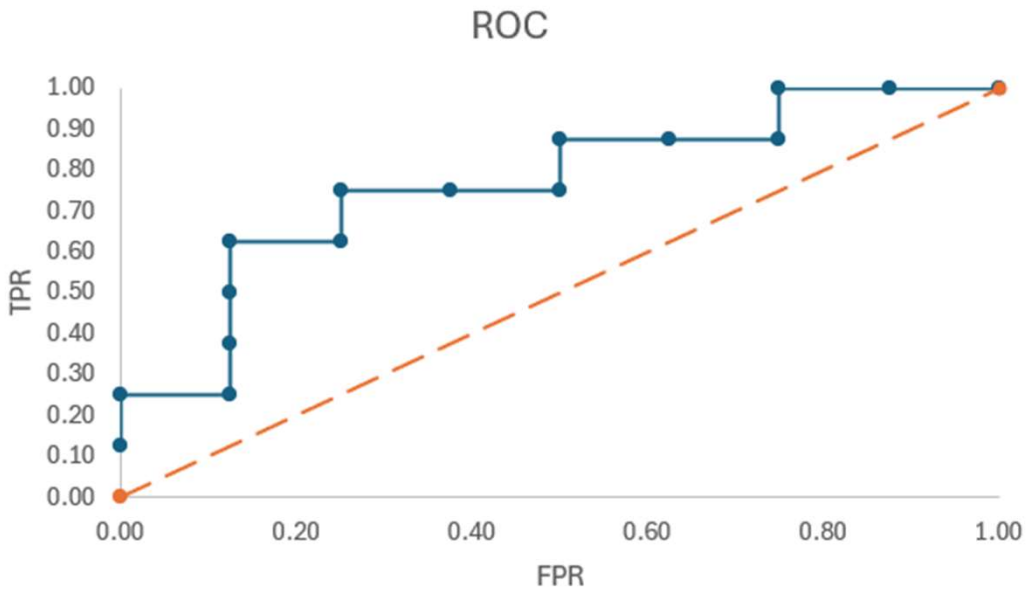
Step 4: Connect the dots in the plot



ROC

Example of ROC Plot

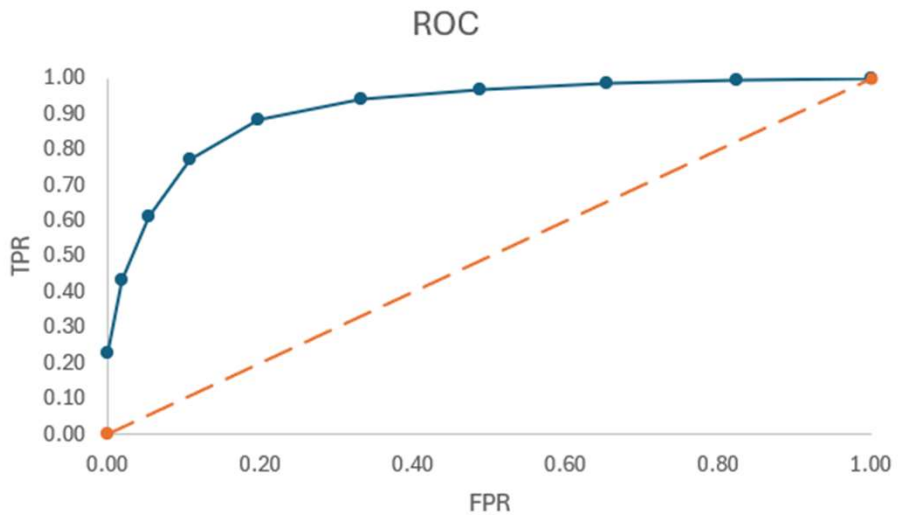
Ex.ID	Predicted	Actual	TPR	FPR
12	0.97	1	1/8 = 0.13	0/8 = 0.00
7	0.96	1	2/8 = 0.25	0/8 = 0.00
15	0.89	0	2/8 = 0.25	1/8 = 0.13
11	0.72	1	3/8 = 0.38	1/8 = 0.13
1	0.68	1	4/8 = 0.50	1/8 = 0.13
5	0.67	1	5/8 = 0.63	1/8 = 0.13
4	0.54	0	5/8 = 0.63	2/8 = 0.25
2	0.51	1	6/8 = 0.75	2/8 = 0.25
8	0.51	0	6/8 = 0.75	3/8 = 0.38
3	0.50	0	6/8 = 0.75	4/8 = 0.50
13	0.48	1	7/8 = 0.88	4/8 = 0.50
6	0.47	0	7/8 = 0.88	5/8 = 0.63
10	0.42	0	7/8 = 0.88	6/8 = 0.75
14	0.17	1	8/8 = 1.00	6/8 = 0.75
9	0.09	0	8/8 = 1.00	7/8 = 0.88
16	0.03	0	8/8 = 1.00	8/8 = 1.00



ROC

More Examples

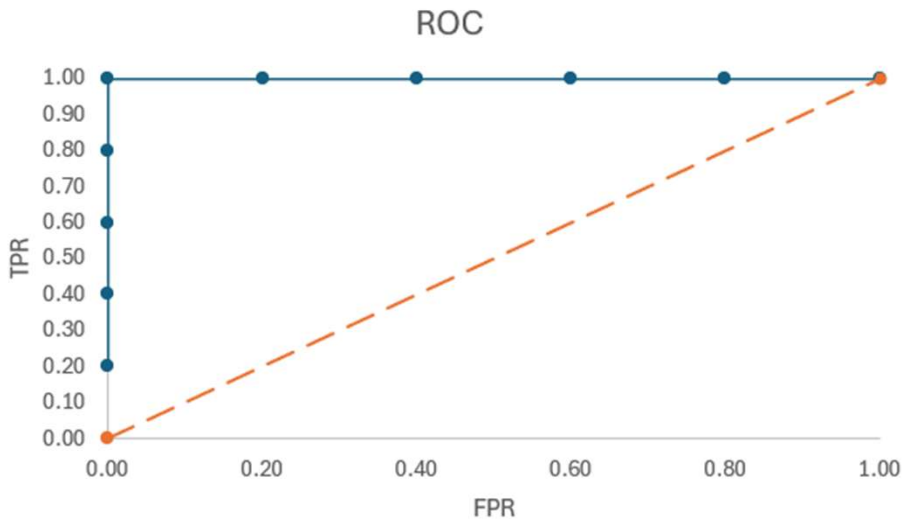
Prob(Y=1)	Distribution	count(True Y=1)	count(True Y=0)	TPR	FPR
[90%, 100%]	1,000	1,000	-	0.23	0.00
[80%, 90%)	1,000	900	100	0.43	0.02
[70%, 80%)	1,000	800	200	0.61	0.05
[60%, 70%)	1,000	700	300	0.77	0.11
[50%, 60%)	1,000	500	500	0.88	0.20
[40%, 50%)	1,000	250	750	0.94	0.33
[30%, 40%)	1,000	120	880	0.97	0.49
[20%, 30%)	1,000	80	920	0.99	0.65
[10%, 20%)	1,000	40	960	1.00	0.82
[0%, 10%)	1,000	20	980	1.00	1.00
Total	10,000	4,410	5,590		



ROC

More Examples

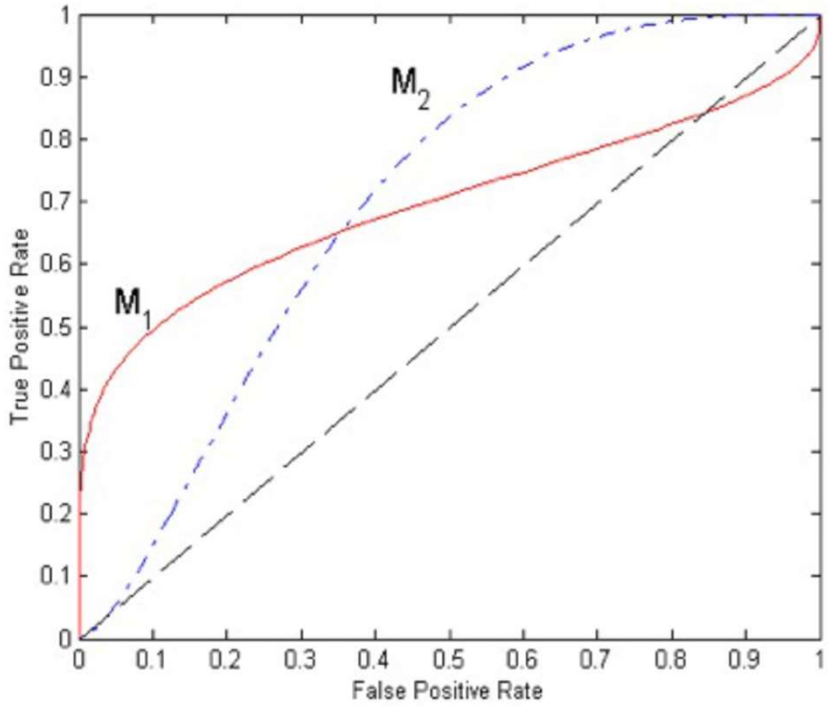
Prob(Y=1)	Distribution	count(True Y=1)	count(True Y=0)	TPR	FPR
[90%, 100%]	1,000	1,000	-	0.20	0.00
[80%, 90%)	1,000	1,000	-	0.40	0.00
[70%, 80%)	1,000	1,000	-	0.60	0.00
[60%, 70%)	1,000	1,000	-	0.80	0.00
[50%, 60%)	1,000	1,000	-	1.00	0.00
[40%, 50%)	1,000	-	1,000	1.00	0.20
[30%, 40%)	1,000	-	1,000	1.00	0.40
[20%, 30%)	1,000	-	1,000	1.00	0.60
[10%, 20%)	1,000	-	1,000	1.00	0.80
[0%, 10%)	1,000	-	1,000	1.00	1.00
Total	10,000	5,000	5,000		



ROC

ROC – Comparing Models

- Different algorithms can work better in different parts of ROC space.
- **M1** is better for small FPR
- **M2** is better for large FPR



Area Under ROC Curve

AUC – Area Under Curve

- Comprehensive measure: AUC evaluates model performance across all thresholds, not just one fixed decision threshold.
- Reflects the model's ability to distinguish between classes.
- Provides a single value for easy comparison of models.