Chapter 8. Classification: Evaluation

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Paper Organization

Suppose we are writing a paper: We propose a "novel" model, the Naïve Bayes model, to address the problem of classification.

1. Introduction

- **(1) Why do we study** the problem of classification? **Applications** such as predicting "play tennis or not" given weather data.
- **(2) Existing models:** Decision Trees. Issue: Ignoring useful though not the "best" features.
- **(3) Major challenges:** Lack of theoretical foundation on considering distributions of all the attributes in massive training instances.
- **(4) Idea:** Borrow Bayes Theorem. **Proposed method:** $P(H|X) = P(X|H)P(H)/P(X)$. **Why it works** (and work better than DTs)?
- **(5) Itemize major contributions**

Paper Organization (cont.)

2. Related Work

Survey two or three fields of work relevant to your paper on **different aspects**: (1) Classification models (e.g., Decision Trees), (2) Studies using Bayes Theorem

3. Problem Definition

Given … training(instances, features, labels) and testing(instances, features), **find** … testing(labels)

4. Proposed Model

and Algorithm (components and pseudo code)

5. Experiments (to demonstrate your itemized contributions)

6. Conclusions/Discussions (followed with Acks and Refs)

“Experiments” Organization

[Questions to answer in this section…]

Q1: Does the proposed method perform effectively on … ?

Q2: …?

5.1 Datasets

5.2 Experimental settings

Baselines (ID3, C4.5, CART, etc.)

Parameter settings (Normalization? Laplacian correction?)

Validation settings (training, testing …) !!!

Evaluation metrics (accuracy, precision, recall …) !!!

5.3 Binary Classification (Q1)

5.3.1 Quantitative analysis

5.3.2 Qualitative analysis (case studies)

5.4 … (Q2)

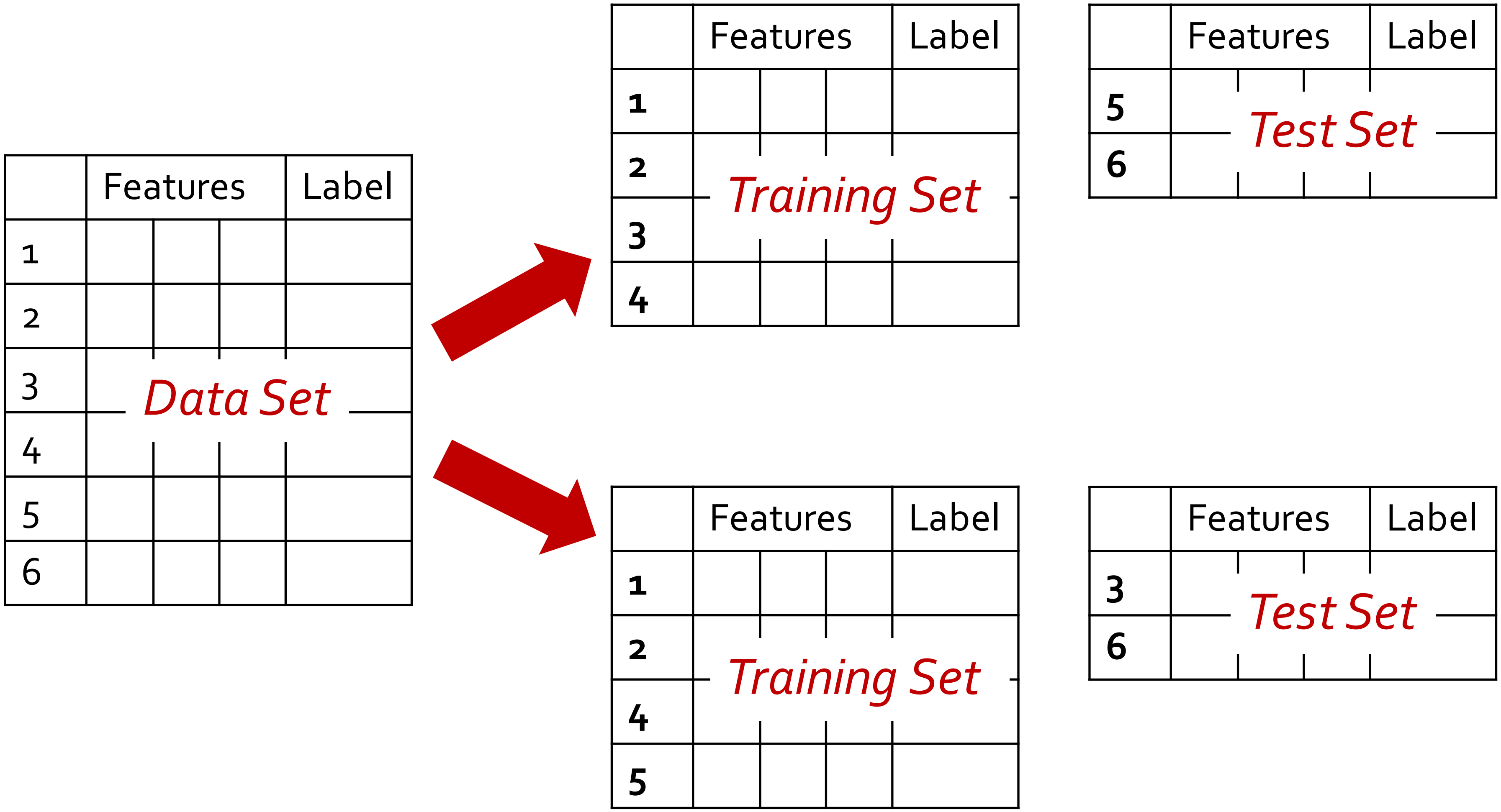
Today's Lecture: Evaluation

- Validation Settings
  - Hold-out validation method
  - Cross-validation methods (+ Stratified)
    - k-fold cross-validation
    - Leave-one-out validation
- Evaluation Metrics
  - Confusion matrix
  - Accuracy, Error rate
  - Sensitivity, Specificity
  - Precision, Recall, F measure, G measure
  - ROC curves, Area Under the Curve (AUC), Precision-Recall Curve
  - Precision@K, Average precision
  - Mean absolute error (MAE), Root mean squared error (RMSE)
  - Ranking-based measures (Kendall's tau, Spearman's rho)

Holdout Validation

- Given data is *randomly* partitioned into *two independent* sets
  - Training set (e.g., 2/3, 3/5, 4/5) for model construction
  - Test set (e.g., 1/3, 2/5, 1/5) for accuracy estimation
- Repeat holdout k times, accuracy = *avg.* of the accuracies obtained
  - Standard deviation?

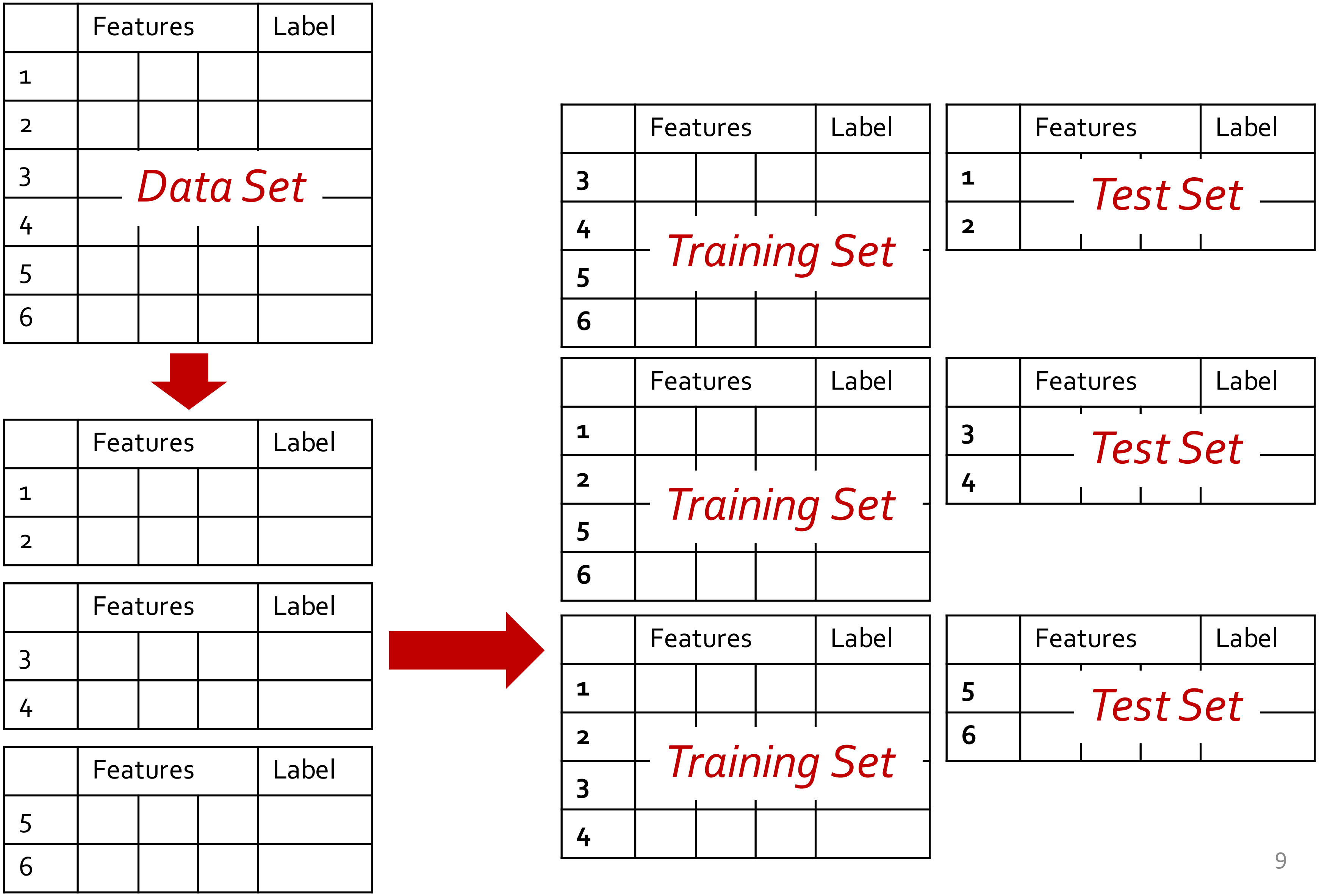
Holdout Validation: Example (k=2)



k-fold Cross Validation

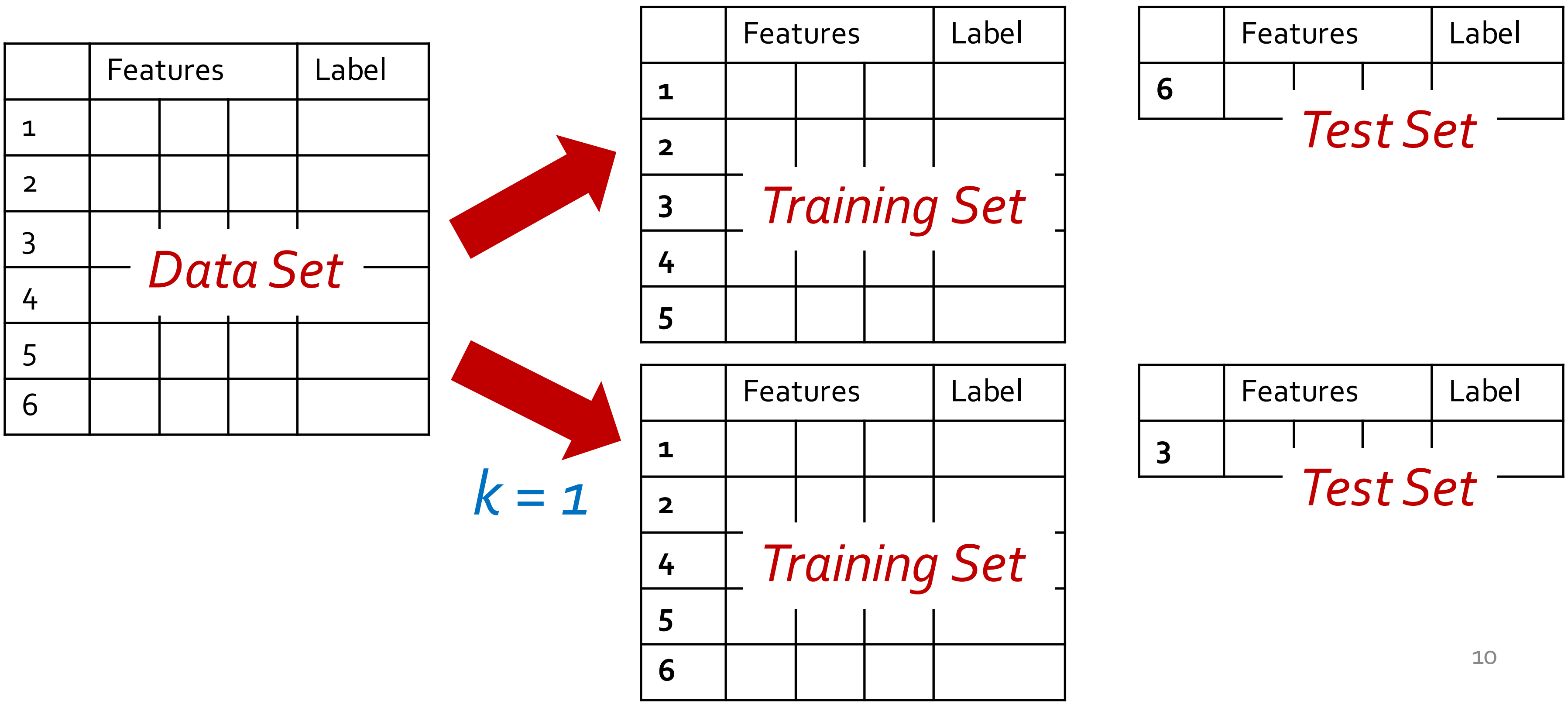
- Given data D is *randomly* partitioned into k *mutually exclusive* subsets D_i ($i = 1, \ldots, k$), each approximately *equal size* $|D_i|$
  - At i-th iteration ($i = 1, \ldots, k$), use D_i as test set for accuracy estimation and others $D_1 \cup \ldots \cup D_{i-1} \cup D_{i+1} \cup \ldots \cup D_k$ as training set for model construction
  - k=10 is the most popular

k-fold Cross Validation: Example (*k*=3)



Leave-one (*k*)-out Validation

- Given *small-sized* data is randomly partitioned into a training set and a test set. The size of the test set is *k*, i.e., number of test tuples.



Stratified Cross-Validation

- Folds are stratified so that class distribution in each fold is approximately the same as that in initial data.

	Features			Label (Win/Loss/Draw)
1 … **500**				Win
501 … **800**				Loss
801 … **850**				Draw

10-fold stratified cross-validation

	Features			Label
*50				Win
*30				Loss
*5				Draw

	Features			Label
*50				Win
*30				Loss
*5				Draw

⋮

	Features			Label
*50				Win
*30				Loss
*5				Draw

Check List: Validation Settings

- ❑ Holdout validation
- ❑ *k*-fold cross-validation
- ❑ Leave-one-out validation
- ❑ Stratified cross-validation

Metrics: (1) Confusion Matrix

- Given m classes, an entry, $CM_{i,j}$ in a confusion matrix ***CM*** indicates the number of tuples in class i (**actual class**) that were labeled by the classifier as class j (**predicted class**)
  - May have extra rows/columns to provide totals

Actual class\Predicted class	C	¬ C
C	**True Positives (TP)**	**False Negatives (FN)**
¬ C	**False Positives (FP)**	**True Negatives (TN)**

Actual class\Predicted class	game_result = "win"	game_result = "loss"	Total
game_result = "win"	**6,954**	**46**	7,000
game_result = "loss"	**412**	**2,588**	3,000
Total	7,366	2,634	10,000

Metrics: (2) Accuracy, Error Rate

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

$$Accuracy = (TP + TN)/All$$

- **Error rate:** $1 - accuracy$, or

$$Error\ rate = (FP + FN)/All$$

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

A\P	C	¬C	
C	**1000**	**1800**	**2800**
¬C	**1200**	**1000**	**2200**
	2200	**2800**	**5000**

Accuracy = *2000/5000 = 0.4*
Error rate = *3000/5000 = 0.6*

Example: C = (game_result = "win")

Metrics: (3) Sensitivity, Specificity

- Class Imbalance Problem:
  - One class may be rare, e.g. fraud, or HIV-positive
    - N >> P
  - Significant majority of the negative class and minority of the positive class
    - Then TN could be high and *Accuracy = (TP + TN)/All* could be high

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

A\P	C	¬C	
C	**1000**	**1800**	**2800**
¬C	**1200**	**96000**	**97200**
	2200	**97800**	**100000**

Accuracy = 97000/100000 = 0.97

Error rate = 3000/100000 = 0.03

Example: C = (cancer= "yes")

Metrics: (3) Sensitivity, Specificity

- Sensitivity: True Positive recognition rate

$$Sensitivity = TP/P$$

- Specificity: True Negative recognition rate

$$Specificity = TN/N$$

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

A\P	C	¬C	
C	**1000**	**1800**	**2800**
¬C	**1200**	**96000**	**97200**
	2200	**97800**	**100000**

Sensitivity = 1000/2800 = 0.357

Specificity = 96000/97800 =0.982

Example: C = (cancer= "yes")

Metrics: (4) Precision, Recall

- **Precision**, or exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$Precision = TP/(TP+FP) = TP/P'$$

- **Recall**, or completeness: what % of positive tuples did the classifier label as positive?

$$Recall = TP/(TP+FN) = TP/P,$$ *the same as sensitivity*

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P′	**N′**	**All**

A\P	C	¬C	
C	**1000**	**1800**	**2800**
¬C	**1200**	**96000**	**97200**
	2200	**97800**	**100000**

Precision = 1000/2200 = 0.455

Recall = 1000/2800 = 0.357

Example: C = (cancer= "yes")

Metrics: (4') F Measure

- **F measure**, or F-score: harmonic mean of precision and recall
  - In general, it is the weighted measure of precision and recall, also called Fβ-score:

$$F = \frac{1}{\alpha \cdot \frac{1}{P} + (1-\alpha) \cdot \frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

Assigning β^2 times as much weight to recall as to precision

$\alpha = 1/(1+\beta^2)$

- F1-measure (balanced F-measure)
  - That is, when $\beta = 1$, $F_1 = \frac{2PR}{P+R}$
- Other two F meausres
  - F_2
  - $F_{0.5}$

A\P	C	¬C	
C	**1000**	**1800**	**2800**
¬C	**1200**	**96000**	**97200**
	2200	**97800**	**100000**

Precision = 1000/2200 = 0.455
Recall = 1000/2800 = 0.357
F_1 = 0.400
F_2 = 0.373
$F_{0.5}$ = 0.431

Metrics: (4'') G Measure

- **G measure**, or Fowlkes-Mallows Index, is the geometric mean of precision and recall:

$$G = \sqrt{PR}$$

A\P	C	¬C	
C	**1000**	**1800**	**2800**
¬C	**1200**	**96000**	**97200**
	2200	**97800**	**100000**

Precision = 1000/2200
= 0.455
Recall = 1000/2800
= 0.357
G = 0.403

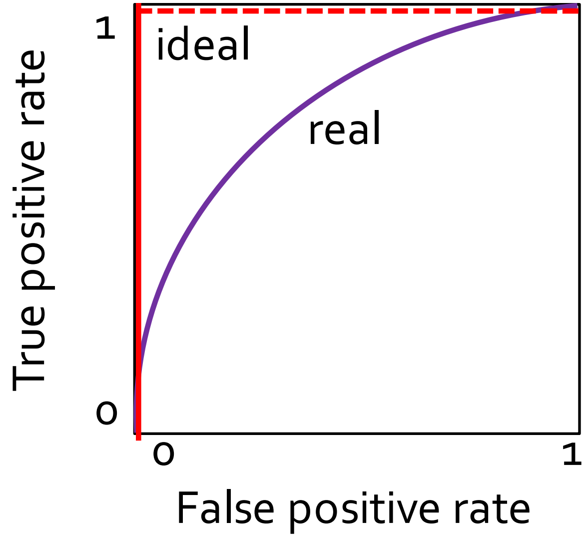
Metrics: (5) ROC Curve

- What is it?
  - ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models.
  - Originated from signal detection theory: Developed in 1950s to analyze noisy signals.
  - Shows the **trade-off between the true positive rate and the false positive rate**.
    - TPR = TP/P = TP/(TP+FN)
    - FPR = FP/N = FP/(FP+TN)

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P’	**N’**	**All**

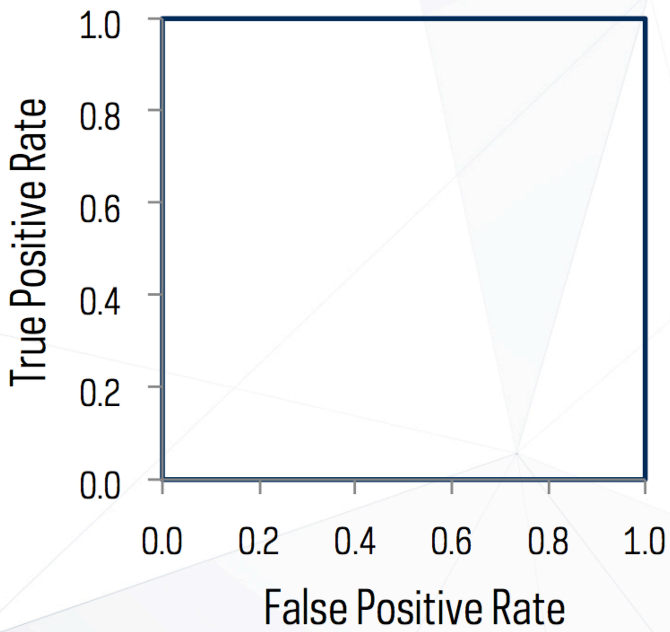
Metrics: (5) ROC Curve

- How to plot?
  - Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
  - **Vertical axis represents the true positive rate**
  - **Horizontal axis represents the false positive rate**



TPR = TP/P = TP/(TP+FN)
FPR = FP/N = FP/(FP+TN)

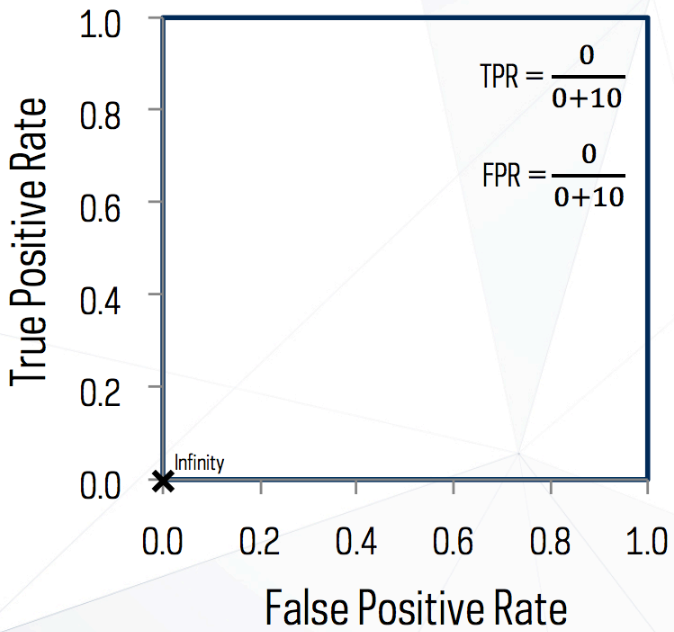
Generating ROC Curves



Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves (cont.)

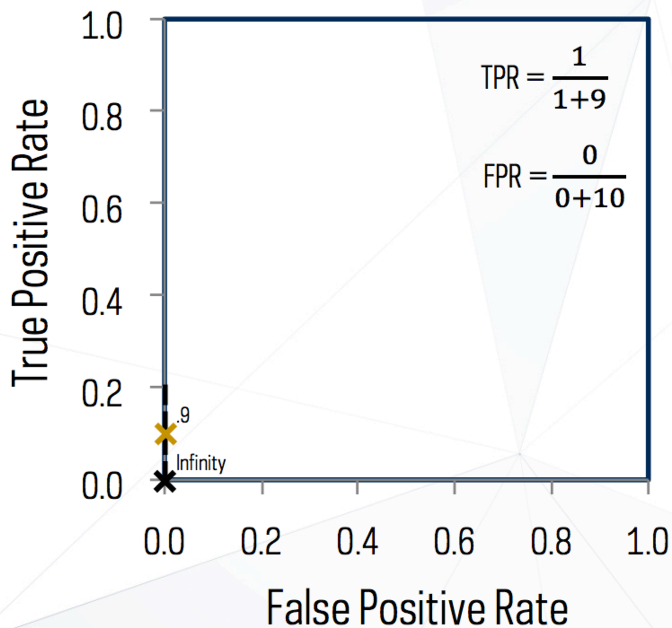
TPR = TP/P = TP/(TP+FN)
FPR = FP/N = FP/(FP+TN)



Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves (cont.)

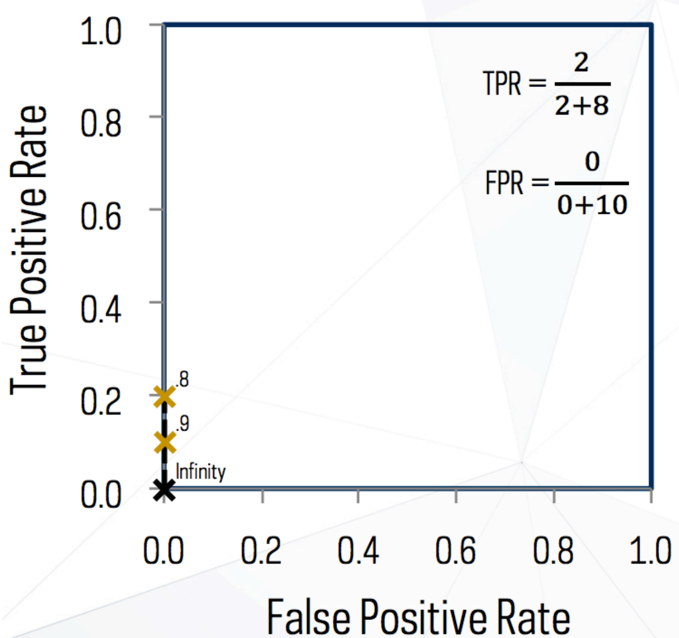
TPR = TP/P = TP/(TP+FN)
FPR = FP/N = FP/(FP+TN)



Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves (cont.)

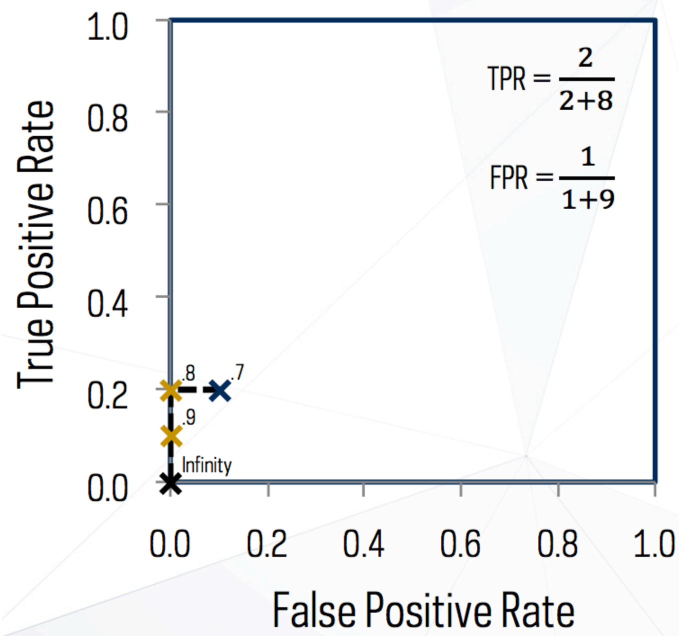
TPR = TP/P = TP/(TP+FN)
FPR = FP/N = FP/(FP+TN)



Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves (cont.)

TPR = TP/P = TP/(TP+FN)
FPR = FP/N = FP/(FP+TN)

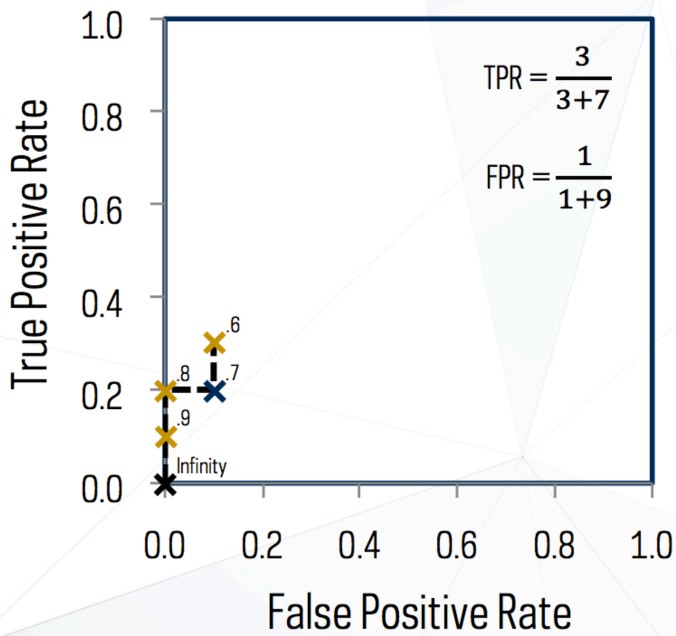


Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves (cont.)

TPR = TP/P = TP/(TP+FN)

FPR = FP/N = FP/(FP+TN)

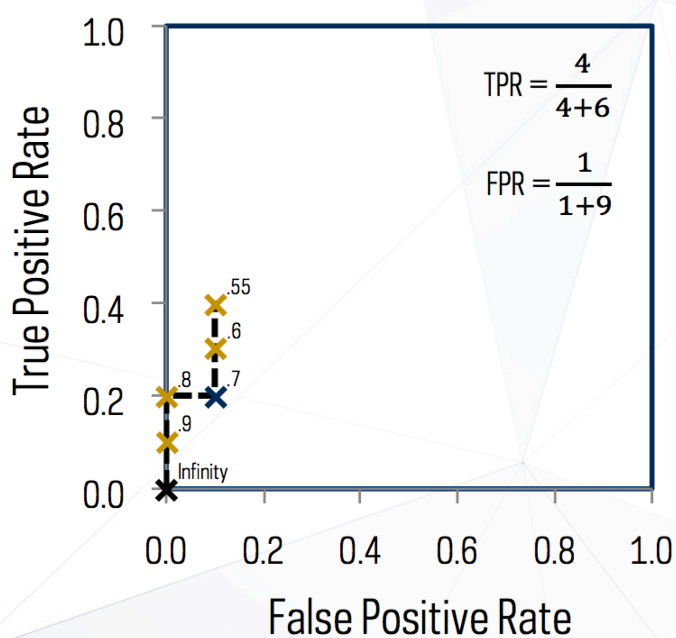


Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves (cont.)

TPR = TP/P = TP/(TP+FN)

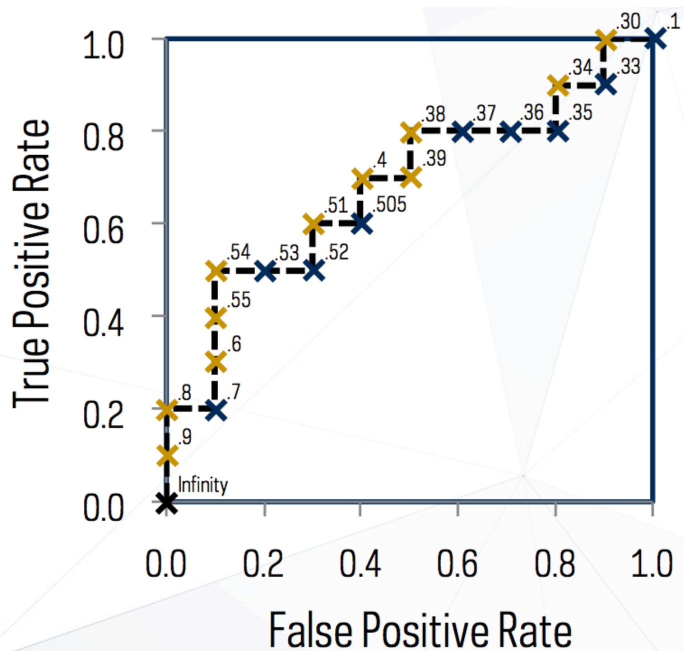
FPR = FP/N = FP/(FP+TN)



Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Generating ROC Curves: Final

TPR = TP/P = TP/(TP+FN)
FPR = FP/N = FP/(FP+TN)

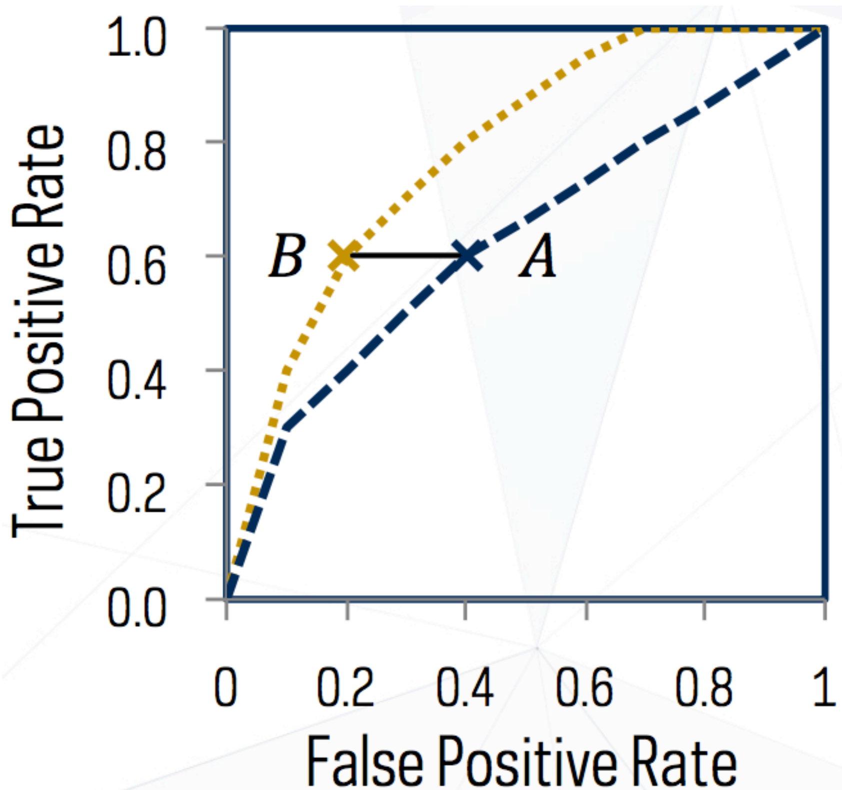


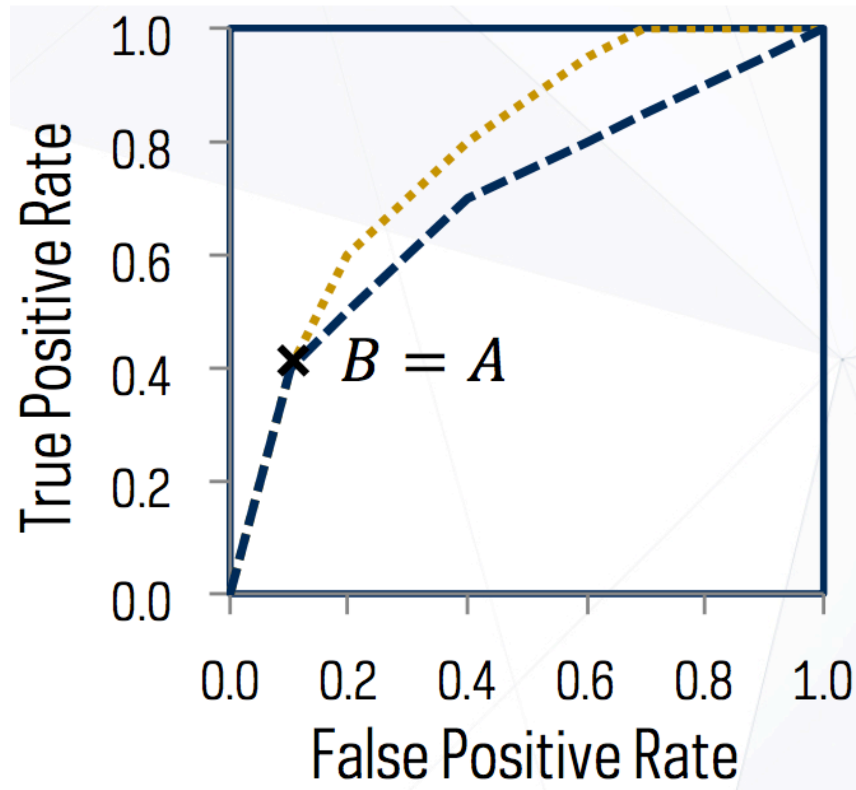
Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Comparing Classifiers in ROC Space

TPR = TP/P = TP/(TP+FN)

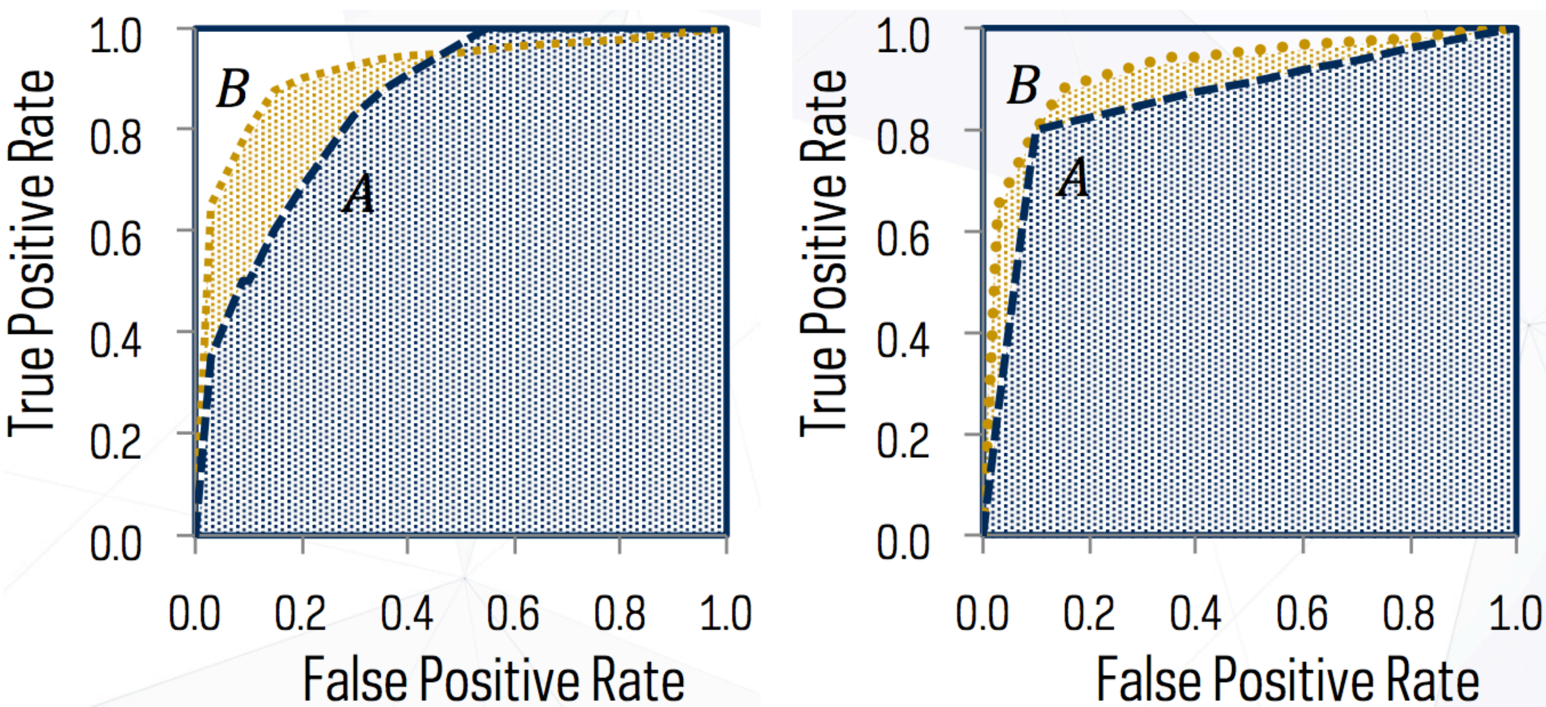
FPR = FP/N = FP/(FP+TN)





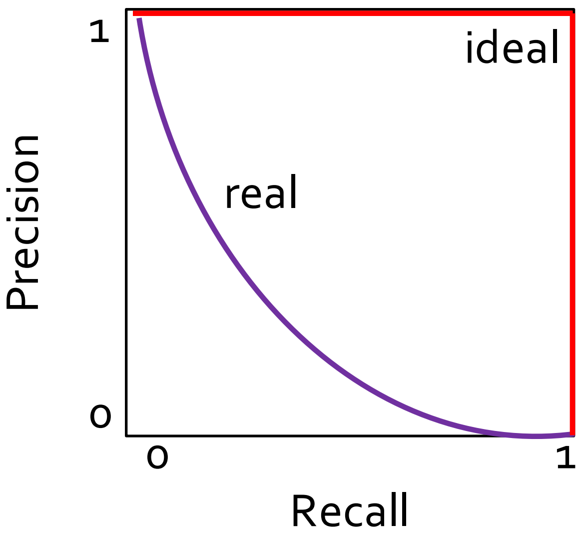
Metrics: (5) AUC

- The **area under the ROC curve** (AUC) is a measure of the accuracy of the model
  - Summarizes model performance across all possible thresholds
  - A model with perfect accuracy will have an area of 1.0



Metrics: (5′) Precision-Recall Curve

- How to plot?
  - Vertical axis represents **Precision**
  - Horizontal axis represents **Recall**



Precision = TP/P′ = TP/(TP+FP)
Recall = TP/P = TP/(TP+FN)

Metrics: (6) Precision@K

- Precision@K
  - P@1 = 1.0
  - P@3 = 0.67
  - P@5 = 0.8
  - P@10 = 0.6

Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Metrics: (6) Average Precision

- Average Precision

K	P@K	K	P@K
1	1.00	11	0.64
2	1.00	12	0.58
3	0.67	13	0.62
4	0.75	14	0.57
5	0.80	15	0.53
6	0.83	16	0.50
7	0.71	17	0.53
8	0.63	18	0.50
9	0.67	19	0.53
10	0.60	20	0.50

Given |P| = |TP+FN| = 10,

Average Precision (A.P.)

$$= \frac{\sum_k Precision@K}{|P|}$$

$$= \frac{1+1+0.75+0.80+0.83+0.67+0.64+0.62+0.53+0.53}{10}$$

= 0.74

Q: When A.P. = 1.0 (maximum)?
What is the minimum of A.P.?

Metrics: (7) Errors

- Mean Absolute Error (MAE)

$$\frac{\sum_i |s_i - c_i|}{n} = \frac{|0.9 - 1.0| + \cdots + |0.1 - 0.0|}{20}$$

Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Metrics: (7) Errors

- Root mean squared error (RMSE)

$$\sqrt{\frac{\sum_i (s_i - c_i)^2}{n}} = \sqrt{\frac{(0.9 - 1.0)^2 + \cdots + (0.1 - 0.0)^2}{20}}$$

Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Metrics: (8) Ranking-based Measures

- Rank correlation coefficients
  - Kendal's tau

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{n(n-1)/2}.$$

  - Spearman's rho

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

where d_i is the difference between two ranks

Instance	Class	Score	Instance	Class	Score
1	positive	.9	11	positive	.4
2	positive	.8	12	negative	.39
3	negative	.7	13	positive	.38
4	positive	.6	14	negative	.37
5	positive	.55	15	negative	.36
6	positive	.54	16	negative	.35
7	negative	.53	17	positive	.34
8	negative	.52	18	negative	.33
9	positive	.51	19	positive	.30
10	negative	.505	20	negative	.1

Check List: Evaluation Metrics

- ❑ Confusion matrix
- ❑ Accuracy, Error rate
- ❑ Sensitivity, Specificity
- ❑ Precision, Recall, F measure, G measure
- ❑ ROC curve, Area Under the Curve (AUC), Precision-Recall Curve
- ❑ Precision@K, Average precision
- ❑ Mean absolute error (MAE), Root mean squared error (RMSE)
- ❑ Ranking-based measures (Kendall's tau and Spearman's rho)

References

- C. Apte and S. Weiss. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13, 1997
- P. K. Chan and S. J. Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. KDD'95
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman & Hall, 1990.
- R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2ed. John Wiley, 2001
- U. M. Fayyad. Branching on attribute values in decision tree generation. AAAI'94.
- Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. J. Computer and System Sciences, 1997.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest: A framework for fast decision tree construction of large datasets. VLDB'98.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, BOAT -- Optimistic Decision Tree Construction. SIGMOD'99.
- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001.
- T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 2000

References (cont.)

- J. Magidson. The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection. In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, Blackwell Business, 1994
- M. Mehta, R. Agrawal, and J. Rissanen. SLIQ : A fast scalable classifier for data mining. EDBT'96
- T. M. Mitchell. Machine Learning. McGraw Hill, 1997
- S. K. Murthy, Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, Data Mining and Knowledge Discovery 2(4): 345-389, 1998
- J. R. Quinlan. Induction of decision trees. Machine Learning, 1:81-106, 1986.
- J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
- J. R. Quinlan. Bagging, boosting, and c4.5. AAAI'96.
- R. Rastogi and K. Shim. **Public: A decision tree classifier that integrates building and pruning**. VLDB'98
- J. Shafer, R. Agrawal, and M. Mehta. **SPRINT : A scalable parallel classifier for data mining**. VLDB'96
- J. W. Shavlik and T. G. Dietterich. **Readings in Machine Learning**. Morgan Kaufmann, 1990
- P. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining**. Addison Wesley, 2005
- S. M. Weiss and C. A. Kulikowski. **Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems**. Morgan Kaufman, 1991
- S. M. Weiss and N. Indurkhya. **Predictive Data Mining**. Morgan Kaufmann, 1997
- I. H. Witten and E. Frank. **Data Mining: Practical Machine Learning Tools and Techniques**, 2ed. Morgan Kaufmann, 2005