Artificial Intelligence for Medical Data with Python

10 SAMPLE SLIDES

9 th session – Evaluation Metrics of Prediction Models

UNIVERSITY OF THE AEGEAN

SCHOOL OF ENGINEERING

DEPARTMENT OF INFORMATION AND COMMUNICATION SYSTEMS ENGINEERING

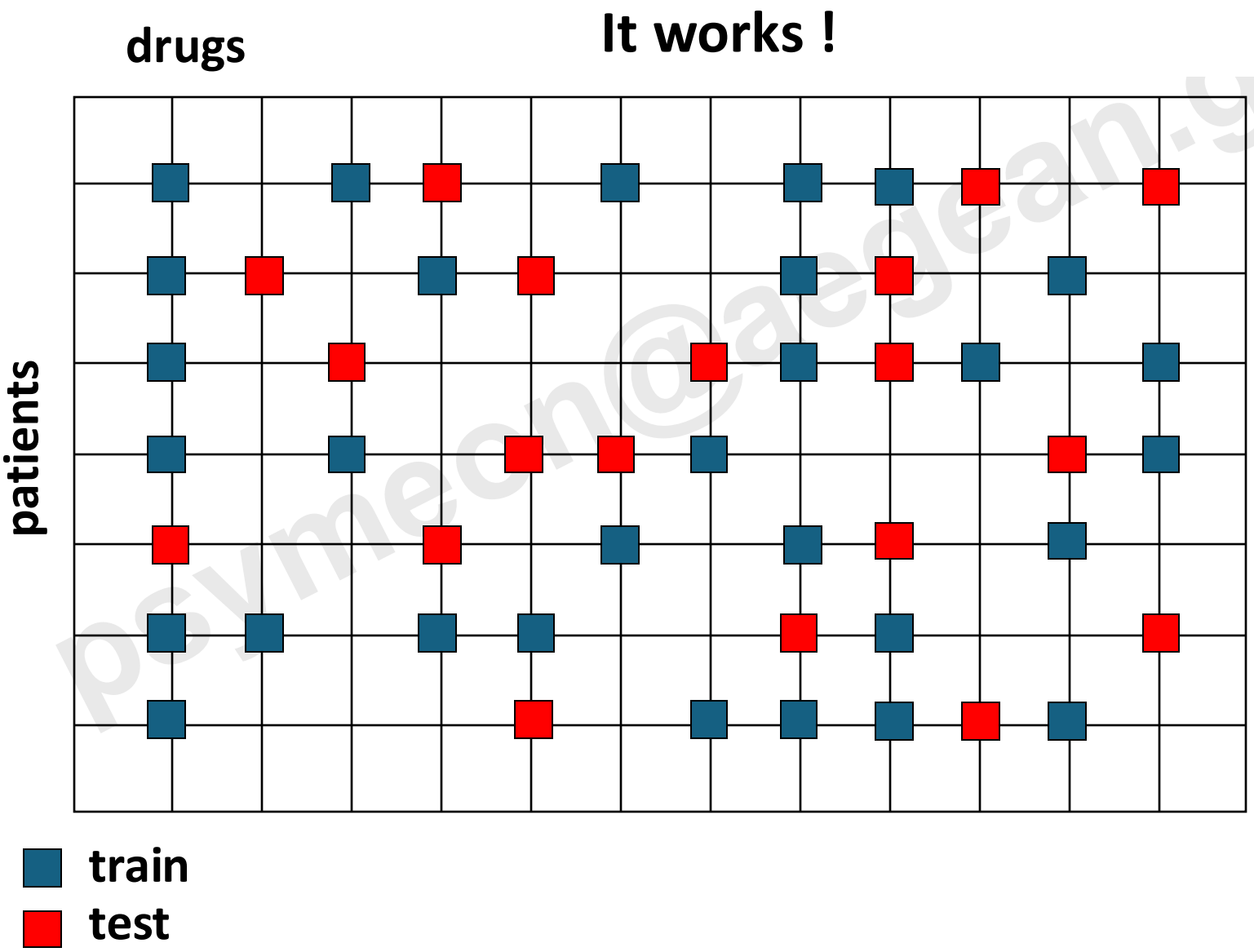
Presenter: Panagiotis Symeonidis

Associate Professor

http://panagiotissymeonidis.com

psymeon@aegean.gr

Splitting the data



Accuracy: Comparing Values

- Measure **how close** the predicted ratings are to the true values

- **Predictive accuracy (rating): Mean Absolute Error** (**MAE**), r^*_{ui} is the **predicted** value and r_{ui} is the **true** one:

$$MAE(r^*) = \frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} |r^*_{ui} - r_{ui}|$$

Variation : *penalize* larger errors - Mean Squared Error (average the square of the differences), Root Mean Squared Error RMSE:

$$RMSE(r^*) = \sqrt{\frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} \left(r^*_{ui} - r_{ui}\right)^2}$$

Performance of binary classification

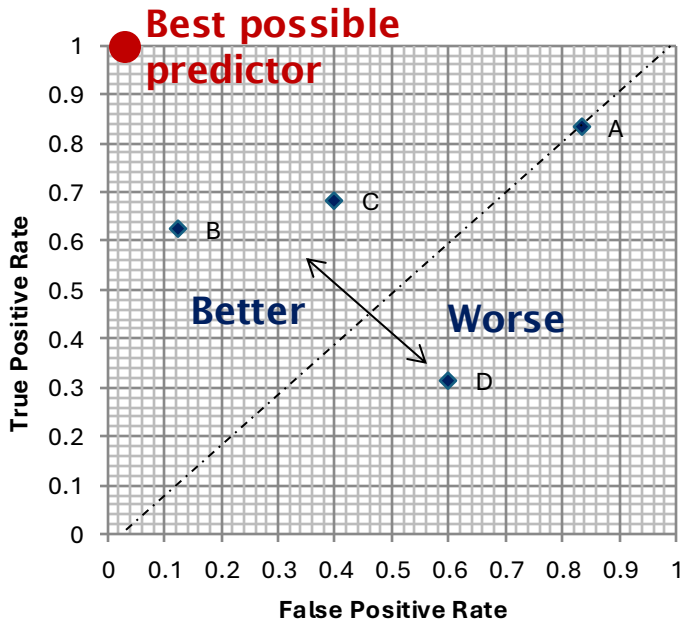
- Sensitivity or True Positive Rate (TPR), also known as Recall:
  - $TPR = \frac{TP}{TP+FN}$
  - Not recommending items that would be actually good recommendations (False Negatives or Type II Error)
- Specificity or True Negative Rate:
  - $TNR = \frac{TN}{FP+TN} = 1\text{- }FPR$
- False Positive Rate (FPR) or Fall-Out
  - $FPR = 1 - TNR = \frac{FP}{FP+TN}$
  - False recommendations of items that are actually bad (False Positives or Type **I Error)**

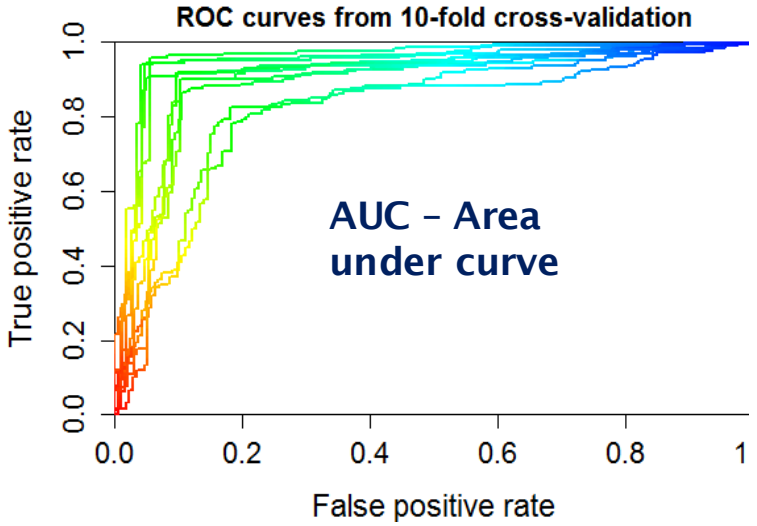
A		Reality		
		Actually Good	Actually Bad	
Prediction	Rated Good	100	100	200
	Rated Bad	20	20	40
		120	120	240

- $TPR = \frac{100}{120} = 0.83$
- $TNR = \frac{20}{120} = 0.16$
- $FPR = 1 - 0.16 = 0.83$

ROC Space

- Receiver Operating Characteristic
  - Plot of FPR vs. TPR
  - The diagonal represents the random guess
  - i.e. length of recommendation list is varied
  - ROC curves can be plotted
  - AUC-ROC – area under ROC curve aggregates performance into one metric

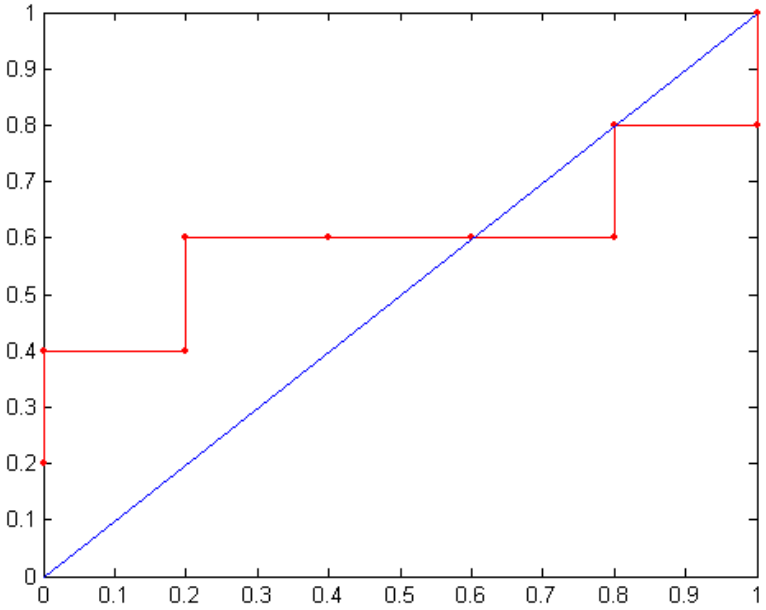




How to construct an ROC curve

	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:



Precision and Recall Example

Predicted

Reality

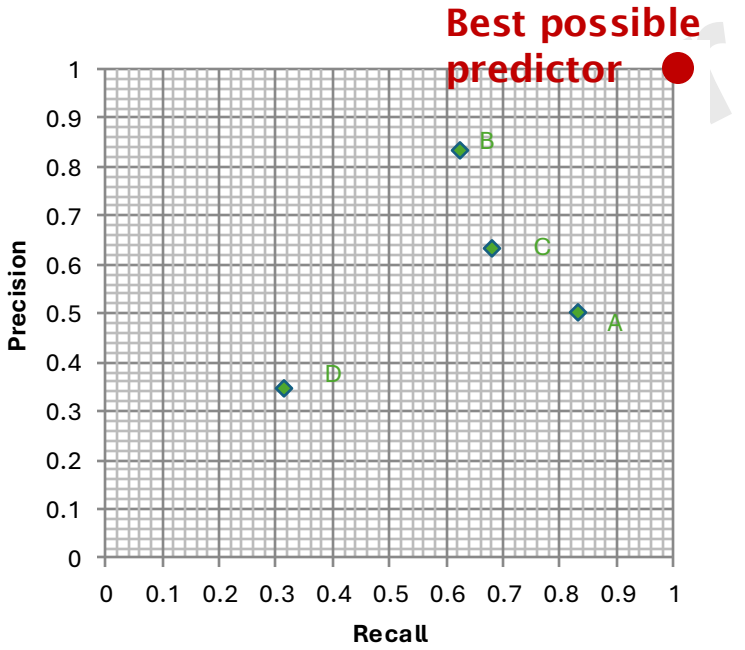
	Selected	*Not Selected*	*Total*
Relevant	TP *9*	FN *15*	*24*
Irrelevant	FP *3*	TN *13*	*16*
Total	*12*	*28*	*40*

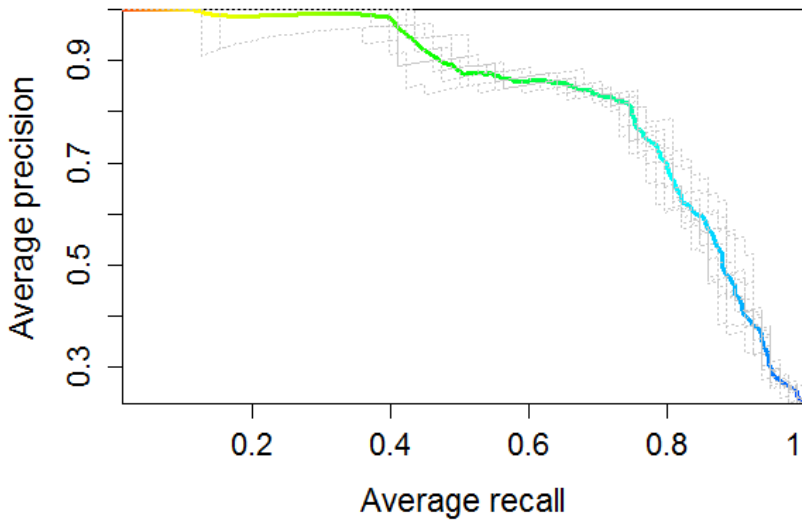
$$P = \frac{9}{12} = 0.75$$

$$R = \frac{9}{24} = 0.375$$

Precision Recall Space

- When a recommender system is tuned to increase precision, recall typically decreases as a result (or vice versa)





Quality of the produced ranking: NDCG

- Discounted cumulative gain (DCG)
  - Logarithmic reduction factor

$$DCG_{pos} = \sum_{i=1}^{pos} \frac{relevance_i}{log_2(i+1)}$$

Where:

- *pos* denotes the position up to which relevance is accumulated
- $relevance_i$ returns the relevance of recommendation at position i

- Idealized discounted cumulative gain (IDCG)
  - *IDCG* represents the maximum achievable DCG with the same set of relevance scores but in the perfect ranking order.
  - In the case of binary relevance scores, all relevant items should be at the start of the list. In the case of graded relevance, you should place all items in a descending order of relevance.

$$IDCG_{pos} = \sum_{i=1}^{pos} \frac{relevance_i}{log_2(i+1)}$$

all items i are now ordered by decreasing relevance

- Normalized discounted cumulative gain (nDCG)
  - Normalized to the interval [0..1]

$$nDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}}$$

Example nDCG

Rank	Hit?
1	X
2	✅
3	✅
4	✅
5	X

$$DCG_5 = \frac{0}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5} + \frac{0}{\log_2 6} = 1.56$$

$$IDCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{0}{\log_2 5} + \frac{0}{\log_2 6} = 2.13$$

$$nDCG_5 = \frac{1.56}{2.13} = 0.73$$