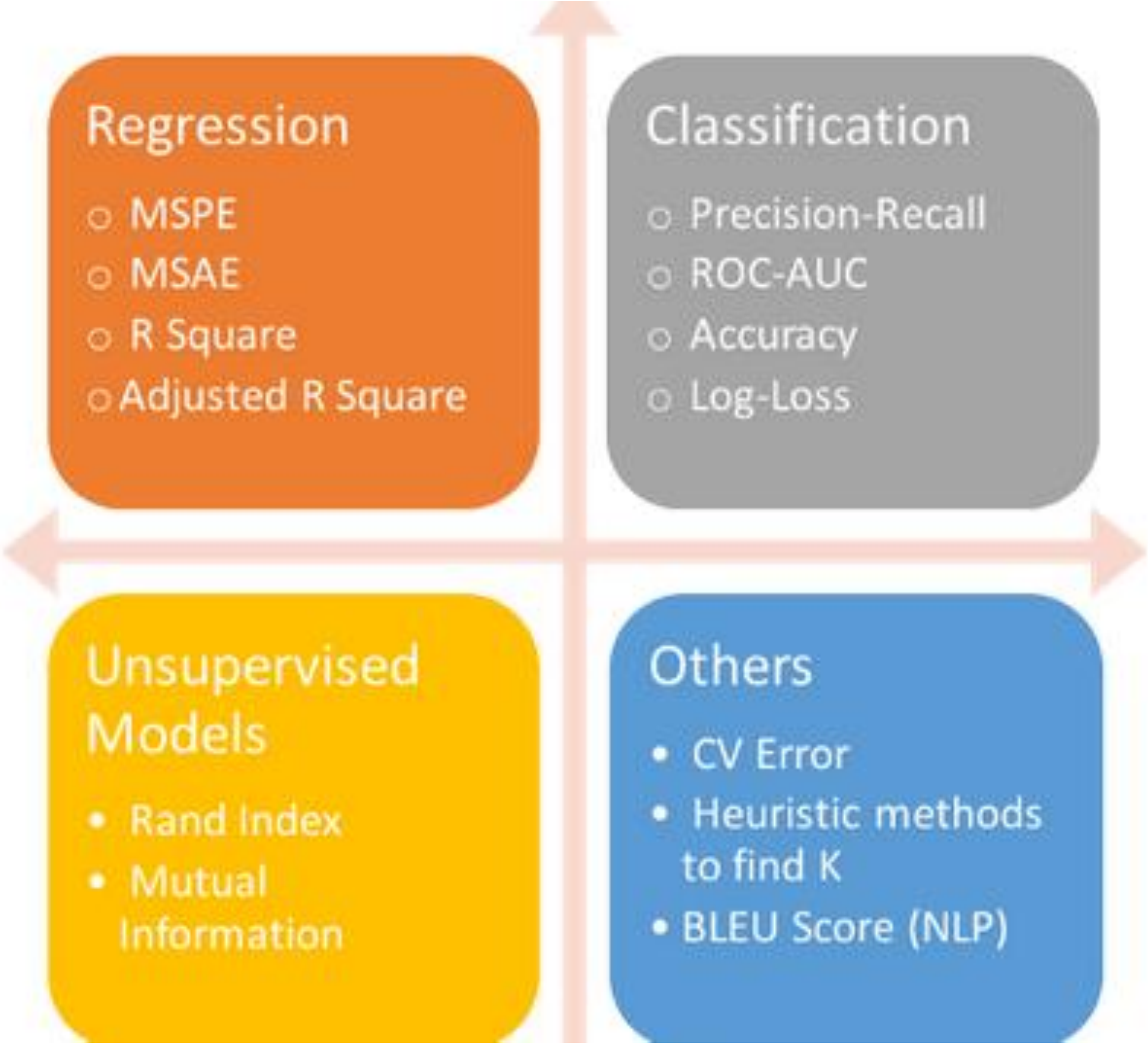
Machine Learning with Python

Measuring Performance of Classifiers

Arghya Ray

Why Evaluate?

- Multiple methods are available to classify or predict
- For each method, multiple choices are available for settings
- To choose best model, need to assess each model's performance



Reference: https://www.kaggle.com/usengecoder/performance-metrics-for-classification-problems

Accuracy Measures (Classification)

Misclassification error

- Error = classifying a record as belonging to one class when it belongs to another class.

- Error rate = percent of misclassified records out of the total records in the validation data

Naïve Rule

Naïve rule: classify all records as belonging to the most prevalent class

- Often used as benchmark: we hope to do better than that
- Exception: when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule (see "lift" – later)

Separation of Records

“High separation of records” means that using predictor variables attains low error

“Low separation of records” means that using predictor variables does not improve much on naïve rule

Confusion Matrix

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Total Value = (TP+FP+FN+TN)

Accuracy = (TP+TN)/Total values

1 – Accuracy = (FP+FN)/Total Values
=Error rate

Source: https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

Confusion Matrix

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

201 1's correctly classified as "1"

85 1's incorrectly classified as "0"

25 0's incorrectly classified as "1"

2689 0's correctly classified as "0"

Accuracy= (201+2689)/(201+85+25+2689)
= 0.96

Error rate = 1-0.96=0.04

Error Rate

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

Overall error rate = (25+85)/3000 = 3.67%

Accuracy = 1 – err = (201+2689)/3000 = 96.33%

If multiple classes, error rate is:

(sum of misclassified records)/(total records)

Cutoff for classification

Most DM algorithms classify via a 2-step process:

For each record,

1. Compute **probability of belonging to class "1"**
2. Compare to cutoff value, and classify accordingly

- Default cutoff value is 0.50

  If >= 0.50, classify as "1"

  If < 0.50, classify as "0"
- Can use different cutoff values
- Typically, error rate is lowest for cutoff = 0.50

Cutoff Table

Actual Class	Prob. of "1"	Actual Class	Prob. of "1"
1	0.996	1	0.506
1	0.988	0	0.471
1	0.984	0	0.337
1	0.980	1	0.218
1	0.948	0	0.199
1	0.889	0	0.149
1	0.848	0	0.048
0	0.762	0	0.038
1	0.707	0	0.025
1	0.681	0	0.022
1	0.656	0	0.016
0	0.622	0	0.004

- If cutoff is 0.50: eleven records are actually in class "1"
- If cutoff is 0.80: seven records are actually in class "1"

Confusion Matrix for Different Cutoffs

Cut off Prob.Val. for Success (Updatable)	0.25

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	11	1
non-owner	4	8

Cut off Prob.Val. for Success (Updatable)	0.75

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	7	5
non-owner	1	11

Other performance measures.

F1-score/ F-score =
2* (Precision*Recall)/(Precision+Recall)

Predicted Class

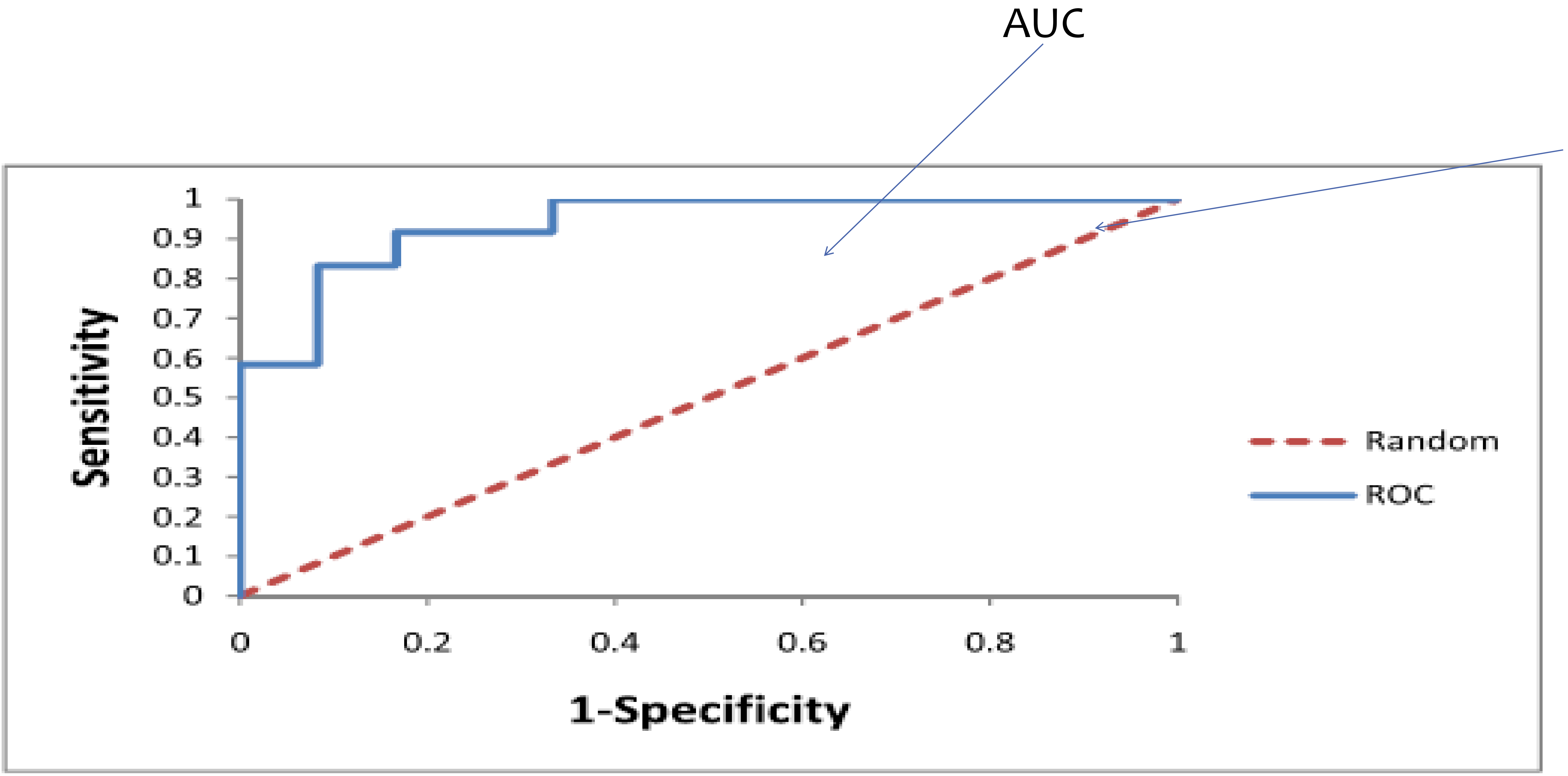
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Recall

Click to add text

Source: https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

Receiver Operating Characteristic curve (ROC Curve)



Random line
Baseline model
Or
No Model
Or
Naïve Model

Along the random line,
Sensitivity = 1 - Specificity
i.e., the system does not know which class the customer belongs to.

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

ROC Curve

Compare performance of DM model to “no model, pick randomly”

Measures ability of DM model to identify the important class, relative to its average prevalence

Charts give explicit assessment of results over a large number of cutoffs

Asymmetric Costs

Misclassification Costs May Differ

The cost of making a misclassification error may be higher for one class than the other(s)

Looked at another way, the benefit of making a correct classification may be higher for one class than the other(s)

Example – Response to Promotional Offer

Suppose we send an offer to 1000 people, with 1% average response rate

("1" = response, "0" = nonresponse)

- "Naïve rule" (classify everyone as "0") has error rate of 1% (seems good)

- Using DM we can correctly classify eight 1's as 1's

  It comes at the cost of misclassifying twenty 0's as 1's and two 0's as 1's.

The Confusion Matrix

	Predict as 1	Predict as 0
Actual 1	8	2
Actual 0	20	970

Error rate = (2+20) = 2.2% (higher than naïve rate)

Introducing Costs & Benefits

Suppose:

- Profit from a “1” is \$10
- Cost of sending offer is \$1

Then:

- Under naïve rule, all are classified as “0”, so no offers are sent: no cost, no profit
- Under DM predictions, 28 offers are sent.

  8 respond with profit of \$10 each

  20 fail to respond, cost \$1 each

  972 receive nothing (no cost, no profit)

- Net profit = \$60

Profit Matrix

	Predict as 1	Predict as 0
Actual 1	$80	0
Actual 0	($20)	0

Generalize to Cost Ratio

Sometimes actual costs and benefits are hard to estimate

- Need to express everything in terms of costs (i.e., cost of misclassification per record)
- Goal is to minimize the average cost per record

A good practical substitute for individual costs is the **ratio** of misclassification costs (e,g,, “misclassifying fraudulent firms is 5 times worse than misclassifying solvent firms”)

Minimizing Cost Ratio

q_1 = cost of misclassifying an actual "1",

q_0 = cost of misclassifying an actual "0"

Minimizing the **cost ratio** q_1/q_0 is identical to minimizing the average cost per record

Software* may provide option for user to specify cost ratio

*Currently unavailable in XLMiner

Note: Opportunity costs

- As we see, best to convert everything to costs, as opposed to a mix of costs and benefits
- E.g., instead of “benefit from sale” refer to “opportunity cost of lost sale”
- Leads to same decisions, but referring only to costs allows greater applicability

Cost Matrix
(inc. opportunity costs)

	Predict as 1	Predict as 0
Actual 1	\$8	\$20
Actual 0	\$20	\$0

Recall original confusion matrix (profit from a "1" = \$10, cost of sending offer = \$1):

	Predict as 1	Predict as 0
Actual 1	8	2
Actual 0	20	970

Multiple Classes

For *m* classes, confusion matrix has *m* rows and *m* columns

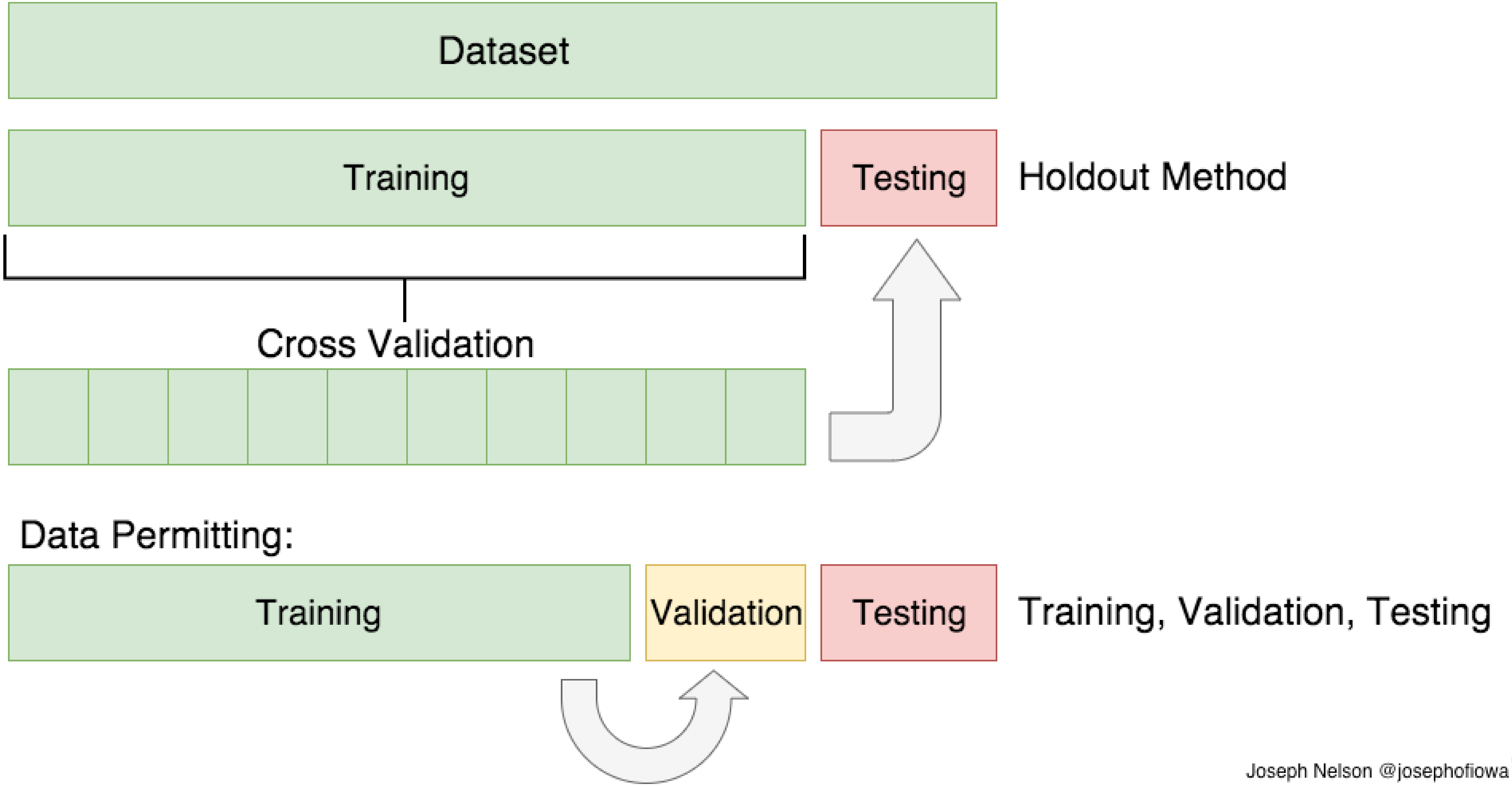
- Theoretically, there are $m(m-1)$ misclassification costs, since any case could be misclassified in $m-1$ ways
- Practically too many to work with
- In decision-making context, though, such complexity rarely arises – one class is usually of primary interest

Confusion Matrix for Multi-class problems

https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/#:~:text=The%20confusion%20matrix%20is%20a,and%20False%20Negative(FN).

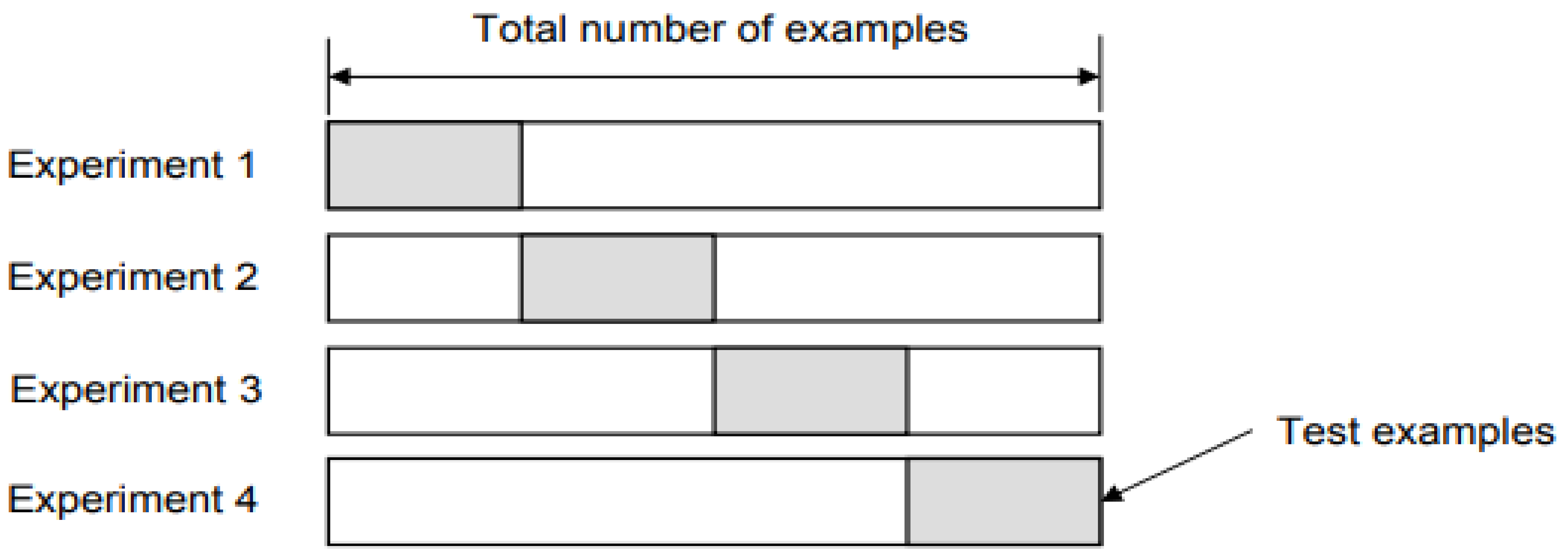
https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Dividing the dataset into training and testing sets



k-Fold Cross Validation

- **Create a K-fold partition of the the dataset**
  - For each of K experiments, use K-1 folds for training and a different fold for testing
    - This procedure is illustrated in the following figure for K=4



- **K-Fold Cross validation is similar to Random Subsampling**
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- **As before, the true error is estimated as the average error rate on test examples**

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Source: https://www.csd.uwo.ca/~oveksler/Courses/Winter2006/CS434_654b/pr_l13.pdf

Summary

- Evaluation metrics are important for comparing across DM models, for choosing the right configuration of a specific DM model, and for comparing to the baseline
- Major metrics: confusion matrix, error rate, predictive error
- Other metrics when
    - one class is more important
    - asymmetric costs
- When important class is rare, use oversampling
- In all cases, metrics computed from validation data

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.

- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.

- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..