Machine Learning & Data Mining

Evaluation of supervised approaches

Kyung-Ah Sohn

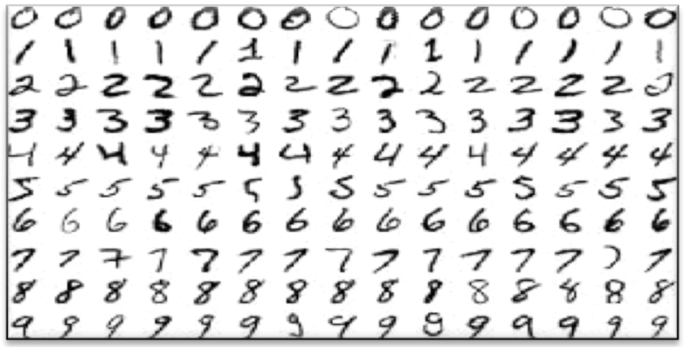
Ajou University

Outline

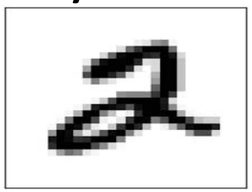
- Performance measure for classification
  - Accuracy
  - Specificity/Sensitivity, Recall/Precision
  - ROC curve

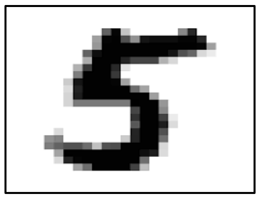
- Model selection
  - Over-fitting
  - Cross validation

MNIST: handwritten digits



Consider Binary classification



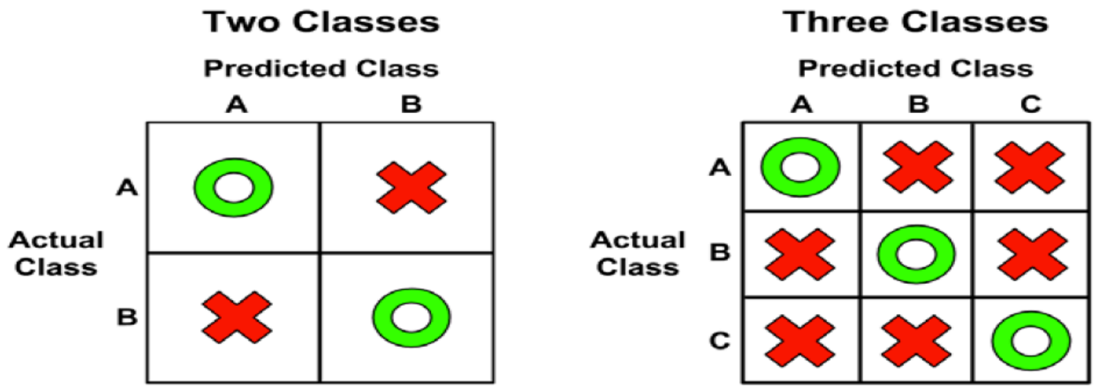


'2' or 'not 2'?

Performance measure

- For a test data *X*, measure of closeness between true label Y_{true} and predicted Y_{pred}
  - Rather than how fast it takes to classify or learn the classifier, scalability, etc.

- Confusion matrix



Binary Classification

1100 test images

Classifier 1

	Predicted ‘2’	Predicted ‘Not 2’
True ‘2’	70	30
True ‘Not 2’	140	860

Classifier 2

	Predicted ‘2’	Predicted ‘Not 2’
True ‘2’	20	80
True ‘Not 2’	50	950

Which classifier is better?

Performance measure

- The class of interest is known as the positive class
- All the others are known as negative

- True Positive (TP): Correctly classified as the class of interest
- False Negative (FN): Incorrectly classified as not the class of interest
- False Positive (FP): Incorrectly classified as the class of interest
- True Negative (TN): Correctly classified as not the class of interest

	Predicted class		
Actual class		Class=1	Class=0
	Class=1	TP	FN
	Class=0	FP	TN

Metrics for Performance Evaluation

	Predicted class		
Actual class		Class=1	Class=0
	Class=1	A	B
	Class=0	C	D

- Widely-used metric:

$$Accuracy = \frac{A+D}{A+B+C+D} = \frac{TP+TN}{TP+TN+FP+FN}$$

Num. correctly classified / total num. of test data

Limitation of Accuracy

- In binary classification
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If a classifier predicts everything to be class 0, accuracy is:
  - Accuracy is misleading because the classifier does not detect any Class 1 example

Sensitivity & Specificity

	Predicted class	
Actual class	Class=1	Class=0
Class=1	TP	FN
Class=0	FP	TN

$$Sensitivity = \frac{TP}{TP + FN}$$ True Positive rate

$$Specificity = \frac{TN}{FP + TN}$$ True Negative rate

Precision & Recall
(in Information Retrieval)

	Predicted class	
Actual class	Class=1	Class=0
Class=1	TP	FN
Class=0	FP	TN

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

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

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Sensitivity = \frac{TP}{TP + FN}$$ True Positive rate

$$Specificity = \frac{TN}{FP + TN}$$ True Negative rate

- High sensitivity = Few false negatives
- High specificity = Few false positives

$\boxed{\text{Tradeoff}}$

e.g. airport alarm system

Actual class	Predicted class
1	0
0	0
0	0
1	1
1	0
0	0
0	0
0	1

X	FN
O	TN
O	TN
O	TP
X	FN
O	TN
O	TN
X	FP

$$Accuracy = \frac{5}{8} = 62.5\%$$

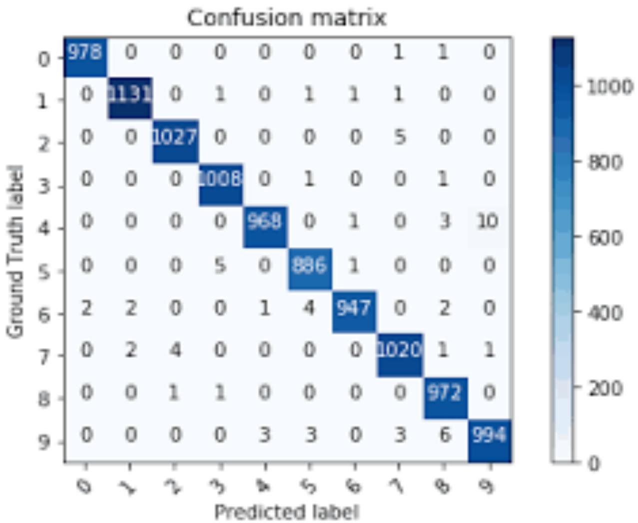
Classification Performances

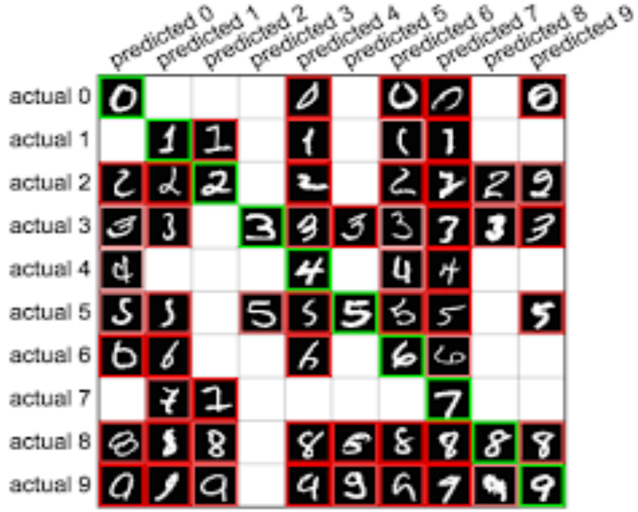
- Confusion matrix

	Predicted class	
Actual class	Class=1	Class=0
Class=1	4	1
Class=0	2	3

  - Accuracy=
  - Misclassification error=1-Accuracy=
  - Sensitivity (true positive, recall) =
  - Specificity (true negative)=
  - Precision =
  - F1 measure =

Multi-class Classification: MNIST

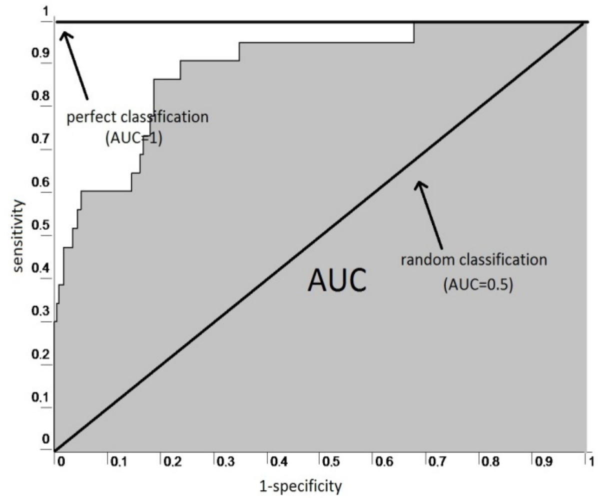




- What if you change a threshold?
  - e.g. If BFP > 25, classify as female

ROC (Receiver Operating Characteristic) curve

- ROC curve plots (1-specificity) (or FP rate) on the x-axis against sensitivity (or TP rate) on the y-axis



AUC: area under the curve

10.0	21.7	8.9	19.9	23.4	28.9	15.7	21.6	21.5	23.2
M	F	M	F	M	F	M	F	M	F

How to construct an ROC curve

	Instance	Sorted	True Class
+	1	28.9	F
+	2	23.4	M
+	3	23.2	F
+	4	21.7	F
+	5	21.6	F
+	6	21.5	M
-	7	19.9	F
	8	15.7	M
	9	10	M
	10	8.9	M

TP	FP	TN	FN	Sensitivity	Specificity
1	0	5	4	0.2	1
1	1	4	4	0.2	0.8

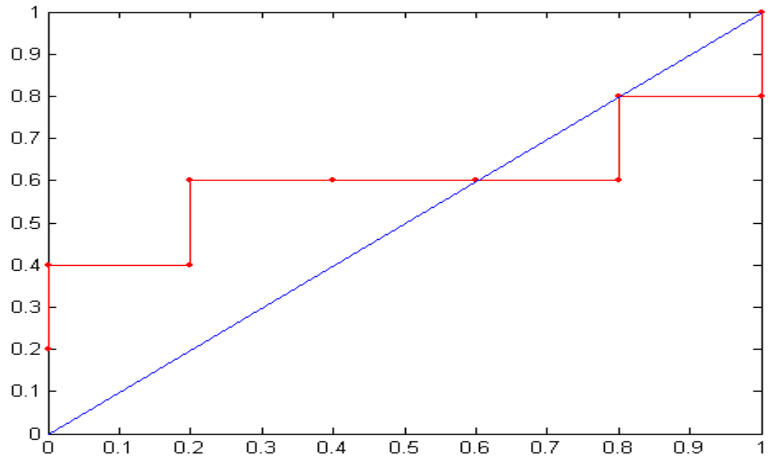
How to construct an ROC curve

	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 test instance P(+|A)
- Sort the instances according to P(+|A) in decreasing order
- Apply threshold at each unique value of P(+|A)
- Count TP,FP,TN,FN at each threshold
- Compute TP rate, FP rate

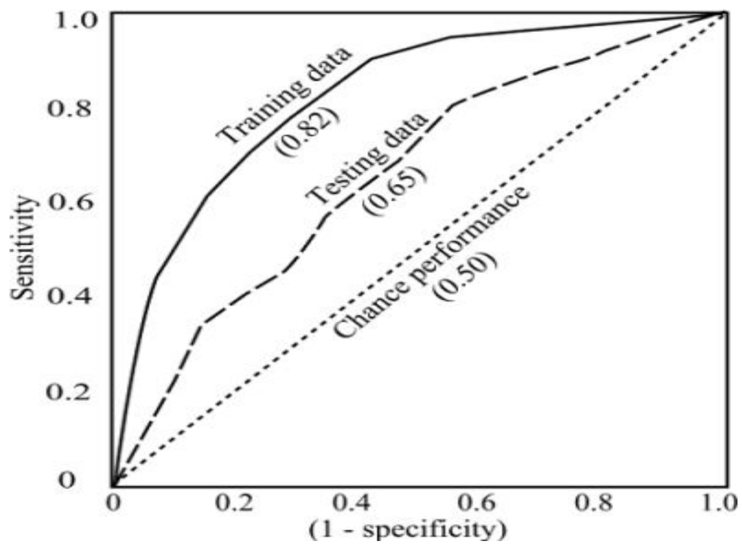
Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:



ROC curves

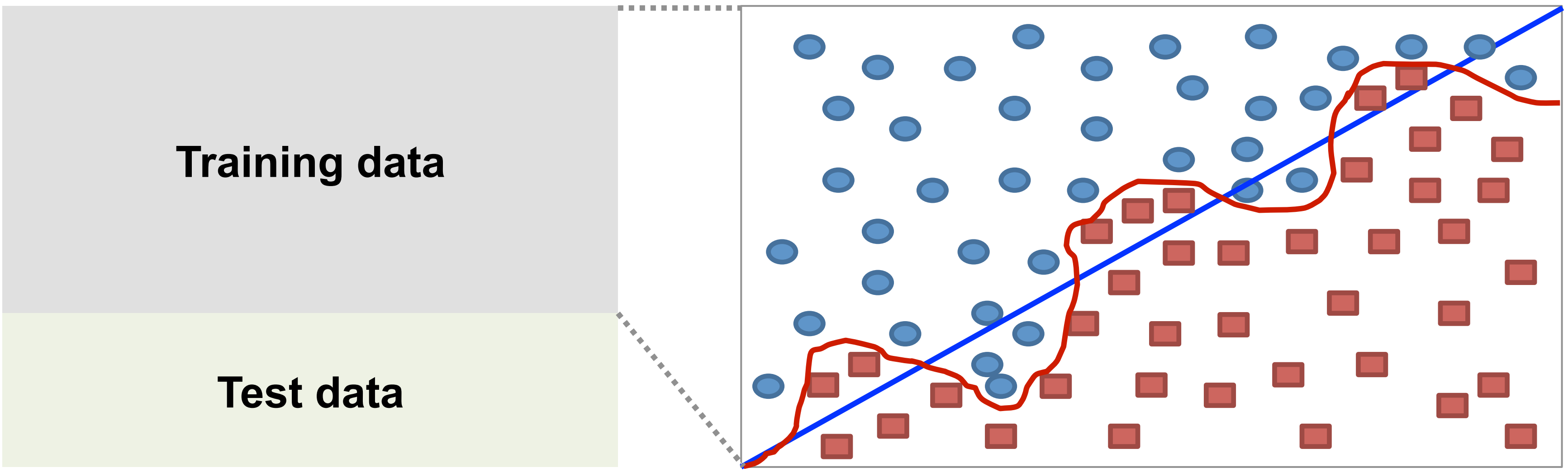
- Typically,



AUC (area under the curve)

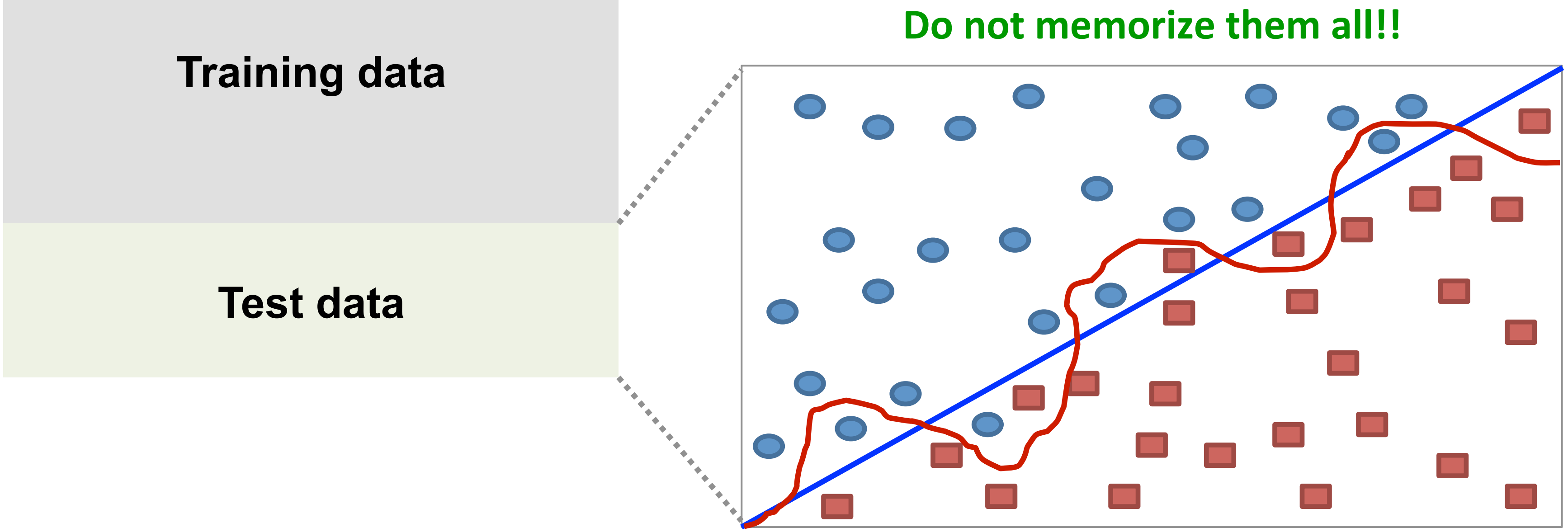
OVER-FITTING AND CROSS VALIDATION

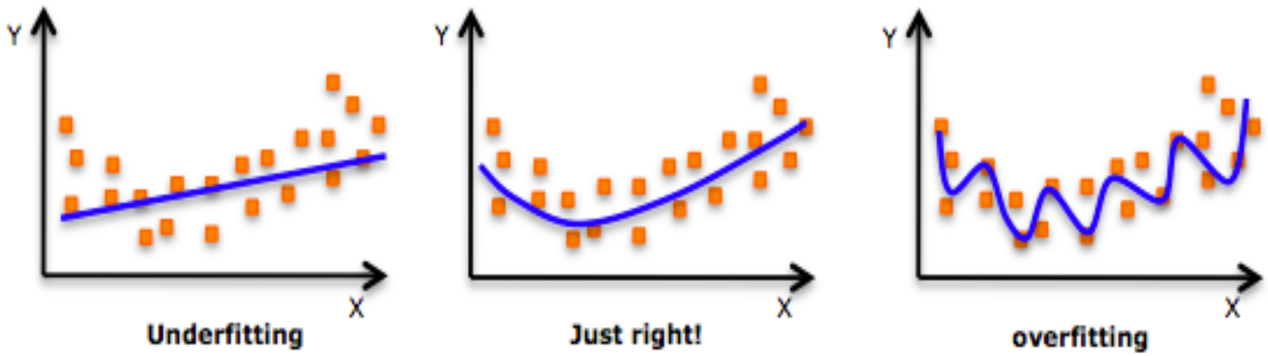
❖ Over-fitting for training data



Is red boundary is better than blue one?

❖ Over-fitting for training data



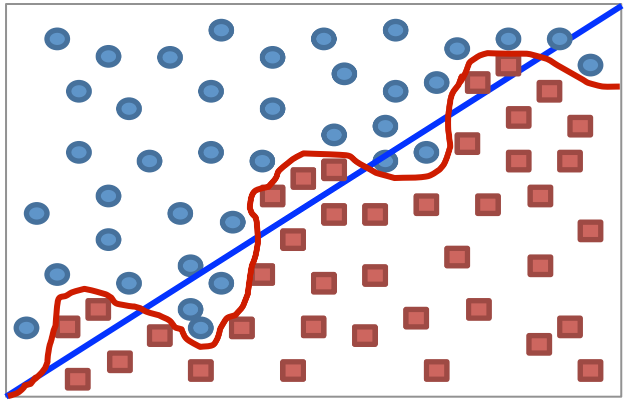


underfitting is not as prevalent
as overfitting

Why evaluate?

- Multiple algorithms for classification available
- For each algorithm, multiple parameter choices are available
  - e.g. the choice of k in k-nearest neighbors
- To choose the best model, one needs to assess each model's performances

Validation

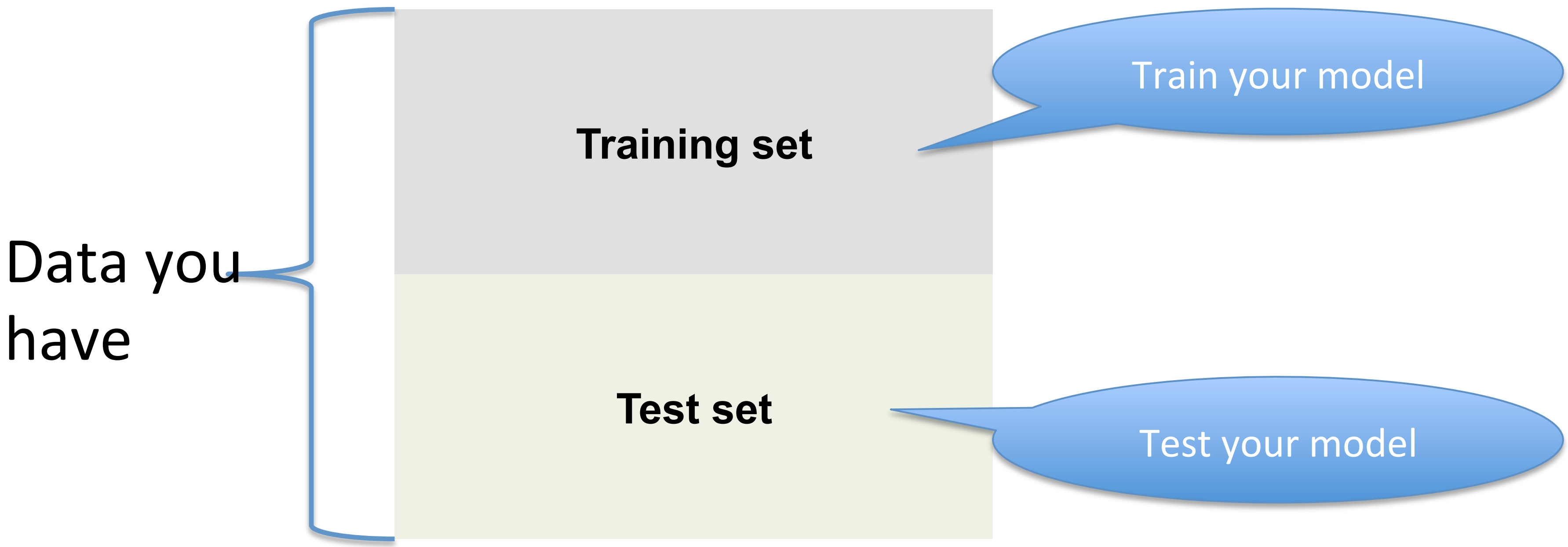


- The problems of over-fitting

- Internal validation: validate your model on your current data set (cross-validation)

- External Validation: Validate your model on a completely new dataset

Holdout (or Test-set) validation

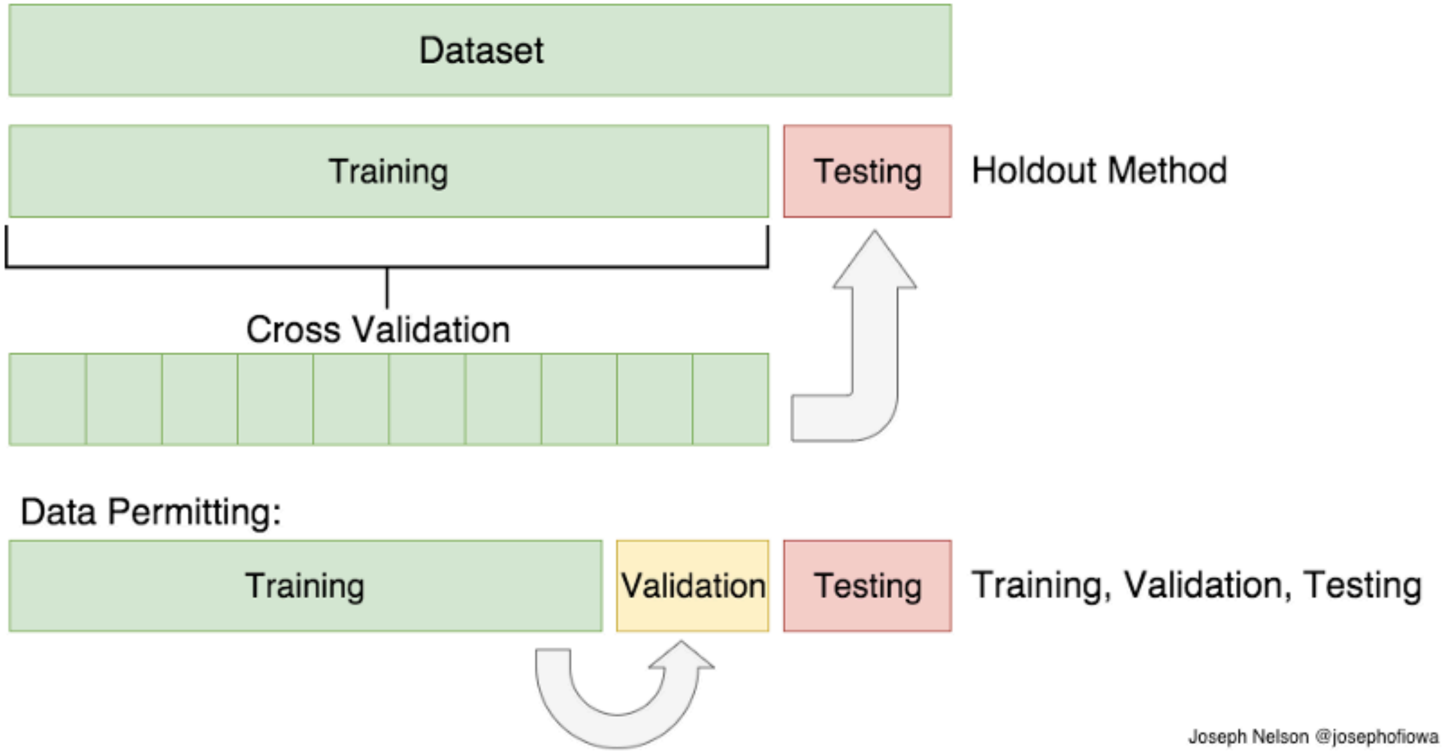


- “waste” half of your data
- often you don’t have enough data to spare

Cross-validation

- When to use?
  - To choose the best parameter setting
  - Anytime you want to prove that your model does not over-fit the training data and it will have good prediction in new datasets

Train/Validation/Test set

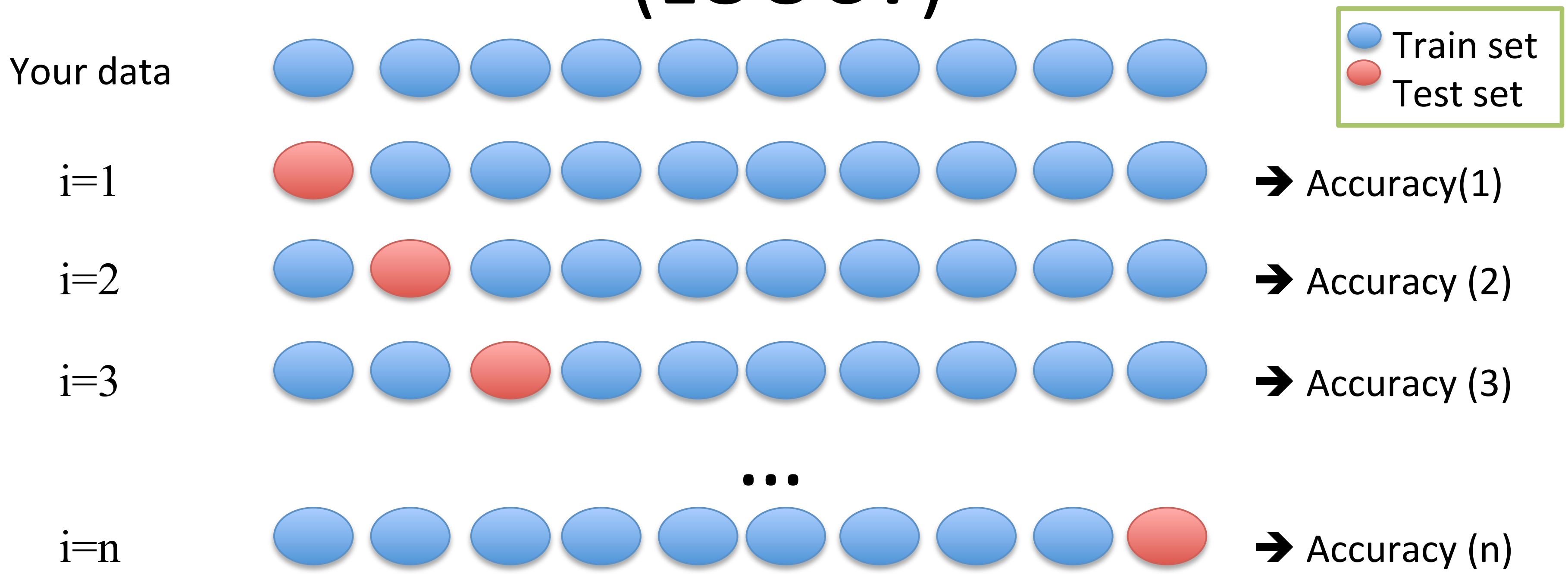


https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

Cross-validation

- Leave-one-out validation
- K-fold cross validation

Leave-One-Out Cross Validation (LOOCV)



Final accuracy = (Accuracy (1)+…+Accuracy(n) / n

LOOCV

- Leave one sample out at a time
- Learn the model on the remaining training data
- Test on the held out data point

- Summarize the performance of each run

LOOCV

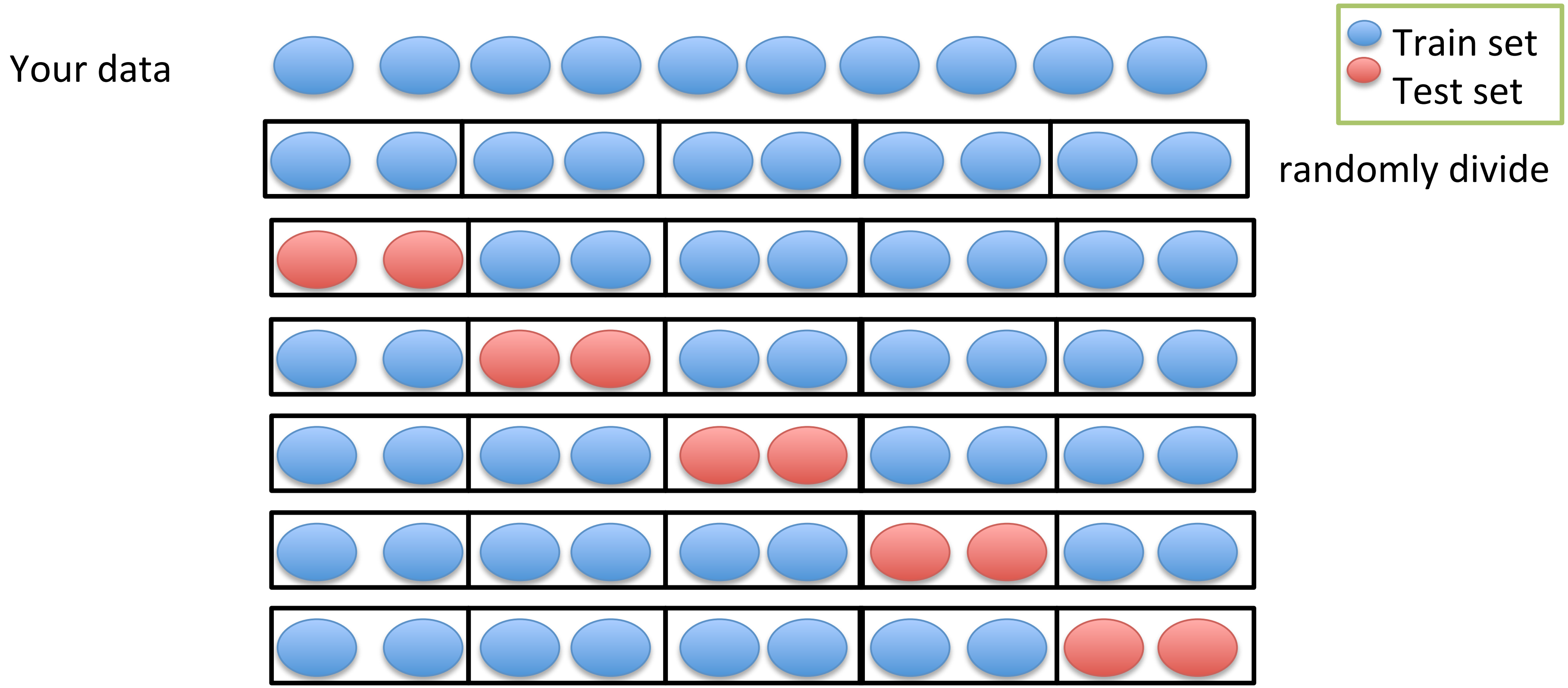
- The evaluation result is good, but it is very expensive to compute
  - n runs of the learning algorithm if you have n data points

    n x (running time of the algorithm)

K-fold cross validation

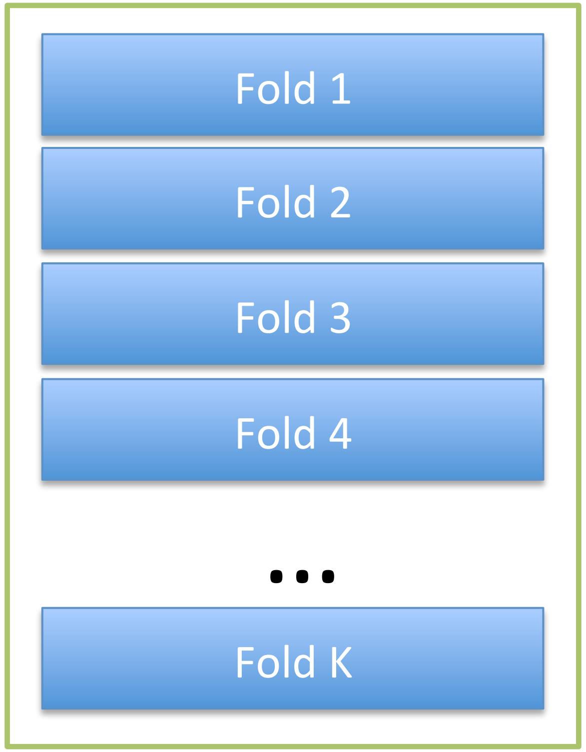
- One way to improve the holdout method
- The data set is divided into k subsets, and the holdout method is repeated k times
- Each time, one of the k subsets is used as the test set, and the remaining subsets are used in training

Example: 5-fold CV



K-fold CV

- The average error rates (or accuracy measures) across all k trials is computed.
- It matters less how the data is divided
- Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times
- Typical choice is 10-fold CV (or 5-fold)



Which kind of Cross Validation?

	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavior	Doesn't waste data
10-fold	Wastes 10% of the data. 10 times more expensive than test set	Only wastes 10%. Only 10 times more expensive instead of R times.
3-fold	Wastier than 10-fold. Expensivier than test set	Slightly better than test-set
R-fold	Identical to Leave-one-out	

http://www.autonlab.org/tutorials/

CV-based model selection

- Example: choosing “k” for k-NN
- Step 1: compute 10-fold-CV error for six different model classes

Algorithm	TRAINERR	10-fold-CV-ERR	Choice
K=1			
K=2			
K=3			
K=4			⌦
K=5			
K=6			

- Step 2: Whichever model class gave best CV score: train it with all the data, and that’s the predictive model you will use

Cross-validation is useful

- Preventing over-fitting
- Comparing different algorithms
- Choosing the optimal parameters
- For any supervised learning approaches

What you should know

- How to measure performance of supervised approaches
- Why you can't use "training-set-error" to estimate the quality of your learning algorithm on your data.
- Why you can't use "training set error" to choose the learning algorithm
- Holdout (Test-set) cross-validation
- Leave-one-out cross-validation
- k-fold cross-validation