

CS109 – Data Science

SVM, Best Practices

Hanspeter Pfister, Mark Glickman, Verena Kaynig-Fittkau

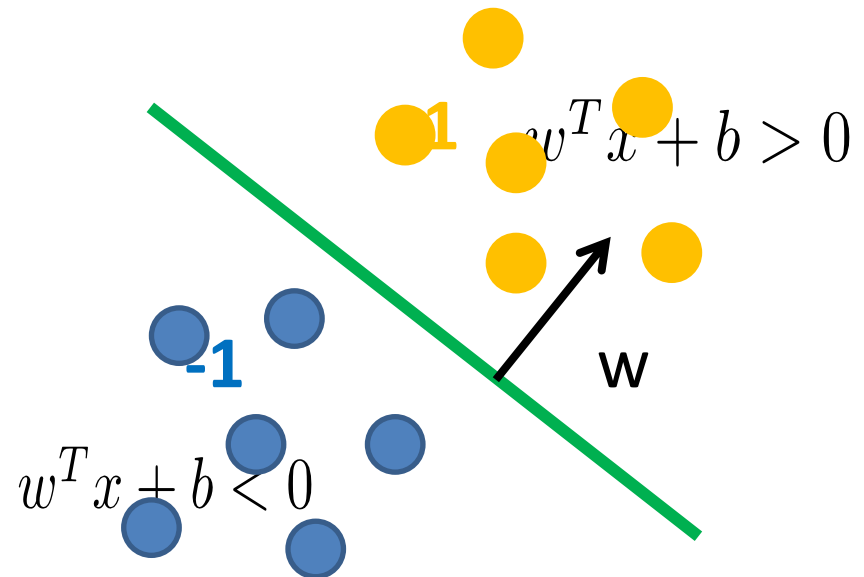


Announcements

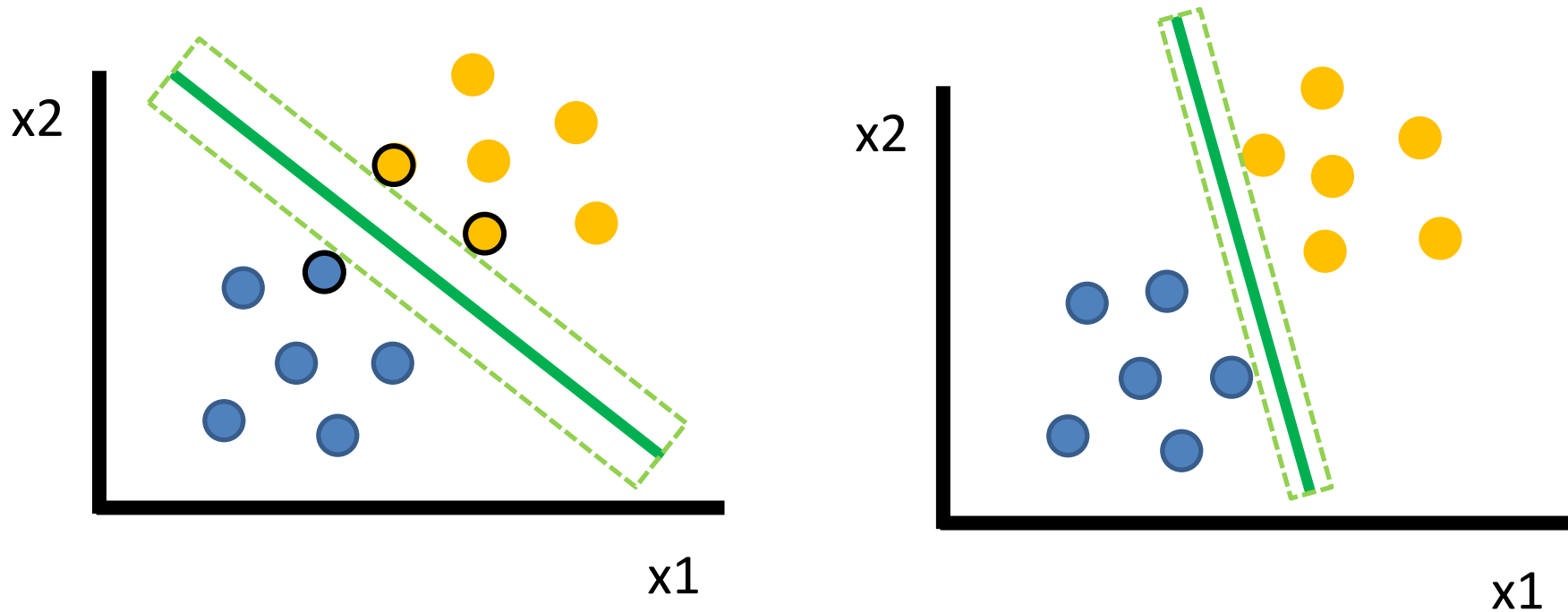
- Make sure you really submit your AWS ID!

Separating Hyperplane

- x : data point
- y : label $\in \{-1, +1\}$
- w : weight vector
- b : bias



Maximum Margin Classification



Solution depends only on the support vectors!

Maximum Margin Classification



Solution depends only on the support vectors!

What about outliers?

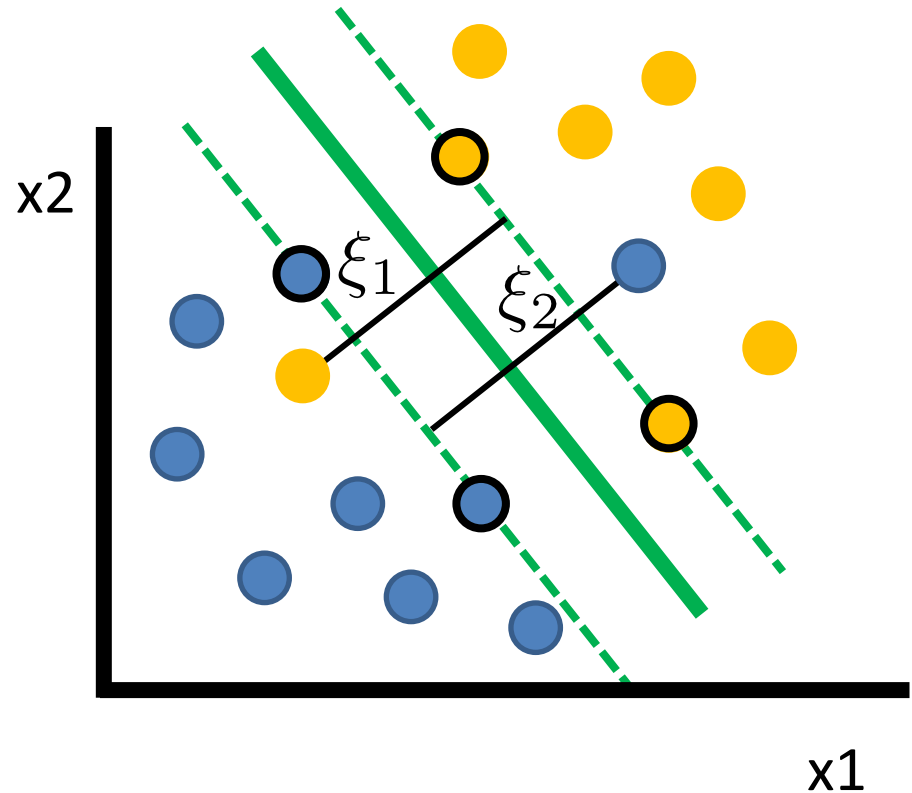
ξ_i : slack variables

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2$$

subject to:

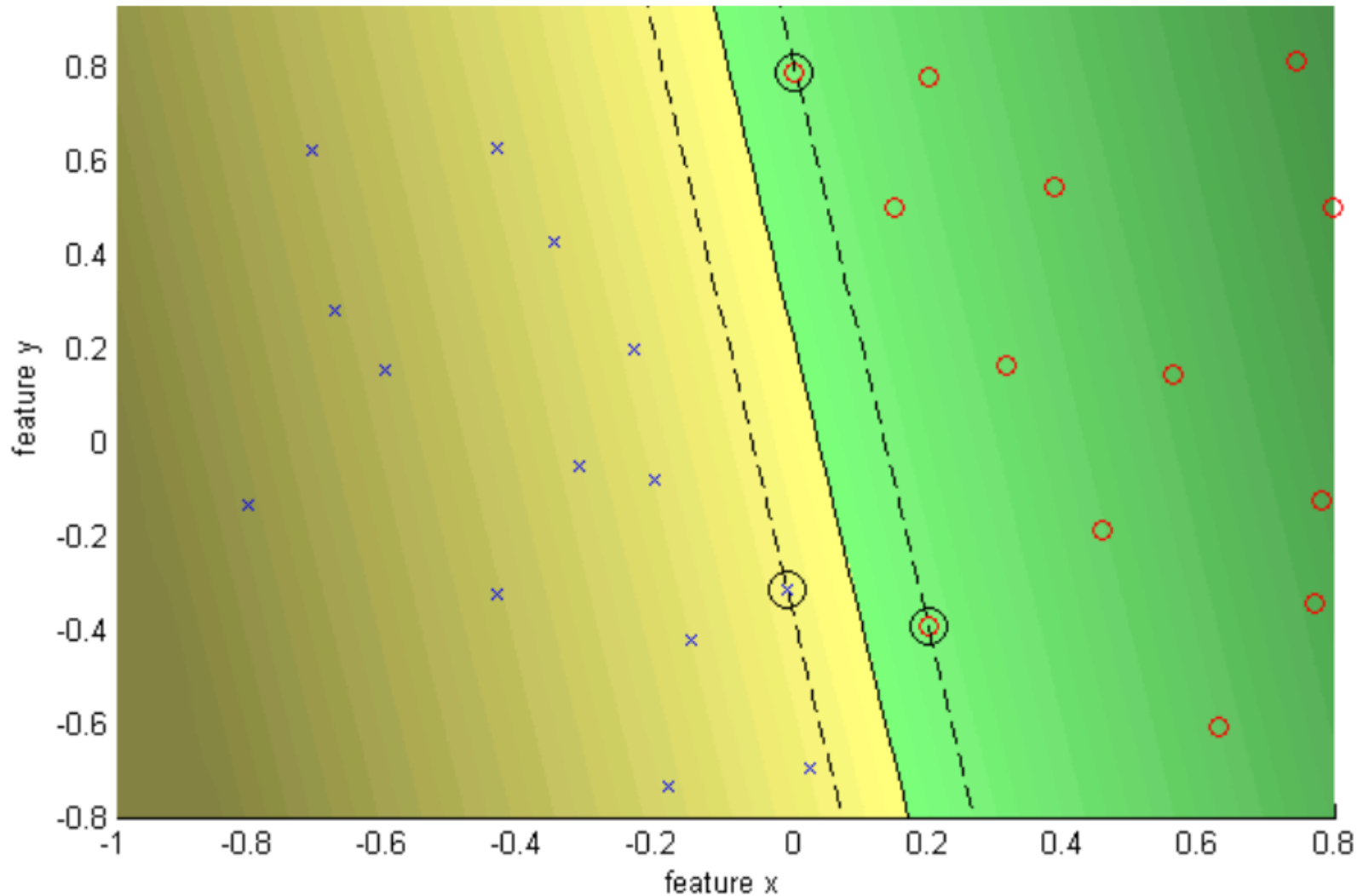
$$y^{(i)}(w^T x^{(i)} + b) \geq 1$$

$$(i = 1, \dots, n)$$

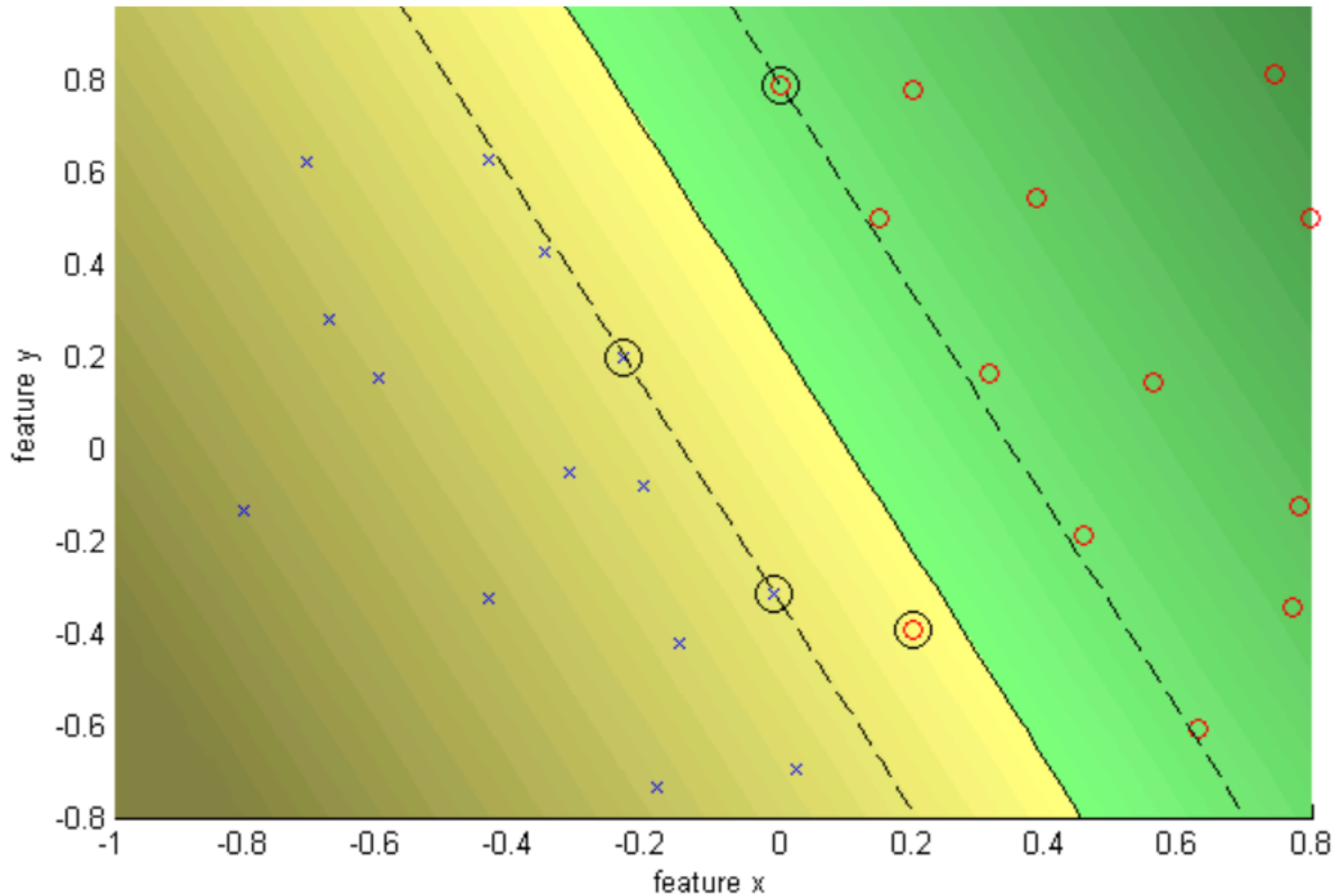


How can we use this to account for imbalanced data?

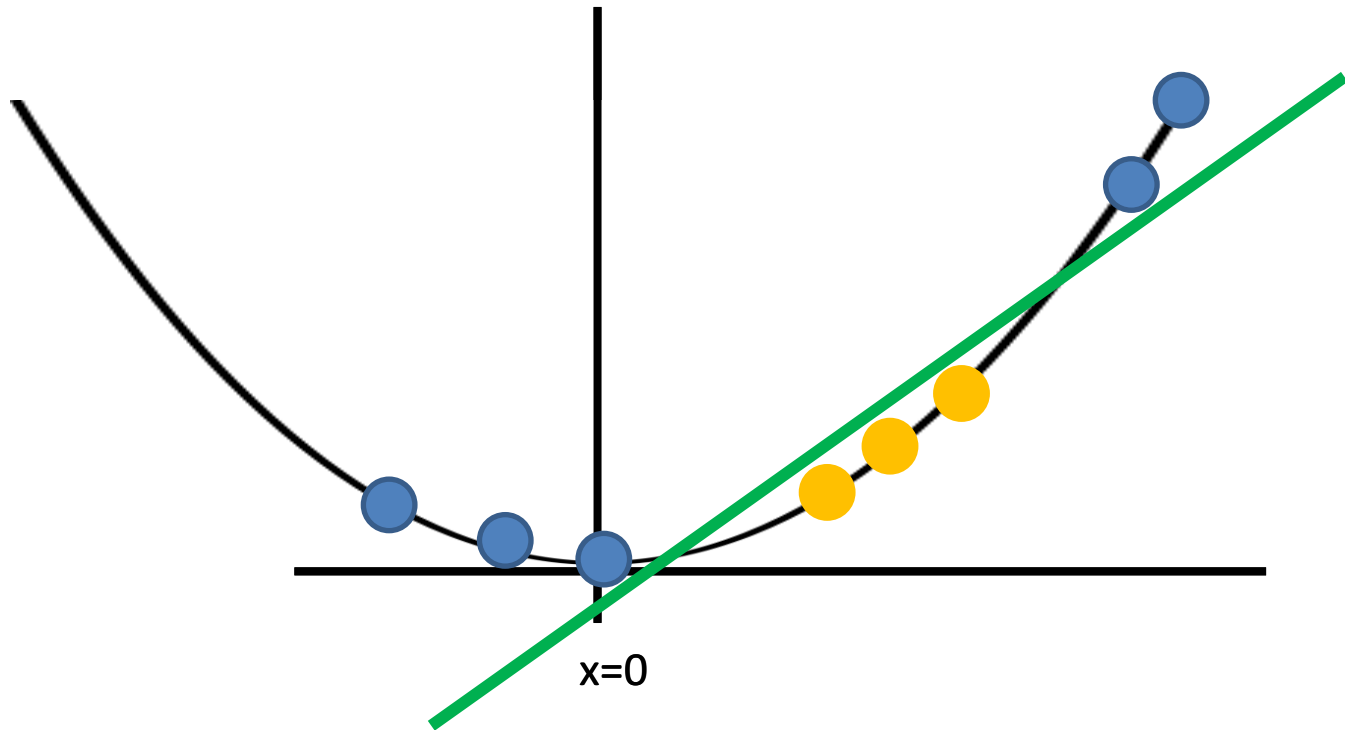
Hard Margin ($C = \text{Infinity}$)



Soft Margin ($C = 10$)



XOR problem revised



Did we add information to make the problem separable?

Kernel Functions

$$K(x, z) = \Phi(x) \cdot \Phi(z)$$

- Polynomial:

$$K(x, z) = (1 + x \cdot z)^s$$

- Radial basis function (RBF):

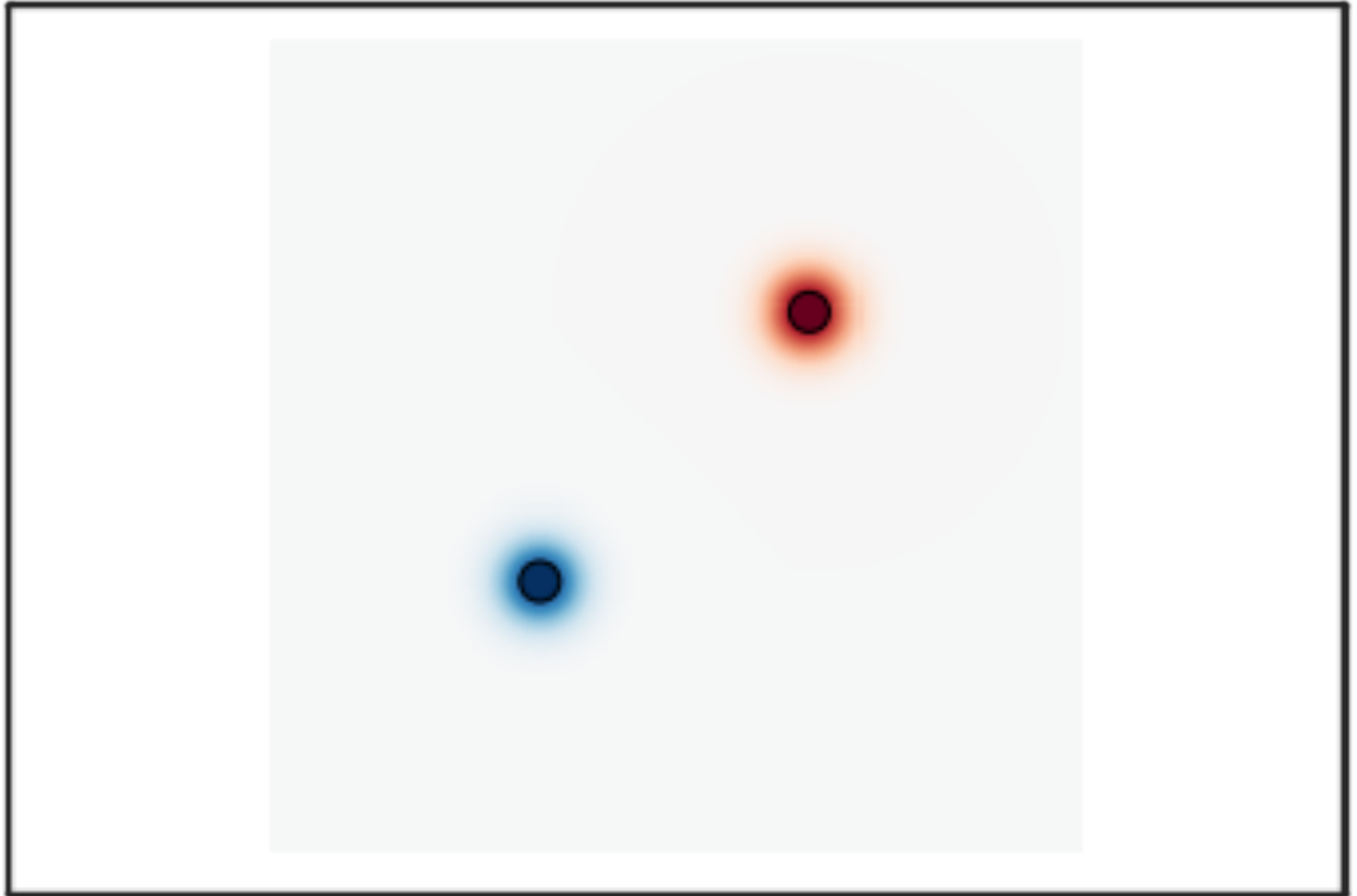
$$K(x, z) = \exp(-\gamma \|x - z\|^2)$$

Prediction

$$w^T x + b = \sum_{i=1}^m \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b.$$

- Again we can use the kernel trick!
- Prediction speed depends on number of support vectors

RBF Intuition



Tips and Tricks

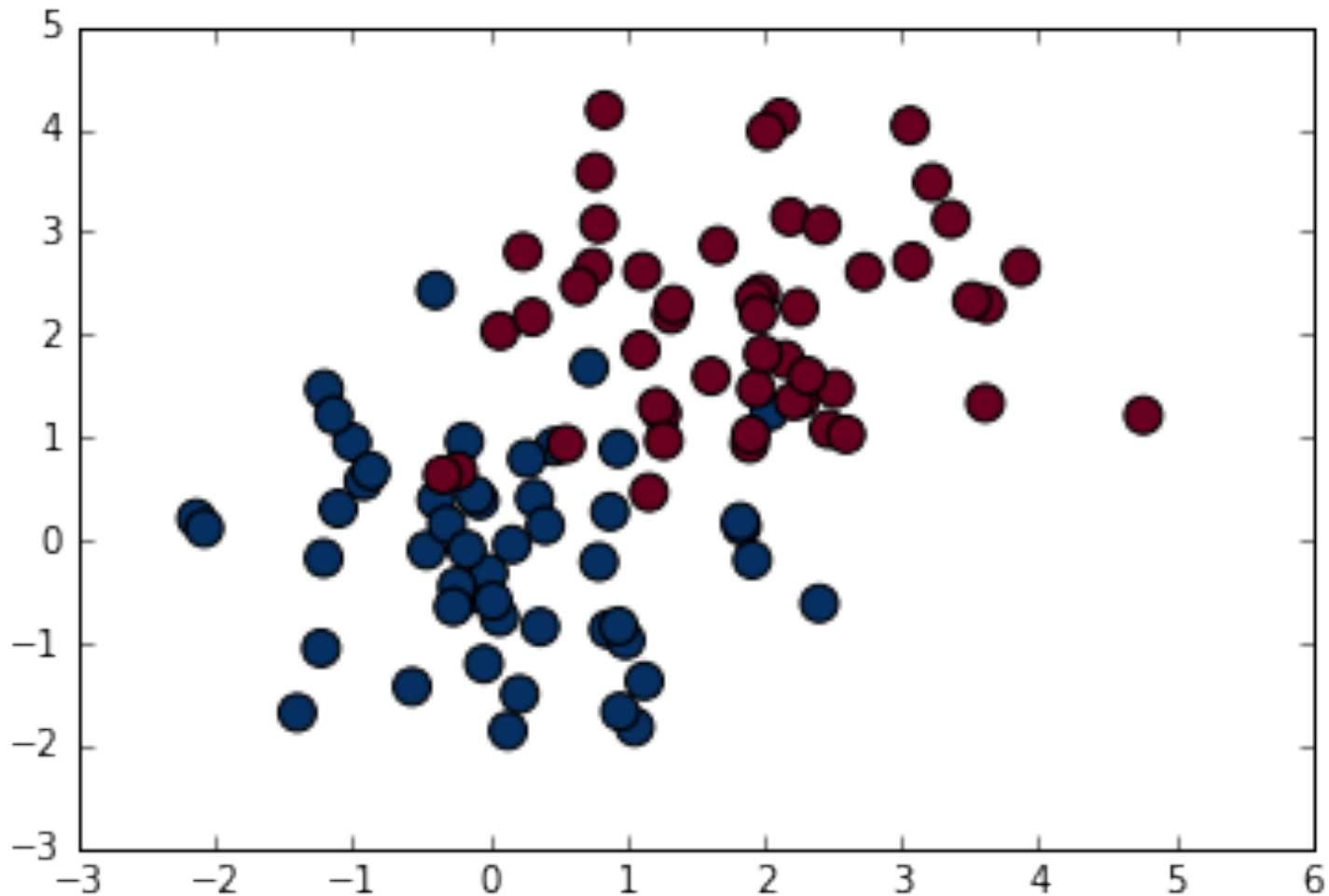
- SVMs are not scale invariant
- Check if your library normalizes by default
- Normalize your data
 - mean: 0 , std: 1
 - map to $[0,1]$ or $[-1,1]$
- Normalize test set in same way!

Tips and Tricks

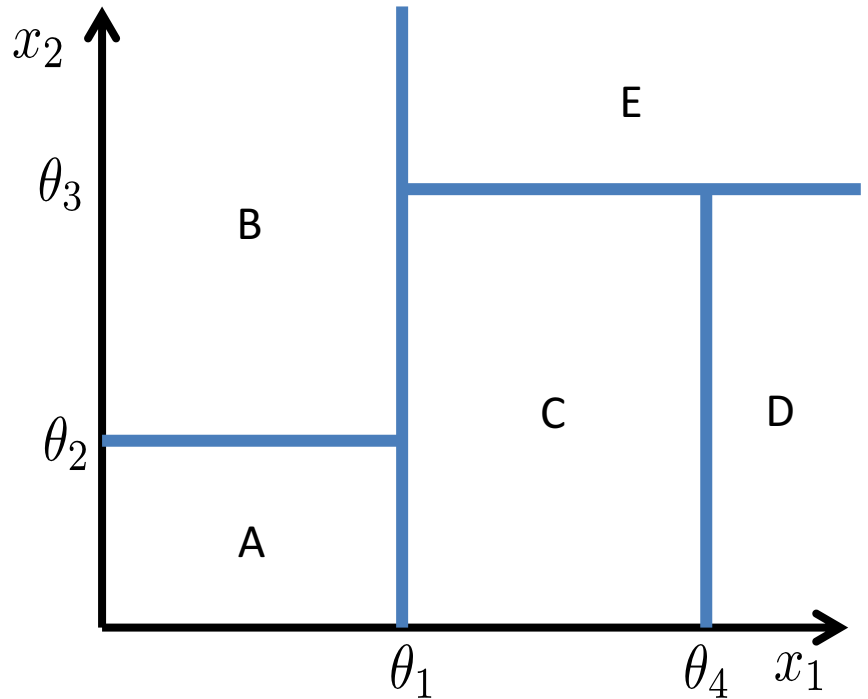
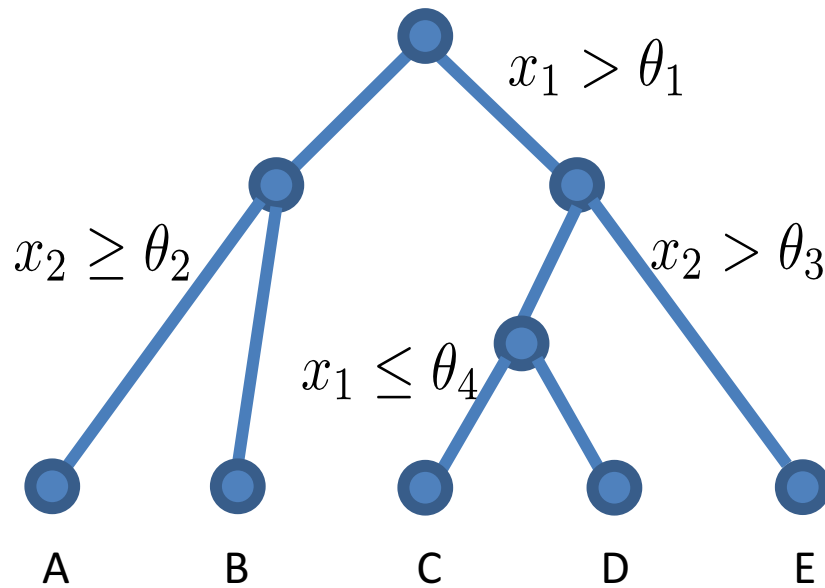
- RBF kernel is a good default
- For parameters try exponential sequences
- Read:

Chih-Wei Hsu et al., “**A Practical Guide to Support Vector Classification**”,
Bioinformatics (2010)

What about that scale invariance



Decision Tree Recap



What about that scale invariance?

Messed up?	RBF SVM	Linear SVM	LR	KNN	Decision Tree	Adaboost
No						
Yes						

Experiments done with simulated data in **two** dimensions.
Data altered by scaling second dimension with a factor of 1000

What about that scale invariance?

Messed up?	RBF SVM	Linear SVM	LR	KNN	Decision Tree	Adaboost
No	0.98	0.99	0.97	0.98	0.95	0.97
Yes	0.96	0.99	0.97	0.87	0.95	0.97

Experiments done with simulated data in **two** dimensions.
Data altered by scaling second dimension with a factor of 1000

What about that scale invariance?

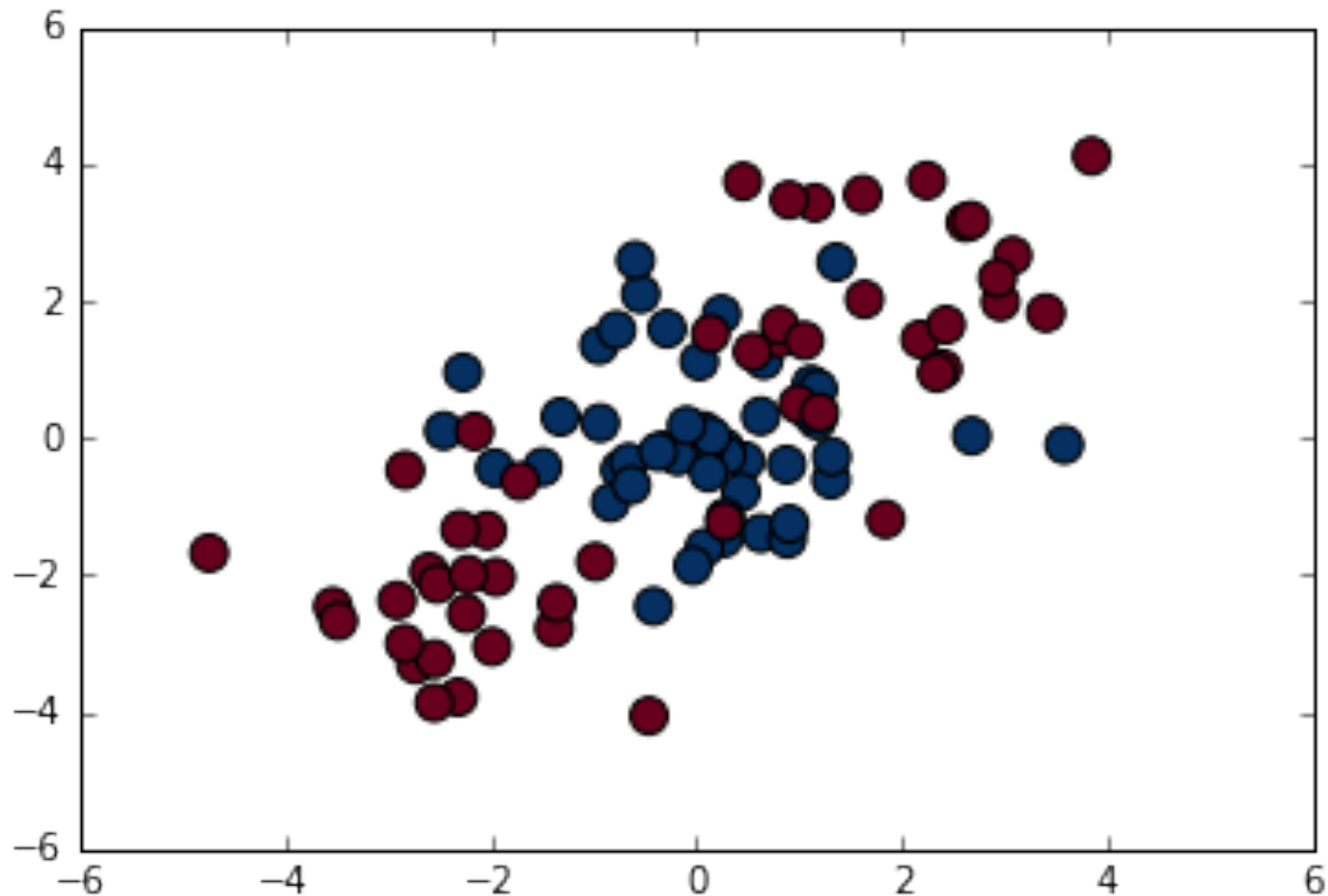
Messed up?	RBF SVM	Linear SVM	LR	KNN	Decision Tree	Adaboost
No	0.99	0.99	0.99	0.99	0.86	0.98
Yes	0.83	0.96	0.96	0.83	0.86	0.98

Experiments done with simulated data in **five** dimensions.

Data altered by scaling

- second dimension with a factor of 1000
- third dimension with a factor of 100
- fourth dimension with a factor of 0.0001

What about that scale invariance



What about that scale invariance?

Messed up?	RBF SVM	Linear SVM	LR	KNN	Decision Tree	Adaboost
No	0.94	0.45	0.5	0.91	0.86	0.87
Yes	0.8	0.54	0.6	0.83	0.85	0.87

Experiments done with simulated data in **five** dimensions.

Data not linearly separable anymore

Data altered by scaling

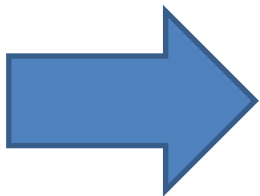
- second dimension with a factor of 1000
- third dimension with a factor of 100
- fourth dimension with a factor of 0.0001

Take Home Message

- Consider normalization if:
 - Your classifier uses Euclidean distance
 - You use L1 or L2 regularization with one parameter for all dimensions
 - Your optimization method uses gradients

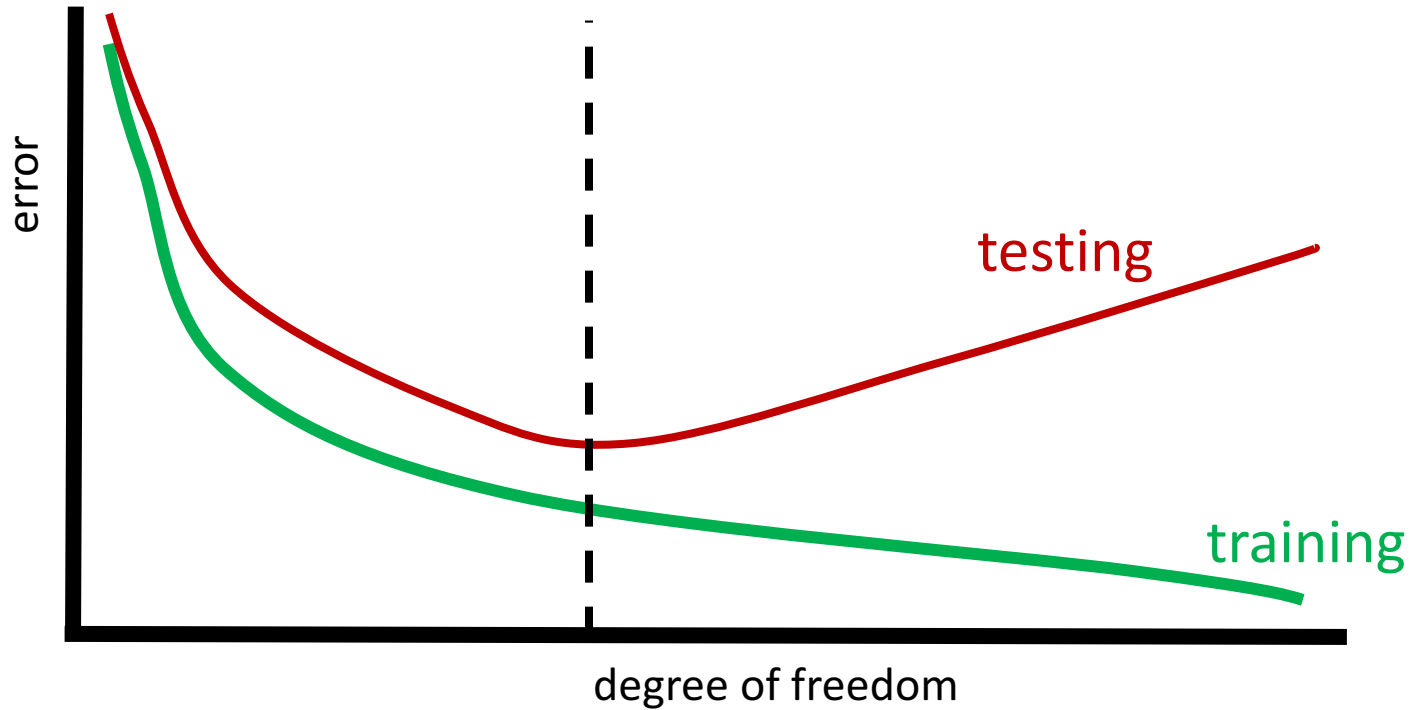
Parameter Tuning

- Given a classification task
- Which kernel ?
- Which kernel parameter values?
- Which value for C?



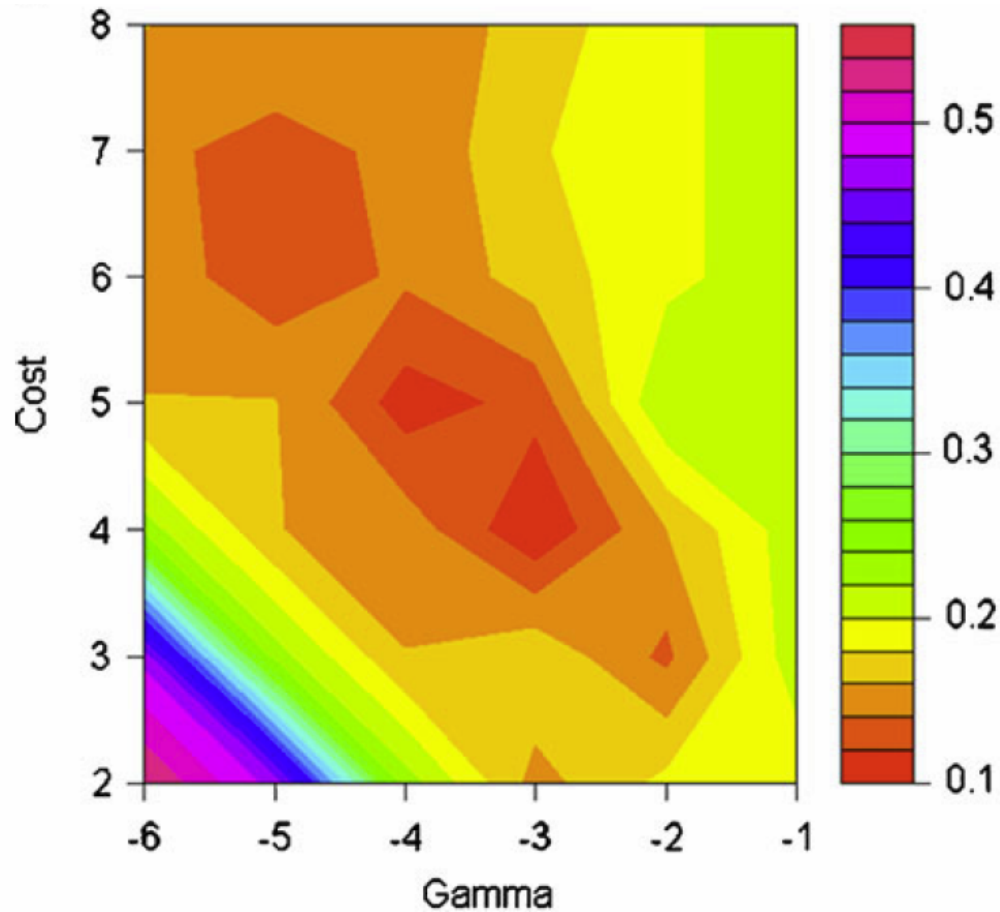
Try different combinations
and take the **best**.

Train vs. Test Error



Where is a SVM with RBF kernel for small or large gamma?

Grid Search



Zang et al., “Identification of heparin samples that contain impurities or contaminants by chemometric pattern recognition analysis of proton NMR spectral data”, Anal Bioanal Chem (2011)

Error Measures

- True positive (tp)
- True negative (tn)
- False positive (fp)
- False negative (fn)

		predicted	
		1	-1
true	1	tp	fn
	-1	fp	tn

TPR and FPR

- True Positive Rate:

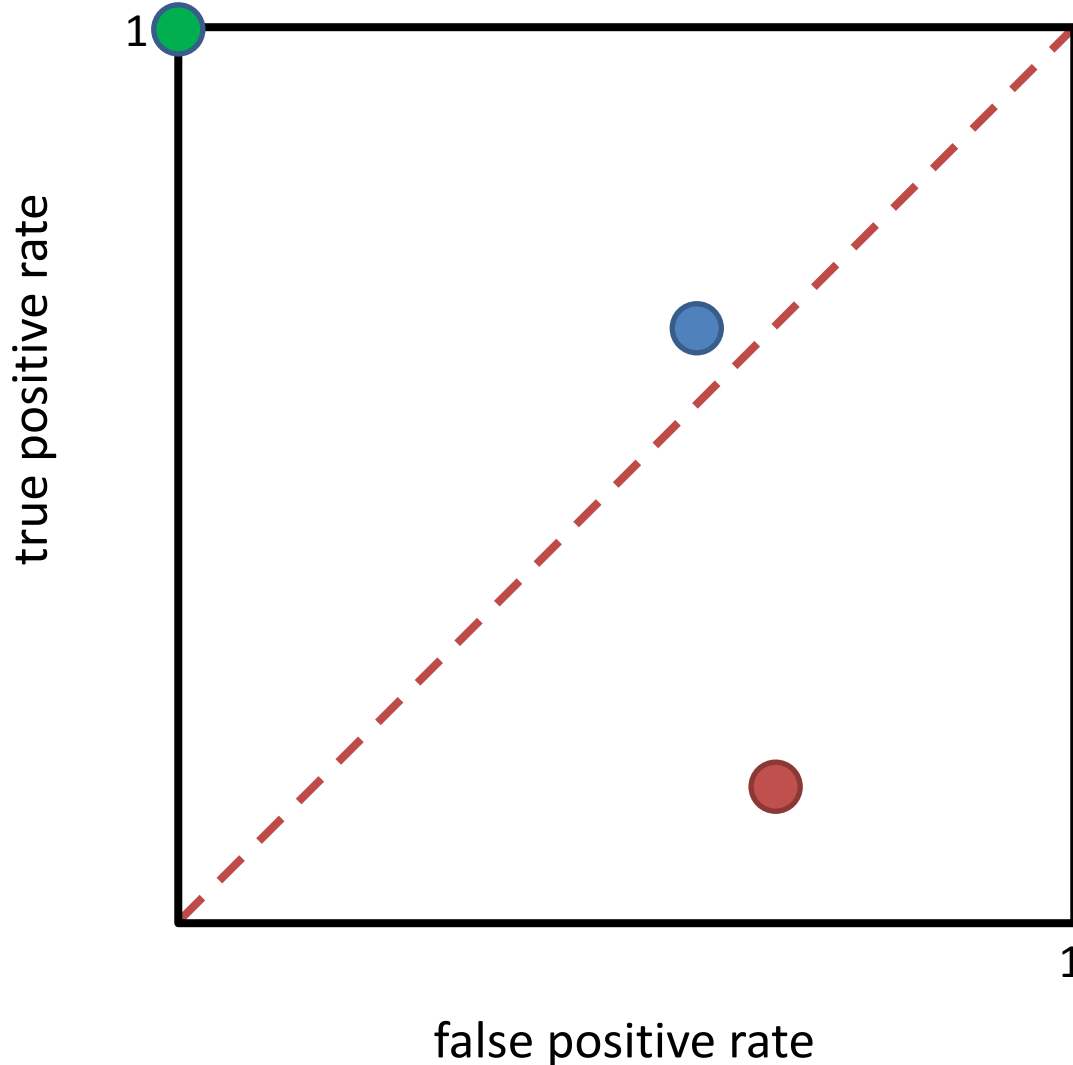
$$\frac{tp}{tp + fn}$$

- False Positive Rate:

$$\frac{fp}{fp + tn}$$

		predicted	
		1	-1
true	1	tp	fn
	-1	fp	tn

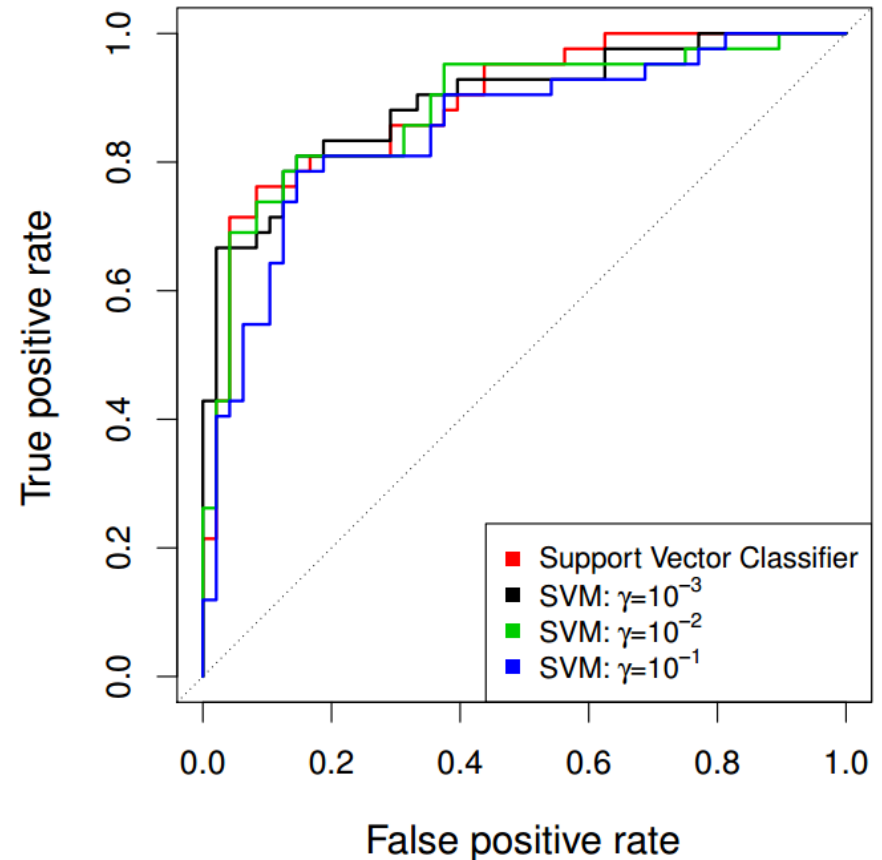
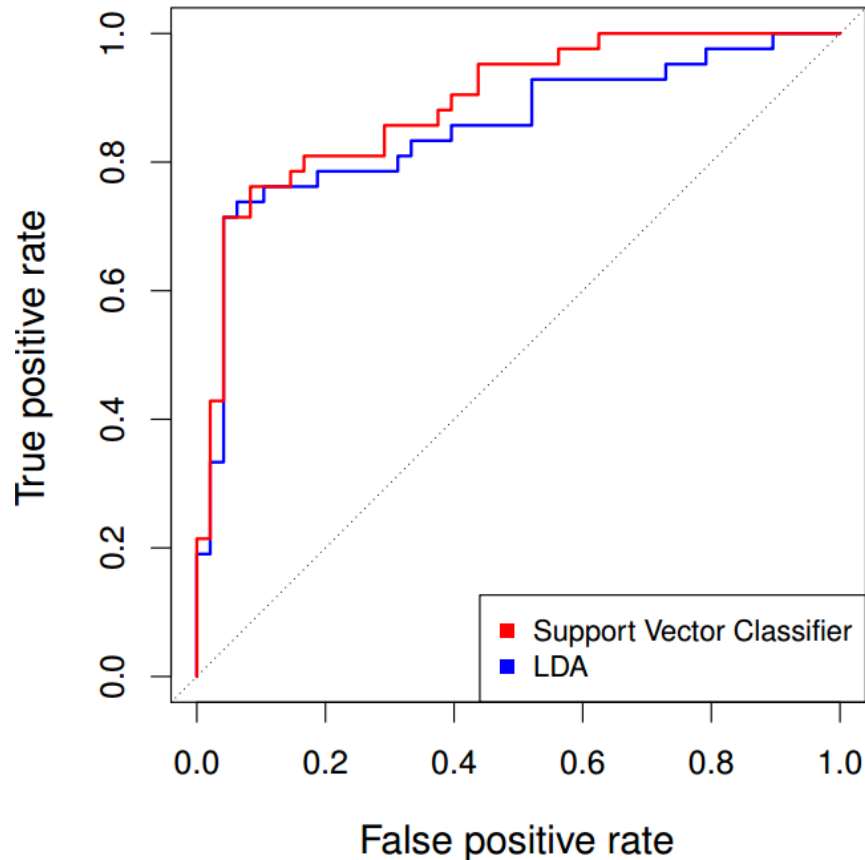
Receiver Operating Characteristic



Pros of ROC

- TPR and FPR are intuitive just from their names
- The random baseline is always given
- It is also valid for imbalanced data sets

ROC Curve for SVM ?



What they did

- Get the functional margin for all data points
- Threshold according to the margin

$$w^T x + b = \sum_{i=1}^m \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b.$$

Why ROC curves not points?

Textbook, page 37-40:

“ROC curves are useful for comparing different classifiers, since they take into account all possible thresholds.

Precision Recall

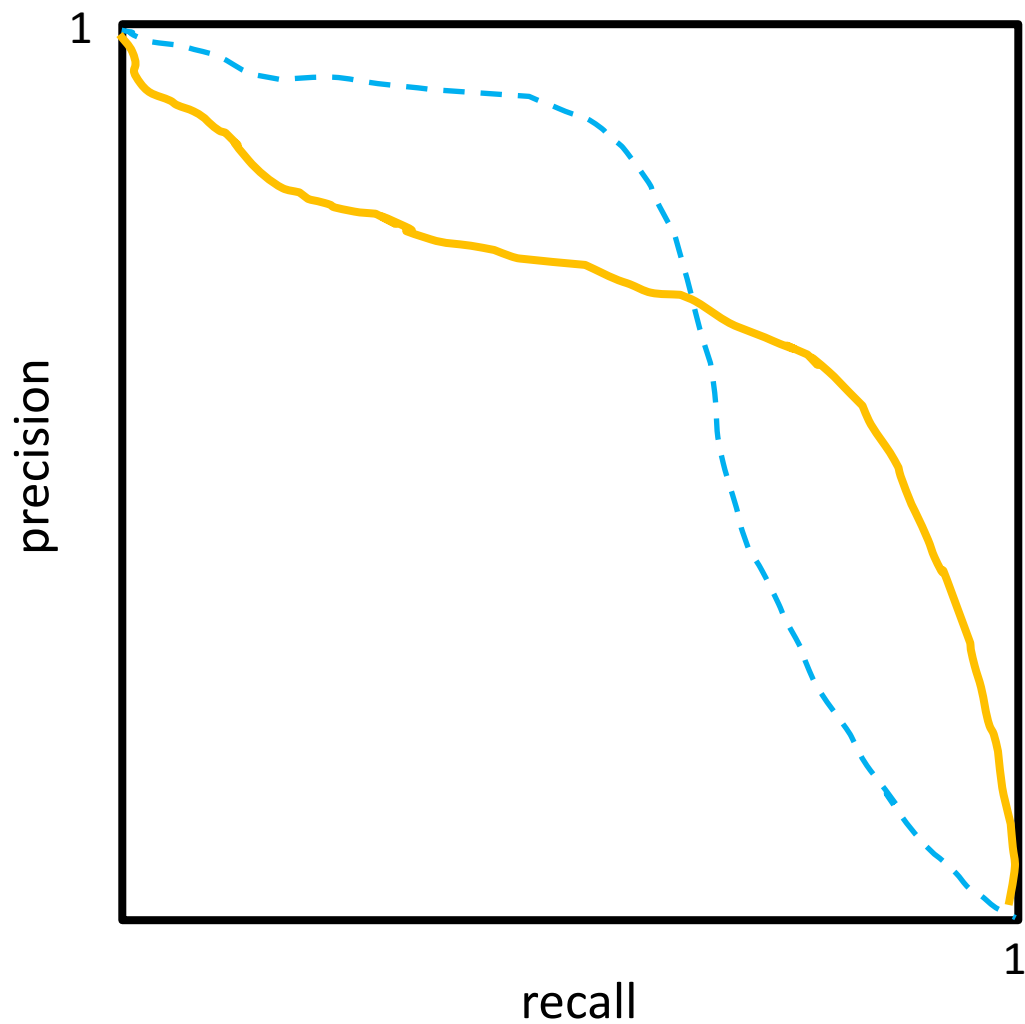
- Recall: $\frac{tp}{tp + fn}$
- Precision: $\frac{tp}{tp + fp}$

		predicted	
		1	-1
true	1	tp	fn
	-1	fp	tn

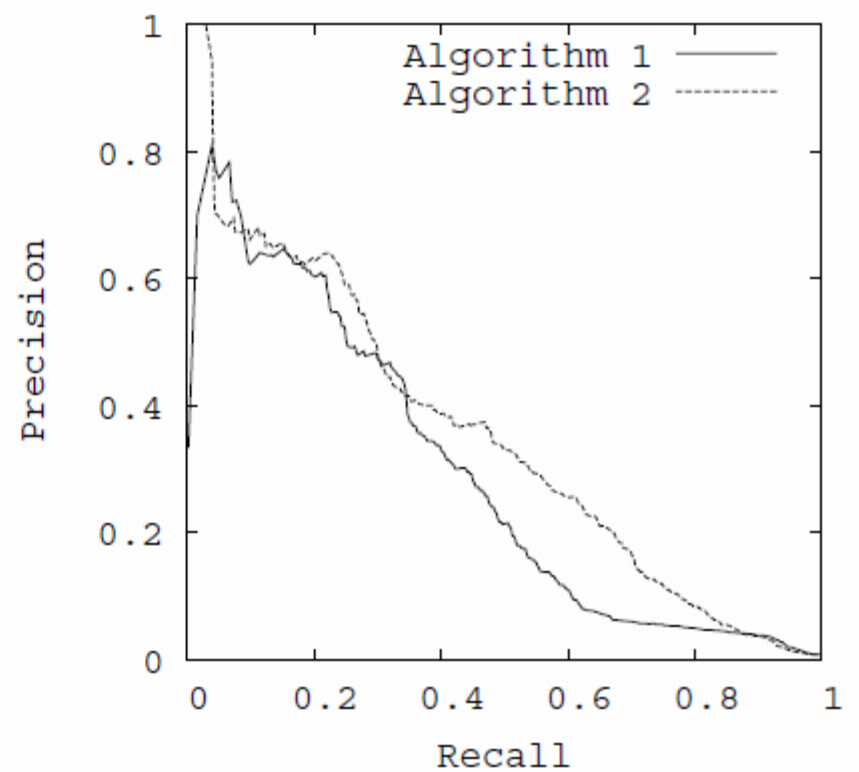
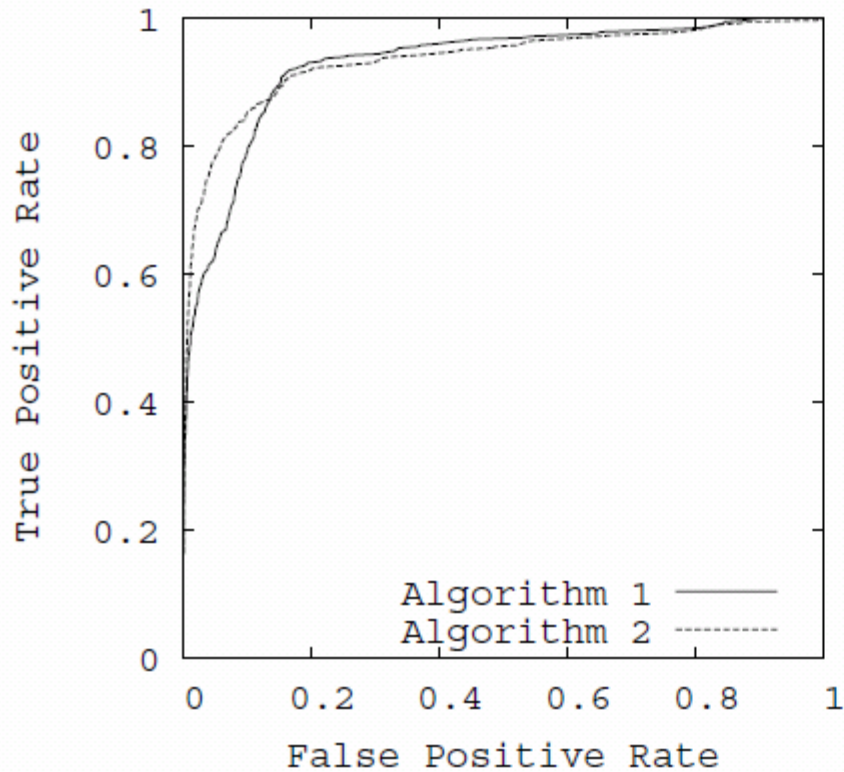
Precision Recall

- **Recall:** If I pick a random positive example, what is the probability of making the right prediction?
- **Precision:** If I take a positive prediction example, what is the probability that it is indeed a positive example?

Precision Recall Curve



Comparison



J. Davis & M. Goadrich,
“The Relationship Between Precision-Recall and ROC Curves.”,
ICML (2006)

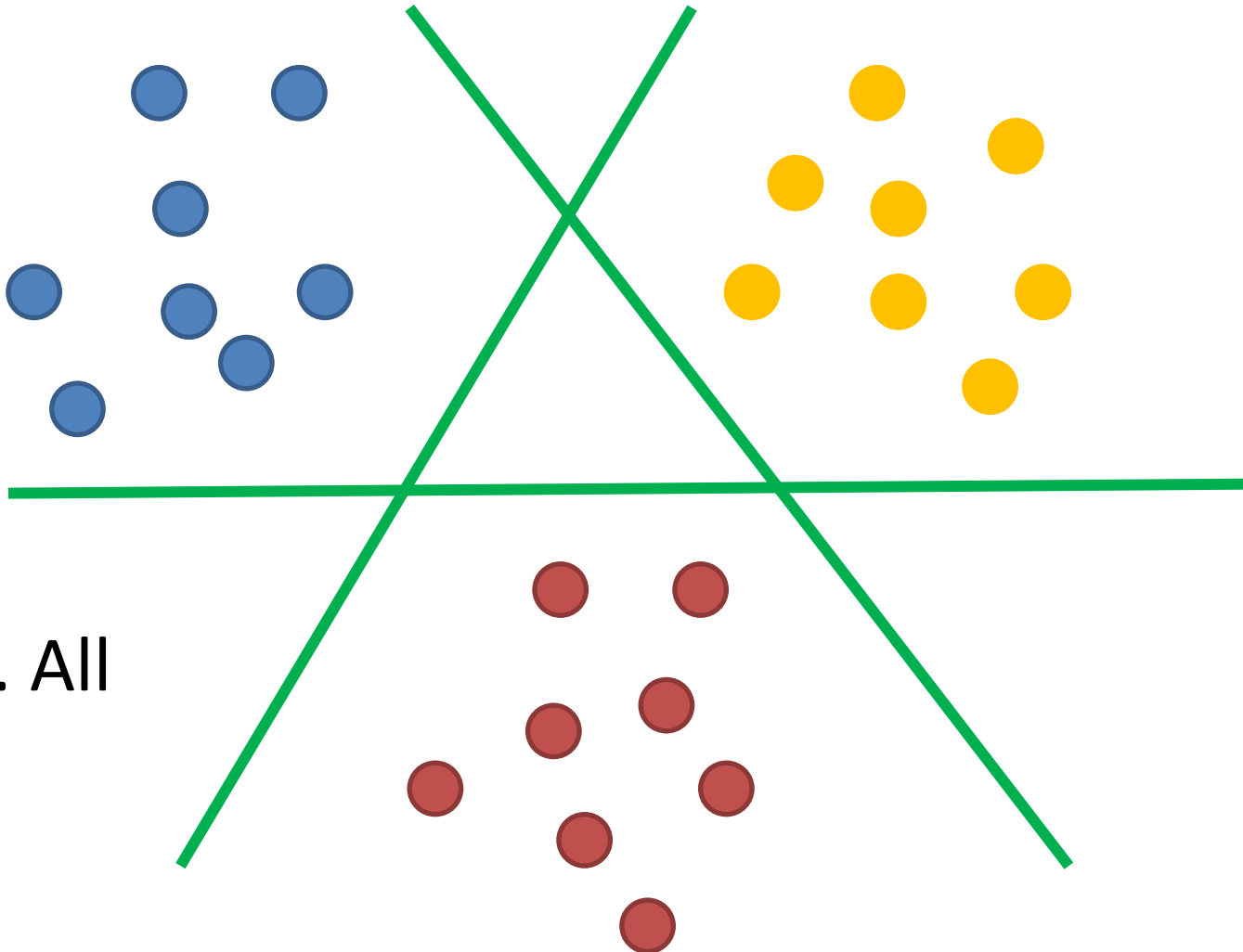
F-measure

- Weighted average of precision and recall

$$F_{\beta} = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

- Usual case: $\beta = 1$
- Increasing β allocates weight to recall

Multi Class

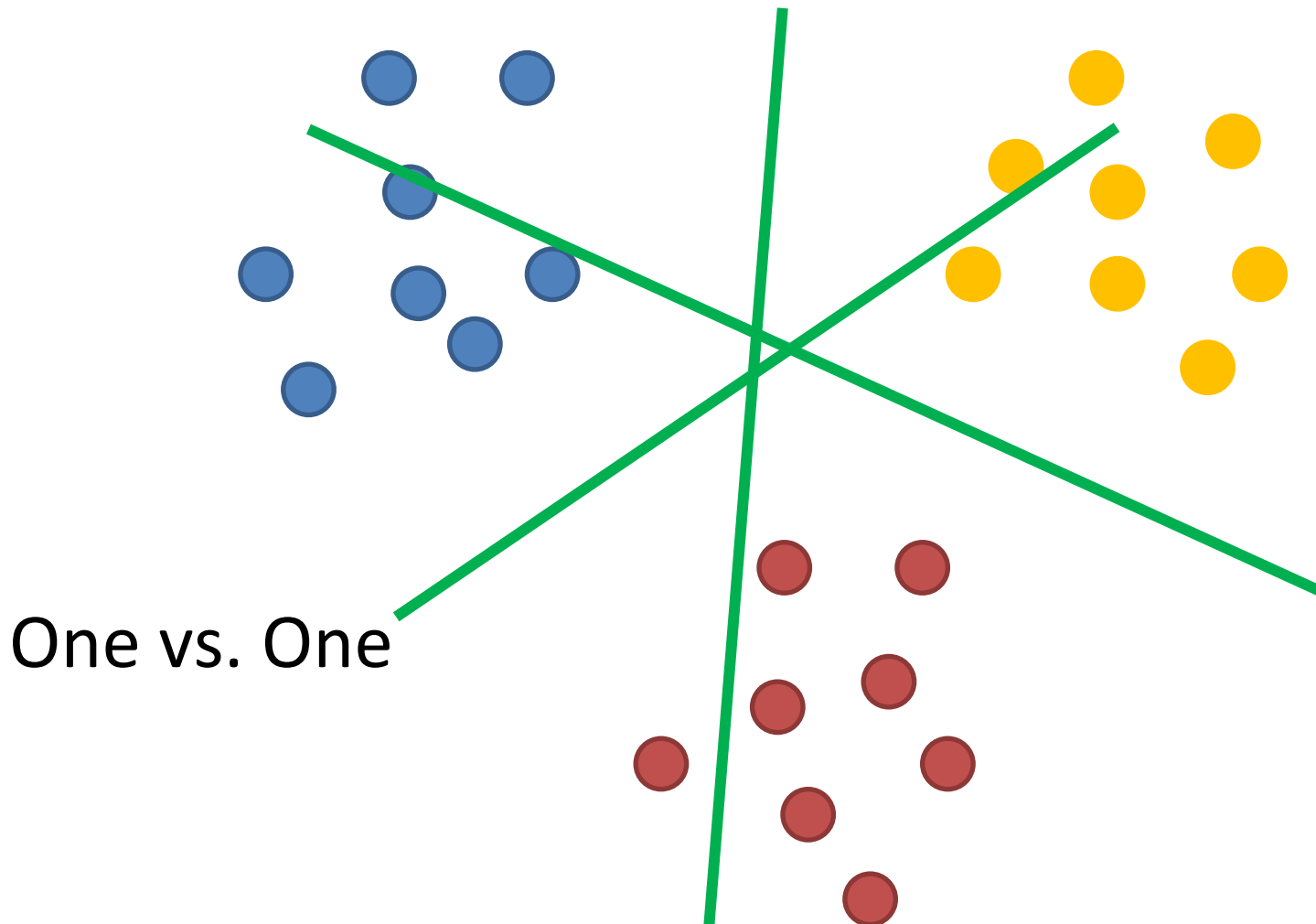


One vs. All

One vs All

- Train n classifier for n classes
- Take classification with greatest margin
- Slow training

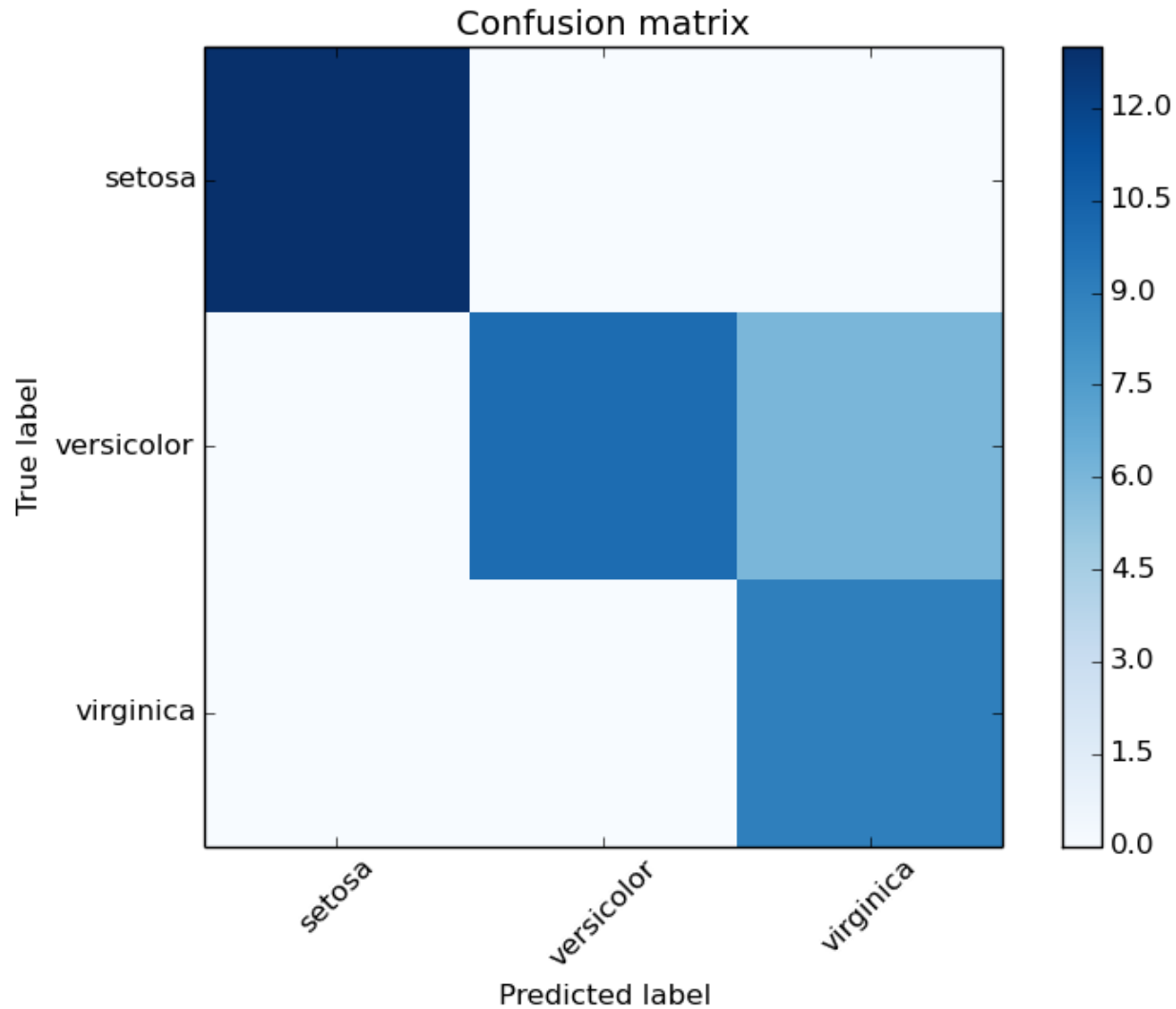
Multi Class



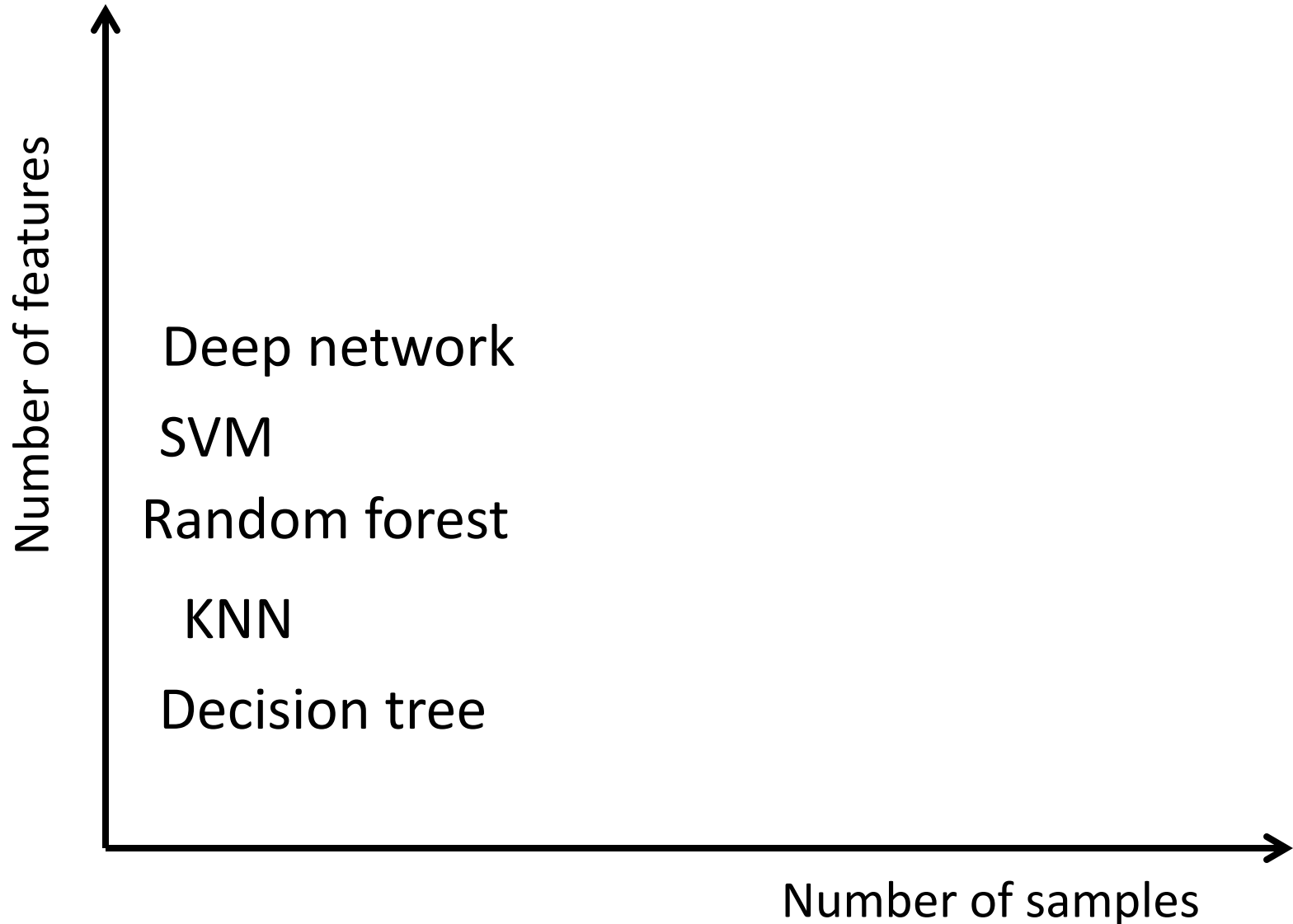
One vs One

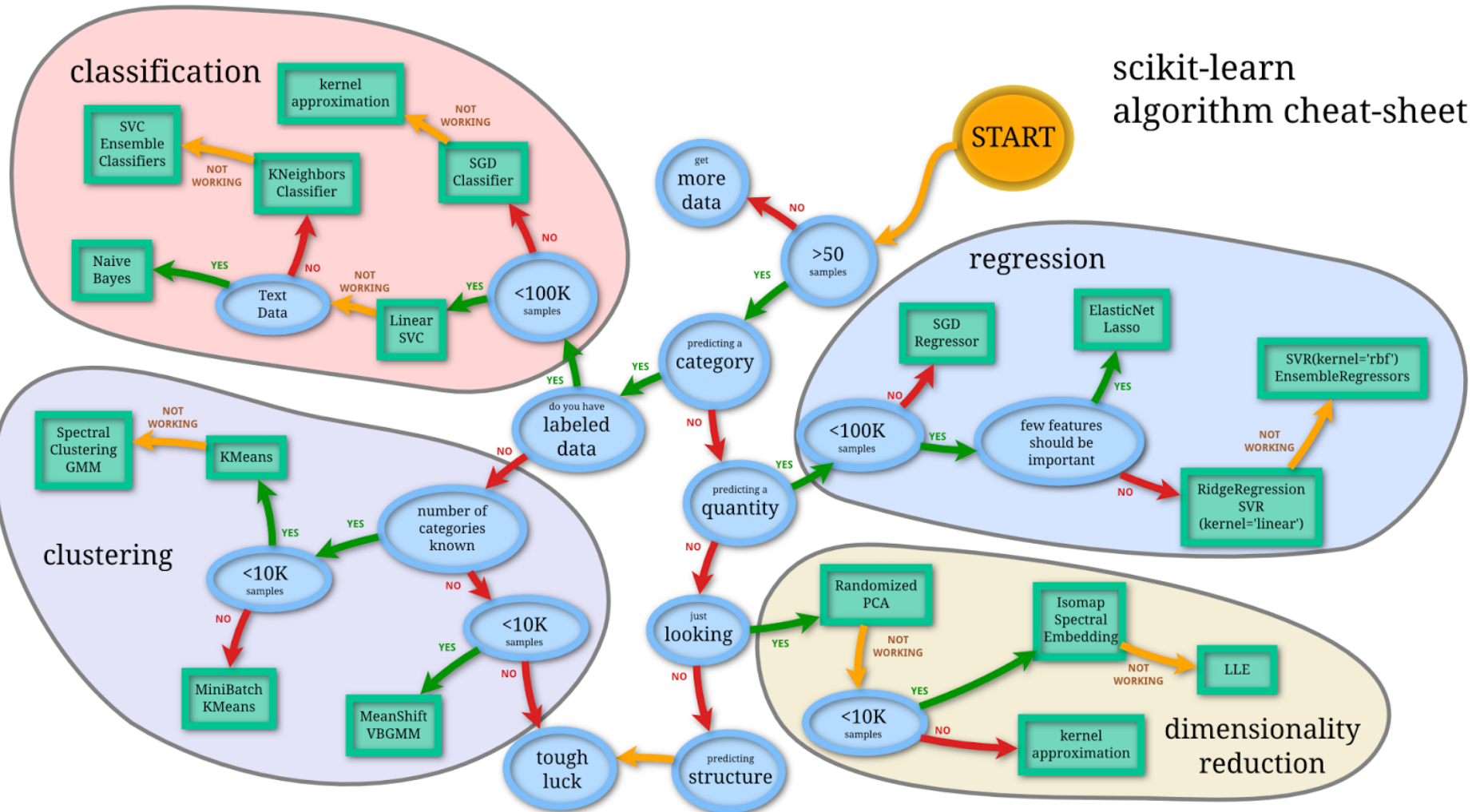
- Train $n(n-1)/2$ classifiers
- Take majority vote
- Fast training

Confusion Matrix



Which classifier for what?



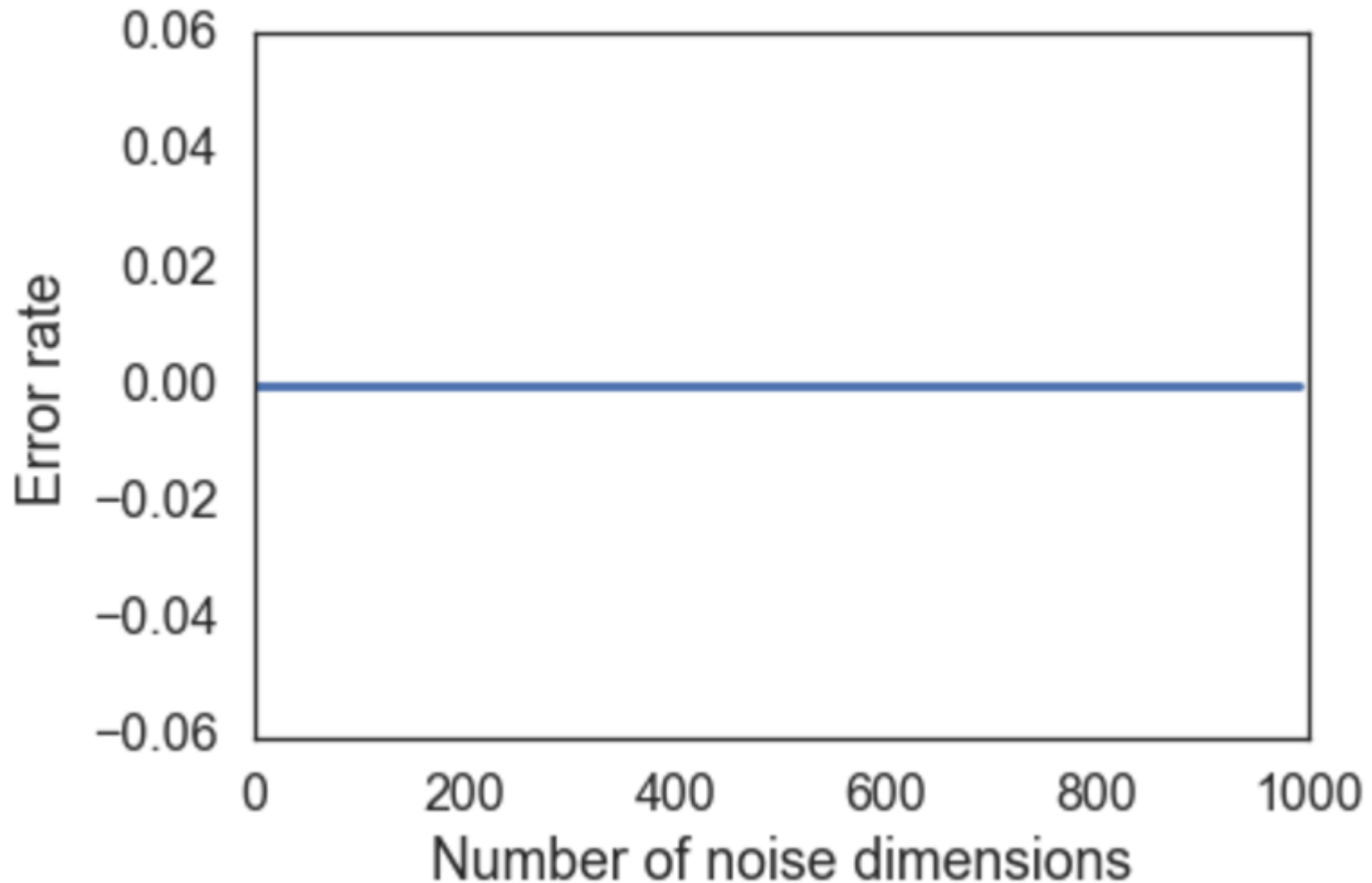
scikit-learn
algorithm cheat-sheet

Extreme Scenario

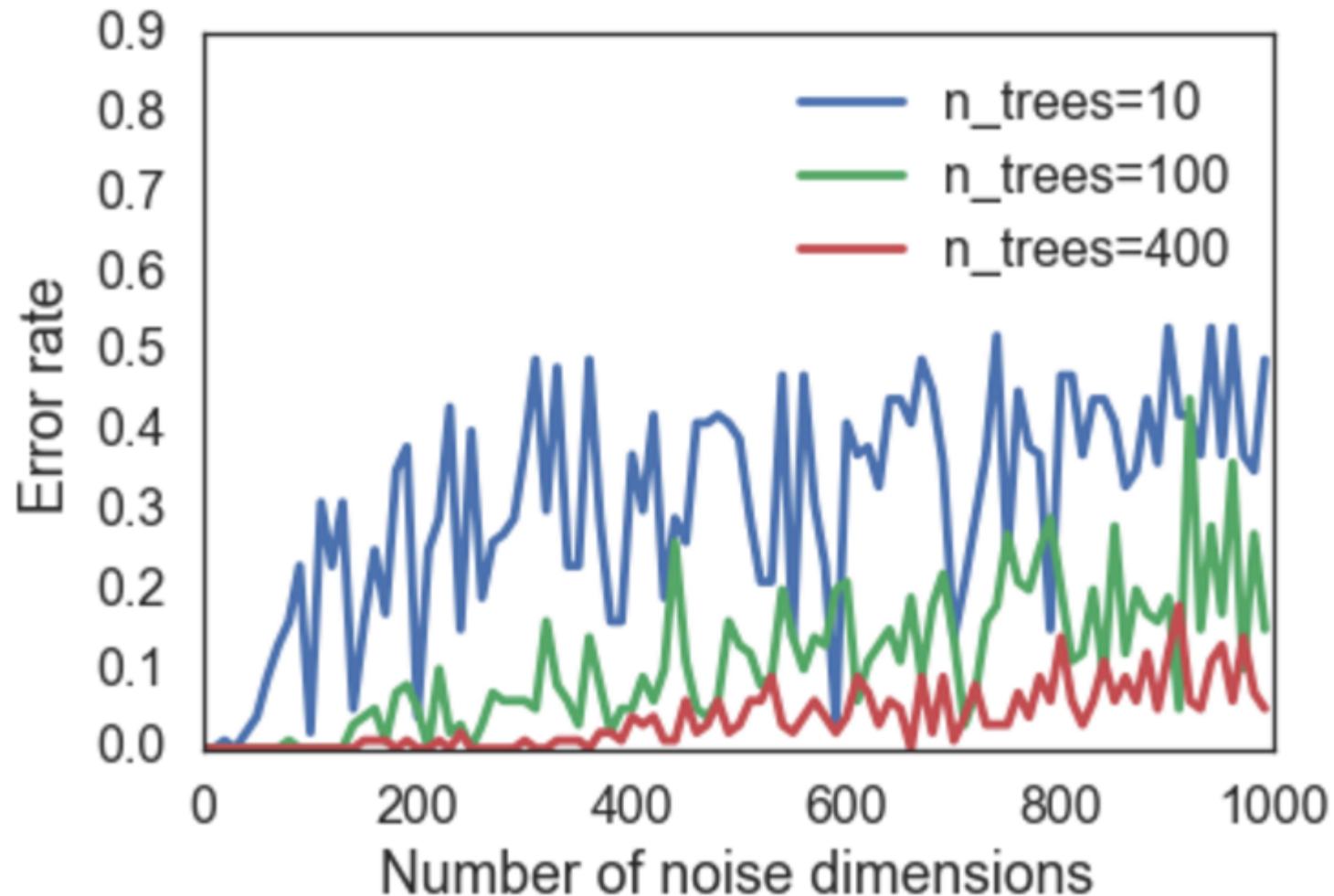
y	x			
1	1	@	&	...
0	0	#	%	...
1	1	\$	#	...
1	1	%	^	...
0	0	^	!	...
1	1	*)	...
0	0)	%	...
...

How would which classifier deal with this problem?
What if the number of noise dimensions increases?

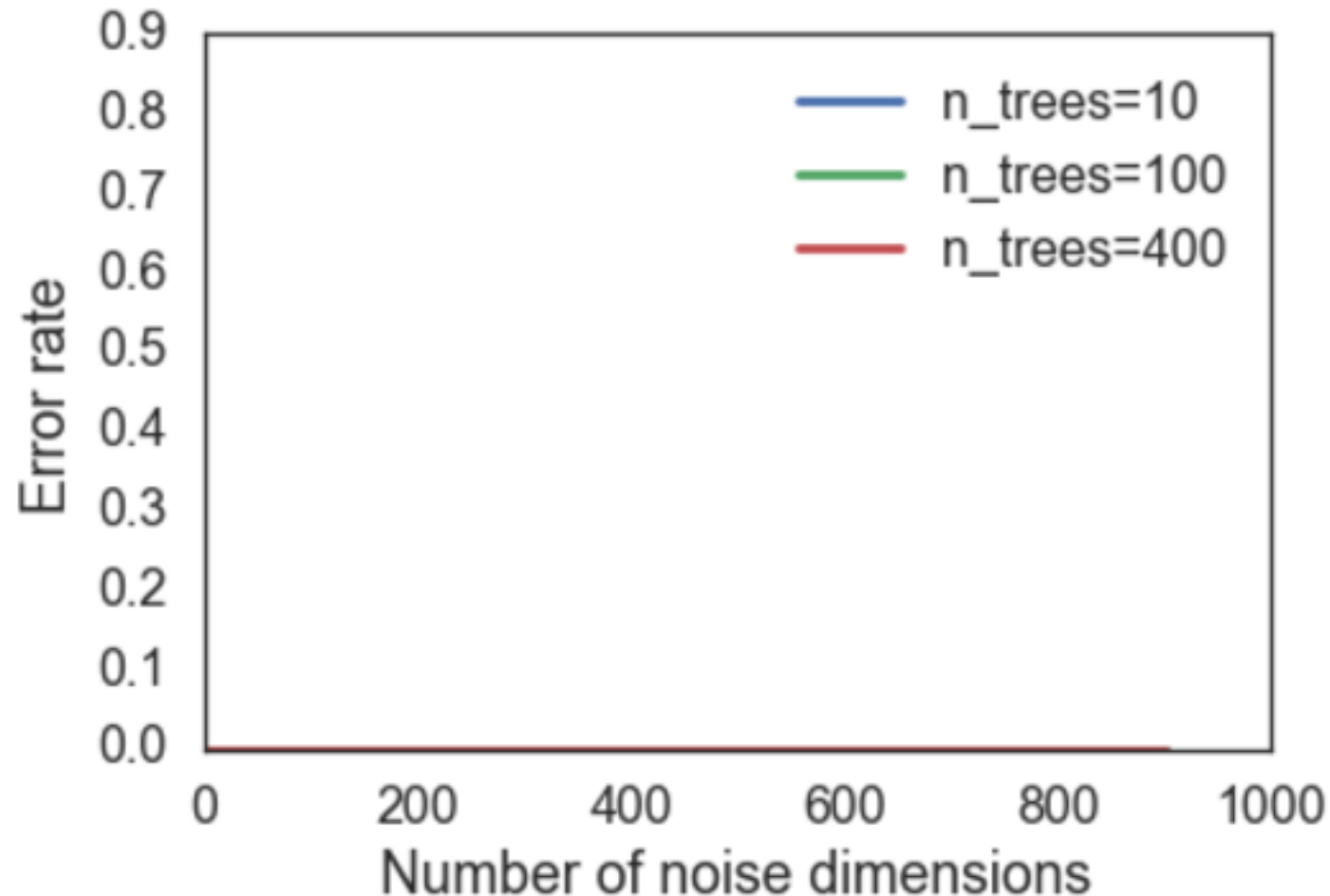
Noise Robustness – Decision Tree



Noise Robustness – Random Forest



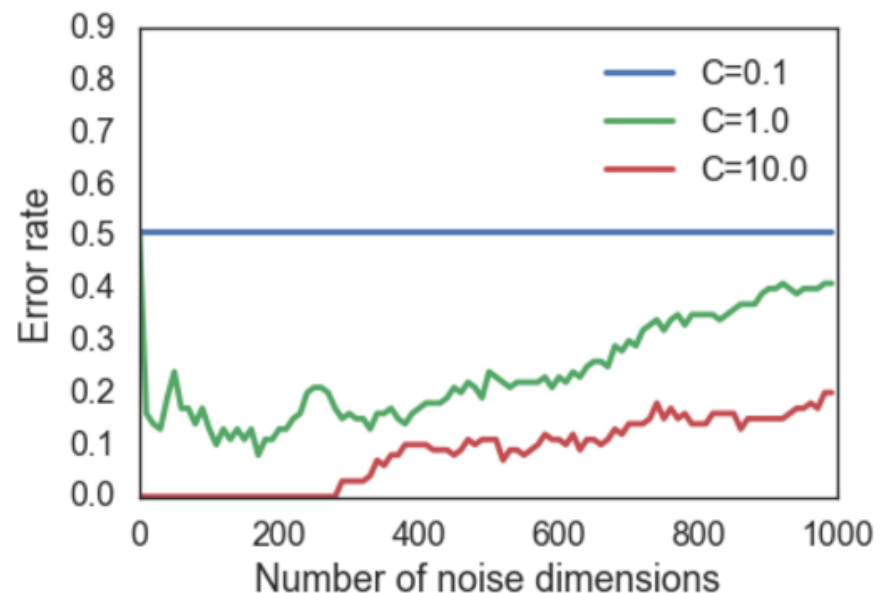
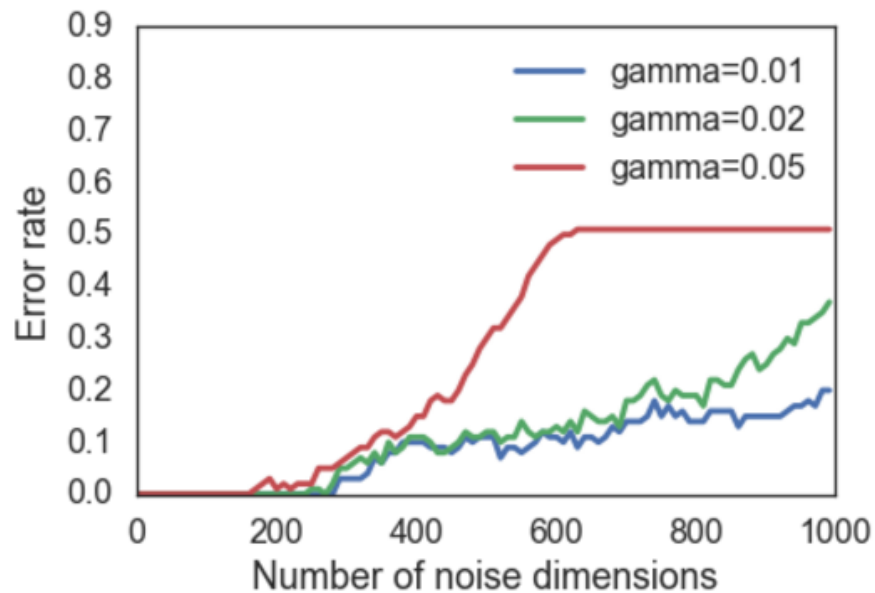
Noise Robustness – Random Forest



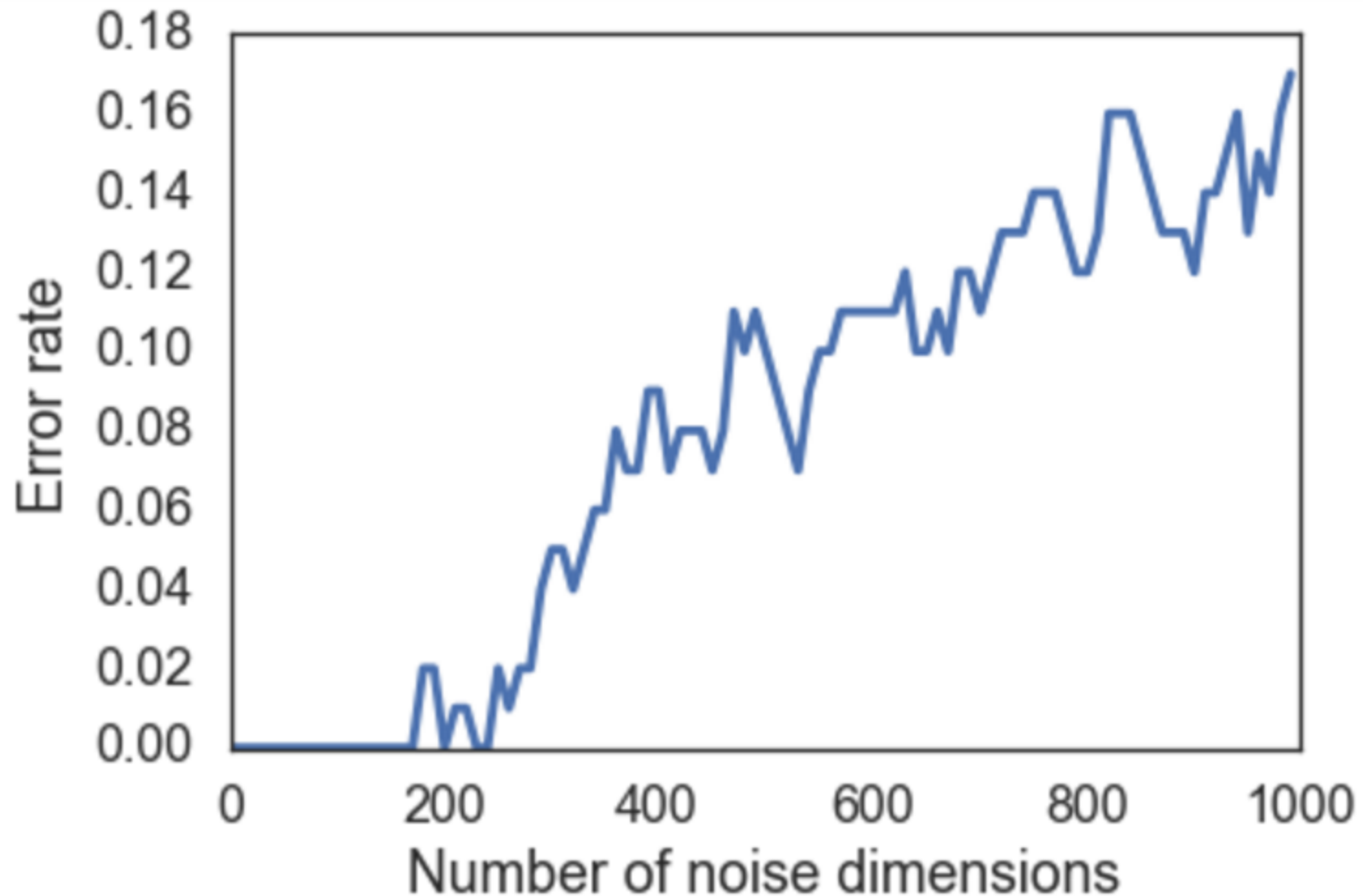
Random Forest Error Rate

- Error depends on:
 - Correlation between trees (higher is worse)
 - Strength of single trees (higher is better)
- Increasing number of features for each split:
 - Increases correlation
 - Increases strength of single trees

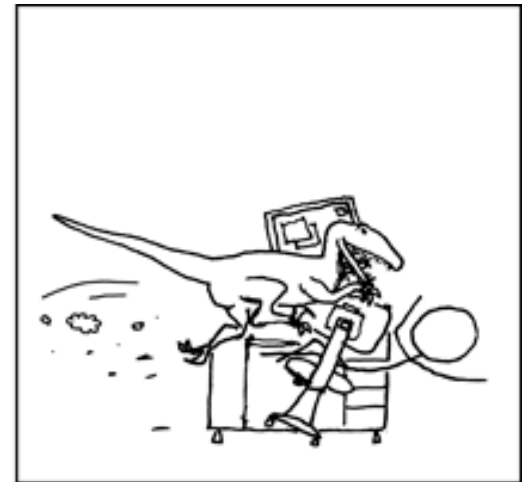
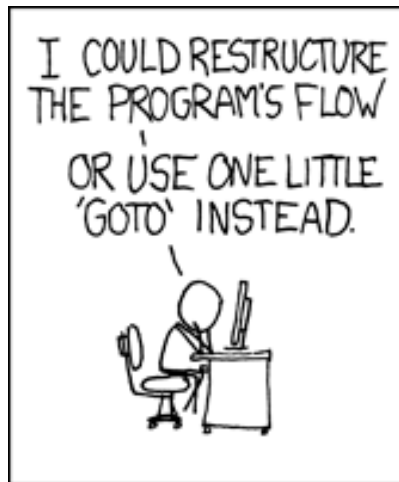
Noise Robustness - SVM



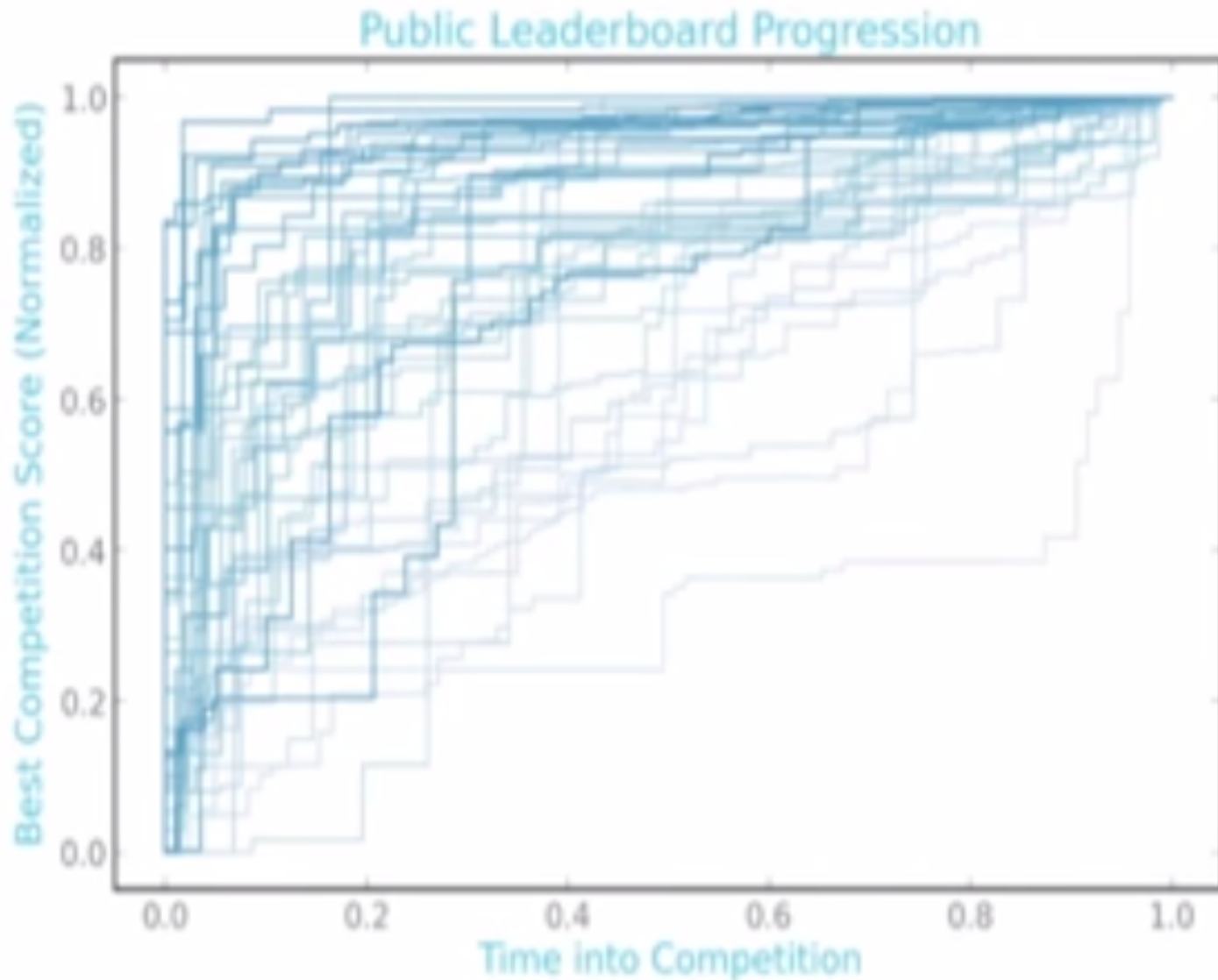
Noise Robustness – Logistic Regression



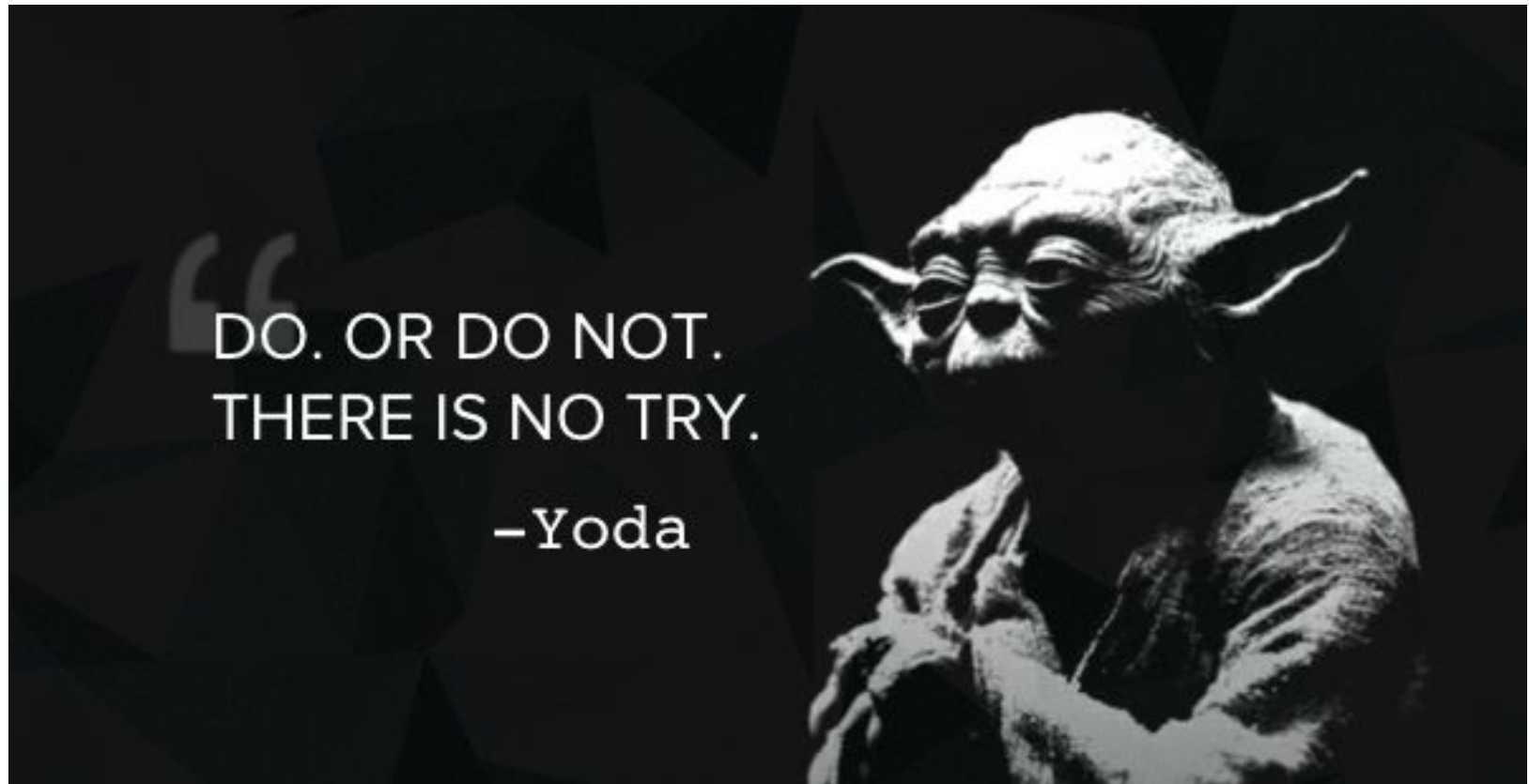
Best Practices



Typical Progress



Under Promise, Over Deliver!



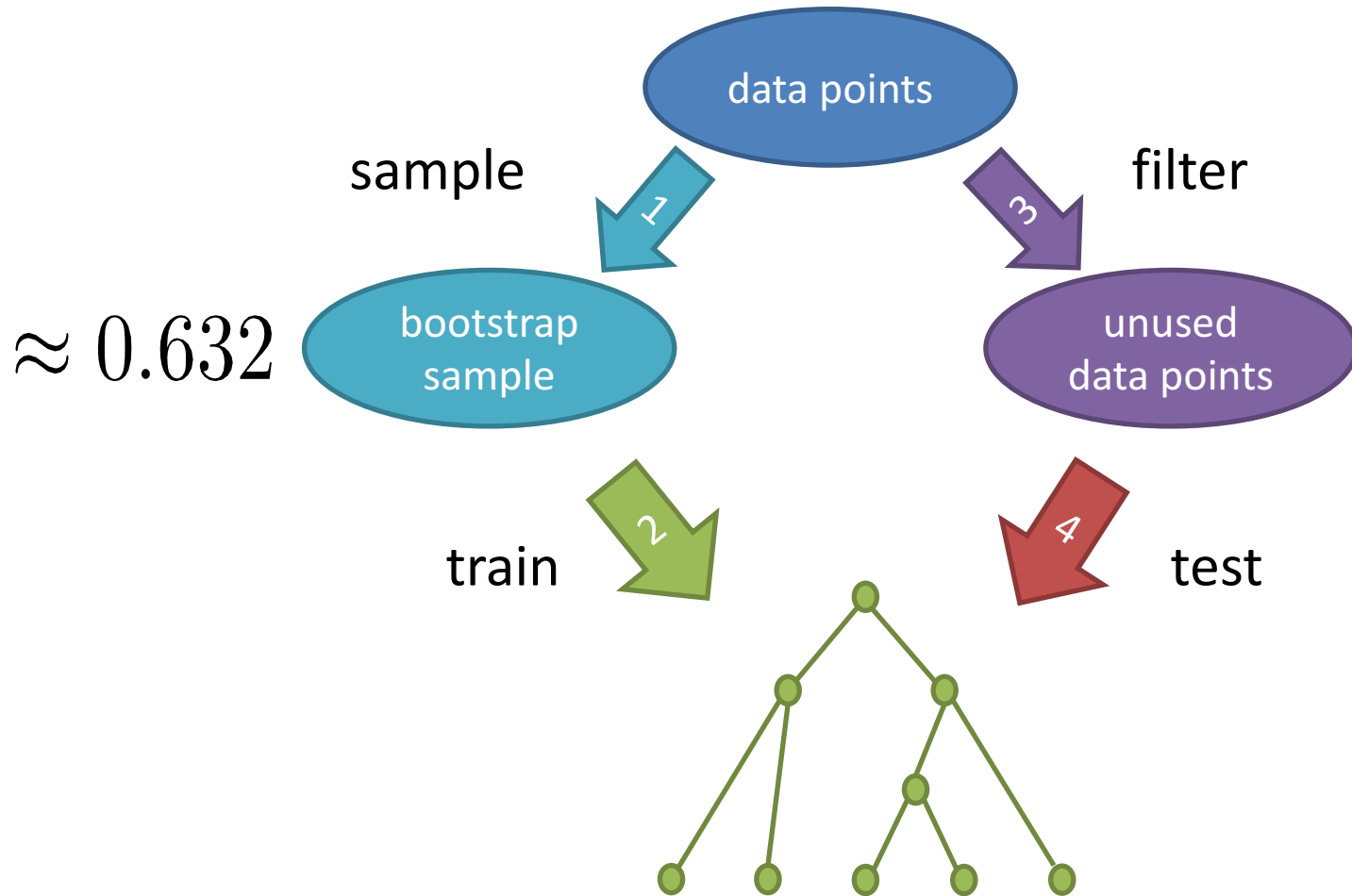
General Best Practices

- it will be harder than it looks
- know your application:
 - zero values
 - outliers
 - where do labels come from
- Document, document, document
 - for yourself! And for others
- commit, pull, push, repeat

From a Kaggle Forum

... it feels weird to be using cross-validation type methods with random forests since they are already an ensemble method using random samples with a lot of repetition. Using cross-validation on random forests feels redundant.

Out of Bag Error



Cross Validation



training
data







validation
data



test
data

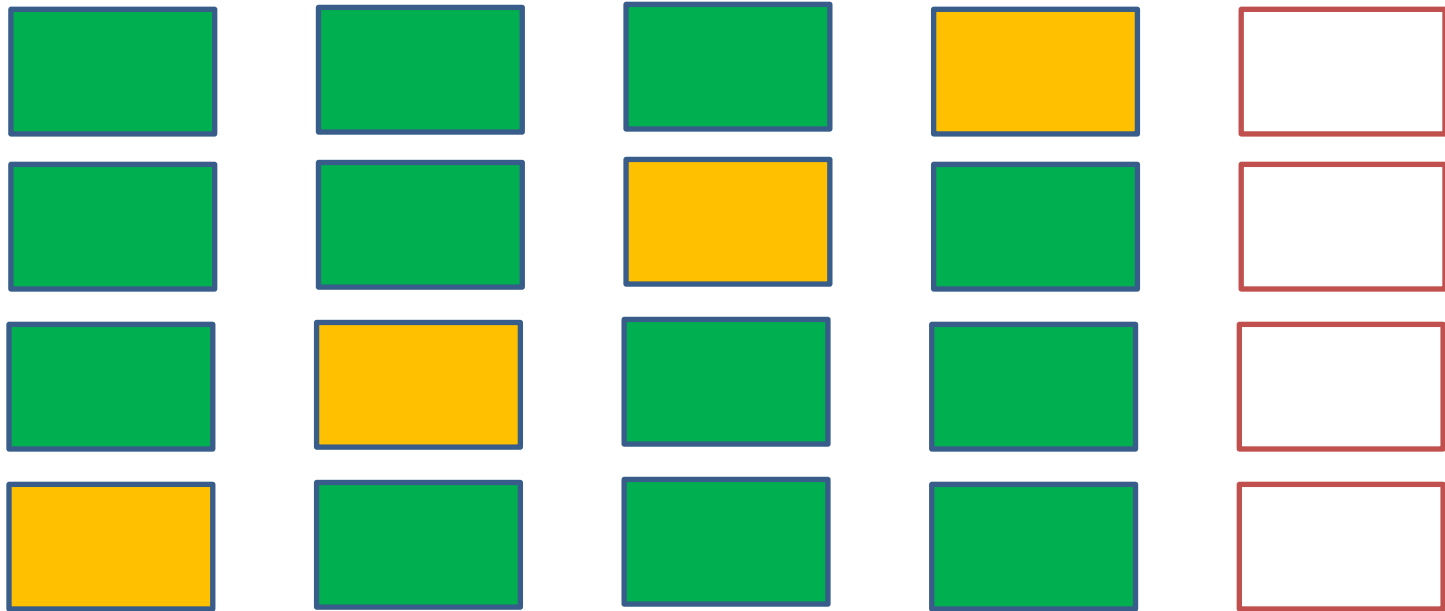
- Training data: train classifier
 - Validation data: estimate hyper parameters
 - Test data: estimate performance
-
- Be mindful of validation and test set, validation set might refer to test set in some papers.

Last Step of Each Fold

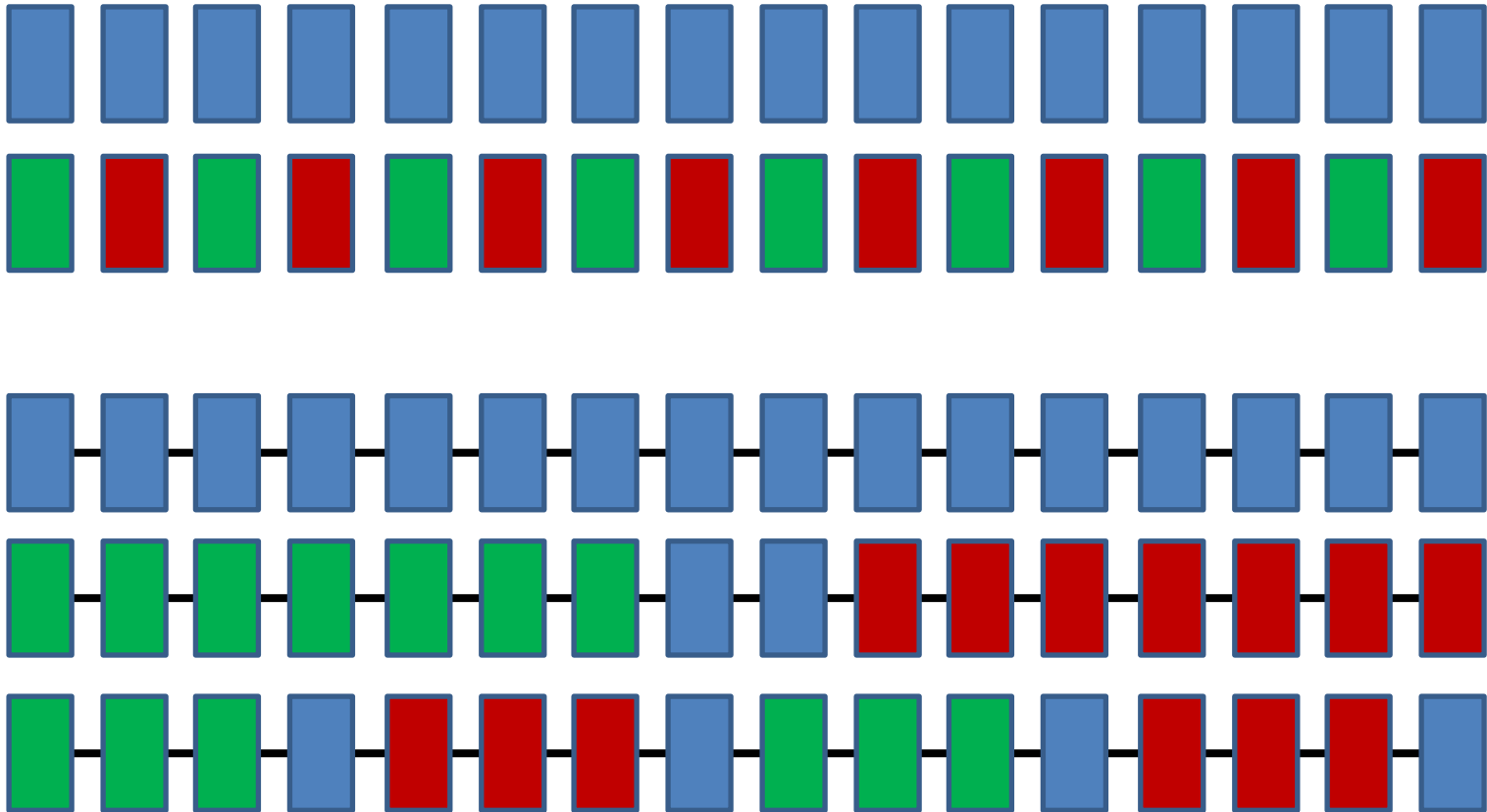
1. Take best parameters 
2. Train on training data and validation data together  
3. Test performance on test data 

This is the **final** result of your method.

5 – Fold Cross Validation



Know Your Data



Normalization

- Be very careful.
- Do not leak into the test data.
- Think about what is useful.

Normalization - 1



training



Estimate
mean
values and
normalize.



validation



Estimate
mean
values and
normalize.



test



Estimate
mean
values and
normalize.

Normalization - 2



training



validation

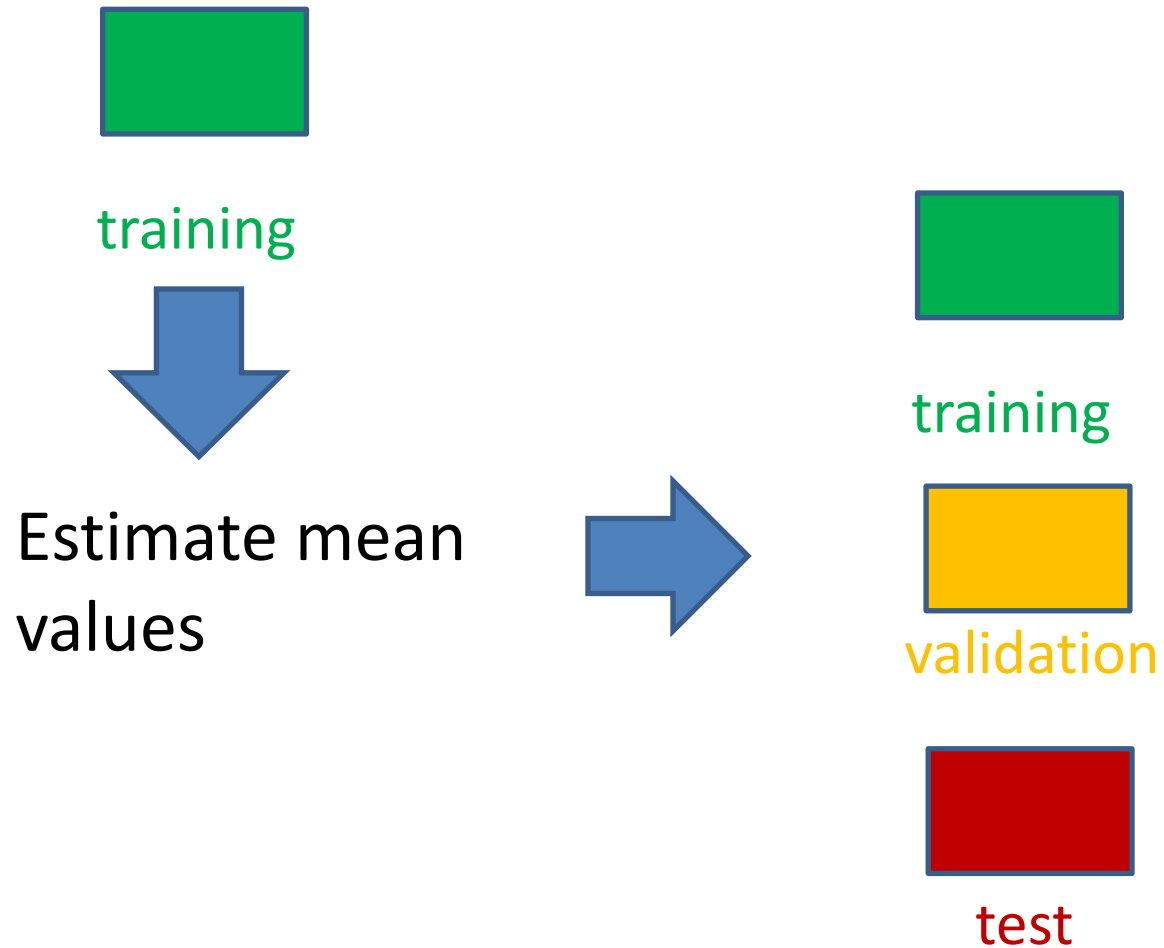


test



Estimate
mean
values and
normalize.

Normalization - 3



Scenario - 1

- 1. Screen the predictors: find a subset of “good” predictors that show fairly strong (univariate) correlation with the class labels
- 2. Using just this subset of predictors, build a multivariate classifier.
- 3. Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.

Scenario - 2

- 1. Divide the samples into K cross-validation folds (groups) at random.
- 2. For each fold $k = 1, 2, \dots, K$
 - Find a subset of “good” predictors that show fairly strong (uni-variate) correlation with the class labels, using all of the samples except those in fold k .
 - Using just this subset of predictors, build a multivariate classifier, using all of the samples except those in fold k .
 - Use the classifier to predict the class labels for the samples in fold k .

Summary

- Data normalization: Recommended for SVM, KNN, KLR, ...
- Evaluation: ROC vs Precision/Recall
- Have more than a hammer in your toolbox
- Under promise, over deliver!
- Really know your data and cross validation