Algorithm Fitting & Tuning

Data Mining for Business and Governance*
19/09/2017

**Formerly known as Social Data Mining*

Course Schedule

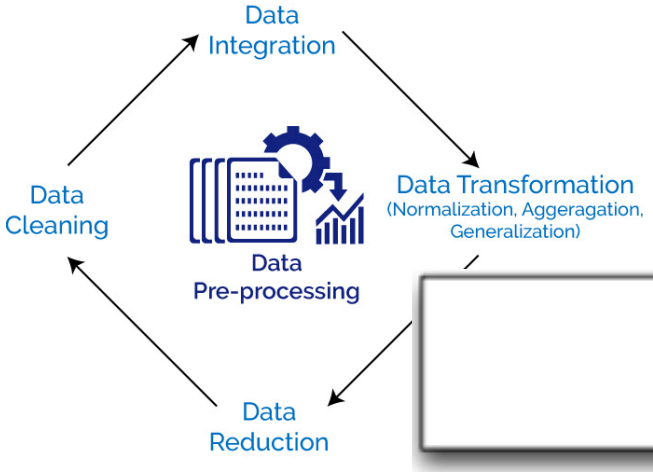
`v05.09.2017` (subject to change – always check the latest version!)

#	Date	Lectures (Theory - Willem)	Date	Video Lectures (Applications - Chris)	Video Practicals & Notebooks
1	29-08	Introduction to Data Mining	31-08	Introduction to Data Science	Introduction to jupyter, pandas, and scikit-learn
2	05-09	Regression	07-09	Representing Data: Vectors, Types, Databases	Handling & Interpreting Data, Plotting
3	12-09	Classification	14-09	Working with Text Data Part 1 (17-09)	DIY Pandas + scikit-learn
4	19-09	Algorithm Fitting & Tuning	21-09	Working with Text Data Part 2	**No practical** -> time to prepare for midterm.
5	26-09	**Midterm**	28-09	Best Practices, Common Pitfalls & Research	Preprocessing + Pipelines, MNIST Challenge
6	03-10	Data Reduction & Decomposition	05-10	Mining Massive Data, Ensemble Methods	Online / Out-of-Core Learning
7	10-10	Clustering and Graphs	12-10	Applications of Deep Learning	Social Media and Multi-modal Data
8	17-10	Time Series Analysis	19-10	Explaining Models, Ethics, Privacy	Unsupervised Learning: Intuitions and Metrics

Overview: fitting & tuning

- **“The Data Mining Procedure”**
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- ROC curves
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

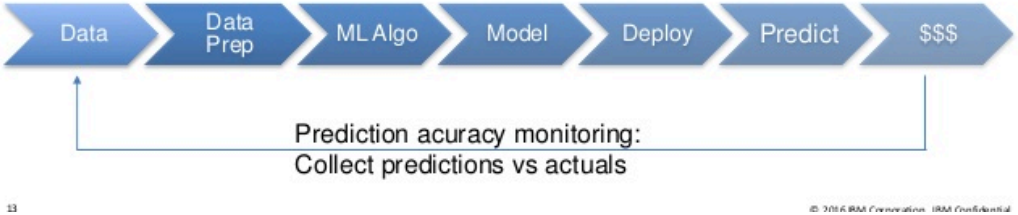
Data Mining Procedure



IBM Analytics

IBM

- **The Machine Learning Workflow**
- Feedback loop

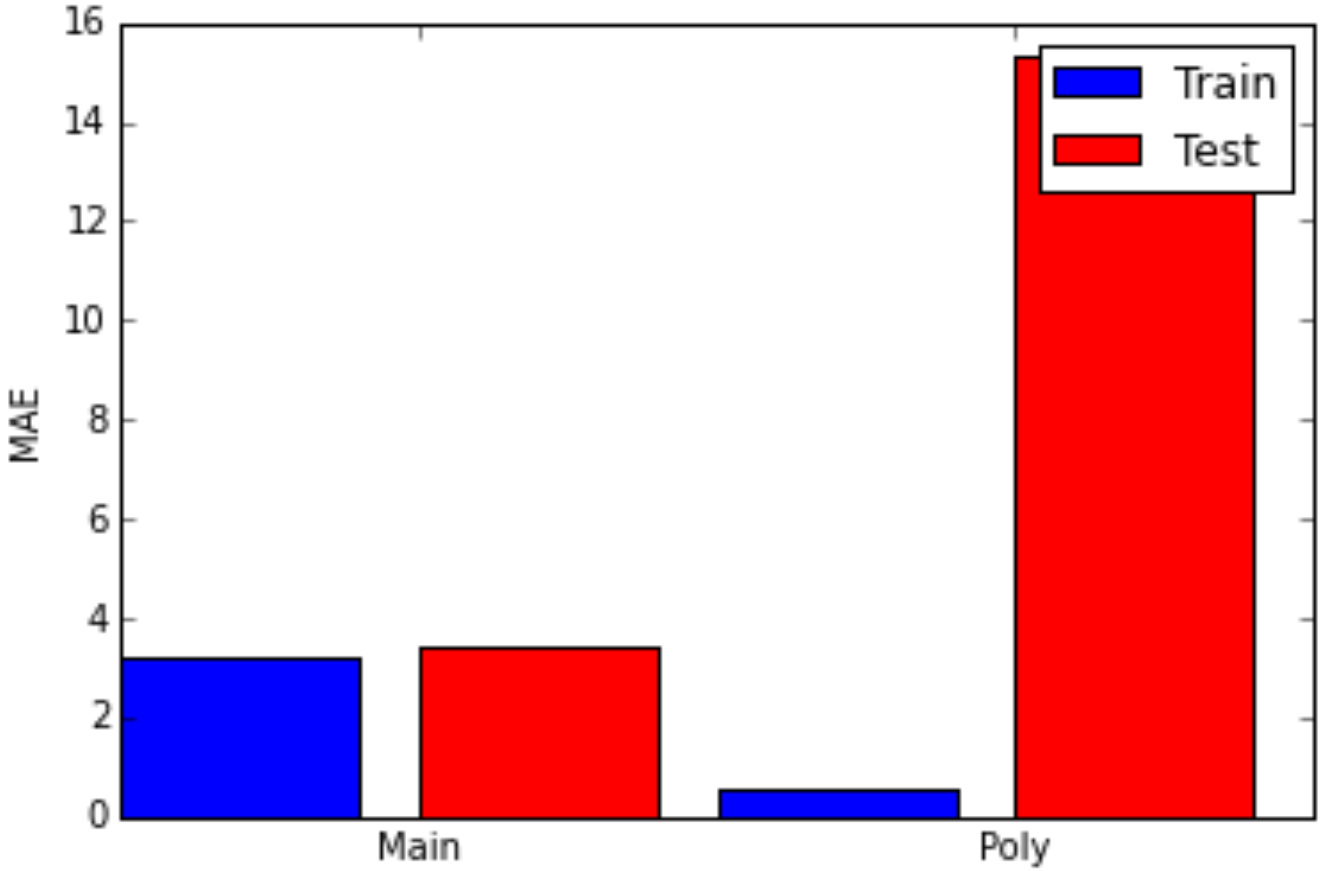


Boston house prices

- CRIM per capita crime rate by town
- ZN proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS proportion of non-retail business acres per town
- CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX nitric oxides concentration (parts per 10 million)
- RM average number of rooms per dwelling
- AGE proportion of owner-occupied units built prior to 1940
- DIS weighted distances to five Boston employment centres
- RAD index of accessibility to radial highways
- TAX full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT % lower status of the population
- MEDV Median value of owner-occupied homes in $1000's

Target: Housing Price

Two linear regression models



Too much training?

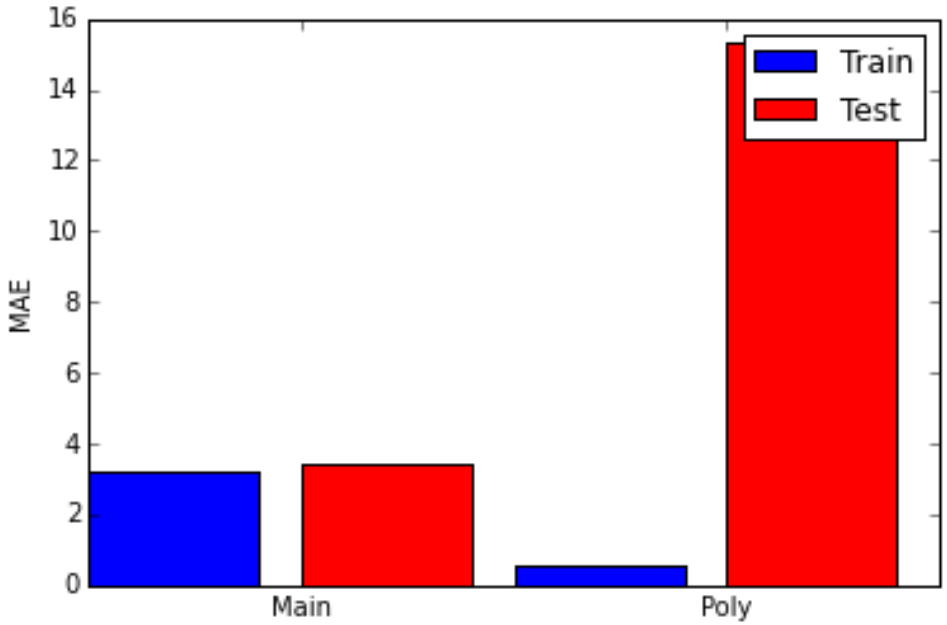
Fit training data very closely

Able to predict very well on training data

Doesn’t generalize well to the test set

- High error on test data

Boston house prices



Main effects

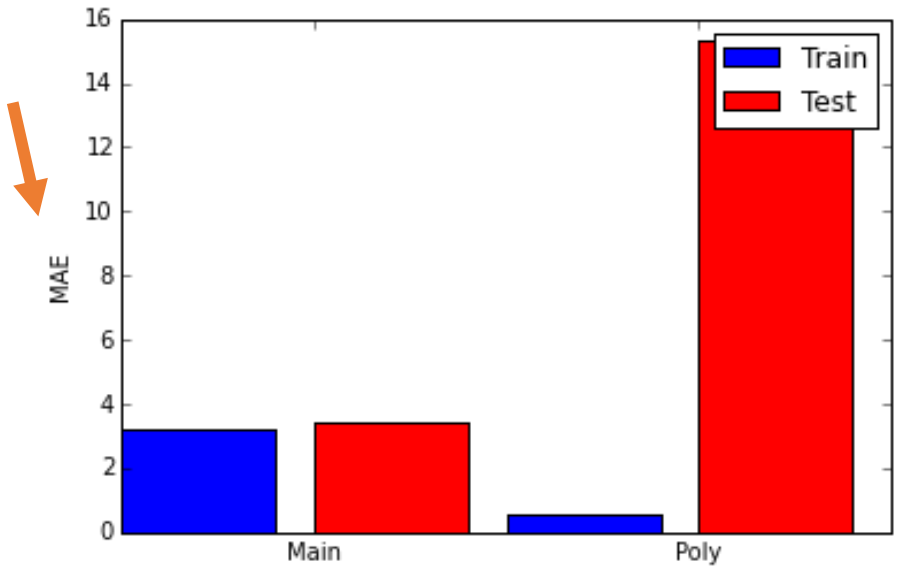
Y = [a b c]

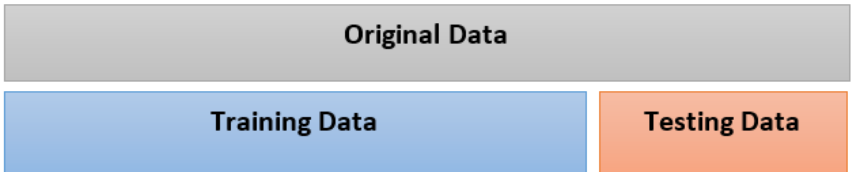
Poly: Main + Interactions

Y = [a b c ab ac abc]

Always easier to fit (and overfit) with more features

Determining the performance of our model

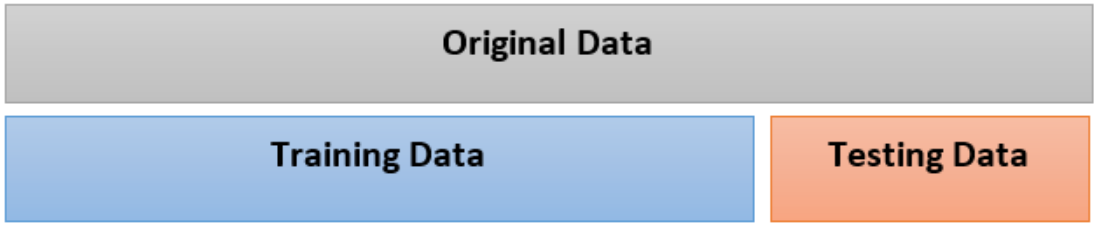




Evaluation: determining how well a model fits the data

In the standard (hold-out) framework we can only do this once

”The” Data Mining or Machine Learning Procedure

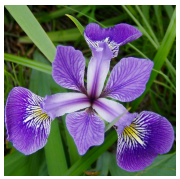


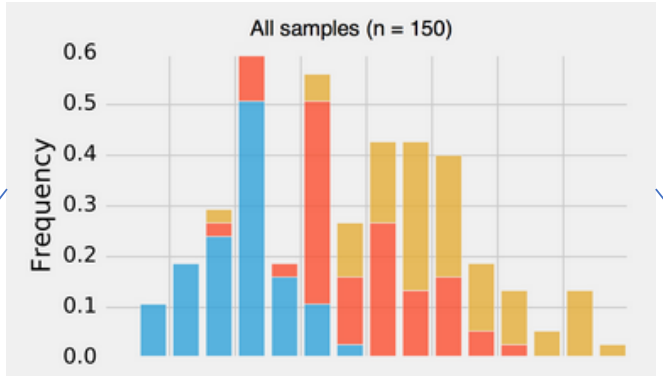
More than just decide on a Split: for example 2/3 vs 1/3

The Iris dataset: 50 Setosa, 50 Versicolor, and 50 Virginica flowers; the flower species are distributed uniformly:

- 33.3% Setosa
- 33.3% Versicolor
- 33.3% Virginia

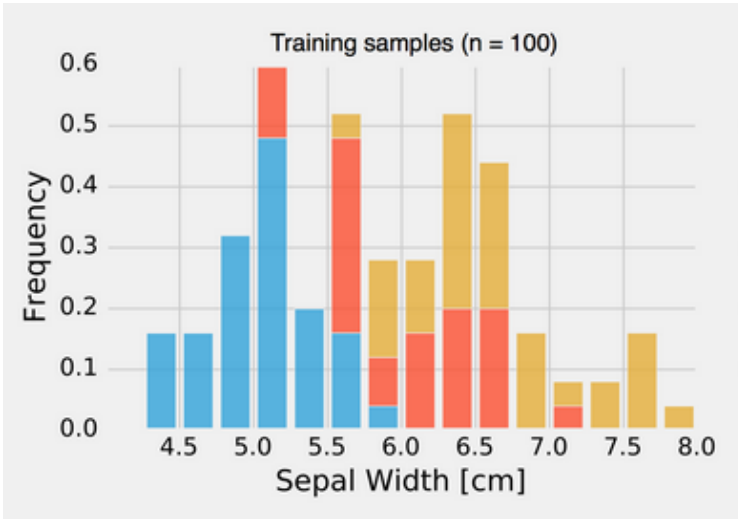
Splitting data: Stratification

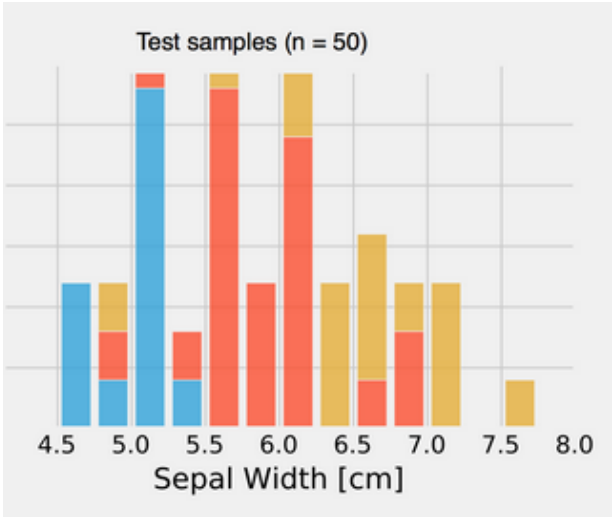




Training Set

Test Set





Bias vs Variance

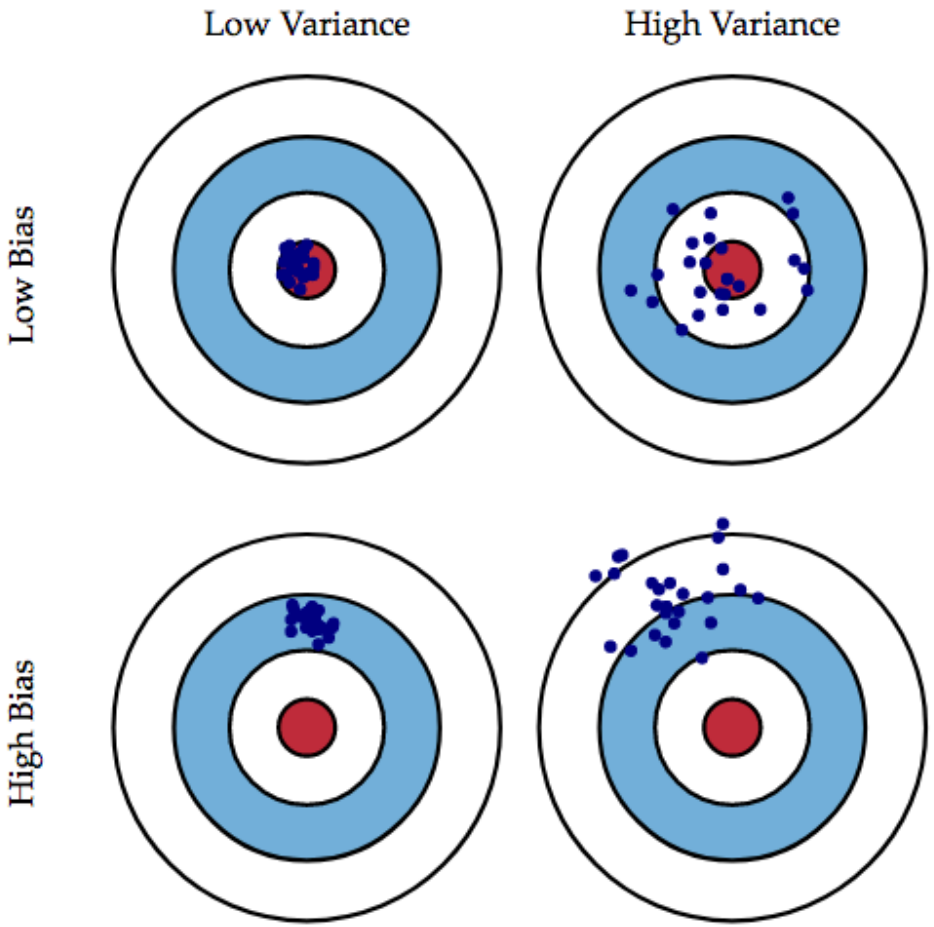


Fig. 1 Graphical illustration of bias and variance.

Example

Suppose you have 1000 instances. You divide these in 500+250+250 instances.

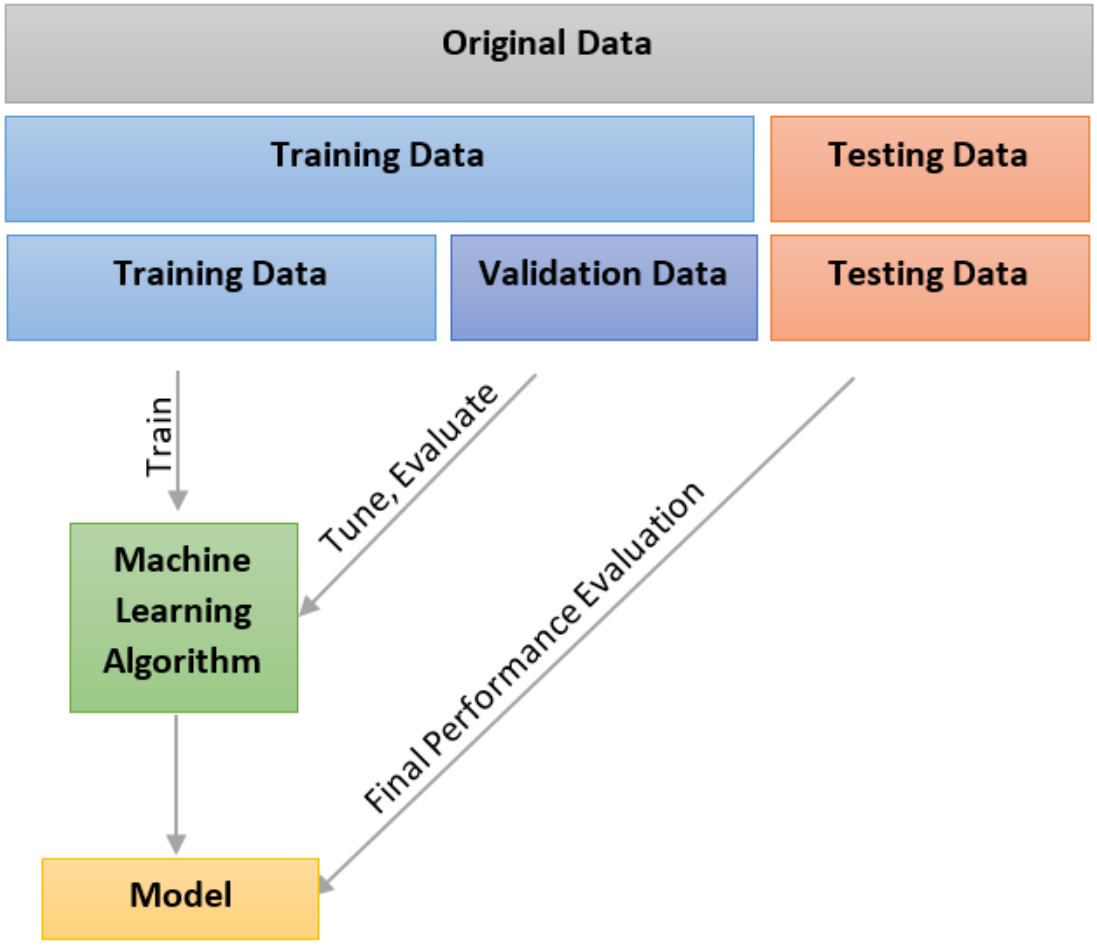
- Training set: 500 instances
- Validation set: 250 instances

With these sets you tune/optimize your model

Test set: 250 instances

The performance of the test set is an estimate of the prediction performance on unseen instances (generalization)

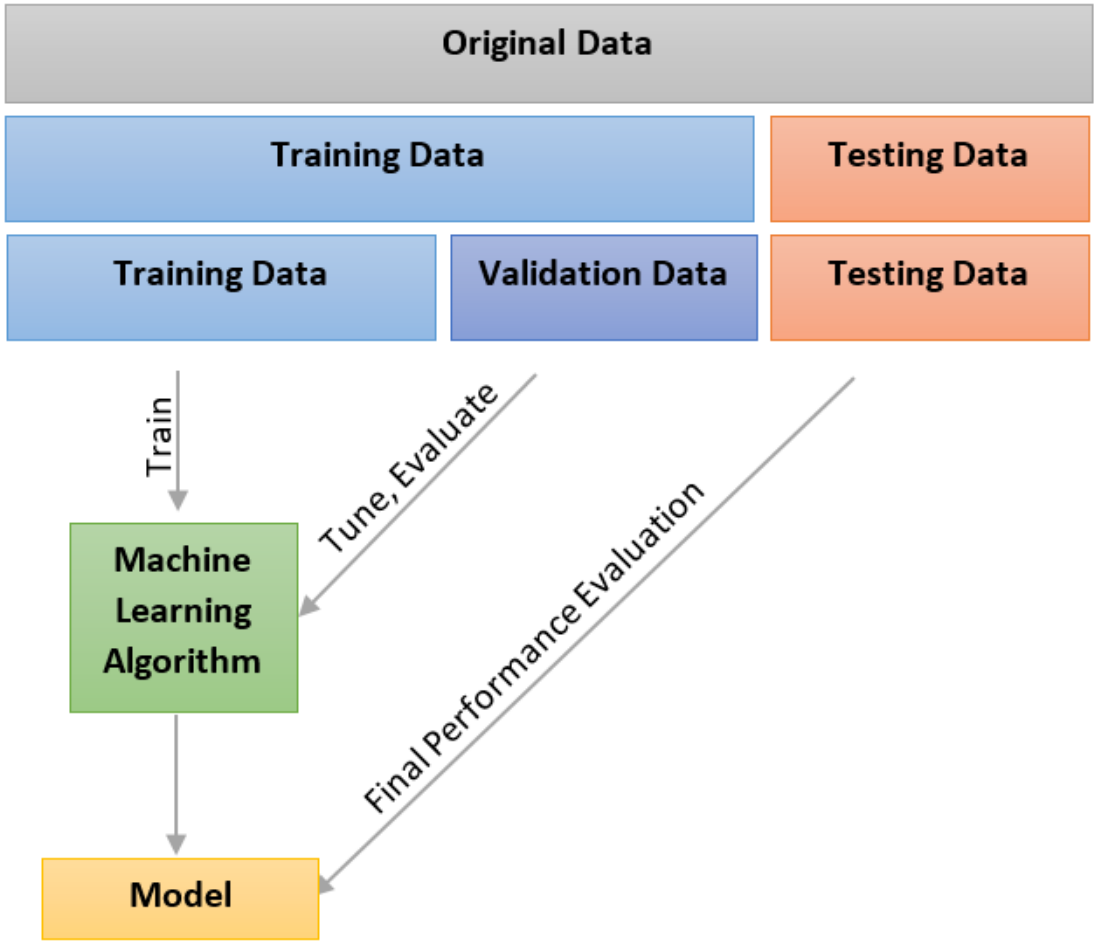
”The Data Mining (ML) Procedure”



Overview: fitting & tuning

- “The Data Mining Procedure”
- **Hyper Parameters**
- Model Evaluation
- Confusion Matrix
- ROC curves
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

Machine Learning Procedure



Hyperparameter

Hyperparameter

variable part of the model which isn't set during learning on training data

Needs to be tuned on a validation set

Model Parameter

- Required by the model when making predictions
- Values define the 'skill' of the model on your problem
- Are learned from data
- Often not set manually by the practitioner
- Often saved as part of the learned model

Example, the coefficient (betas) in a linear or logistic regression

Model Hyperparameter

- Often used in processes to help estimate model parameters.
- Often specified by the practitioner.
- Can be set using heuristics.
- Often tuned for a given predictive modeling problem.

Example, the K in K-nearest neighbors

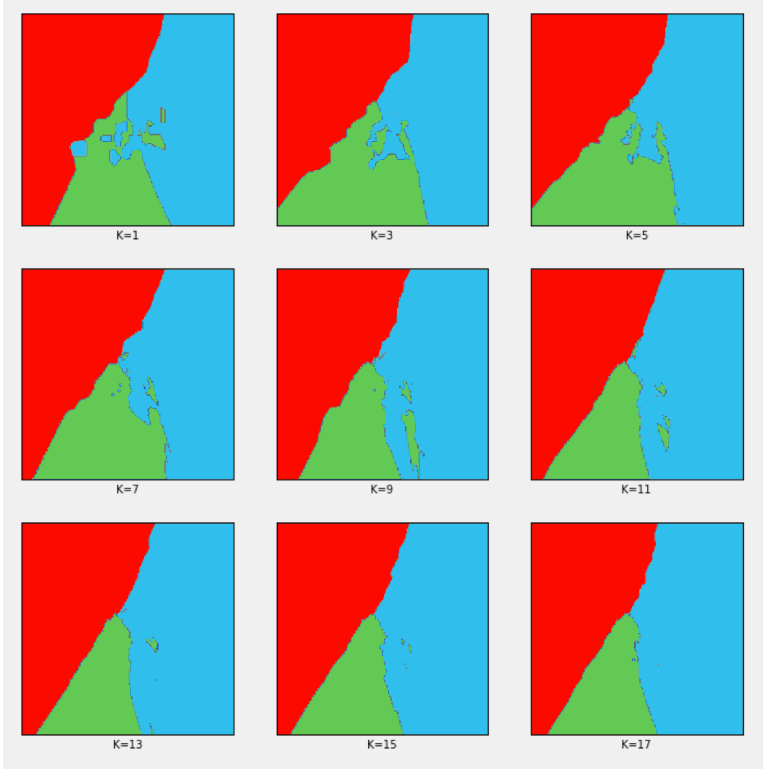
If you have to specify a model parameter manually then it is probably a model hyperparameter.

Example: Tuning K-NN

How can we control fit in K-NN?

by changing K

- Smaller K – more fit
- Larger K – less fit

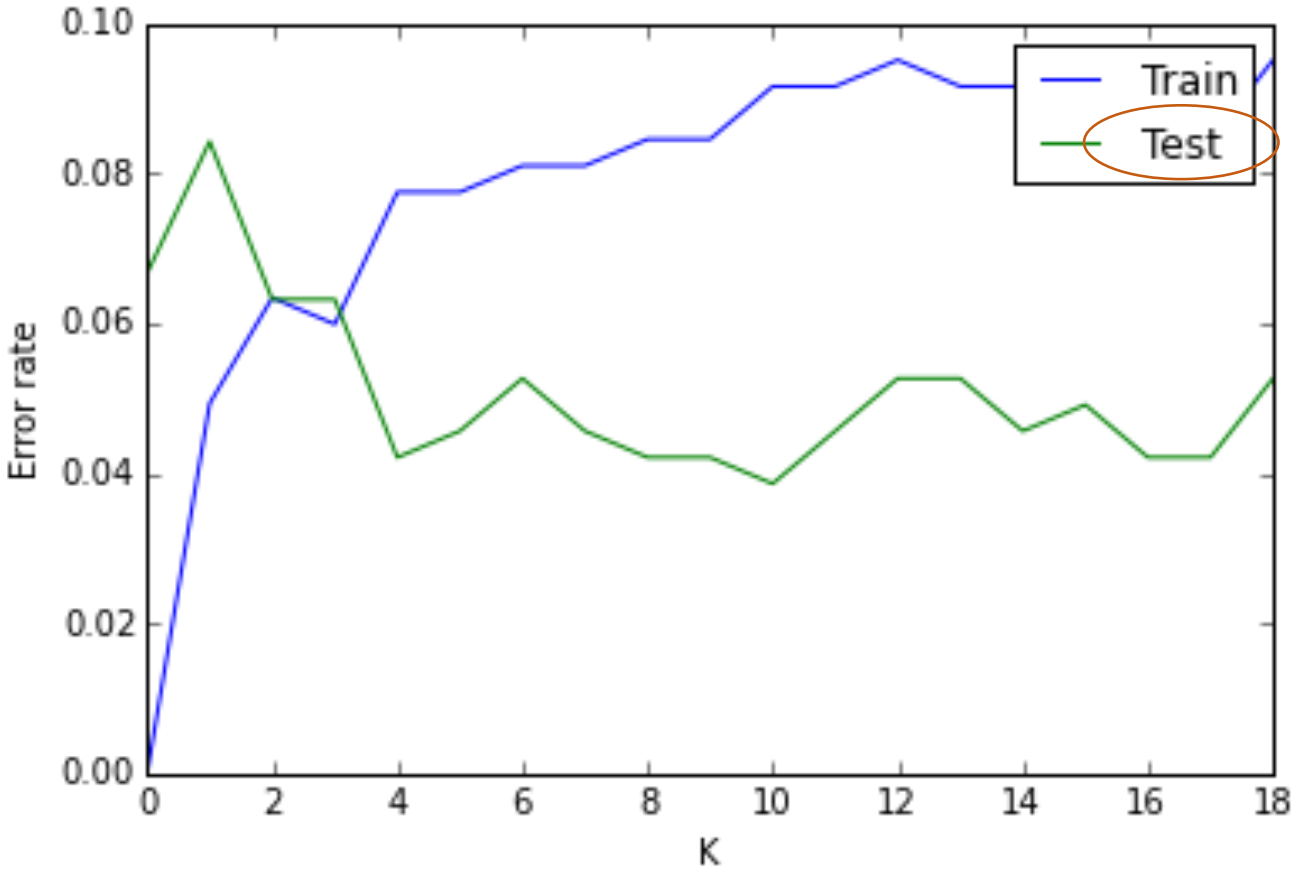


Breast cancer diagnostic dataset

Diagnosis (M = malignant, B = benign)

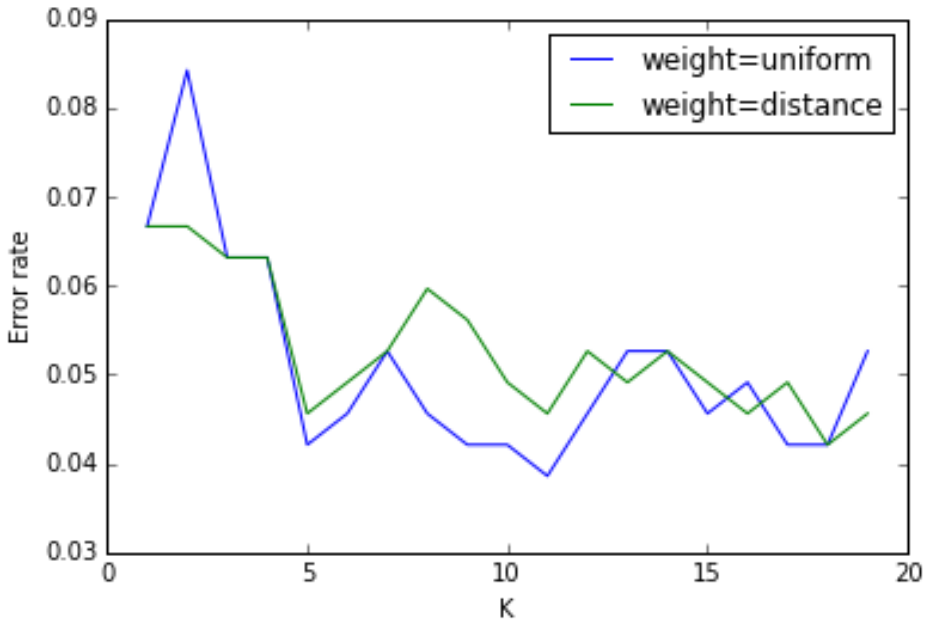
- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

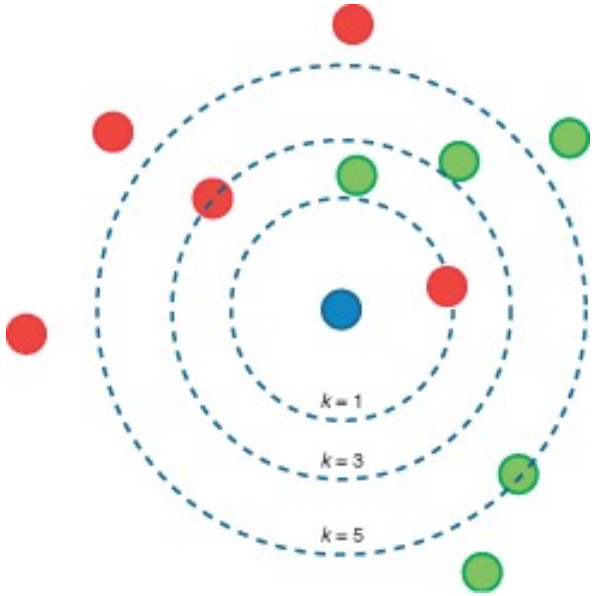
Train and test error as a function of K



Tuning multiple hyperparameters

$$\text{weight} \in \{\text{uniform}, \text{distance}\}$$
$$K \in \{1, \ldots, 19\}$$





K-NN hyperparams

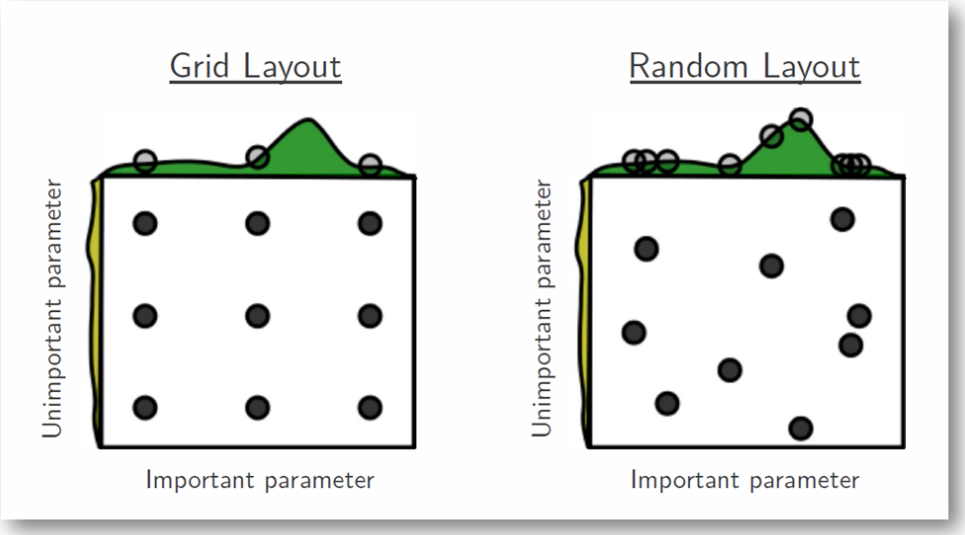
- K
- Neighbor weights
- Distance metric

Grid search: hyperparameter optimization

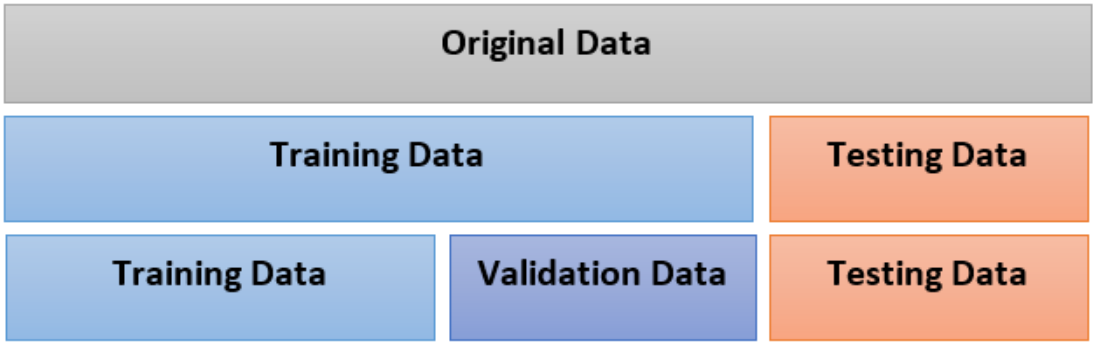
Systematic search for best hyperparameter settings

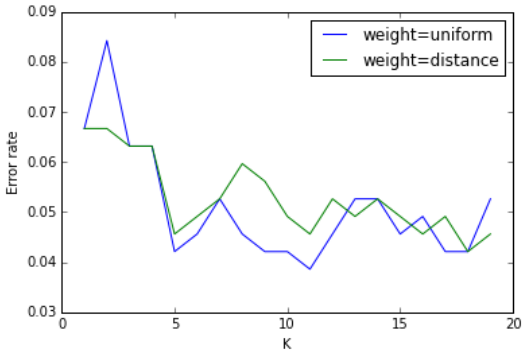
- Manually choose values to test for each hyperparameter
- Check validation accuracy/error for all combinations
- Number of parameters expand, but often behave parallel (relatively independent)

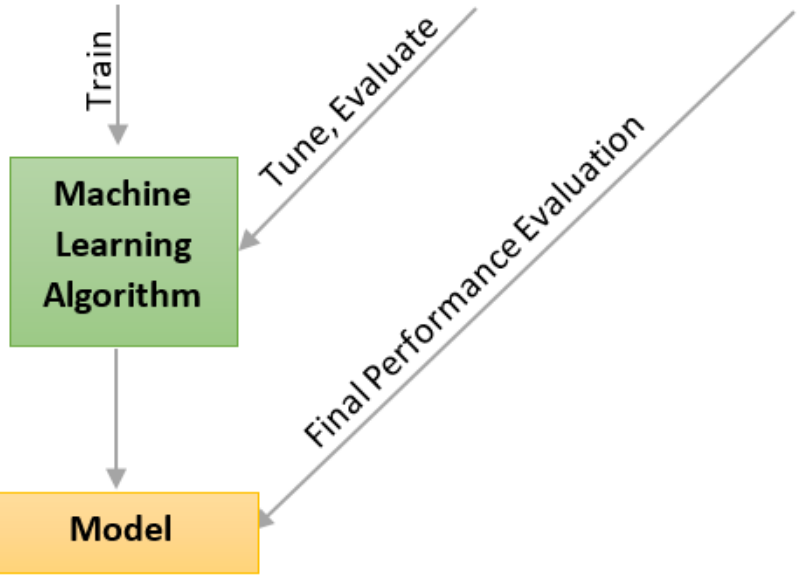
Lots of computation!

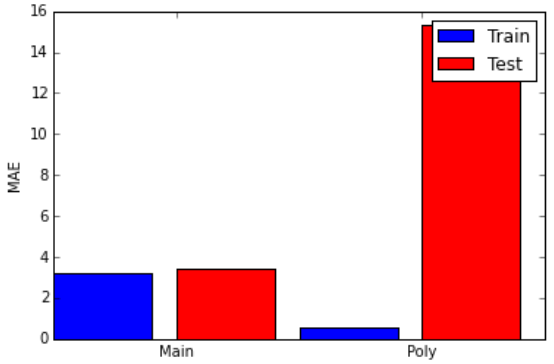


Machine Learning Procedure









Overview: fitting & tuning

- “The Data Mining Procedure”
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- ROC curves
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

Evaluating model performance

Metrics for a Regression Task:

- Coefficient of determination (R^2)
- Mean Absolute Error
- Root Mean Squared Error

Metrics for Classification Task:

R^2: coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y^{(i)} - f(x^{(i)}))^2}{\sum_{i=1}^{N}(y^{(i)} - \text{mean}(x))^2}$$

How well model f predicts targets relative to mean

Equivalent to proportion of variance explained by f

The mean is not always a the suitable baseline

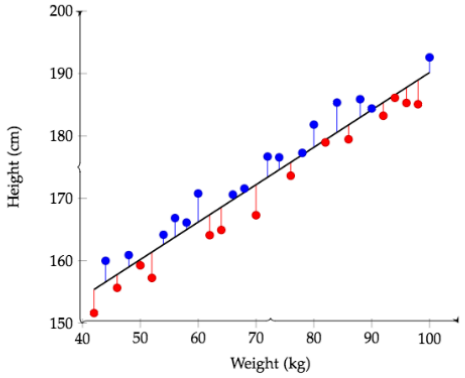
Measures of the Error

Error: difference between true value y and predicted value

Err = f(x) - $\hat{y}$

Errors may be positive or negative

- Summing errors can be misleading
- Square/Absolute values prior to summing



Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \qquad \mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

RMSE vs MAE

CASE 1: Evenly distributed errors

ID	Error	|Error|	Error^2
1	2	2	4
2	2	2	4
3	2	2	4
4	2	2	4
5	2	2	4
6	2	2	4
7	2	2	4
8	2	2	4

MAE	RMSE
2.000	2.236

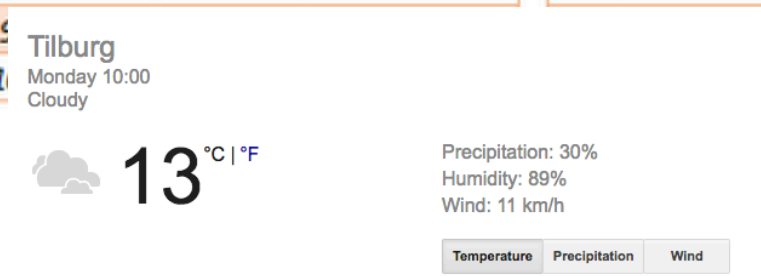
CASE 2: Small variance in errors

ID	Error	|Error|	Error^2
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	9
7	3	3	9
8	3	3	9
9	3	3	9
10	3	3	9

CASE 3: Large error outlier

ID	Error	|Error|	Error^2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	20	20	400

MAE	RMSE
2.000	6.325



Conclusion: MAE is easier to interpret, so present results, but the RMSE has some nice properties (often use RMSE for training your model).

Evaluation performance

Metrics for a Regression Task:

- Coefficient of determination (R^2)
- Mean Absolute Error
- Root Mean Squared Error

Typically report R^2 and Error (both)

Metrics for Classification Task:

- Confusion Matrix
- Accuracy, Precision, Recall, F1 score
- Area Under ROC Curve

What are we predicting?

Is there a dominant class?

- Does it matter?

Example:

- Predict if a TiU student will win European Cup Soccer?
- Claim: have a classifier that will at least be 99% accurate
- Always predict “no” → correct or almost all students

Accuracy and unbalanced classes

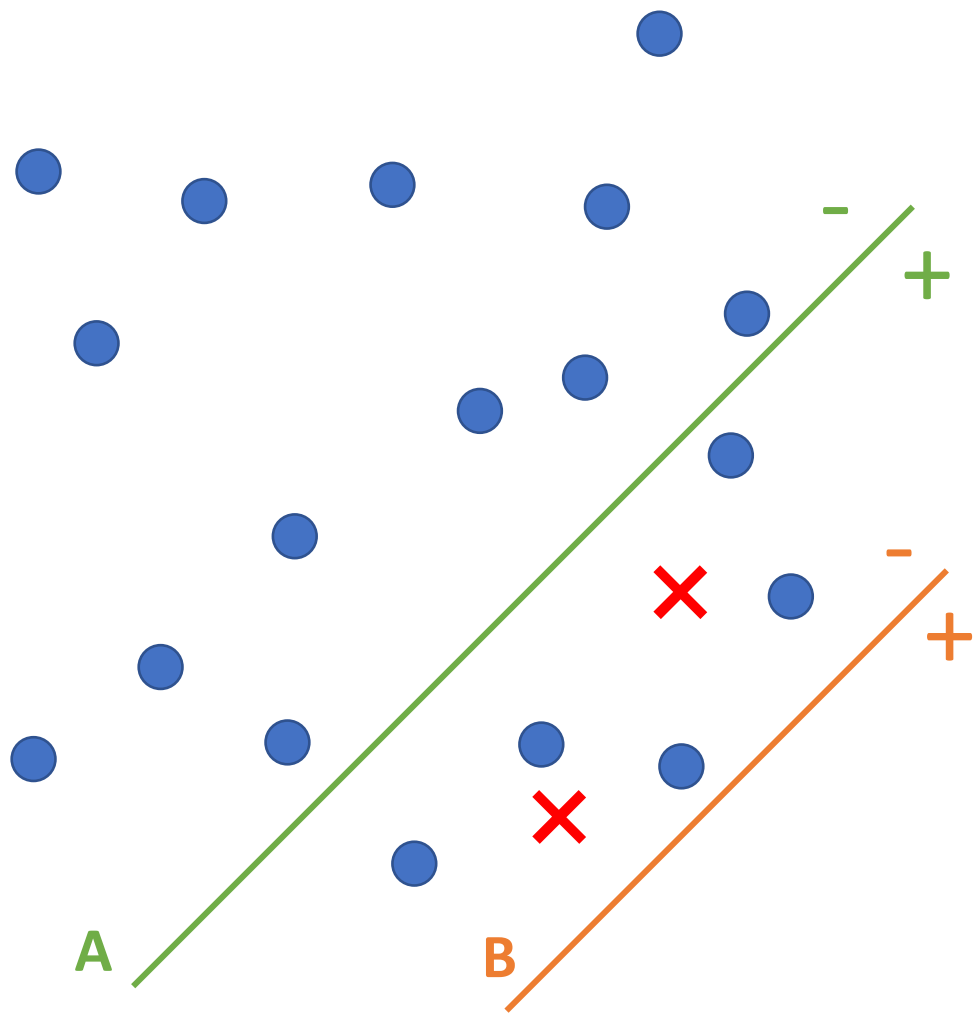
Predict Euro Cup (✕)or Not (●)

Would Your Prefer classifier
A or B?

Is accuracy (% correct)
higher for A or B?

Accuracy and error rate
are not enough

Can also be problem when you
split your data



Evaluation performance

Metrics for a Regression Task:

- Coefficient of determination (R^2)
- Mean Absolute Error
- Root Mean Squared Error

Typically report R^2 and Error (both)

Metrics for Classification Task:

- Confusion Matrix
- Accuracy, Precision, Recall, F1 score
- Area Under ROC Curve

Overview: fitting & tuning

- "The Data Mining Procedure"
- Hyper Parameters
- Model Evaluation
- **Confusion Matrix**
- ROC curves
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

Overview: fitting & tuning

- “The Data Mining Procedure”
- Hyper Parameters
- **Model Evaluation**
- Confusion Matrix
- ROC curves
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

Confusion Matrix

Assignment to a class: $y \in \{0,1\}$

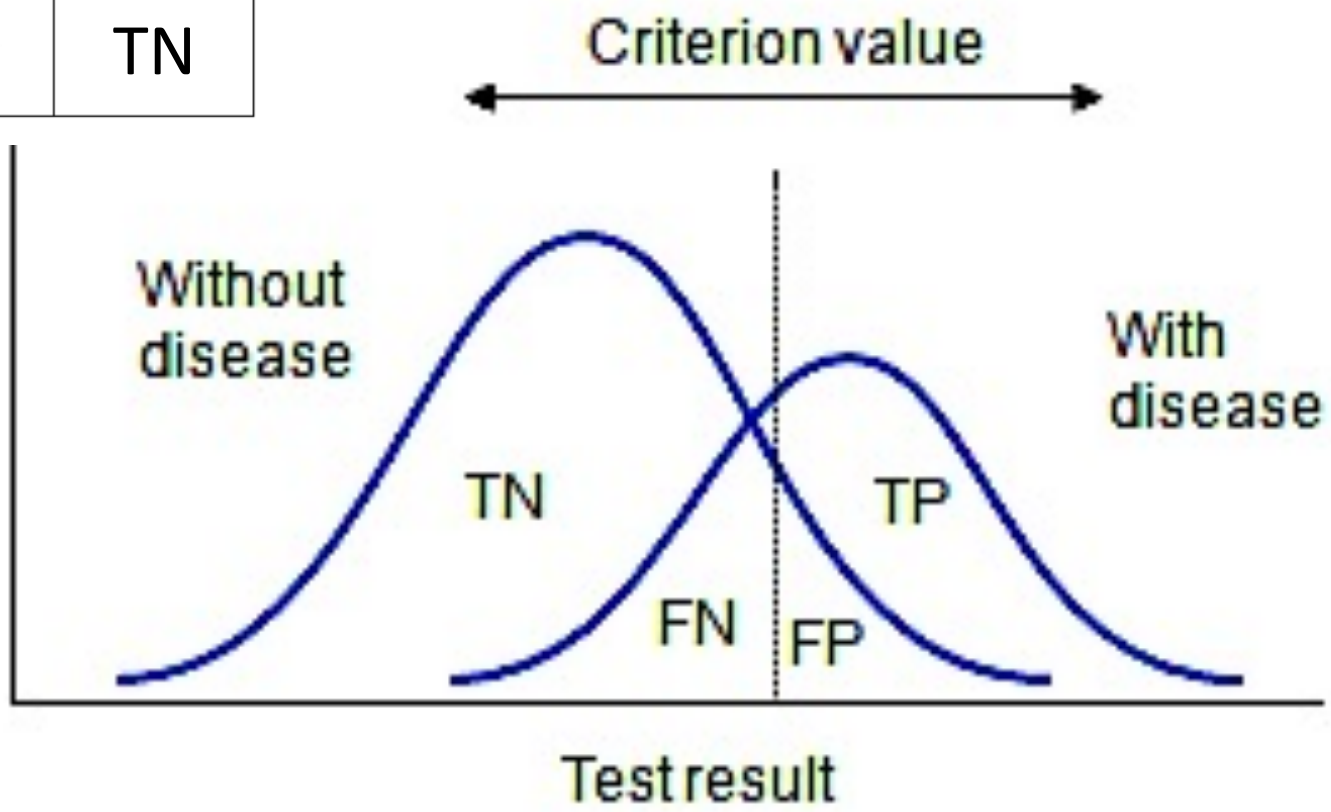
0 = "negative class" (N) e.g legit, benign "dominant class"

1 = "positive class" (P) e.g. fraudulent, malignant "rare class"

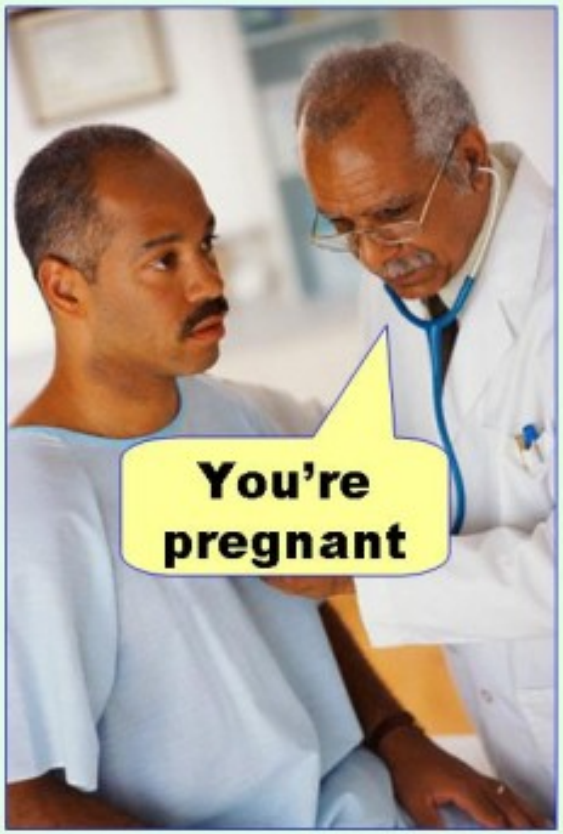
	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Confusion Matrix

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN



Type I error
(false positive)



Type II error
(false negative)

Accuracy

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Positives + Negatives}$$

“the curious case of unbalanced classes”

	Predicted Negative	Predicted Positive
Negative Cases	9,700	150
Positive Cases	50	100

$$\mathrm{A}(M) = \frac{9,700 + 100}{9,700 + 150 + 50 + 100} = 98.0\%$$

	Predicted Negative	Predicted Positive
Negative Cases	9,850	0
Positive Cases	150	0

$$\mathrm{A}(M) = \frac{9,850 + 0}{9,850 + 150 + 0 + 0} = 98.5\%$$

Precision: hit-rate

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

Of all transaction where we predicted fraud (y = 1), what fraction actually was fraud

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall: true positive rate

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

Of all fraudulent transaction, what fraction actually did we correctly detect as fraud

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Machine Learning vs. Medicine

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP} = \frac{TN}{healthy}$$

$$Sensitivity = \frac{TP}{TP+FN} = \frac{TP}{diseased}$$

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

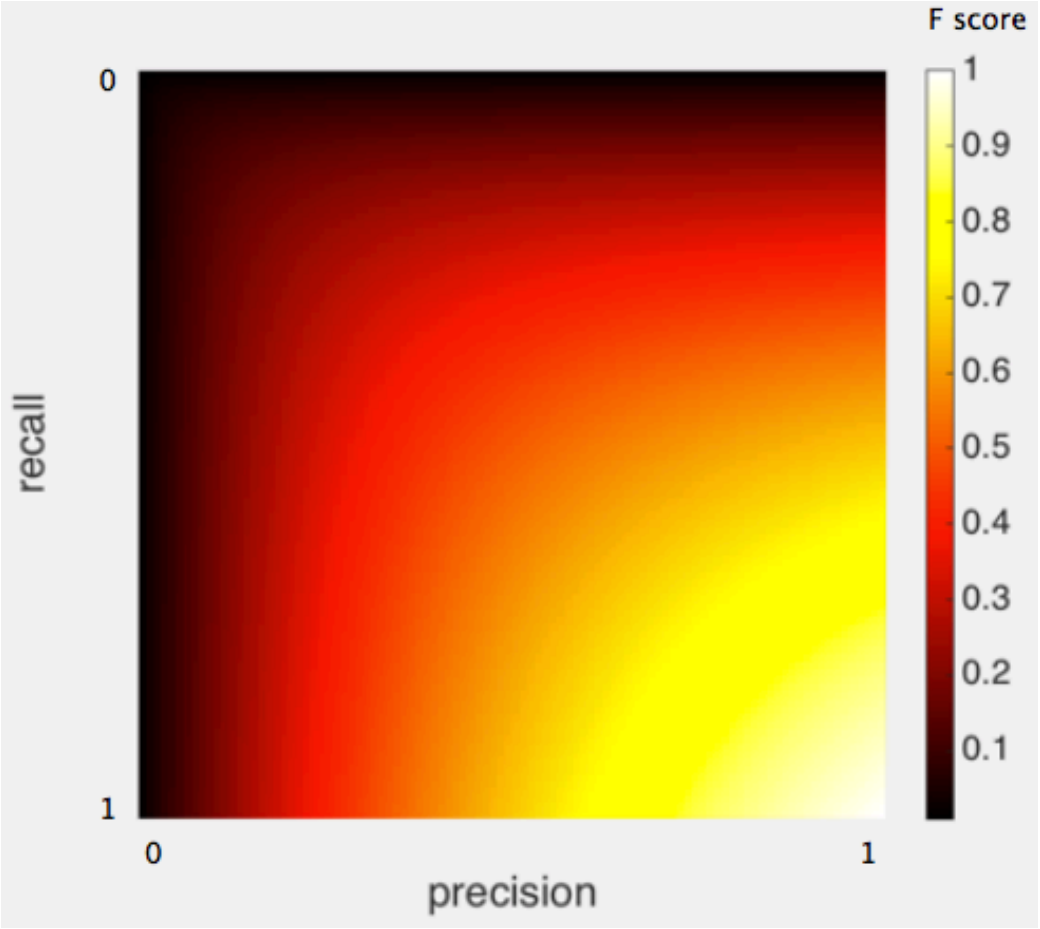
F or F1-score

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$F1 = 2\,\frac{Precision * Recall}{Precision + Recall}$$

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

F1 Score: recall vs. precision



Example 1

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

	P'	N'
P	485	515
N	9	991

- Accuracy: 0.738
- Precision: 0.982
- Recall: 0.485
- F score: 0.649

Example 2: unbalanced classes

	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

	P'	N'
P	493	507
N	1017	98983

Medical Screening

- Accuracy: 0.985
- Precision: 0.327
- Recall: 0.493
- F score: 0.393

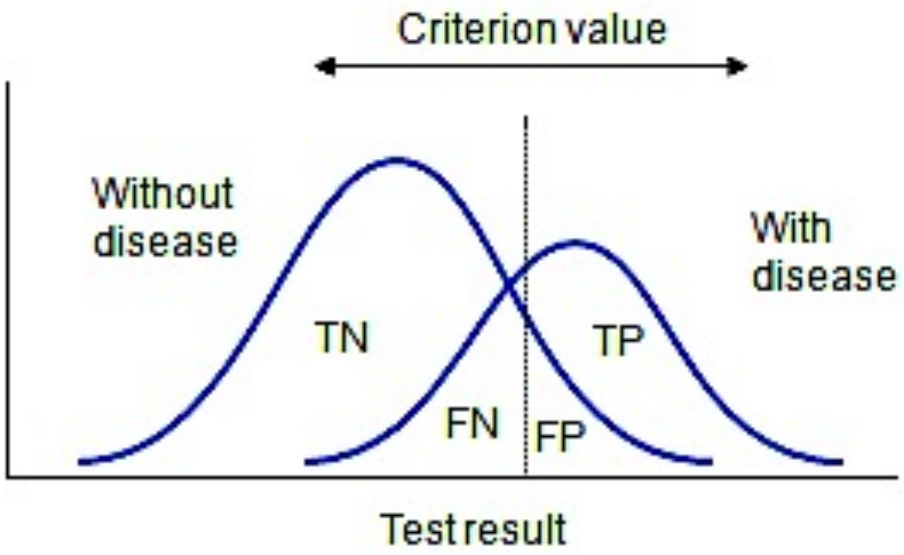
Overview: fitting & tuning

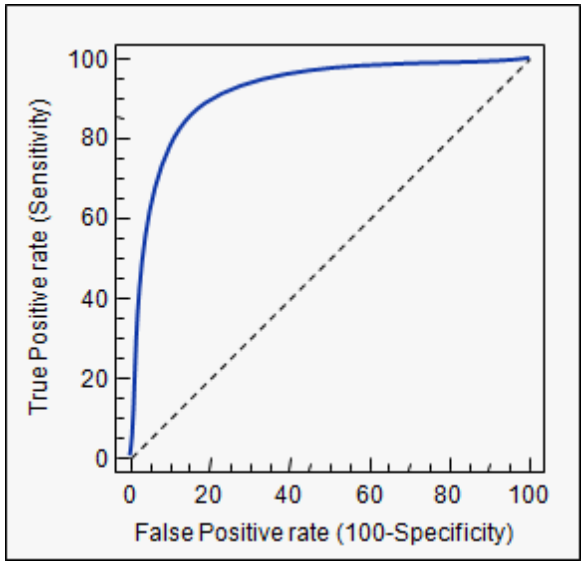
- "The Data Mining Procedure"
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- **ROC curves**
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

ROC curves (area under the curve)

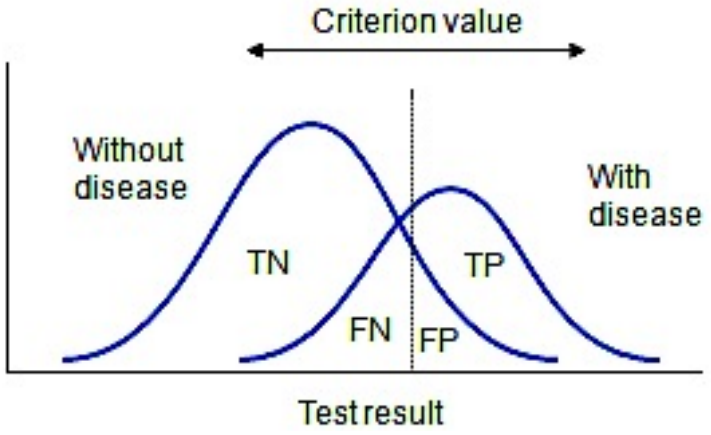
- Widely used measure for classification
- Measure between 0.5 and 1.0
- Can inspect outcome for different thresholds (criterion)

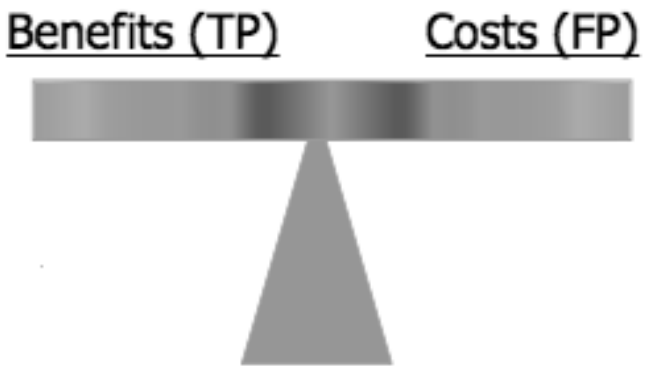
	p' (Predicted)	n' (Predicted)
p (Actual)	TP	FN
n (Actual)	FP	TN

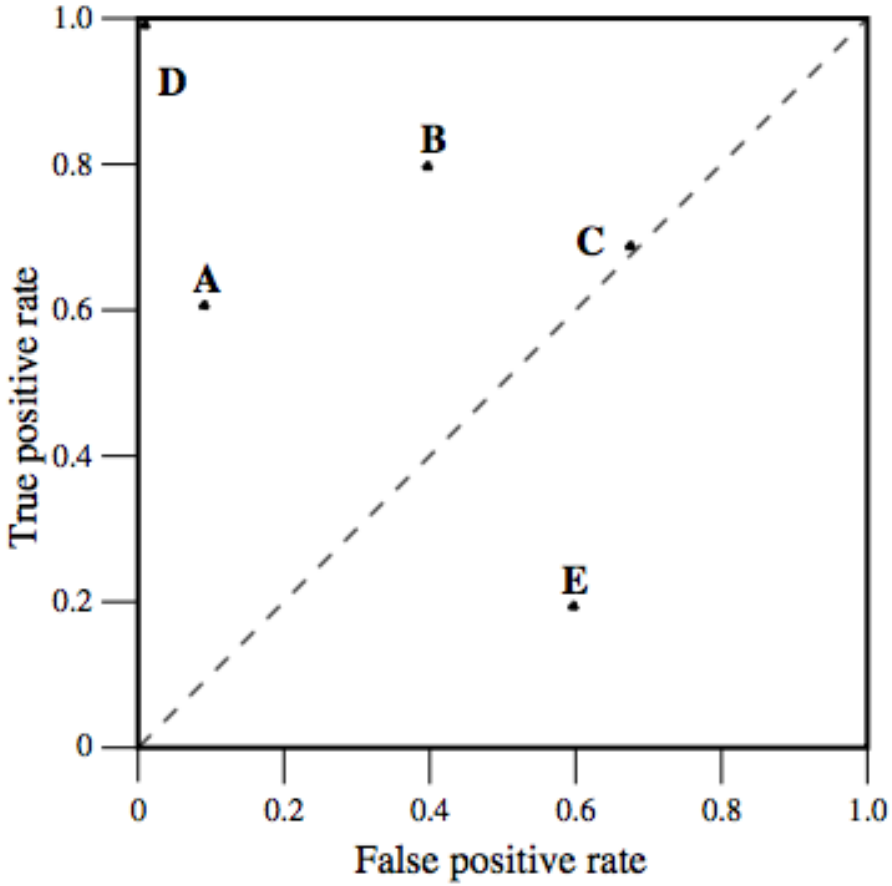




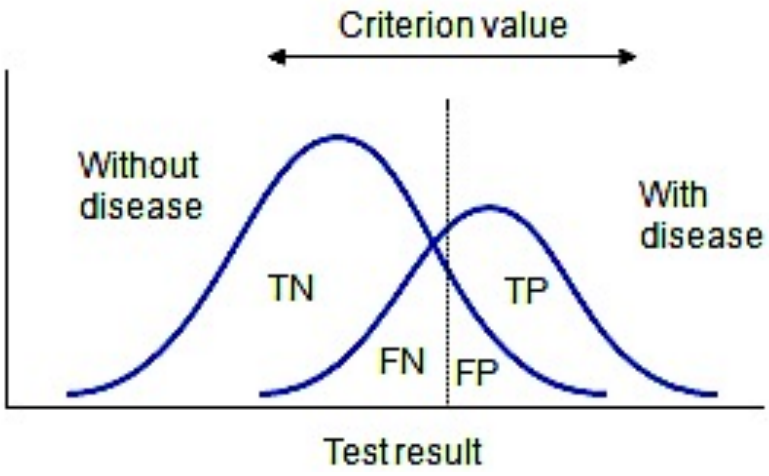
ROC curves

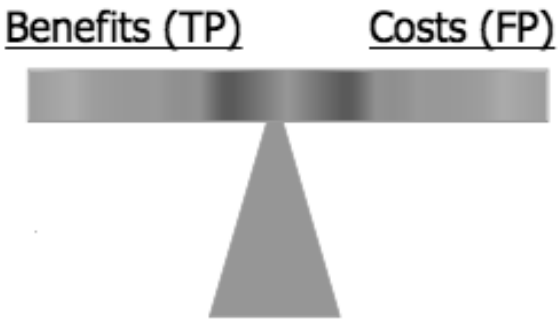

Criterion value
Without disease
With disease
TN
TP
FN
FP
Test result

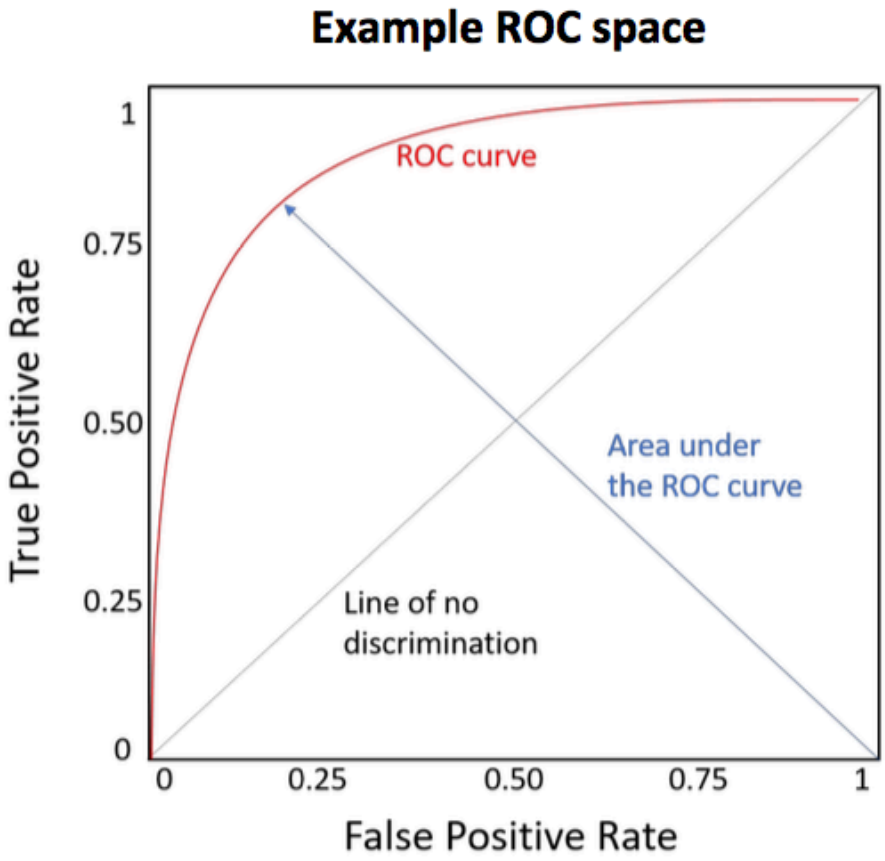

Benefits (TP)
Costs (FP)


1.0
0.8
0.6
0.4
0.2
0
D
B
C
A
E
True positive rate
0
0.2
0.4
0.6
0.8
1.0
False positive rate

ROC curves

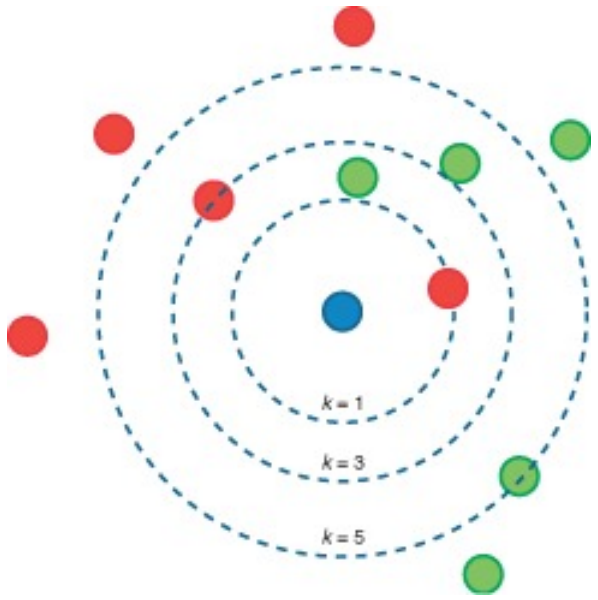

Criterion value
Without disease
With disease
TN
TP
FN
FP
Test result


Benefits (TP)
Costs (FP)


Example ROC space
True Positive Rate
False Positive Rate
ROC curve
Area under the ROC curve
Line of no discrimination
0
0.25
0.50
0.75
1

Example of test with 20 instances

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



Example of test with 20 instances

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

Threshold Score $\geqq$ 0.54

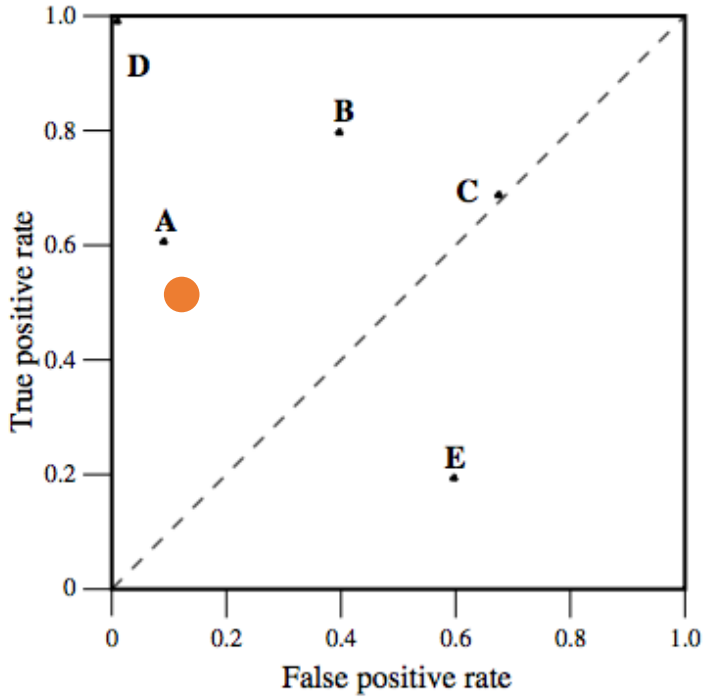
	p' (Predicted)	n' (Predicted)
p (Actual)	5	1
n (Actual)	5	9

True Positive Rate = 5/10 = 0.50

False Positive Rate = 1/10 = 0.10

Example of test with 20 instances

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

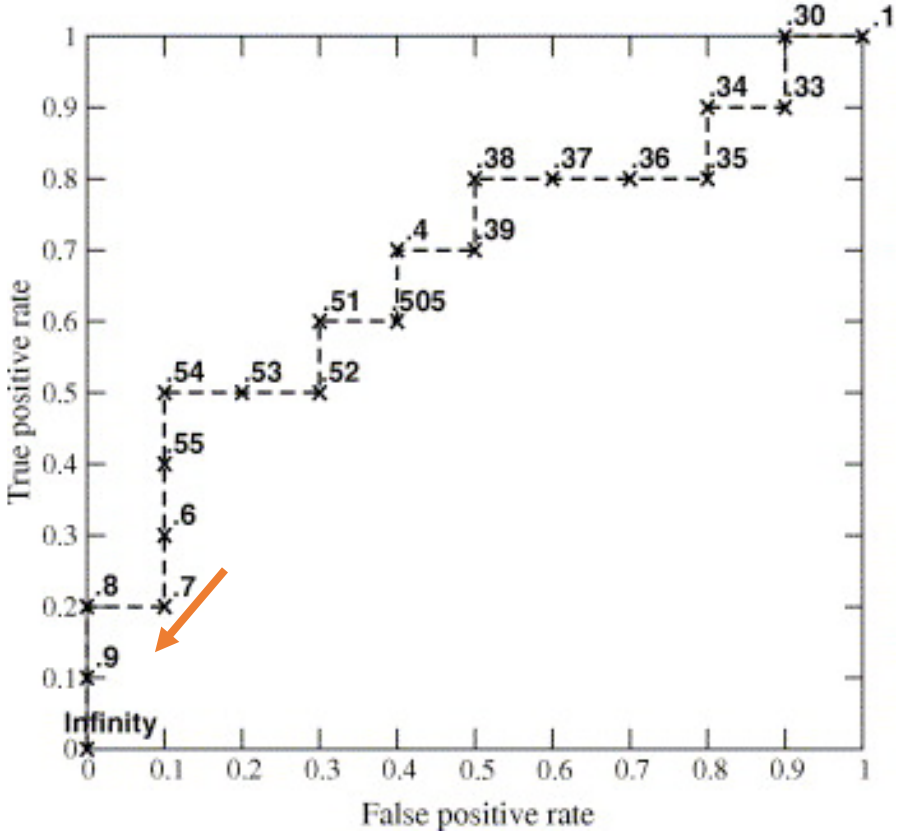


True Positive Rate = 5/10 = 0.50

False Positive Rate = 1/10 = 0.10

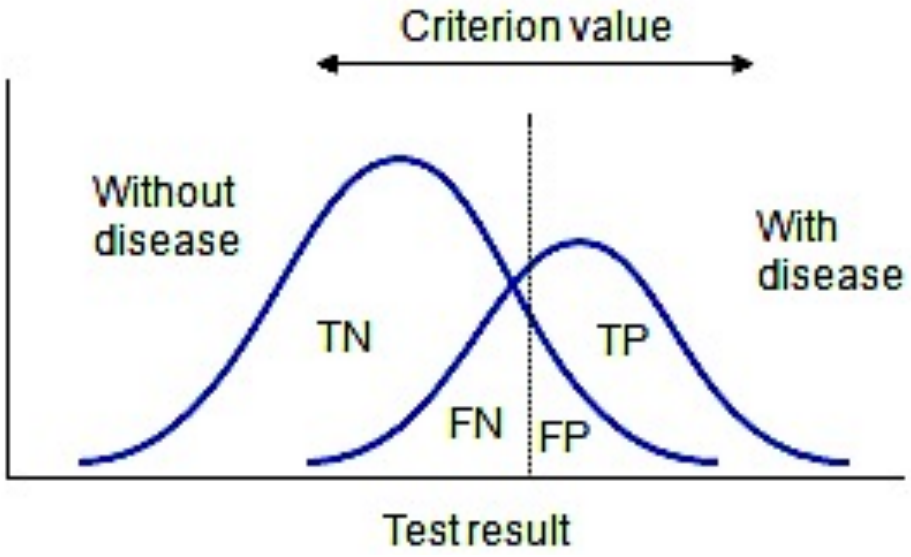
Example of test with 20 instances

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

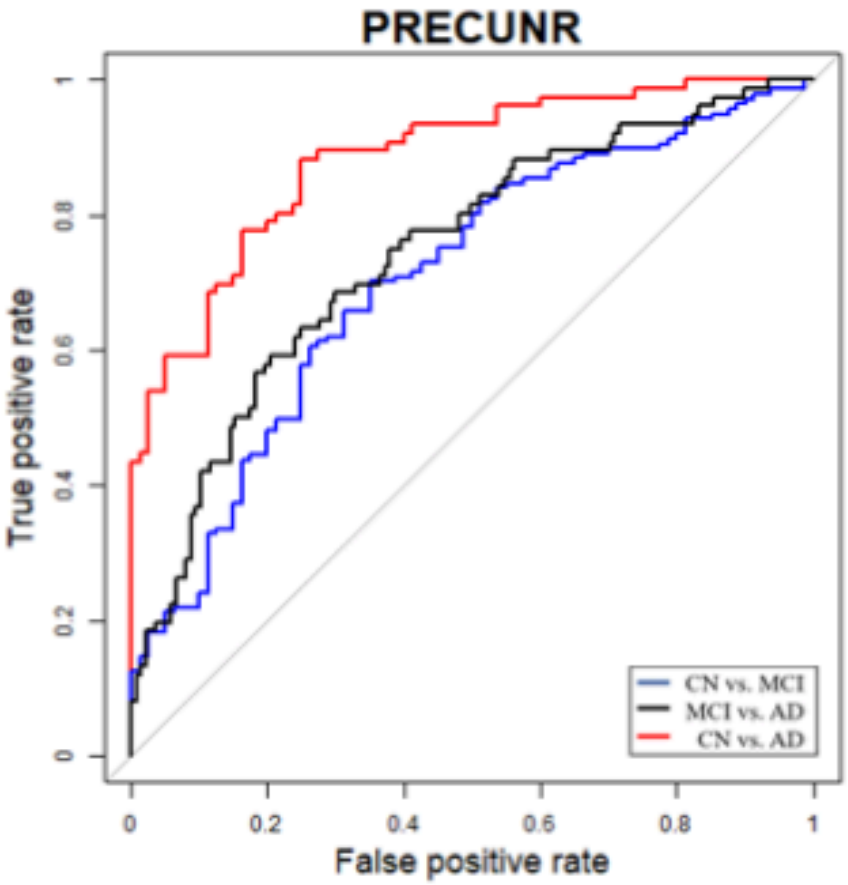


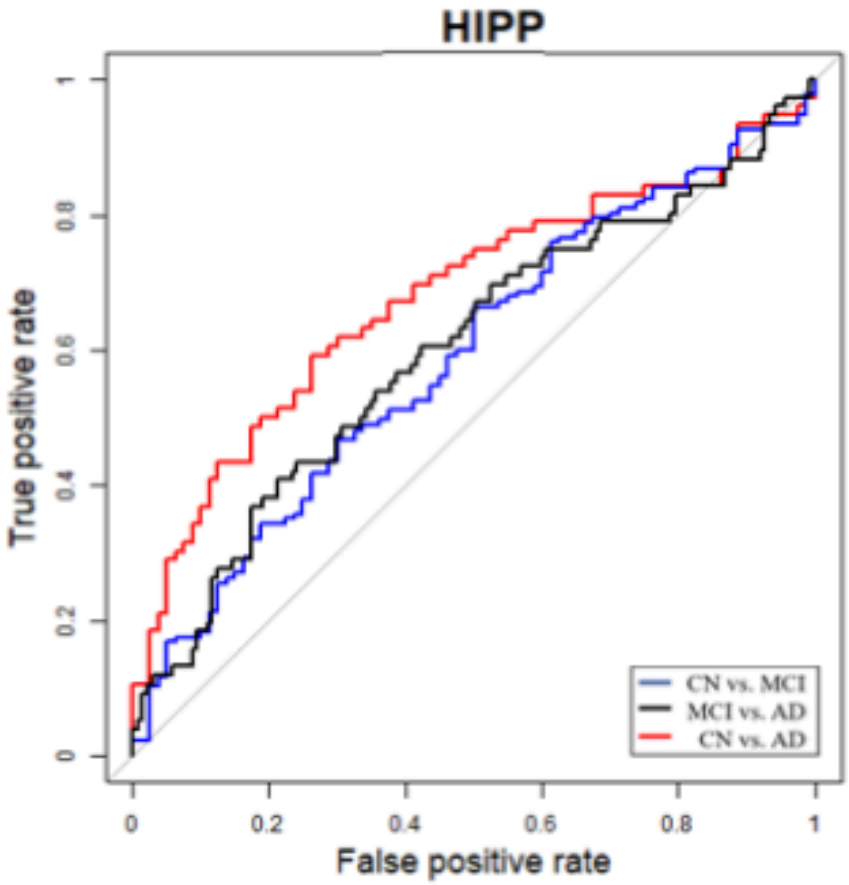
Example of test with 20 instances

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



Example: ROC curves

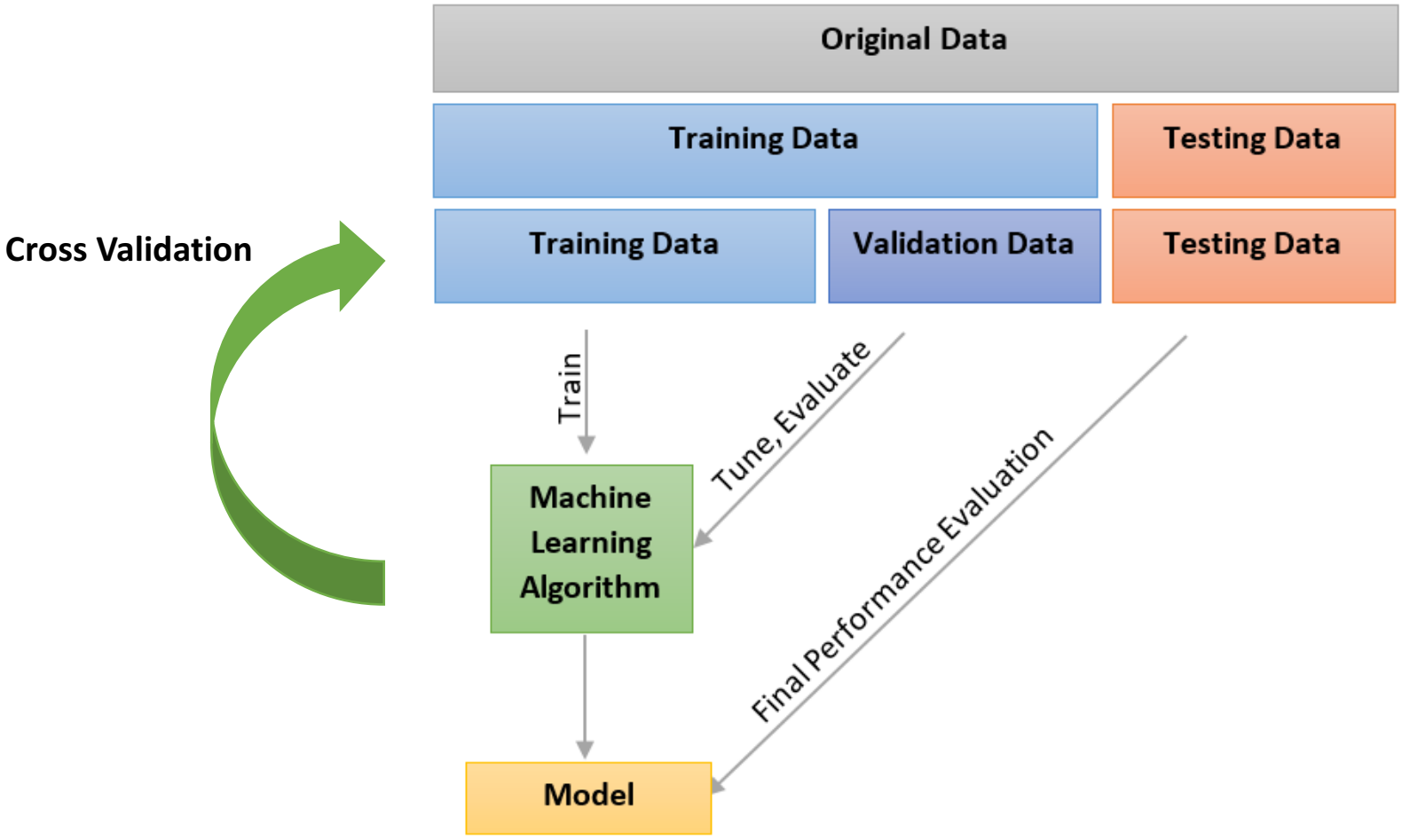




Overview: fitting & tuning

- "The Data Mining Procedure"
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- ROC curves
- **Cross Validation**
- The curse of dimensionality
- Under-fitting and Over Fitting

Machine Learning Procedure



Cross Validation

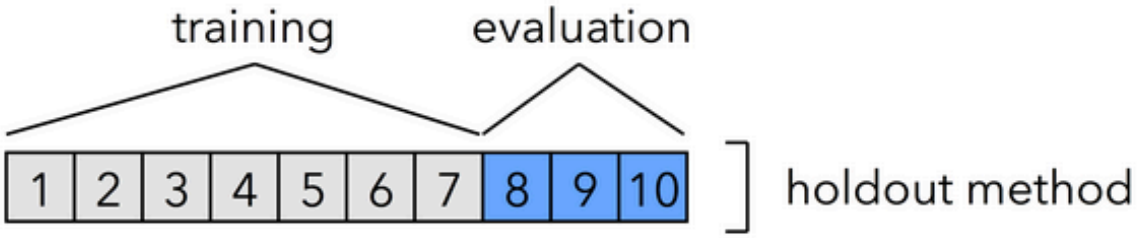
To estimate performance of the learned model from available data using one algorithm

- Often motivated by small datasets

To compare the performance of two or more different algorithms and find out the best algorithm for the available data

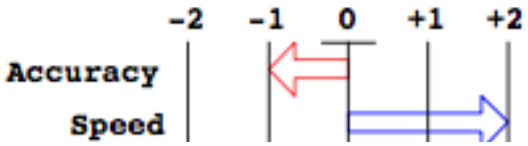
- Often motivated by search optimal solution

Hold out method

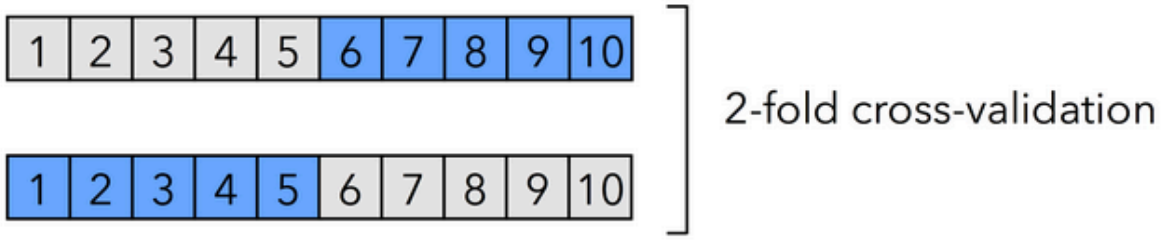


1. Separate the available examples into training set (examples used for training) and testing set (the rest of the data).

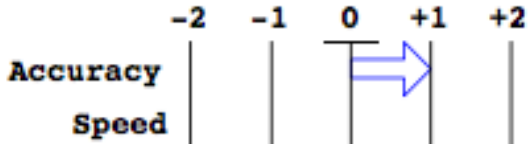
2. Calculate the accuracy over the testing



K-fold Cross Validation

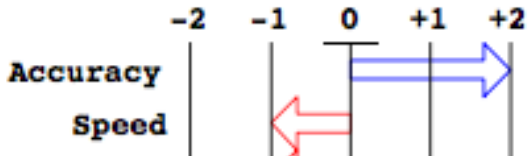


1. Partition randomly the available examples into k disjoint subsets.
2. Use one of the partitions as a testing set to evaluate a model generated by considering the other subsets as the training set.
3. Repeat this process with all subsets and average the obtained errors.



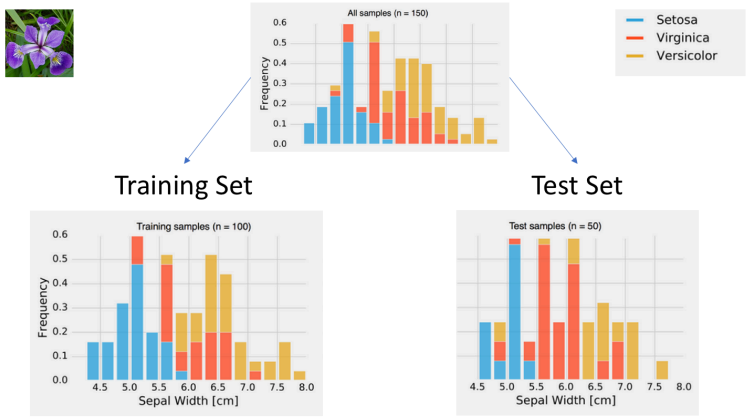
Leave one out

1. Use one of the examples to test a model generated by considering the other examples.
2. Calculate judgement for this model: 0 if model and example are consistent, 1, otherwise.
3. Repeat these two steps for all examples and average the obtained judgement values.

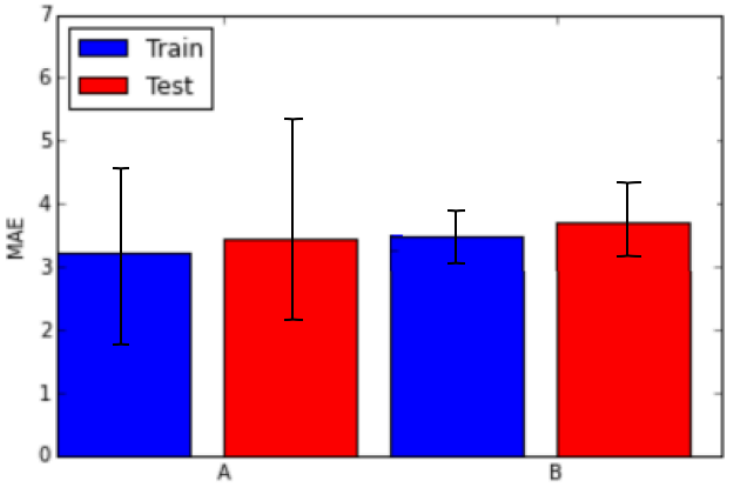


Cross Validation: advantages

To estimate performance of the learned model from available data using one algorithm (small dataset)



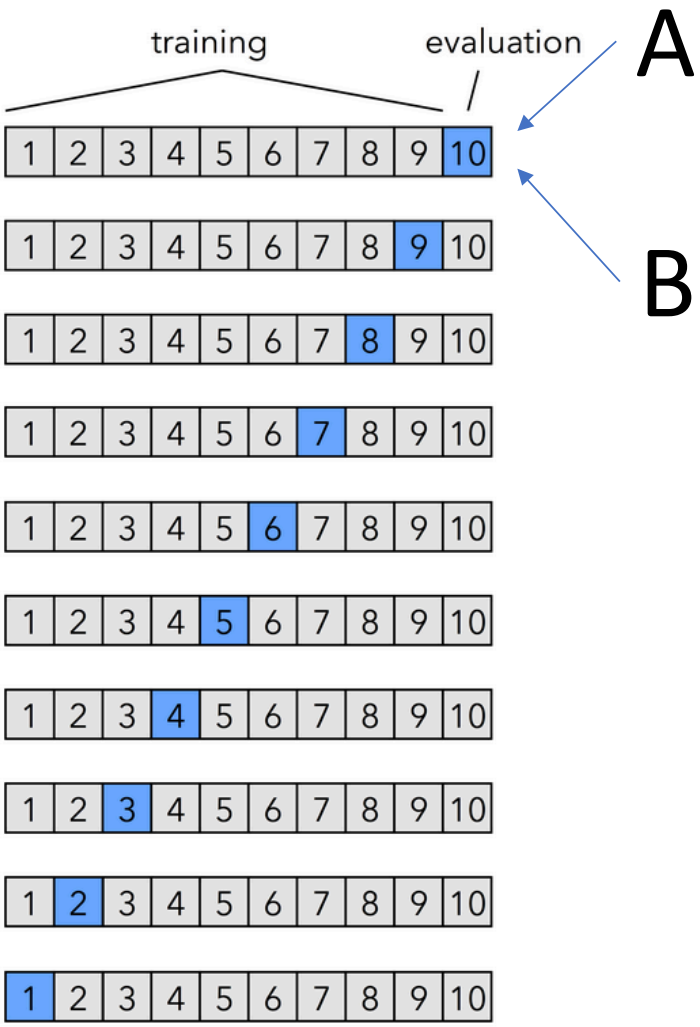
To compare the performance of two or more different algorithms



Model Selection

Pair-wise Comparison of algorithm A and B

~(allows pair-wise t-test)

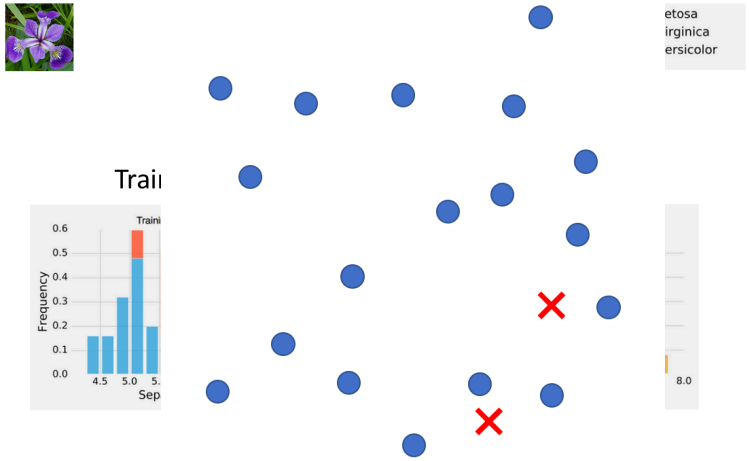


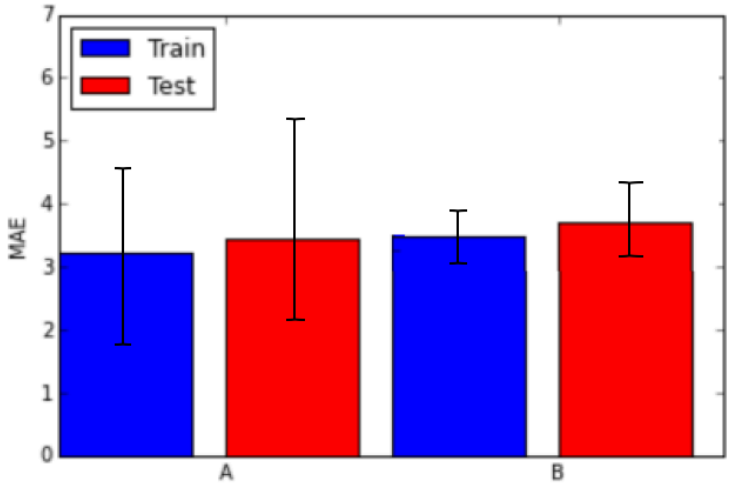
Cross Validation: advantages

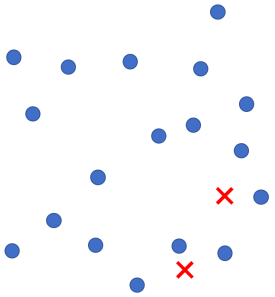
To estimate performance of the learned model from available data using one algorithm (small dataset)

"Unbalanced is a sort of small data"

To compare the performance of two or more different algorithms



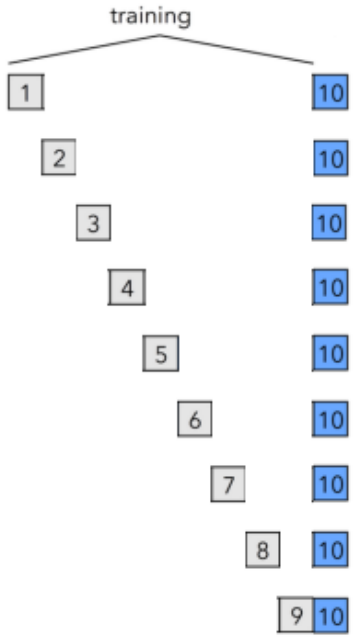


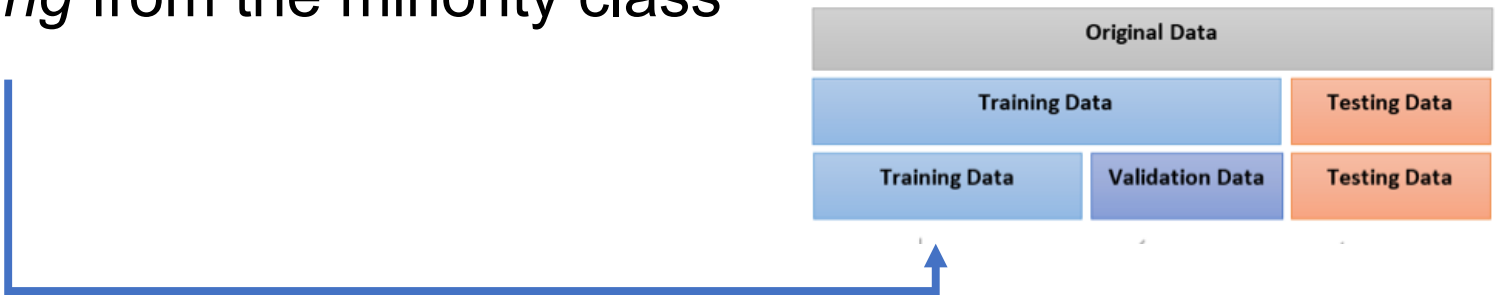


Resample Training data: deal with unbalanced classes

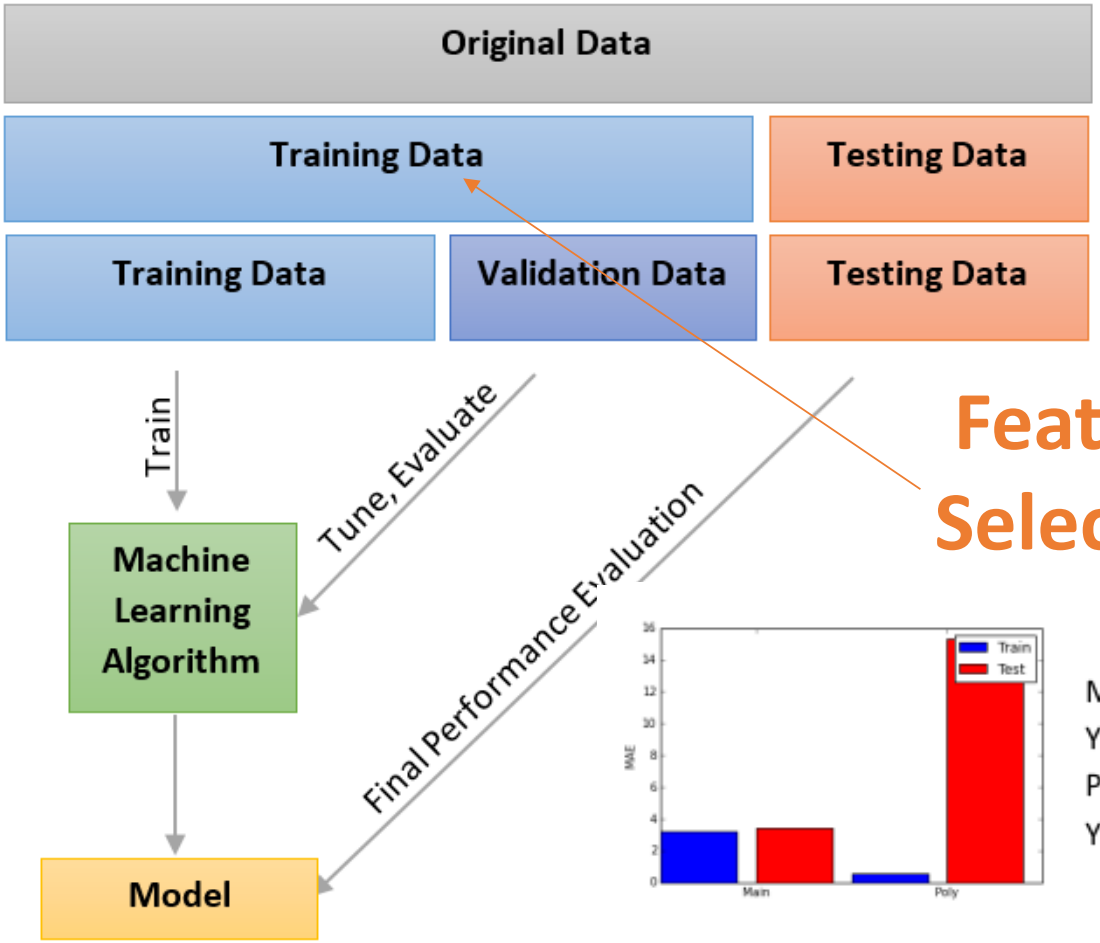
- **Downsampling** creates a balanced dataset by matching the number of samples in the minority class with a random sample from the majority class

- **Upsampling** matches the number of samples in the majority class with *resampling* from the minority class





Cross Validation



Main effects

Y = [a b c]

Poly: Main + Interactions

Y = [a b c ab ac abc]

Overview: fitting & tuning

- "The Data Mining Procedure"
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- ROC curves
- Cross Validation
- **The curse of dimensionality**
- Under-fitting and Over Fitting

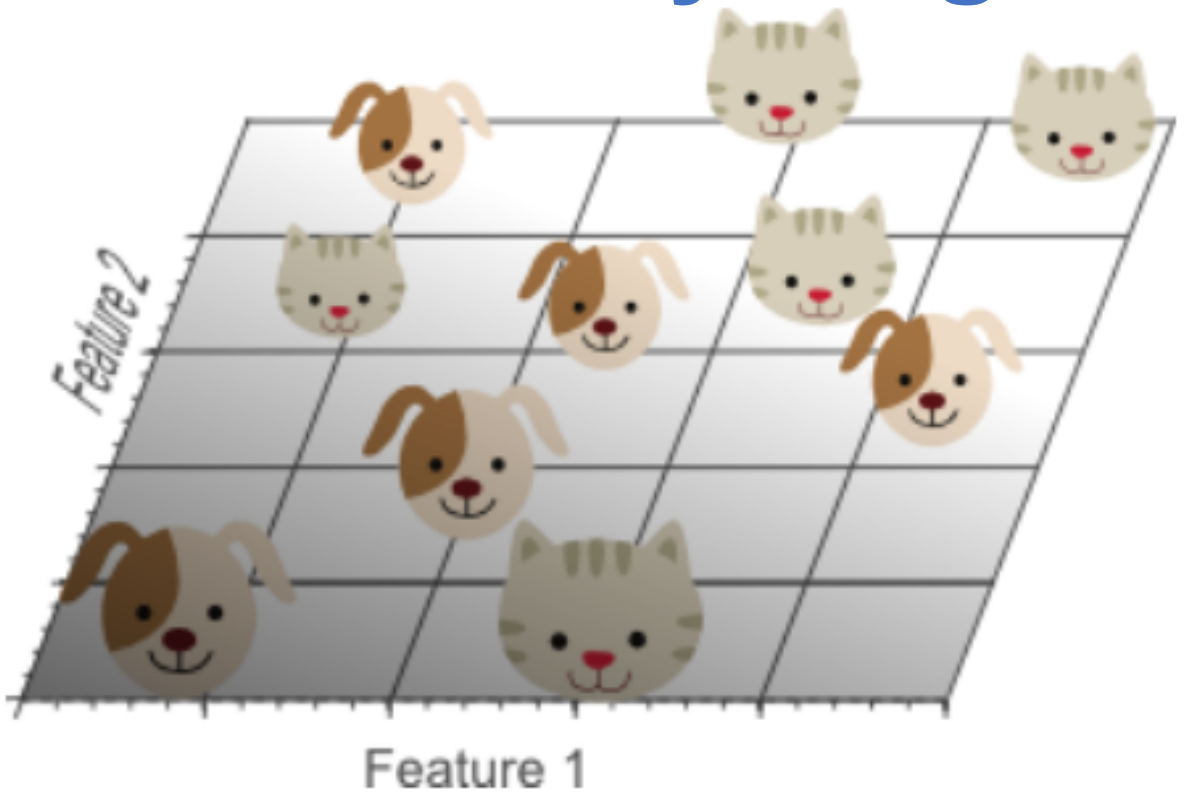
Feature Selection: curse of dimensionality

- We consider a simple linear decision boundary to solve the CAT-DOG classification task...

More features may be good (1)

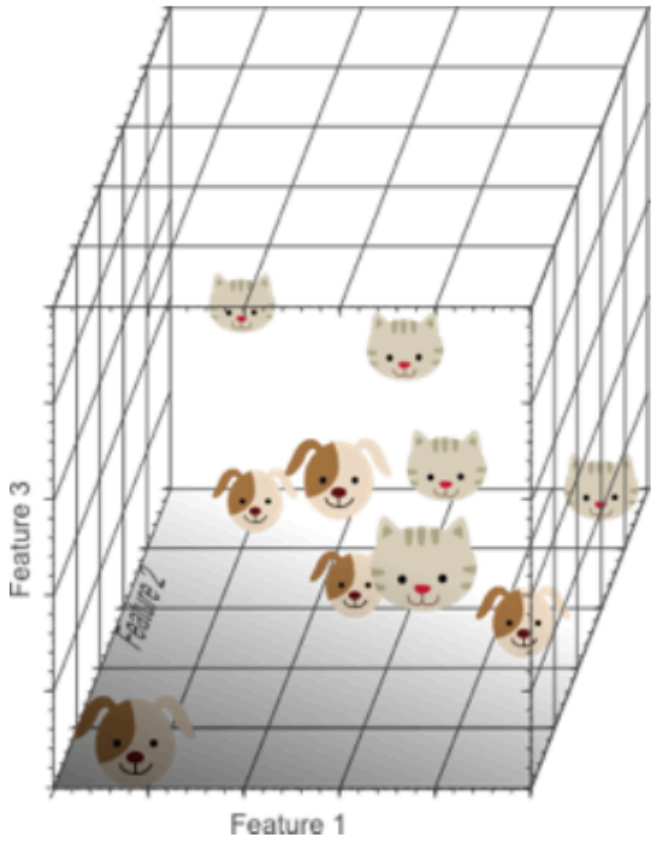
- 1 feature is insufficient to separate cats and dogs

More features may be good (2)



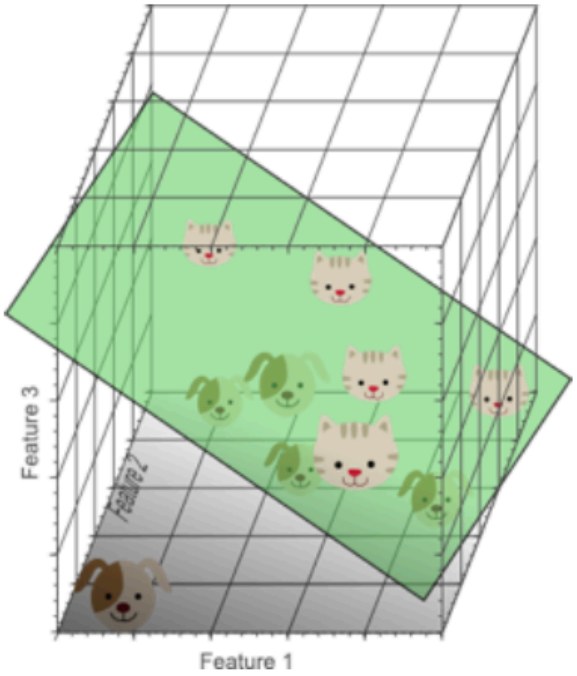
- 2 feature is insufficient to separate cats and dogs

More features may be good (3)



- 3 feature is insufficient to separate cats and dogs

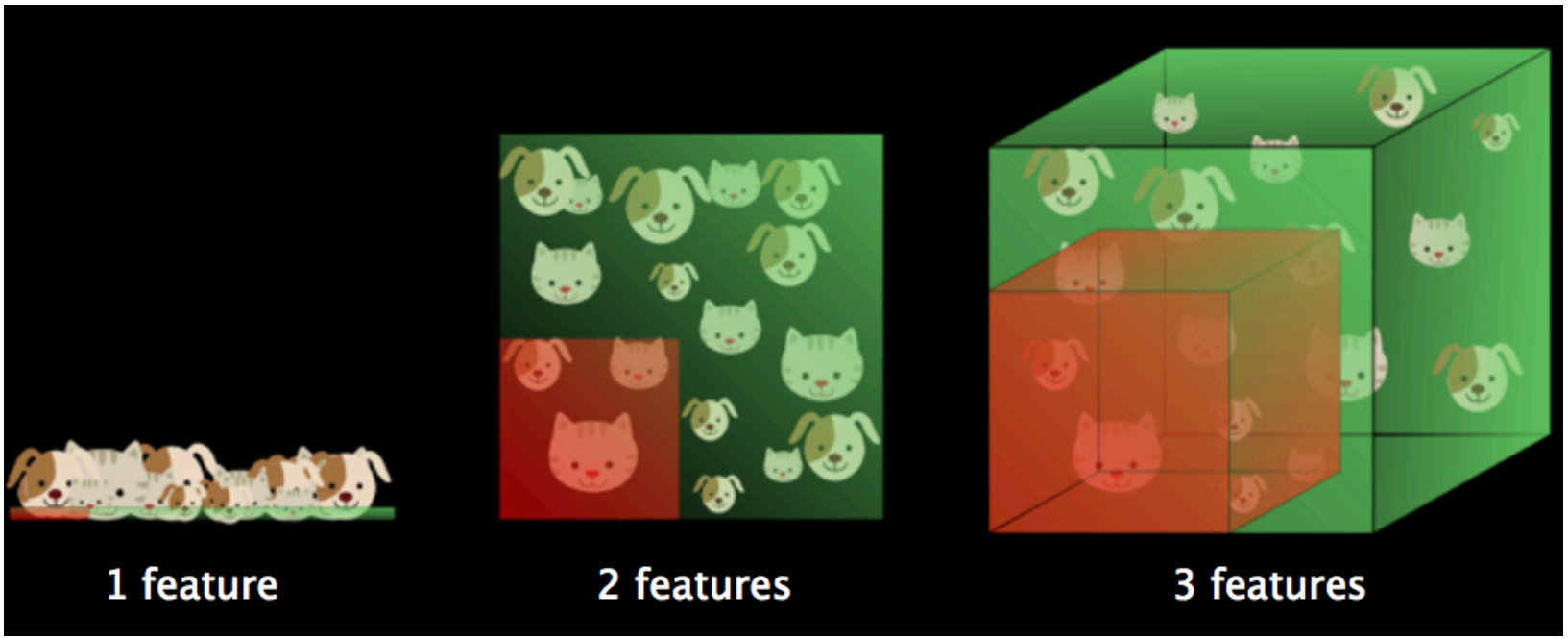
More features may be good



- 3 features are sufficient to separate cats and dogs

Cats versus Dogs

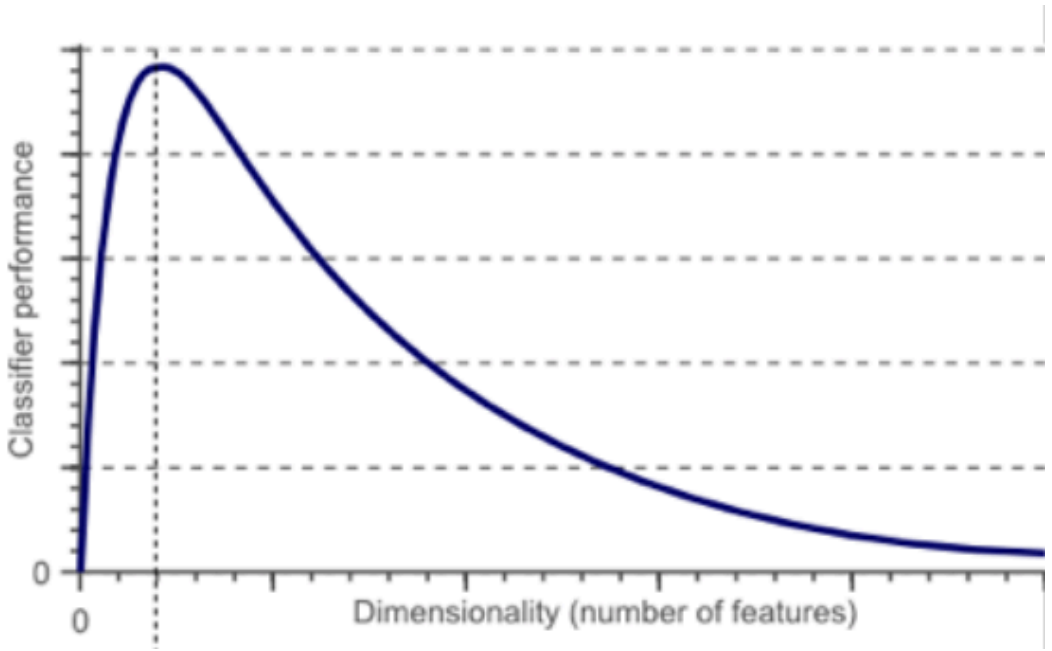
The amount of training data needed to cover grows exponentially with the number of dimensions



More features may be bad

- With more features, each instance might become unique

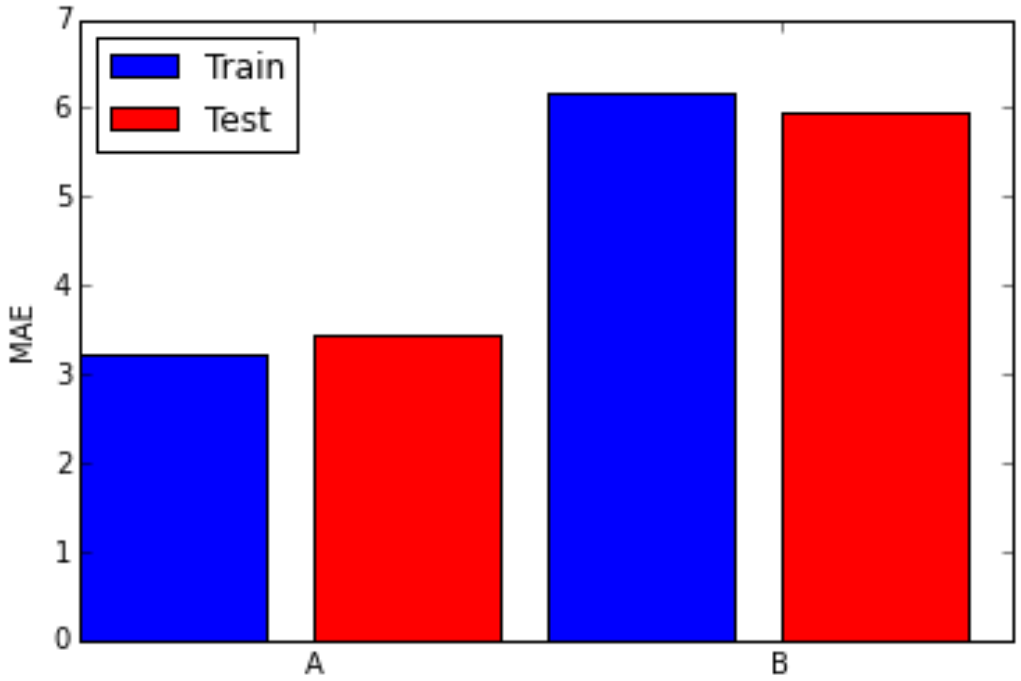
- We assume that the CAT-DOG classification task requires 20% of the data for training
- If the number of features becomes too large, generalisation performance drops



Overview: fitting & tuning

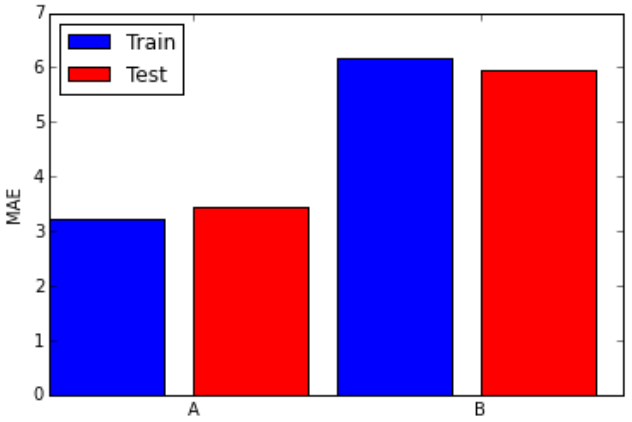
- “The Data Mining Procedure”
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- ROC curves
- Cross Validation
- The curse of dimensionality
- **Under-fitting and Over Fitting**

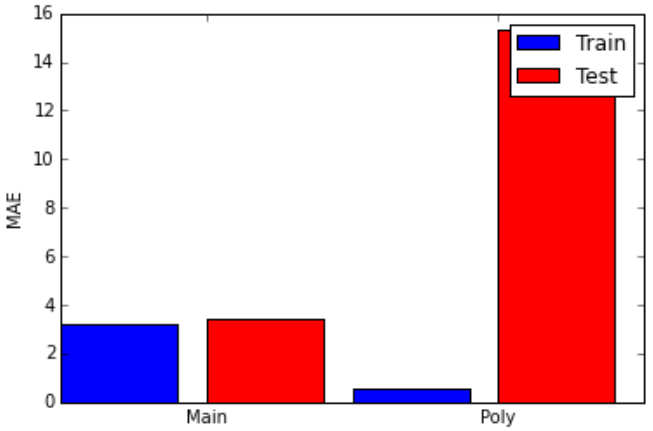
The mystery of high error



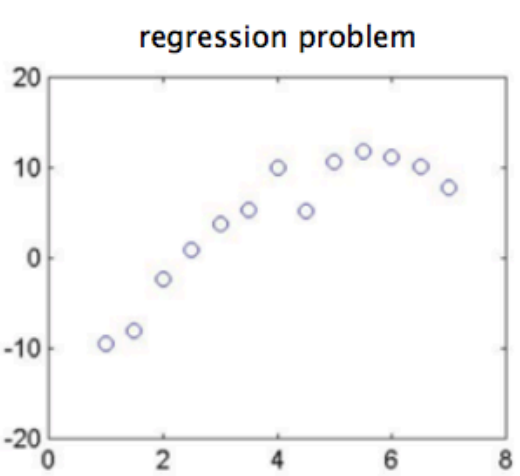
The mystery of high error

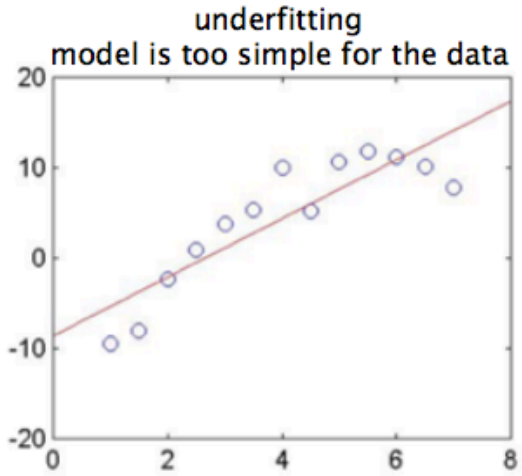
- Is either model **A** or model **B** overfitting?
- Why is the test error of model **B** so high?
- Training error of model **B** is high
  - Model cannot even fit training data
  - **Underfitting**

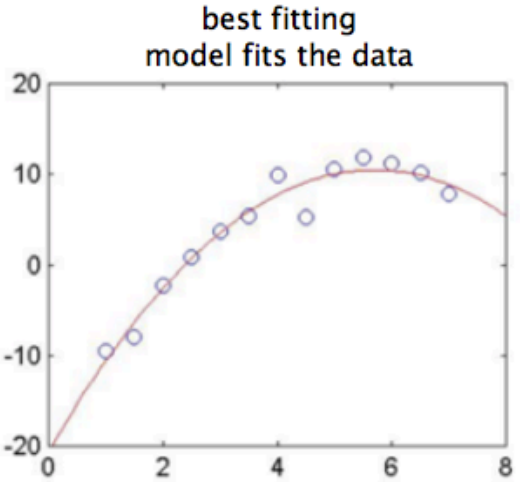


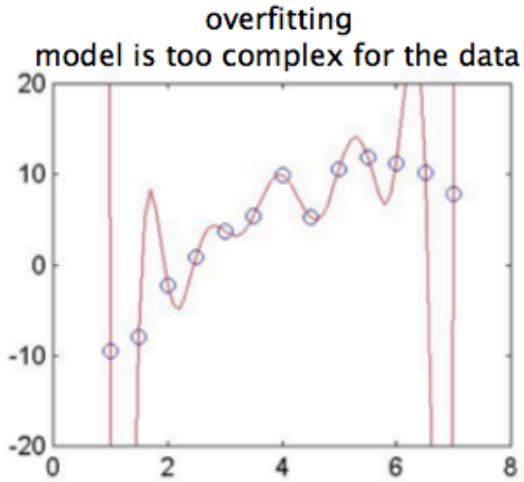


Under vs Over-Fitting







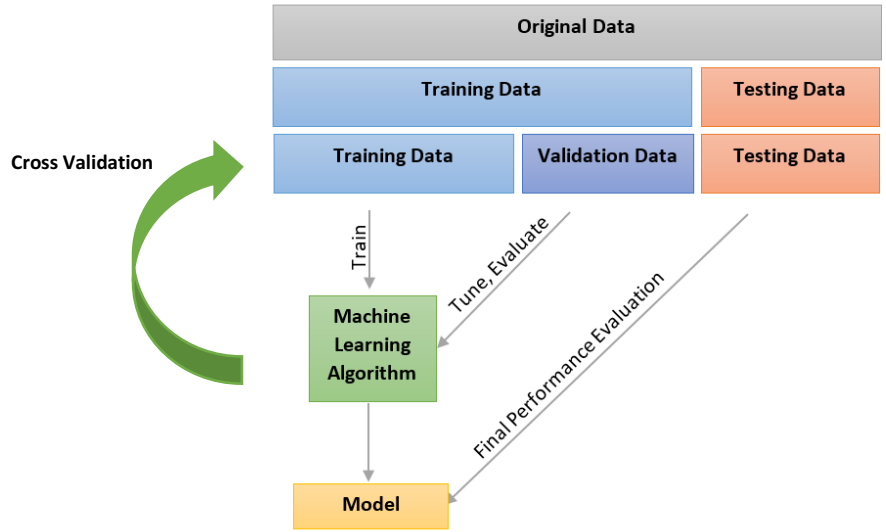


Your model isn't working

- Is it underfitting or overfitting?

	TRAIN ERROR	VAL ERROR
OVER-FIT	LOW	HIGH
UNDER-FIT	HIGH	HIGH
FIT	LOW	LOW

Model selection



- Define **feature sets**
- Choose **hyperparameters**
  - Choose range of **values** to check
- Check performance on train and validation data
  - For as many **combinations** of the above as you can afford
- Pick combination with lowest error

Overview: fitting & tuning

- “The Data Mining Procedure”
- Hyper Parameters
- Model Evaluation
- Confusion Matrix
- ROC curves
- Cross Validation
- The curse of dimensionality
- Under-fitting and Over Fitting

Course Schedule

`v05.09.2017` (subject to change – always check the latest version!)

#	Date	Lectures (Theory - Willem)	Date	Video Lectures (Applications - Chris)	Video Practicals & Notebooks
1	29-08	Introduction to Data Mining	31-08	Introduction to Data Science	Introduction to jupyter, pandas, and scikit-learn
2	05-09	Regression	07-09	Representing Data: Vectors, Types, Databases	Handling & Interpreting Data, Plotting
3	12-09	Classification	14-09	Working with Text Data Part 1 (17-09)	DIY Pandas + scikit-learn
4	19-09	Algorithm Fitting & Tuning	21-09	Working with Text Data Part 2	**No practical** -> time to prepare for midterm.
5	26-09	**Midterm**	28-09	Best Practices, Common Pitfalls & Research	Preprocessing + Pipelines, MNIST Challenge
6	03-10	Data Reduction & Decomposition	05-10	Mining Massive Data, Ensemble Methods	Online / Out-of-Core Learning
7	10-10	Clustering and Graphs	12-10	Applications of Deep Learning	Social Media and Multi-modal Data
8	17-10	Time Series Analysis	19-10	Explaining Models, Ethics, Privacy	Unsupervised Learning: Intuitions and Metrics