Recap Data Mining

Data Mining for Business and Governance*
17/10/2017

**Formerly known as Social Data Mining*

Course Schedule

`v04.10.2017` (subject to change – always check the latest version!)

#	Date	Lectures (Theory - Willem)	Date	Video Lectures (Applications - Chris)	Practicals & Notebooks
1	29-08	Introduction to Data Mining	31-08	Introduction to Data Science	Setting up Jupyter
2	05-09	Regression	07-09	Representing Data	Raw Data to Observations
3	12-09	Classification	14-09	Working with Text Data Part 1	DIY Pandas + scikit-learn
4	19-09	Algorithm Fitting & Tuning	21-09	Best Practices, Common Pitfalls	*No practical*
5	26-09	**Midterm**	28-09		DIY Pandas + scikit-learn
6	03-10	Data Reduction & Decomposition	05-10	Working with Text Data Part 2	Preprocessing + Pipelines
7	10-10	Clustering and Graphs	12-10	Mining Massive Data	Unsupervised Learning
8	17-10	Recap Lecture	19-10	Applications of Deep Learning*	MNIST Challenge*

*Will not be exam material.

Planned bonus videos: "Data Science Research", "Explaining models, Ethics, Privacy".

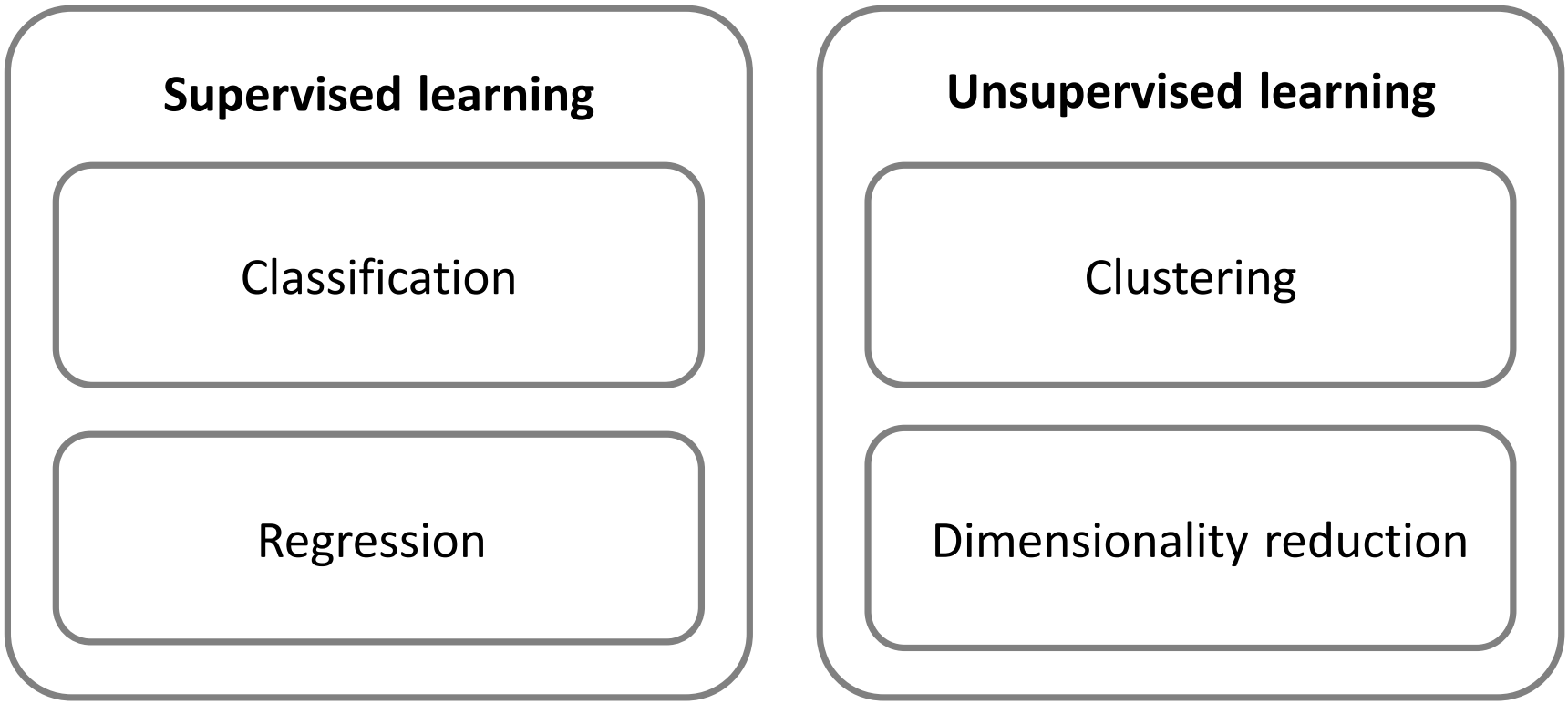
Disclaimer Recap lecture

Not everything what is on the final exam will be covered today

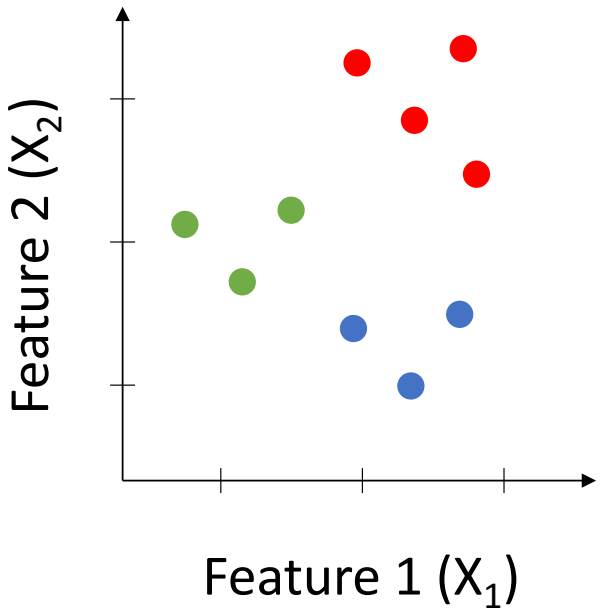
Not all information covered today will necessarily be on the final exam, some is context*

Question about the organization of the final exam?

Supervised vs Unsupervised Learning

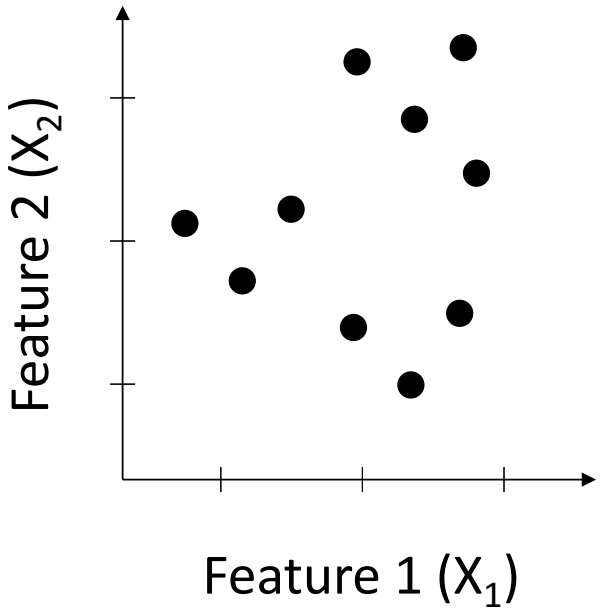


Supervised Learning



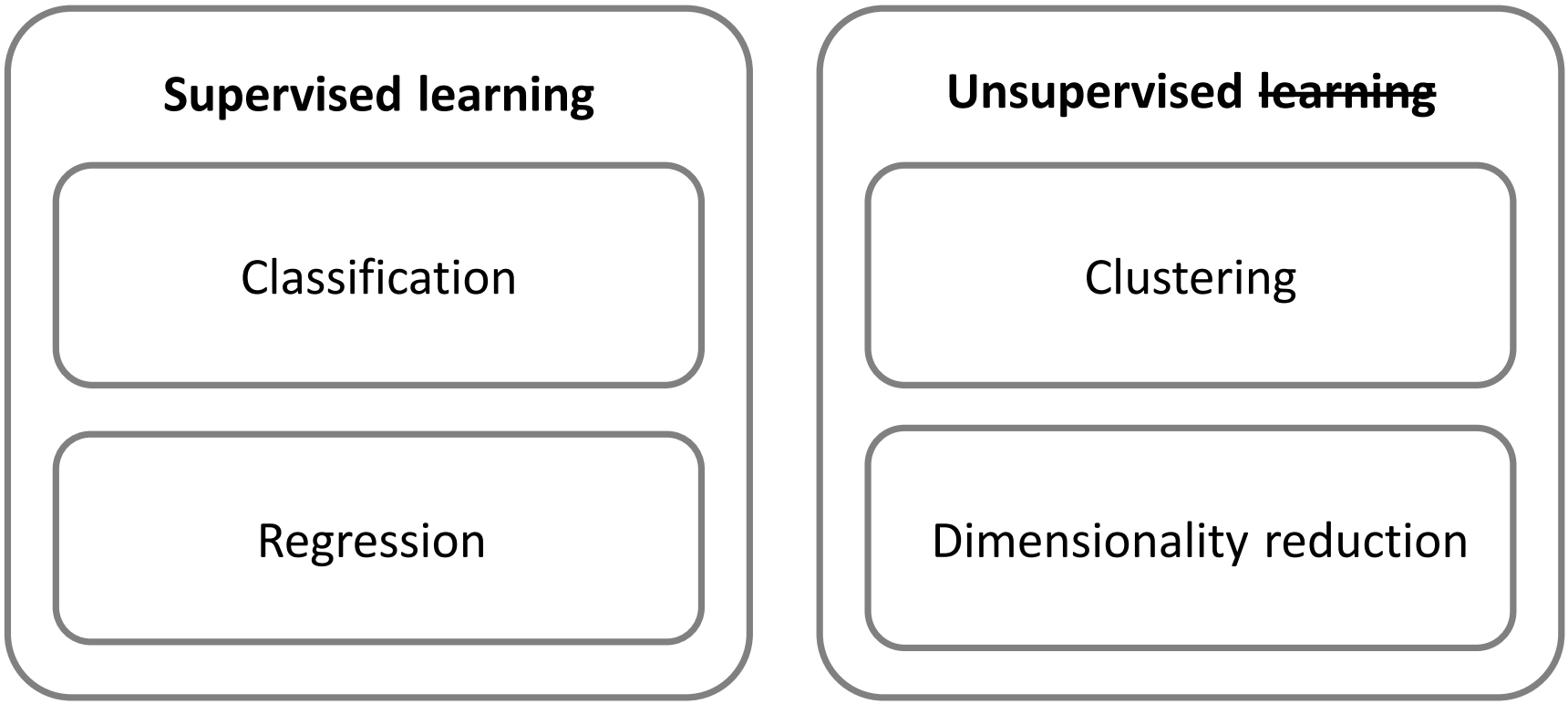
Training Set: $\{(x^{(1)},y^{(1)}), (x^{(2)},y^{(2)}), (x^{(3)},y^{(3)}), \ldots , (x^{(i)},y^{(i)})\}$

Unsupervised Learning

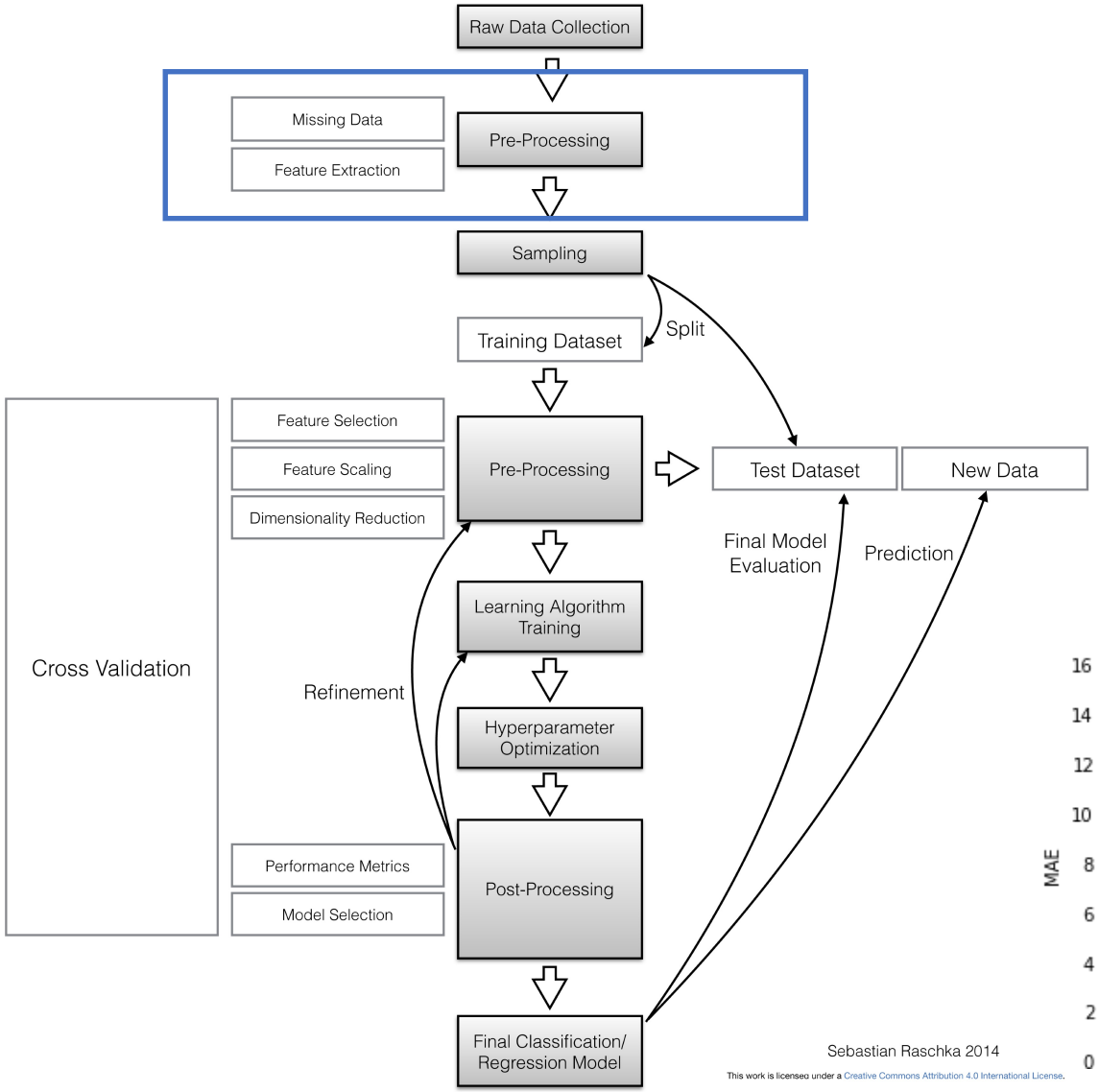


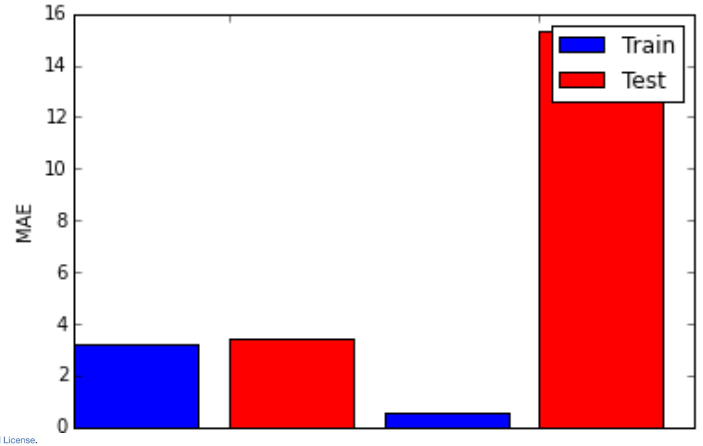
Training Set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(i)}\}$

Supervised vs Unsupervised Learning



A flowchart for supervised learning





Preprocessing: feature transformation (categorical variables)

	color	size	prize	class
0	green	M	10.1	class1
1	red	L	13.5	class2
2	blue	XL	15.3	class1

nominal

green → (1,0,0)
red → (0,1,0)
blue → (0,0,1)

ordinal

M → 1
L → 2
XL → 3

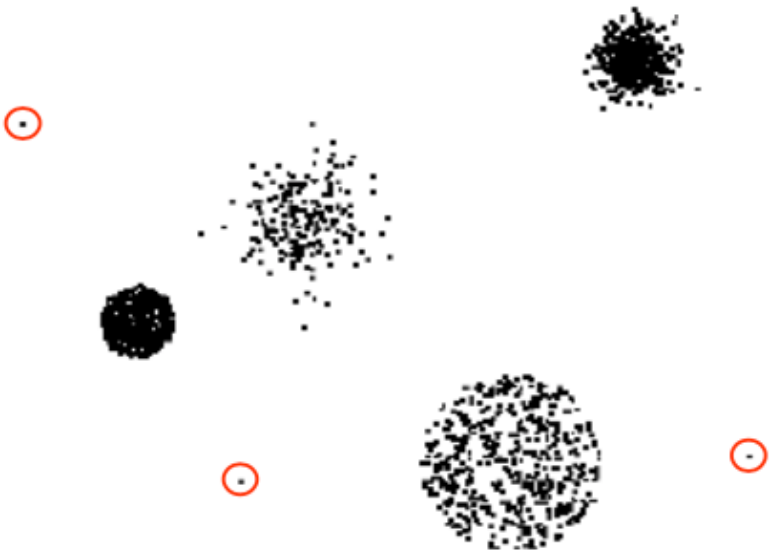
	class	color=blue	color=green	color=red	prize	size
0	0	0	1	0	10.1	1
1	1	0	0	1	13.5	2
2	0	1	0	0	15.3	3

Preprocessing: feature transformation normalization and outlier removal

Z-score $z_i = \frac{x_i - \bar{x}}{s}$

$\bar{X} = 2.5$ S (standard deviation) = 1.87

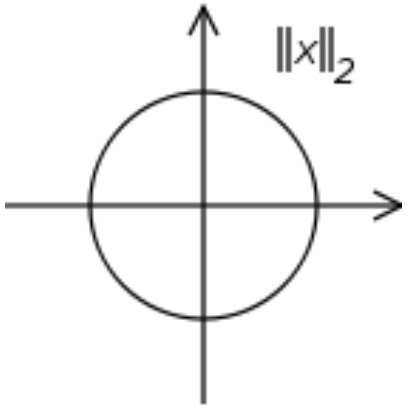
	input	Standardized (z-scoring)	Feature scaling
0	0	-1.336	0
1	1	-0.802	0.2
2	2	-0.267	0.4
3	3	0.267	0.6
4	4	0.802	0.8
5	5	1.336	1



Remove Outliers: data objects with characteristics that are considerably different than most of the other data objects in the data set. Depends strongly on the goal (~ Anomaly detection)

Preprocessing (feature transformation) vector normalization

L2-Norm is $|x|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$

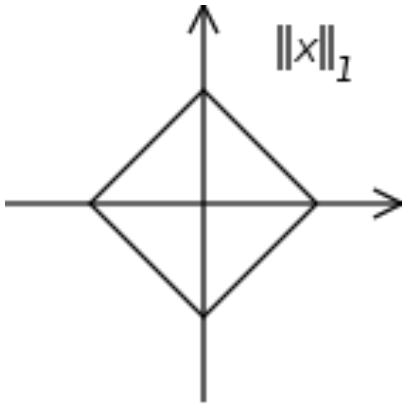


$X_i = (1, -4, 5)$

$l2 = \sqrt{|1|^2 + |-4|^2 + |5|^2} = \sqrt{42}$

$|1, -4, 5|_2 = x_i / l2 = (0.015, 0.617, 0.772)$

L1-Norm is $|x|_1 = \sum_{i=1}^{n} |x_i|$



$l1 = |1| + |-4| + |5| = 10$

$|1, -4, 5|_1 = x_i / L1 = (0.1, -0.4, 0.5)$

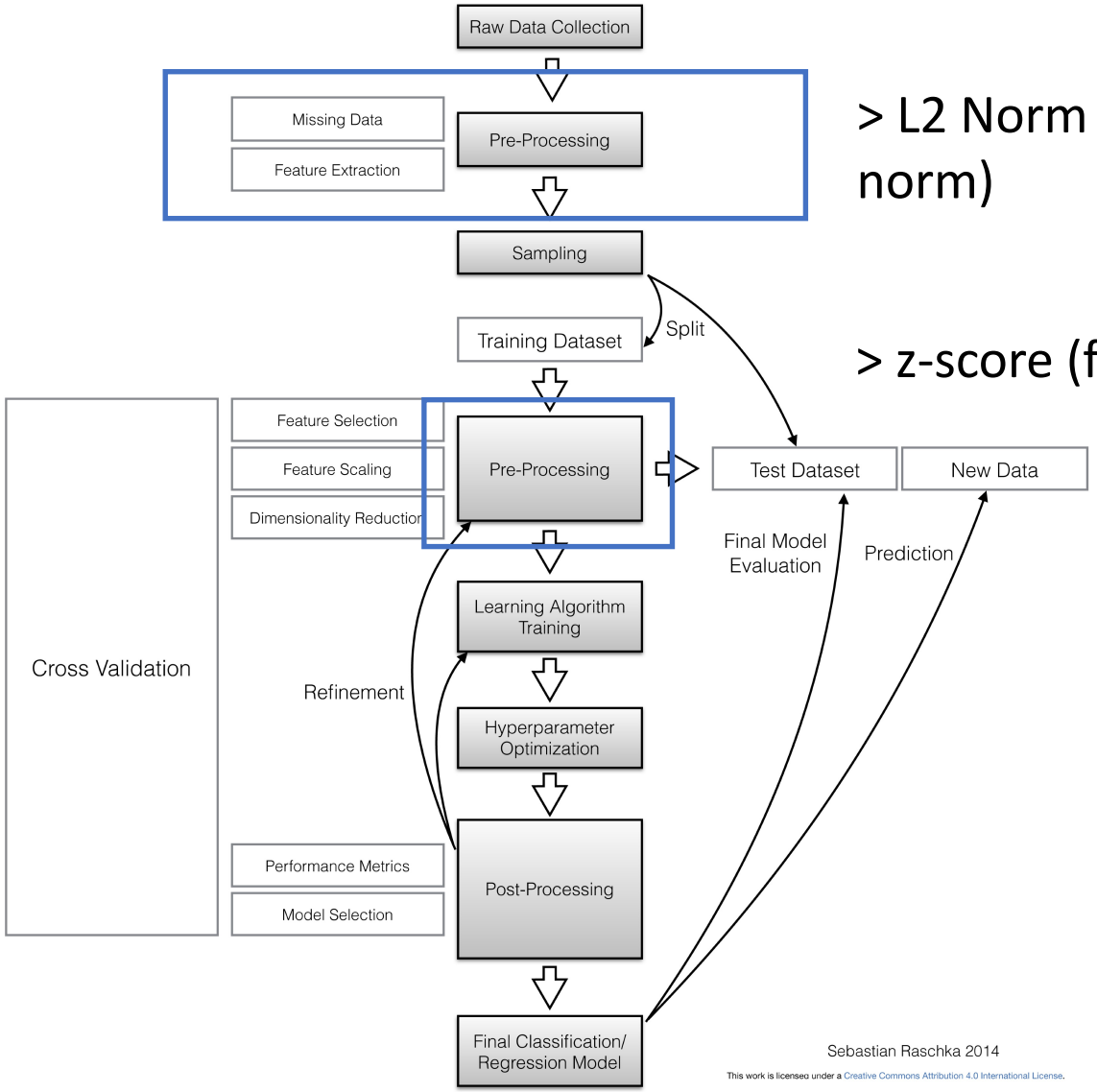
Example Normalization

>>>>> L2 or L1 Norm

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV	CAT. MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24	0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6	0
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	1
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	1
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2	1
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7	0
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9	0
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1	0
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5	0
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9	0
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15	0
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9	0
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7	0
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4	0
0.63796	0	8.14	0	0.538	6.096	84.5	4.4619	4	307	21	380.02	10.26	18.2	0

<<<<< Z-score

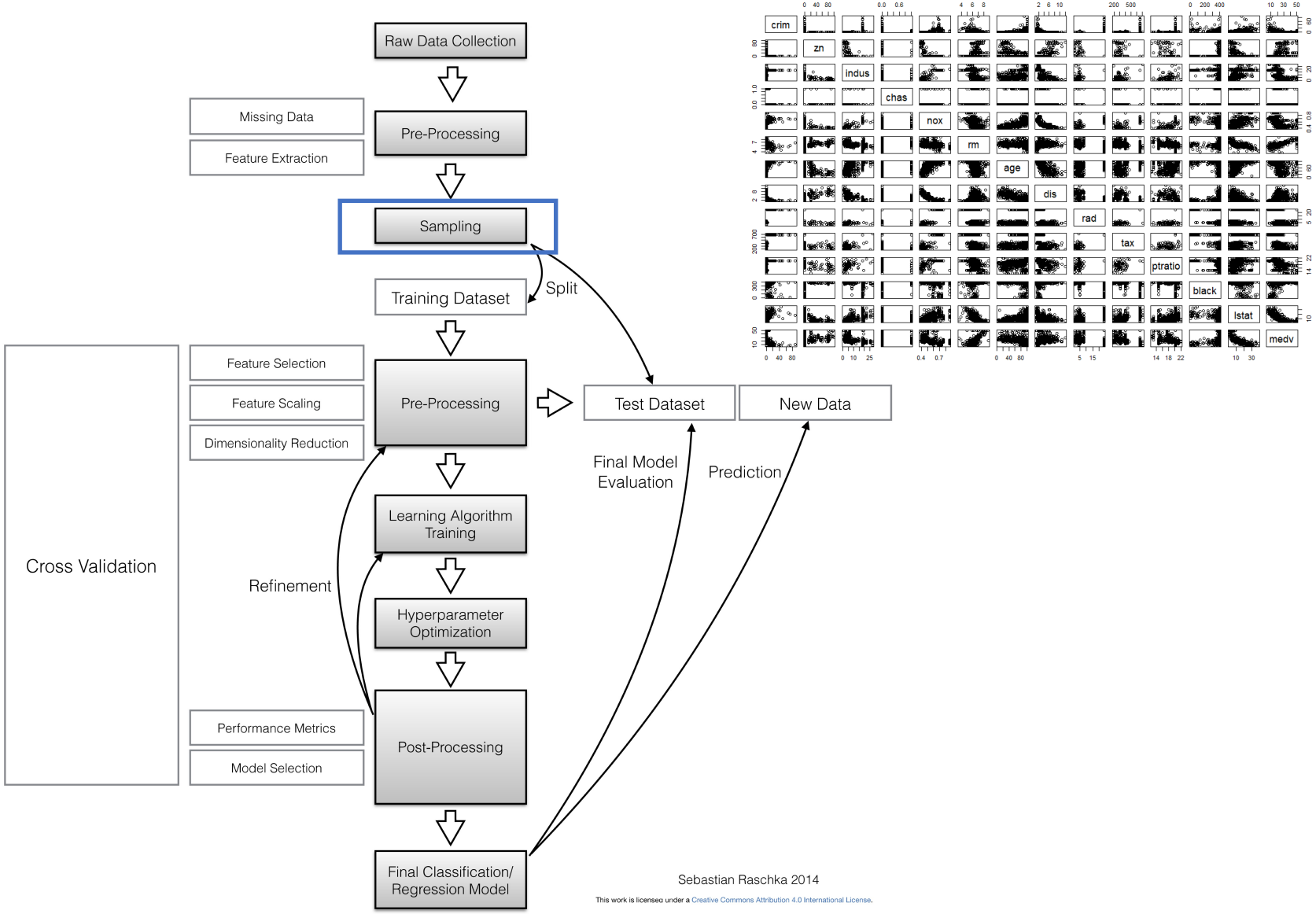
A flowchart for supervised learning



> L2 Norm (vector or instance norm)

> z-score (feature normalization)

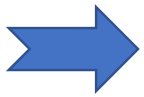
Typical flowchart supervised learning



Data Exploration and Visualization (descriptive analysis)

Sort or rearrange your data

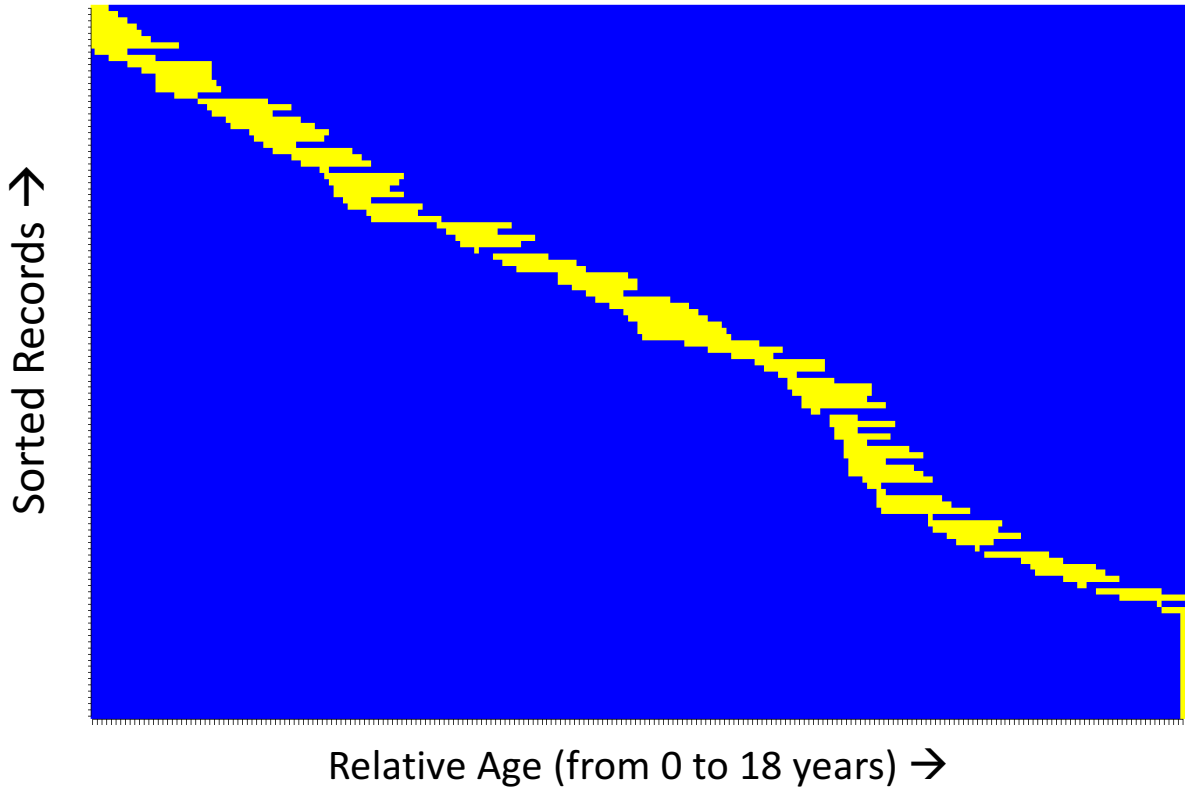
	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0



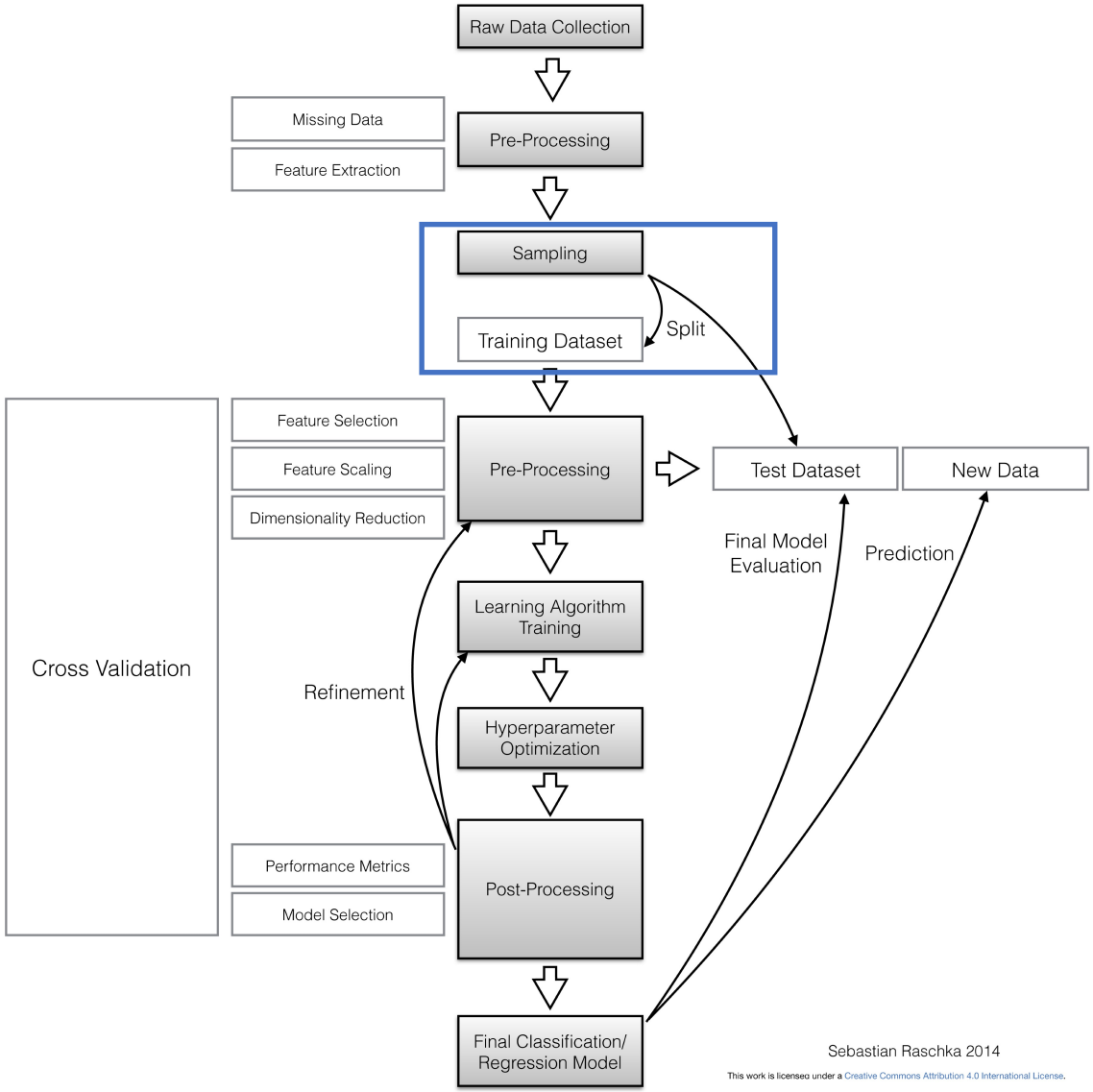
	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

Data Exploration (descriptive analysis)

Electronic Patient Data (EPD), ~60k records of ~180 patients, with each sorted by date of first entry relative to date of birth



Typical flowchart supervised learning



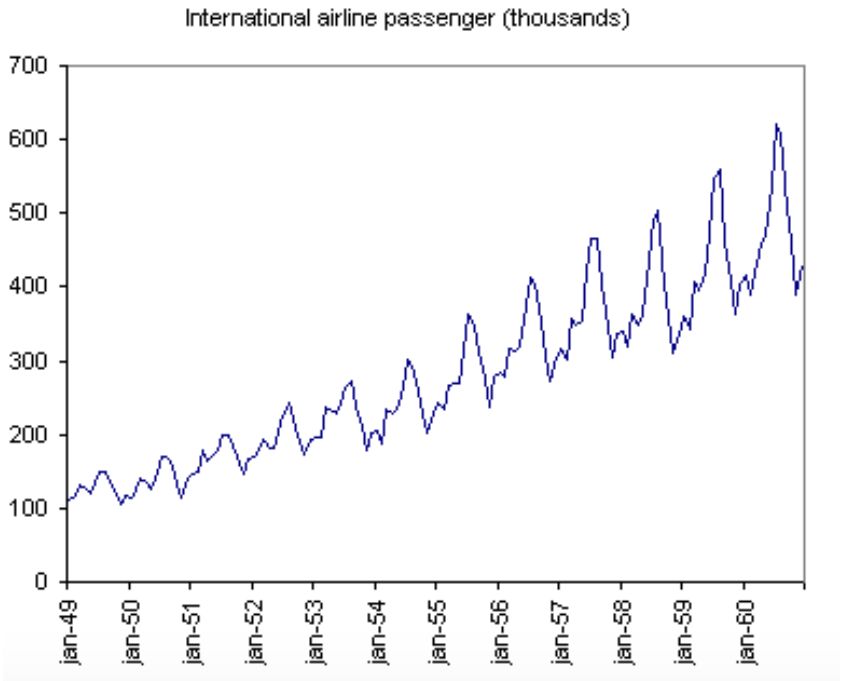
Splitting your data

The fundamental goal is to *generalize* beyond the data instances used to train models

Never touch the test data (until the end)

Test data must belong to the same (statistical) distribution as the training data

Splitting your data



Sequential Split: for example a time series: typically train on a period, for example one 1-6 and test on 7-8, etc.

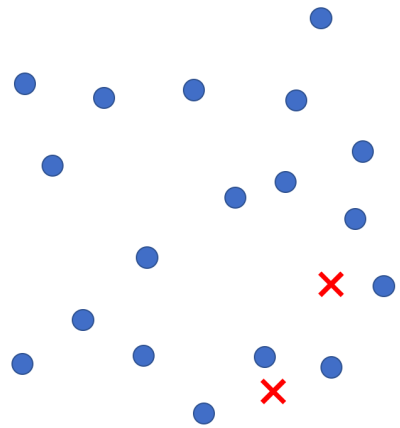
common pitfall is cycles in the data (on different time-scales)

Random Split: blindly assign instances to training and test set

common pitfall is that the training is not similar to the test data

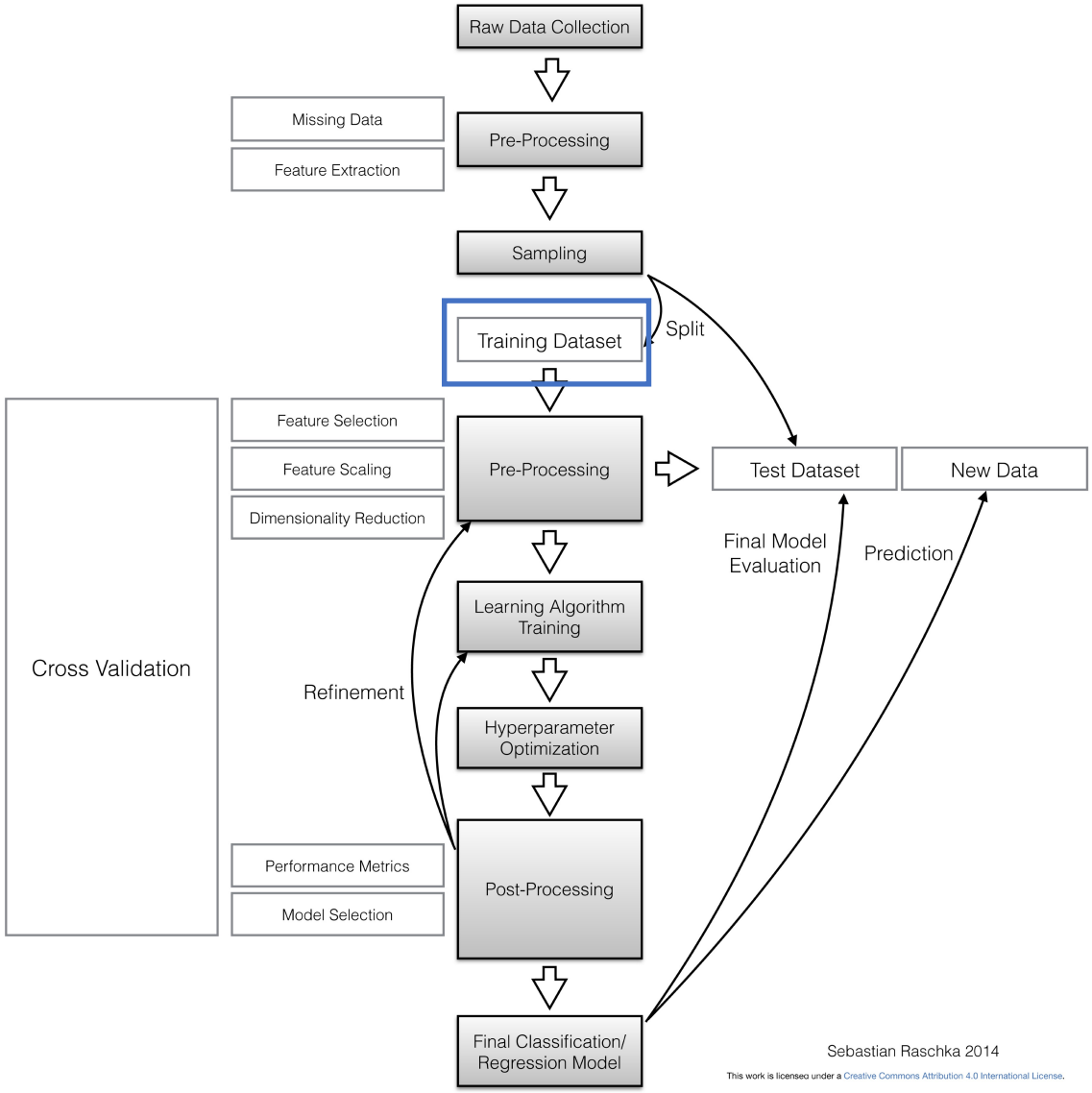
Sampling and splitting your data

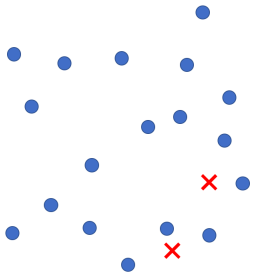
In the case of small data, you want to check (stratify) your data in terms of target, or at least check if the ratio's are representative



In the case of unbalanced data you might want to stratify your data

Typical flowchart supervised learning

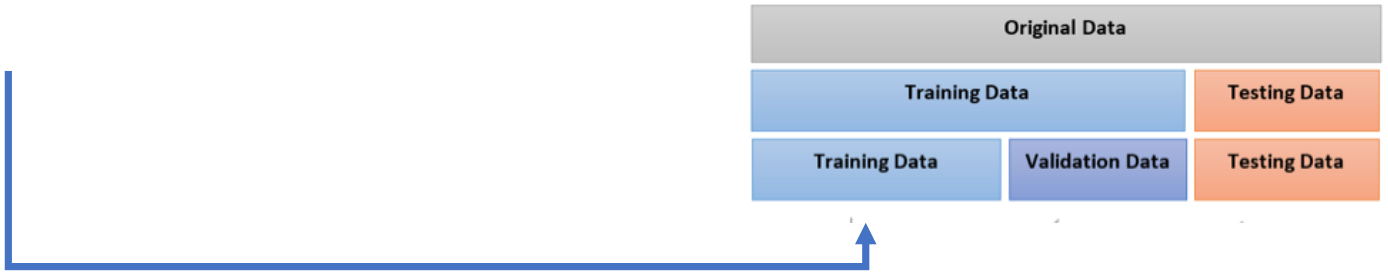




Resample Training data: a strategy to deal with unbalanced classes

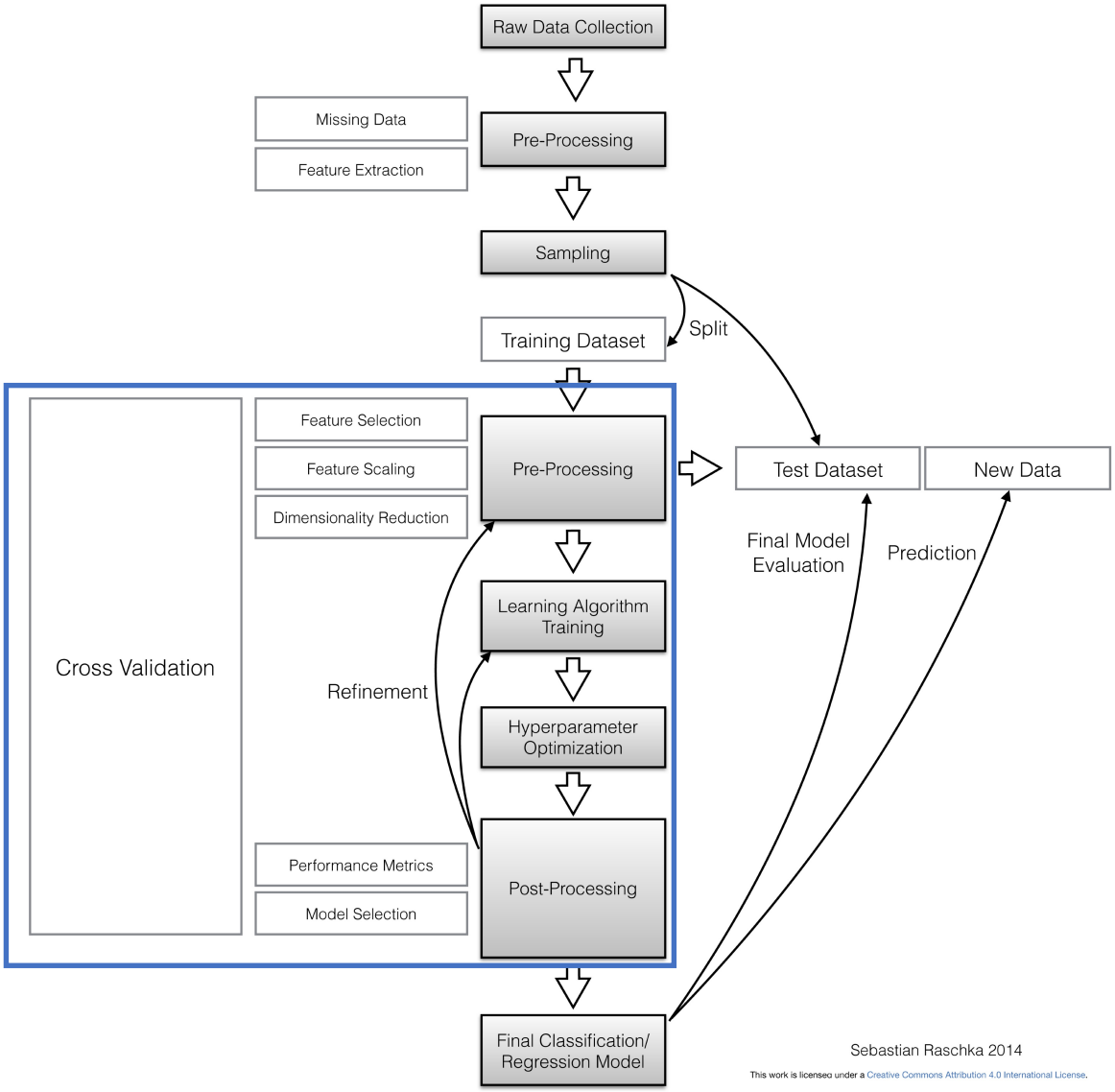
Downsampling creates a balanced dataset by matching the number of samples in the minority class with a random sample from the majority class

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



”Unbalanced data is a sort of small data”

Typical flowchart supervised learning



Cross Validation

Cross-validation is a statistical technique to estimate the prediction error rate by splitting the data into training and **evaluation** datasets

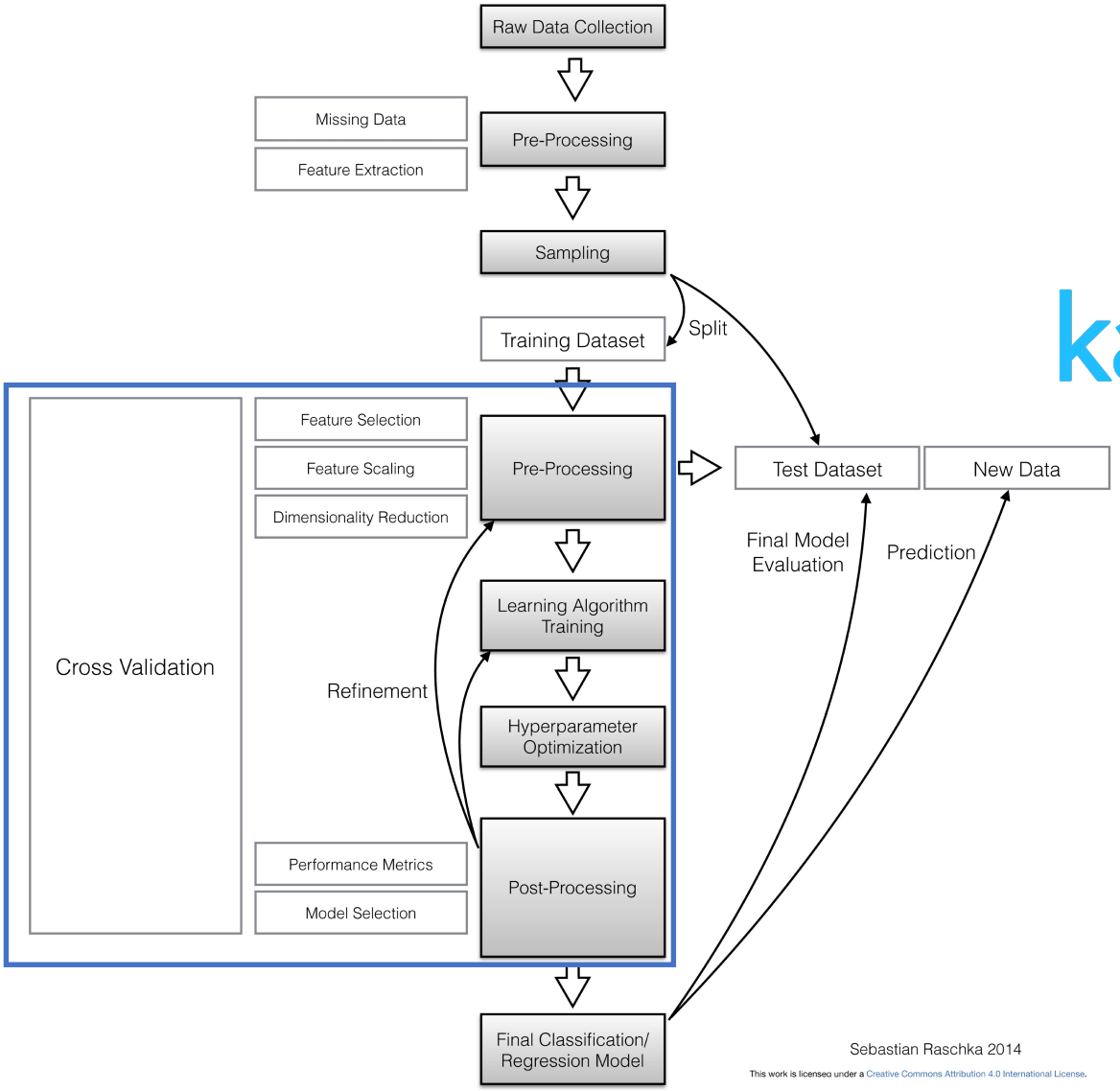
Reason 1: To estimate the performance of the **learned model** from available data using one algorithm. Often motivated by small datasets

When you apply cross validation on test data you can no longer know how well your findings generalize

Reason 2/3: To compare the performance of two or more different algorithms, or to find the best hyperparameters, for the available data. Often motivated by the search for an optimal solution

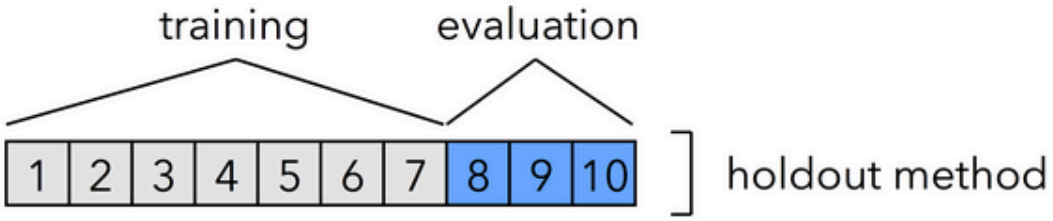
*Requires a training, **validation** and test set*

Typical flowchart supervised learning



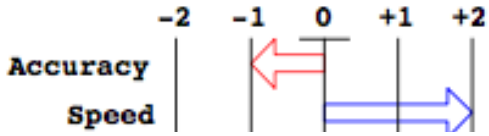
kaggle™

Hold out method

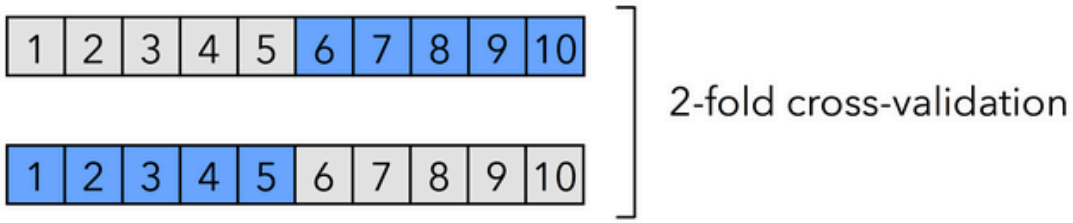


1. Separate the available examples into training set (examples used for training) and validation set (the rest of the data).

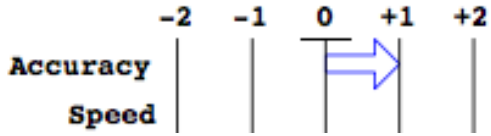
2. Calculate the accuracy over the testing



K-fold Cross Validation

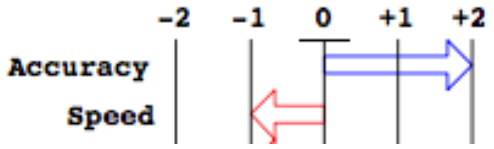


1. Partition randomly the available examples into k disjoint subsets.
2. Use one of the partitions as a validation set to evaluate a model generated by considering the other subsets as the training set.
3. Repeat this process with all subsets and average the obtained errors.

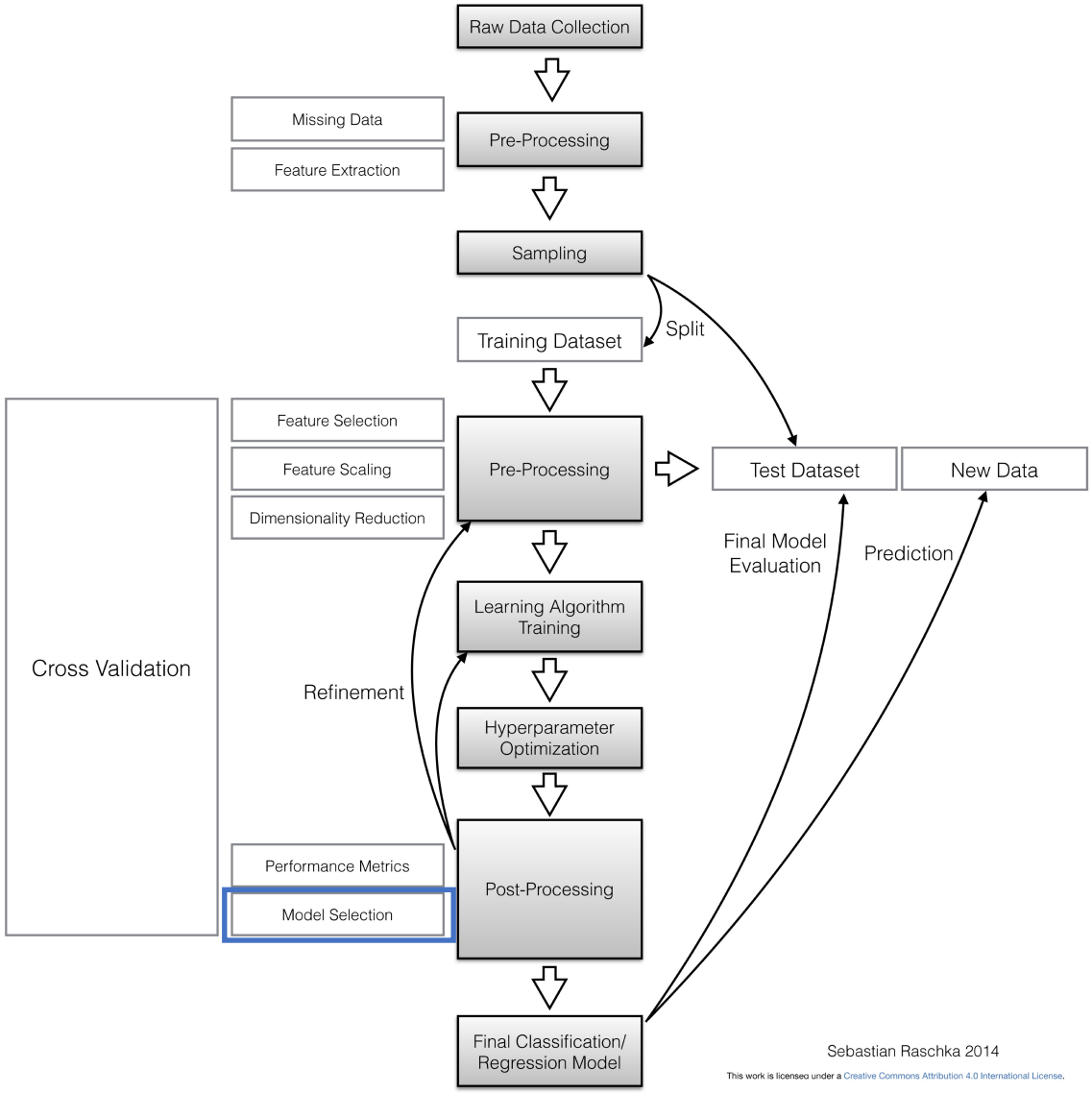


Leave one out

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.

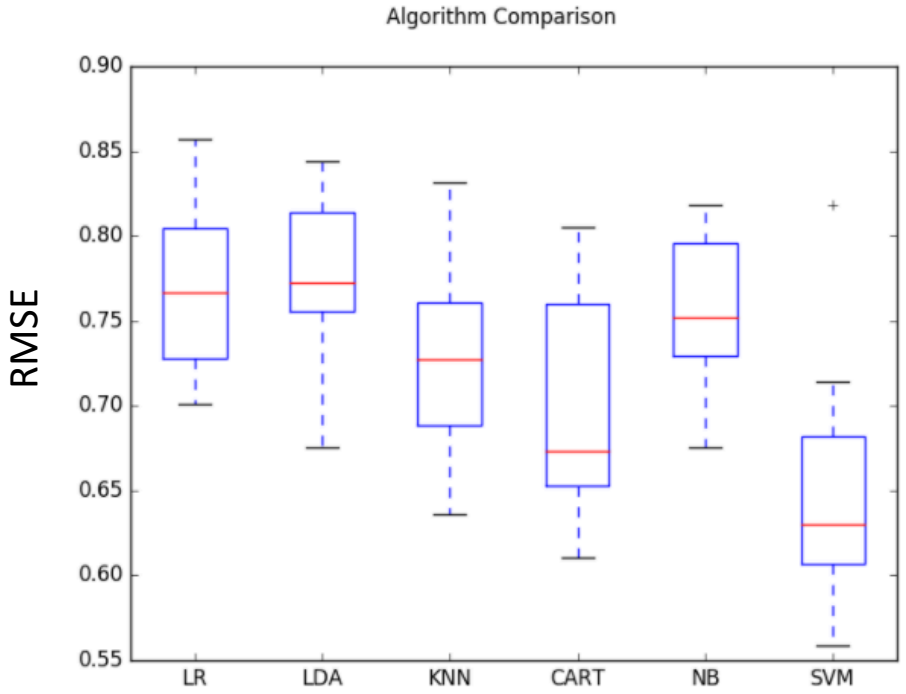


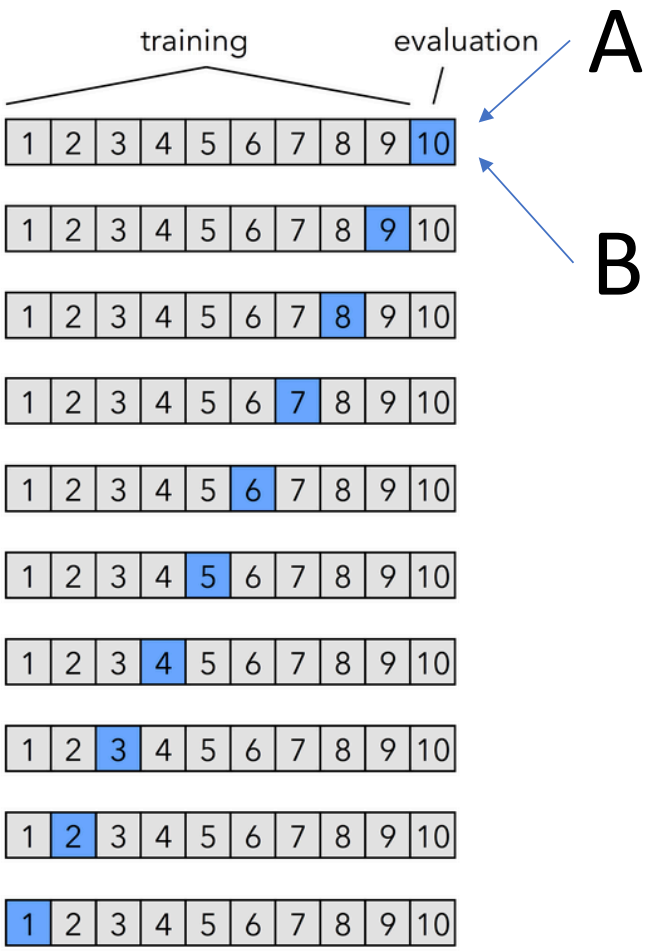
Typical flowchart supervised learning



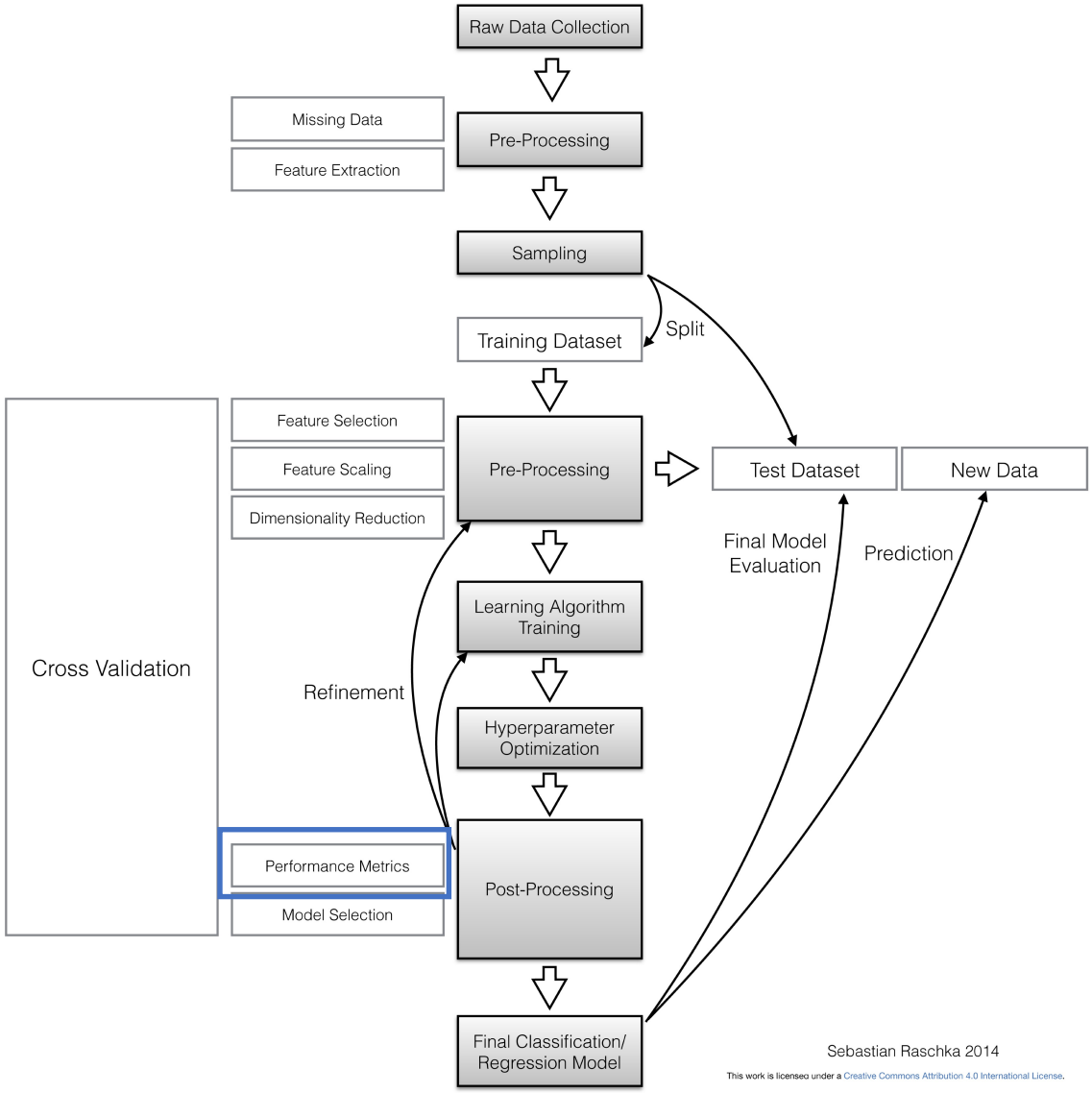
Model Selection: use validation data

Pair-wise Comparison
of algorithm A and B
~(allows pair-wise t-test)





Typical flowchart supervised learning



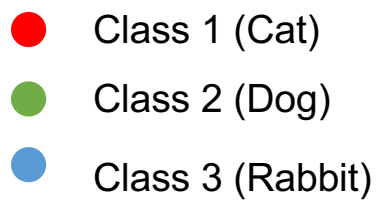
Multiclass Classification with Logistic Regression

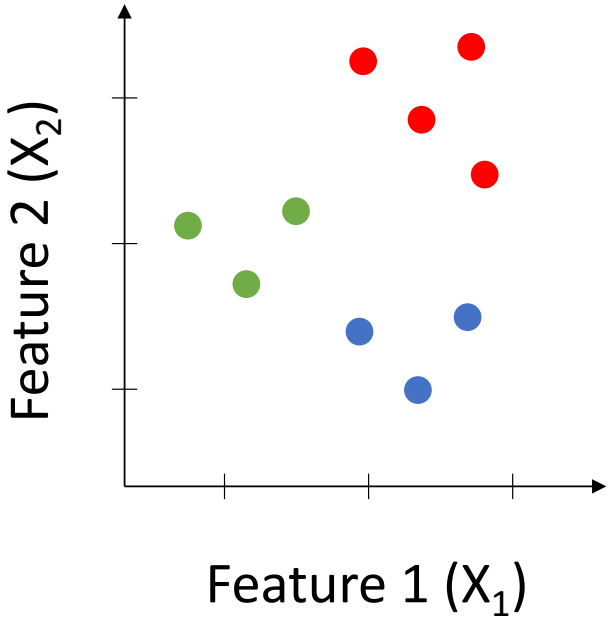
Examples of classification tasks:

- Iris Species → {setosa, versicolor, virginica}
- Email folder tagging → {primary, social, promotions, updates}
- Weather → {sunny, cloudy, snow, rain}

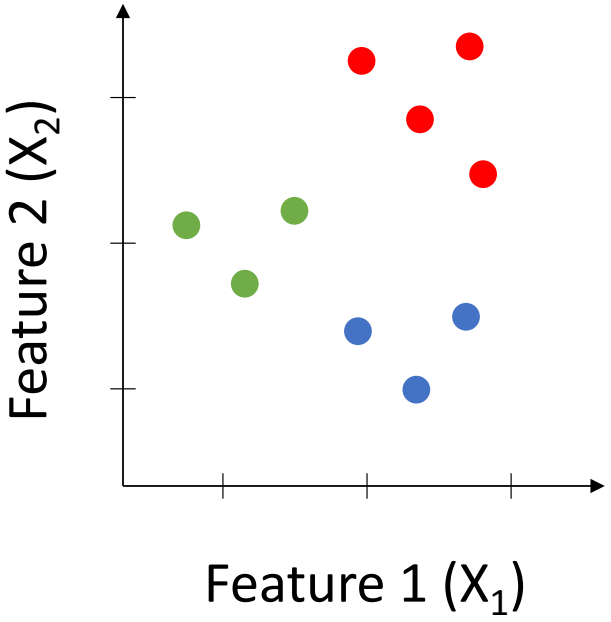
Assignment to a class: $y \varepsilon \{1,2….\}$

0 = can also start at 0





Multiclass Classification
Example logistic regression: of one versus all

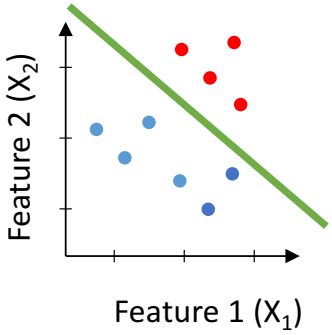


Class 1 (Cat)

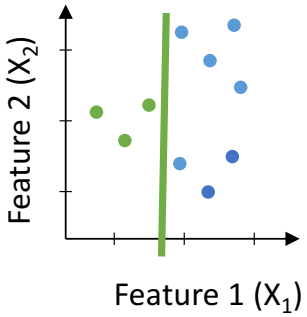
Class 2 (Dog)

Class 3 (Rabit)

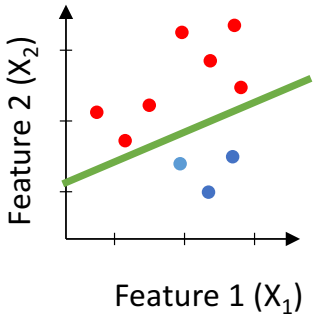
$f(X)^i = P(Y=i|X)$ $\quad i = \{1,2,3\}$



$f(X)^1$

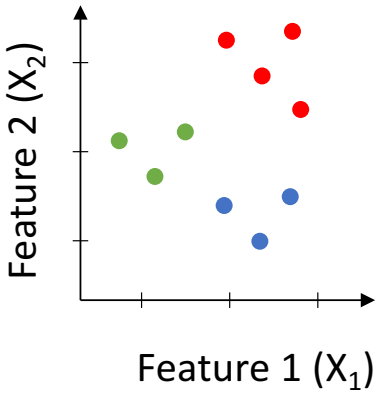


$f(X)^2$



$f(X)^3$

Multiclass Classification
Example logistic regression: of one versus all



$$f(X)^i = P(Y=i|X) \qquad i = \{1,2,3\}$$

$$\max_i f(X)^i$$

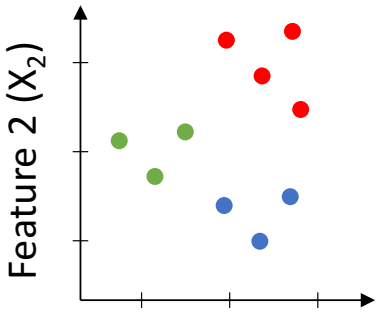
**number of classifiers expands rapidly)*

	X1	X2	Y
1	2.781	2.551	0
2	1.465	2.362	0
3	3.397	4.400	0
5	3.064	3.005	1
6	7.628	2.759	1
7	5.332	2.089	1
8	6.923	1.771	2
9	8.675	-0.242	2
10	7.674	3.509	2

$\hat{Y}$(1vs0)	$\hat{Y}$(2vs0)	$\hat{Y}$(2vs1)
0.299	0.329	0.423
0.146	0.147	0.672
0.085	0.082	0.424
0.831	0.247	0.542
0.955	0.672	0.324
0.862	0.782	0.450
0.677	0.972	0.645
0.891	0.999	0.558
0.754	0.672	0.981

Y
0
0
0
1
1
1
2
2
2

Multiclass Classification
Example logistic regression: of one versus all



$$f(X)^i = P(Y=i|X) \qquad i = \{1,2,3\}$$

$$\max_i f(X)^i$$

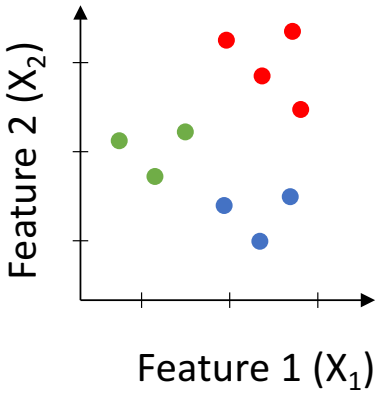
**number of classifiers expands rapidly)*

	X1	X2	Y
1	2.781	2.551	0
2	1.465	2.362	0
3	3.397	4.400	0
5	3.064	3.005	1
6	7.628	2.759	1
7	5.332	2.089	1
8	6.923	1.771	2
9	8.675	-0.242	2
10	7.674	3.509	2

Y(1v0)	Y(2v0)	Y(2v1)	Y(0v1)	Y(0v2)	Y(1v2)
0.299	0.329	0.423	0.701	0.671	0.577
0.146	0.147	0.672	0.854	0.853	0.328
0.085	0.082	0.424	0.915	0.918	0.576
0.831	0.247	0.542	0.169	0.753	0.458
0.955	0.672	0.324	0.045	0.328	0.676
0.862	0.782	0.45	0.138	0.218	0.55
0.677	0.972	0.645	0.323	0.028	0.355
0.891	0.999	0.558	0.109	0.001	0.442
0.754	0.672	0.981	0.246	0.328	0.019

Y
0
0
0
1
1
1
2
2
2

Multiclass Classification
Example logistic regression: of one versus all



$$f(X)^i = P(Y=i|X) \qquad i = \{1,2,3\}$$

$$\max_i f(X)^i$$

**number of classifiers expands rapidly)*

	X1	X2	Y
1	2.781	2.551	0
2	1.465	2.362	0
3	3.397	4.400	0
5	3.064	3.005	1
6	7.628	2.759	1
7	5.332	2.089	1
8	6.923	1.771	2
9	8.675	-0.242	2
10	7.674	3.509	2

$\hat{Y}$(1vs0)	$\hat{Y}$(2vs0)	$\hat{Y}$(2vs1)
0	0	1
0	0	2
0	0	1
1	0	2
1	2	1
1	2	1
1	2	2
1	2	2
1	2	2

$\hat{Y}$
0
0
0
1
1
1
2
2
2

Multiclass classification: evaluation

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

Accuracy is on vs off diagonal

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Accuracy = (5 + 3 + 11) / 27 = 0.70

Multiclass classification: evaluation

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

Some other metrics have a "micro" or "macro" solution, For example, take precision (PRE)

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

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

$$PRE_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FP_1 + \cdots + FP_k}$$

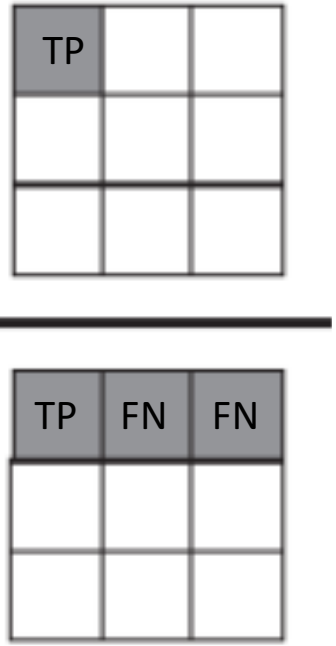
$$PRE_{macro} = \frac{PRE_1 + \cdots + PRE_k}{k}$$

Multiclass classification: evaluation

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

$$Recall = \frac{TP}{TP + FN}$$

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11



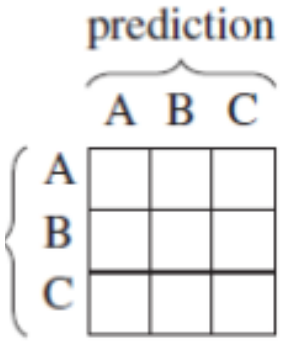
$Recall_{(cat)} = 5 / (5 + 3 + 0)$

$Recall_{(dog)} = 3 / (2 + 3 + 1)$

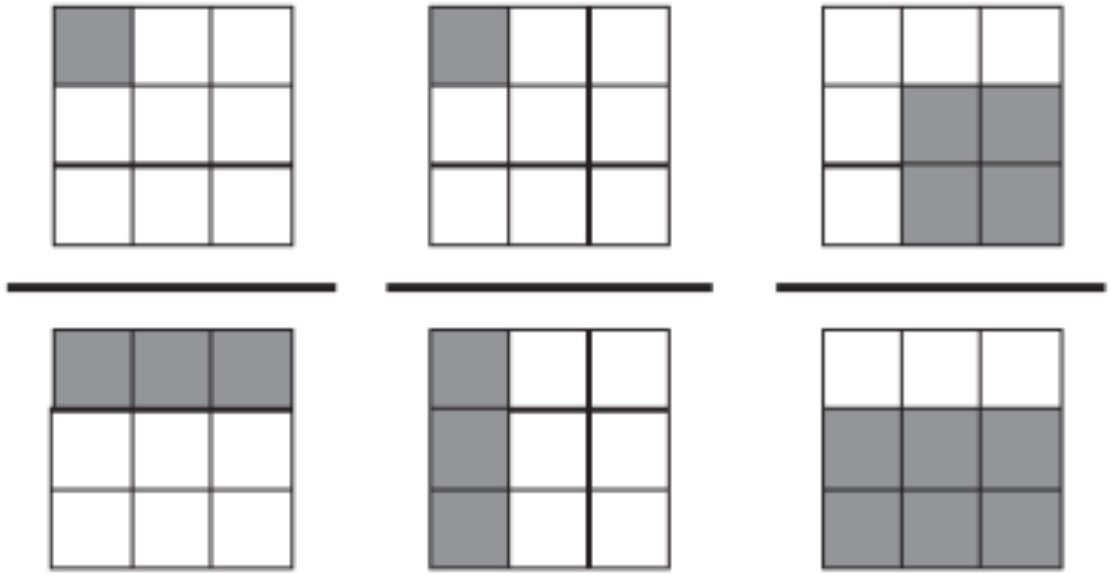
$Recall_{(rabit)} = 11 / (0 + 2 + 11)$

$$Recall_{(macro)} = \frac{\frac{5}{8} + \frac{3}{6} + \frac{11}{13}}{3} = 0.657$$

Multiclass classification: evaluation



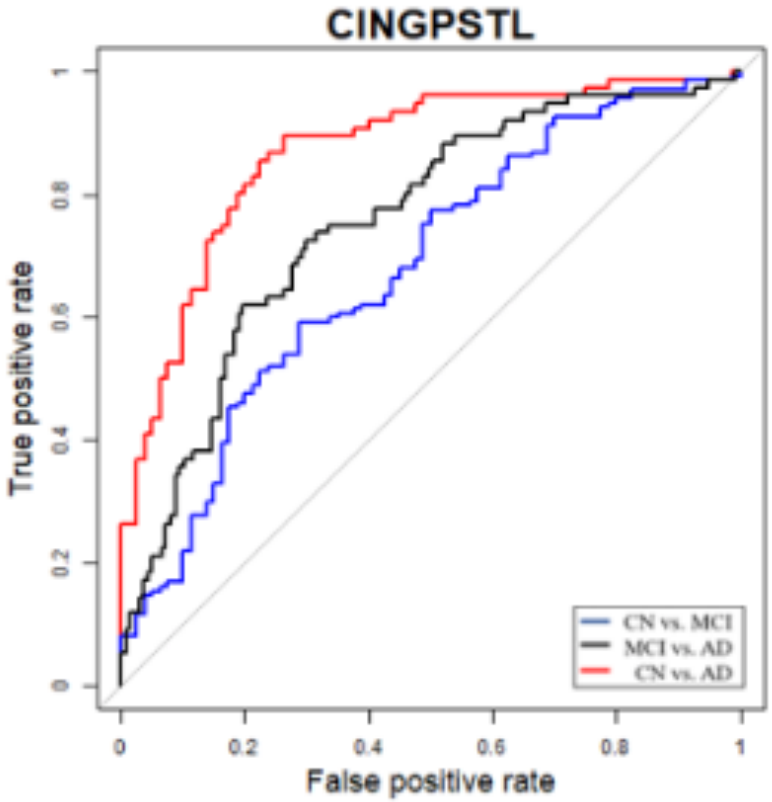
confusion matrix



Recall_A Precision_A Specificity_A

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

Example thesis last year
ROC curves: multiclass classification



CN vs MCI

MCI vs AD

CN vs AD

Example scientific paper:
~~"One metric to rule them all"~~

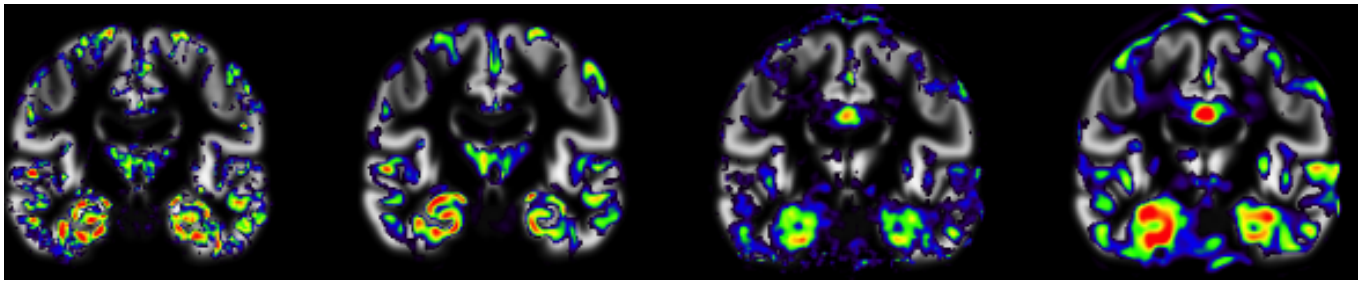
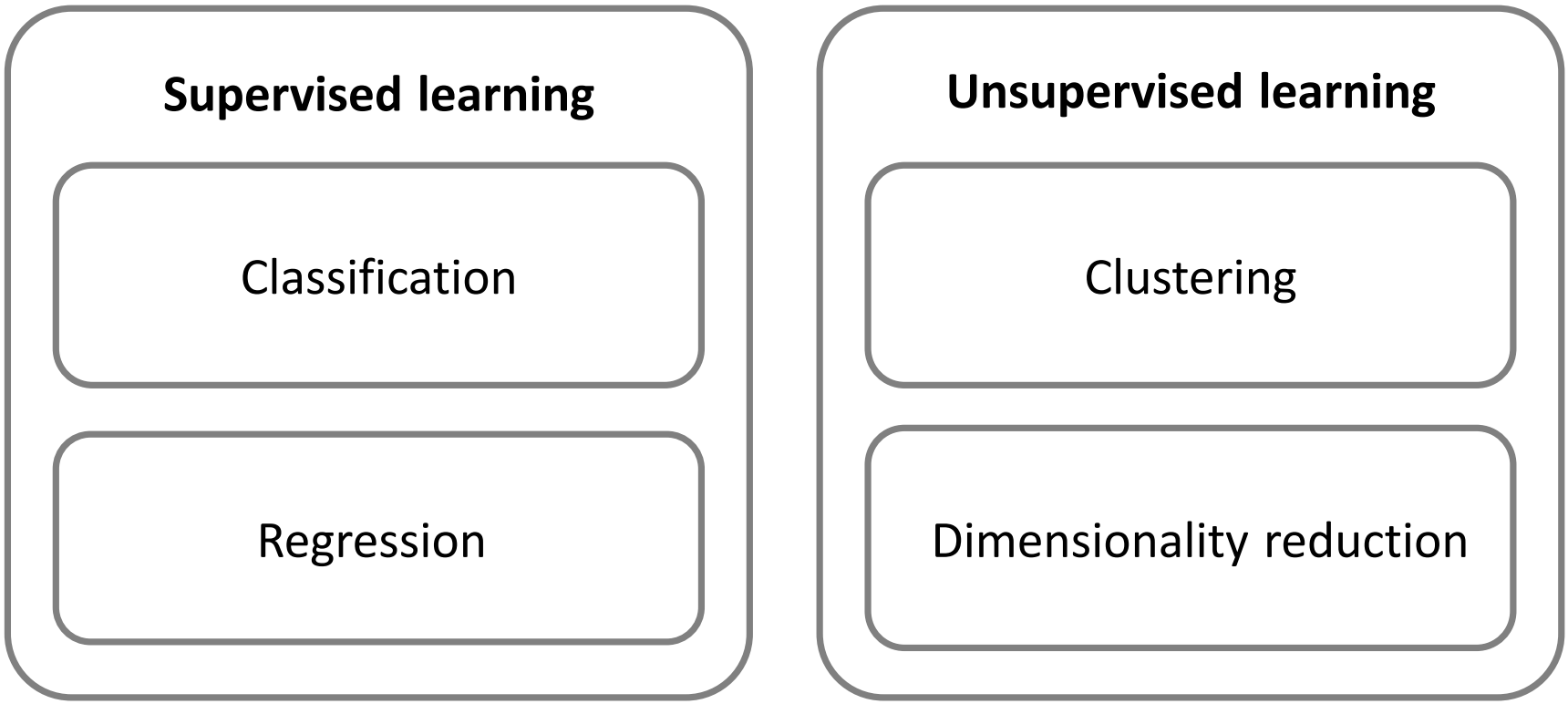
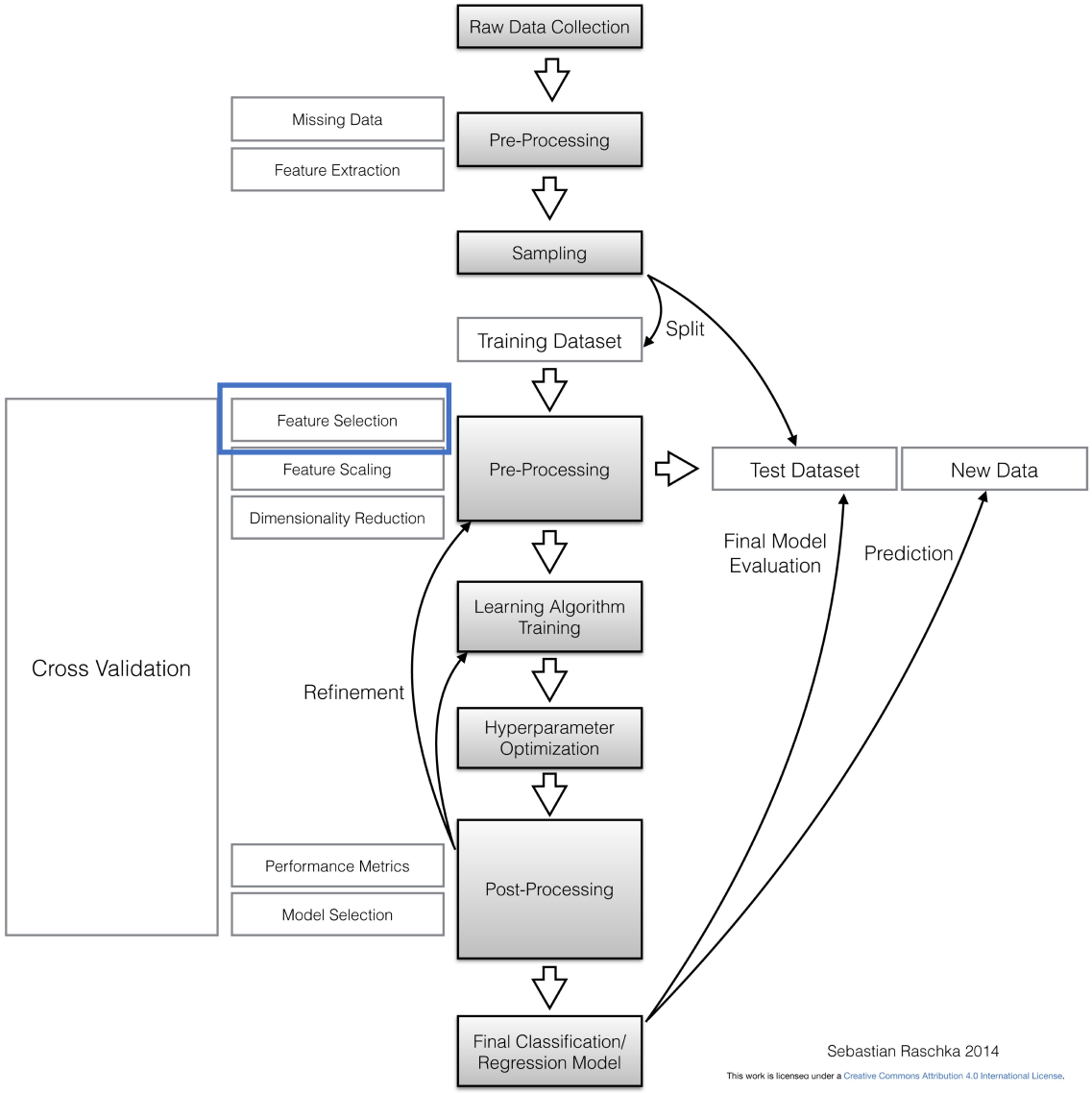


Image type	Classifier	Task	AUC	Bal acc	Sens	Spec
T1	Linear SVM	AD vs CN	94.2%	88.8%	92.8%	84.8%
		MCIc vs CN	85.5%	80.5%	66.8%	89.3%
		MCIc vs MCInc	73.8%	66.5%	64.9%	69.3%
		AD-Aβ+ vs CN-Aβ-	93.6%	87.9%	90.5%	85.5%
		MCIc-Aβ+ vs MCInc-Aβ+	73.6%	65.8%	69.8%	60.3%
FDG PET	Linear SVM	AD vs CN	96.7%	91.1%	95%	87.2%
		MCIc vs CN	89.1%	83.5%	74.7%	89.2%
		MCIc vs MCInc	80.5%	75.2%	78.5%	69.4%
		AD-Aβ+ vs CN-Aβ-	98.6%	94.4%	95.6%	93.4%
		MCIc-Aβ+ vs MCInc-Aβ+	80.8%	72.8%	77%	67%

Supervised vs Unsupervised Learning



Typical flowchart supervised learning



Clustering

Identify a finite set of categories, classes or groups (clusters) in the data

Objects in the *same* cluster should be as similar as possible

Objects in *different* clusters should be as dissimilar as possible

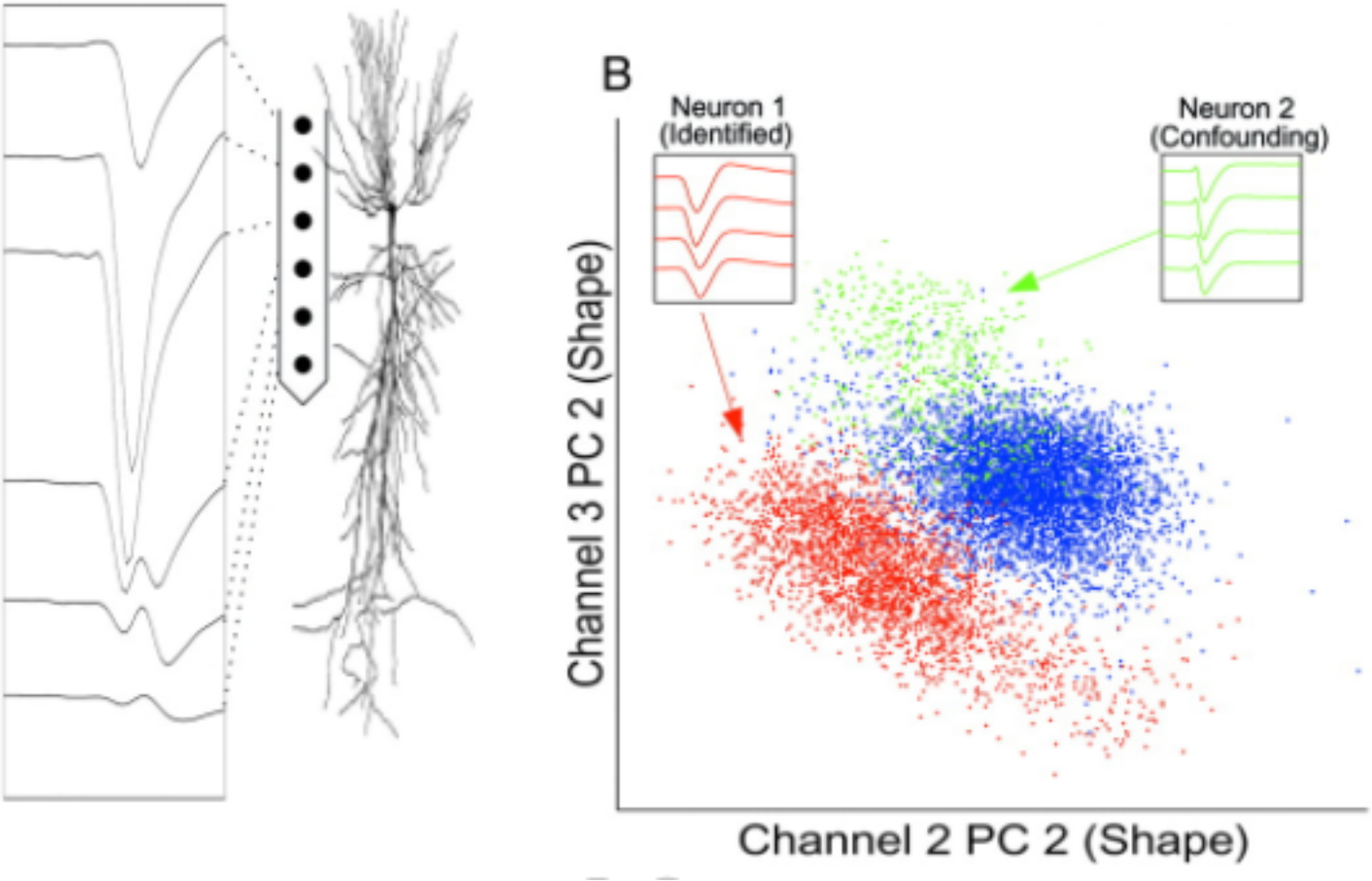
Clustering

Depends on distance or similarity metric *(euclidian, Manhattan, hamming, cosine, jaccard..)*

Different algorithms: K-means, Density – based, hierarchical clustering

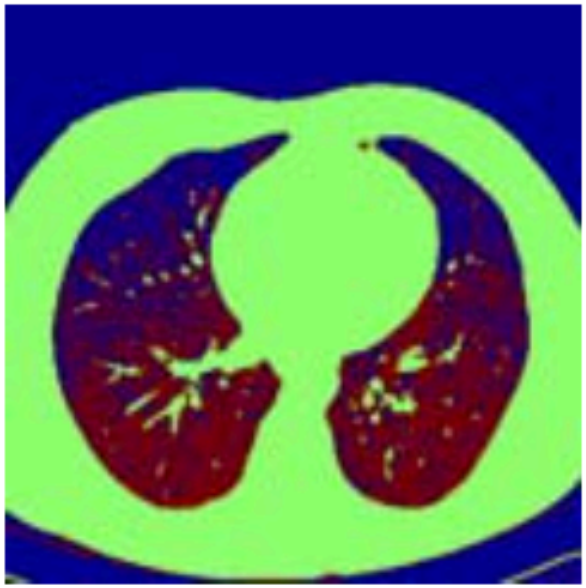
Graphs basic principles and use for feature extractions

Example cluster analysis



Examples of clustering

An image (I)



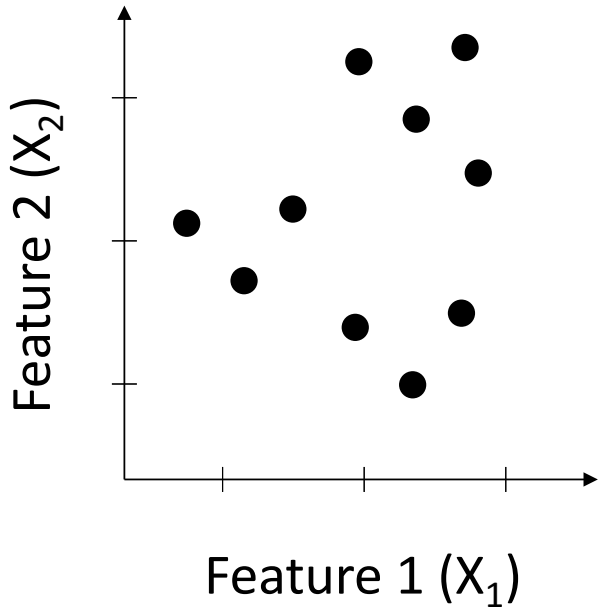
Three-cluster image (J) on gray values of I

K- means cluster algorithm

Input

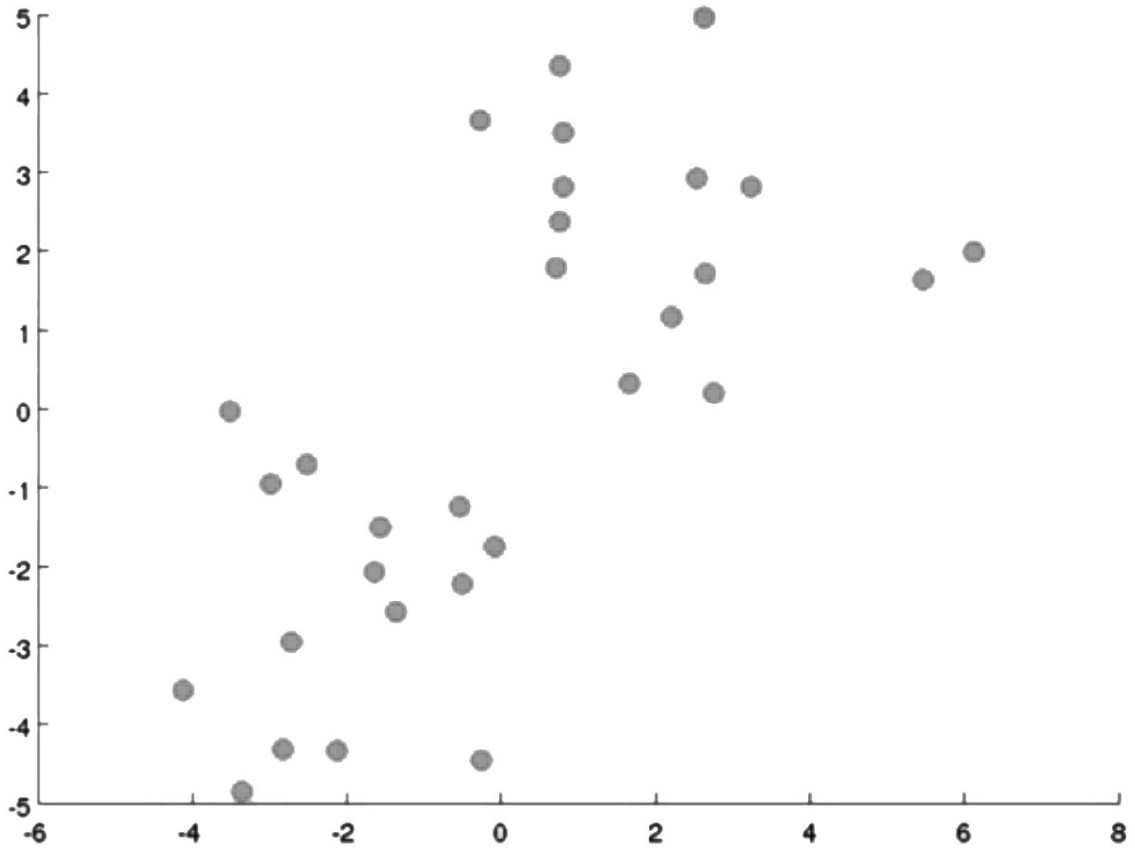
- K (number of clusters)
- Training Set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(i)}\}$

Can have multiple dimensions (R^n)

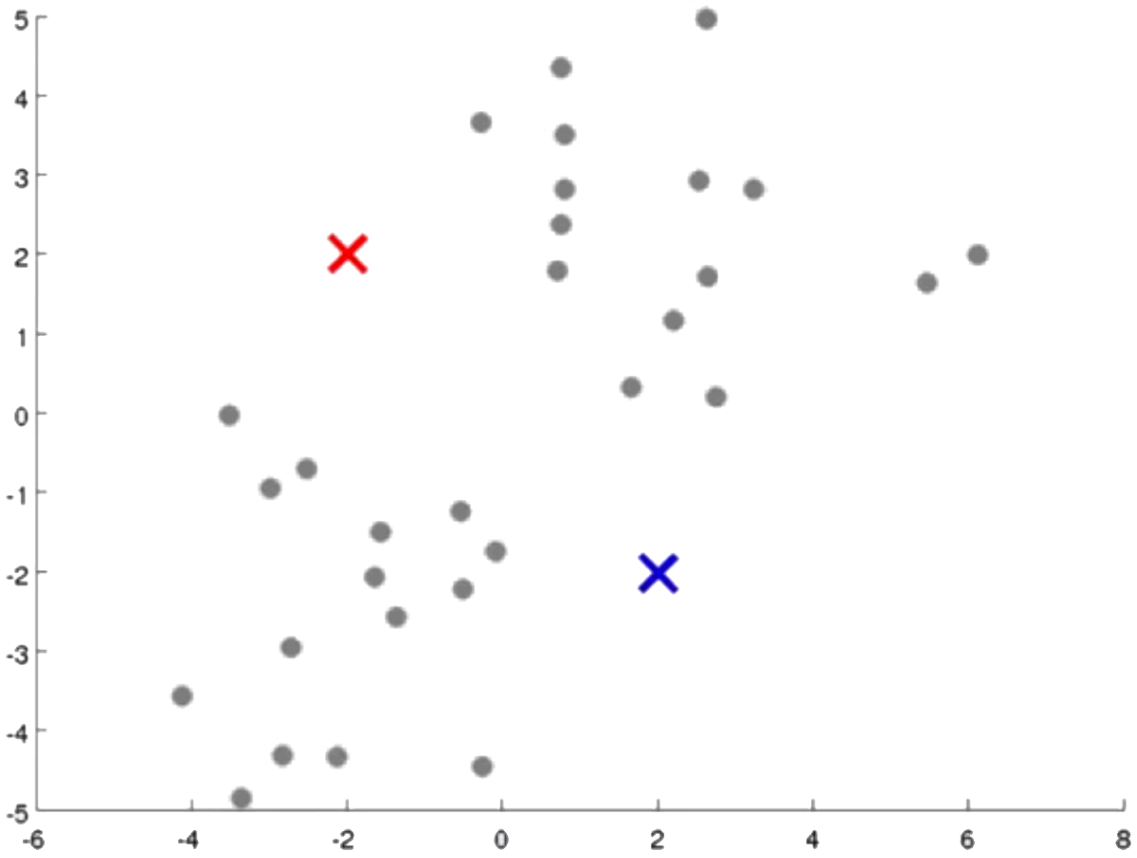


K-means is the most popular partitioning clustering method

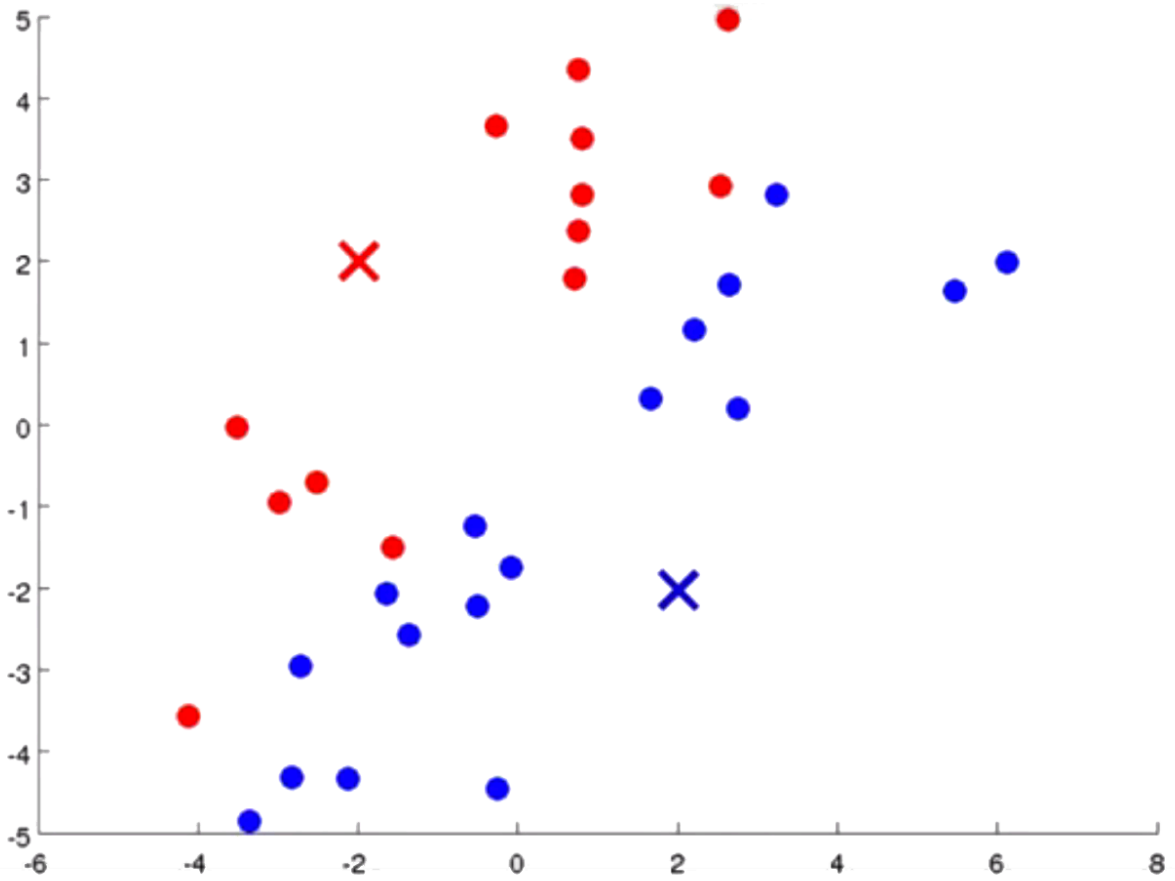
K- means cluster algorithm



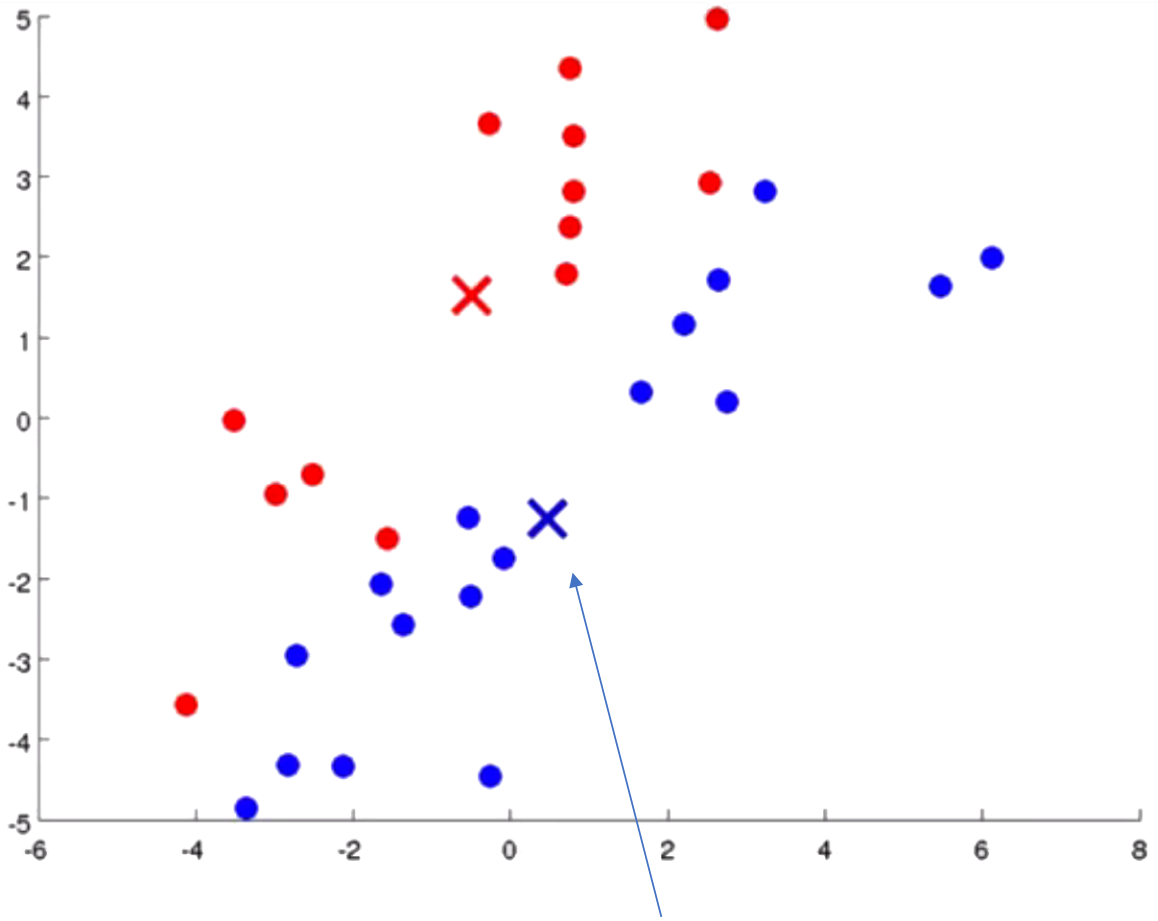
K- means cluster algorithm



K- means cluster algorithm

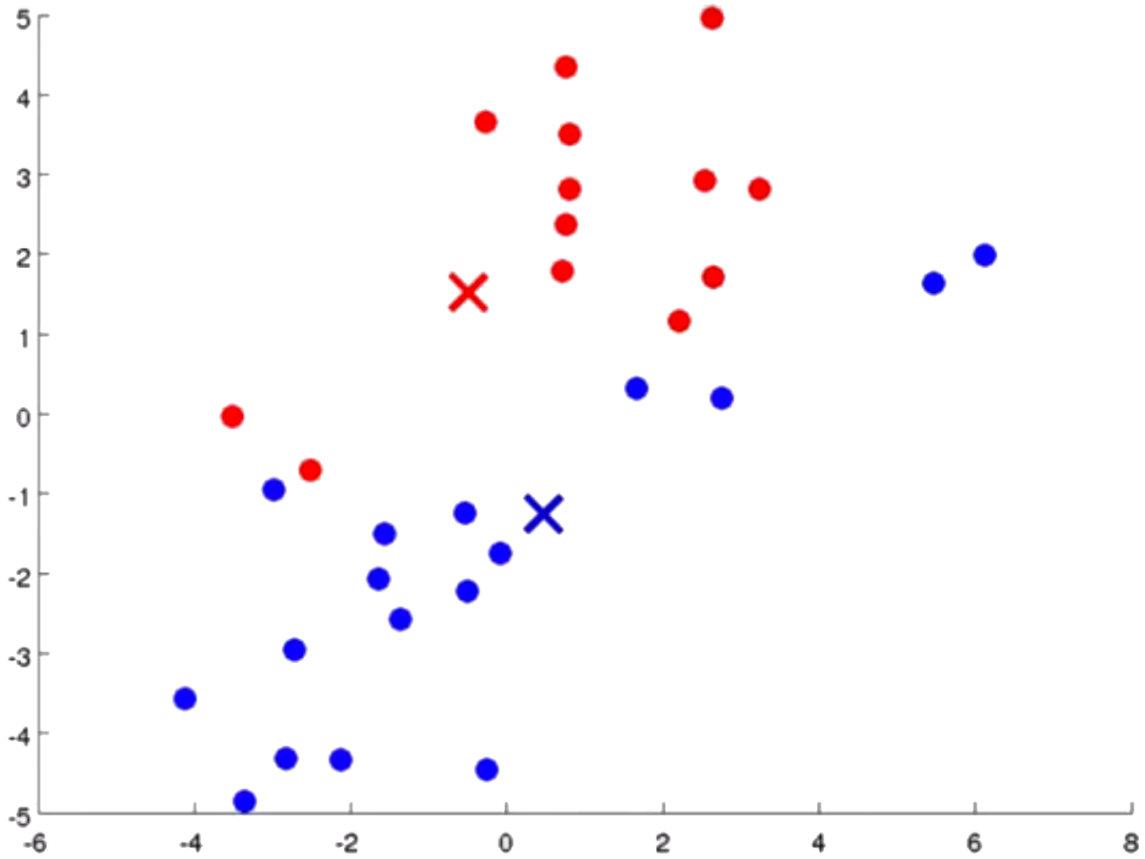


K- means cluster algorithm

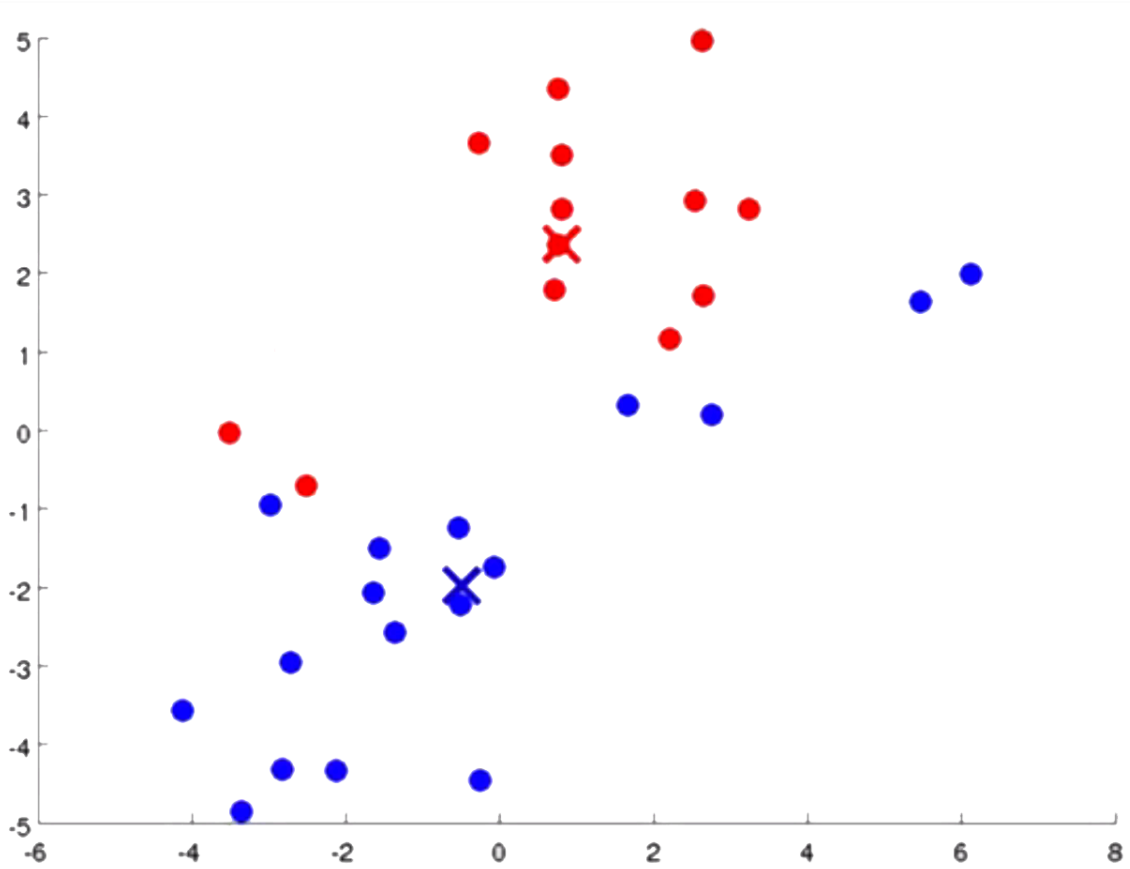


Centroid μ_c: Mean of all points in cluster C (blue)

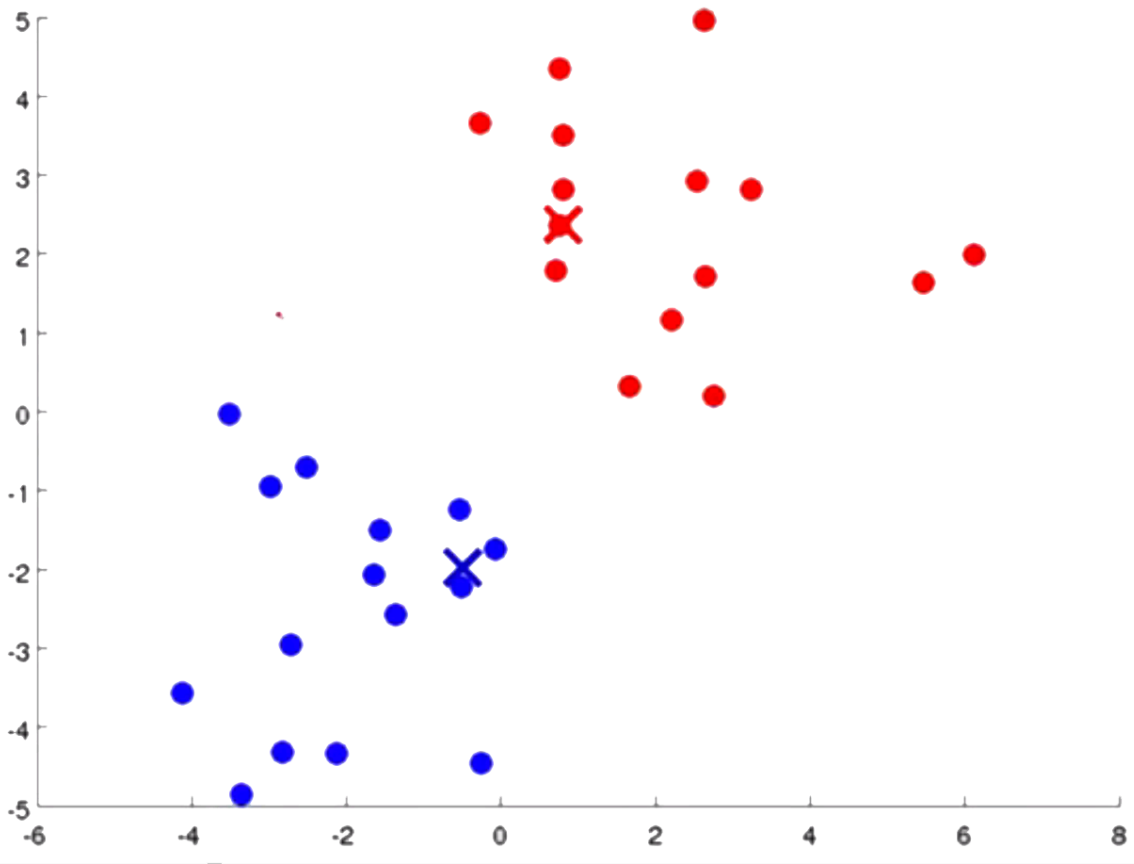
K- means cluster algorithm



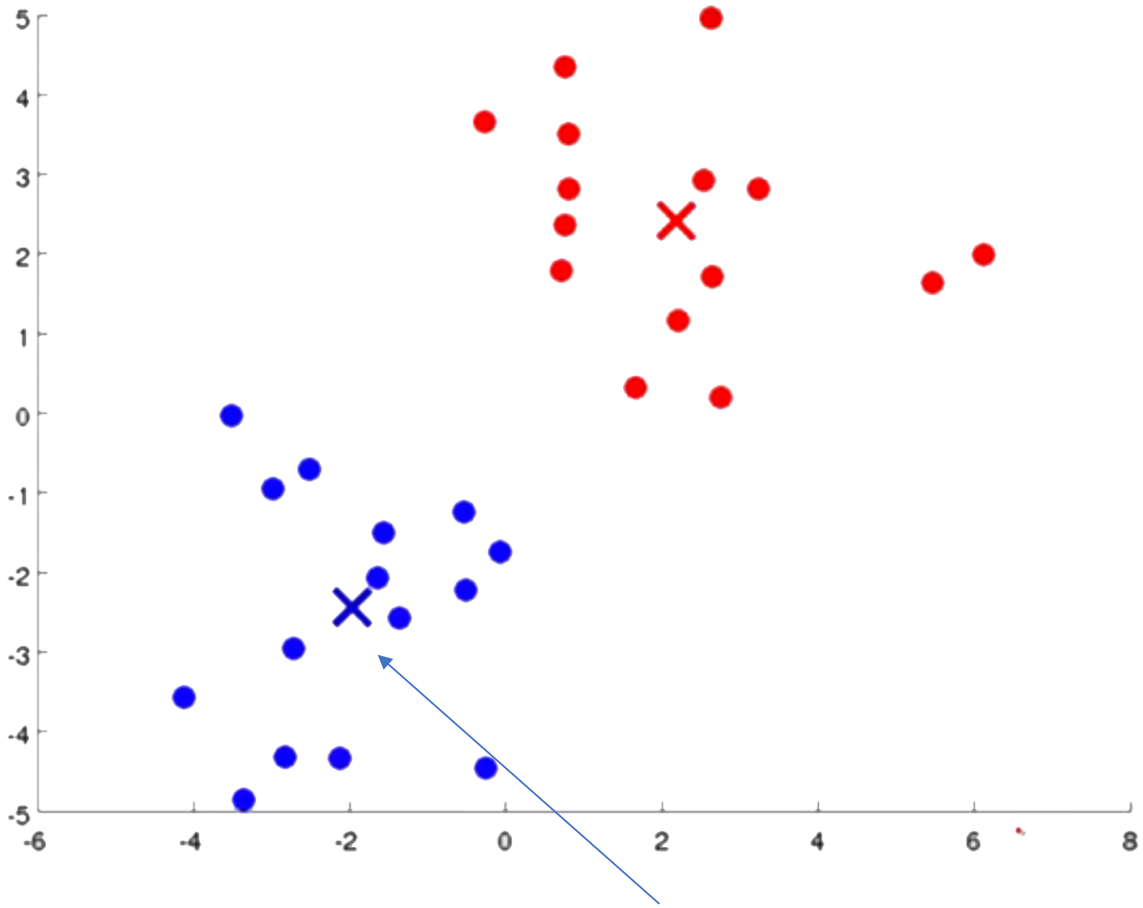
K- means cluster algorithm



K- means cluster algorithm



K- means cluster algorithm



Centroid μ_c: No longer moves (after updating)

K- means cluster algorithm

Randomly initialize *K* cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$

Repeat {

for i = 1 to m

$c^{(i)}$ ← index (from 1 to *K*) of cluster centroids

closest to $x^{(i)}$

for k = 1 to *K*

μ_k ← average (mean) of points assigned to cluster *k*

}

K- means cluster algorithm

Randomly initialize *K* cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$

Repeat {

cluster assignment step

for i = 1 to m

$c^{(i)} \leftarrow$ index (from 1 to *K*) of cluster centroids closest to $x^{(i)}$

$$c^{(i)} \leftarrow \min_k \| x^{(i)} - \mu_k \|$$

move centroid

for k = 1 to *K*

$\mu_k \leftarrow$ average (mean) of points assigned to cluster *k*

}

$x^{(1)}, x^{(3)}, x^{(4)} \rightarrow c^{(1)} = 2, c^{(3)} = 2, c^{(4)} = 2$

Example:

$$\mu_2 \leftarrow [\, x^{(1)} + x^{(3)} + x^{(4)} \,] / 3)$$

(updated centroid)

K- means cluster algorithm

Start with randomly initialize *K* cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$

Repeat {

cluster assignment step

for i = 1 to m (until m steps or centroids stop moving)

"Evaluate the distance of each point to the centroid and update the cluster assignment"

move centroid

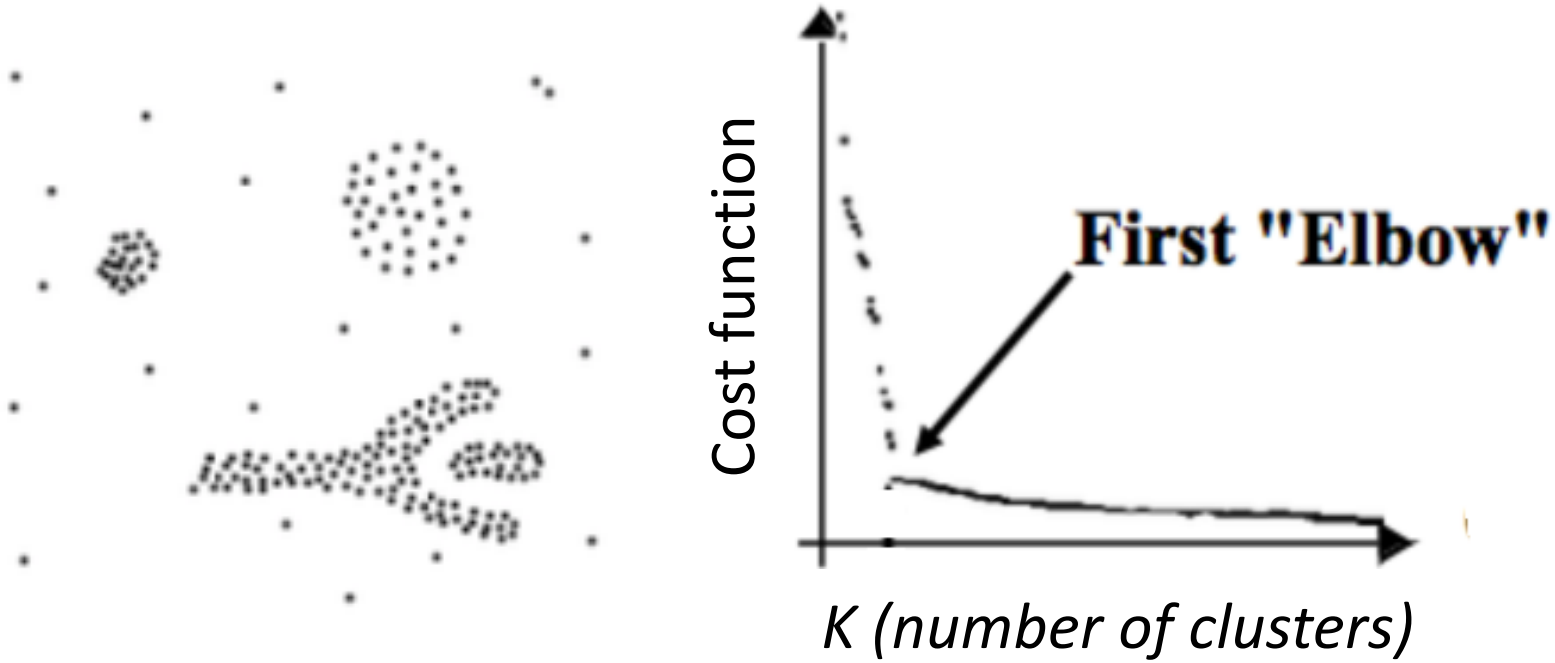
for k = 1 to *K (for each cluster)*

"calculate the centroid location and update"

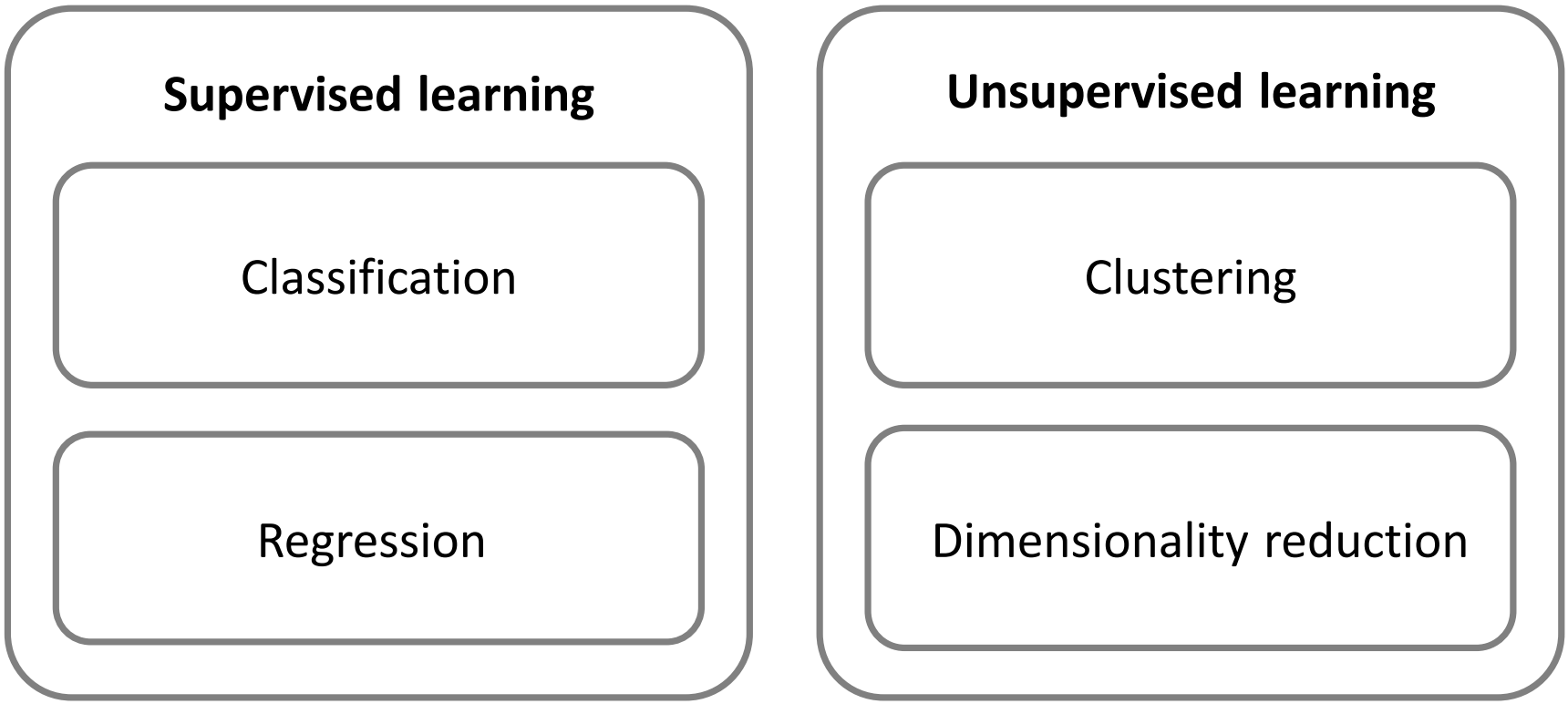
}

How do you choose the right value for K?

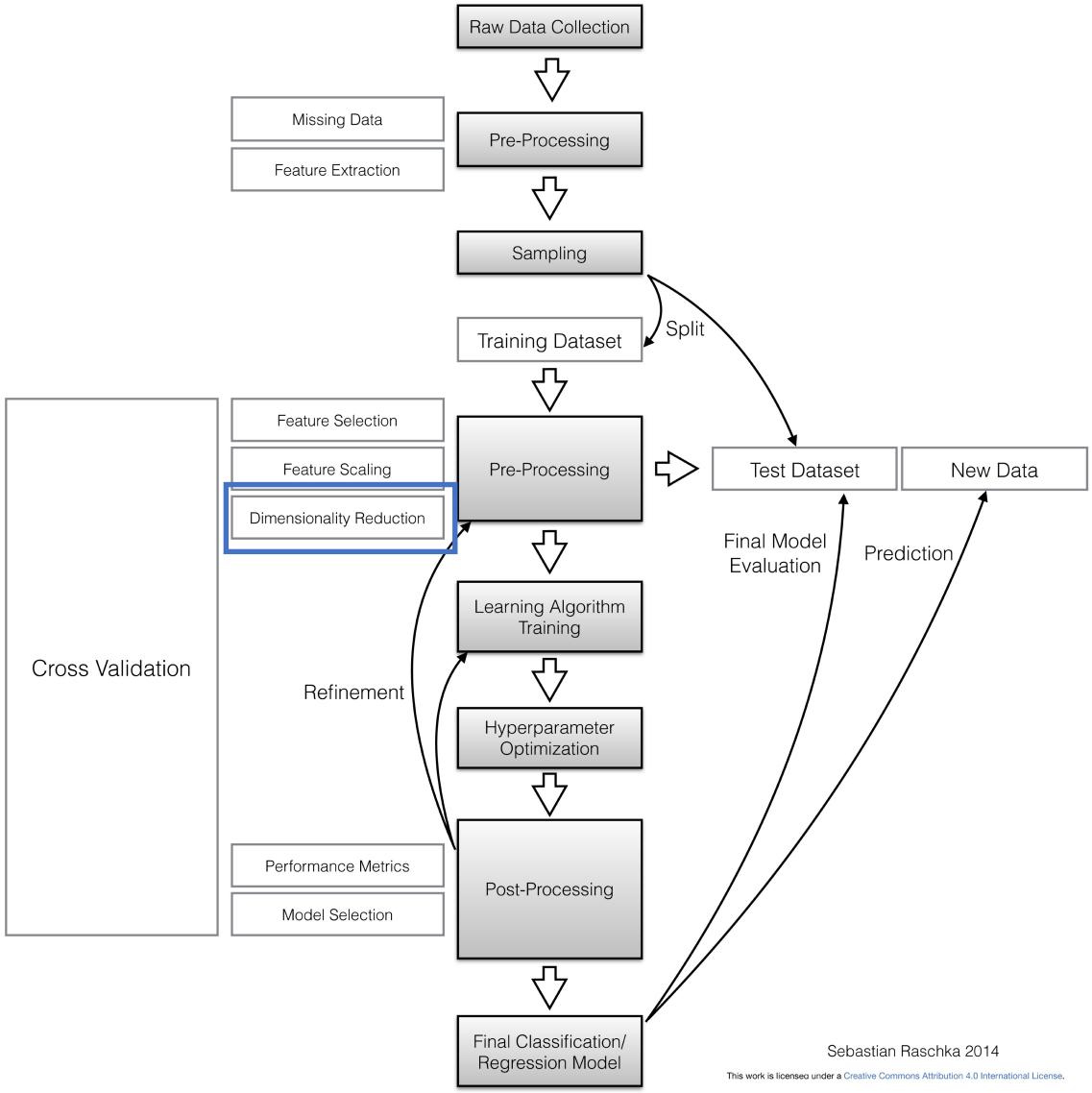
Often a challenge (hence unsupervised learning)



Supervised vs Unsupervised Learning



Typical flowchart supervised learning



Dimensionality Reduction

Why dimension reduction?

- Visualization
- The curse of dimensionality

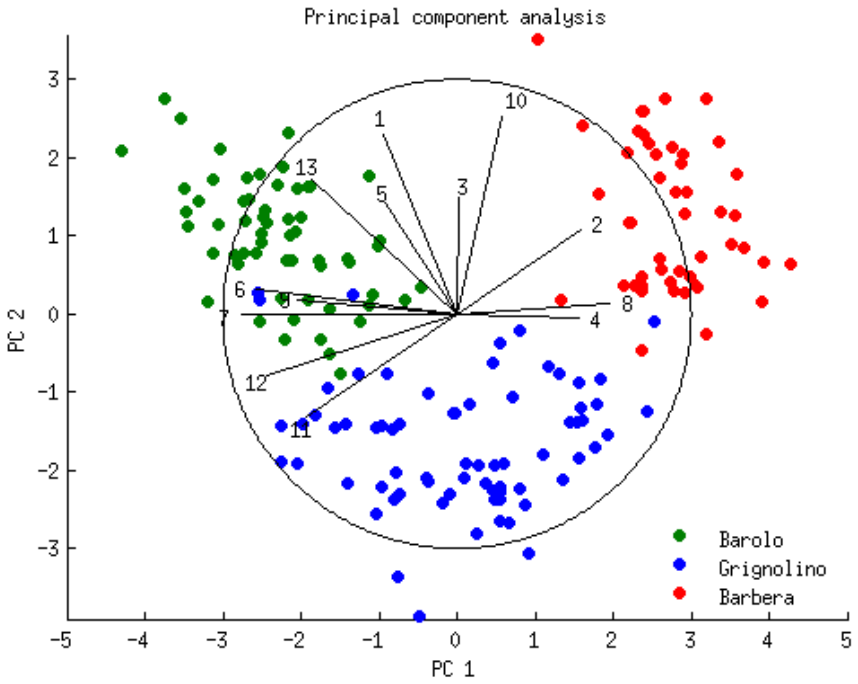
How: Feature Selection

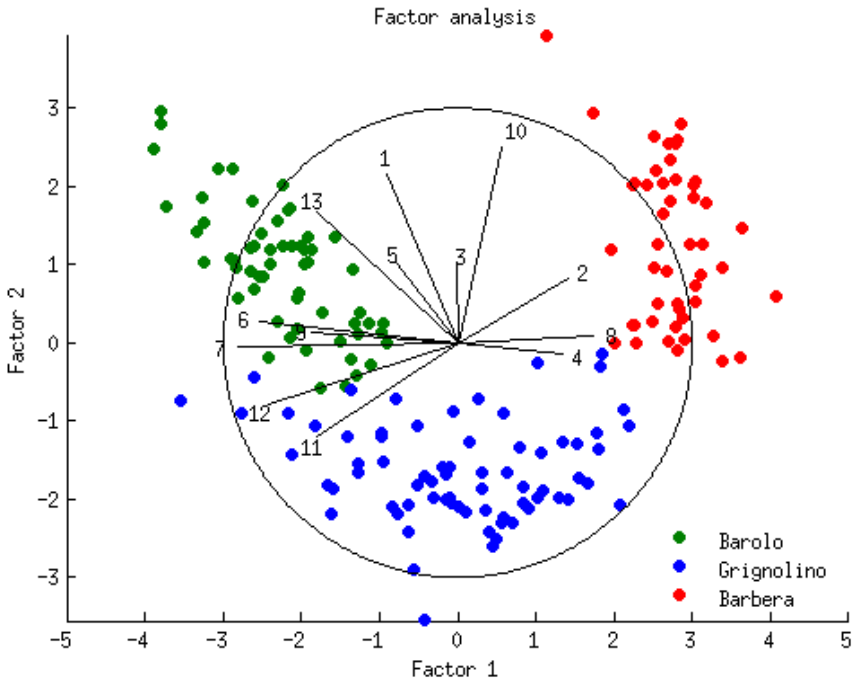
- Filtering strategy
- Wrapper strategy
- Embedding strategy

How: Feature Extraction

- Linear: PCA, Factor analysis
- Non-linear: Kernel PCA, Curves, Manifolds

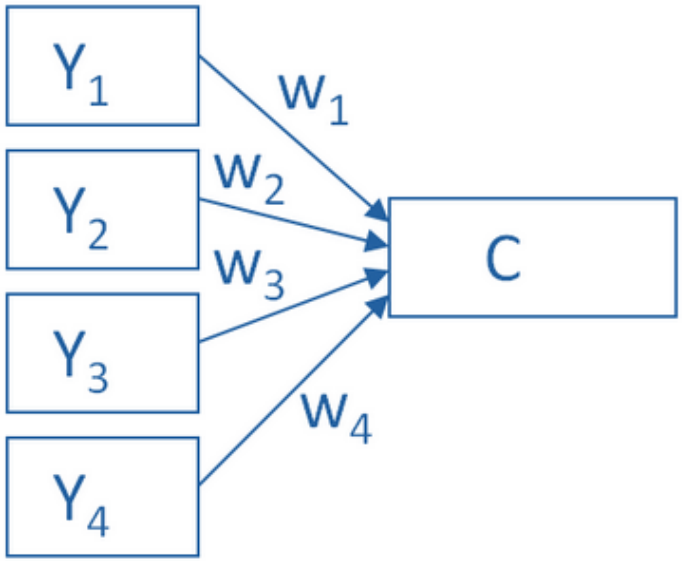
PCA vs Factor Analysis

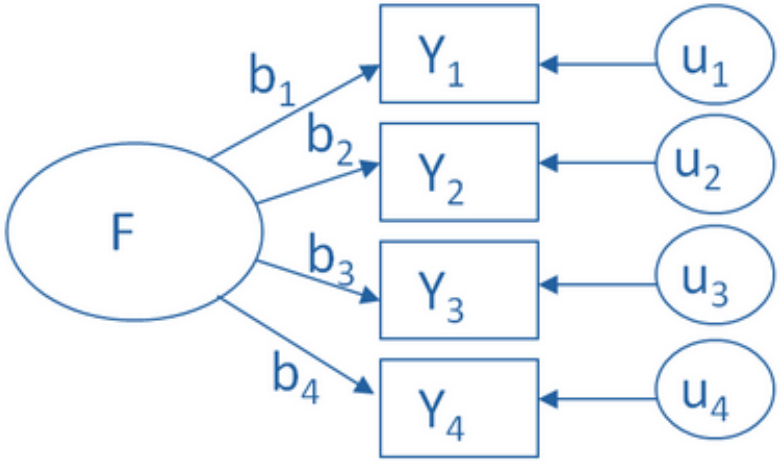




Similar but Different

PCA vs FA

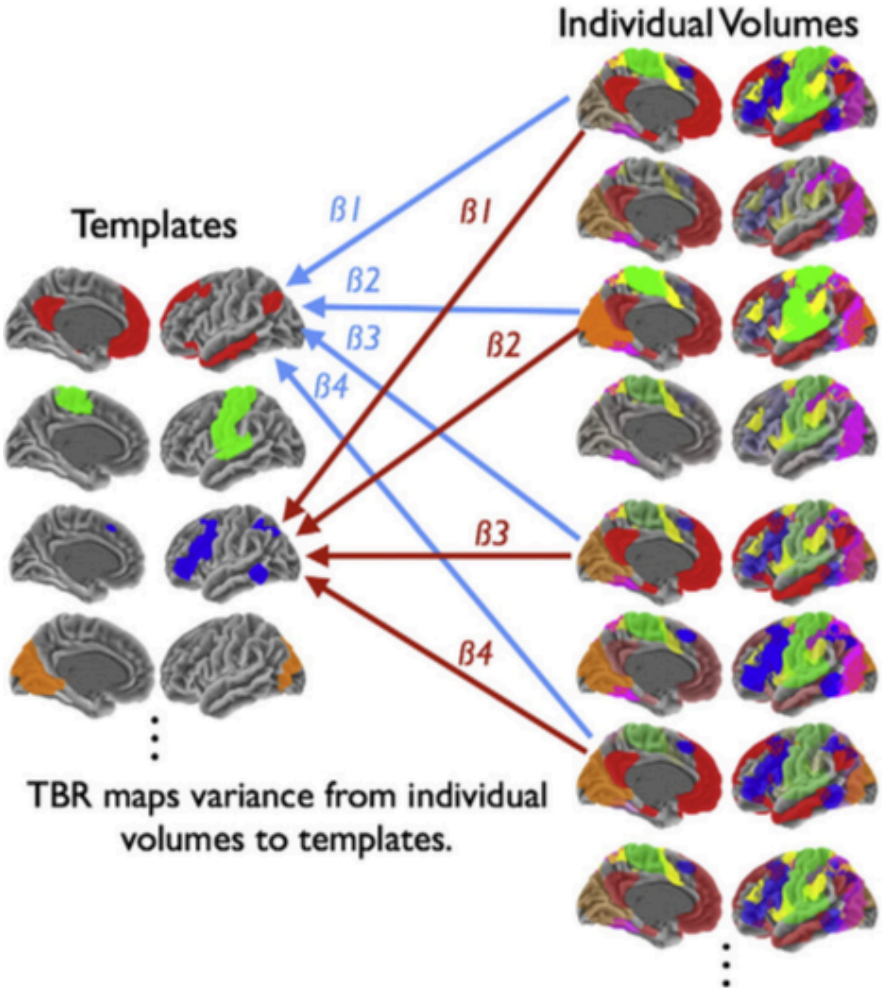


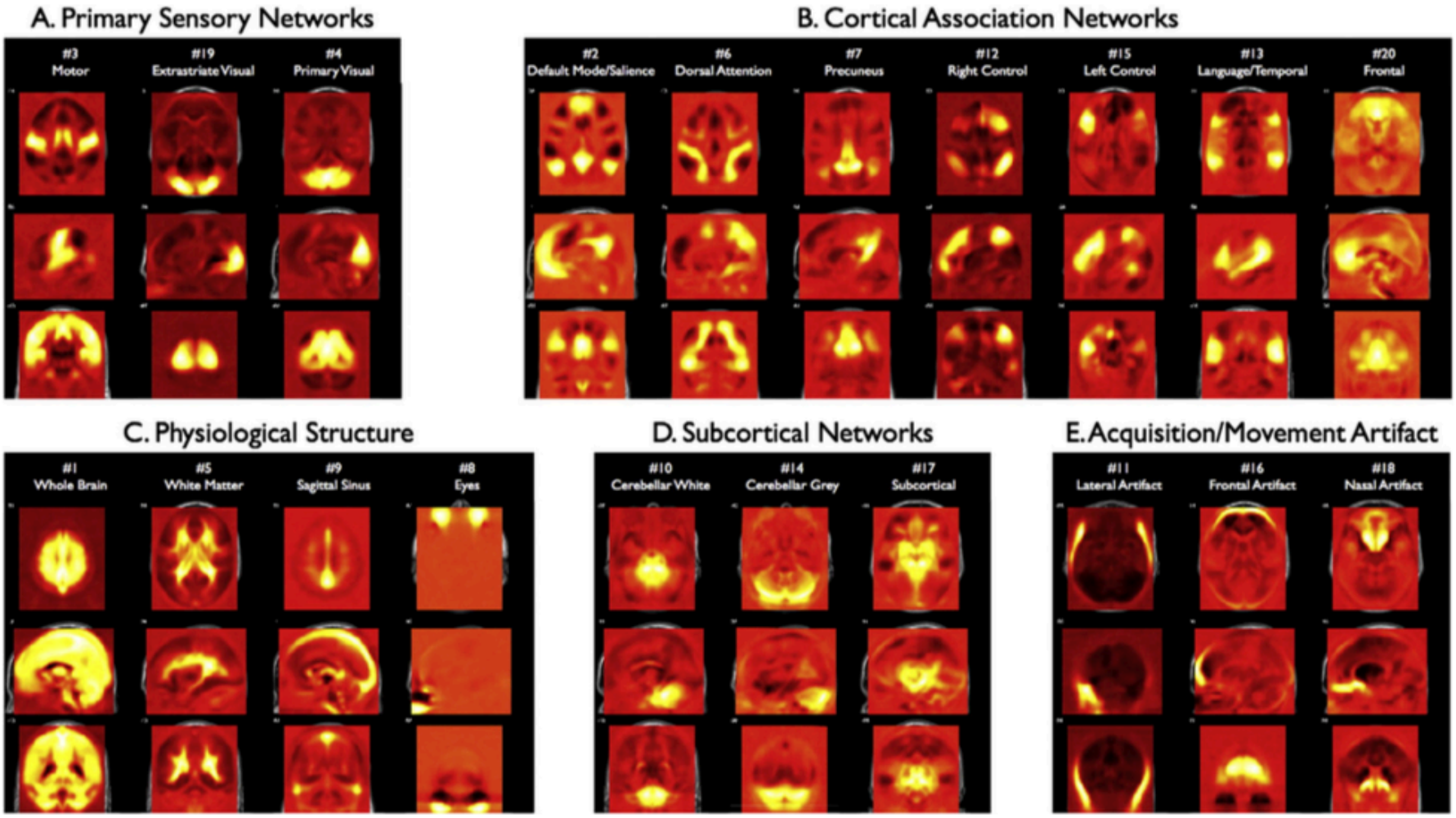


$C = w_1(Y_1) + w_2(Y_2) + w_3(Y_3) + w_4(Y_4)$

$Y_1 = b_1 * F + u_1$
$Y_2 = b_2 * F + u_2$
$Y_3 = b_3 * F + u_3$
$Y_4 = b_4 * F + u_4$

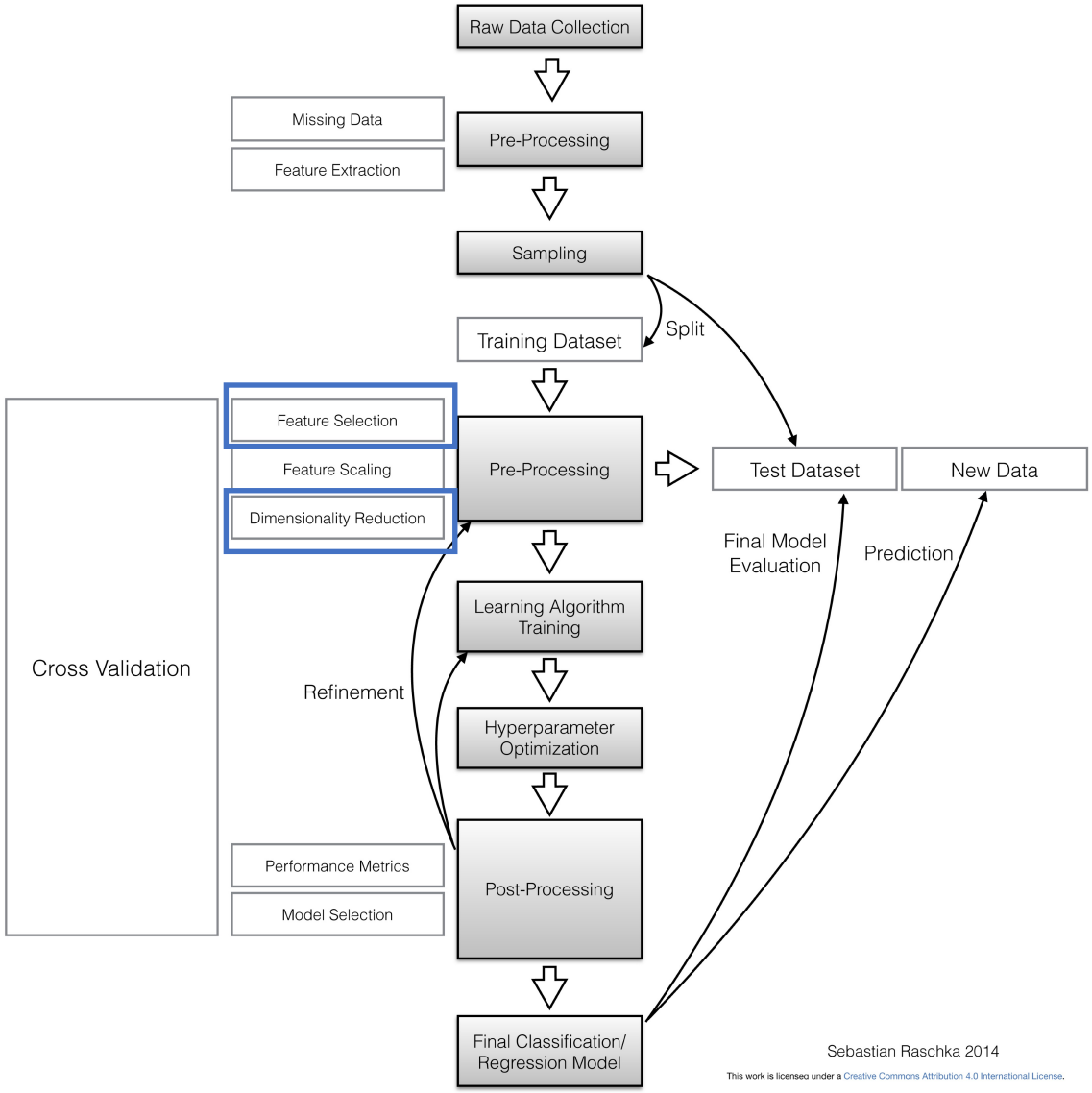
Factor Analysis on Brain data



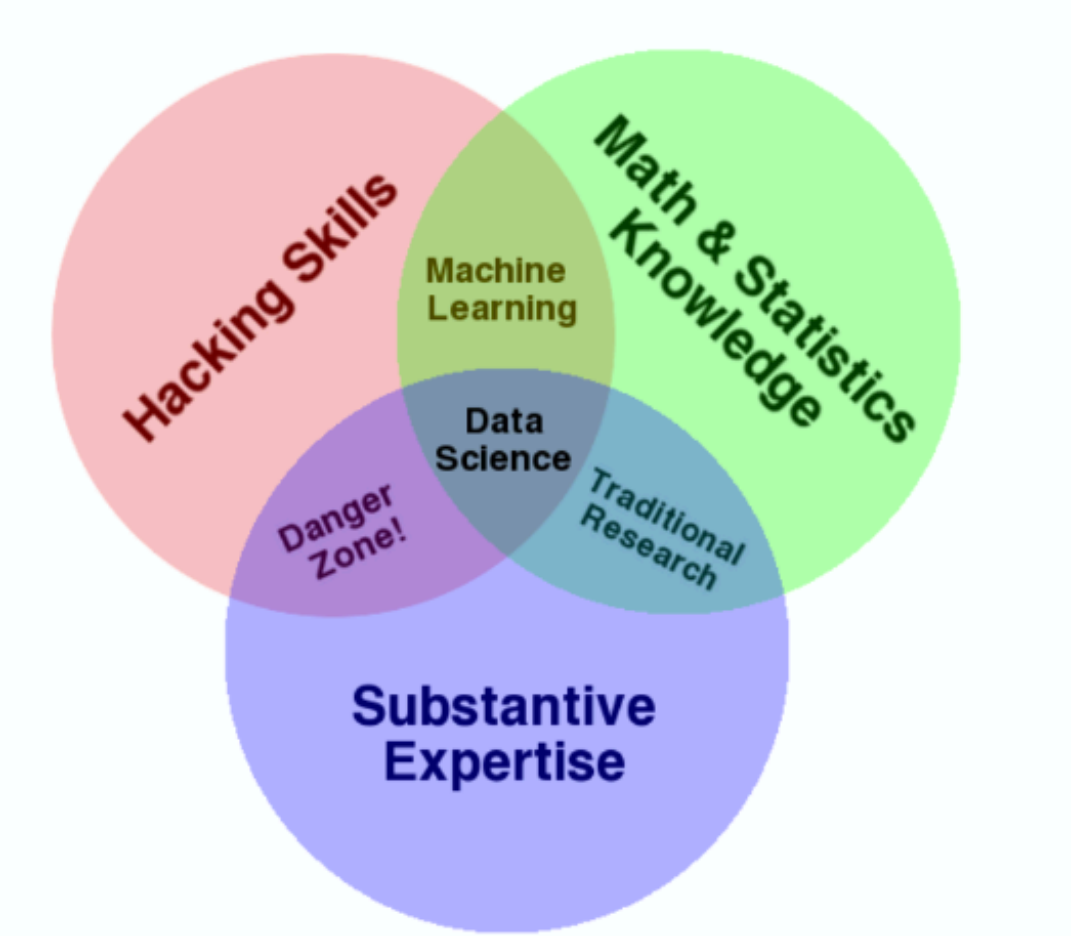


Not all Factors or All components are meaning full (noise)

Typical flowchart supervised learning



What is data science?



http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Course Schedule

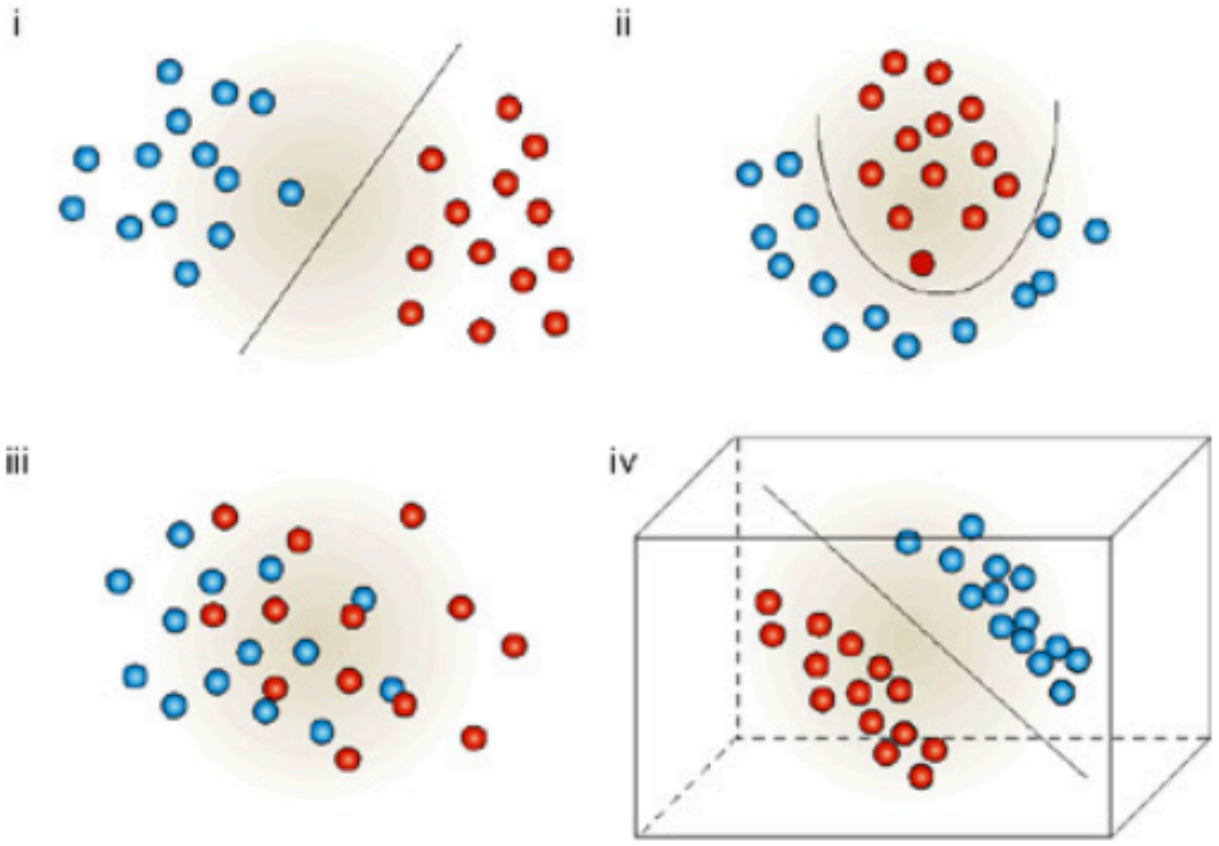
`v04.10.2017` (subject to change – always check the latest version!)

#	Date	Lectures (Theory - Willem)	Date	Video Lectures (Applications - Chris)	Practicals & Notebooks
1	29-08	Introduction to Data Mining	31-08	Introduction to Data Science	Setting up Jupyter
2	05-09	Regression	07-09	Representing Data	Raw Data to Observations
3	12-09	Classification	14-09	Working with Text Data Part 1	DIY Pandas + scikit-learn
4	19-09	Algorithm Fitting & Tuning	21-09	Best Practices, Common Pitfalls	*No practical*
5	26-09	**Midterm**	28-09		DIY Pandas + scikit-learn
6	03-10	Data Reduction & Decomposition	05-10	Working with Text Data Part 2	Preprocessing + Pipelines
7	10-10	Clustering and Graphs	12-10	Mining Massive Data	Unsupervised Learning
8	17-10	Recap Lecture	19-10	Applications of Deep Learning*	MNIST Challenge*

*Will not be exam material.

Planned bonus videos: "Data Science Research", "Explaining models, Ethics, Privacy".

Which of the examples are not Linearly Separable?



www.sli.do #DM4BG