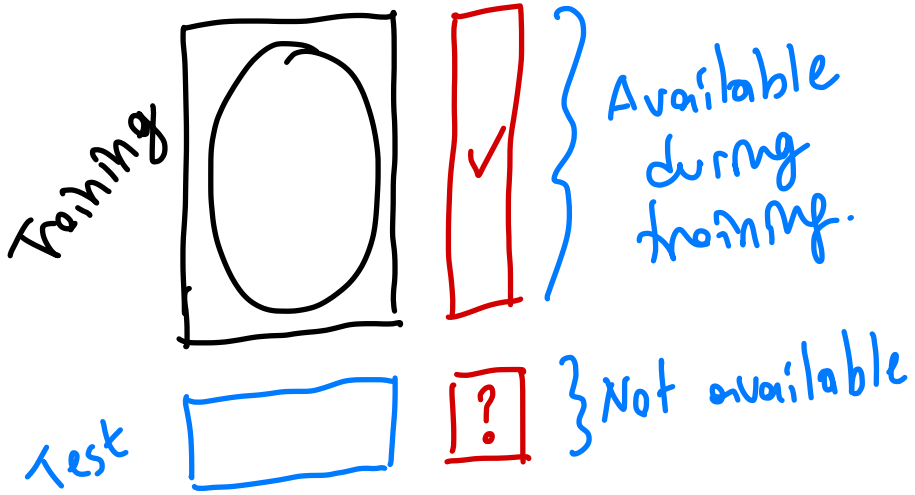
Design and Analysis of Machine Learning Experiments

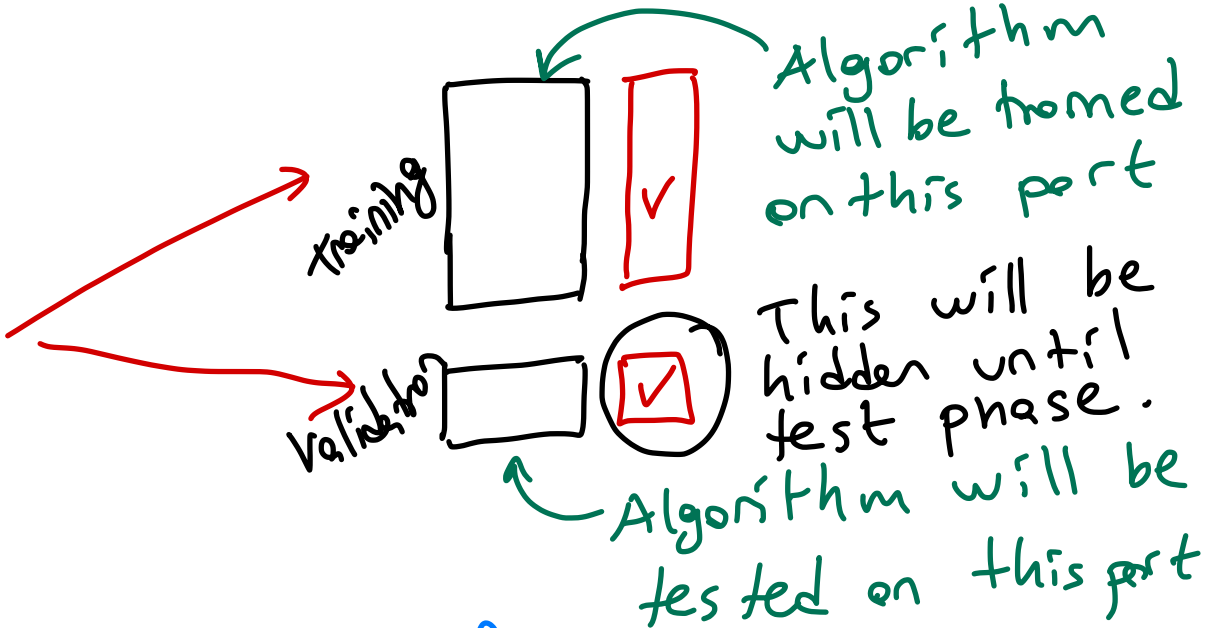
1. How can we assess the expected error of a learning algorithm for a given problem?
2. Given two or more algorithms, how can we say that one is better than the other(s) for a given problem?

WE CAN NOT USE THE TRAINING SET TO ANSWER 1 & 2.

training set error < test set error

VALIDATION SETS





"simulate the future by performing test on the validation data"

A1
A2
A3

R = # of replications

R_1	T_1	V_1
R_2	T_2	V_2
R_3	T_3	V_3
R_R	T_R	V_R

$e_{1,1}$ = misclassification error of A_1 on R_1

$e_{3,2}$ = misclassification error of A_3 on R_2

measures
- time complexity
- space complexity
- interpretability
- easy programmability
- loss functions
- ⋮

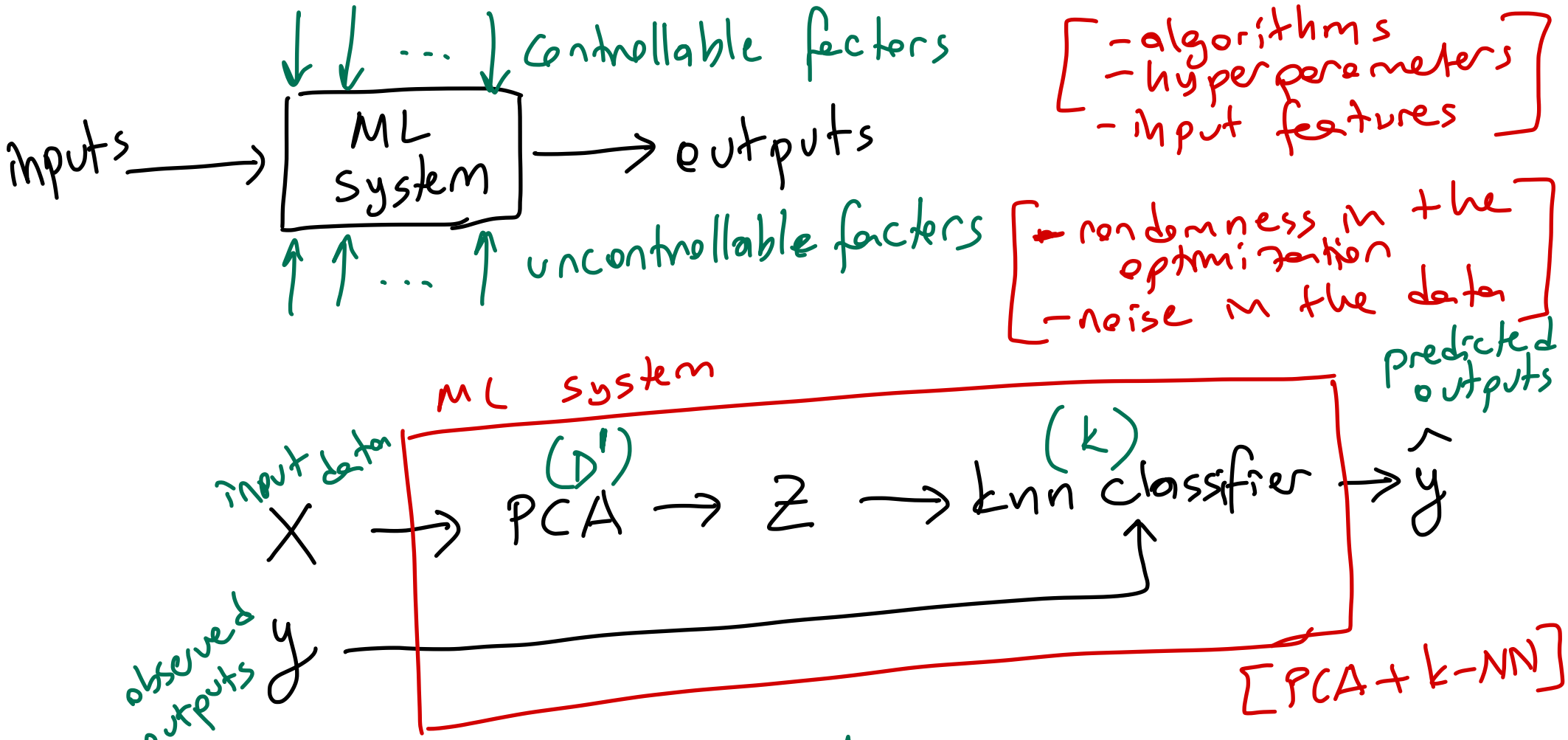
	A_1	A_2	A_3
R_1	$e_{1,1}$	$e_{2,1}$	$e_{3,1}$
R_2	$e_{1,2}$	$e_{2,2}$	$e_{3,2}$
⋮	⋮	⋮	⋮
R_R	$e_{1,R}$	$e_{2,R}$	$e_{3,R}$
	e_1	e_2	e_3

e_2 minimum

e_1 = average performance of A_1

A^* = algorithm with the minimum average error.

$A^* = A_2$

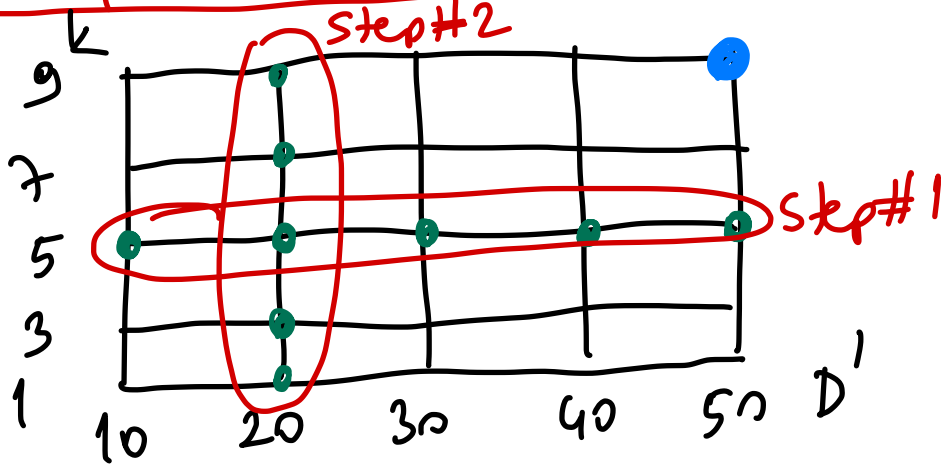


Optimization problem $\Rightarrow (D'^{*}, k^{*})$

Exhaustive Enumeration: Try all possible combinations.

! Not possible due to "computational complexity"

One factor at a time:



25 possible combinations

⇓

We tried only 9 out of 25 Combinations.

(1) Find best D' by setting k to a specified value.

Assume $k=5 \Rightarrow D' = ?$

A1 (10,5) | A2 (20,5) ★ | A3 (30,5) | A4 (40,5) | A5 (50,5)

(2) Find best k by using D' from step #1.

Assume $D'=20 \Rightarrow k = ?$

A1 (20,1) | A2 (20,3) | A3 (20,5) | A4 (20,7) ★ | A5 (20,9)

$(D'^{*}, k^{*}) \Rightarrow (20, 7)$

Guidelines for ML Experiments

1. Aim of the study
   - → evaluate a single algorithm
   - → pick the best algorithm for a specific problem
   - → pick the best algorithm for a set of problems
2. Selection of the response variable → performance criteria
3. Choice of factors and their levels
   - → algorithms
   - → hyper-parameters
   - → ...
4. Choice of experimental design
   - → ✓ exahustive enumeration
   - → factorial design
   - → ✓ one factor at a time
   - → response surface design
   - → ...

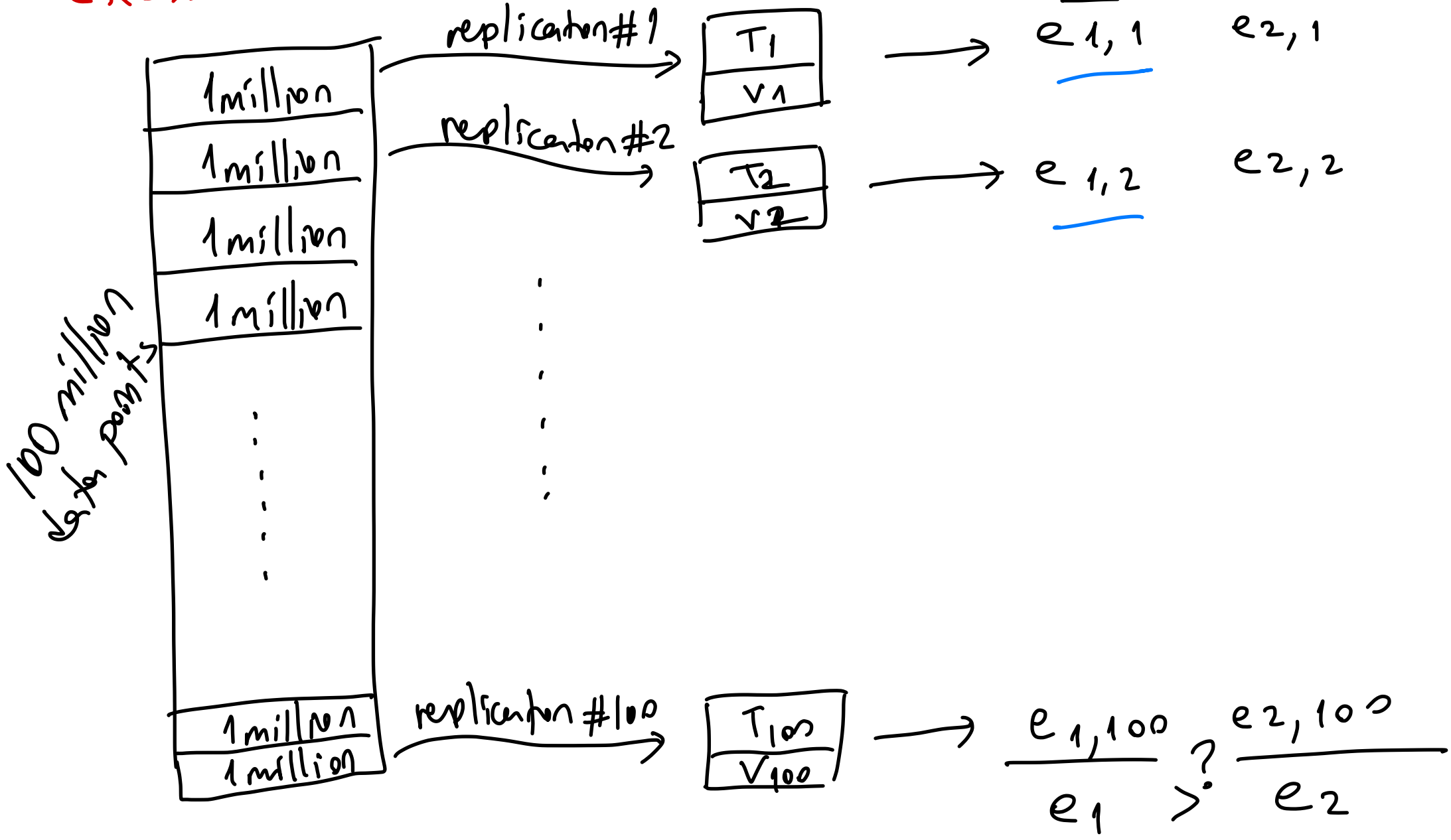
⑤ Run experiments → use [parallel / cloud] computing if possible

⑥ Statistical analysis of the results → Alg 1 > Alg 2 ?

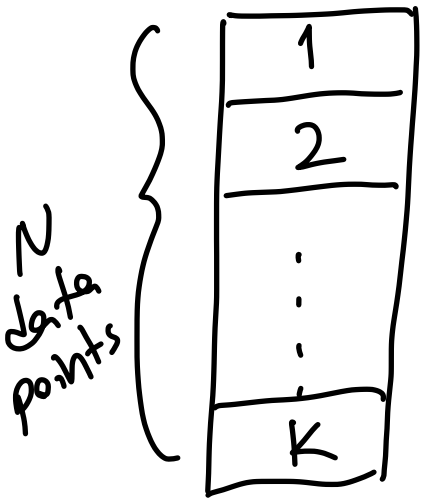
- confidence interval
- hypothesis testing.
- ...

⑦ Conclusions & Recommendations

CROSS-VALIDATION & RESAMPLING



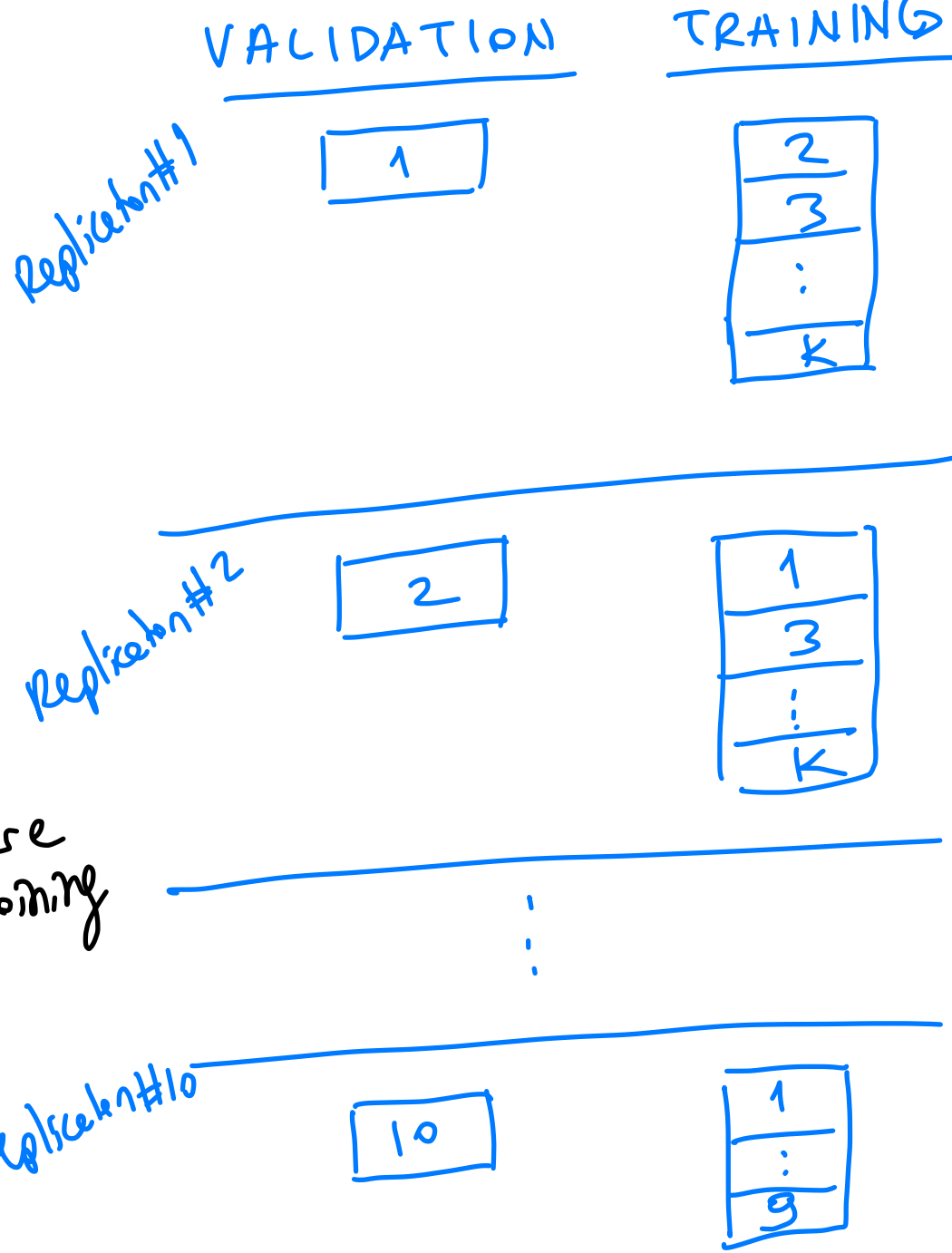
K-FOLD CROSS-VALIDATION



almost equal "K" partitions

* No overlap in validation!
* (K-2) out of (K-1) blocks are common in training

Overlap between training data is quite large



Step 1: Shuffle data points.

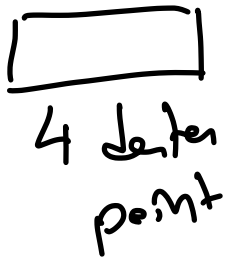
Step 2: Divide the data set into K (almost) equal sized partitions

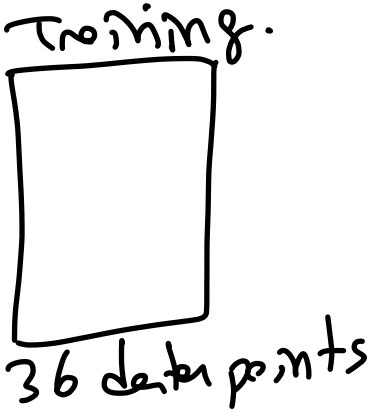
Step 3: Use each partition as the validation data set in one replication

LEAVE-ONE-OUT CROSS-VALIDATION

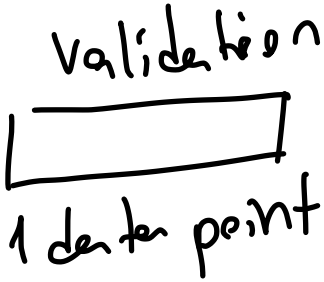
((K-FOLD Cross validation where K=N)

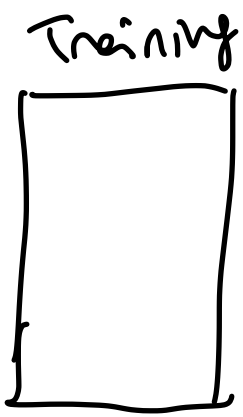
K=10



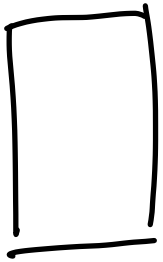


(LOO)
K=40



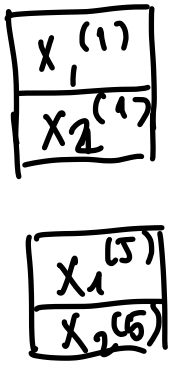


5X2 CROSS-VALIDATION



randomly shuffle

randomly shuffle



Training	Validation
$x_1^{(1)}$	$x_2^{(1)}$
$x_2^{(1)}$	$x_1^{(1)}$
⋮	⋮
$x_1^{(5)}$	$x_2^{(5)}$
$x_2^{(5)}$	$x_1^{(5)}$