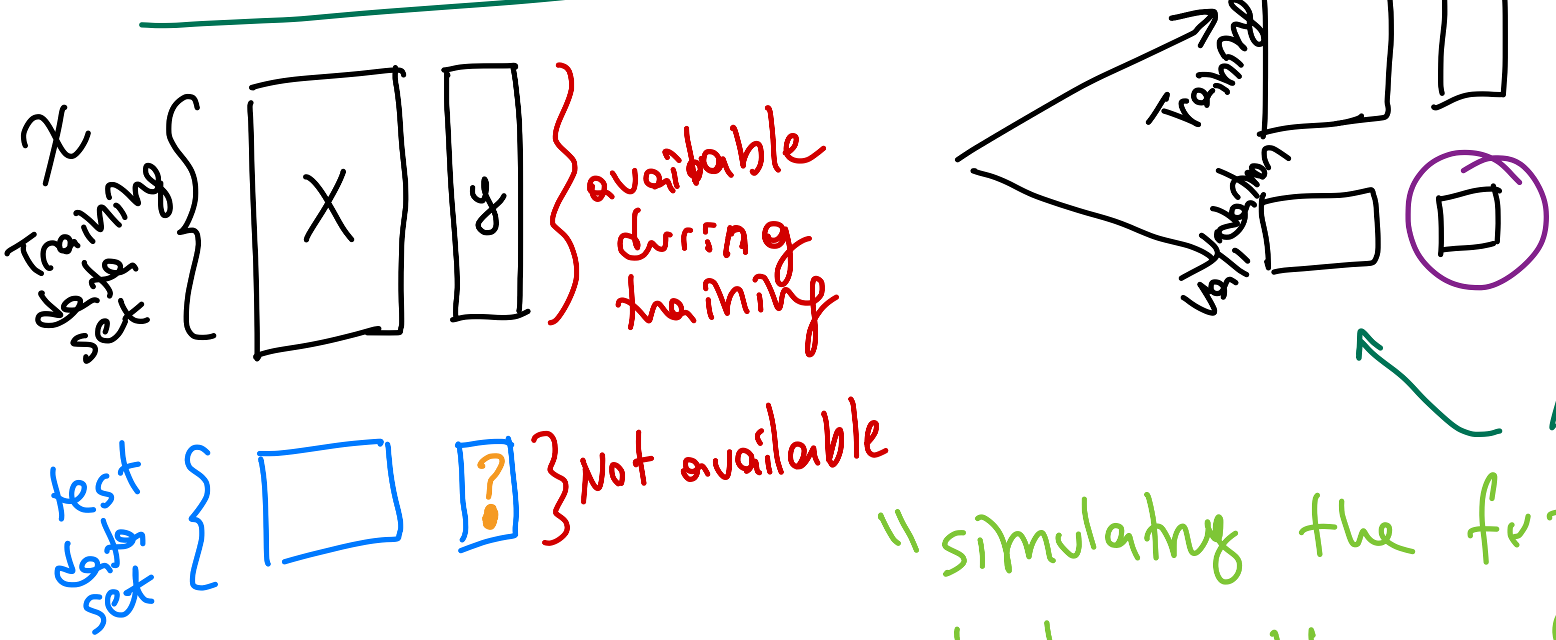
Design and Analysis of ML Experiments

1. How can we assess the expected performance 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 CANNOT USE THE TRAINING SET TO ANSWER 1 & 2.

Usually $\Rightarrow$ training set error < test set error

VALIDATION SETS



Algorithm will be trained on this part

This will be hidden until test phase.

Algorithm will be tested on this part

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

A1

A2

A3

train

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

test

R = # of replications

- time complexity
- space complexity
- interpretability
- easy programmability
- ⋮

$e_{2,2}$ = misclassification error of A_2 on R_2

$e_{3,1}$ = misclassification error of A_3 on R_1

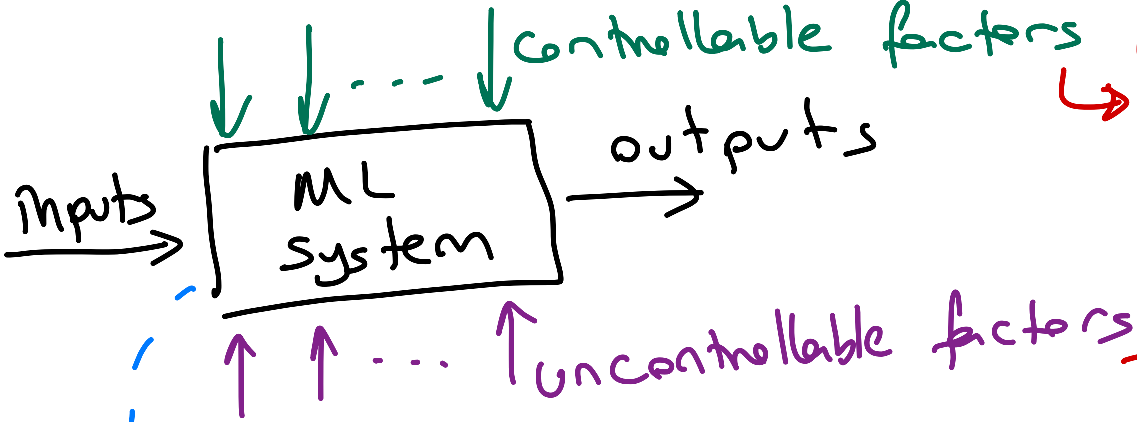
	A_1	A_2	A_3
R_1	$e_{1,1}$	$e_{2,1}$	$e_{3,1}$
R_2	$e_{1,2}$	$e_{2,2}$	$e_{3,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
R_R	$e_{1,R}$	$e_{2,R}$	$e_{3,R}$
	e_1	e_2	e_3

e_1 = average performance of A_1

A^* = algorithm with the best average performance.

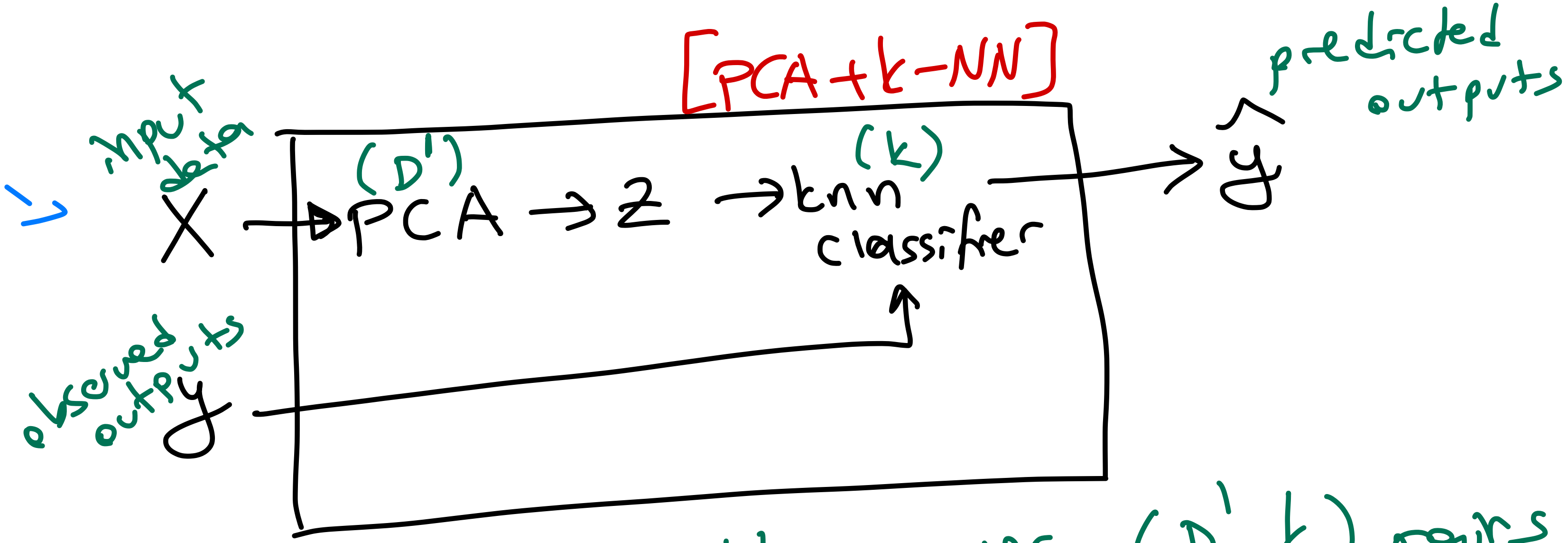
minimum

$A^* = A_2$



- algorithm
- hyperperameters
- input features

- randomness in the optimization
- noise in the data

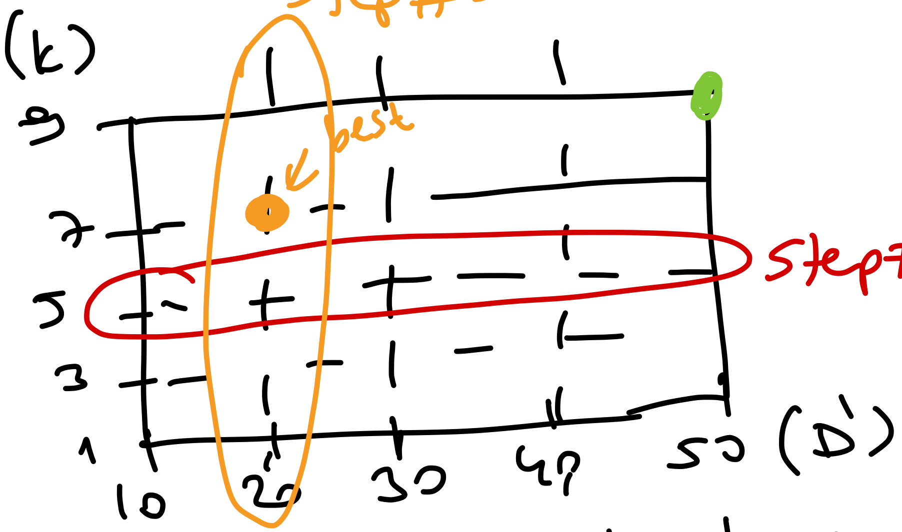


optimization problem over (D', k) pairs

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 A2 A3 A4 A5

$(D',k) \Rightarrow (10,5)\ \boxed{(20,5)}\ (30,5)\ (40,5)\ (50,5)$

A^*

$A2 \equiv A8$

② Find best k by using D' from Step#1

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

A6 A7 A8 A9 A10

$(D',k) \Rightarrow (20,1)\ (20,3)\ (20,5)\ \boxed{(20,7)}\ (20,9)$

A^*

⇓

$$(D^*, k^*) = (20,7)$$

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
   → hyperparameters
   → ...

4. Choice of experimental design
   → exhaustive enumeration
   → factorial design
   → one factor at time.
   → response surface design
   → ...

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

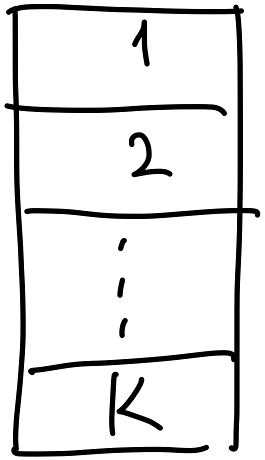
6. statistical analysis of the results → Alg 1 > Alg 2 ?

→ confidence interval

→ hypothesis testing.

7. Conclusions & Recommendations

K-FOLD CROSS-VALIDATION



almost equal-sized "K" partitions

	Val	Tra
Replication 1	1	2, 3, ⋮, K
⋮		
Replication K	K	~~1~~, 2, ⋮, K-1

overlap ratio $\frac{(K-2)}{(K-1)}$