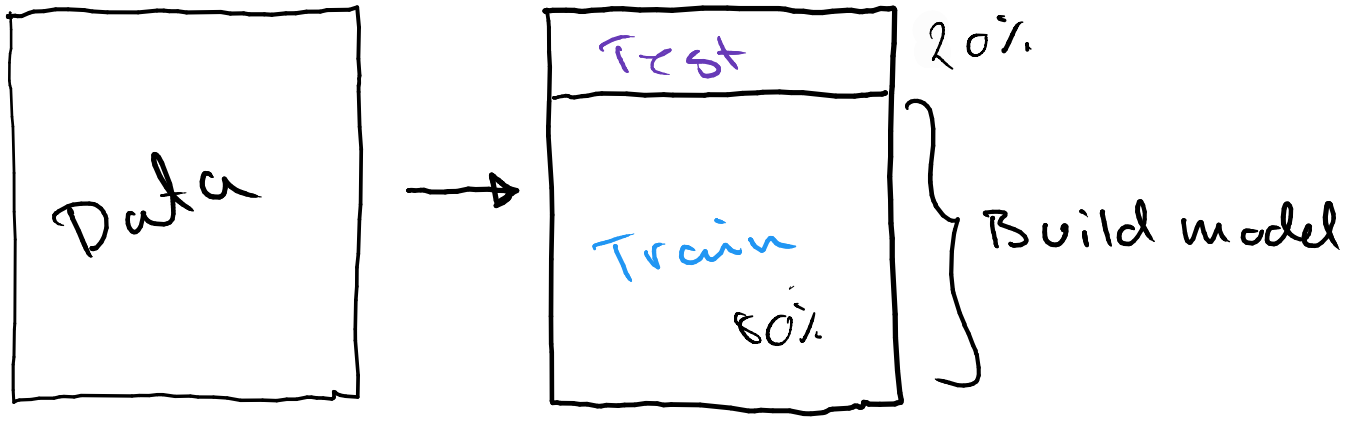
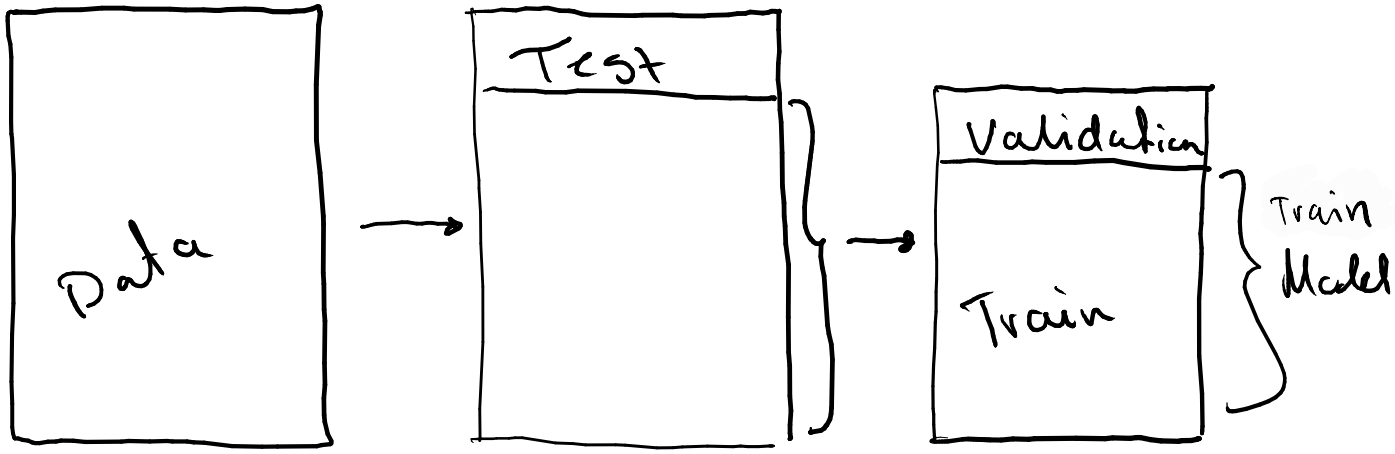
Methodology:

1. Simple train-test:

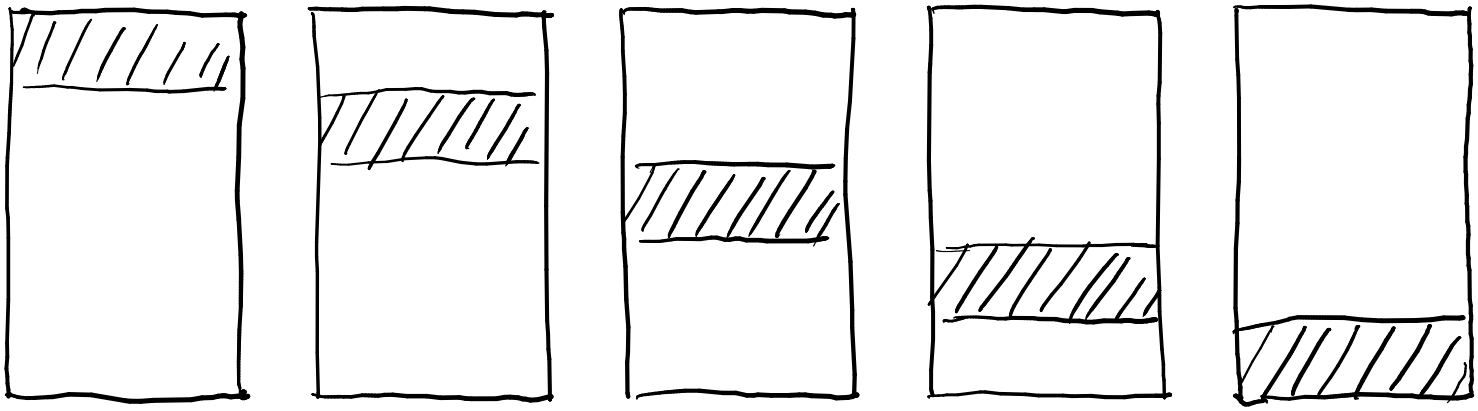


2. Validation:



* 60 - 20 - 20
* 64 - 16 - 20
* Train + Validation used for final model.

3. Cross Validation:



* split data into folds (e.g 5, 7, 10)
* train on larger set, test on small
* Accuracy = avg. of all models

Depends on size of data-set:
1) Could do train-test split and then do C.V. on train
2) Use all data for C.V

4. Stratified C.V. and validation:

* Items in folds/validation set are chosen such that distribution of population is retained
  - e.g. 80% male, 20% female

5. leave P-out C.V.

* General structure of C.V.
* C.V above is a special instance of leave-P-out C.V.

→ Leave p observations as the validation set and use the remaining (n-p) as train

→ Repeat all possible combinations of n and p:

$$C_p^n = \frac{n!}{(n-p)! \cdot p!}$$

e.g. $n = 100, p = 30$

$$C_{30}^{100} = \frac{100!}{70! \cdot 30!} \approx 3 \cdot 10^{25}$$

Special case: Leave-one-out C.V:

$$p = 1: \quad C_1^n = \frac{n!}{(n-1)! \, 1!} = n$$

Overfitting Revisited:

$S_{Train} \gg S_{Test}$ → overfitting → Low external validity

How to prevent:

→ Cross validation
→ More data
→ Feature engineering
→ Regularisation (next week)

Jupyter: Validation

Performance:

How do we assess whether our models are good?

i) Supervised vs. Unsupervised

ii) Classification vs. regression

Today

Next week

supervised: True label is known

↓

predicted vs. True

Wikipedia

A)

Predicted Label \ Actual Label	1	0
1	TP	FP
0	FN	TN

B)

Predicted Label \ Actual Label	0	1
0	TN	FN
1	FP	TP

C)

Actual Label \ Predicted Label	1	0
1	TP	FN
0	FP	TN

D)

Actual Label \ Predicted Label	0	1
0	TN	FP
1	FN	TP

Python!

What is the default output of confusion_matrix from sklearn? Image by Author

TP: True Positive

TN: True Negative

F.P: False Positive (Type I error, α)

F.N: False Negative (Type II error, β)

Accuracy:

TN	FP (Type I, α)
FN (Type II, β)	TP

$$\text{Accuracy} = \frac{TP + TN}{TN + FP + FN + TP}$$

* Most used metric
* Ratio of correct predictions

Precision:

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

* ratio of relevant instances among the retrieved instances:

  Example: 12 dogs, 10 cats

  ML model: 8 dogs < 5 dogs, 3 cats

  out of all items predicted "Dog" how many were right: $\frac{5}{8}$

* Often used when we are interested in false positives.

* Positive Predictive Rate/Value PPR/PPV

* Has to do with quality

Recall:

TN	Type I FP α
Type II FN β	TP

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

* True Positive Rate (TPR)
  - Hitrate
  - Sensitivity
  - Power

Example: 12 dogs, 10 cats

ML model: 8 dogs < 5 dogs, 3 cats

"out of all the dogs, how many were right" $\frac{5}{12}$

* often used when we are interested in false negatives.
* Has to do with quantity

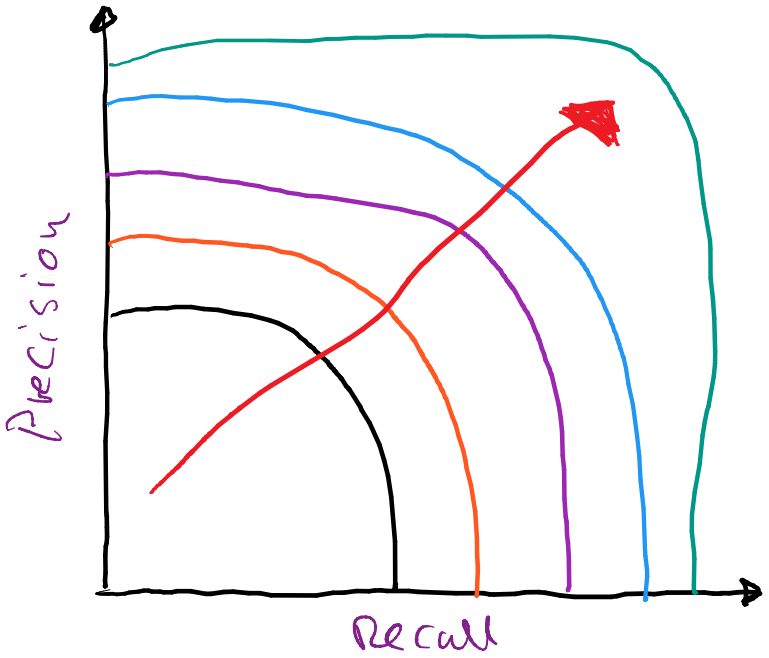
Trade-off:

$$\frac{TP}{TP+FP} \quad \text{vs.} \quad \frac{TP}{TP+FN}$$

F-score:

$$F = 2 \cdot \frac{\text{Prec} \times \text{Recall}}{\text{Prec} + \text{Recall}}$$

Precision-Recall curve:



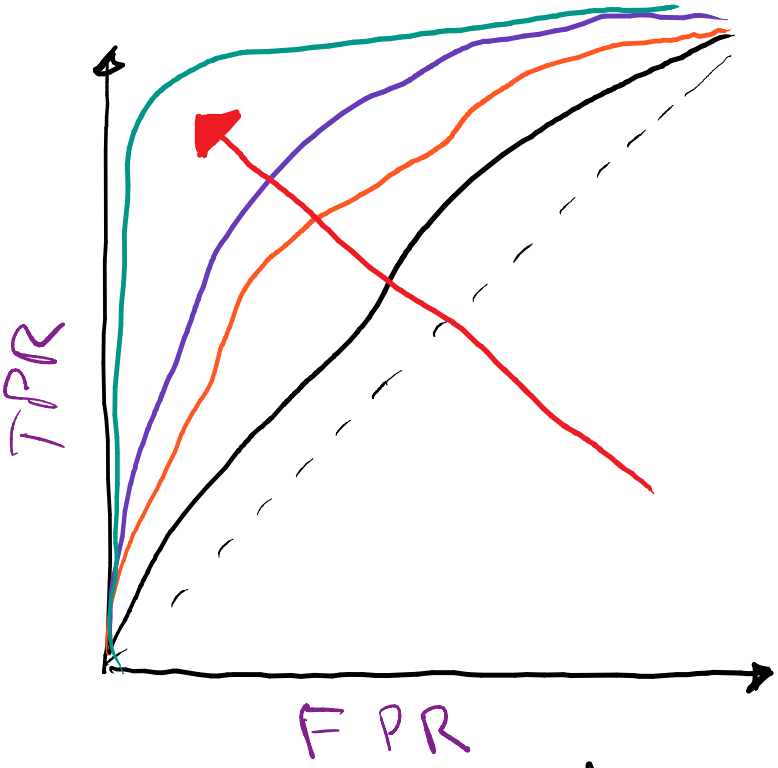
AUC: Area under the curve

Max = 1

RoC-Curve:

Recall (TPR) vs. FPR = $\frac{FP}{FN+TN}$

(TP vs. FP)



Many other metrics. Just go to Wikipedia!