Lecture 6 : More Evaluation

Two reasons to evaluate :

① get a measure of gen. perf.

② model selection :

e.g. choosing a tuning parameter

↳ k in kNN

↳ which covariates to include

Q: Can we use a test/train split (or X-validation) to do ② ?

A: Yes, but we need to be careful.

Proper way to do this: split data into 3 datasets

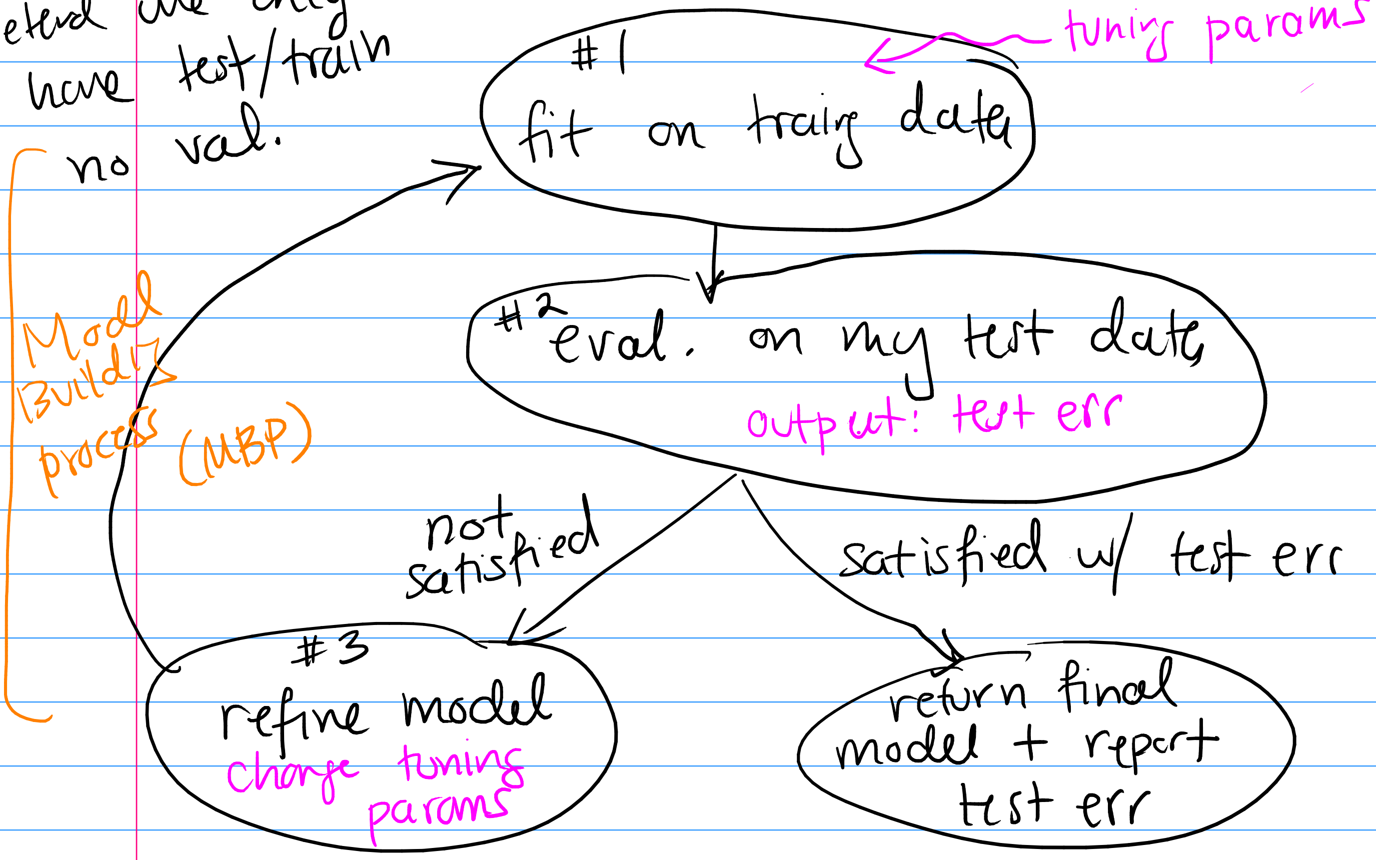
① traing used to directly fit the model (e.g. min RSS)

② validation used to select amog my models (e.g. Choose k)

③ test to estimate gen. perf. of my final chosen model.

Why do we need sep. val/test?

Pretend we only have test/train no val.



Cardinal Sin! can't let my ML algo see the test data

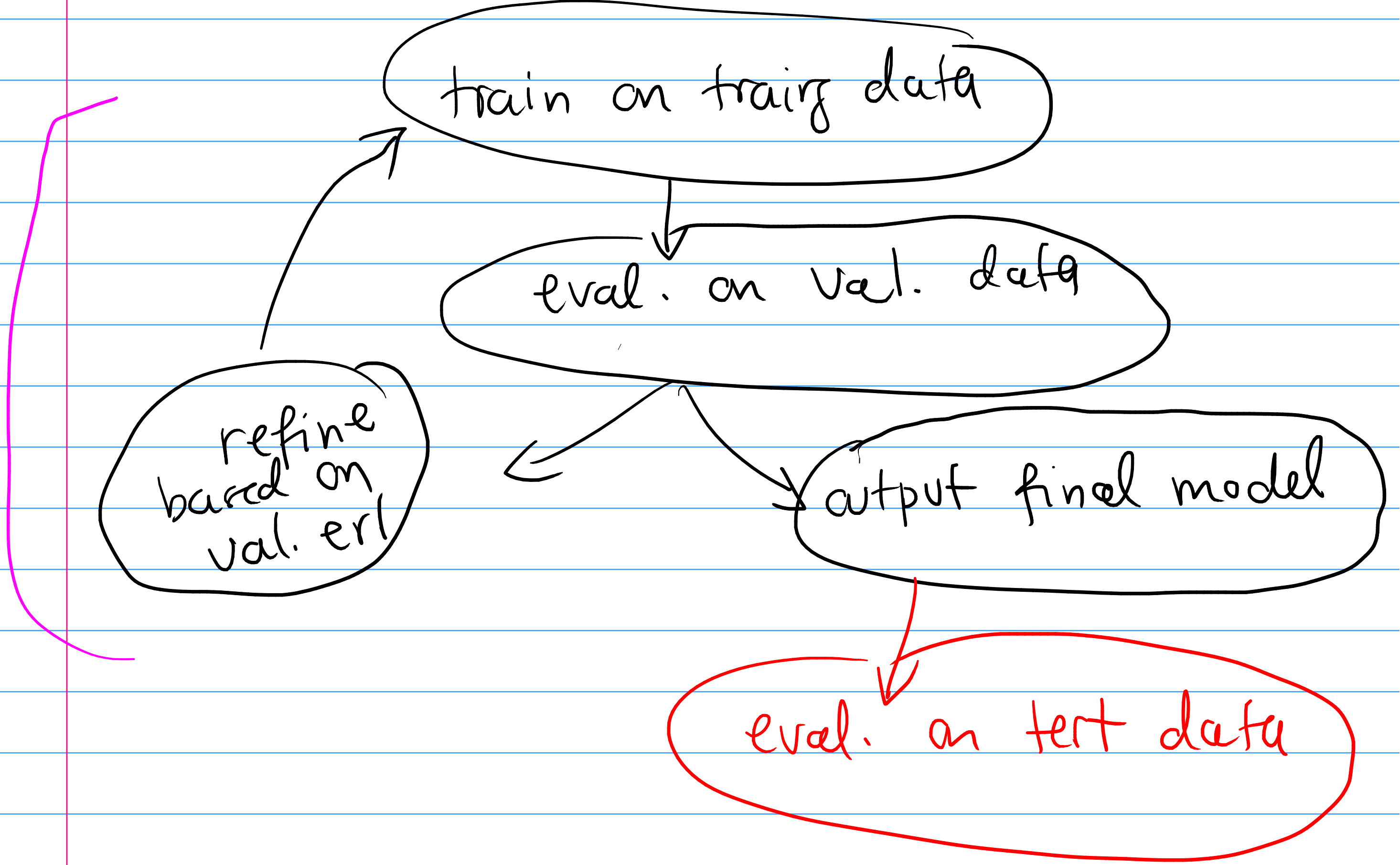
Q: Did my model see the test data?
After all, my algo only fit using the traing data.

A: Yeah, kinda. By iteratively refing model based on test err. — the test data has influenced the MBP.

Still need a true "hold-out" dataset to est. gen. perf.

Consequently, we split into 3 datasets

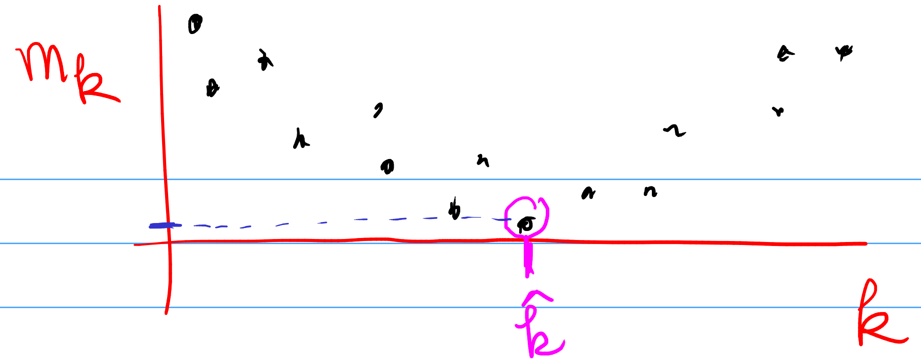
① train } used to build / choose model.
② val.
③ test



Ex. Choosing k for kNN

```
Split data into train, val., test
For k in seq of ks
    → train on training data using k neighbors
    → eval. on val. data
          $m_k$ = RMSE of val. preds
END FOR
$\hat{k} = \arg\min_k m_k$ = i.e. val. of k that produces min $m_k$
```

MBP



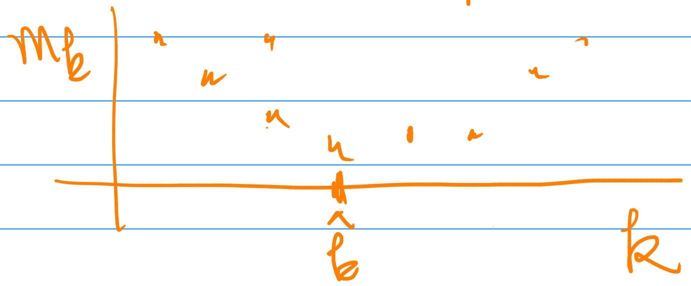
THEN to est. my gen. perf. I fit kNN using $\hat{k}$ neighbors on combined train/val and then eval. on my testing data.

Can we do this in a x-validated way?

Called: nested cross validation

```
Split data into I folds
For $i = 1, \ldots, I$
  → hold out fold i for testing
  → use other folds for train-val
  → split train-val into J folds
  For $j = 1, \ldots, J$
    → hold out fold j as val. data
    → use other folds as traing data
    For k in seq of ks
      → fit kNN usiy train and k neighbors
      → eval on val
        $m_k^{(j)}$ = RMSE on val.
  → $m_k = \text{mean}(m_k^{(j)})$
```

→ choose $\hat{k} = \arg\min_{k} m_k$



→ fit model w/ $\hat{k}$ on train-val

→ eval ony my test data

M_i = RMSE of $\hat{k}$ model on test

So after all of this I have

$M_1, \ldots, M_I$

So e.g. gen. perf. might be $\text{mean}(M_i)$