# Lecture 6: More Evaluation

Two reasons to evaluate:

1. get measure of gen. performance
2. model selection
   - → choosing K for KNN
   - → # covariates for OLS

Q: Can I use a test/train split to do this? (equiv. X-validation)

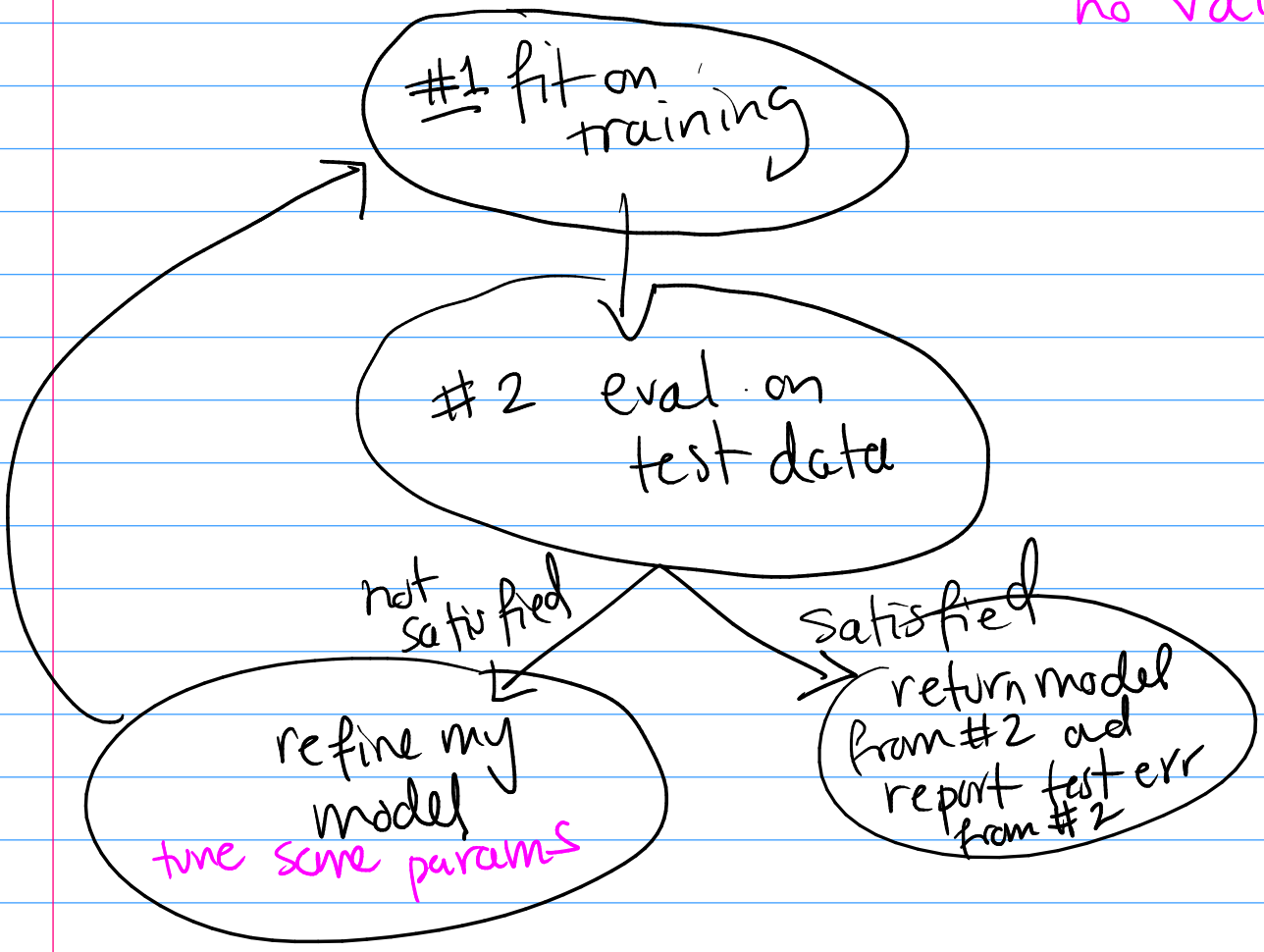
A: Yes, but need to be careful.

Proper way to do this: split data into 3 sets

1. training data: directly used to fit model (e.g. min RSS over training)
2. Validation data: used to select among my models (e.g. choose K)
3. test data to estimate gen. perf. of final model

# Why do I need sep. val / test?

Ex. model building process w/ just a train/test no val.




Need to avoid: can't let my MBP see or be influenced by test data

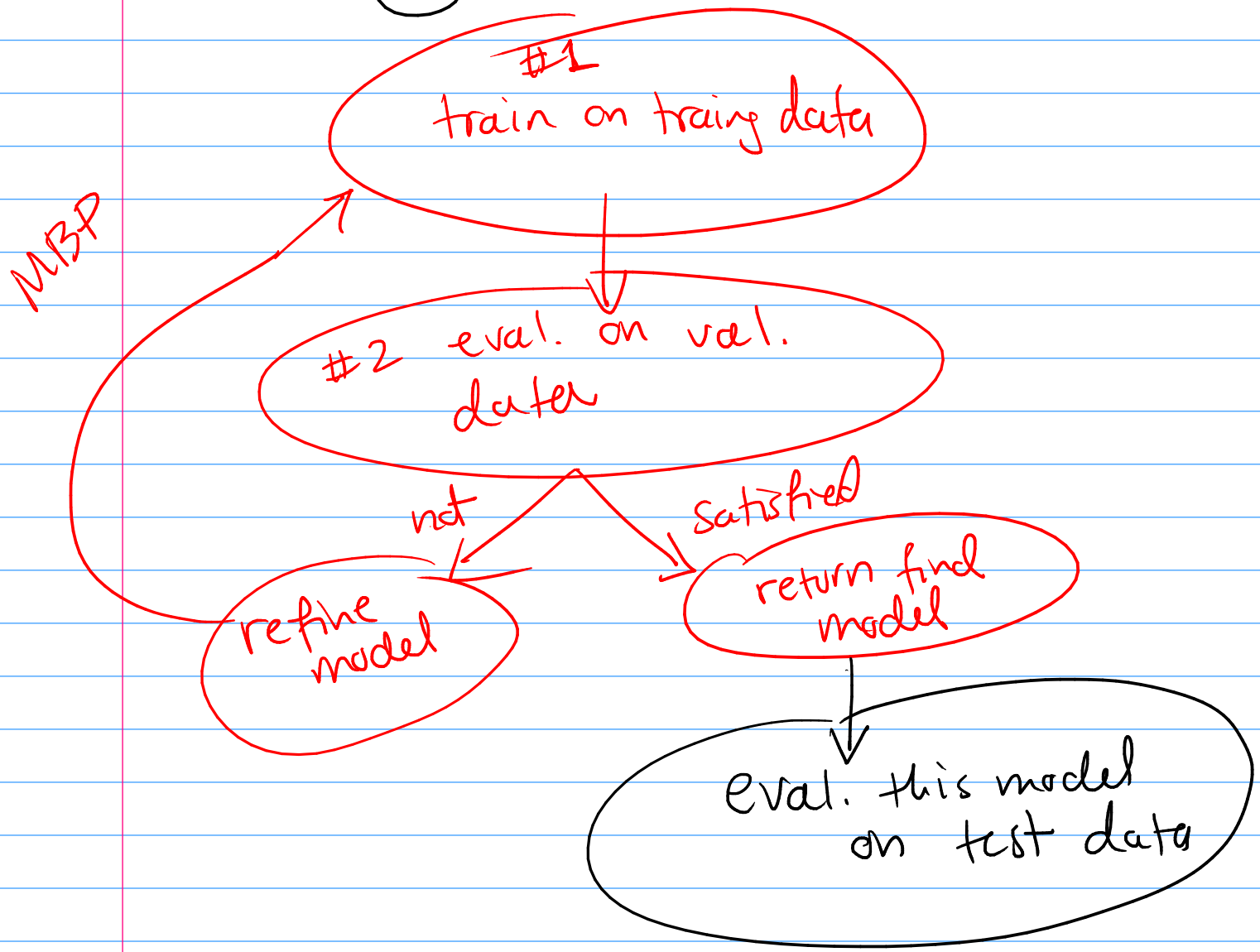
Q: Does my MBP see the test data?

A: Yes, basically hand-fitting my model. The refinement is directly influenced by test err.

Still need a true "hold-out" data set.

Consequently split into 3:

1. train
2. val.
3. test.




---

Ex. choosing K for KNN

Split data into train/test/val

For K in K-seq

① train on training using K neighbors

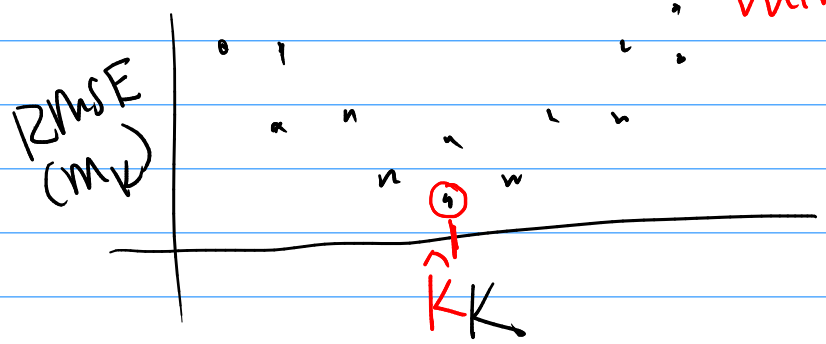
② eval on val.

$m_K$ = RMSE of val preds

END

MBP

$\hat{K} = \underset{K}{\text{argmin}}\ m_K =$ chose val K that has min val. err.




THEN fit KNN using $\hat{K}$ and all of the train/val. data and eval. this on test data.

↳ gives some est. of err for this MBP

Q: Can I do this in a X-val. way?

Yes, called nested x-val.

Split data into I folds

For $i = 1, \ldots, I$

→ hold out fold i for testing

→ use rest for train/val

→ split train/val into J folds

For $j = 1, \ldots, J$

→ hold out jth fold for val.

→ use rest for train

Fit my model

For K in K-seq

→ fit KNN on train w/ K neighbors

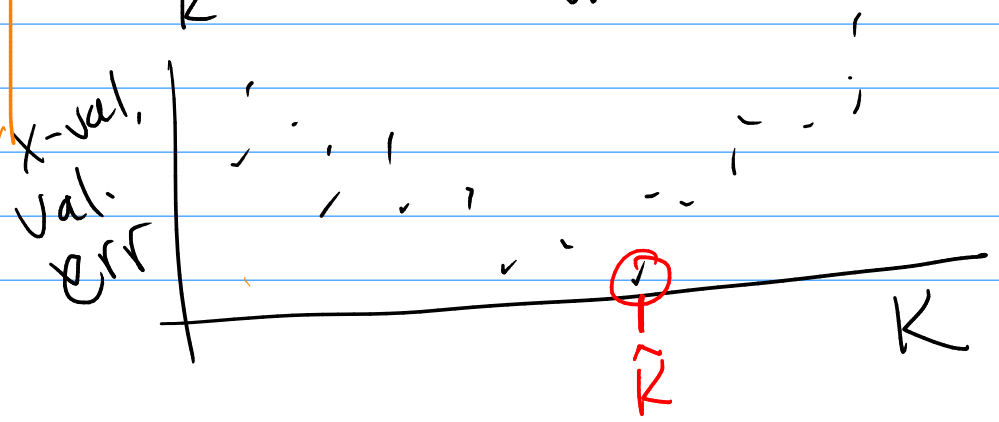
→ eval. on val.

→ $m_{jK}$ = RMSE

END

END

→ $m_K = \text{mean}(m_{jK})$



MBP

→ $\hat{K} = \underset{K}{\operatorname{argmin}}\ m_K$

→ fit using $\hat{K}$ on all my combined train/val.

→ eval on test data ⇝ $M_i$

END

end up w/ $M_1, \ldots, M_I$

maybe report $\text{mean}(M_i)$

↑ est. of gen-perf. for my MBP