Lecture 5: Evaluation

How do I choose among the various models?

e.g. choose k for kNN

e.g. choose design for OLS

We need some way to evaluate which model building process is going to perform better.

- → use OLS w/ 5 vars
- → use kNN w/ $k=5$
- → use kNN and optimize over possible values of k

A (maybe not always great) way of doing this!

Calculate some performence metric on $\hat{f}$ from the training data

e.g.

① training residual sums of squares

$$RSS_{train} = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$$

② training mean sq. error

$$MSE_{train} = \frac{RSS_{train}}{N}$$

③ traing root mean sq. err: $RMSE_{train} = \sqrt{MSE_{train}}$

④ training $R^2_{train} = 1 - \frac{RSS_{train}}{TSS_{train}}$

$\uparrow$ % of var. explained by $\hat{f}$

$TSS_{train} = \sum_n (y_n - \bar{y})^2$

Why isn't a training metric always a good measure of performance?

→ I don't actually care about performance on my traing data (I already know the answer)

→ I actually care about is perfomance of $\hat{f}$ on new/unseen data.
(generalization performance)

ERM:

$$\underbrace{\frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n))}_{\text{avg. loss over training data}} \approx \underbrace{\mathbb{E}\left[L(Y, f(X))\right]}_{\text{actual loss}}$$

Similarly:

$$\underset{\substack{\uparrow \\ \text{training metric/perf.}}}{\text{Metric}(\{y_n, f(x_n)\})} \approx \underset{\substack{\uparrow \\ \text{generalization performance}}}{\mathbb{E}\left[\text{Metric}(\{Y, f(X)\})\right]}$$

Focusing too much on training metric can be misleading.

This happens b/c when I evaluate $\hat{f}$ on my training data I am evaluating it on the same data used to build $\hat{f}$.

So my model building process has already seen the traing data - its not a fair evaluation.

Analogy: Test 1: I give you 10 practice problems and a week from now I give exam w/ exactly those 10 problems.

eval. on traing data →

Test 2: I give you 10 practice problem and a week from now I give exam w/ 10 related but not exactly the same questions.

generalization perf. →

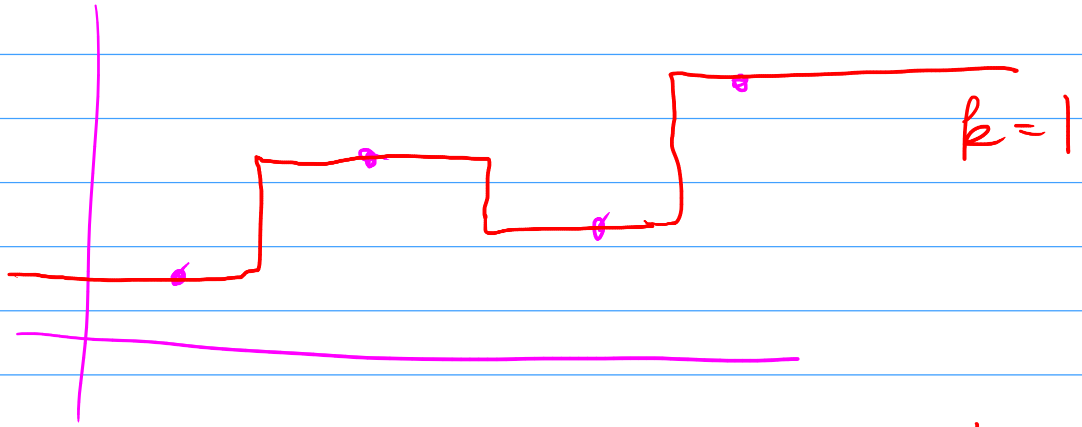
Ex. This can happen in SML too.

Consider optimizing the value of k for kNN by choosing the value of k that minimizes RSS_{train}.

What value of k do I choose?

$k = 1$

When $k=1$ I interpolate my traing data



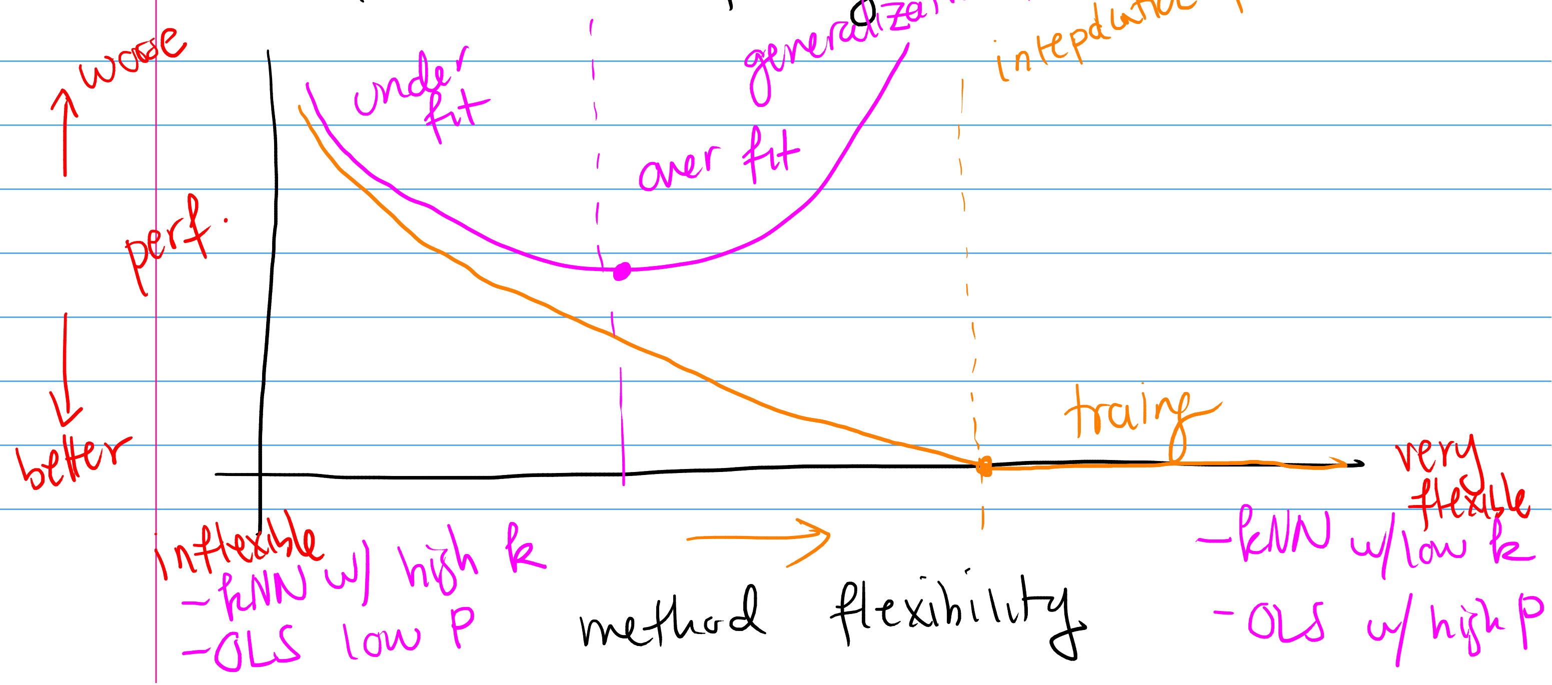
So $\hat{y}_n = y_n$ fer traing data $n=1,\ldots,N$

hence $RSS_{train} = \sum_n (\underbrace{y_n-\hat{y}_n}_{0})^2 = 0$

A similar story is true fer OLS when choosing what terms to include in the design.

as $p \to N$ I will interpdate my traing data

We call this "over-filting"



How do we solve this?

Let's estimate generalization perf.

Need independent set of (unseen) data to evaluate our model on.

Testing Data: $\{(\underset{\sim}{x}_{test,n}, y_{test,n})\}_{n=1}^{N_{test}}$

Procedure: ① use traing data to buid $\hat{f}$

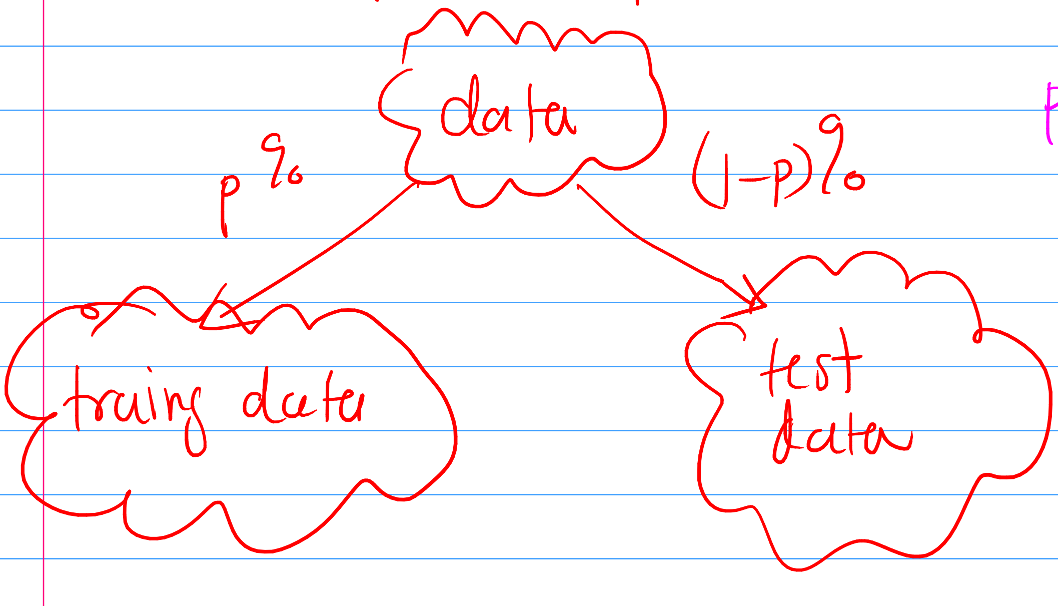
② eval. perf. of $\hat{f}$ on testing data

RSS_{test}, MSE_{test}, $RMSE_{test}$, R^2_{test}

↳ an estimate of generalizatin perf. of my model buildig process.

How do I get testing data?

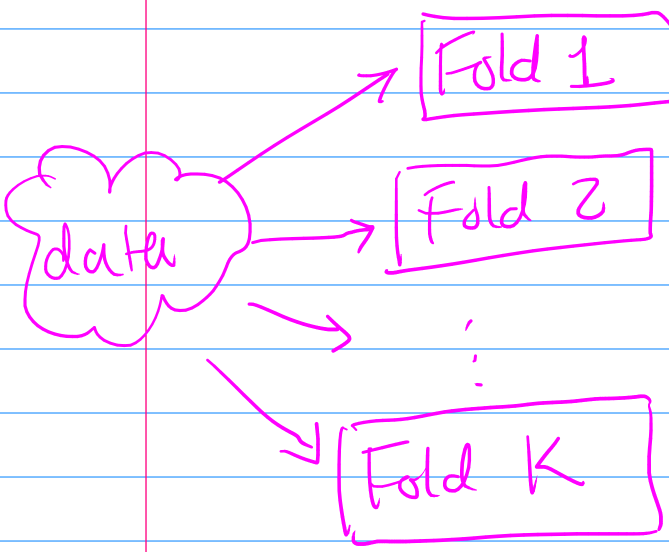
Do a test/train split.



p = 90, 80, 50

One can do this split multiple times which is called cross-validation

K-Fold Cross Validation



Cycle through folds

For $k=1,\dots,K$

① train model on all but fold k

② test model on fold k
call this test metric m_k

So at the end of loop I have

$$m_1, m_2, \dots, m_K$$

Can combine to get overall perf. metric

$$m = \text{median}(m_1, \dots, m_K)$$

But wait! If I train K different models which do I use?

None. I train a model using all of my data.

We want to evaluate perfomance generally for two reasons:

1. I want to know how good this MBP might be
2. I want to choose among varias models.

We can use a test/train split or X-validation to choose among models - but we need to be very careful.