Cross Validation

Objectives

- Understand Model fitting and Bias-Variance Trade off
- Be able to explain cross validation and why it is used
- Be able to explain how to account for different model fits and steps to take to move towards a better fit.
- Understand difference between hyperparameters and parameters
- Aside: introduce feature selection/elimination
- Contrast cross validation with other statistical model comparison methods

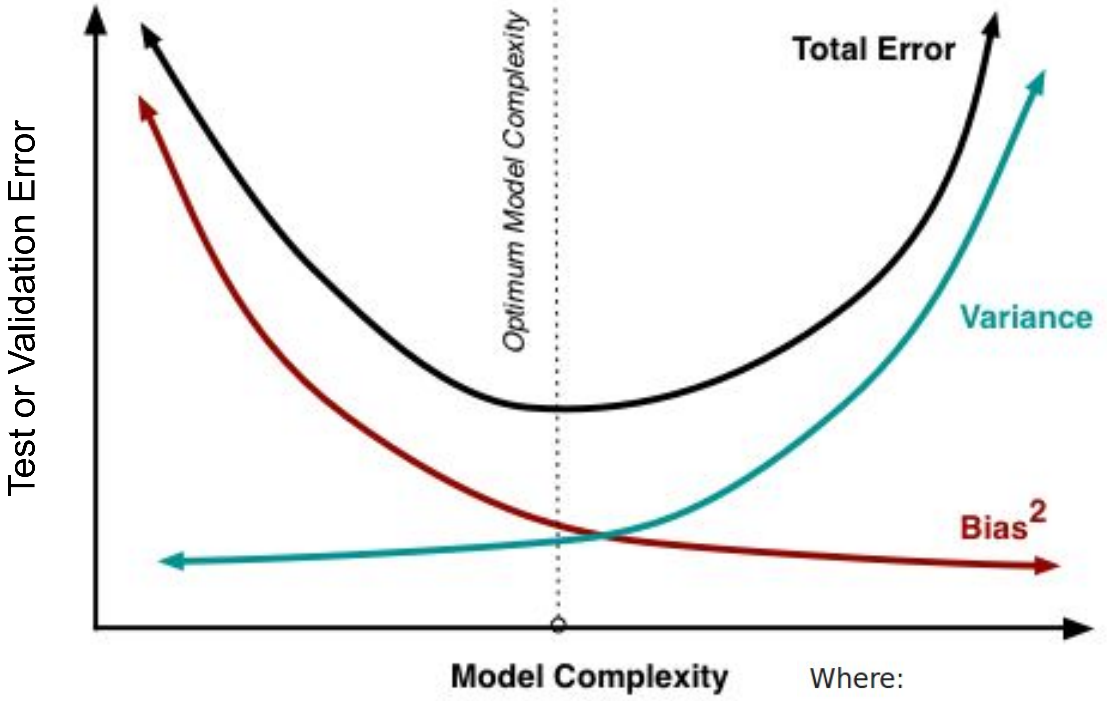
Motivation: We’re stuck with the bias-variance tradeoff

So how is the “correct” model complexity chosen?

Model complexity can be the order of the fit, the number of features, interaction of features, number of splits (decision tree), number of neurons/layer in a neural net, number of layers...

We can’t do anything to reduce the sampling error from the population, but can we find the model complexity that minimizes the sum of the bias^2 and the variance?

Typical Bias-Variance tradeoff behavior



$$\mathrm{E}\Big[\big(y-\hat{f}\,(x)\big)^2\Big] = \mathrm{Bias}\big[\hat{f}\,(x)\big]^2 + \mathrm{Var}\big[\hat{f}\,(x)\big] + \sigma^2$$

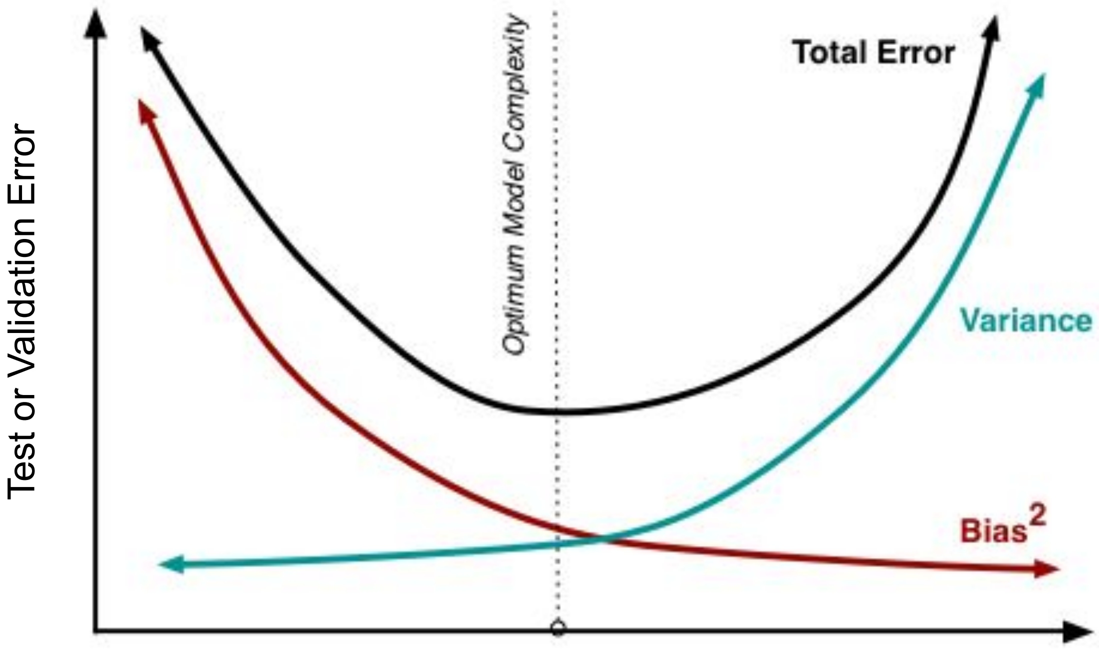
Where:

$$\mathrm{Bias}\big[\hat{f}\,(x)\big] = \mathrm{E}\big[\hat{f}\,(x) - f(x)\big]$$

and

$$\mathrm{Var}\big[\hat{f}\,(x)\big] = \mathrm{E}[\hat{f}\,(x)^2] - \mathrm{E}[f(x)]^2$$

Typical Bias-Variance tradeoff behavior



Bias: unknown

Variance: unknown

But, if we have data with targets that the model has **not** trained on, we can plot the expected residual for a given model complexity, and choose the complexity that gives the lowest residual.

$$\mathrm{E}\Big[\big(y-\hat{f}(x)\big)^2\Big] = \mathrm{Bias}\big[\hat{f}(x)\big]^2 + \mathrm{Var}\big[\hat{f}(x)\big] + \sigma^2$$

Where:

$$\mathrm{Bias}\big[\hat{f}(x)\big] = \mathrm{E}\big[\hat{f}(x) - f(x)\big]$$

and

$$\mathrm{Var}\big[\hat{f}(x)\big] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[f(x)]^2$$

Bias-Variance Trade off Example

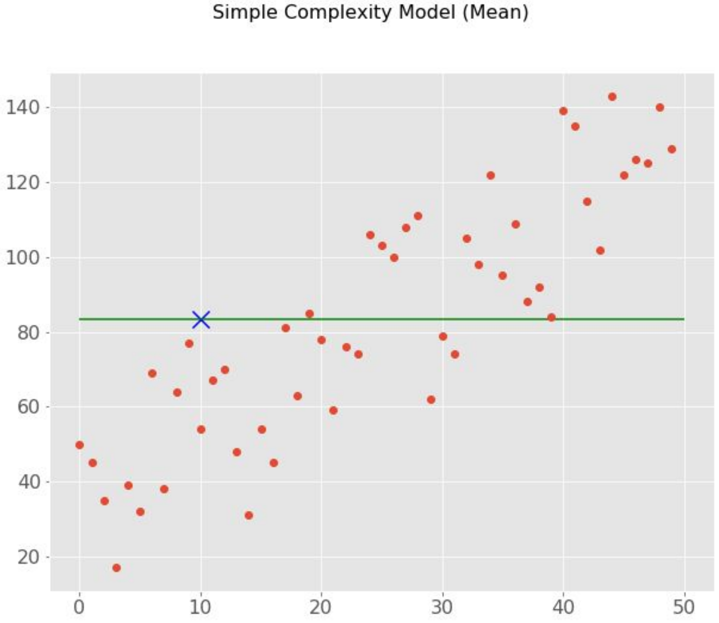
Let's say you and a friend are having an argument about which team is better in a game, the Red team or Blue team. You like the Blue team and your friend likes the Red team

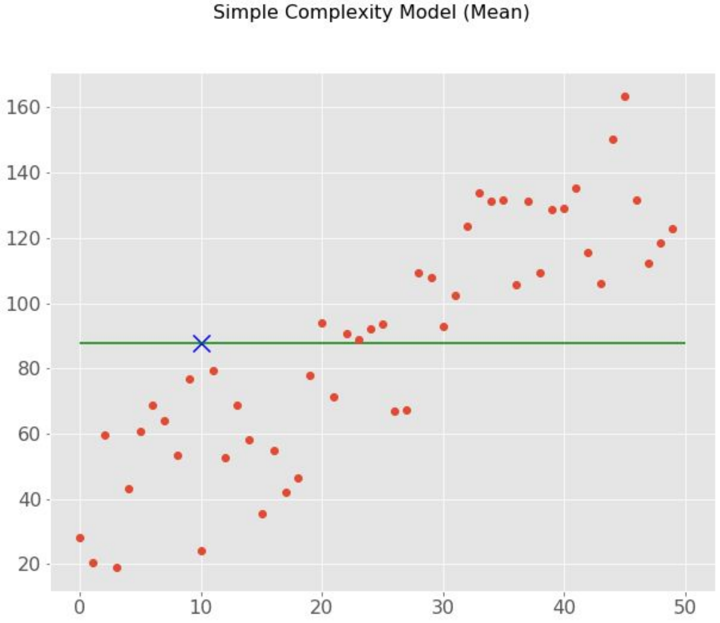
Simple Mode Complexity

- We have **HIGH** bias, with **LOW** variance **OF PREDICTED VALUE**

- Your friend is really likes the red team because he loves the color red and everyone in his family and friends likes the color red(**high bias**) so no matter how much you try to convince him, he will not deviate his expected decision(**low variance**).

Simple Model Complexity



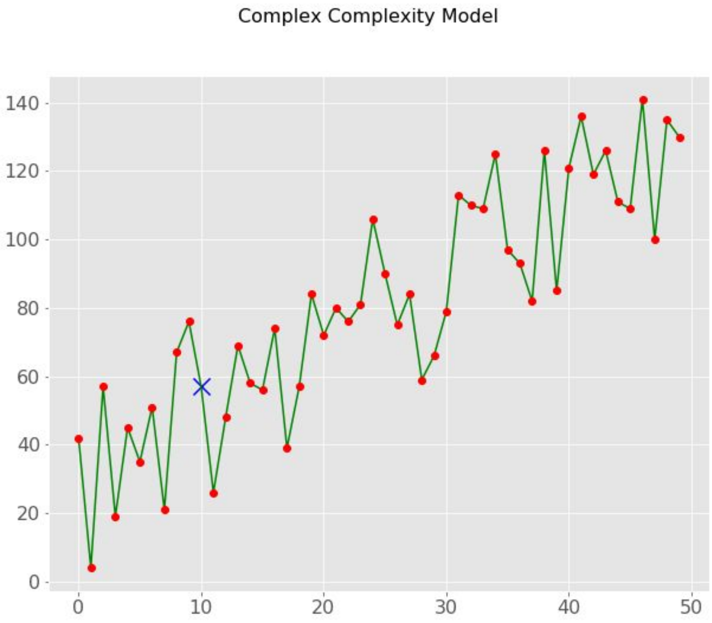


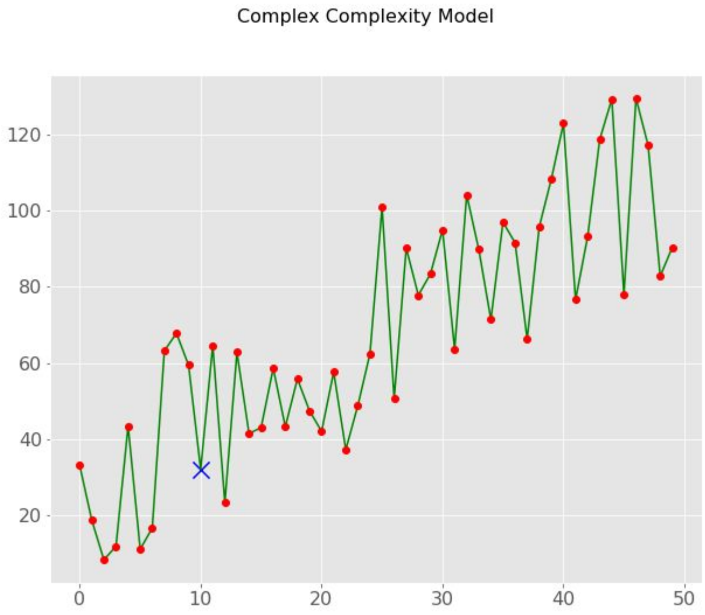
Complex Model Complexity

- We have **LOW** bias, with **HIGH** variance **OF PREDICTED VALUE**

- Your friend is very unsure about both teams and just likes the red team because he read something on the internet(**low bias**). He is very easily convinced, all you have to do is tell him your opinion whether it's true or not and he will shift his view based on what you tell him(**high variance**).

Complex Model Complexity





Check-in!

- Come up with an example of Simple Complexity Model
  - What’s its Variance, and what’s its Bias

- Come up with an example of a Complex Complexity Model
  - What’s its Variance, and what’s its Bias

- How can we mitigate both Bias and Variance when it comes to modeling?
  - Think about it… (we haven’t covered it yet)

Cross validation

“Cross-validation ... is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set.”
- Wikipedia

We use cross-validation for two things:

1) Attempting to quantify how well a model (of some given complexity) will predict on an unseen data set

2) Tuning hyperparameters of models to get best predictions.

Scikit-learn tangent (hyperparameters?)

Hyperparameters Vs. Parameters

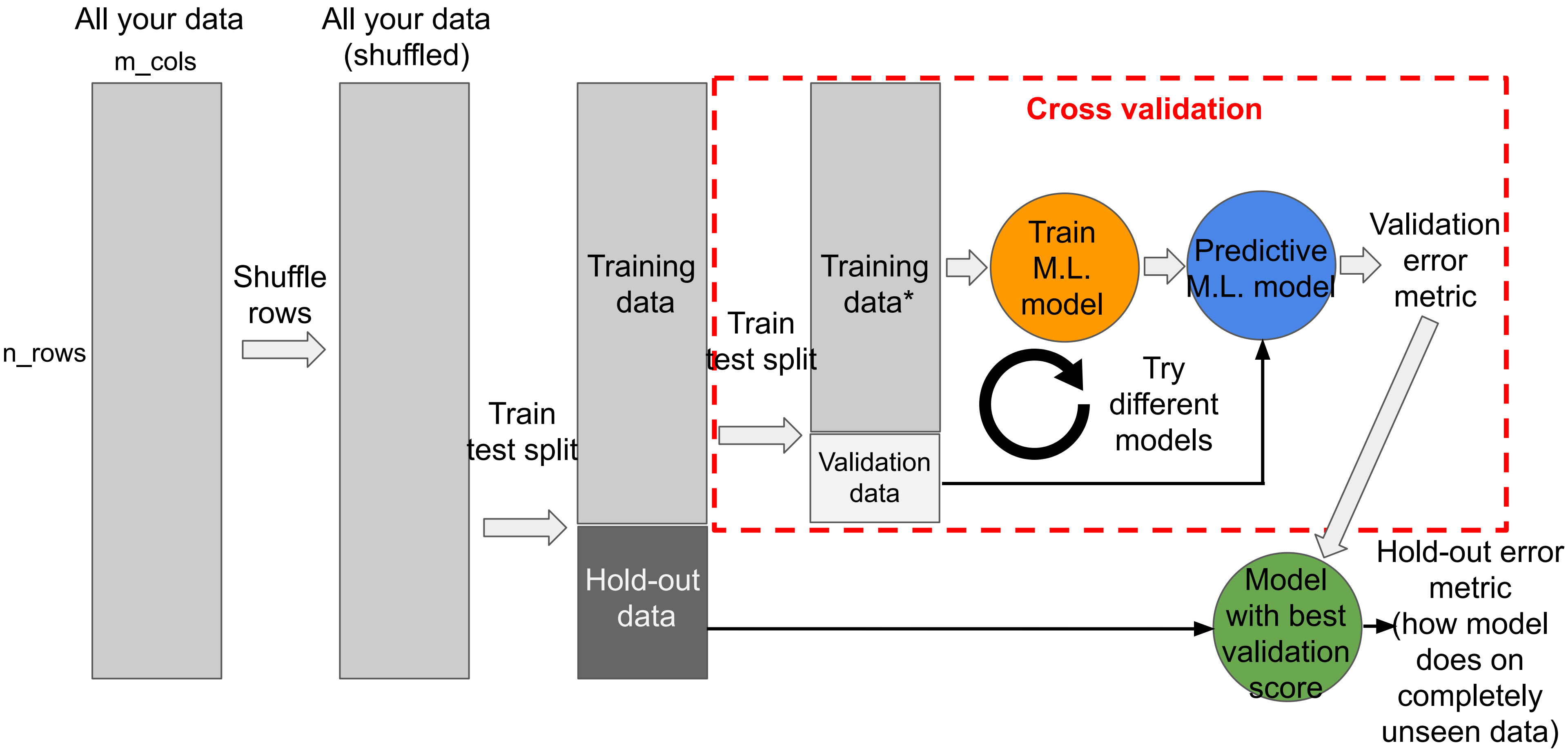
Parameters

- They are required by the model when making predictions.
- They are often not set manually and estimated or learned from data.
- I.E. Normal distribution parameters - Mean,Standard Deviation

Hyperparameters

- Set manually by user
- They are often used in processes to help estimate model parameters.
- Tuned to better predict for model.
- I.E. KNN Hyper Parameters - Number neighbors, distance metric, etc...

Cross validation - illustrated



Cross validation - in words

1. Split your data (after splitting out hold-out set) into training/validation sets.

   70/30, 80/20 or 90/10 splits are commonly used

2. Use the training set to train several models of varying complexity.

   e.g. linear regression (w/ and w/out interaction features), neural nets, decision trees, etc.

3. Evaluate each model using the validation set.

   calculate R^2, MSE, accuracy, or whatever you think is best

4. Keep the model that performs best over the **validation** set.

People use different terms for the splits.

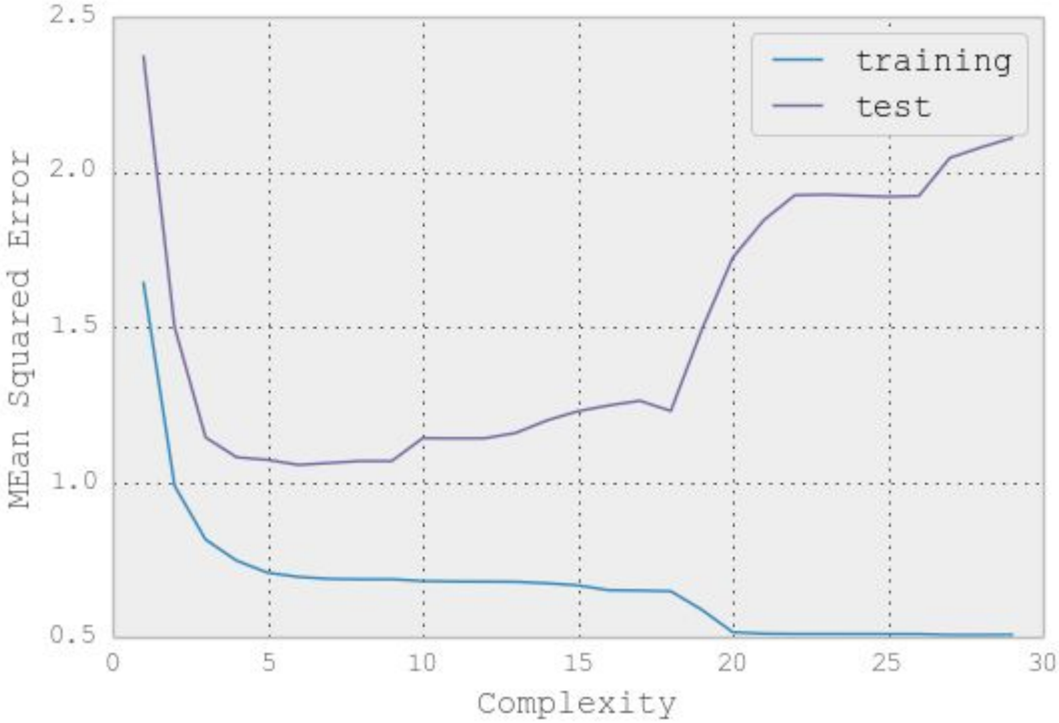
All data -> Train, Test, Hold-out

All data -> Train, Validate, Test

All data -> Train, Validate, Hold-out

All the same idea.

Cross validation - what you should get



Number of features, interaction between features, order of features

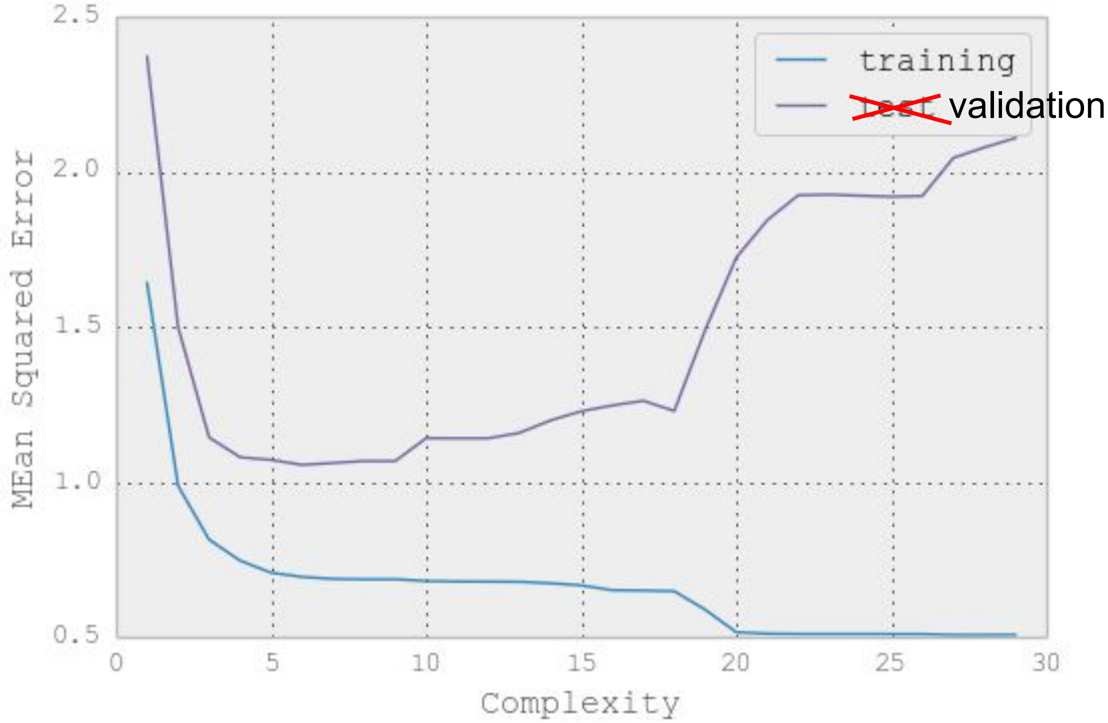
Which error is most important to minimize? Why?

What model complexity is best? How did you decide?

At the optimum model complexity, what is the bias? What is the variance?

Is it possible for the test error to be lower than the training error?

Cross validation - what you should get



Minimize validation error due to it mimicking “Unseen” data

Around 5 is the optimal complexity due to minimizing both errors

Bias and variance is minimized, we do not know the underlying Bias and Variance

Possible, could be due to data leakage between train and test data. Can also happen randomly.

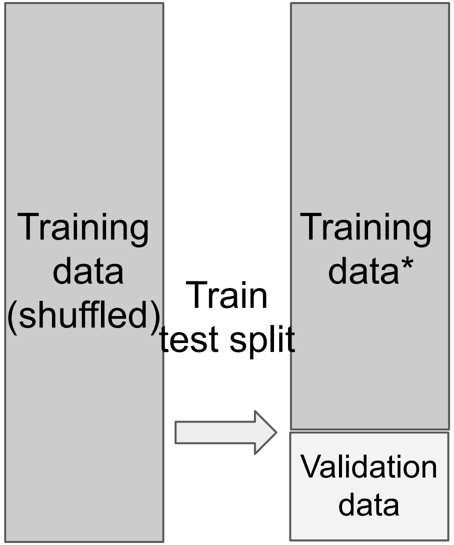
Number of features, interaction between features, order of features

Four Cases of Fitting

1. Underfitting – Validation and training error high
2. Overfitting – Validation error is high, training error low
3. Good fit – Validation error low, slightly higher than the training error
4. Unknown fit - Validation error low, training error 'high -- Caution!

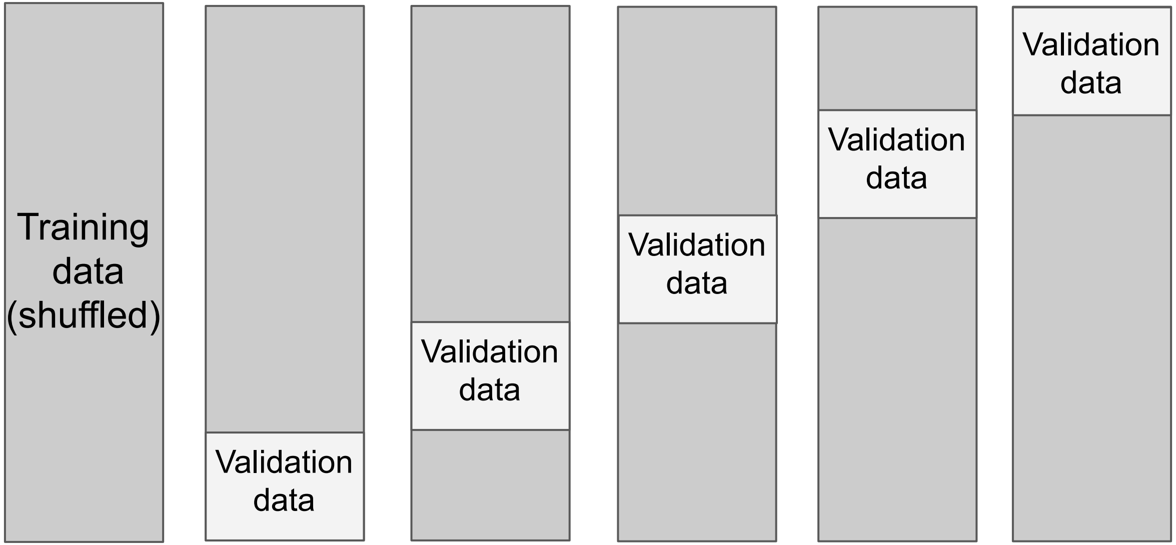
Types of Cross Validation

A single train-test split



After training get only 1 estimate of validation error (what if validation data very different from training data by chance?!?)

k-fold (showing k=5)



After training get k estimates of validation error from the same model complexity, so calculate the mean validation error from those five estimates. This gives a more robust, less variable estimate.

Special case of k-fold: k = n (Leave one out CV).
Models are highly correlated in LOOCV.

What to do if your model is overfitting

Pretty common. If you are starting with 5-10 features that can already be pretty complex. Often assume that if we start with linear regression this is “simple.” This is not necessarily so.

1. **Get more data…** (not usually possible/practical)

2. **Subset Selection:** keep only a subset of your predictors (i.e, dimensions)

3. **Regularization:** restrict your model’s parameter space (Wednesday)

4. **Dimensionality Reduction:** project the data into a lower dimensional space (later in course)

Subset selection

Best subset: Try every model. Every possible combination of *p* predictors

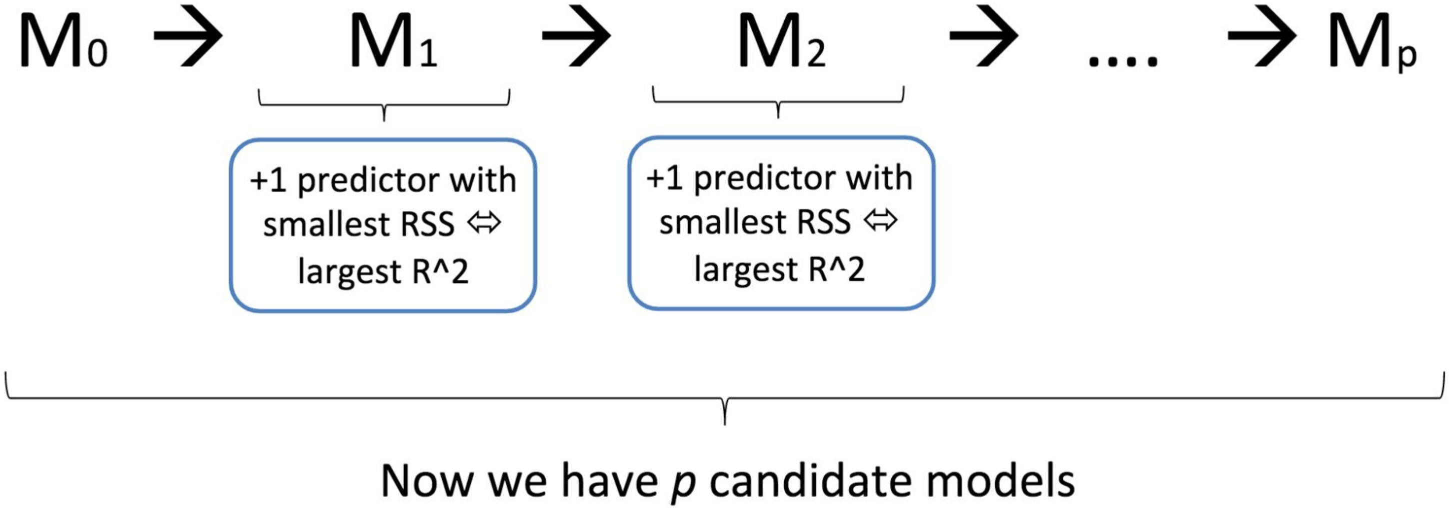
- Computationally intensive. 2^p possible subsets of *p* predictors
- High chance of finding a “good” model by random chance.

… A sort-of monkeys-Shakespeare situation …

Stepwise: Iteratively pick predictors to be in/out of the final model.

- Forward, backward, forward-backward strategies
  - Forward: starting with just one and adding more features, one-by-one
  - Backward: starting with them all, and removing one-by-one
- Sklearn features only backward recursive elimination.

Forward step-wise selection



$M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow .... \rightarrow M_p$

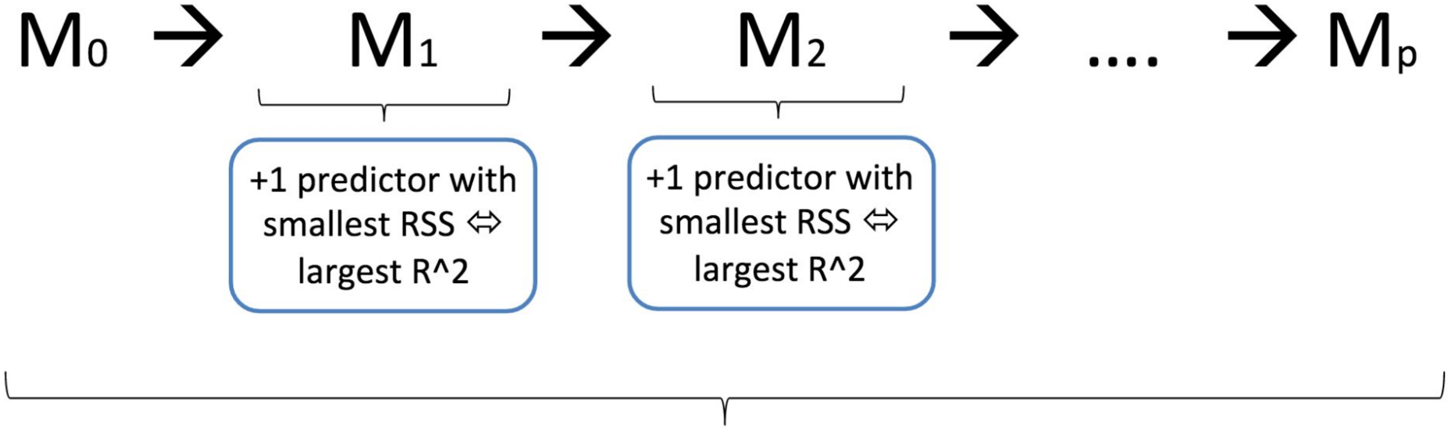
+1 predictor with smallest RSS ⇔ largest R^2

+1 predictor with smallest RSS ⇔ largest R^2

Now we have *p* candidate models

Is R^2 a good way to decide amongst the resulting *(p+1)* candidates?

Forward step-wise selection



$M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow .... \rightarrow M_p$

+1 predictor with smallest RSS ⇔ largest R^2

+1 predictor with smallest RSS ⇔ largest R^2

Now we have *p* candidate models

Is R^2 a good way to decide amongst the resulting *(p+1)* candidates?

R^2 doesn’t penalize for model complexity, so no.
Use Mallow’s C_p, or AIC, or BIC, or Adjusted R^2.

… or just use cross-validation with any error measurement.

Statistical metrics that penalize model complexity

$$C_p = \frac{1}{n}(RSS + 2\underline{p}\hat{\sigma}^2)$$

Mallow's Cp
p is the total # of parameters
$\hat{\sigma}^2$ is an estimate of the variance of the error, ε

$$AIC = -2logL + 2 \cdot \underline{p}$$

L is the maximized value of the likelihood function for the model estimated

$$BIC = \frac{1}{n}(RSS + log(n)\underline{p}\hat{\sigma}^2)$$

This is Cp, except 2 is replaced by log(n). log(n) > 2 for n>7, so BIC generally exacts a heavier penalty for more variables

$$Adjusted\ R^2 = 1 - \frac{RSS/(n-\underline{p}-1)}{TSS/(n-1)}$$

Similar to R^2, but pays price for more variables

Side Note: Can show AIC and Mallow's Cp are equivalent for linear case

Statistical metrics that penalize model complexity

$$C_p = \frac{1}{n}(RSS + 2\underline{p}\hat{\sigma}^2)$$

Mallow's Cp
p is the total # of pa
$\hat{\sigma}^2$ is an estim ror, ε

$$AIC = -2logL + 2 \cdot p$$

likelihood
stimated

$$BIC$$

is Cp, except 2 is replaced by log(n).
log(n) > 2 for n>7, so BIC generally exacts a heavier penalty for more variables

$$Adj = 1 - \frac{RSS/(n-\underline{p}-1)}{TSS/(n-1)}$$

Similar to R^2, but pays price for more variables

Yes, you can use these, but why?
You have something better….

Side Note: Can show AIC and Mallow's Cp are equivalent for linear case

Something better

Use cross validation

Questions

- Describe the Bias-Variance Trade-off and how we find the optimal model
- Describe the four forms of fitting for a model
- Describe Hyperparameters vs. Parameters
- Describe why cross validation is used
- Describe how to do it (especially k-fold cross validation)
- What do you do if your model is overfitting
- What do you do if your model is underfitting
- What are the ways to statistically compare models and select the best one

- What are your questions!

Objectives and Going forward

- Understand Model fitting and Bias-Variance Trade off
- Be able to explain cross validation and why it is used
- Be able to explain how to account for different model fits and steps to take to move towards a better fit.
- Understand difference between hyperparameters and parameters
- Aside: introduce feature selection/elimination
- Contrast cross validation with other statistical model comparison methods

This afternoon:

- In the individual assignment, code cross validation from scratch