RESAMPLING METHODS

Chapter 05

Validation for Decisionmaking

- The Validation Set Approach
- Leave-One-Out Cross Validation
- K-fold Cross Validation
- Bias-Variance Trade-off for k-fold Cross Validation
- Cross Validation on Classification Problems

What are resampling methods?

- Tools that involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain more information about the fitted model
  - Model Assessment: estimate error rates on unseen data
  - Model Selection: select appropriate model hyperparameters (e.g. level of model flexibility)
- They are computationally expensive! But these days we have powerful computers
- Two resampling methods:
  - Cross Validation
  - Bootstrapping

Borghetti's "Golden Rule" of Performance Reporting

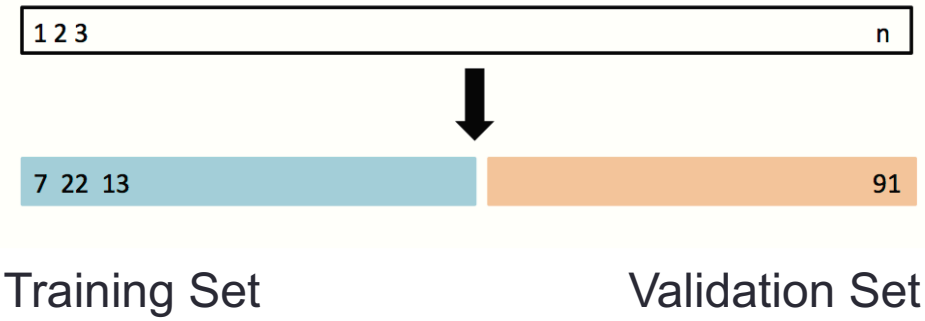
- You already know that:
  - Training sets are used to fit models, and training set accuracy may vary greatly from test set performance, so it is inappropriate to report model quality based on the training set performance
- Golden Rule: If you use an observation as part of your decision-making process, then you should NOT use it as part of a set of data you are using to *report* performance
  - Decision-making includes things like fitting a model, choosing hyperparameters, and selecting which model to use
- If you need to choose hyperparameters or select the best model, don't use the test-set data to make the decision!

Training v. Validation v. Test sets

- TEST SET: Used for performance prediction **only**. It estimates performance of a model on unseen data. ***Sequester the test set before ML!***
- NON-TEST-SET DATA:
  - Training Set: Used to "fit" parameterized models (e.g. find coefficients/weights
  - Validation Set: Use when considering multiple models or making hyperparameter decisions
    - Estimate of each models quality from non-training data once the model has been fit on the training data
    - Used to make selection decisions (e.g. pick the best k in KNN; select whether LDA or QDA model works best)

5.1.1 Model Selection using The Validation Set Approach

- Goal: select the best model (e.g. LDA vs. QDA)
- If we have a large data set, randomly* split the non-test data into *training* and *validation* sets
- Use the training set to build each possible model (i.e. the different combinations of variables)
- Select the model that gave the lowest error rate when applied to the validation set



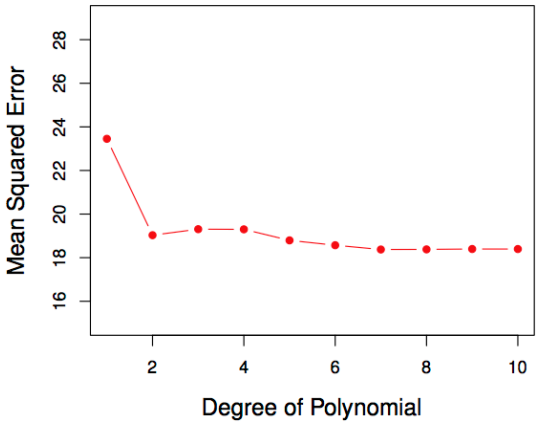
*be careful when randomizing selection from time-series or sequence-based data

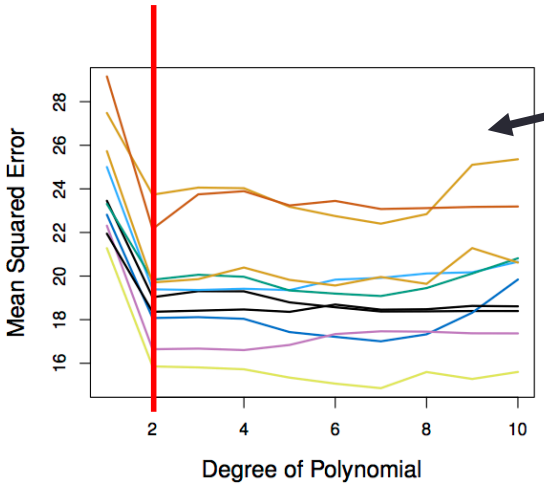
Model Selection using Validation Set Example: Auto Data

- Suppose that we want to predict mpg from horsepower
- Goal: select the best of 10 possible models:
  - Model 1: features = horsepower (Linear only)
  - Model 2: features = horsepower + horspower^2 (Linear + Quadratic)
  - …
  - Model 10: features = horsepower + horspower^2 +…+ horsepower^{10}
- Which model gives a better fit?
  - Randomly split Auto data set into training (196 obs.) and validation set (196 obs.)
  - Fit both models using the training data set
  - Evaluate MSE on each model using the validation data set
  - Select model with the lowest validation MSE
- **Question: If we use the validation set to determine which model is the better fit, is the MSE of the winning validation set a good estimate for MSE on novel/unseen data?**

Validation Set Variability

- Model selection task: which order of polynomial model fits best (from polynomials with orders 1:10)
- Left: Validation error rate for a single (random) 50/50 split
- Right: Validation method repeated 10 times, each time the split is done randomly
- There is a lot of variability among the MSE's…





"Spaghetti plot"

Validation Set Approach in summary

- Advantages:
  - Simple
  - Easy to implement

- Disadvantages:
  - The validation MSE of a single split can be highly variable
  - Only a subset of observations are used to fit the model (training data). Statistical methods tend to perform worse when trained on fewer observations

Cross Validation

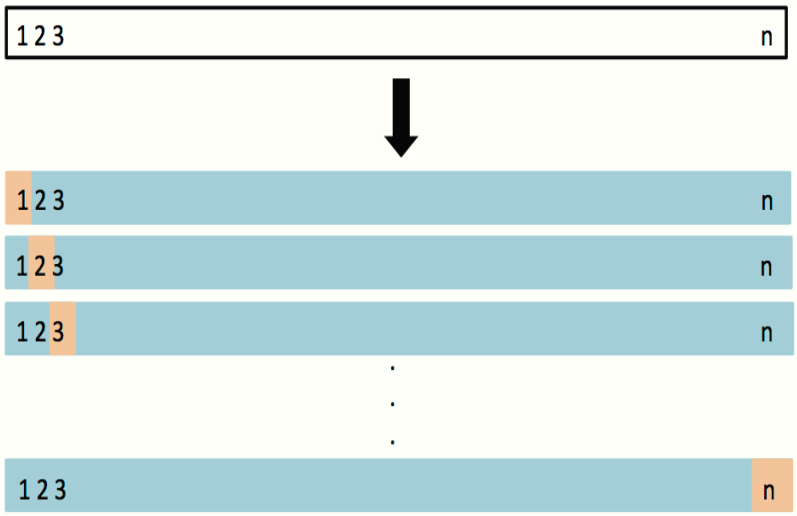
- Goal: reduce variability of the results of the validation set method
- Intuition: repeated resampling (*bootstrapping*) can yield better statistical properties on the estimate of a value
- Procedure:
  - Repeatedly conduct a train-then-evaluate process using different subsets of the data.
  - Estimate the performance as the mean of the (lower variance) performance estimates

Bootstrap (5.2)

- In class exercise

5.1.2 Leave-One-Out Cross Validation (LOOCV)

- This method is similar to the Validation Set Approach, but it tries to address the ValSet's disadvantages
- For each suggested model, do:
  - Split the data set of size n into
    - Training data set (blue) size: n -1
    - Validation data set (beige) size: 1
  - Fit the model using the training data
  - Validate model using the validation data, and compute the corresponding MSE
  - Repeat this process n times
  - The MSE for a model is computed as follows:



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i$$

LOOCV vs. the Validation Set Approach

- LOOCV has less chance of statistical sample bias
  - We repeatedly fit the statistical learning method using training data that contains n-1 observations - almost all the data set is used for training
- LOOCV produces a single MSE
  - The validation set approach produces different MSE when applied repeatedly due to randomness in the splitting process, while performing LOOCV multiple times will always yield the same results, because we split based on 1 obs. each time
- LOOCV is computationally intensive (disadvantage)
  - We fit each model n times

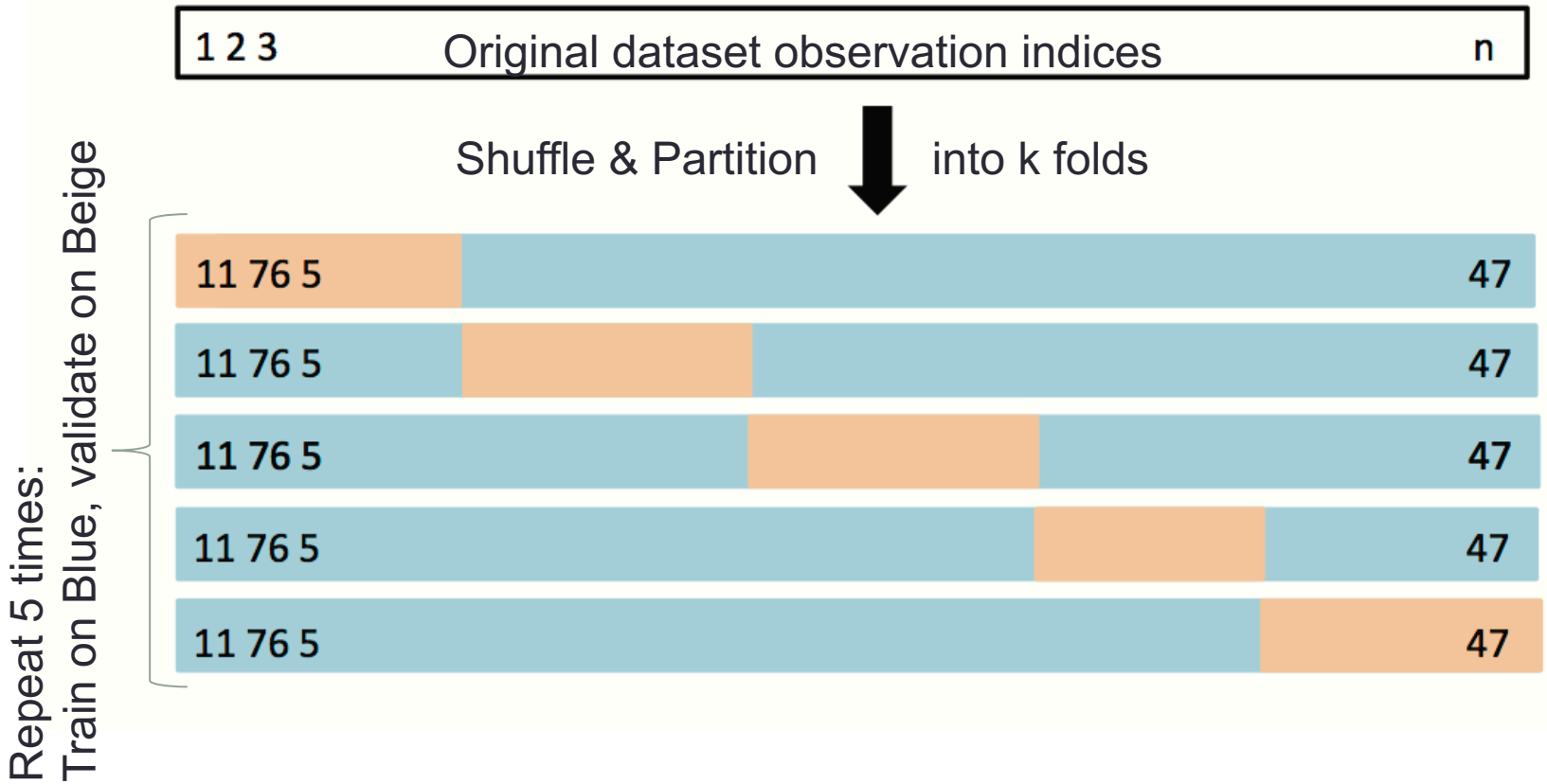
5.1.3 k-fold Cross Validation

- LOOCV is computationally intensive, and the validation set approach has high variability… is there a hybrid approach?
- **k-fold Cross Validation:**
- Randomly divide the data set into k different partitions (e.g. k = 5, or k = 10) known as "folds"
- Repeat a train-validate process k times using those folds:
  - In the i^{th} iteration, we train using all of the folds *except* the i^{th} and we validate on the data from the i^{th} fold.
- By averaging the k different MSE's we get an estimated error rate for unseen observations

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i$$

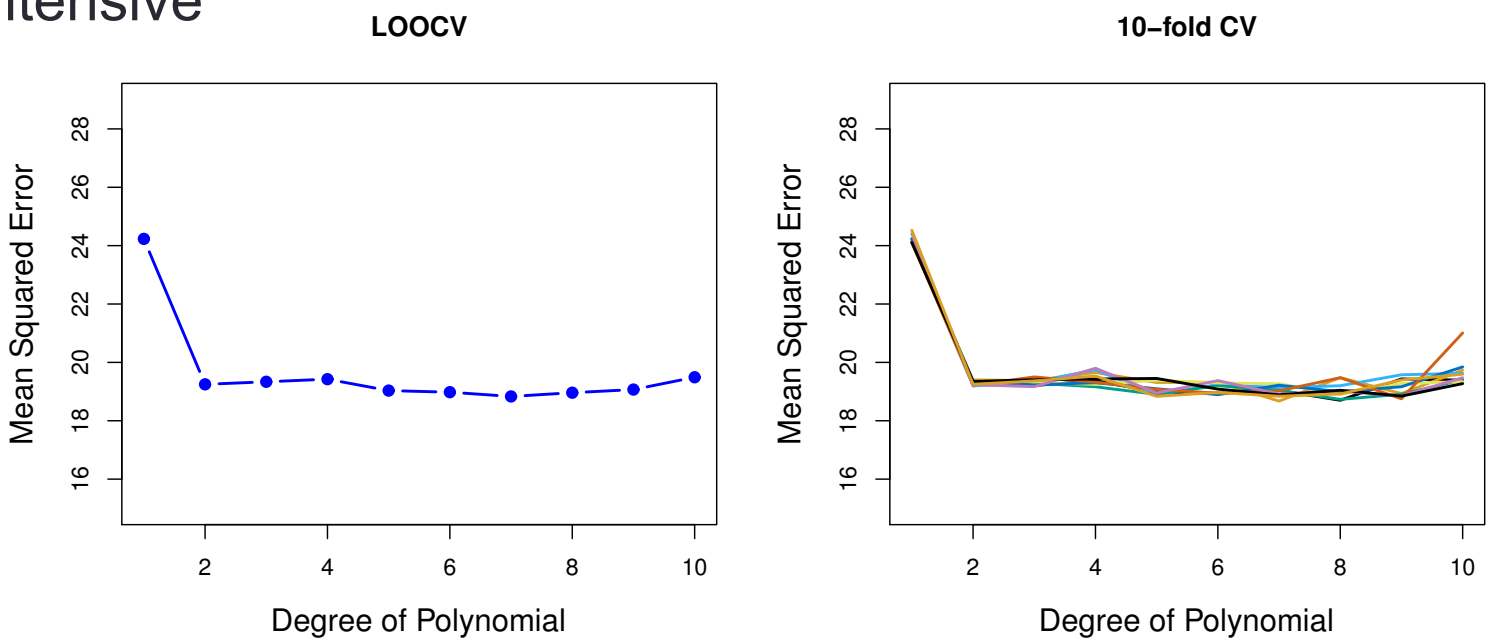
k-fold Cross Validation

Example: k = 5



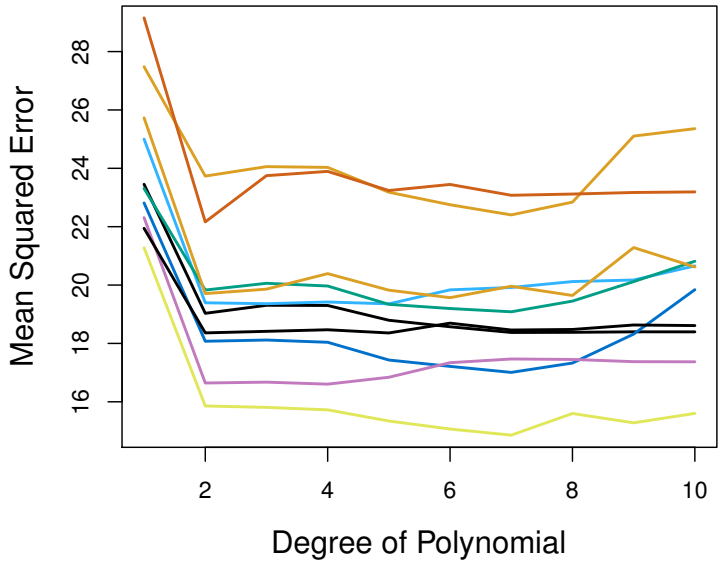
Auto Data: LOOCV vs. k-fold CV

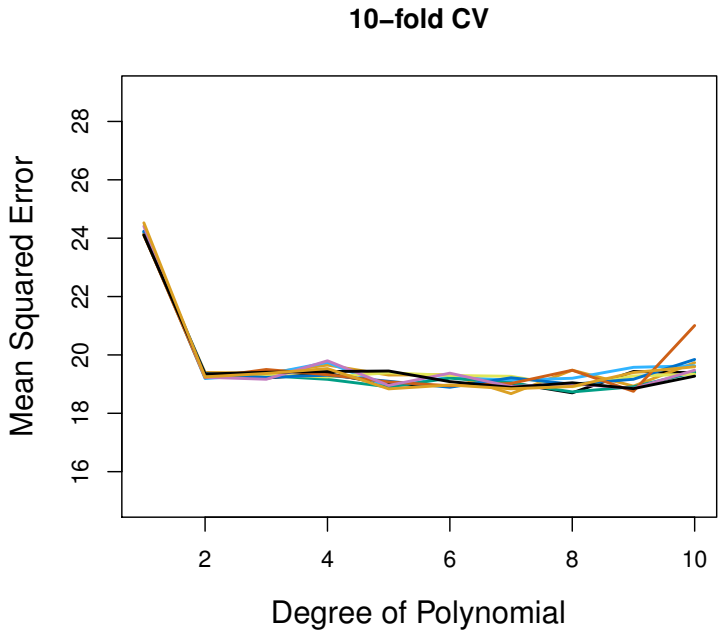
- Left: LOOCV error curve
- Right: 10-fold CV was repeated 9 times, and the figure shows the slightly different CV error rates
- LOOCV is a special case of k-fold, where k = n
- They are both stable, but LOOCV is more computationally intensive



Auto Data: Validation Set Approach vs. k-fold CV Approach

- Left: Validation Set Approach
- Right: 10-fold Cross Validation Approach
- 10-fold CV is more stable

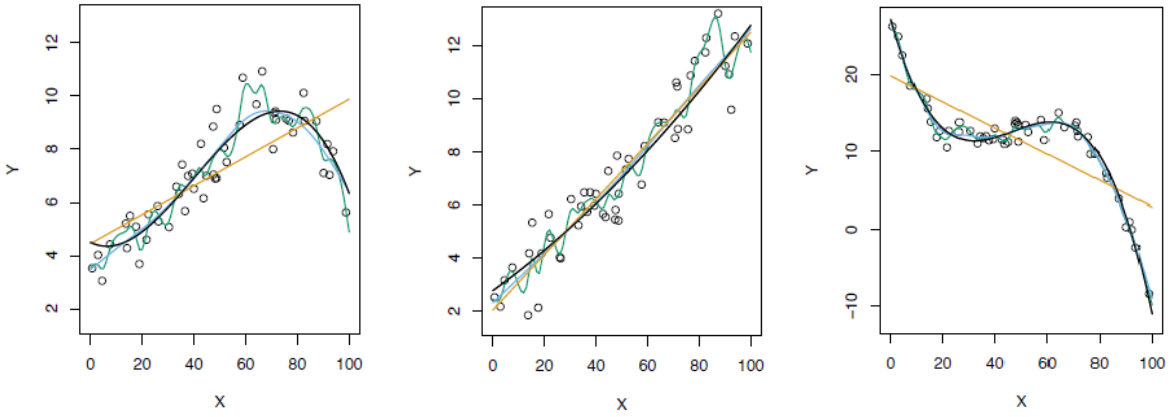




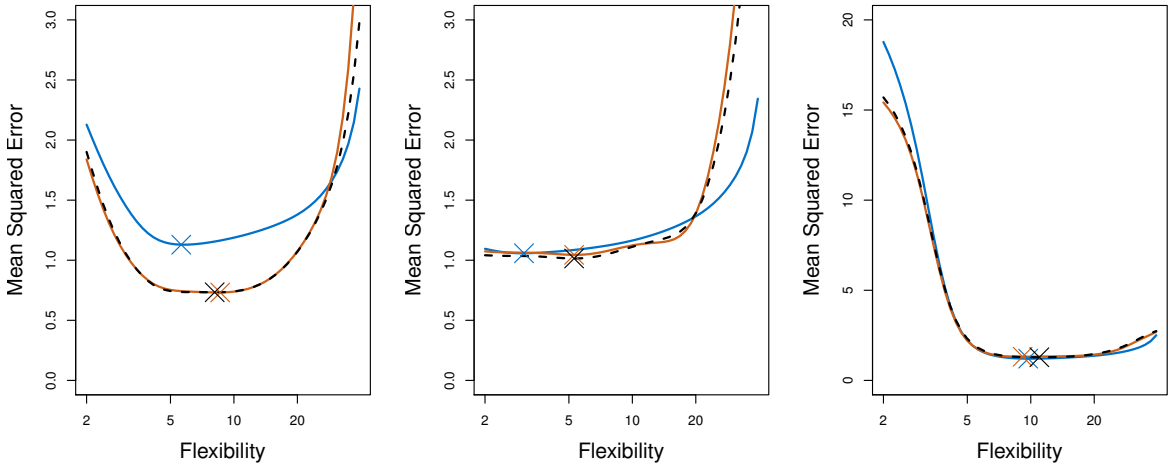
k-fold Cross Validation on Three Simulated Data Sets

Black: True Model
Orange: Linear Regression
Blue: Low Flexibility Spline
Green: High Flexibility Spline

From Chapter 2
Fig 2.9, 2.10, and 2.11



- Blue: True Test MSE
- Black-dashed: LOOCV MSE
- Orange: 10-fold MSE
- Refer to chapter 5 for the bottom graphs, Fig 5.6 page 182



5.1.4 Bias- Variance Trade-off for k-fold CV

- Putting aside that LOOCV is more computationally intensive than k-fold CV… Which is better LOOCV or k-fold CV?
  - LOOCV is less bias than k-fold CV (when $k < n$)
  - But, LOOCV has higher variance than k-fold CV (when $k < n$)
  - Thus, there is a trade-off between what to use
- Conclusion:
  - We tend to use k-fold CV with (k = 5 and k = 10)
  - It has been empirically shown that they yield test error rate estimates that suffer neither from excessively high bias, nor from very high variance

5.1.5 Cross Validation on Classification Problems

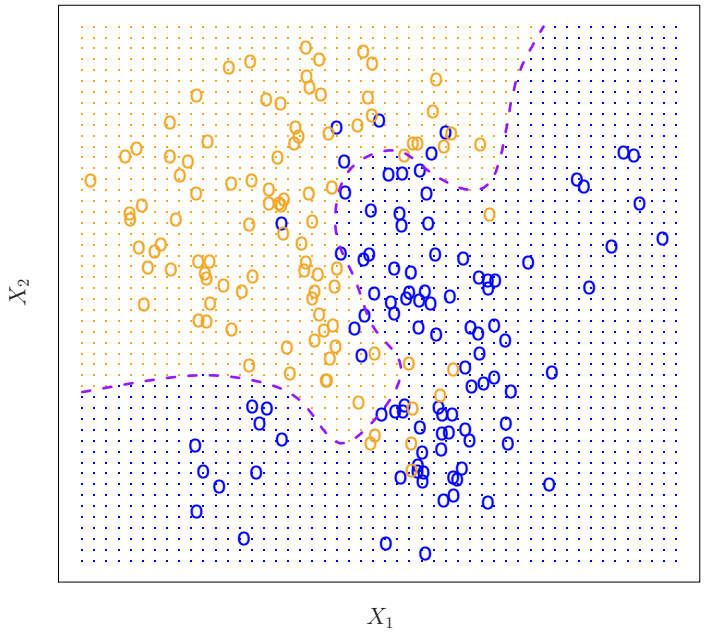
- So far, we have been dealing with CV on regression problems
- We can use cross validation in a classification situation in a similar manner
  - Divide data into k parts
  - Repeat k times:
    - Hold out one part, fit the model using the remaining data and compute the error rate on the hold out data
  - CV error rate is the average over the k error rates we have computed

Cross Validation in Practice

- CV can help estimate MSE or Classification Accuracy
- If we can estimate MSE before running the actual MSE how could we use it in our *model selection* process?
- What types of things could we decide about our Machine Learning modeling process?
- **MINUTE PAPER: Brainstorm and write down your specific uses for CV.**

Model Selection Example: Use CV to Select Classification Model

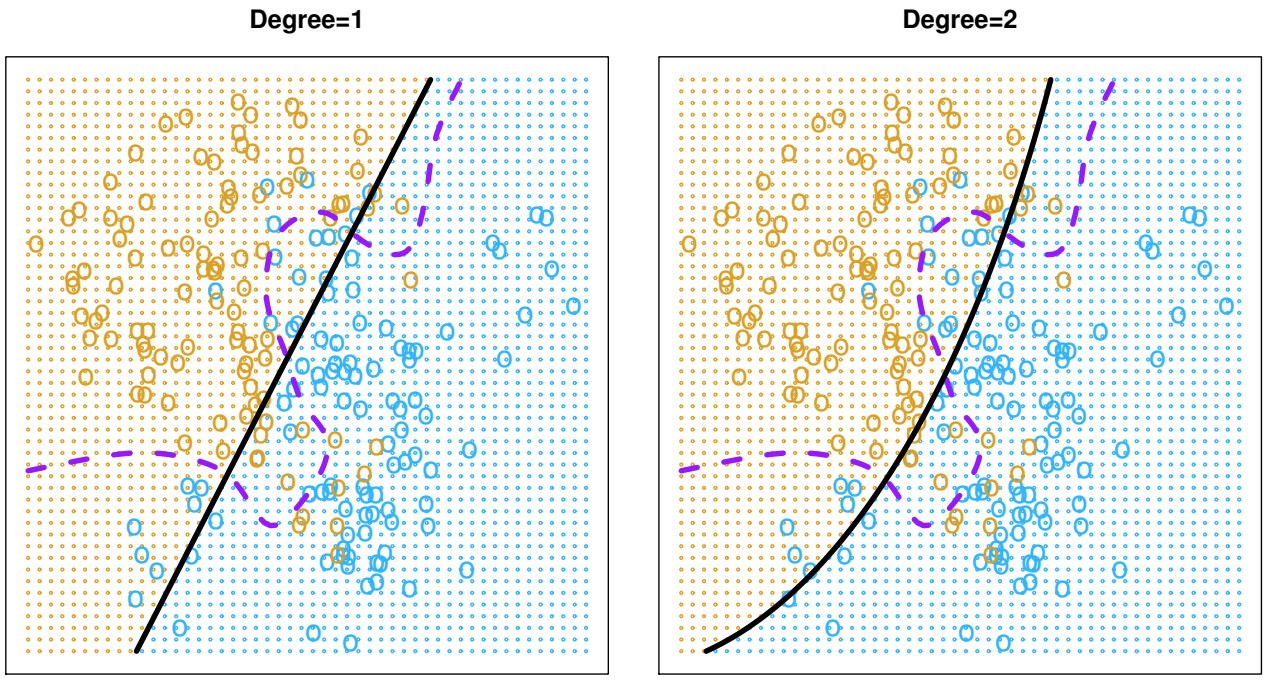
- Model Choice includes Logistic Regression (with terms of deg = 1, 2, …10) and KNN with many choices for K
- The data set used is simulated (refer to Fig 2.13)
- The purple dashed line is the Bayes' boundary



Bayes' Error Rate: 0.133

Use CV to Select Classification Model

- Linear Logistic regression (Degree 1) is not able to fit the Bayes' decision boundary
- Quadratic Logistic regression does better than linear

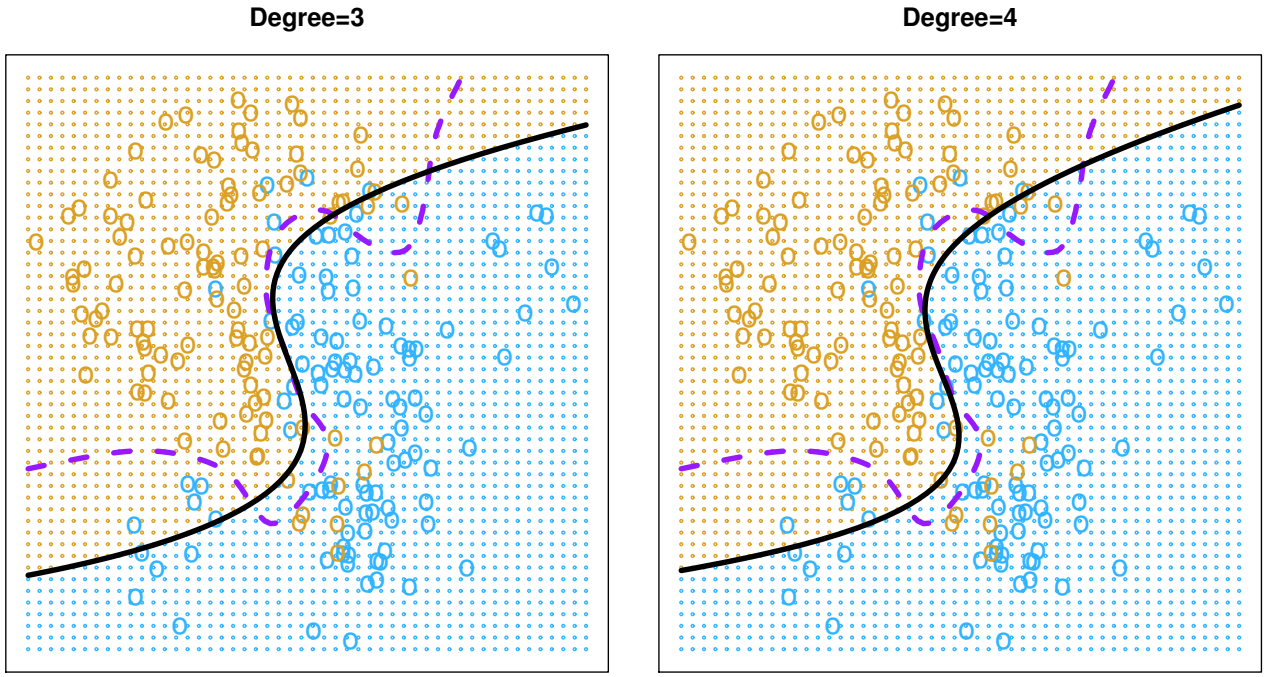


Error Rate: 0.201

Error Rate: 0.197

Use CV to Select Classification Model

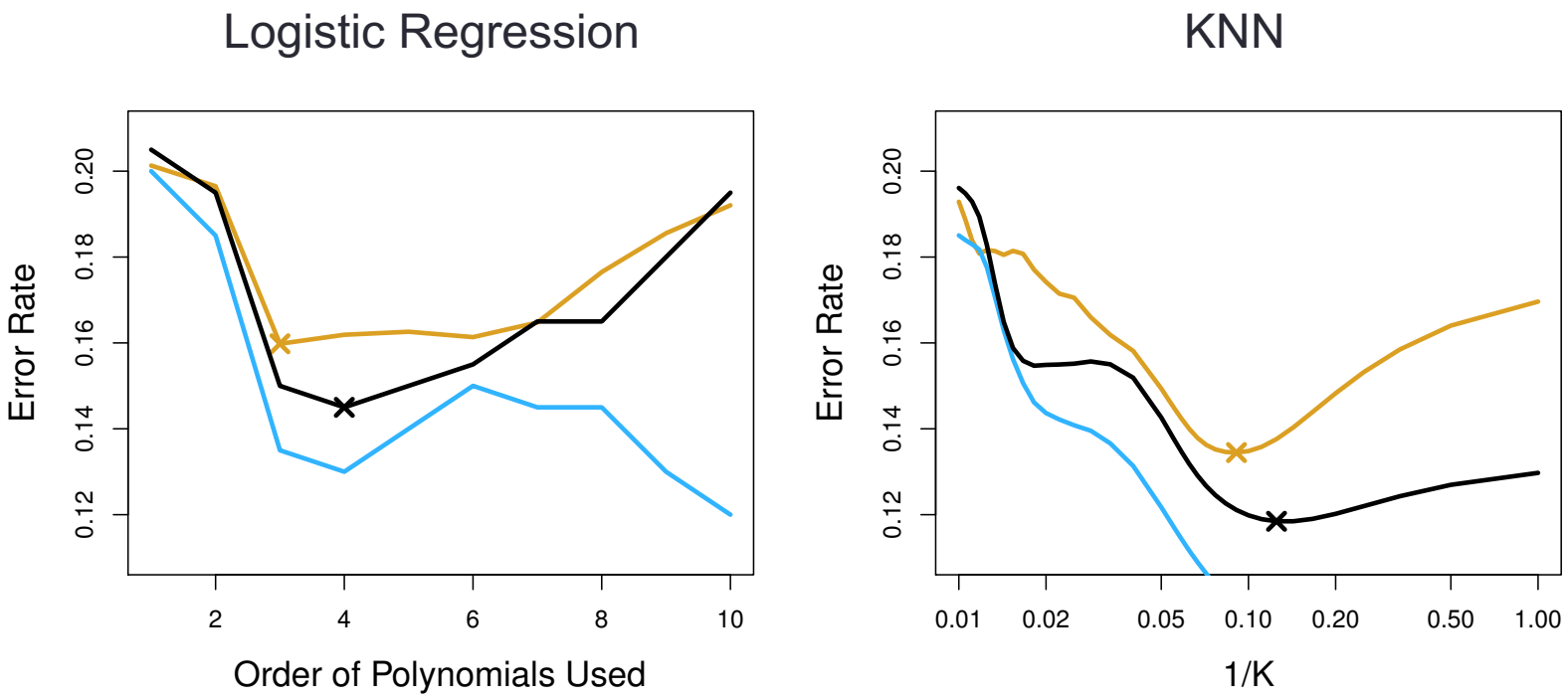
- Using cubic and quartic predictors, the accuracy of the model improves



Error Rate: 0.160

Error Rate: 0.162

Use CV to Select Classification Model



Blue: Training Error
Black: 10-fold CV Error
Brown: Test Error

Concept Check:
How do we interpret the results of these graphs?
What value polynomial should we choose?