METIS

Feature Engineering and Cross-Validation

INTRODUCTION TO DATA SCIENCE – FALL 2018

SESSION 7

AGENDA

1. Bias and variance tradeoff
2. Feature engineering
3. Cross validation

Introduction to Data Science

- Learning the steps in the Data Science Process
- Learning multiple model methodologies

Data Evaluation	Cross Validation	Advanced Error Metrics	Visualization
Deep Learning	Neural Networks	Forecasting or NLP -RNN	Computer Vision -CNN
Machine Learning	Decision trees and Ensembles	Time Series	Clustering
Statistical Learning	Linear Regression	Logistic Regression	PCA–dimensionality reduction
Data Prep	Data Cleaning	Feature Engineering	Visualization
Foundation	Statistics	Python	Linear Algebra

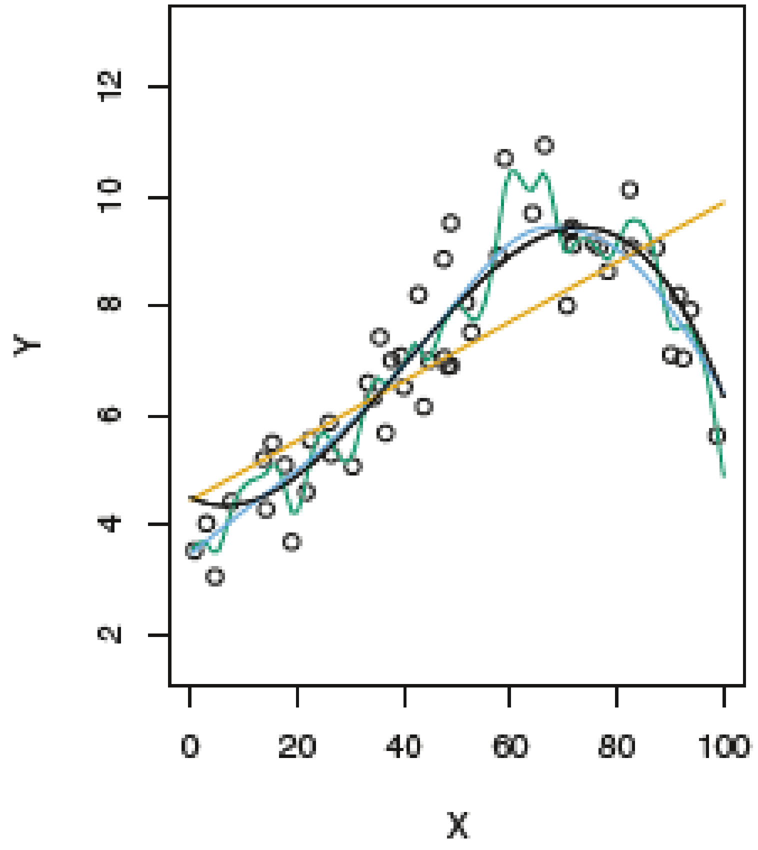
Bias and variance

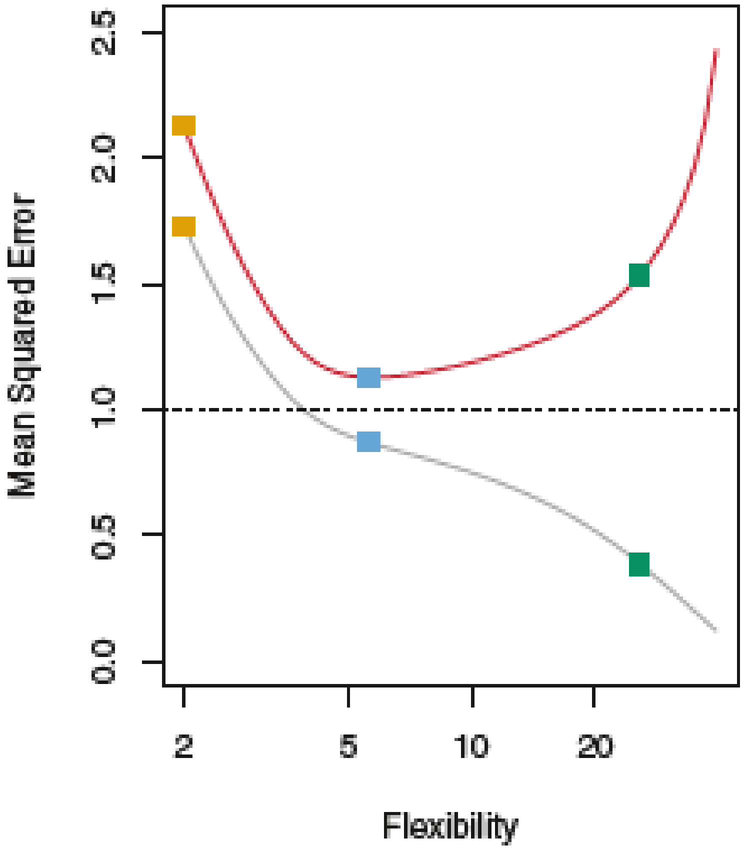
HOW DOES THE MODEL FIT THE DATA

Examining model fits

- Left: Three estimates of observed data

- Right:
    - MSE of training set (gray)
    - MSE of test set (red)
    - Colored squares represent MSE of fitted models on left

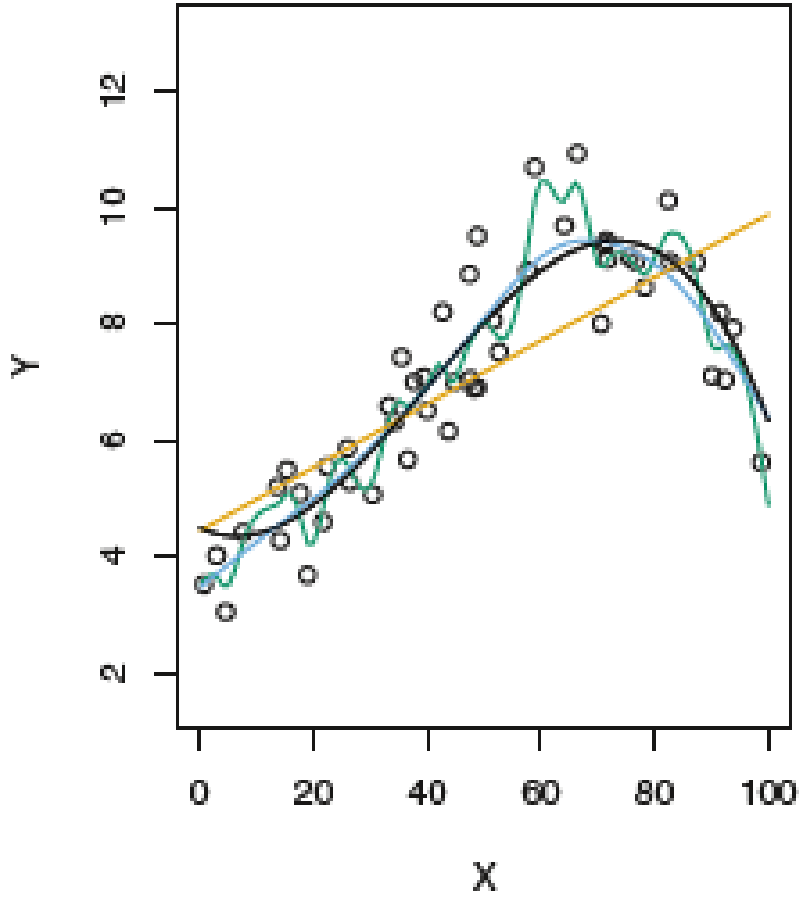




Model variance

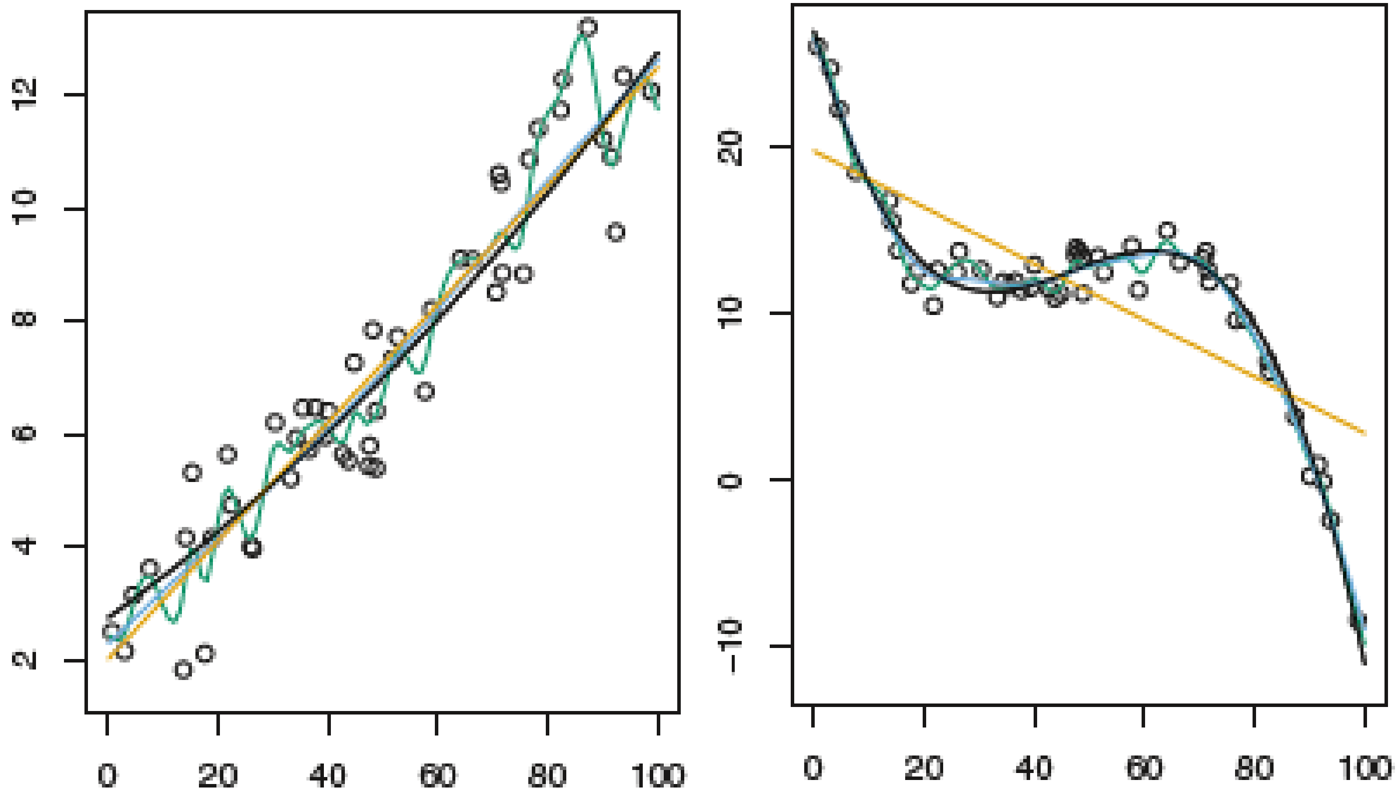
- Variance is the amount by which $\hat{f}$ would change if we estimated it using a different train set

  - Since different training data sets are used to fit our models, different sets result in different $\hat{f}$
  - If a method has a high variance, then small changes in data will cause large changes in $\hat{f}$
  - More flexible methods have higher variance
    - Green: High variance
    - Orange: Low variance



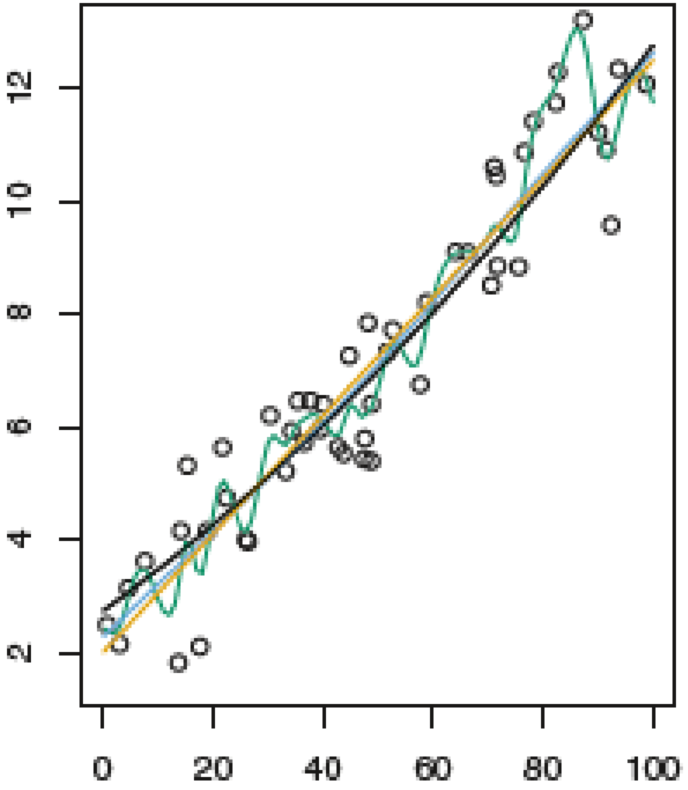
Model bias

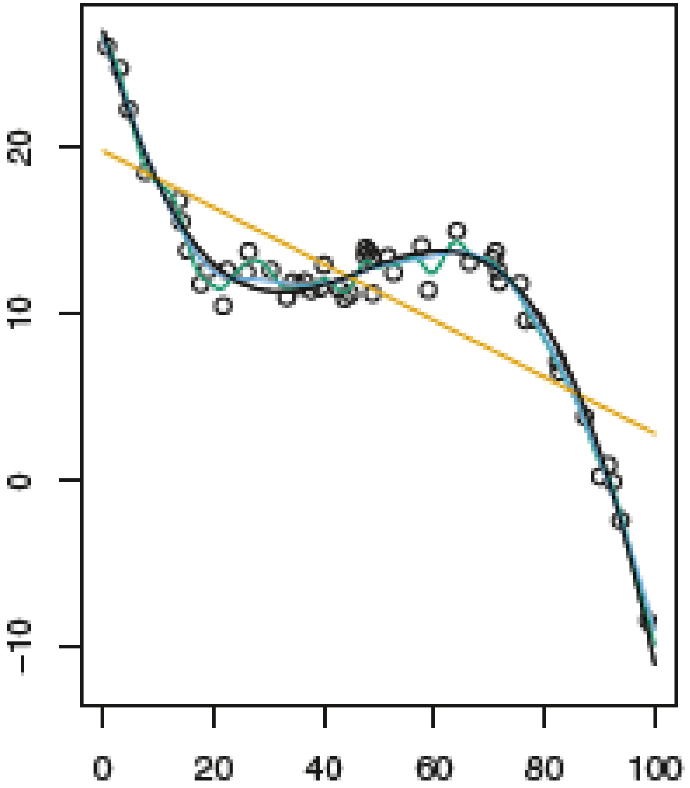
- Bias is the error introduced by solving a real-life problem with a simple model
  - No problem is truly linear
  - Linear **data:** Linear model has LOW bias
  - Non-linear **data:** Linear model has HIGH bias



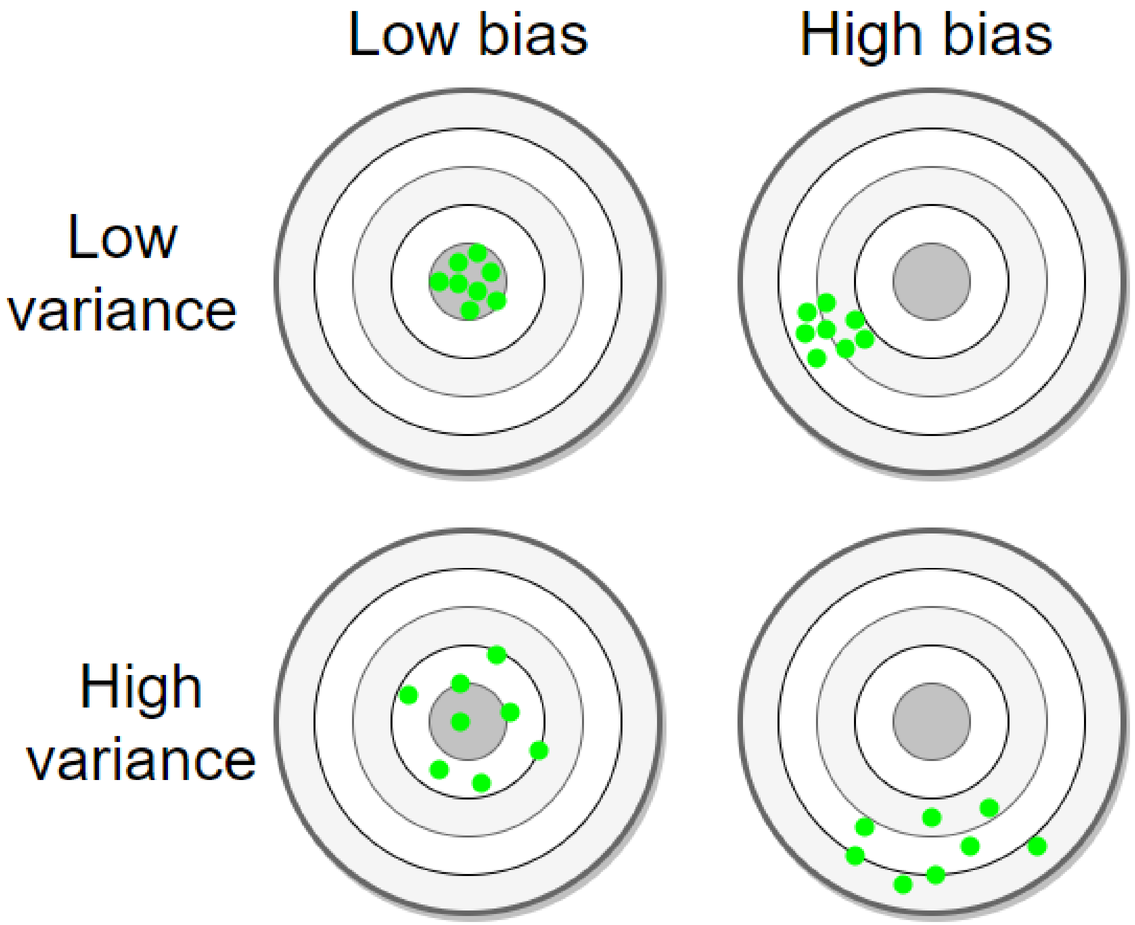
Model bias

- Bias is the error introduced by solving a real-life problem with a simple model
  - No problem is truly linear
  - Linear data: Linear model has LOW bias
  - Non-linear data: Linear model has HIGH bias





Another bias/variance visualization



Feature engineering

MODIFYING THE INDEPENDENT VARIABLES

What is feature engineering?

The process of creating features to make machine learning algorithms work more efficiently and accurately

This requires an understanding of your data and the algorithm

You have already been doing this

Movie ratings dataset from Pandas session

Reformatting timestamp and extracting features

	UserID	MovieID	Rating	FormattedTimestamp	day_of_month	year	month
0	1	1193	5	2000-12-31 22:12:40	31	2000	12
1	1	661	3	2000-12-31 22:35:09	31	2000	12
2	1	914	3	2000-12-31 22:32:48	31	2000	12
3	1	3408	4	2000-12-31 22:04:35	31	2000	12
4	1	2355	5	2001-01-06 23:38:11	6	2001	1

You have already been doing this

Concrete data set from the Tree and Forest session

Binning the responses into categorical data

compressive_strength__28-day_mpa	compressive_strength_bins	compressive_strength_bins_range
34.99	38	(33.726, 41.994]
41.14	38	(33.726, 41.994]
41.81	38	(33.726, 41.994]
42.08	46	(41.994, 50.262]
26.82	30	(25.458, 33.726]
25.21	21	(17.149, 25.458]

Today we add to our experience

- Standard scaling – mean and unit variance
- Min max scaling – results between 0 and 1
- Feature transformations – reduce skew in feature distributions
- Handling categorical features – using dummy variables
- Handling missing values - imputation

Which models require scaling?

Regularized regression

Linear classifiers

Principle Components Analysis

Clustering Methods

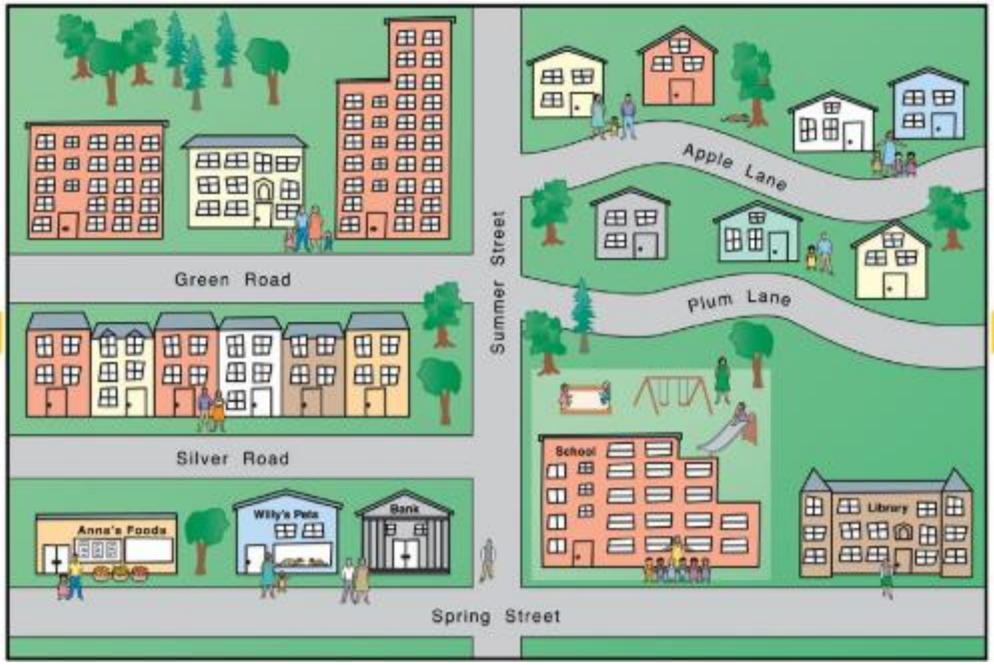
Decision Trees

Random Forest

Boosted trees

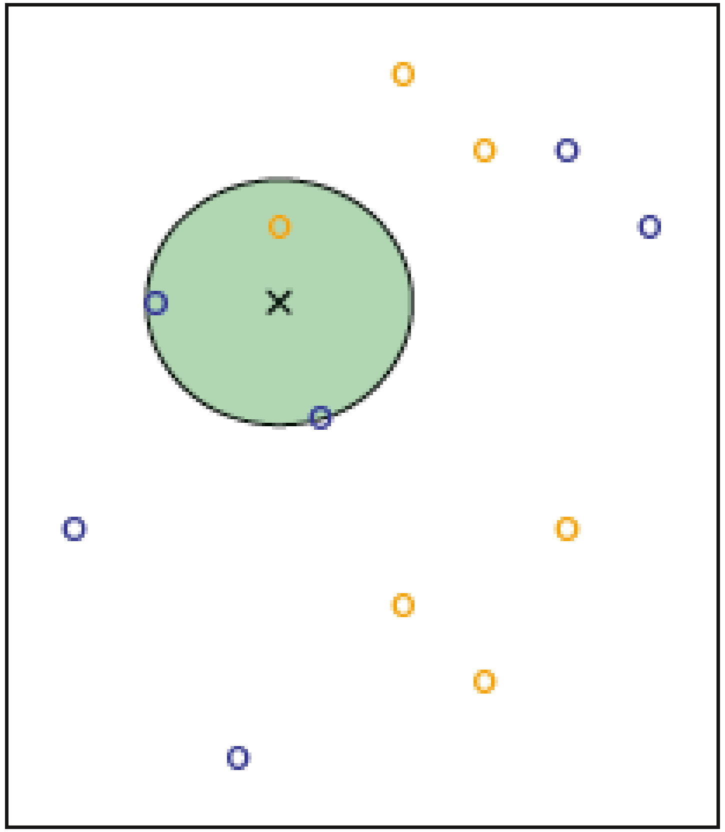
What is K nearest neighbors?

- Classification algorithm – discrete response
- Supervised learning – labeled response
- Non-parametric – no coefficients
- Instance-based – doesn’t hard code a model
- Distance metric – Euclidean distance
- Which K – lowest test error metric

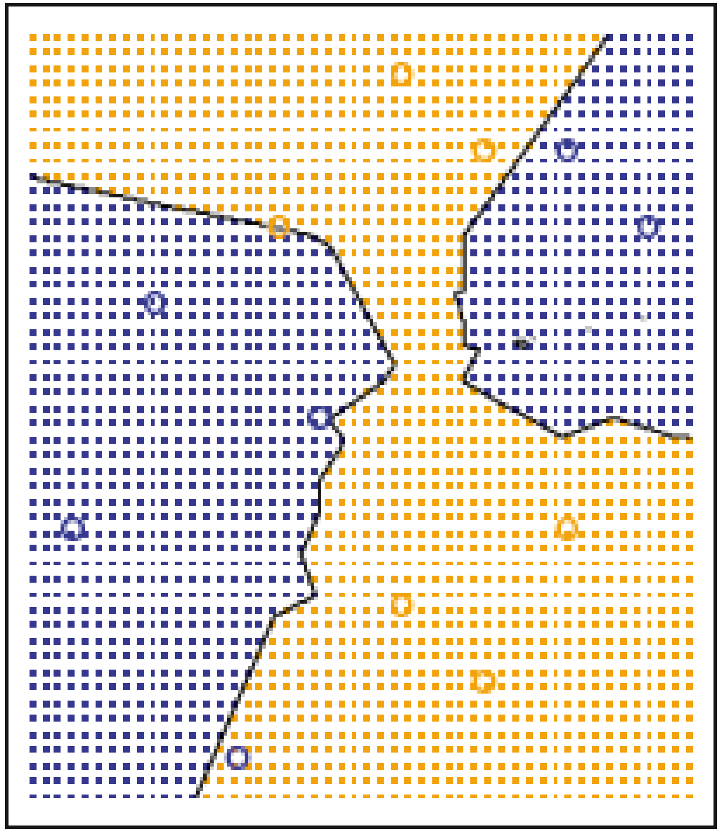


KNN approach where K = 3

- Classification algorithm
- Identify the K (3) **closest points**
- Calculate the probability of classes (blue or orange)
- Apply rule to classify the test observation (black x)



Training data set: 6 blue, 6 orange
Goal: Predict color of black x



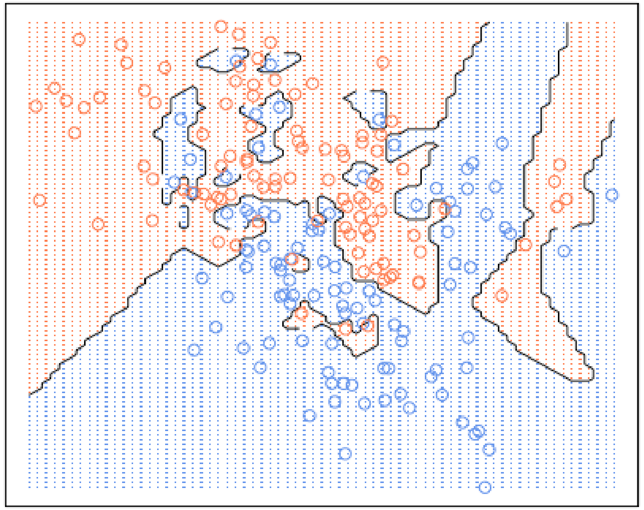
Results: Predictions for all possible values of our feature space (X1, X2)
Decision Boundary: Black line

K nearest neighbors

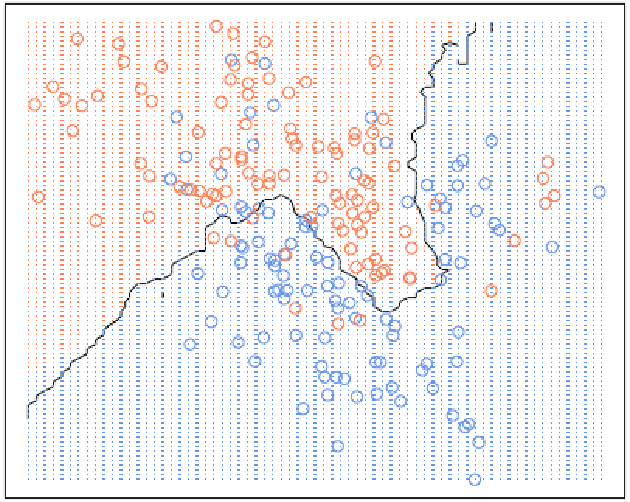
The choice of K has a drastic effect on the KNN classifier obtained

- When K =1, the decision boundary is overly flexible (low bias/high variance)
- What is the training error for K=1?
- As K grows, the boundary becomes less flexible and approaches a linear boundary (low variance / high bias)

nearest neighbour (k = 1)



20-nearest neighbour



KNN pros and cons

Simple to understand

Easy to implement

Works with multiclass or binary

Non-parametric is good for unusual data

Computationally expensive

Skewed class distributions in train/test

Accuracy can suffer with high-dimensional data

Cross-validation

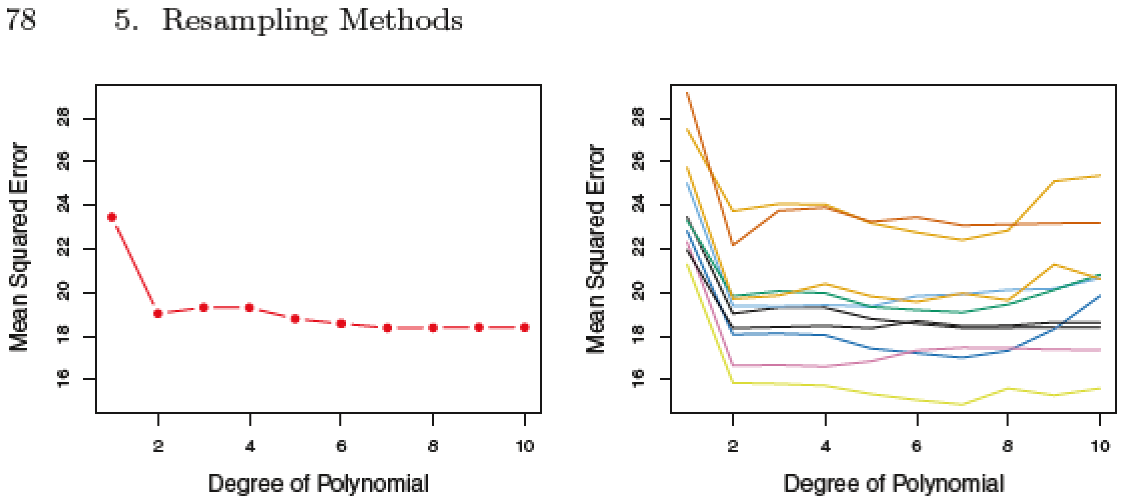
IMPROVING UPON TEST/TRAIN METHODOLOGY

Validation set approach

- Divide the entire set into two parts
- Fit the model on the training set
- Use that fitted model to predict on the test set
    - aka validation set, hold-out set
- Examine error metric for fitted model

Validation set approach

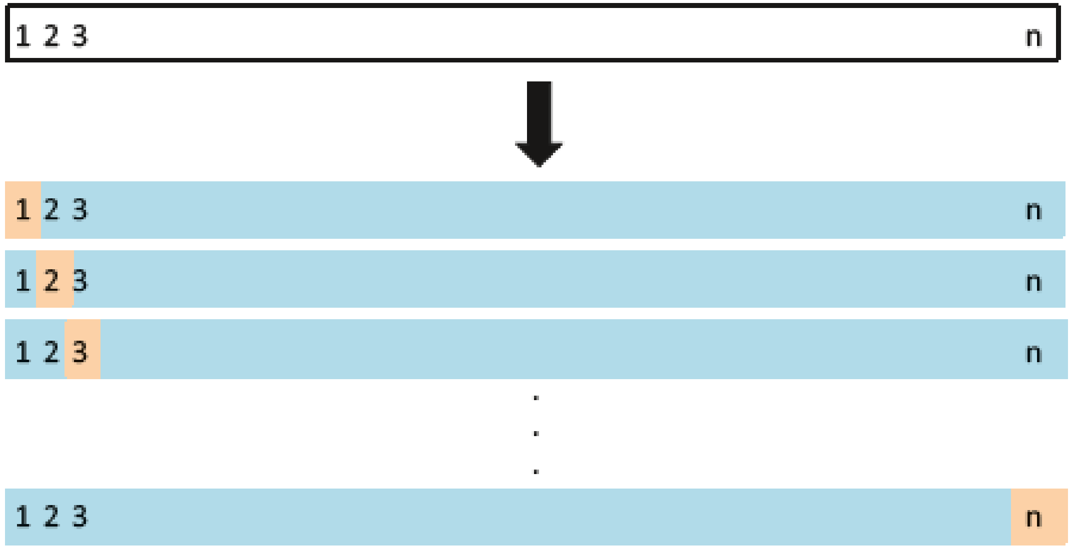
- Left: Validation (test set) error estimates for a singular (what we have been doing so far) split on training and test set
- Right test/train methodology was repeated ten times, each using a different random split.
- **Takeaway:** Simple train/test splitting can lead to varied MSE metrics



LOOCV - Leave-one-out cross-validation

- Split a data set with n observations into a test set of 1 and a train set of n-1
- Calculate the MSE
- Repeat procedure n times
- Aggregate the error metrics

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



LOOCV pros and cons

Less bias than validation approach

Always get the same results, no randomness in the training/validation set splits

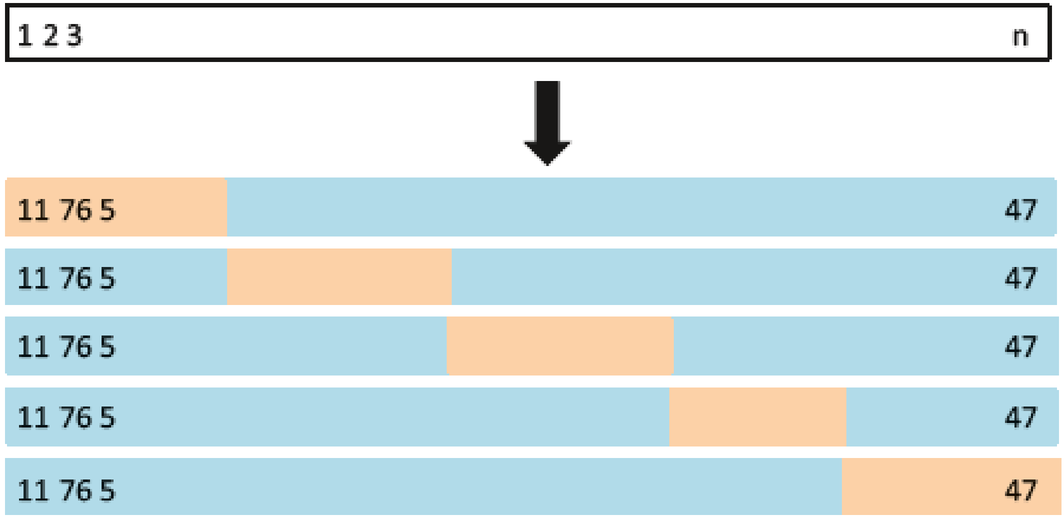
Can be used with any kind of models

Potentially expensive to implement based on n

K-fold cross-validation

- Randomly divide the set into k groups or folds of equal size
- The first fold is a hold out, method is fit on remaining k-1 folds
- MSE is calculated on the hold olut fold
- Repeat process k times and average MSEs

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



K-fold cross-validation pros and cons

Faster than LOOCV

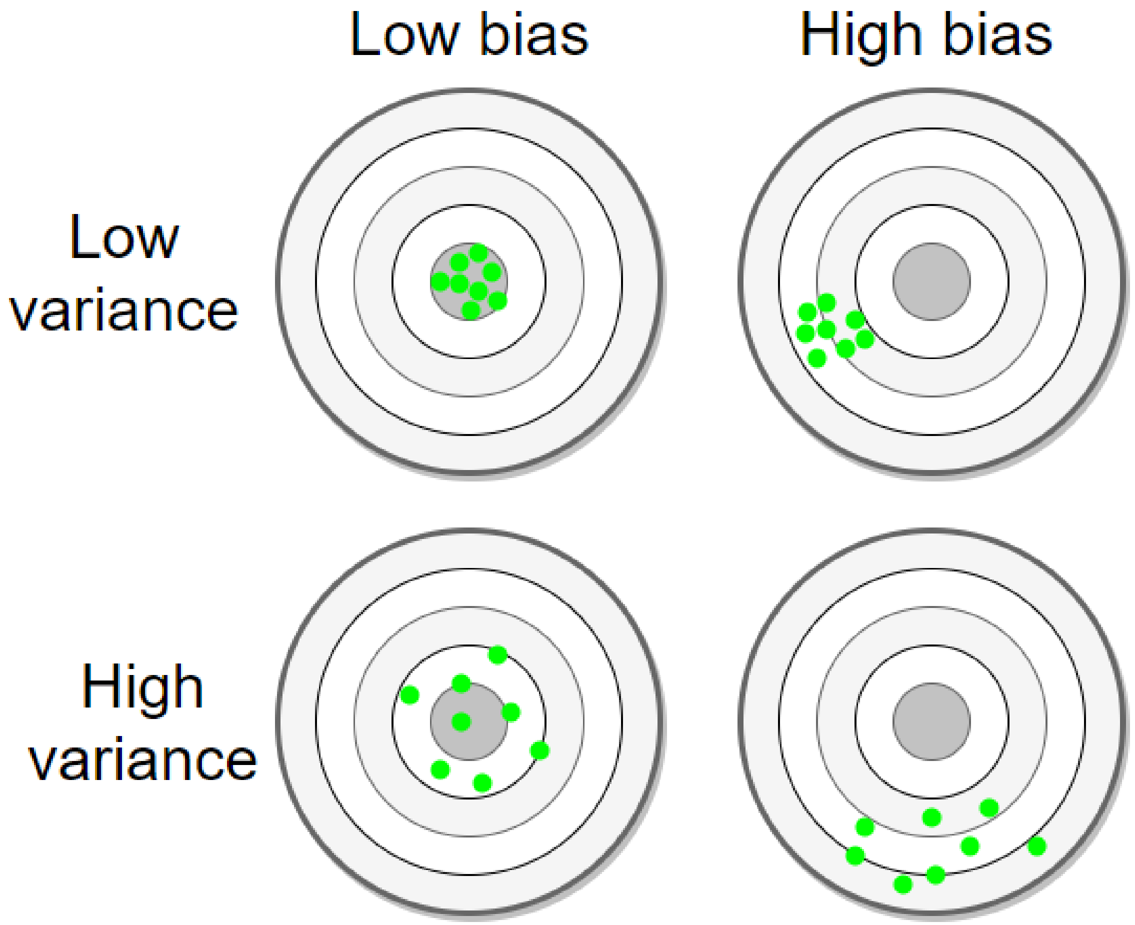
Variability in MSE is lower than validation approach

Use to select the “best fit” model

Variability in MSE is higher than LOOCV

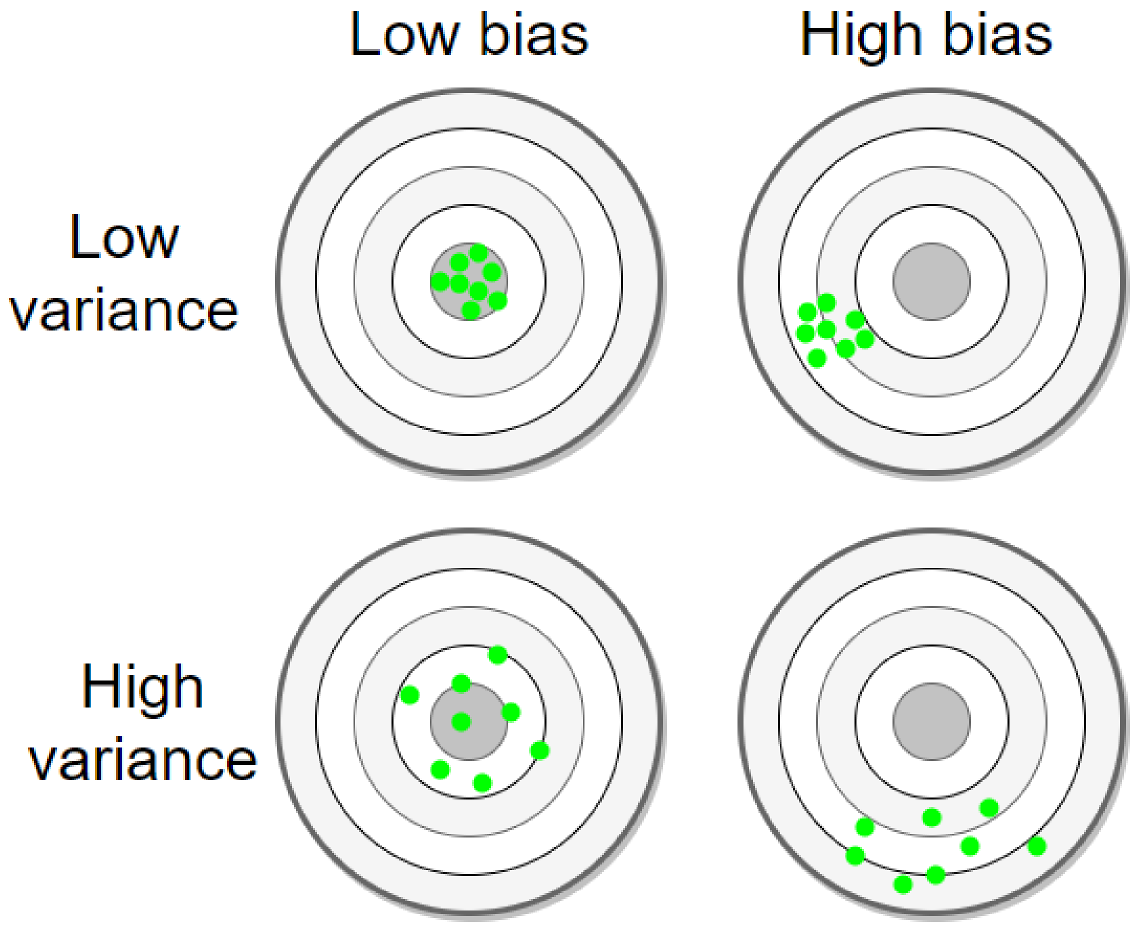
Bias-variance tradeoff for K-folds CV

- LOOCV gives unbiased predictions
- KFCV will give more biased predictions
- LOOCV gives a higher variance
- Why is this?



Bias-variance tradeoff for K-folds CV

- LOOCV gives unbiased predictions
- KFCV will give more biased predictions
- LOOCV gives a higher variance
  - Each of the n fitted models are trained on an almost identical set of data
- KFCV is fit with less correlated sets
- The mean of many highly correlated quantities has **higher variance** than the man of many quantities that are not as highly correlated
- K=5 or k=10 have been shown to yield test error estimate rates that don't suffer from very high bias nor from very high variance



Appendix

Transformations

MODIFYING THE DEPENDENT VARIABLE