BIOS 635: Dimensionality and Assessing Model Accuracy

Kevin Donovan

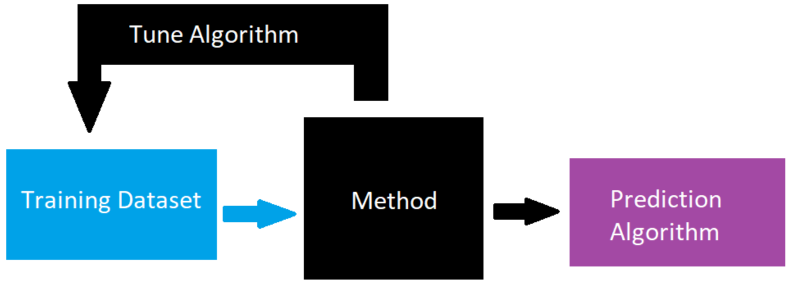
1/21/2021

Review

- First lecture on 1/19
- GitHub accounts created, usernames shared with me
- Course syllabus, schedule, and first recorded lecture posted to Sakai
- Homework 1 assigned, due on 1/27 through GitHub Classroom

Prediction Model

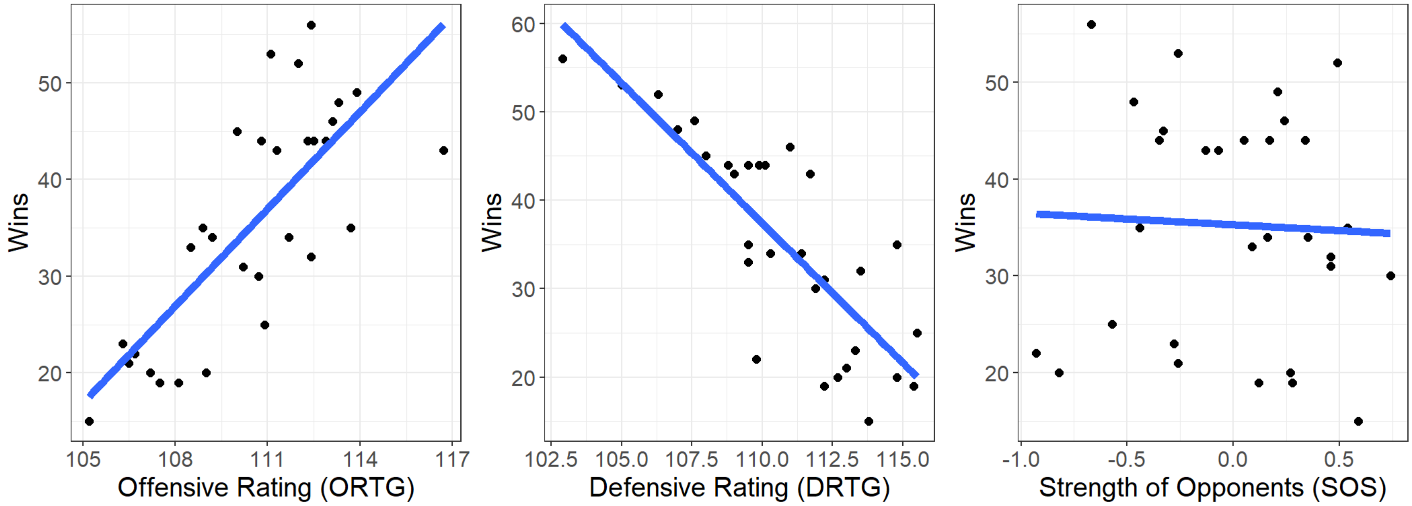
Recall: Pattern recognition algorithm "trained" using observed dataset



Let's formalize this idea using math notation

Prediction Model

Example: Consider data on basketball teams (NBA)



Consider: Predict teams wins using three variables

First, visualize data with line of best fit

Prediction Model

Let's predict wins using all three variables simultaneously

Model: $wins \approx f(ORTG, DRTG, SOS)$

Notation: $wins$ is the variable we want to predict $\equiv$ *response*

$ORTG, DRTG, SOS$ are variables used to predict response $\equiv$ *feature* or *predictor*

Response denoted mathematically by variable Y

Features denoted by variables $X_1, X_2, \ldots, X_p$

Prediction Model

Combining everything together

Can denote set of features as vector:

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_p \end{pmatrix} = \begin{pmatrix} ORTG \\ DRTG \\ SOS \end{pmatrix}$$

With model denoted by

$$Y = f(X) + \epsilon$$

where ϵ denotes model *error*

Prediction Model

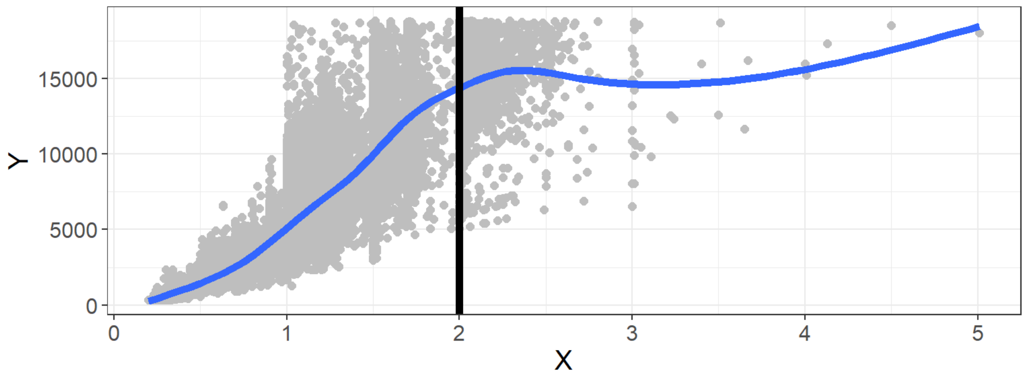
How to use model f:

- For $X = x$, can predict Y
- Which variables are “important” in predicting $Y \equiv wins$?
- Which are not important in the prediction?
- How is each variable in X associated with Y?

Prediction Model

Often, we define model as

$$f(X) = E(Y|X = x)$$



Predict outcome based on *expected value* (mean) at specific feature value(s)

$$f(2) = E(Y|X = 2)$$

Model Evaluation

How to evaluate model f?

Supposed we are in **supervised learning** context:

One measure of accuracy: *mean squared error* (MSE)

$MSE(x) = E[(Y - f(X))^2|X = x]$ across all possible x

Ideal or *optimal* f which minimizes $MSE(x)$ across all x

Model Evaluation

For given model f, *residual* at $X=x$ for Y is

$$\epsilon = Y - f(x)$$

Cannot zero out residual for all cases due to variability around mean at $X=x$

Thus, referred to as *irreducible* error

Goal: if $f = E(Y|X=x)$ want to estimate it with model **as close as possible**

Estimate denoted by $\hat{f}(x)$

MSE for estimate at $X=x$ can be decomposed into

$$MSE_{\hat{f}}(x) = E[(Y - \hat{f}(X))^2|X=x] = [f(x) - \hat{f}(x)]^2 + Var(\epsilon)$$

Model Evaluation

Goal: find best estimate of $f(x)$ using, estimate $\hat{f}(x)$, define as prediction model

Can see best estimate **minimizes difference with true mean**

How to estimate?

- Line of best fit
- Non-linear best fit

Model Estimation

Nearest Neighbor: For given $X = x$ estimate expected Y based on observed values of Y near x in data

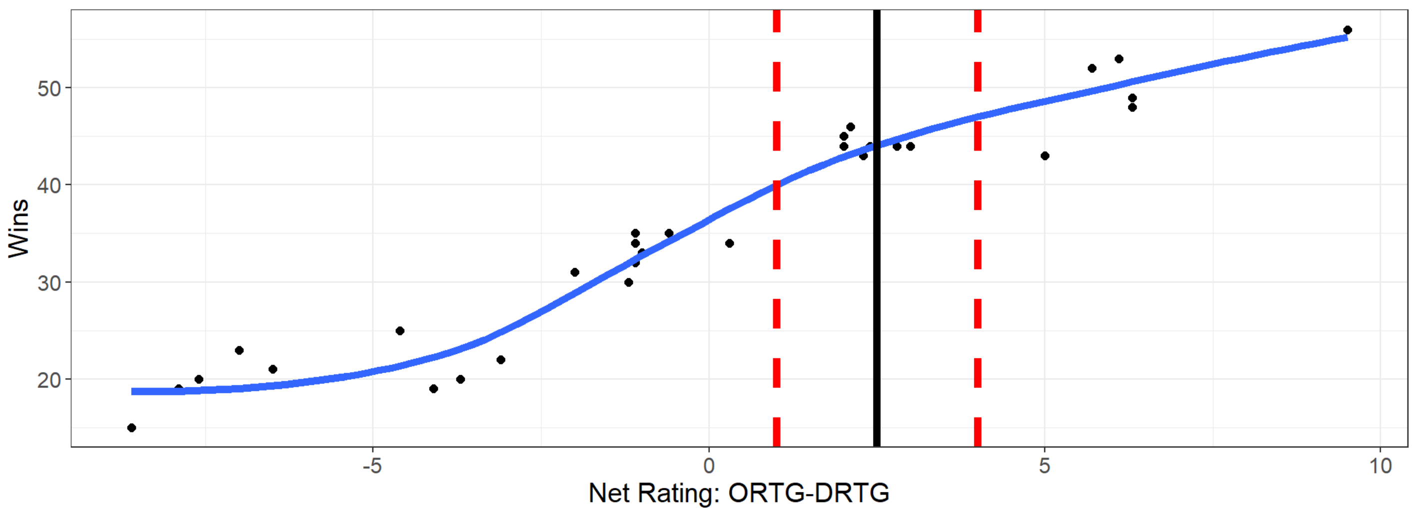
Cannot estimate based on values of Y **at** x since amount of data there likely small

Mathematically: $\hat{f}(x) = \text{Ave}[Y|X \in \delta(x)]$

where Ave denotes a weighted average

$\delta(x)$ is some neighborhood around x

Ex. Predicting wins using net rating $(ORTG - DRTG)$



Curse of Dimensionality

Idea: Higher dimensional models (more features) may have **worse** performance then smaller models

Why?: Higher dimension $\implies$ higher prediction variability

May result in *overfitting* to training data

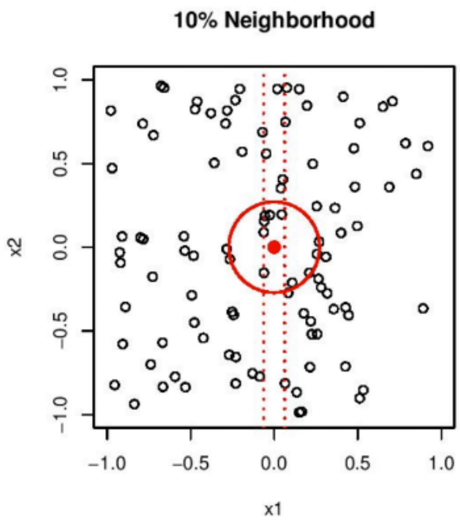
Ex. with nearest neighbor

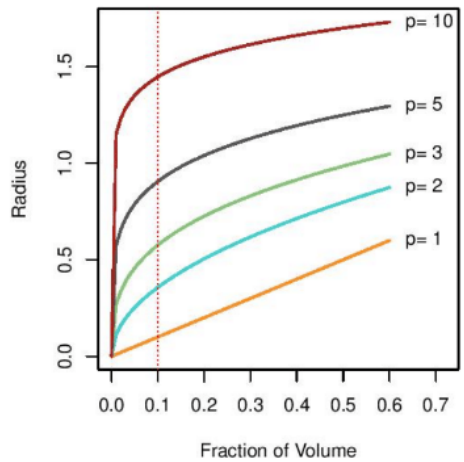
Need large enough $\delta(x)$ to have a good, stable estimate

Too large $\implies$ estimate is inaccurate, **lose benefit** of local averaging

Many features/high dimension $\implies$ neighbors tend to be far away

Curse of Dimensionality





Parametric Modeling

Simpler estimate of f: linear regression model

$f_L(X) \approx \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

where we estimate f_L best estimating $\beta_0, \ldots, \beta_1$ using *line of best fit*

$\hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \ldots + \hat{\beta}_p X_p$

Ex. predicting wins in NBA data using $ORTG, DRTG, SOS$

Mean Squared Error = 9.48

Mean Absolute Error = 2.63

Feature Parameter	Estimate
(Intercept)	35.3
ORtg	2.3
DRtg	-2.5
SOS	2.3

Parametric vs Nonparametric

- **Parametric Models**:

  Algorithm based on a **finite set** of parameters.

  Often algorithm is based on specific functional form (ex. linear)

- **Nonparametric Models**:

  Algorithm based on a **infinite set** of parameters.

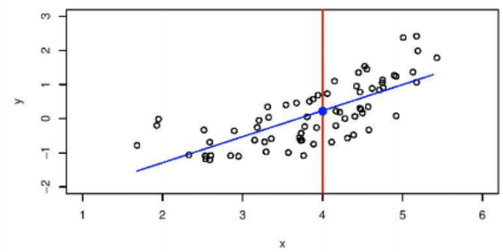
  Algorithm generally more flexible, data-driven, **but**

  Has higher variance, more difficult interpretation, more prone to overfitting

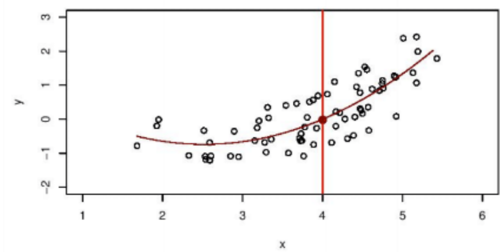
Parametric Functional Form

In two-dimensions:

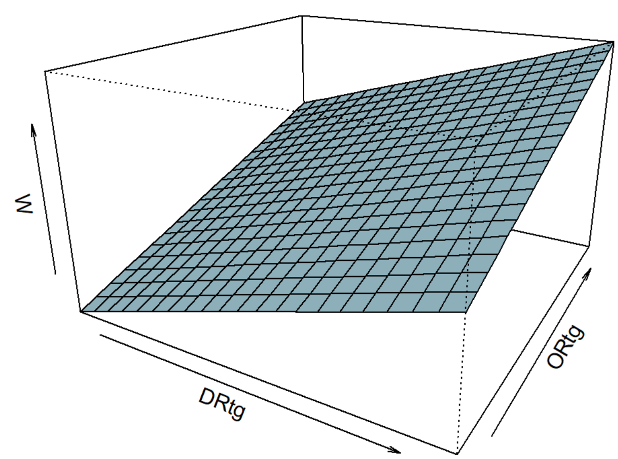
A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit here



A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.



Parameteric Functional Form



In three-dimensions:

Assessing Model Accuracy

Denote training data by $Tr = \{x_i, y_i\}_1^N$

Denote *independent dataset* by $Te = \{x_i, y_i\}_1^M$ as *testing data*

- Could generate and evaluate model using training set, with MSE

$$MSE = \mathrm{Ave}_{i \in Tr}[(y_i - f(\hat{x}_i))^2]$$

- Could generate model using training set then evaluate using testing set

$$MSE = \mathrm{Ave}_{i \in Te}[(y_i - f(\hat{x}_i))^2]$$

Assessing Model Accuracy

Denote training data by $Tr = \{x_i, y_i\}_1^N$

Denote *independent dataset* by $Te = \{x_i, y_i\}_1^M$ as *testing data*

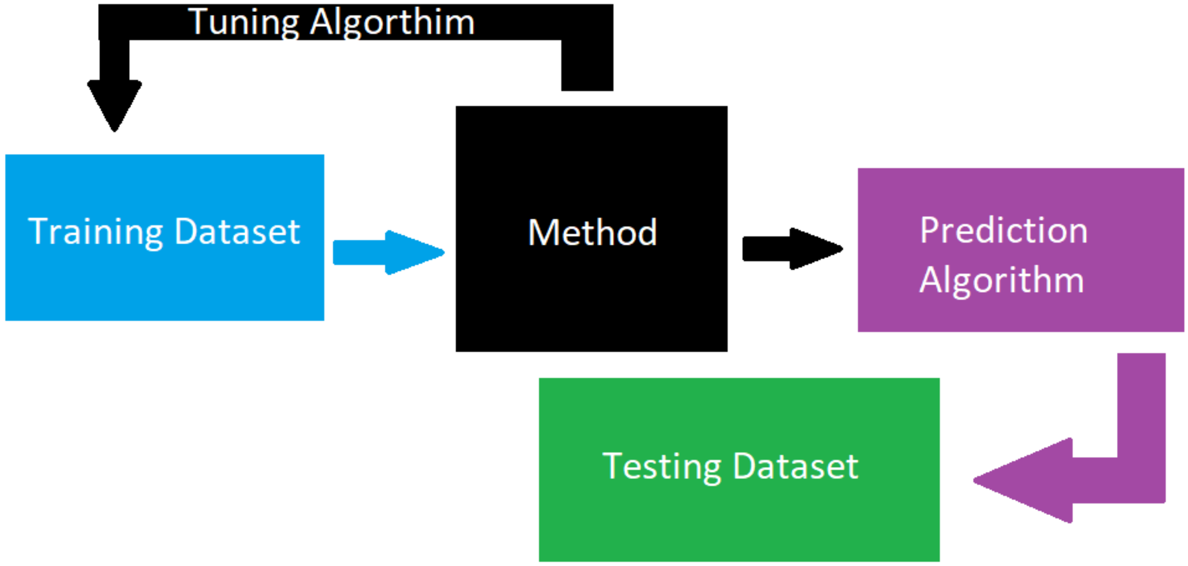
- Could generate and evaluate model using training set, with MSE

$MSE = \text{Ave}_{i \in Tr}[(y_i - f(\hat{x_i}))^2]$

- Could generate model using training set then evaluate using testing set

$MSE = \text{Ave}_{i \in Te}[(y_i - f(\hat{x_i}))^2]$

Assessing Model Accuracy



Testing vs Training Data

- Training Data: Used to build model **only**
- Testing Data: Used to evaluate model **only**
- Why use separate testing data?
  - *Algorithm may **overfit** to training data*
  - *Biased reflection of performance to general samples (generalization)*
  - *Performance on test data better indicator of generalization performance*

Testing vs Training Data

Ex. Predicting NBA team wins

Recall: Evaluating on training set

```
## [1] "MSE = 9.48"
```

```
## [1] "Mean Absolute Error = 2.63"
```

Now, let's **randomly split** the data into two equal sized sets

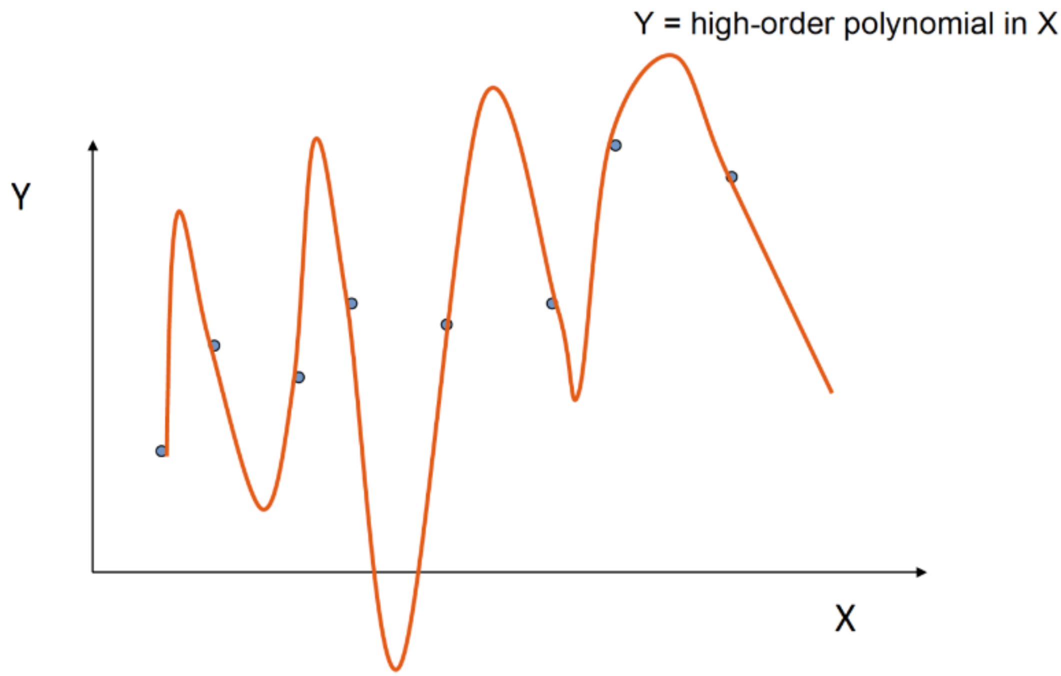
```
## [1] "Training Set; MSE = 9.21"
```

```
## [1] "Training Set; Mean Absolute Error = 2.43"
```

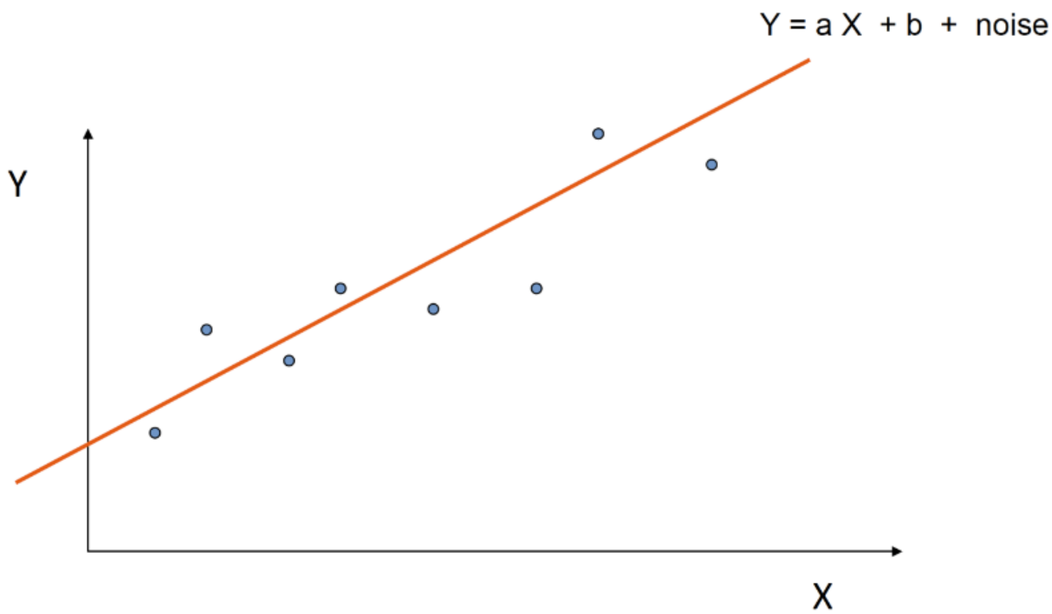
```
## [1] "Testing Set; MSE = 13.44"
```

```
## [1] "Testing Set; Mean Absolute Error = 2.73"
```

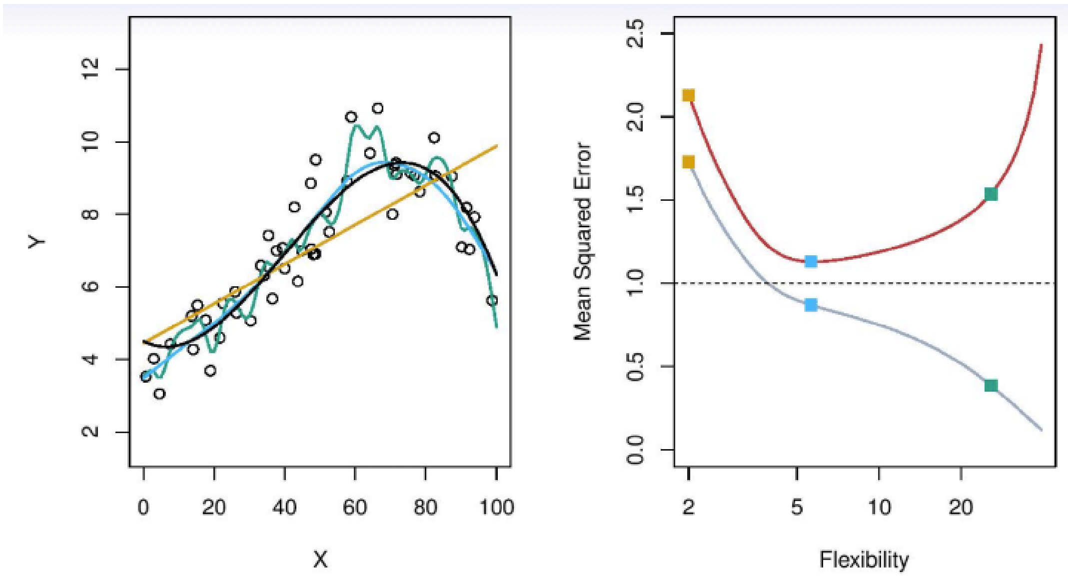
Overfitting (Complex Model)



Overfitting (Simple Model)

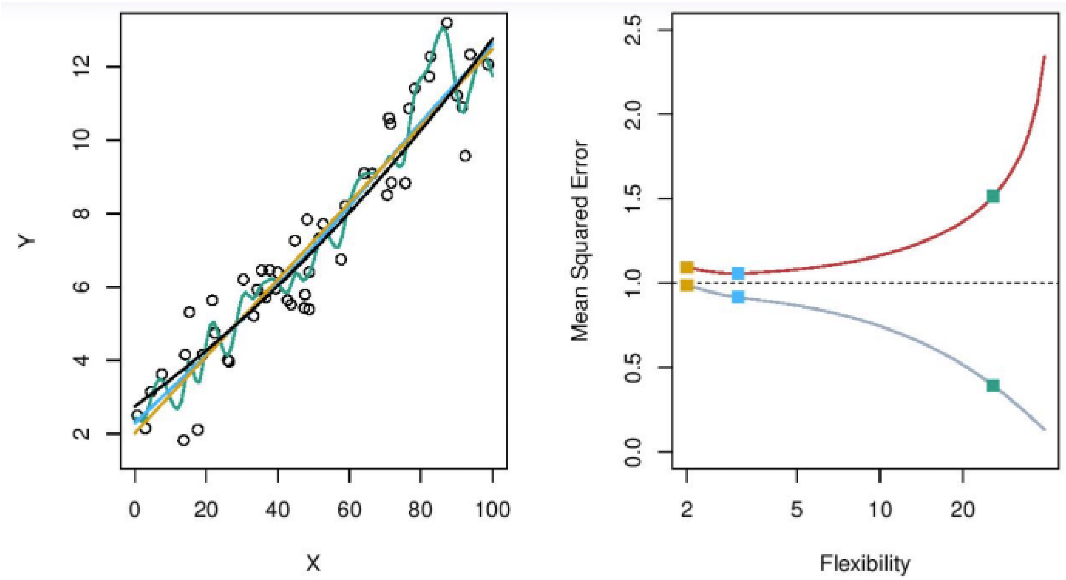


Testing vs Training Performance



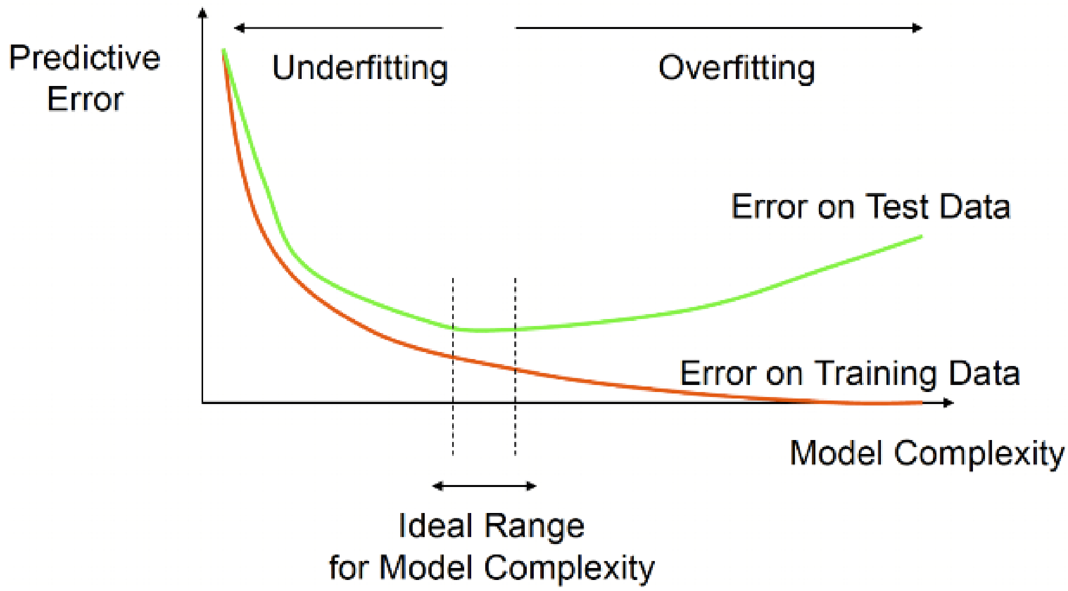
Black curve is truth. Red curve on right is $\mathrm{MSE}_{\mathsf{Te}}$, grey curve is $\mathrm{MSE}_{\mathsf{Tr}}$. Orange, blue and green curves/squares correspond to fits of different flexibility.

Testing vs Training Performance



Here the truth is smoother, so the smoother fit and linear model do really well.

Testing vs Training Performance



Song of the Session

Lovers Rock directed by Steve McQueen

Silly Games by Janet Key

Hello Stranger by Brown Sugar

I'm in Love With a Dreadlocks by Brown Sugar

Lovers Rock by The Clash