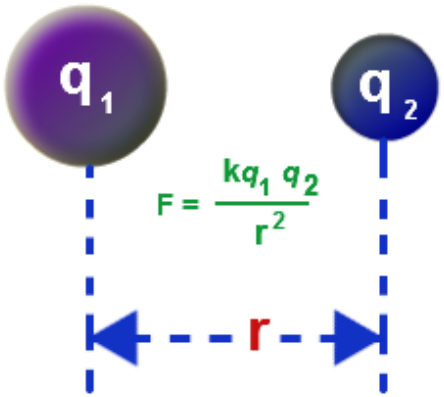
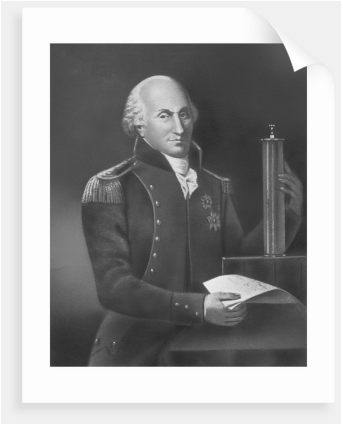
Introduction to Machine Learning

Yet Looking for Tools to Model

- We have all seen simple physics equations, discovered hundreds of years ago
- Since then we have been curious about how systems work and to predict their behavior before constructing them





- Back then, many derivations were based on experiments
- For instance deriving an equation like $F = 8.99 \times 10^9 \frac{q_1 q_2}{r^2}$ could involve fixing all the parameters except one and evaluating the response behavior
- In this simple example q_1, q_2, r are variables and F is a response

Yet Looking for Tools to Model

- What has changed?
  - Back then, we did not have accurate measurement systems but now we have very accurate measurement sensors for pretty much all physical phenomena
  - Back then, we were only able to collect limited data, now we are able to collect huge data
  - Back then, we did not have computational resources, now we have supercomputers
  - Back then, we alway liked more compact models for an easier application, now we can have super-complex models and ask the computers to apply them
- **Then why not model large and complex systems??!!**

Yet Looking for Tools to Model

– Models that can predict the weather, the stock market value, political interactions, ...

– In our century, having more data means more power to predict!

More Examples: Spam Detection

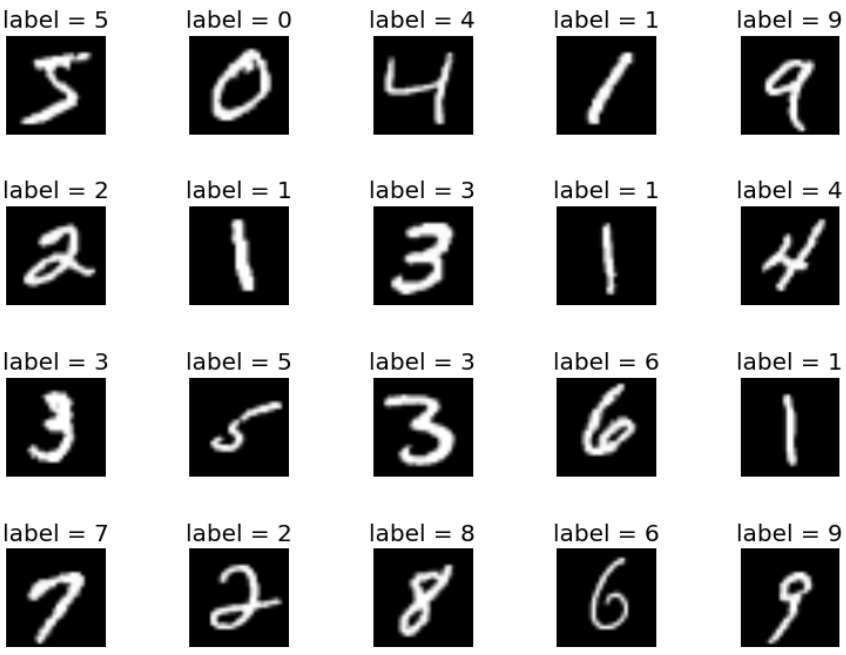
- data from 4601 emails sent to an individual (named George, at HP labs, before 2000). Each is labeled as spam or email
- goal: build a customized spam filter
- input features: relative frequencies of 57 of the most commonly occurring words and punctuation marks in these email messages

	george	you	hp	free	!	edu	remove
spam	0.00	2.26	0.02	0.52	0.51	0.01	0.28
email	1.27	1.27	0.90	0.07	0.11	0.29	0.01

Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between **spam** and **email**

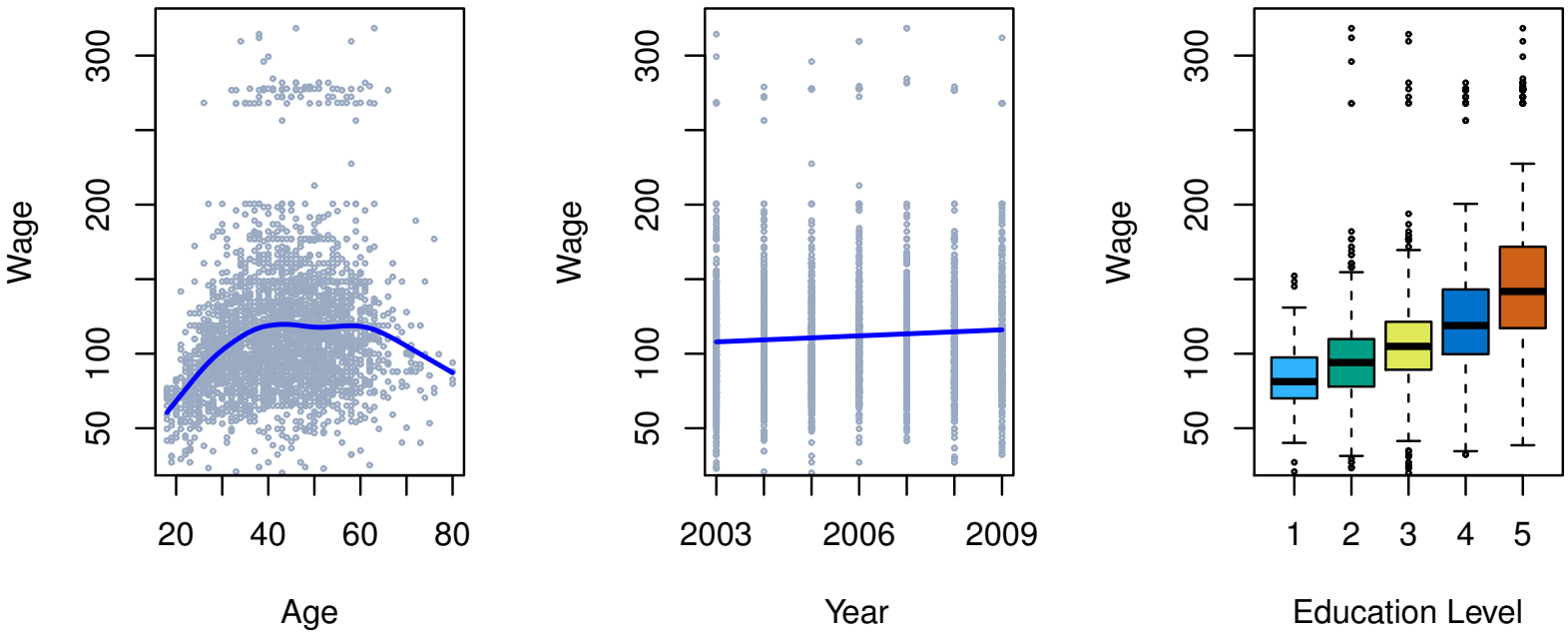
More Examples: Optical Character Recognition

identify hand-written characters based on the combination of all pixel values



More Examples: Salary Prediction

– Establish the relationship between salary and demographic variables in population survey data

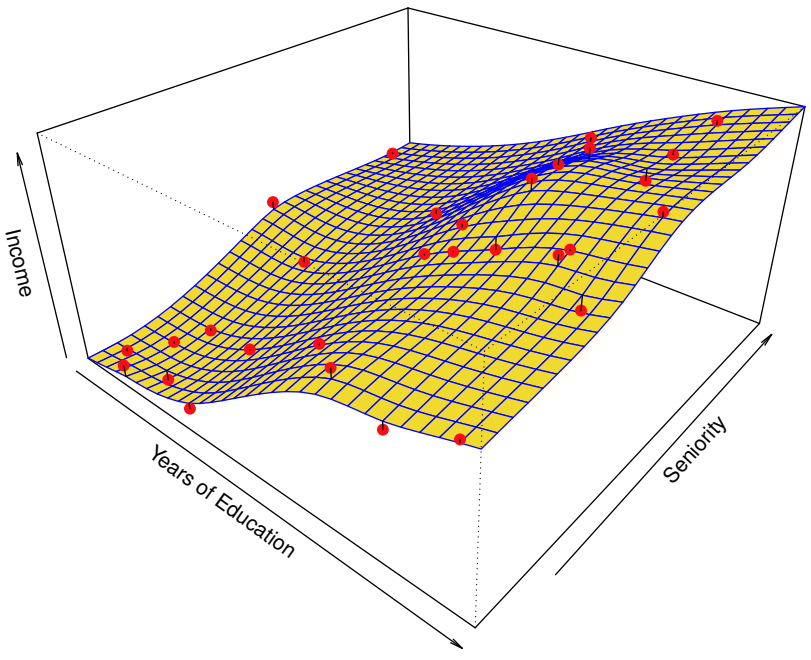


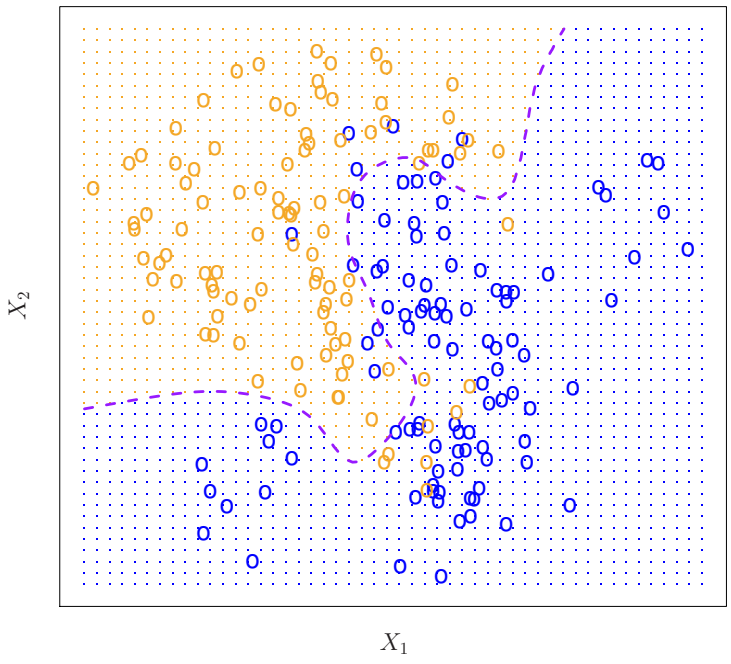
Supervised Learning

- Recall the equation $F = 8.99 \times 10^9 \frac{q_1 q_2}{r^2}$ or more generally $Y = f(\boldsymbol{X})$ where $\boldsymbol{X} = (X_1, \ldots, X_p)^\top$
- In the case above, $Y = F$ and $\boldsymbol{X} = (q_1, q_2, r)^\top$ features
- Outcome measurement Y (also called dependent variable, response, target)
- Vector of p predictor measurements $\boldsymbol{X}$ (also called inputs, regressors, covariates, features, independent variables)
- In the **regression** problem, Y is quantitative (e.g price, blood pressure)
- In the **classification** problem, Y takes values in a finite, unordered set (survived/died, digit 0-9, cancer class of tissue sample) Discrete Number
- We have training data $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_N, Y_N)$. These are observations (examples, instances) of these measurements

  [**Bold** faces correspond to vectors]

Examples of Regression and Classification





Supervised Learning

On the basis of the training data we would like to:

- Accurately predict unseen test cases
- Understand which inputs affect the outcome, and how
- Assess the quality of our predictions and inferences

– Normally from the available examples $(\boldsymbol{X}_1, Y_1), ..., (\boldsymbol{X}_N, Y_N)$ we choose N_1 of them (around 80%) for training and $N - N_1$ of them for the later evaluation of the model (testing)

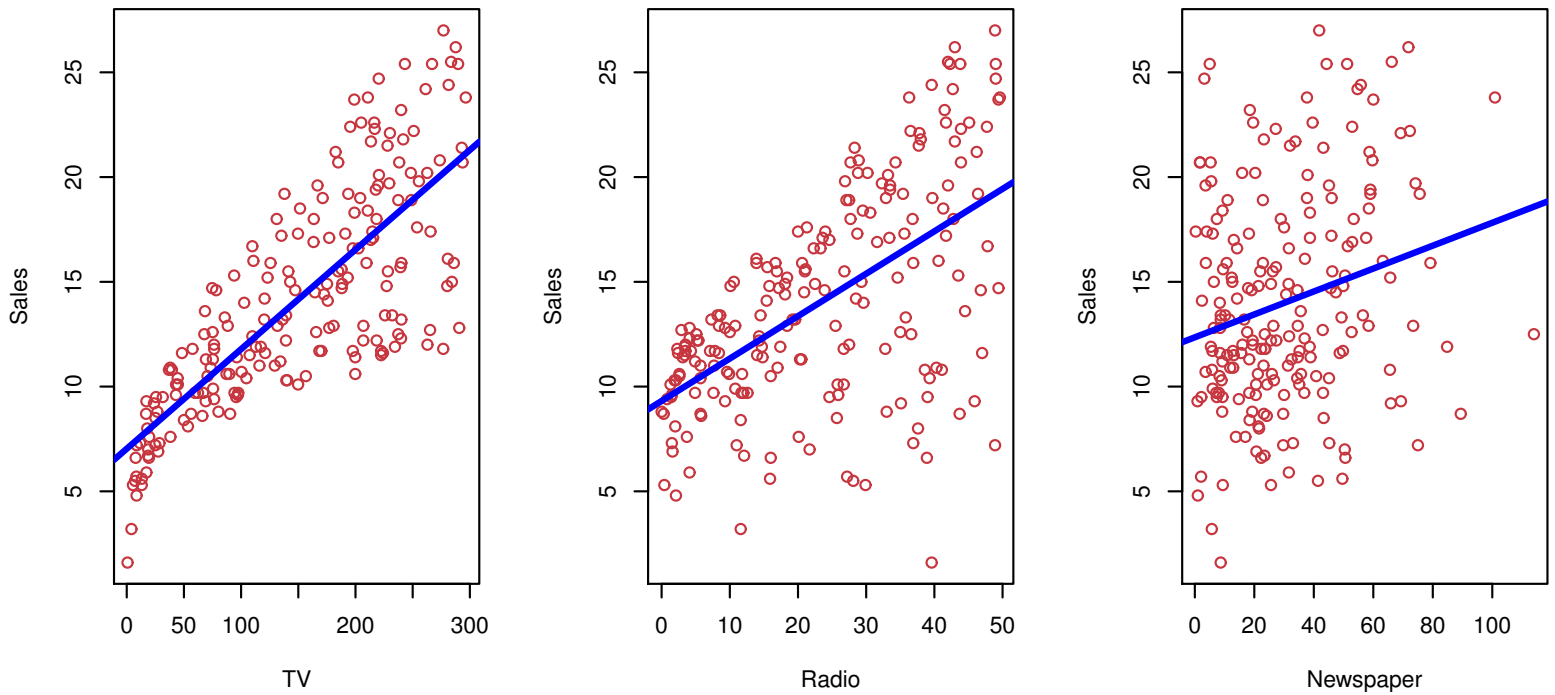
Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.
- objective is more fuzzy — find groups of samples that behave similarly, find features that behave similarly, find linear combinations of features with the most variation.
- difficult to know how well your are doing.
- different from supervised learning, but can be useful as a pre-processing step for supervised learning.
- Examples
  - Cocktail Party problem [e.g., see this https://tinyurl.com/33sma6yt]
  - [Just like the Facebook tagging system] you have collection of photos of multiple people, without information who is on which. You want to divide this dataset into multiple piles, each with photos of one individual
  - Generative Adversarial Networks (GANs)

More on the Basics of Statistical Learning

An Example

Shown are Sales vs the advertising budget on TV, Radio and Newspaper. Blue linear-regression line fits separately to each:



Can we predict Sales using these three? Basically,

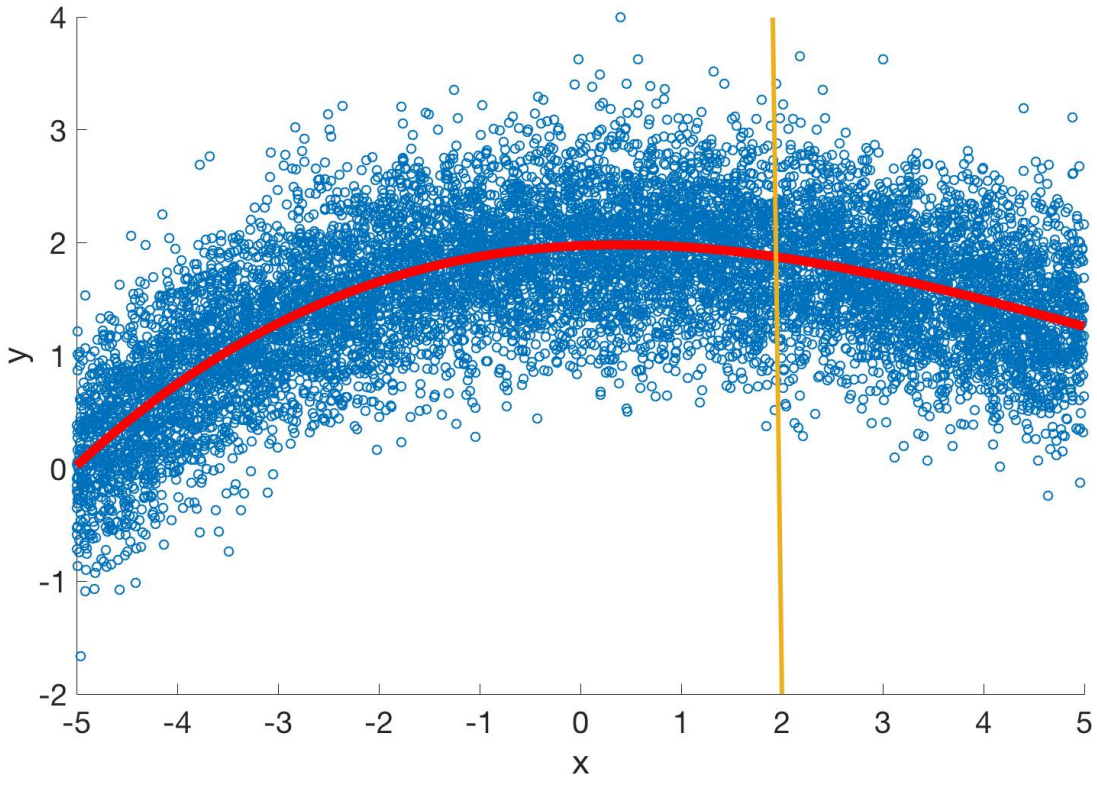
$\texttt{sales} = f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$

Why do we Need Predictive Models

- With a good fit we can make predictions of Y at new points $\boldsymbol{X} = \boldsymbol{x}$
- We can understand which features (i.e., components of $\boldsymbol{X} = (X_1, X_2, ..., X_p)^\top$ are important in explaining Y
- For instance in the sales example TV, Radio and Newspaper can be important but "which restaurant is close to the company" might be an irrelevant feature
- Depending on the complexity of the fit, we may be able to understand how each component X_j of $\boldsymbol{X}$ affects Y

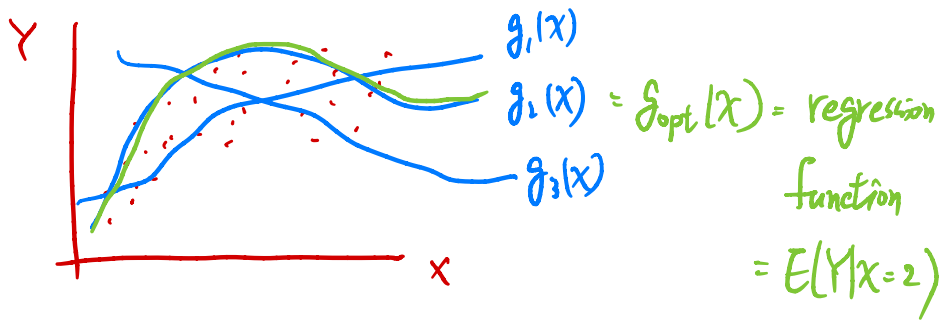
Best Predictive Model if We Had Enough Data

- Suppose we had many data samples (X, Y) as below
- What would have been a good estimate of Y for $X = 2$



- In mathematical terms what we are after is $E(Y|X = 2)$

Best Predictive Model if We Had Enough Data



- In an ideal setting of having many data, the reasonable value of Y corresponding to $X = \boldsymbol{x}$ is the average of all responses at $\boldsymbol{X} = \boldsymbol{x}$
- It can be mathematically shown that $f(\boldsymbol{x}) = E(Y|\boldsymbol{X} = \boldsymbol{x})$ is the function that minimizes $E[(Y - g(\boldsymbol{X}))^2|\boldsymbol{X} = \boldsymbol{x}]$ over all functions g at all points $\boldsymbol{X} = \boldsymbol{x}$.
- This ideal $f(\boldsymbol{x}) = E(Y|\boldsymbol{X} = \boldsymbol{x})$ is called the **regression function**

aka "God's Model"

Question: Can anyone show that $f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions g?

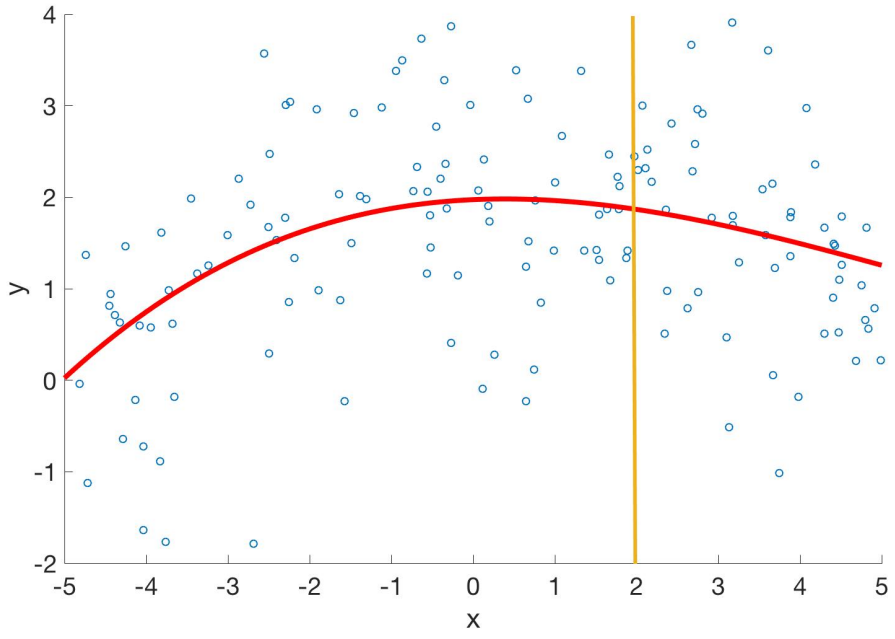
$$C = \mathbb{E}_{|X}\left[(y - g(x))^2\right] = \mathbb{E}_{|X}\left(y^2 - 2y \cdot g + g^2\right) = \mathbb{E}_{|X}(y^2) - 2g \cdot \mathbb{E}_{|X}(y) + g^2$$

$$\frac{\partial C}{\partial g} = 0 - 2\,\mathbb{E}_{|X}(y) + 2g = 0 \Rightarrow g_{opt} = \mathbb{E}_{|X} = E(y|X = x)$$

What Happens in Practice

In practice we don't have enough data to have access to f

Even if that could have been the case, doing it for all points $\boldsymbol{x}$ in the space is not practical



One immediate suggestion is considering neighborhoods around $\boldsymbol{x}$

Nearest neighbor averaging can only be good for small p (say < 4) and large N. [Example:] Points $\boldsymbol{x} = (0.9, \ldots, 0.9) \in \mathbb{R}^{100}$ and $\hat{\boldsymbol{x}} = (1, \ldots, 1) \in \mathbb{R}^{100}$ are quite far

Fundamental Limit

- As we observed the best we can do is the regression function

$$Y = f(\boldsymbol{x}) + \epsilon$$

- $f(\boldsymbol{x})$ can be considered as the actual physical model that generates data at $\boldsymbol{X} = \boldsymbol{x}$ and ϵ as the noise and uncertainty
- There is nothing we can do about ϵ, since it is a random variable not under our control
- Even in practice we cannot estimate $f(\boldsymbol{x})$ for $\boldsymbol{X} = \boldsymbol{x}$ and our estimate is something like $\hat{f}(\boldsymbol{x})$ different than $f(\boldsymbol{x})$
- Hence, the overall prediction error at a point $\boldsymbol{X} = \boldsymbol{x}$ is

$$E\left((Y - \hat{f}(\boldsymbol{x}))^2 | \boldsymbol{X} = \boldsymbol{x}\right) = \left(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\right)^2 + \text{Var}(\epsilon)$$

GM Actual Model

- First term can be improved by improving $\hat{f}(x)$, but the second term is constant and nothing can be done about it

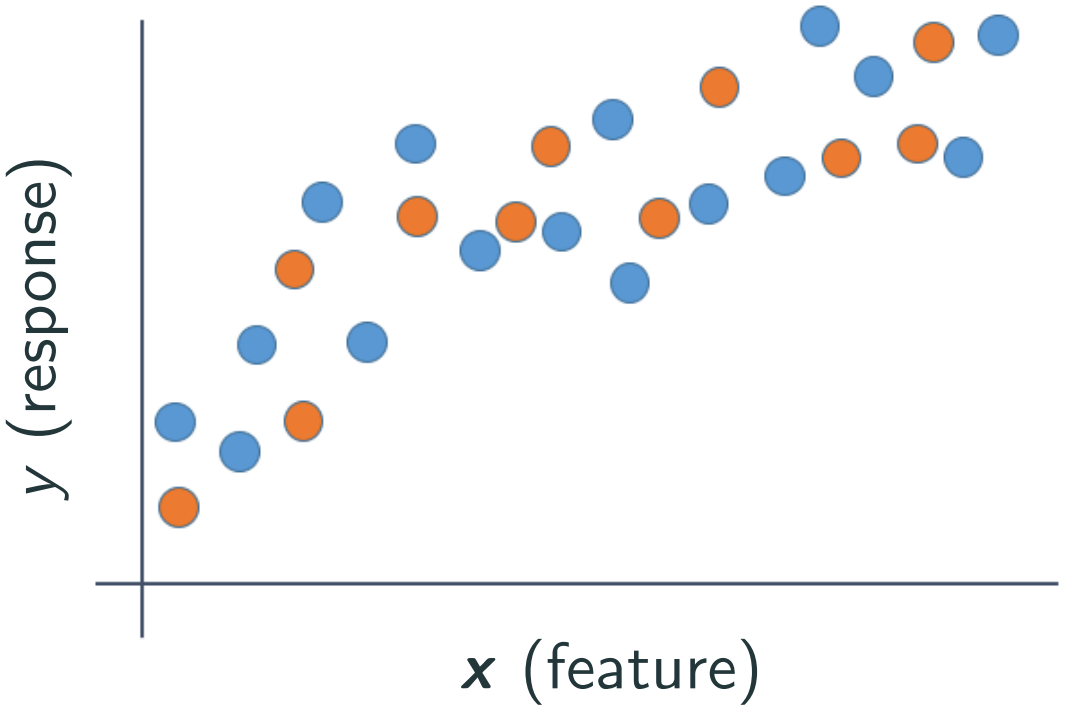
$$E_{|X}\left[(y-\hat{f}(x))^2\right] = E_{|X}\left(f+\varepsilon-\hat{f}\right) = E_{|X}\left[(f-\hat{f})^2+2\varepsilon(f-\hat{f})+\varepsilon^2\right]$$

$$= (f-\hat{f})^2 + 2(f-\hat{f})\cdot \overset{0}{E(\varepsilon)} + E(\varepsilon^2)$$

$$= (f-\hat{f})^2 + \mathrm{Var}(\varepsilon)$$

How to Assess Model Accuracy in Practice

- As we said before, we don't normally have access to $f(\boldsymbol{x})$, so how can we assess the accuracy of an estimated model $\hat{f}(\boldsymbol{x})$?
- Usually we split the data into a training set and a test set to later evaluate the accuracy



Parametric vs Nonparametric Models

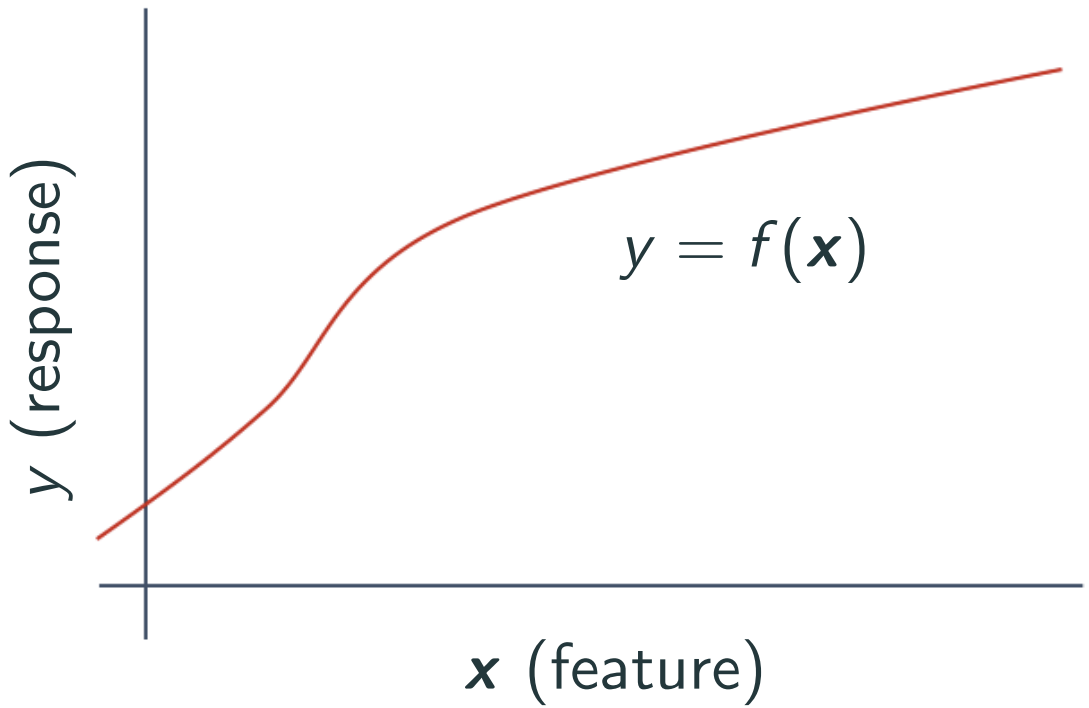
- In parametric models, we use some predefined forms for the model $\hat{f}$ that we would like to fit
- For instance in linear regression we use the following general form and the fitting aims to find the coefficients

$$Y = f_L(\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p$$

- Non-parametric methods do not make explicit assumptions about the functional form of the fit. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly
- Thin-plate spline is an example of nonparametric fit, where you can control the smoothness of the fit vs its flexibility to match the training data

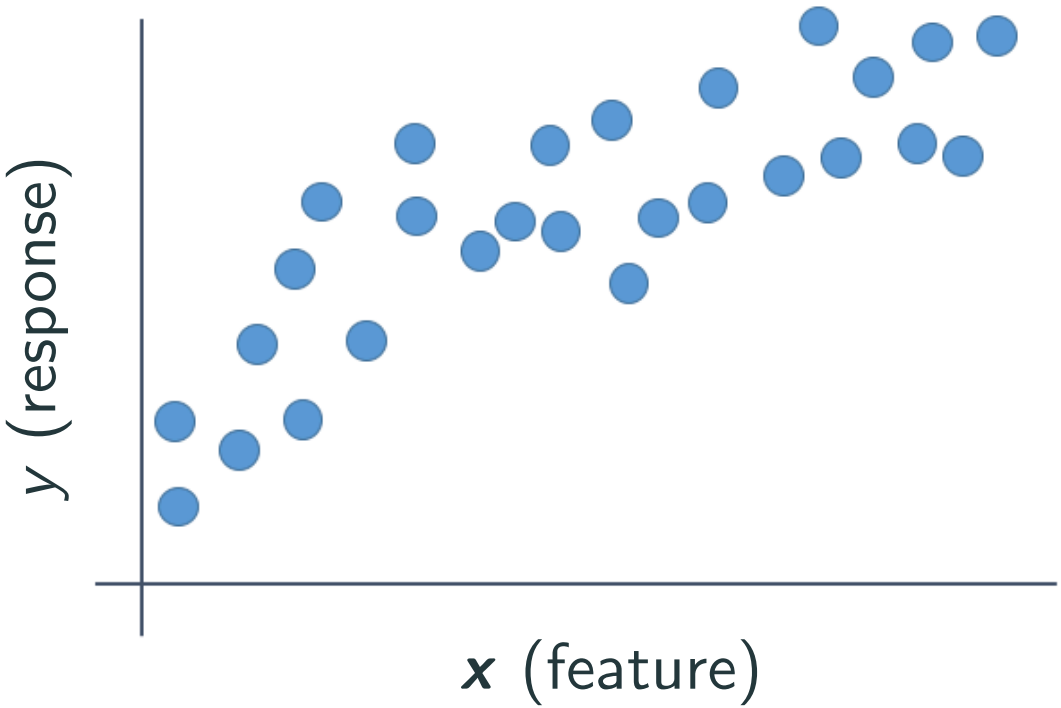
Basics of Model Fitting

- We have a (physical) system which produces a response y to an input $\boldsymbol{x}$



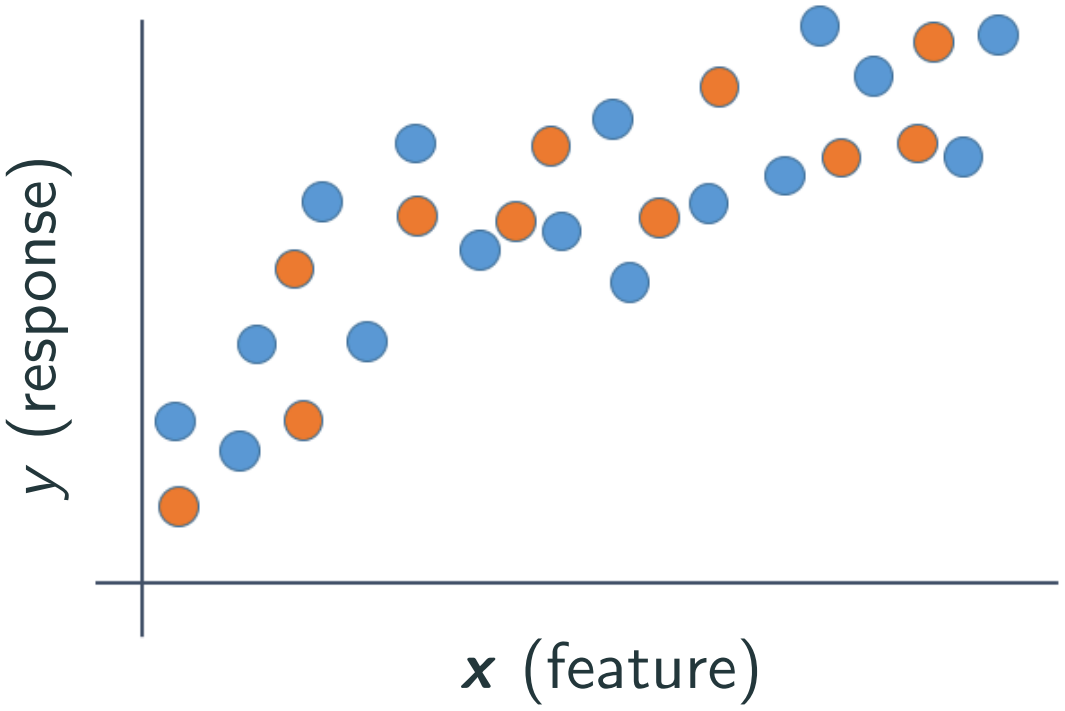
Basics of Model Fitting

- The system is a black box and all we have are samples of input/output



Basics of Model Fitting

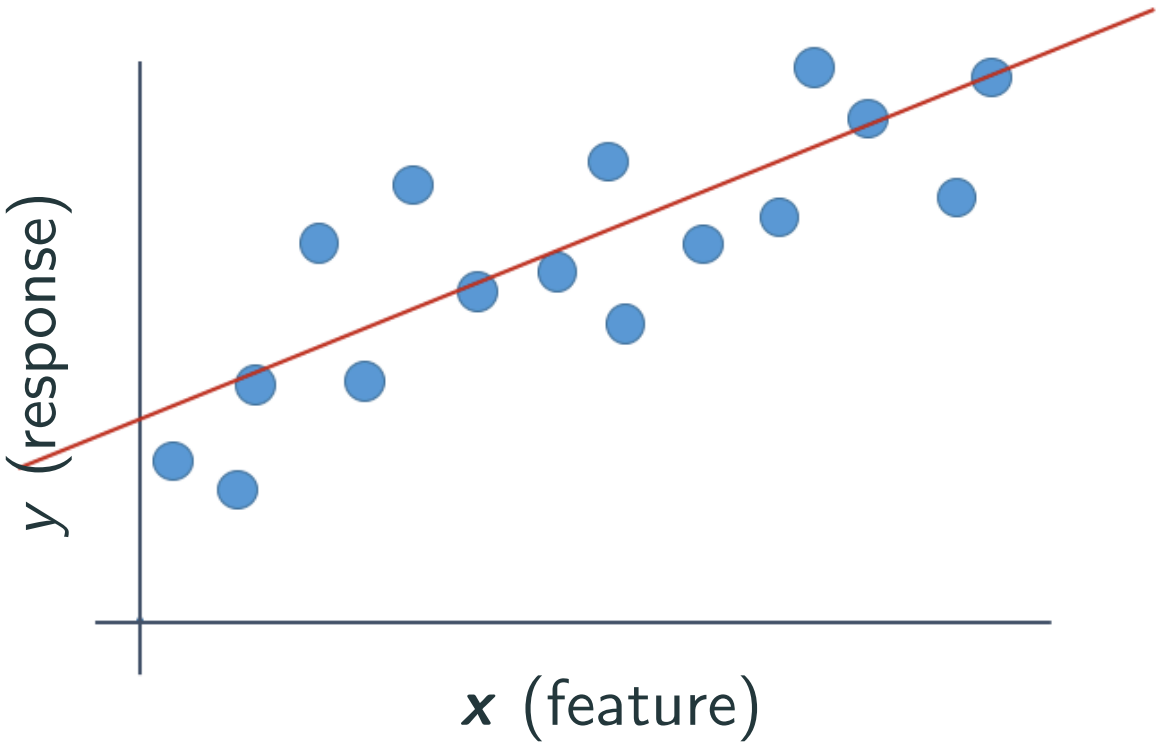
- We may try fitting a model to the data



- We split the data into a training set and a test set to later evaluate the accuracy

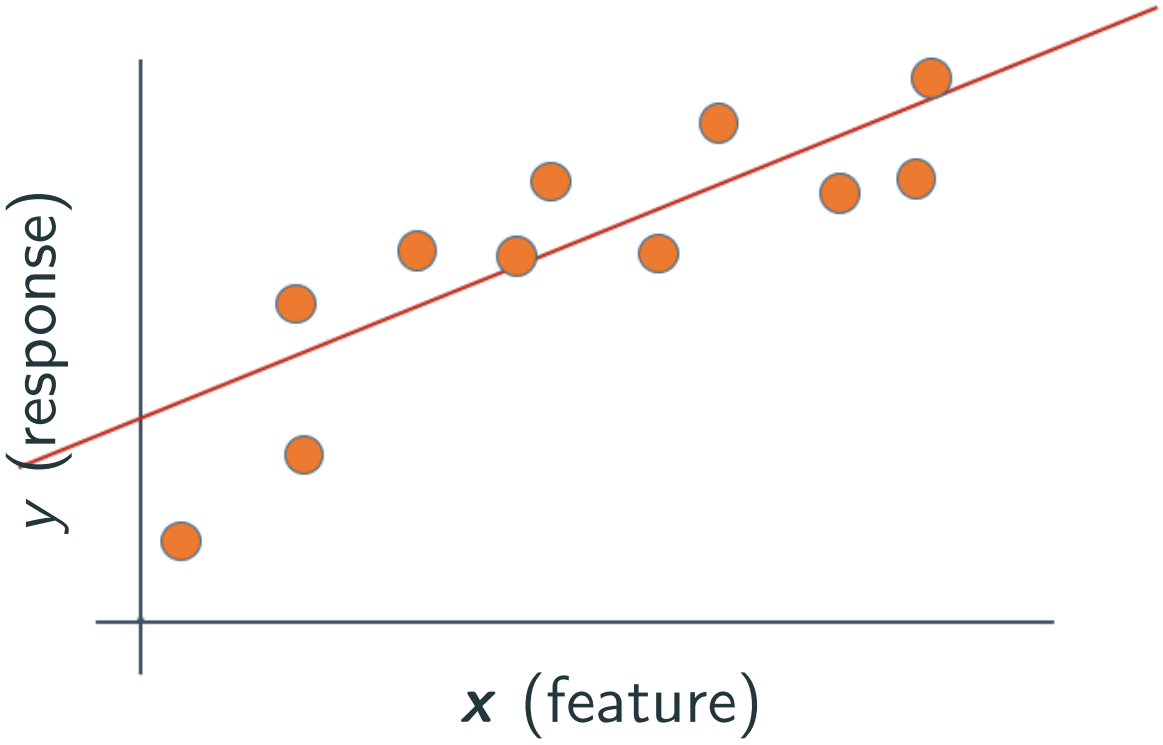
Basics of Model Fitting

- A linear model: $y = w_1 x + w_0$



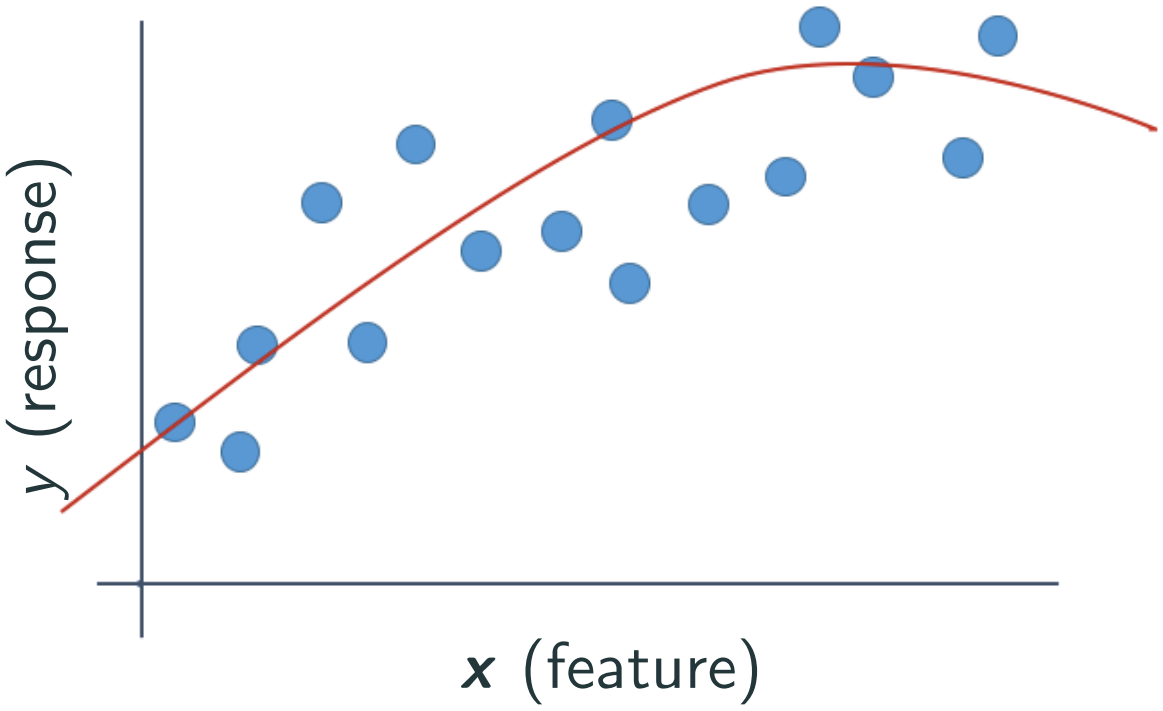
Basics of Model Fitting

- Testing the linear model: $y = \hat{f}(x) = w_1 x + w_0$



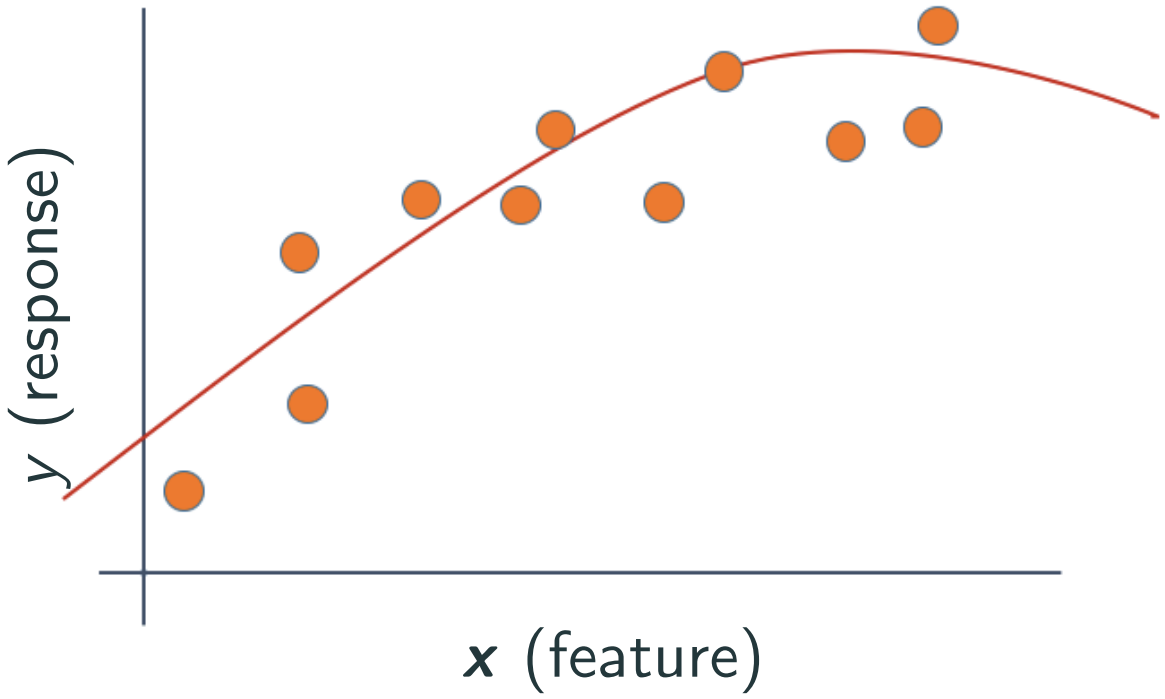
Basics of Model Fitting

- Second order polynomial model: $y = \hat{f}(x) = w_2 x^2 + w_1 x + w_0$



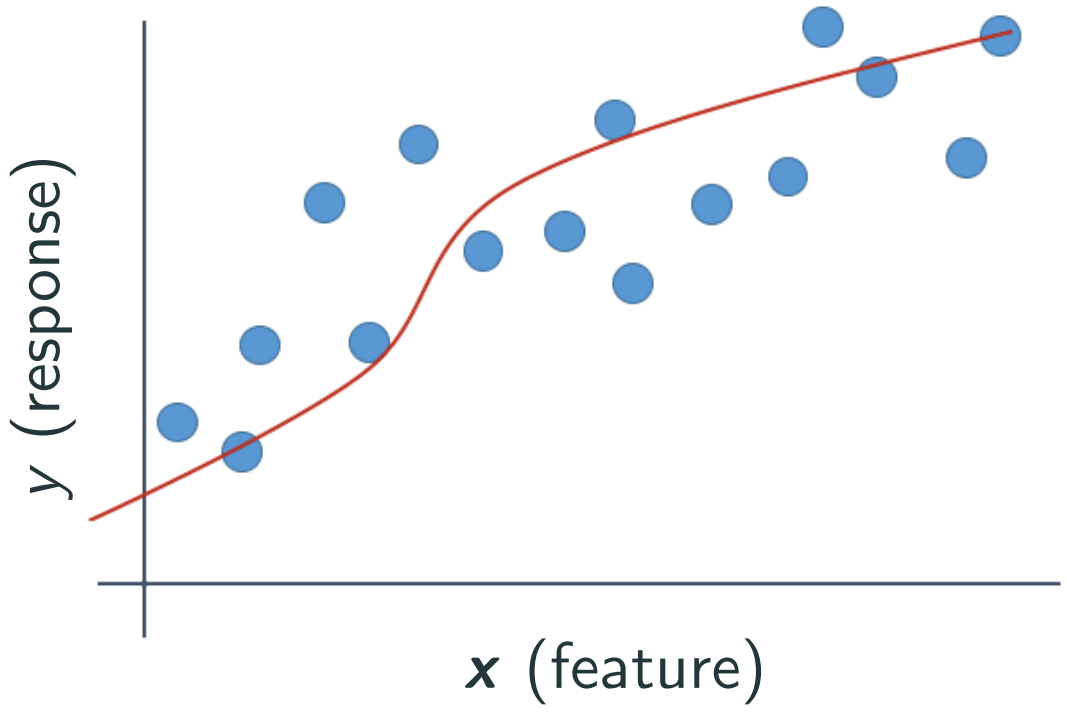
Basics of Model Fitting

- Testing the second order polynomial model:
  $y = \hat{f}(x) = w_2 x^2 + w_1 x + w_0$



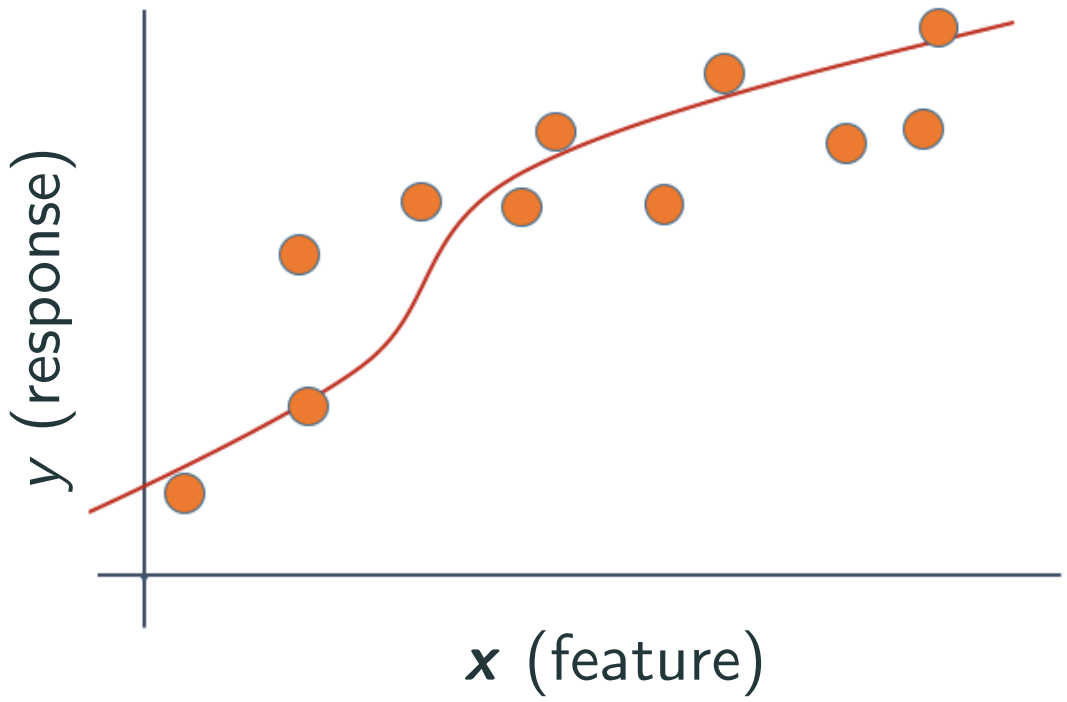
Basics of Model Fitting

- Third order polynomial model: $y = \hat{f}(x) = w_3 x^3 + w_2 x^2 + w_1 x + w_0$



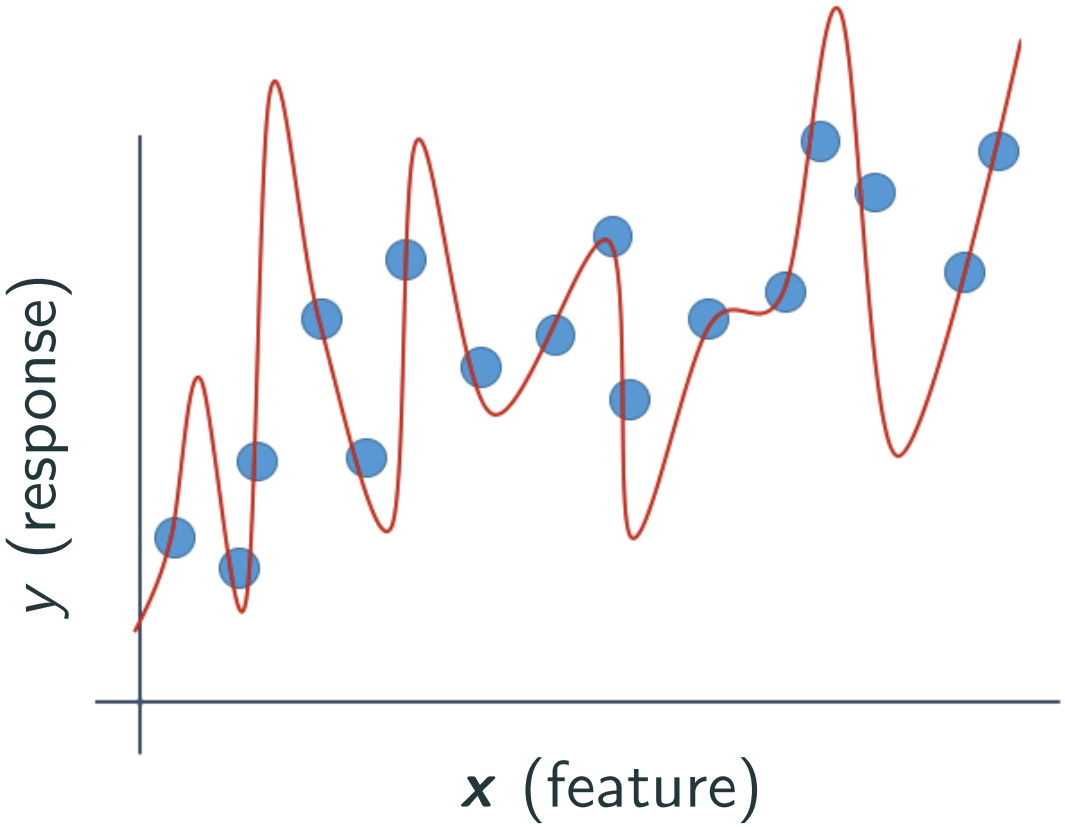
Basics of Model Fitting

- Testing the third order polynomial model:
  $y = \hat{f}(x) = w_3 x^3 + w_2 x^2 + w_1 x + w_0$



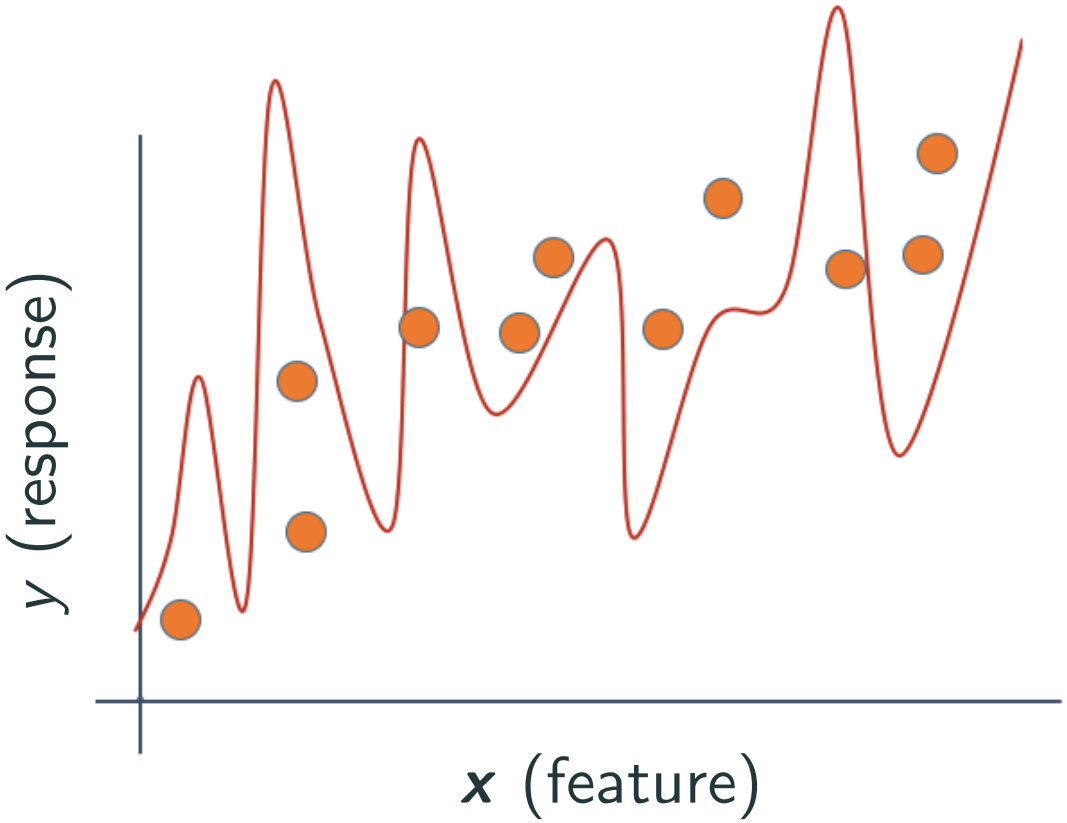
Basics of Model Fitting

- High order polynomial model: $y = \hat{f}(x) = \sum_{p=0}^{P} w_p x^p$ or a thin-plate spline with minor smoothness control



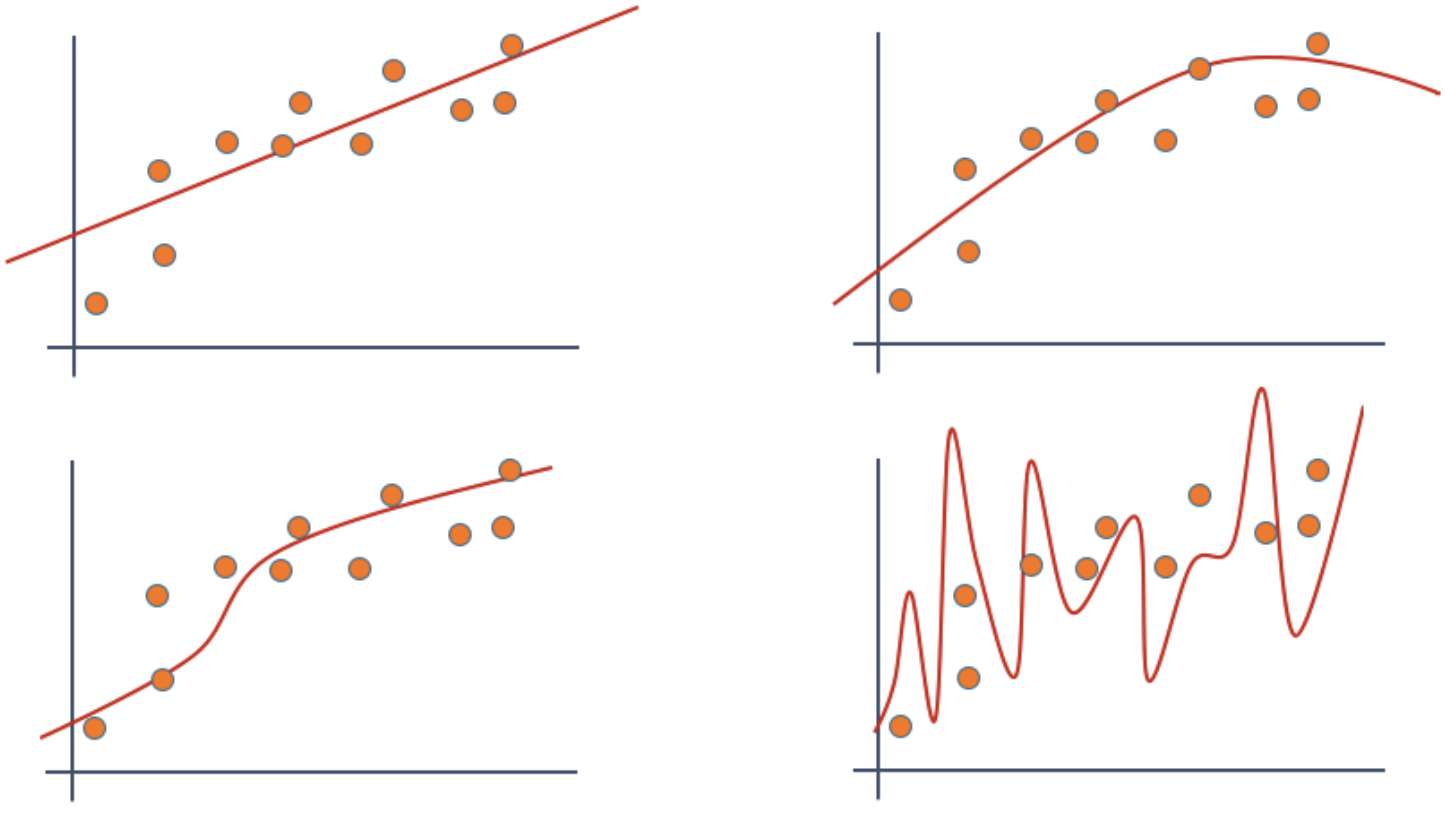
Basics of Model Fitting

- Testing the high order polynomial model: $y = \hat{f}(x) = \sum_{p=0}^{P} w_p x^p$ or a thin-plate spline with minor smoothness control



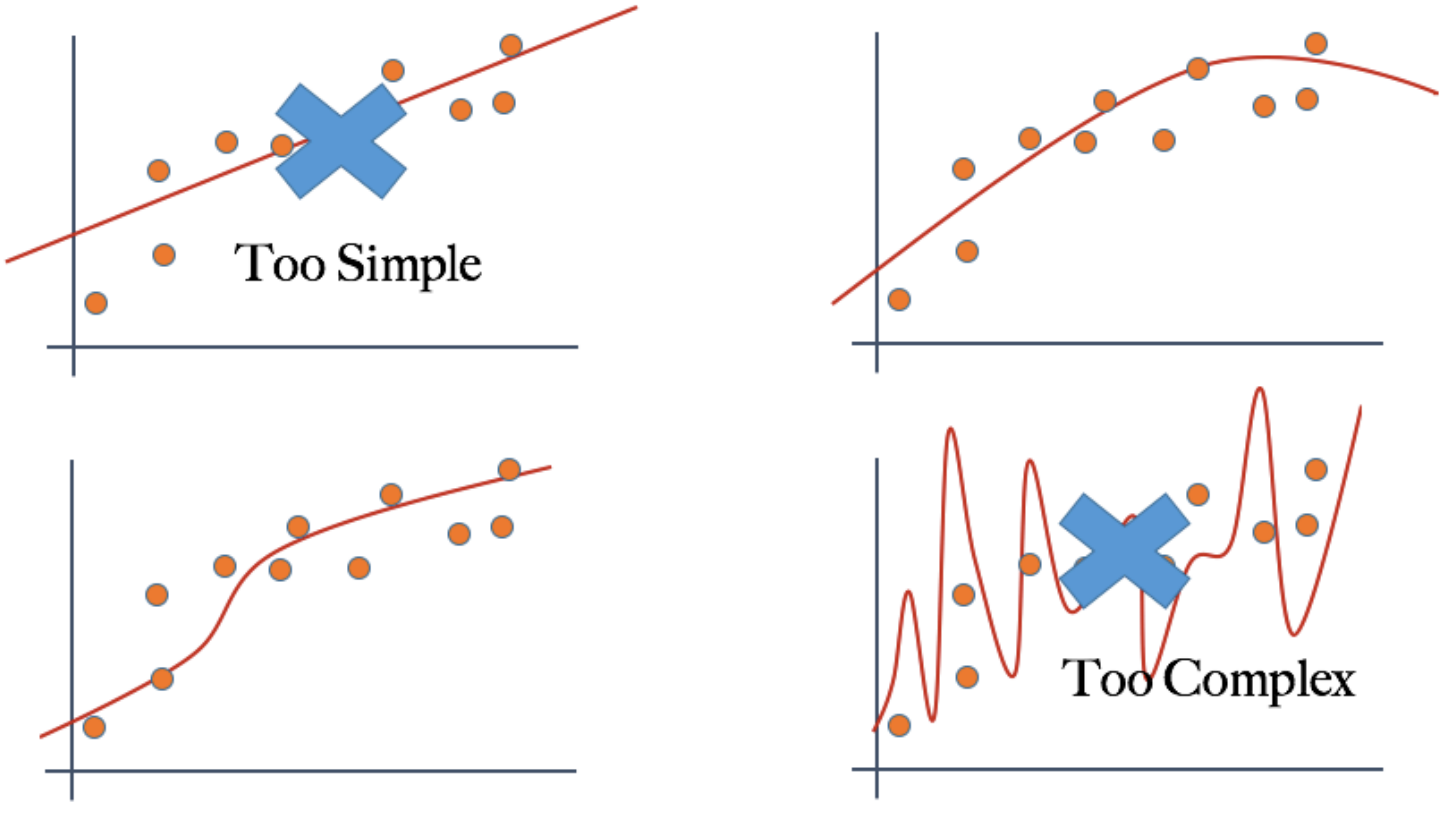
Basics of Model Fitting

- Which model to pick? Underfitting vs overfitting



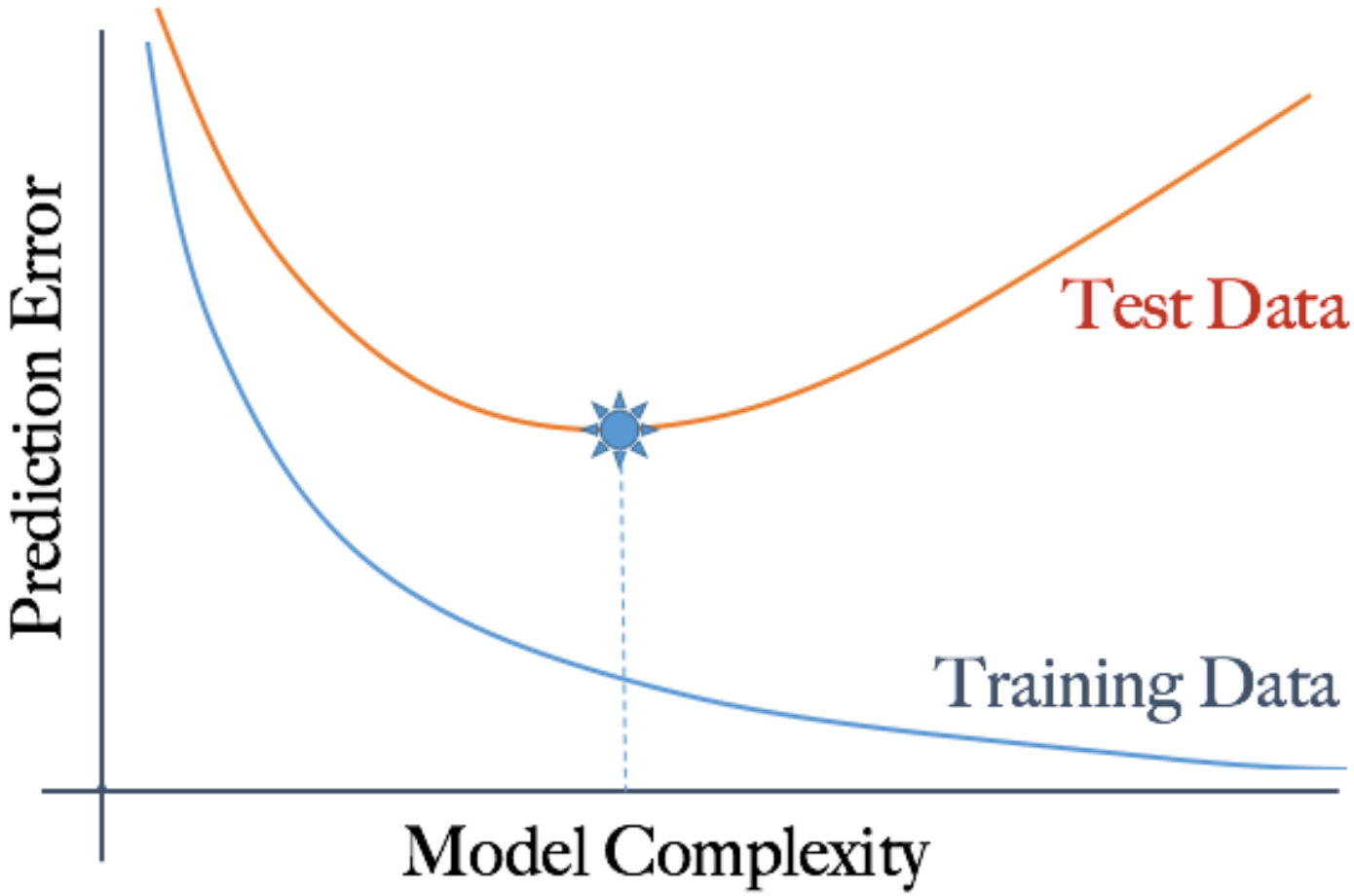
Basics of Model Fitting

- Which model to pick? Underfitting vs overfitting



Typical Curve

– The balance between model flexibility and test error typically looks as below



Assessing Model Accuracy

- Once we fit our model $\hat{f}$ we want to evaluate its performance
- Of course the evaluation should not be with respect to the training data, since that would not be a fair evaluation
- Instead the evaluation is performed on the test data (that has not been used in the fitting)
- Mean Squared Error $= \text{Ave}_{i \in Test}(Y_i - \hat{f}(\boldsymbol{x}_i))^2$

Bias vs Variance(Theory)

- Suppose we fit a model $\hat{f}$ and we have a large test set to accurately calculate the MSR with respect to the test data
- Assuming that $(\boldsymbol{x}_t, Y_t)$ is a fixed point in the test set and T is the training set based on which $\hat{f}$ is derived, we generally have

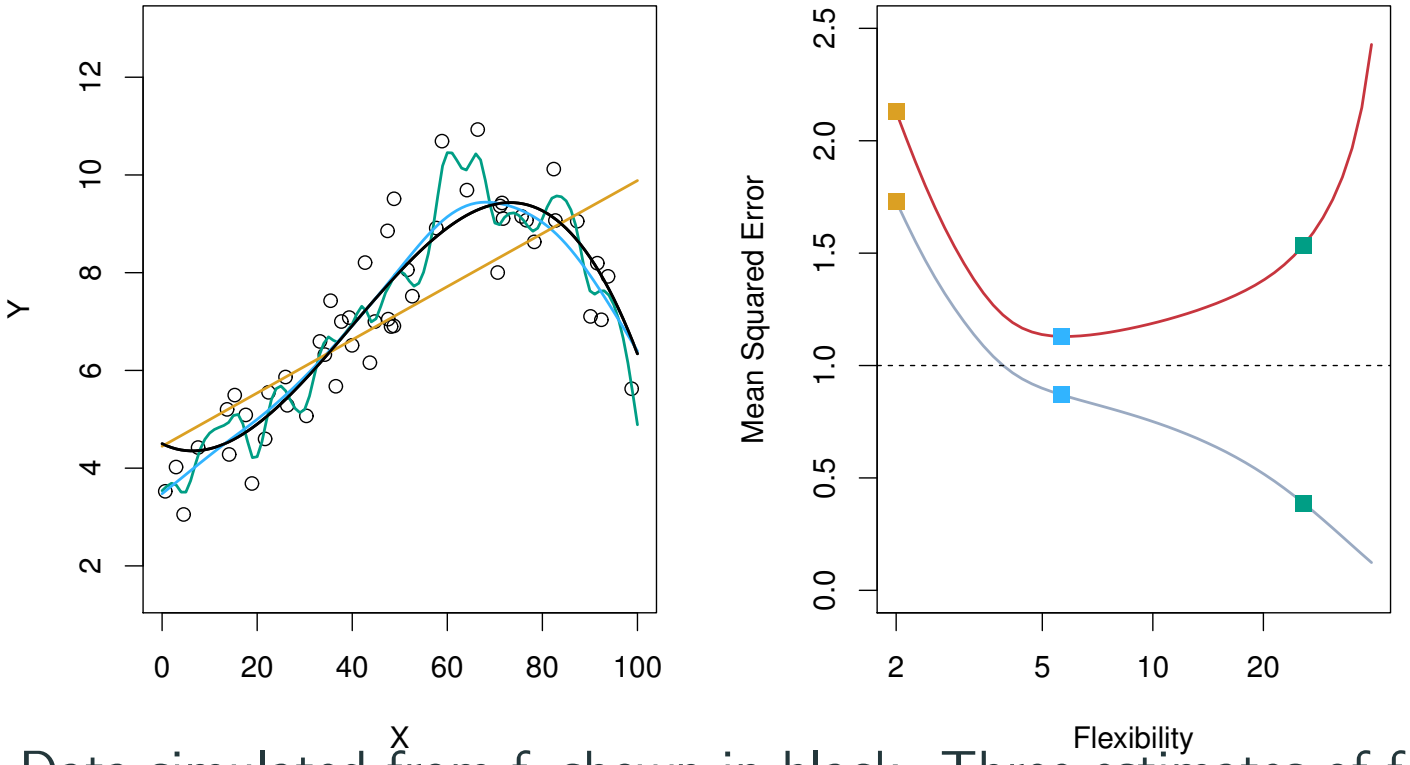
Model Complexity overfitting evaluation

$$E_{tot}\left(Y_t - \hat{f}(\boldsymbol{x}_t)\right)^2 = \text{var}(\hat{f}(\boldsymbol{x}_t)) + \left(\text{Bias}(\hat{f}(\boldsymbol{x}))\right)^2 + \text{var}(\epsilon)$$

$$\text{Bias}(\hat{f}(\boldsymbol{x})) = E_T(\hat{f}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t))$$

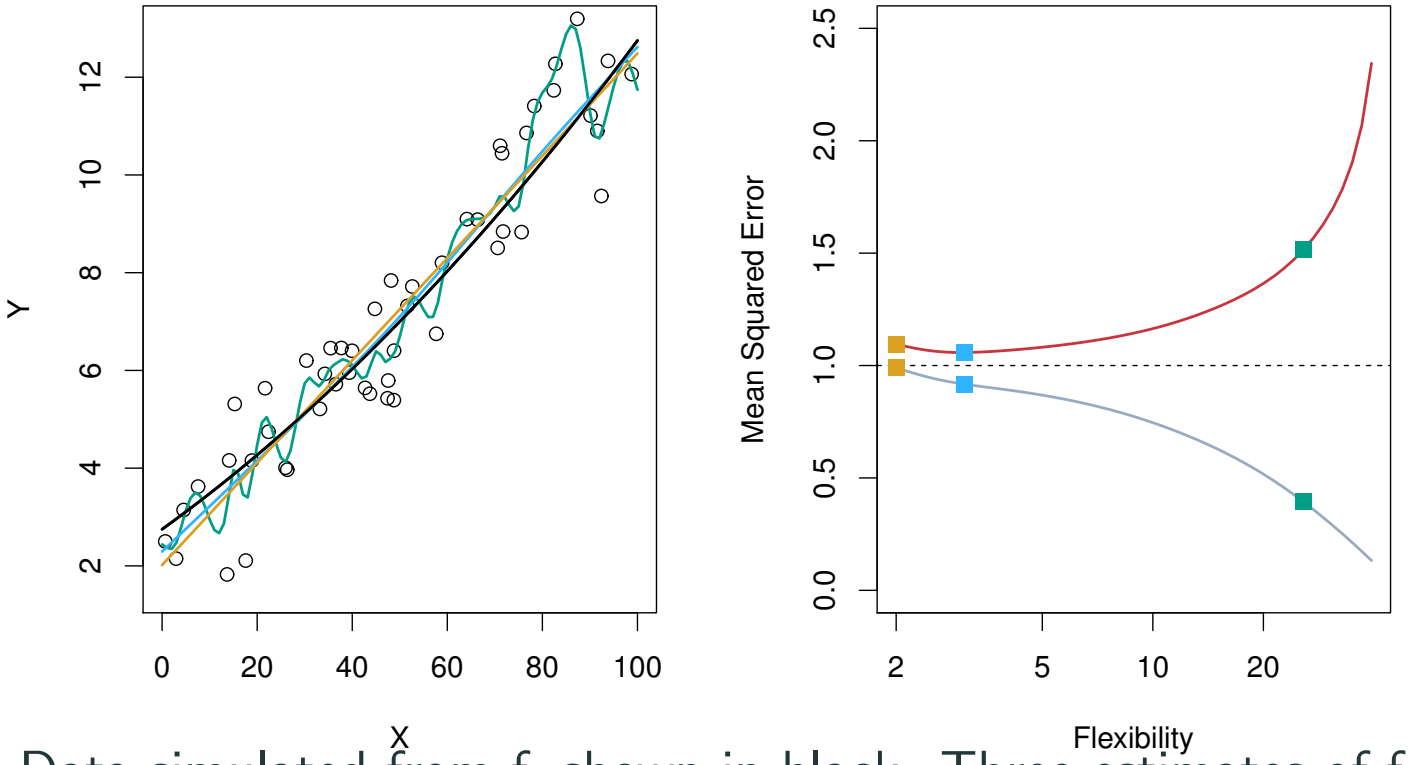
- [Lets derive the equation above]
- Normally as we make $\hat{f}$ more flexible by making it complex (using more sophisticated formulations and more features) the **model variance** term $\text{var}(\hat{f}(\boldsymbol{x}_t))$ increases. This on the other hand decreases the bias (which we want to happen)
- So reducing the test error becomes a trade-off between the bias and the variance

Bias vs Variance



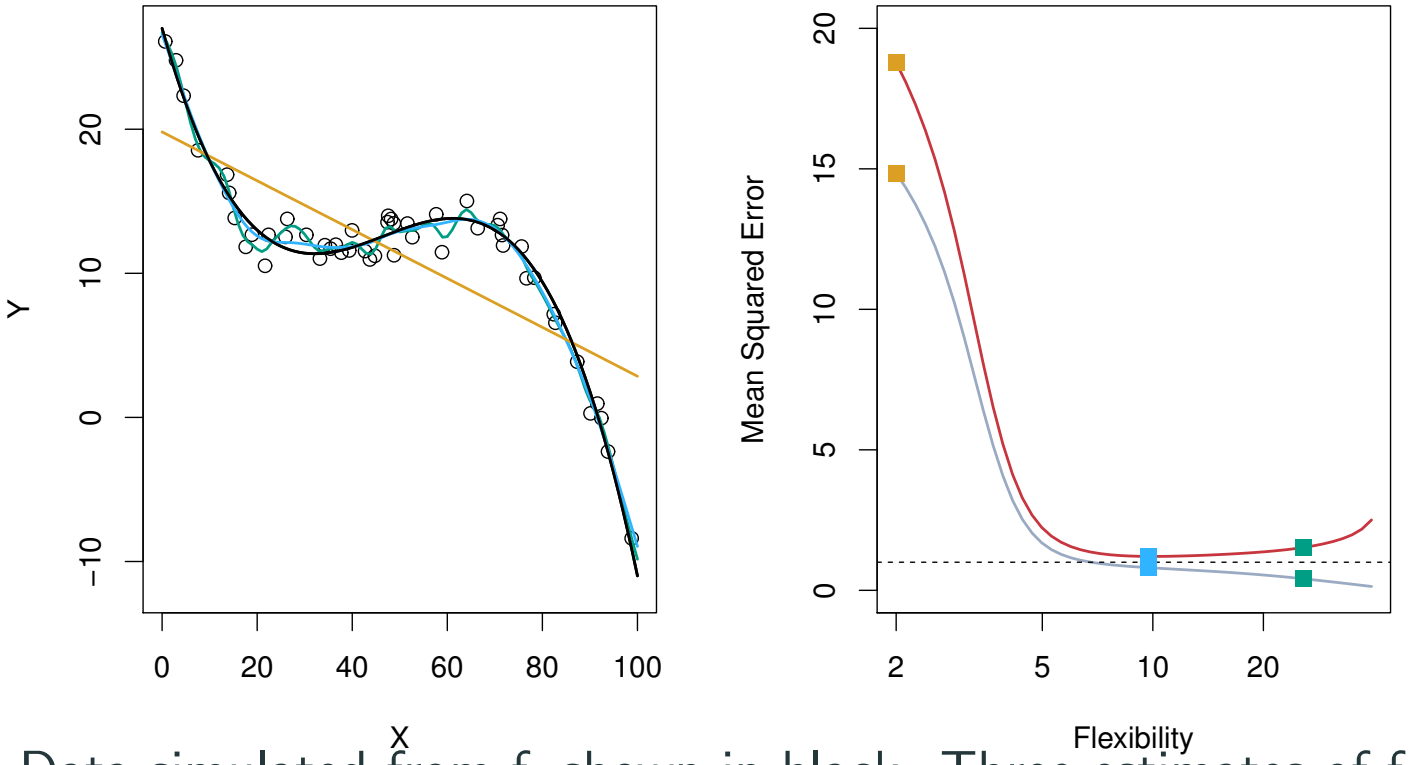
Left: Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Bias vs Variance



Left: Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Bias vs Variance



Left: Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

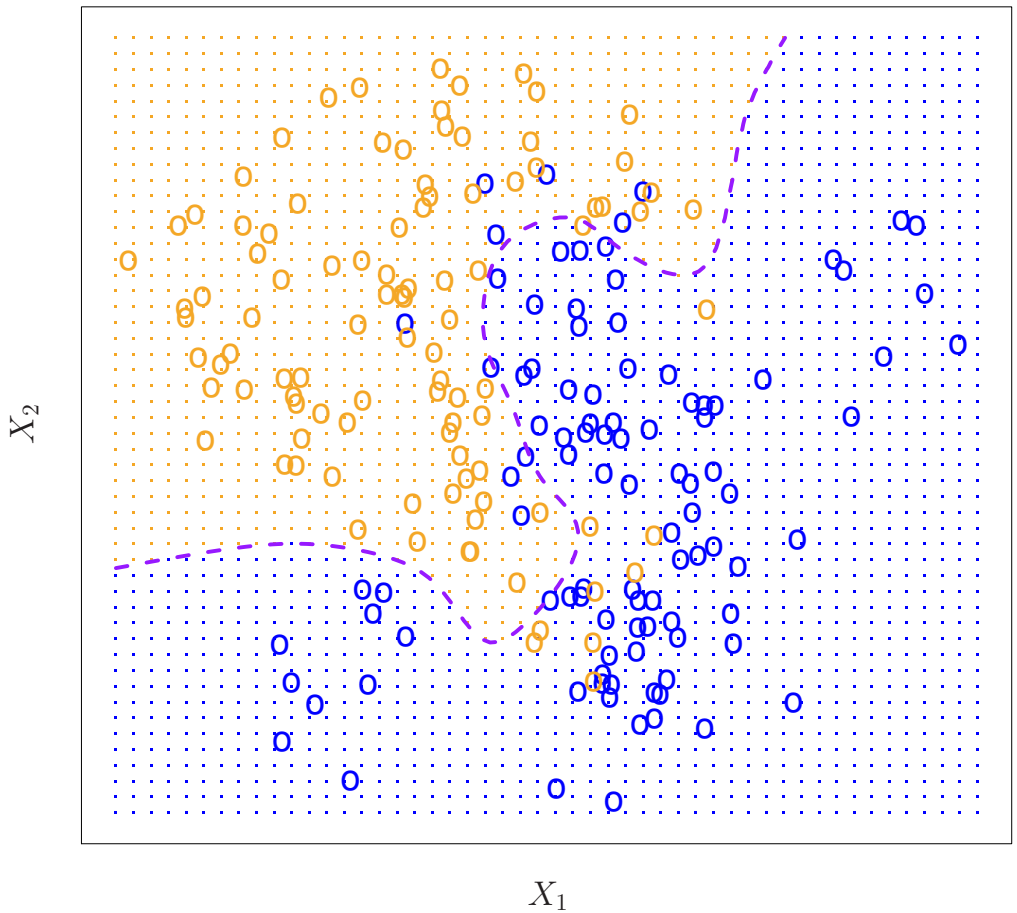
Trade Offs

We just discussed the bias vs variance trade off. In general we deal with various trade offs

- Prediction accuracy versus interpretability:
  e.g., linear models are easy to interpret; thin-plate splines are not.
- Good fit versus over-fit or under-fit:
  How do we know when the fit is just right? (just discussed on the figures above)
- Parsimony versus black-box:
  We often prefer a simpler model involving fewer variables over a black-box predictor involving them all

Classification

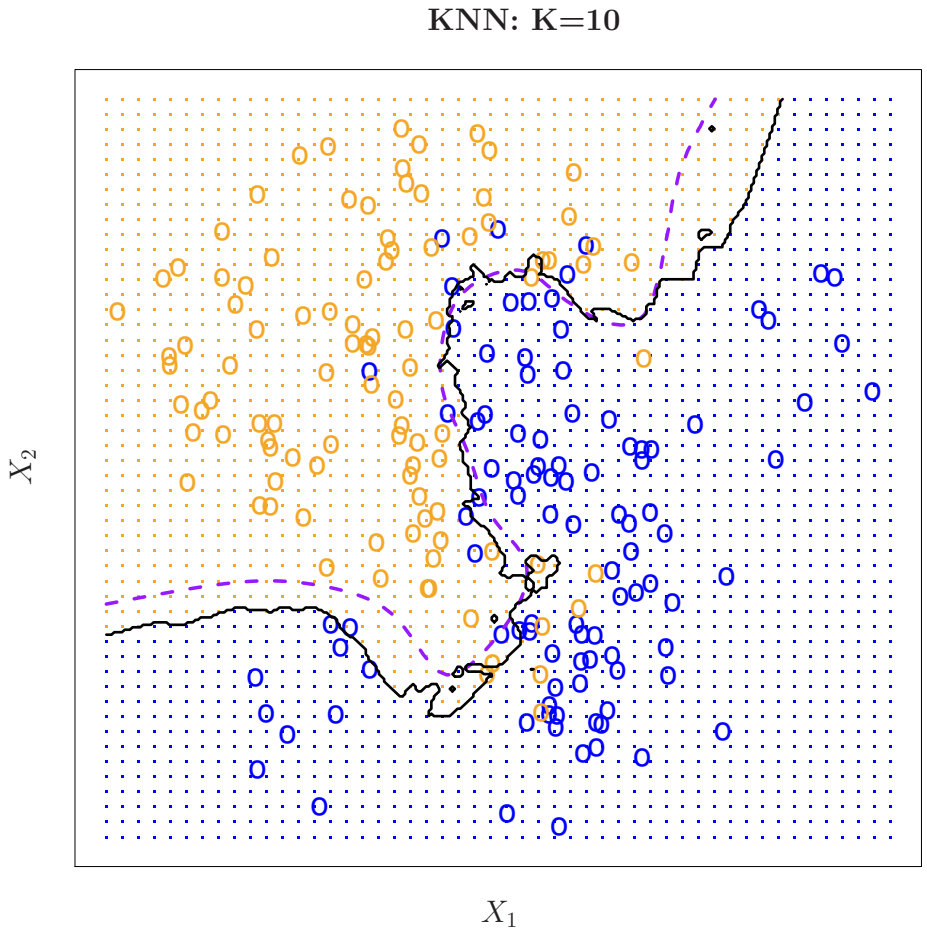
- How overfitting looks for classification problems
- K- Nearest Neighbors Approach



(ideal classifier)

Classification

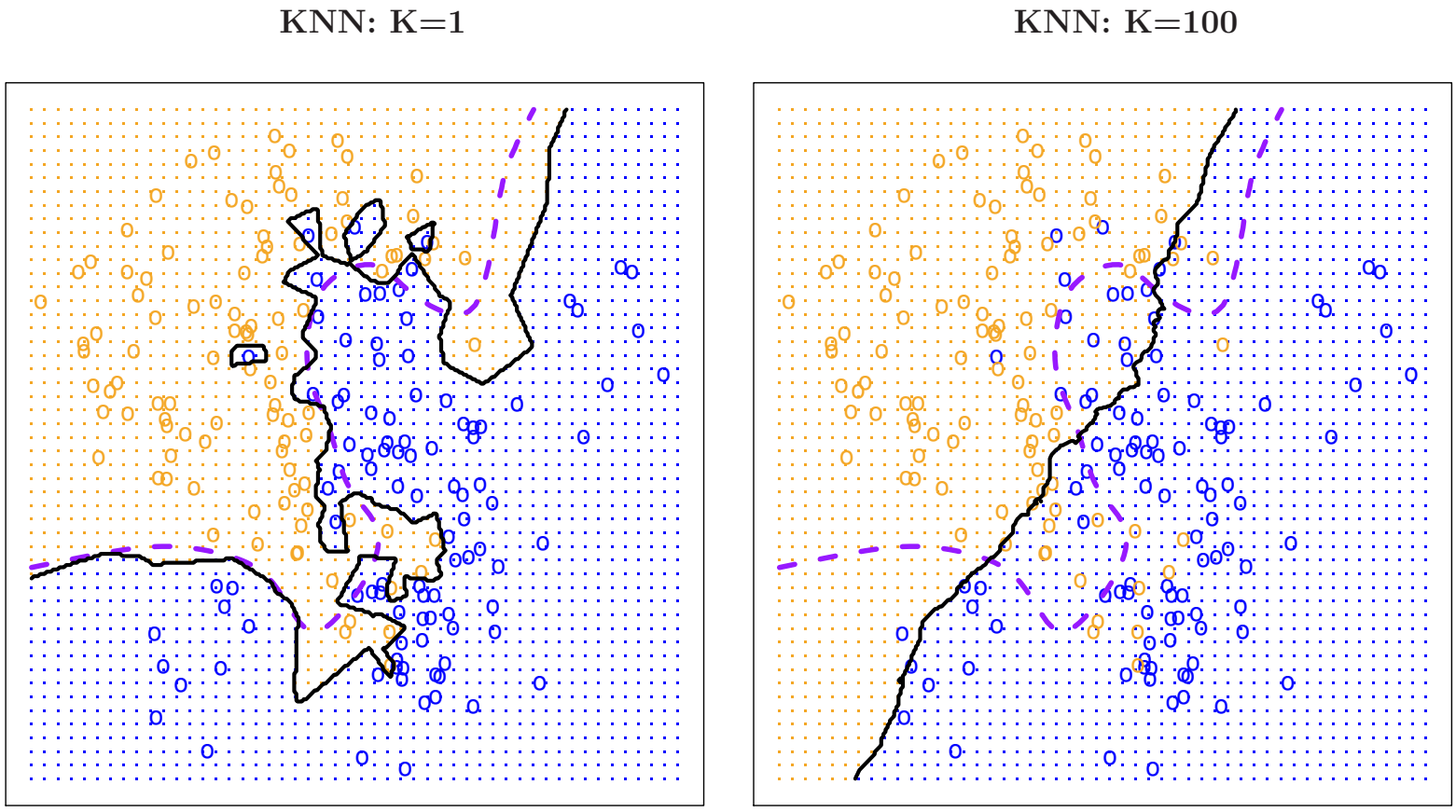
- How overfitting looks for classification problems
- K- Nearest Neighbors Approach (the dense grid is somehow our test)



(KNN Classifier)

Classification

- How overfitting looks for classification problems
- K- Nearest Neighbors Approach (the dense grid is somehow our test)



(KNN Classifier)

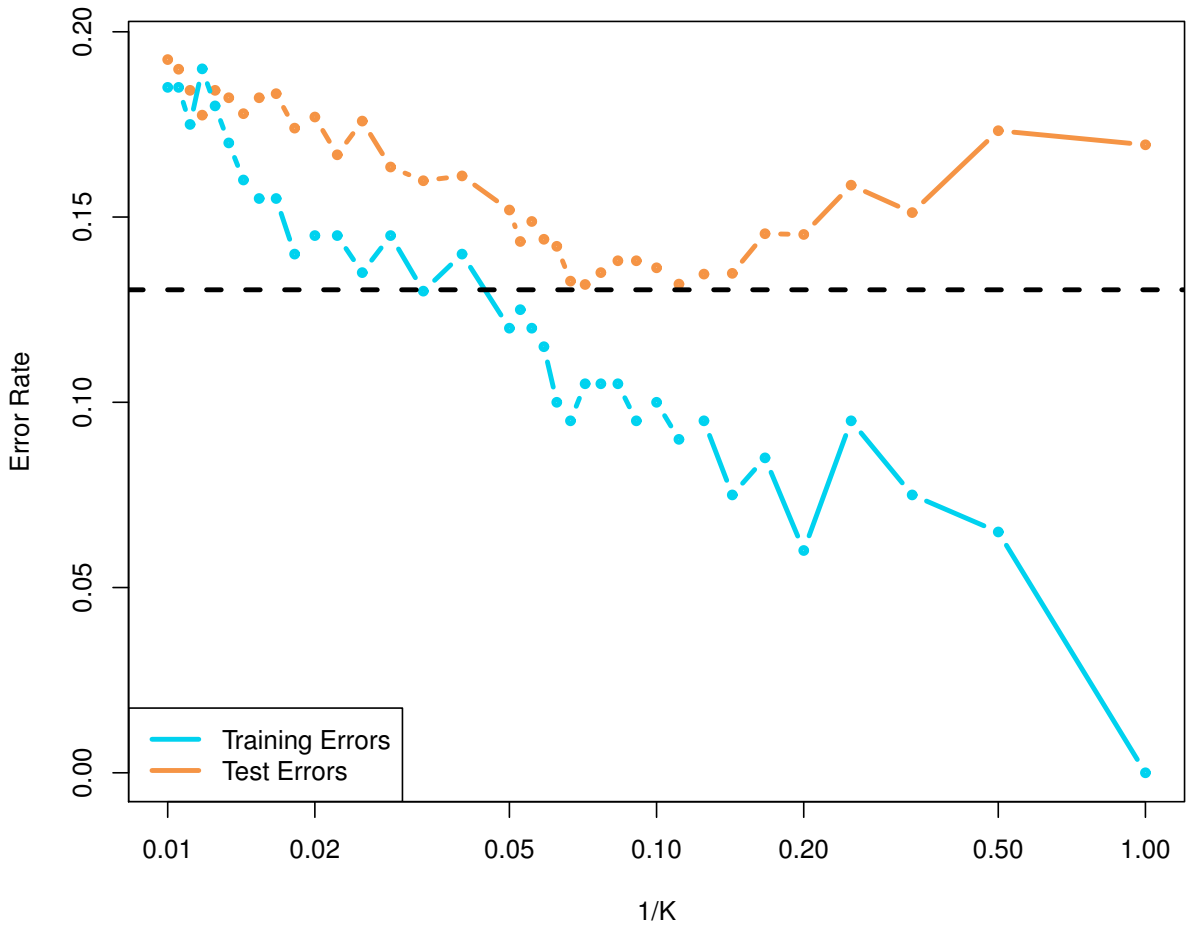
Classification

- K- Nearest Neighbors Approach
- Note that in KNN flexibility is inversely proportional to K

(Just like trying to walk among many people surrounding you)

Classification

- The Trade Off Curve
- K- Nearest Neighbors Approach



(Trade Off)

Lecture Overview

Overview

- Discussed the course information
- Related machine learning to model understanding in science and engineering
- Presented some machine learning examples in real world
- Talked about supervised vs unsupervised learning
- Talked about regression and classification
- Discussed some basics of statistical learning
  - Regression function
  - Test vs training data
  - Overfitting and model trade off
- **Don't worry if some of the topics covered look vague!** We will revisit all this material in depth later in the course

Bias ?