ADEC 7430: Big Data Econometrics

Introduction to Machine Learning

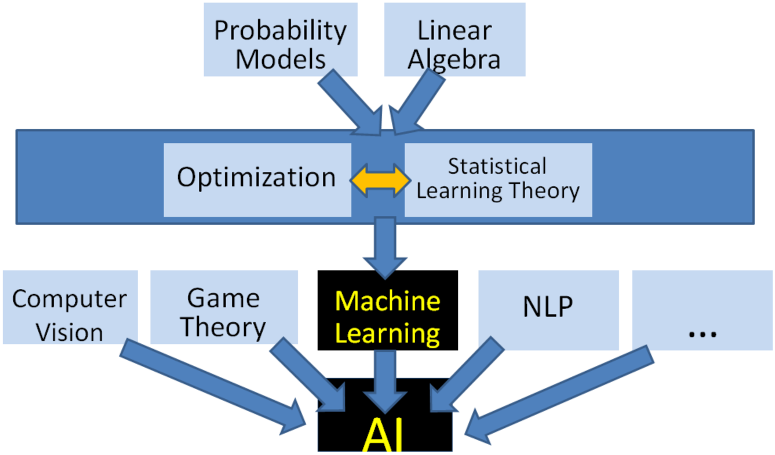
Dr. Nathan Bastian

Woods College of Advancing Studies

Boston College

Assignment

- **Reading:** Ch. 1, Ch. 2
- **Study:** Lecture Slides, Lecture Videos
- **Activity:** Quiz 1, R Lab 1, Discussion #1



References

- *An Introduction to Statistical Learning, with Applications in R* (2013), by G. James, D. Witten, T. Hastie, and R. Tibshirani.
- *The Elements of Statistical Learning* (2009), by T. Hastie, R. Tibshirani, and J. Friedman.
- *Learning from Data: A Short Course* (2012), by Y. Abu-Mostafa, M. Magdon-Ismail, and H. Lin.
- *Machine Learning: A Probabilistic Perspective* (2012), by K. Murphy
- *R and Data Mining: Examples and Case Studies* (2013), Y. Zhao.

Lesson Goals

- Describe the basic concepts of the learning problem and why/how machine learning methods are used to learn from data to find underlying patterns for prediction and decision-making.

- Explain the learning algorithm trade-offs, balancing performance within training data and robustness on unobserved test data.

- Differentiate between supervised and unsupervised learning methods as well as regression versus classification methods.

- Summarize the basic concepts of assessing model accuracy and the bias-variance trade-off.

- Use the R statistical programming language and practice by exploring data using basic statistical analysis.

Big Data is Everywhere

- We are in the era of **big data**!
  - 40 billion indexed web pages
  - 100 hours of video are uploaded to YouTube every minute

- The deluge of data calls for automated methods of data analysis, which is what **machine learning** provides!

What is Machine Learning?

- **Machine learning** is a set of methods that can *automatically* detect patterns in data.

- These uncovered patterns are then used to predict future data, or to perform other kinds of decision-making under uncertainty.

- The key premise is *learning* from data!!

What is Machine Learning?

- Addresses the problem of analyzing huge bodies of data so that they can be understood.
- Providing techniques to automate the analysis and exploration of large, complex data sets.
- Tools, methodologies, and theories for revealing patterns in data – critical step in knowledge discovery.

What is Machine Learning?

- <u>Driving Forces</u>:
  - Explosive growth of data in a great variety of fields
    - Cheaper storage devices with higher capacity
    - Faster communication
    - Better database management systems
  - Rapidly increasing computing power

- We want to make the data work for us!!

Examples of Learning Problems

- Machine learning plays a key role in many areas of science, finance and industry:
  - Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack. The prediction is to be based on demographic, diet and clinical measurements for that patient.
  - Predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data.
  - Identify the numbers in a handwritten ZIP code, from a digitized image.
  - Estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum of that person's blood.
  - Identify the risk factors for prostate cancer, based on clinical and demographic variables.

Research Fields

- Statistics / Statistical Learning
- Data Mining
- Pattern Recognition
- Artificial Intelligence
- Databases
- Signal Processing

Applications

- Business
  - Walmart data warehouse mined for advertising and logistics
  - Credit card companies mined for fraudulent use of your card based on purchase patterns
  - Netflix developed movie recommender system
- Genomics
  - Human genome project: collection of DNA sequences, microarray data

Applications (cont.)

- Information Retrieval
  - Terabytes of data on internet, multimedia information (video/audio files)
- Communication Systems
  - Speech recognition, image analysis

The Learning Problem

- Learning from data is used in situations where we don't have any analytic solution, but we do have data that we can use to construct an empirical solution

- The basic premise of learning from data is the use of a set of observations to uncover an underlying process.

The Learning Problem (cont.)

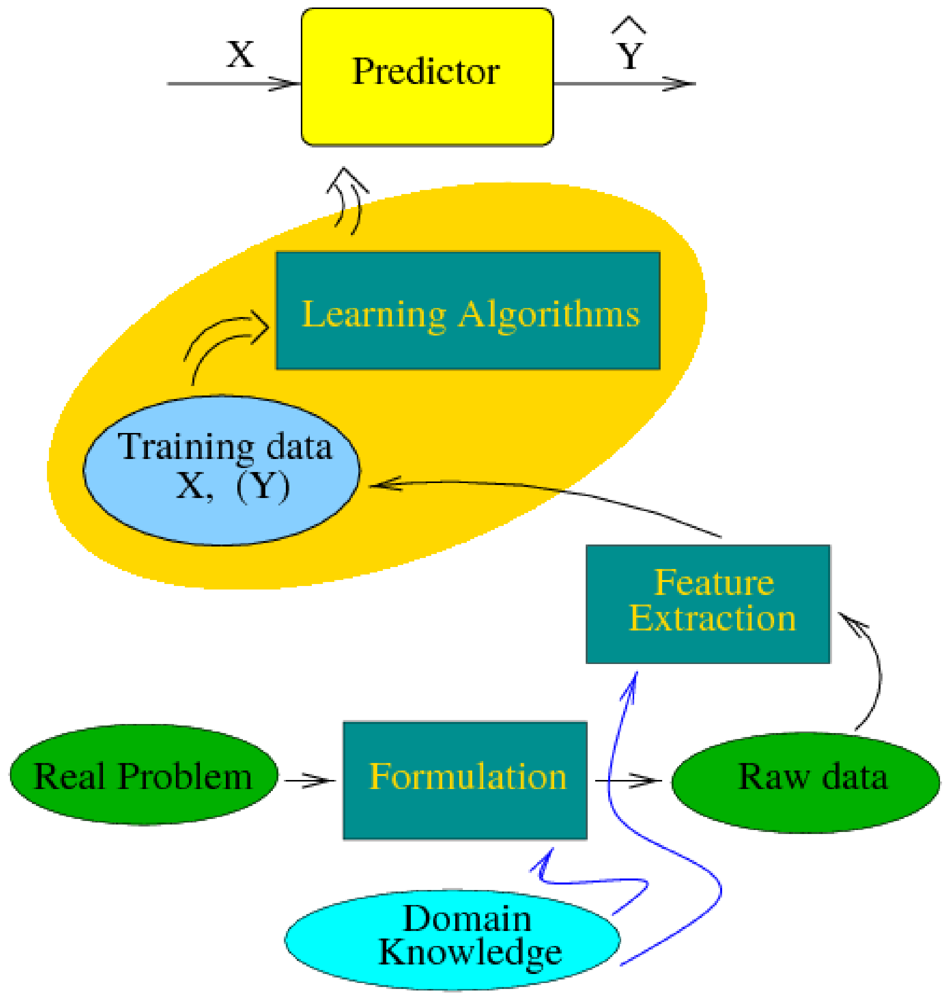
- Suppose we observe the output space Y_i and the input space $X_i = (X_{i1}, \ldots, X_{ip}) \; \forall \; i = 1, \ldots, n$

- We believe that there is a *relationship* between *Y* and at least one of the *X*'s.

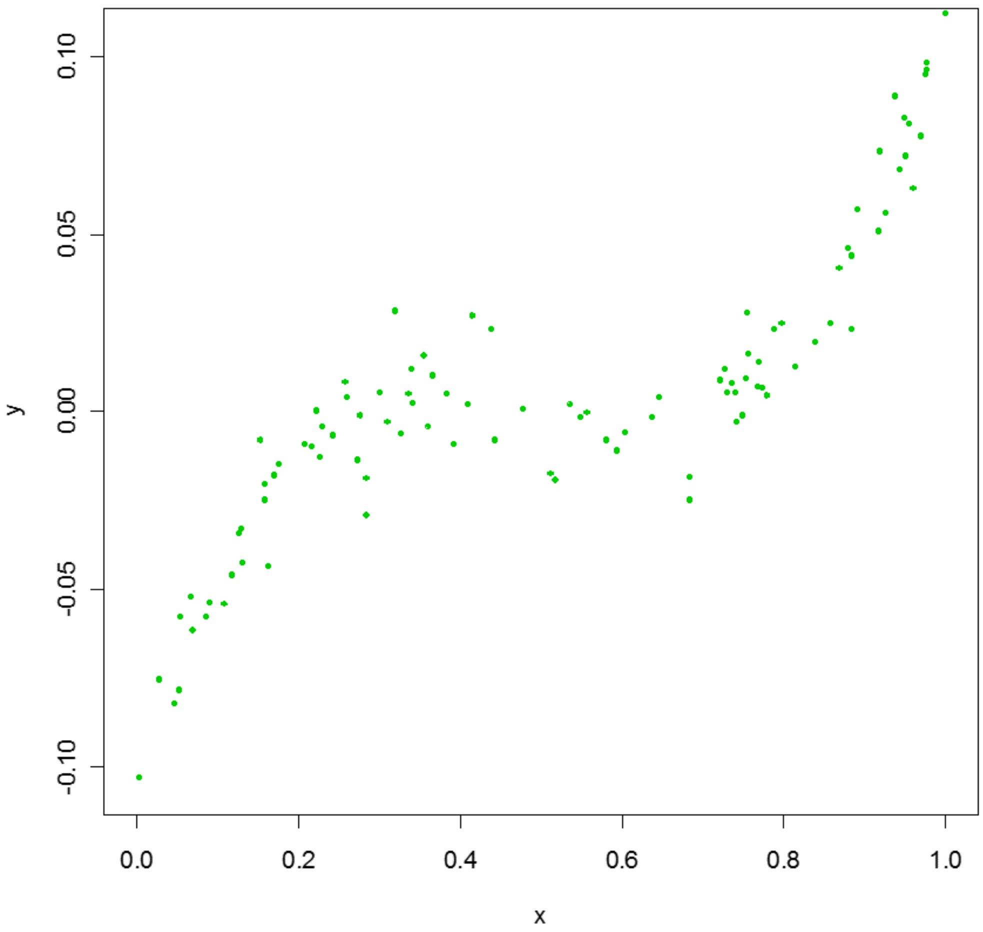
- We can model the relationship as: $Y_i = f(\boldsymbol{X_i}) + \varepsilon_i$

where *f* is an unknown function and *ε* is a random error (noise) term, independent of ***X*** with mean zero.

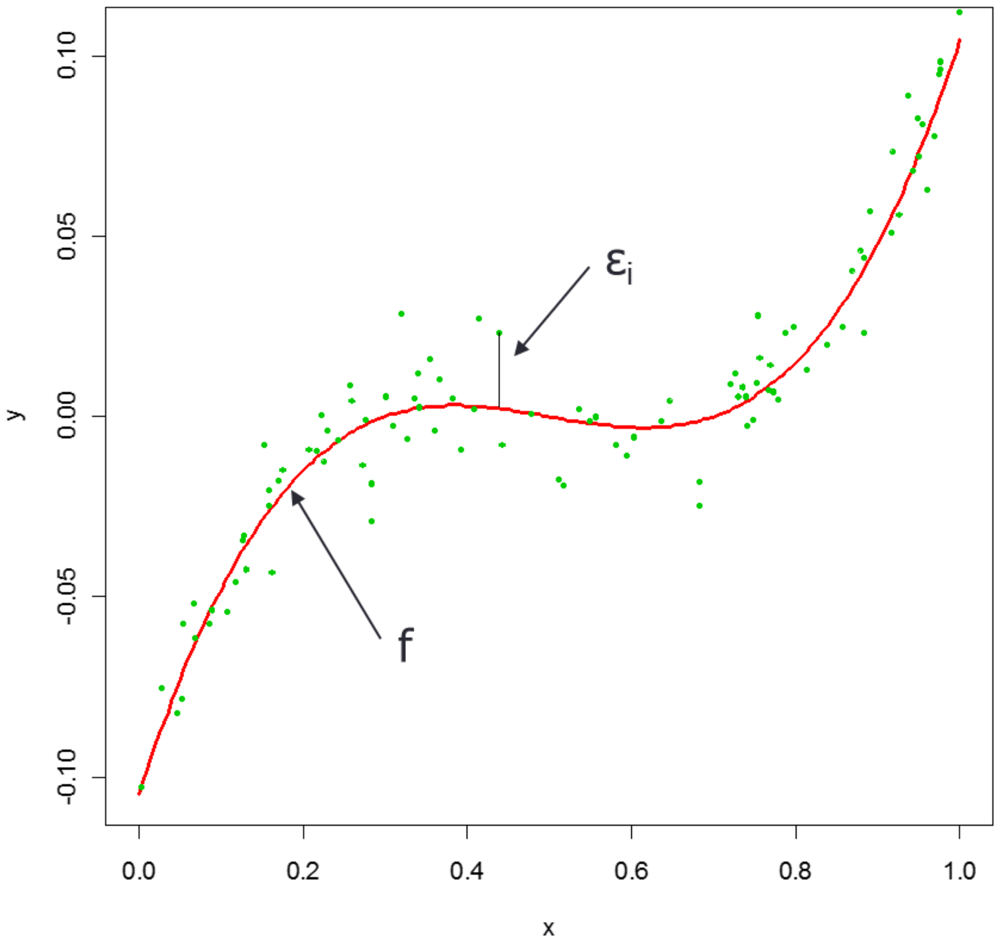
The Learning Problem (cont.)



The Learning Problem: Example

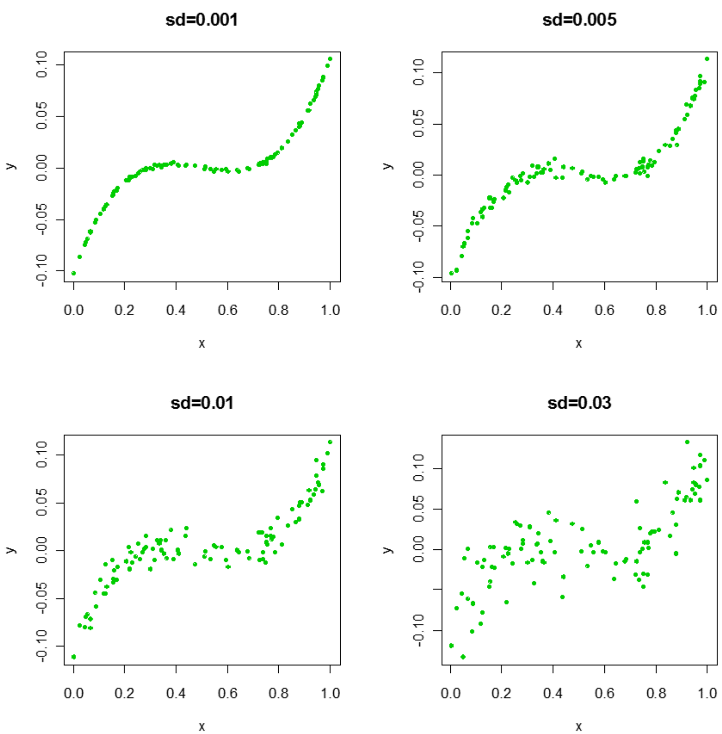


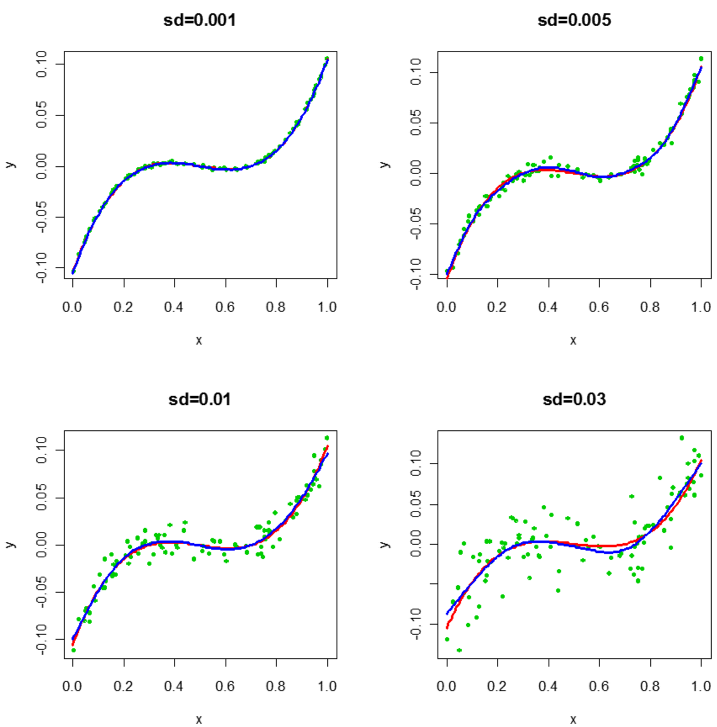
The Learning Problem: Example (cont.)



The Learning Problem: Example (cont.)

- Different estimates for the target function *f* that depend on the standard deviation of the ε's





Why do we estimate *f*?

- We use modern machine learning methods to estimate *f* by *learning* from the data.

- The target function *f* is unknown.

- We estimate *f* for two key purposes:
  - Prediction
  - Inference

Prediction

- By producing a good estimate for f where the variance of ε is not too large, then we can make accurate predictions for the response variable, Y, based on a new value of $\boldsymbol{X}$.

- We can predict Y using $\hat{Y} = \hat{f}(\boldsymbol{X})$

where $\hat{f}$ represents our estimate for f, and $\hat{Y}$ represents the resulting prediction for Y.

Prediction (cont.)

- The accuracy of $\hat{Y}$ as a prediction for Y depends on:
  - Reducible error
  - Irreducible error

- Note that $\hat{f}$ will not be a perfect estimate for f; this inaccuracy introduces error.

Prediction (cont.)

- This error is *reducible* because we can potentially improve the accuracy of the estimated (i.e. hypothesis) function $\hat{f}$ by using the most appropriate learning technique to estimate the target function *f*.

- Even if we could perfectly estimate *f*, there is still variability associated with ε that affects the accuracy of predictions = *irreducible* error.

Prediction (cont.)

- Average of the squared difference between the predicted and actual value of *Y*.

- Var(ε) represents the *variance* associated with ε.

$$E[(Y - \hat{f}(X))^2 | X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{\textit{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\textit{Irreducible}}$$

- Our aim is to minimize the reducible error!!

Example: Direct Mailing Prediction

- We are interested in predicting how much money an individual will donate based on observations from 90,000 people on which we have recorded over 400 different characteristics.
- We don't care too much about each individual characteristic.
- Learning Problem:
  - For a given individual, should I send out a mailing?

Inference

- Instead of prediction, we may also be interested in the type of relationship between Y and the X's.

- Key questions:
  - Which predictors actually affect the response?
  - Is the relationship positive or negative?
  - Is the relationship a simple linear one or is it more complicated?

Example: Housing Inference

- We wish to predict median house price based on numerous variables.

- We want to *learn* which variables have the largest effect on the response and how big the effect is.

- For example, how much impact does the number of bedrooms have on the house value?

How do we estimate *f*?

- First, we assume that we have observed a set of **training data**.

$$\{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n)\}$$

- Second, we use the training data and a **machine learning method** to estimate *f*.
  - Parametric or non-parametric methods

Parametric Methods

- This reduces the *learning problem* of estimating the target function *f* down to a problem of estimating a set of **parameters**.

- This involves a two-step approach…

Parametric Methods (cont.)

- **Step 1:**
  - Make some assumptions about the functional form of *f*. The most common example is a linear model:

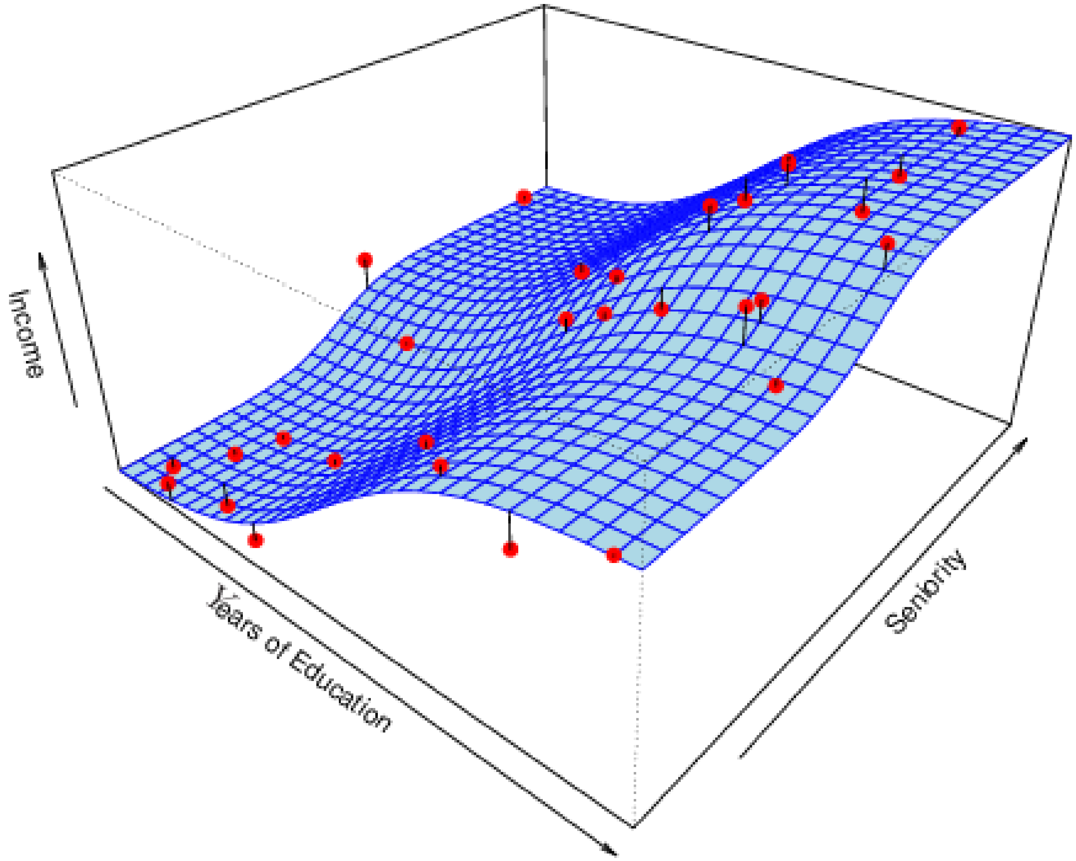
$$f(\mathbf{X}_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}$$

  - In this course, we will examine far more complicated and flexible models for *f*.

Parametric Methods (cont.)

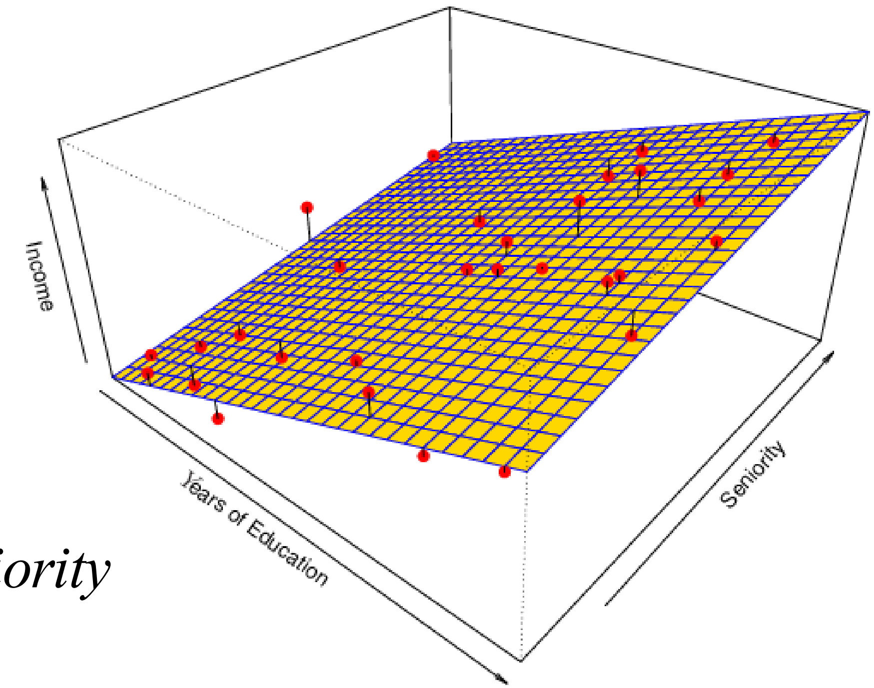
- **Step 2:**
  - We use the *training data* to fit the model (i.e. estimate *f*….the unknown parameters).
  - The most common approach for estimating the parameters in a linear model is via ordinary least squares (OLS) linear regression.
  - However, there are superior approaches, as we will see in this course.

Example: Income vs. Education Seniority



Example: OLS Regression Estimate

- Even if the standard deviation is low, we will still get a bad answer if we use the incorrect model.



$$f = b_0 + b_1 \times Education + b_2 \times Seniority$$

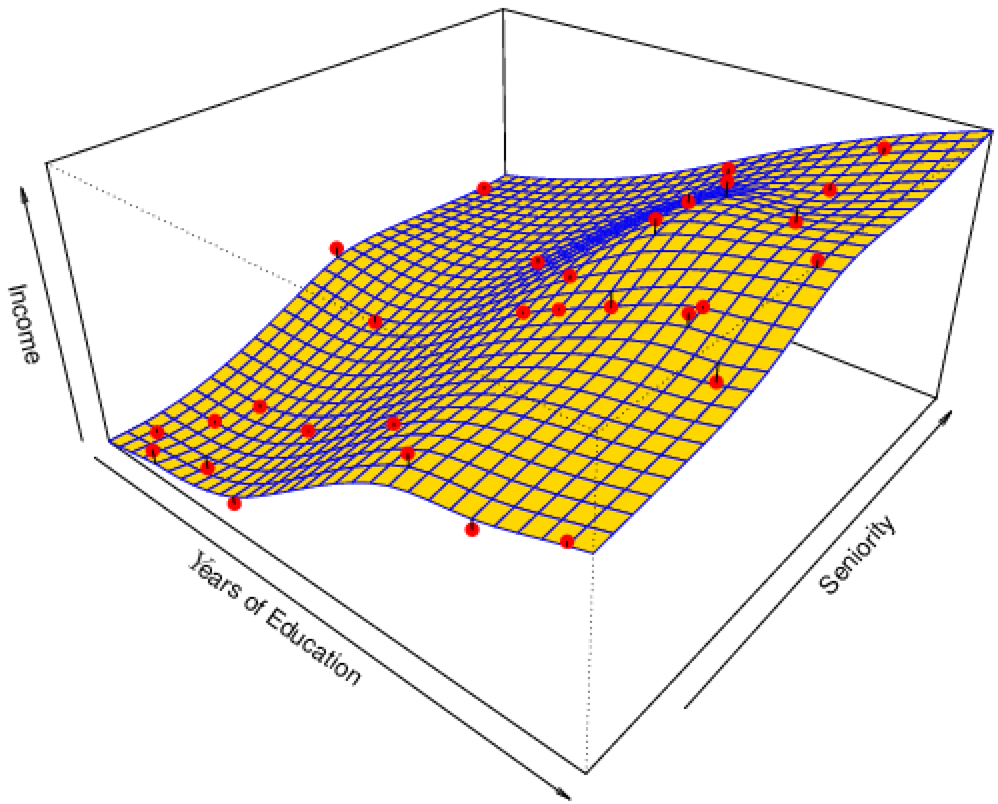
Non-Parametric Methods

- As opposed to parametric methods, these do not make explicit assumptions about the functional form of f.

- <u>Advantages</u>:
  - Accurately fit a wider range of possible shapes of f.

- <u>Disadvantages</u>:
  - Requires a very large number of observations to acquire an accurate estimate of f.

Example: Thin-Plate Spline Estimate



- Non-linear regression methods are more flexible and can potentially provide more accurate estimates.
- However, these methods can run the risk of over-fitting the data (i.e. follow the errors, or noise, too closely), so too much flexibility can produce poor estimates for *f*.

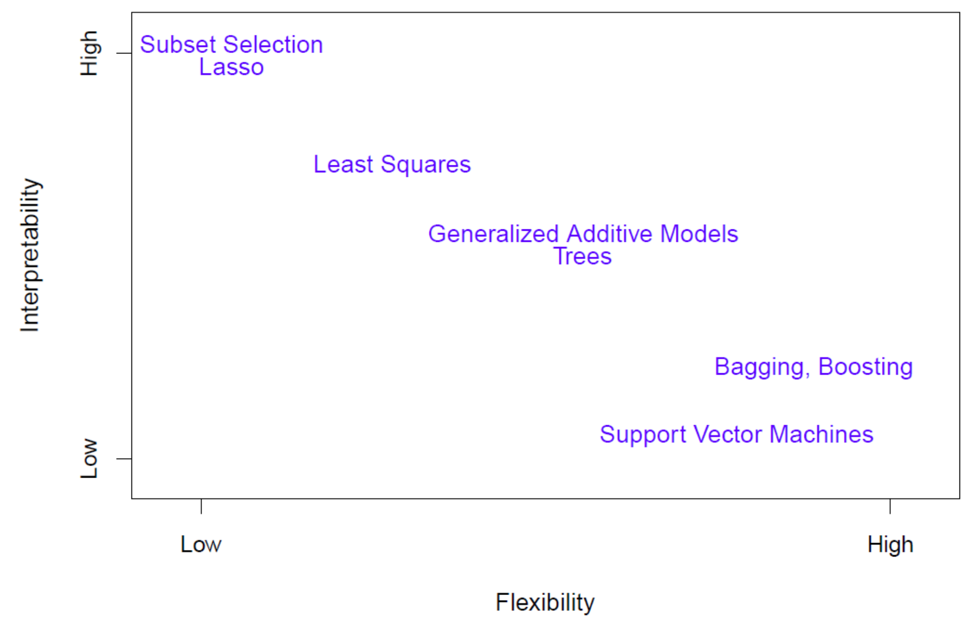
Predictive Accuracy vs. Interpretability

- Conceptual Question:
  - Why not just use a more flexible method if it is more realistic?

- Reason 1:
  - A simple method (such as OLS regression) produces a model that is easier to interpret (especially for inference purposes).

Predictive Accuracy vs. Interpretability (cont.)

- <u>Reason 2</u>:
  - Even if the primary purpose of learning from the data is for prediction, it is often possible to get more accurate predictions with a simple rather than a complicated model.



Learning Algorithm Trade-off

- There are always two aspects to consider when designing a learning algorithm:
  - Try to fit the data well
  - Be as robust as possible

- The predictor that you have generated using your training data must also work well on new data.

Learning Algorithm Trade-off (cont.)

- When we create predictors, usually the simpler the predictor is, the more robust it tends to be in the sense of begin able to be estimated reliably.

- On the other hand, the simple models do not fit the training data aggressively.

Learning Algorithm Trade-off (cont.)

- Training Error vs. Testing Error:
  - Training error → reflects whether the data fits well
  - Testing error → reflects whether the predictor actually works on new data

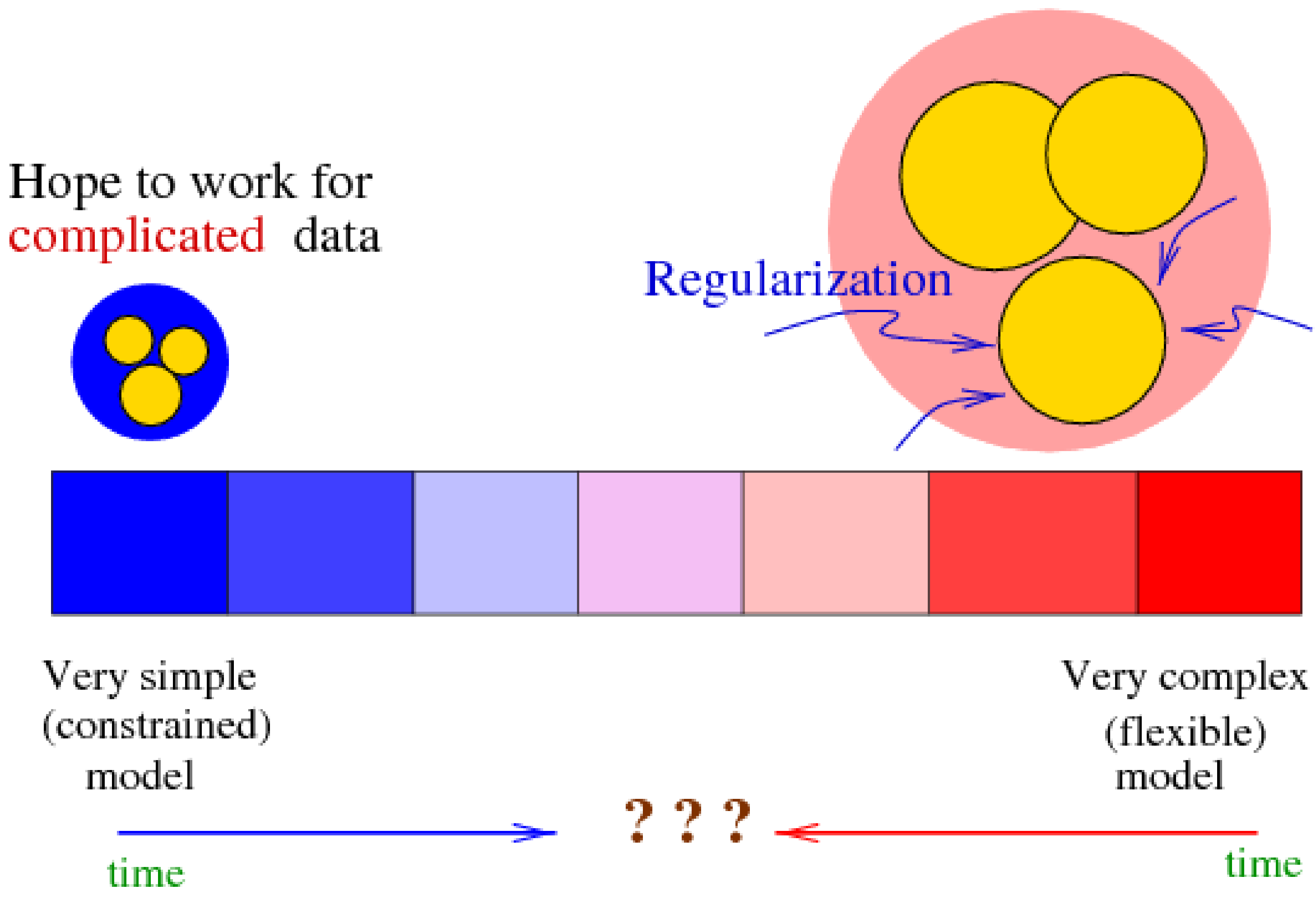
- Bias vs. Variance:
  - Bias → how good the predictor is, on average; tends to be smaller with more complicated models
  - Variance → tends to be higher for more complex models

Learning Algorithm Trade-off (cont.)

- <u>Fitting vs. Over-fitting</u>:
  - If you try to fit the data too aggressively, then you may over-fit the training data. This means that the predictors works very well on the training data, but is substantially worse on the unseen test data.

- <u>Empirical Risk vs. Model Complexity</u>:
  - Empirical risk → error rate based on the training data
  - Increase model complexity = decrease empirical risk but less robust (higher variance)

Learning Spectrum



Supervised vs. Unsupervised Learning

- <u>Supervised Learning</u>:
  - All the predictors, $\boldsymbol{X}_i$, and the response, Y_i, are observed.
    - Many regression and classification methods
- <u>Unsupervised Learning</u>:
  - Here, only the $\boldsymbol{X}_i$'s are observed (not Y_i's).
  - We need to use the $\boldsymbol{X}_i$'s to guess what Y would have been, and then build a model form there.
    - Clustering and principal components analysis

Terminology

- **Notation**
  - Input *X*: *feature, predictor,* or *independent variable*
  - Output *Y*: *response, dependent variable*

- **Categorization**
  - Supervised learning vs. unsupervised learning
    - *Key question*: Is *Y* available in the training data?
  - Regression vs. Classification
    - *Key question*: Is *Y* quantitative or qualitative?

Terminology (cont.)

- **Quantitative:**
  - Measurements or counts, recorded as numerical values (e.g. height, temperature, etc.)
- **Qualitative:** group or categories
  - <u>Ordinal</u>: possesses a natural ordering (e.g. shirt sizes)
  - <u>Nominal</u>: just name the categories (e.g. marital status, gender, etc.)

Terminology (cont.)

Training Samples

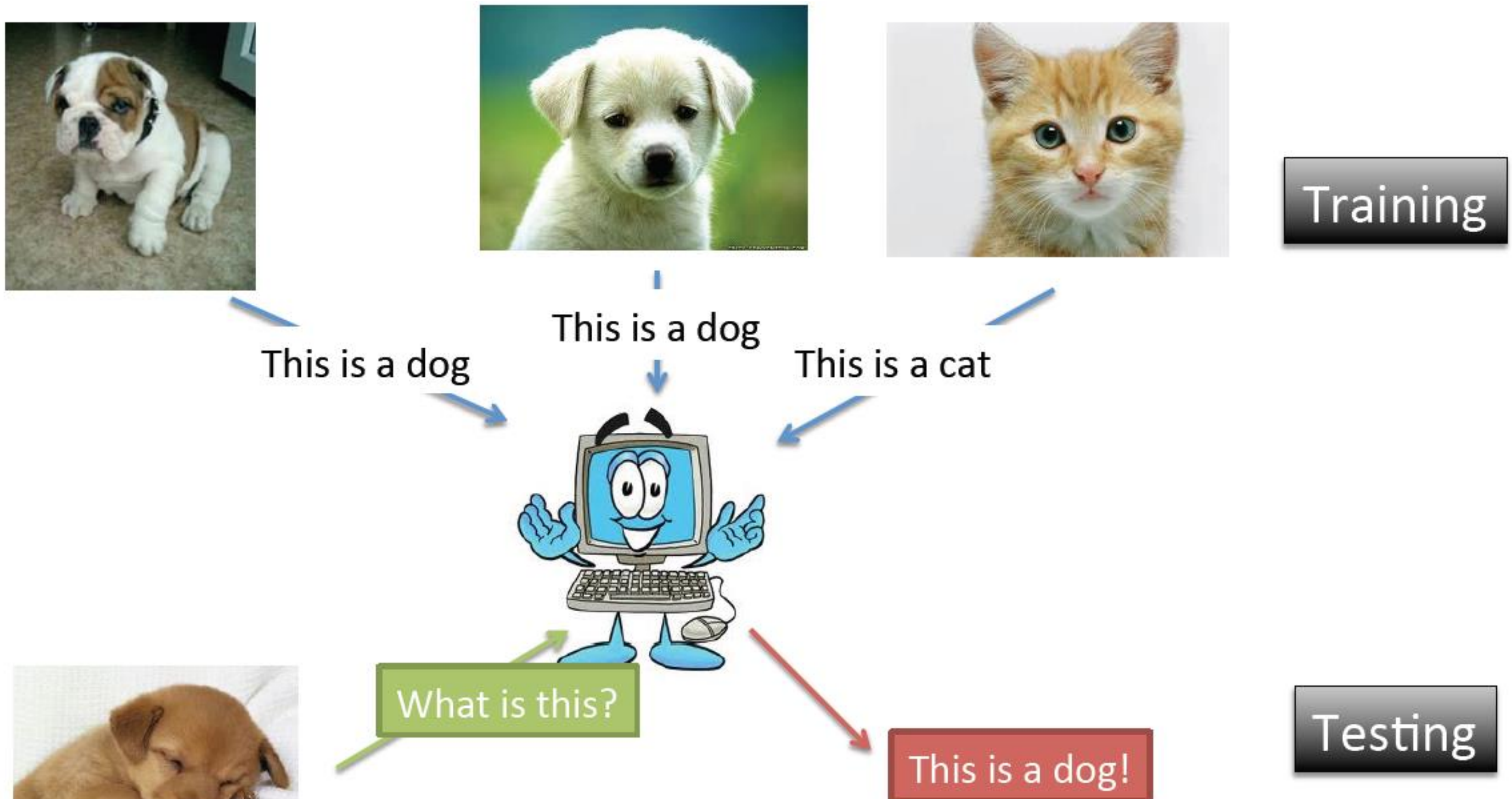
Feature X					Label Y
x_1	**x_2**	**x_3**	**…**	**x_p**	**Y**
3	5	2	…	1	**A**
4	2	3	…	2	**B**
…	…	…	…	…	**…**
4	2	3	…	3	**A**

Model

Testing

Feature X					Label Y (unknown)
x_1	**x_2**	**x_3**	**…**	**x_p**	**Y**
5	5	2	…	1	**?**
2	2	1	…	2	**?**
…	…	…	…	…	**…**
1	3	2	…	4	**?**

Supervised Learning



Classification: Categorical output
Regression: Continuous output

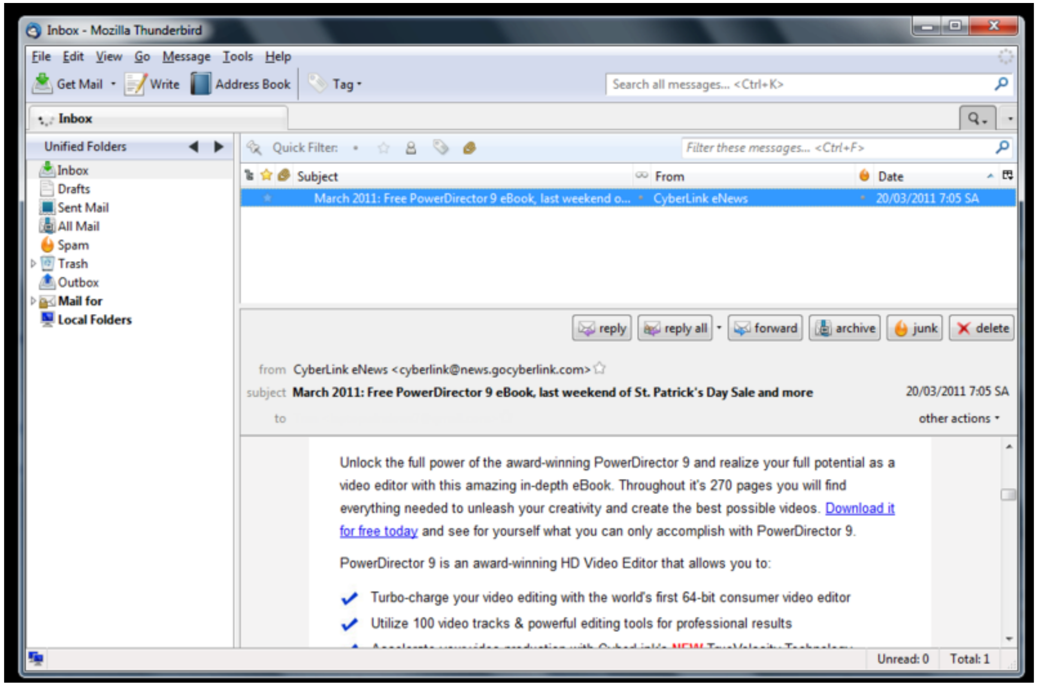
Supervised Learning: Regression vs. Classification

- <u>Regression</u>
  - Covers situations where *Y* is continuous (quantitative)
  - E.g. predicting the value of the Dow in 6 months, predicting the value of a given house based on various inputs, etc.

- <u>Classification</u>
  - Covers situations where *Y* is categorical (qualitative)
  - E.g. Will the Dow be up or down in 6 months? Is this email spam or not?

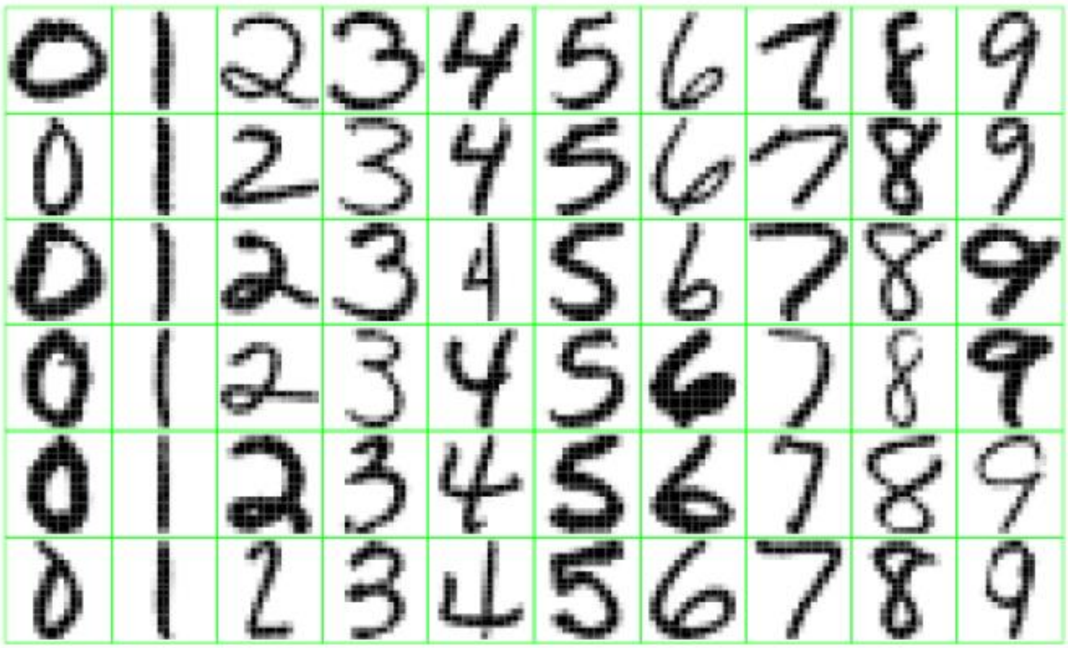
Supervised Learning: Examples

- <u>Email Spam</u>:
  - predict whether an email is a junk email (i.e. spam)



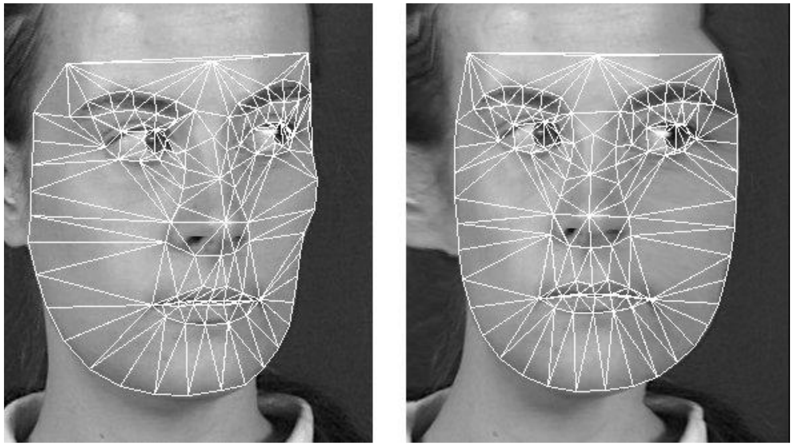
Supervised Learning: Examples

- <u>Handwritten Digit Recognition</u>:
  - Identify single digits 0~9 based on images



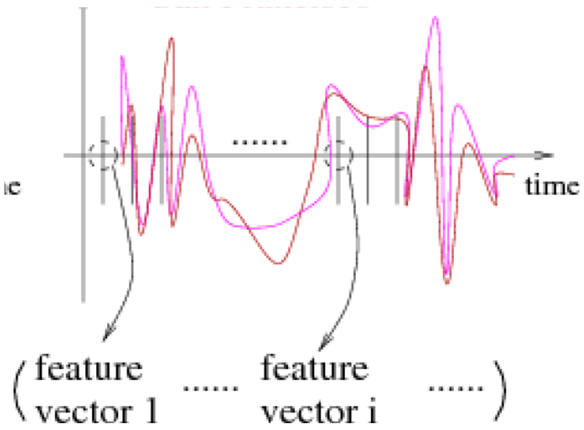
Supervised Learning: Examples

- Face Detection/Recognition:
  - Identify human faces

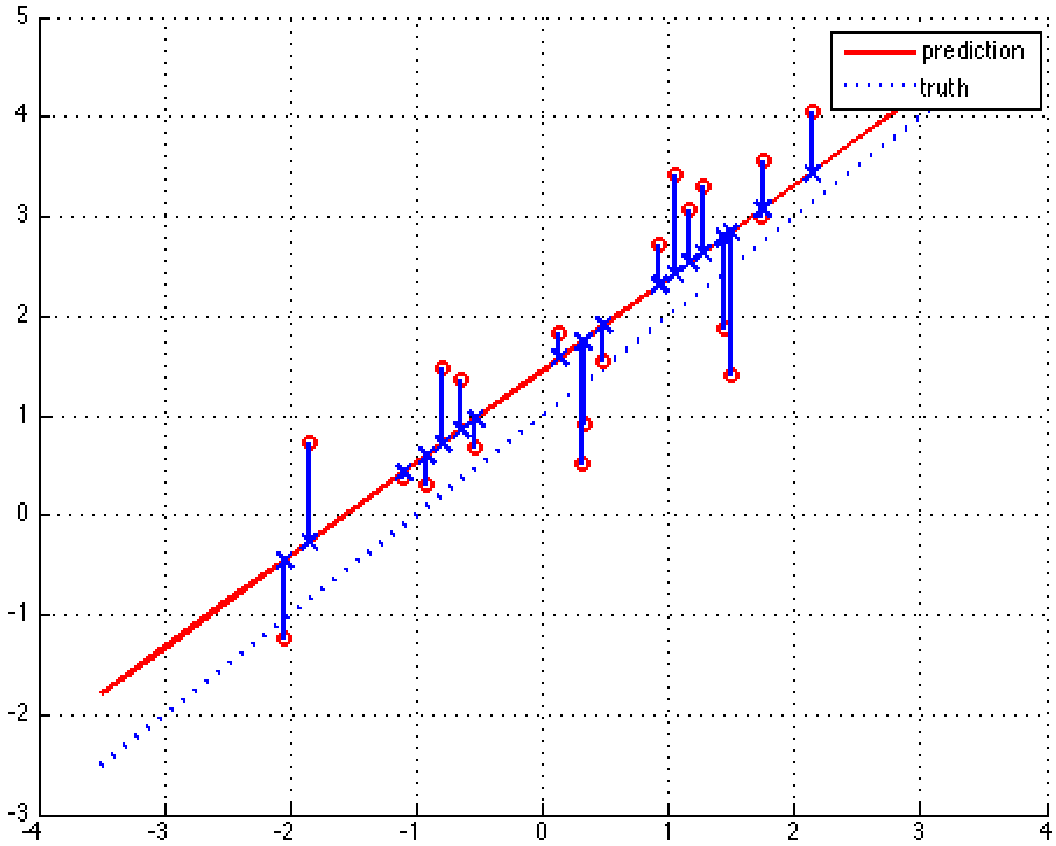


Supervised Learning: Examples

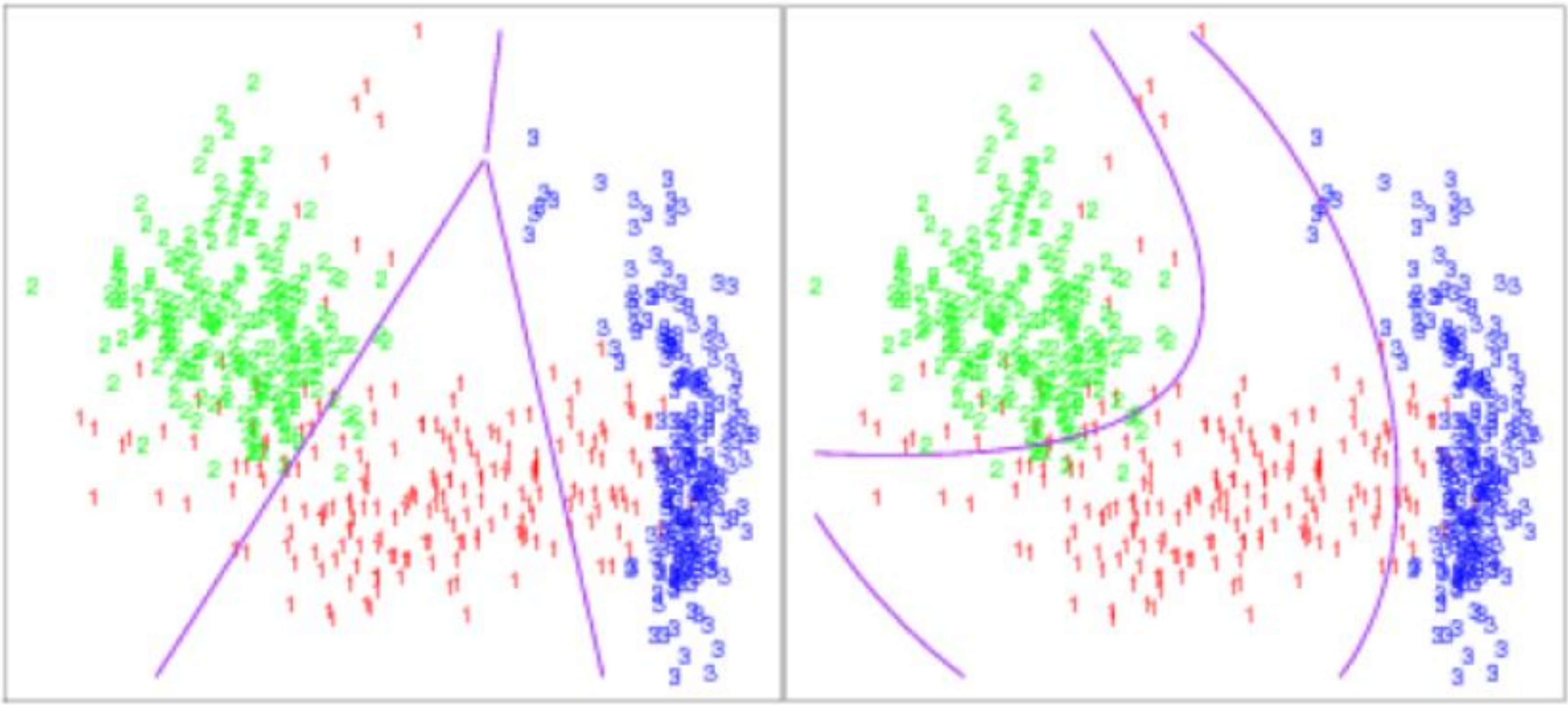
- <u>Speech Recognition</u>:
  - Identify words spoken according to speech signals
    - Automatic voice recognition systems used by airline companies, automatic stock price reporting, etc.



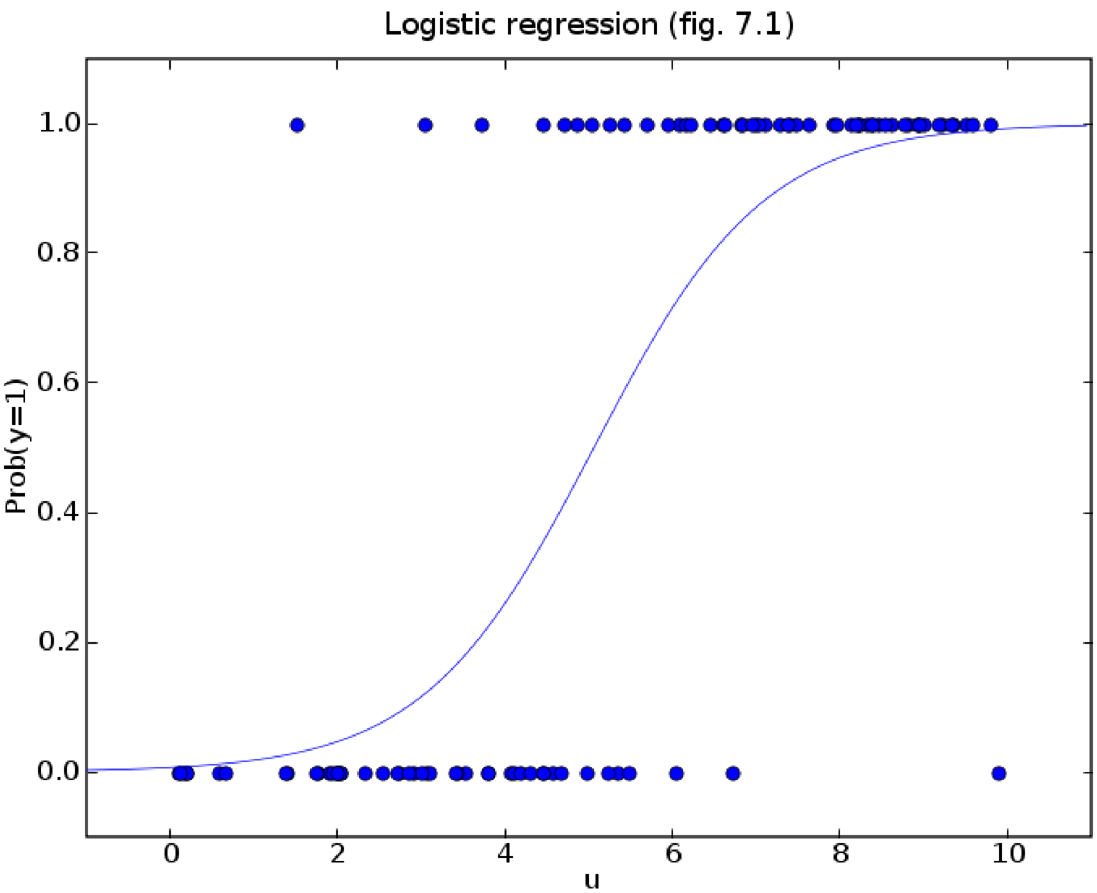
Supervised Learning: Linear Regression



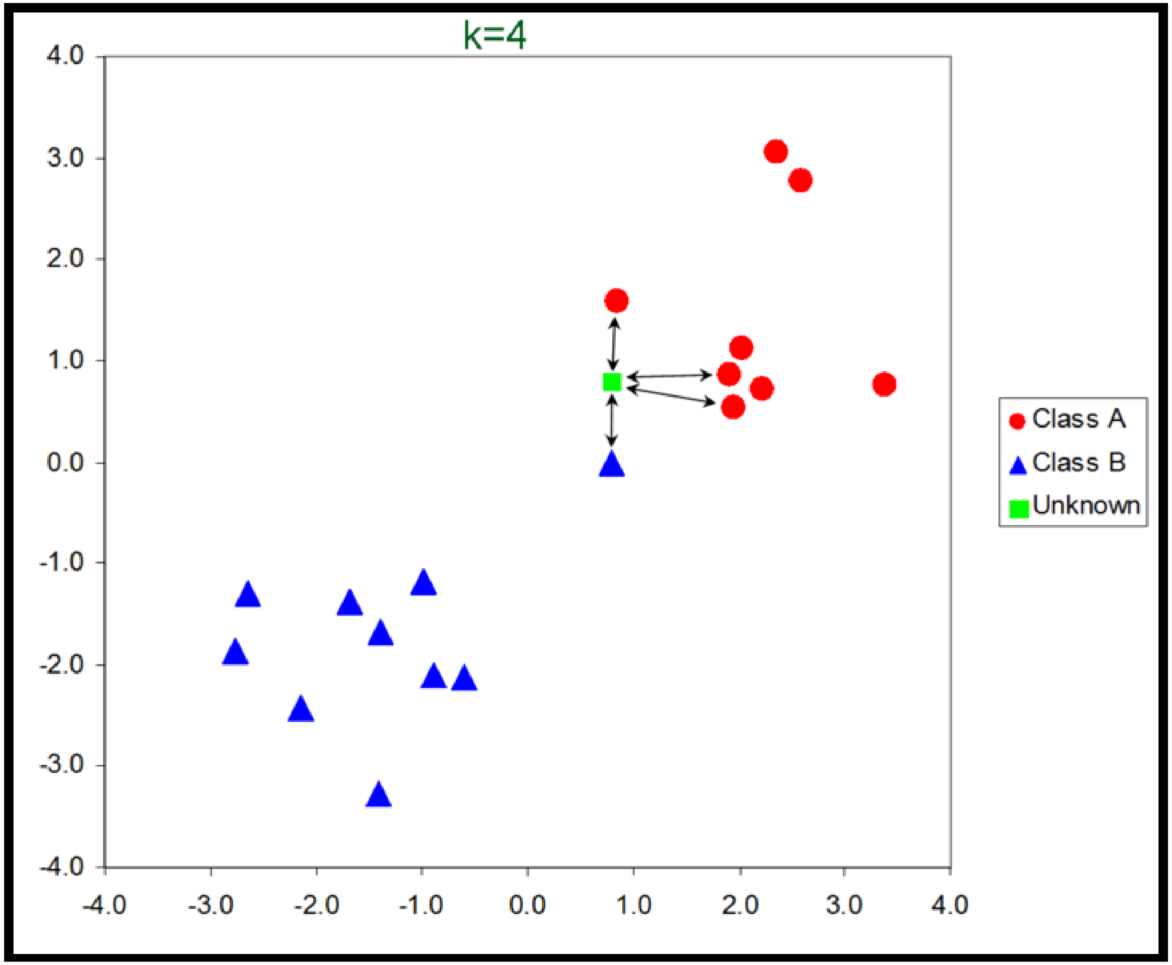
Supervised Learning: Linear/Quadratic Discriminant Analysis



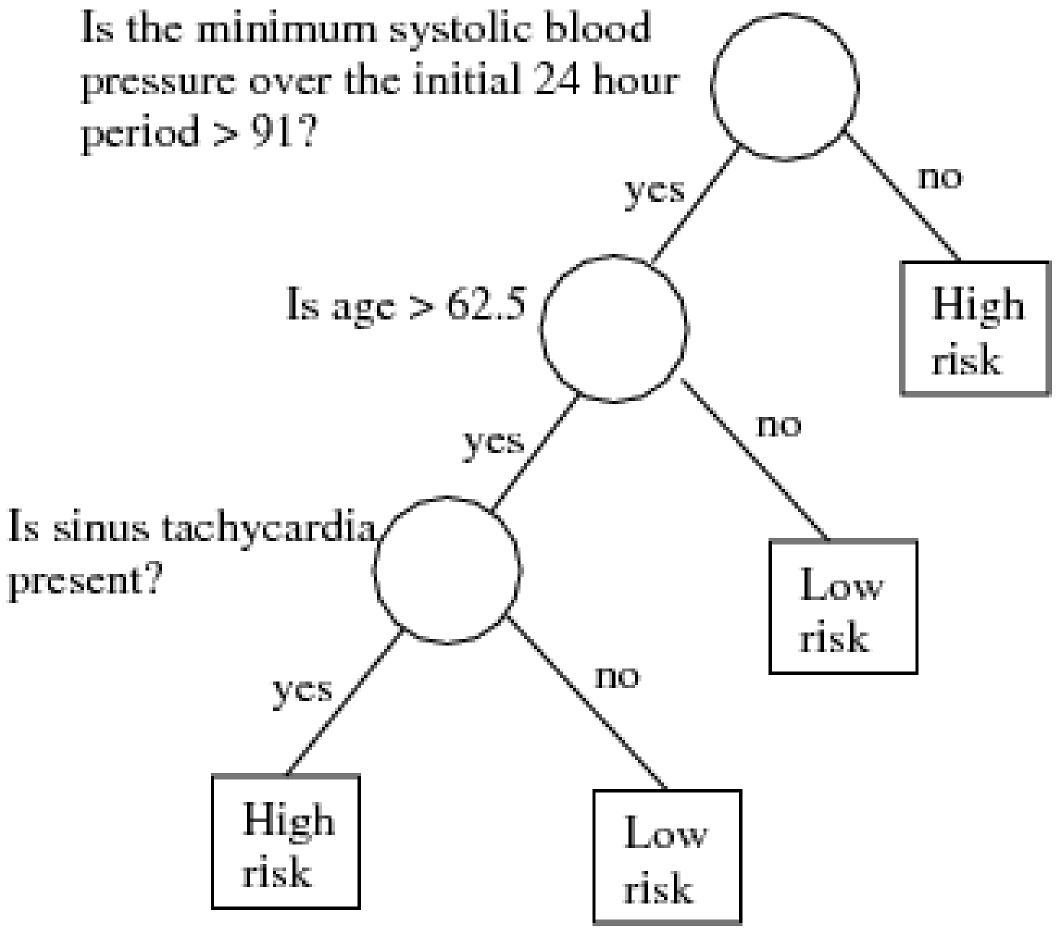
Supervised Learning: Logistic Regression



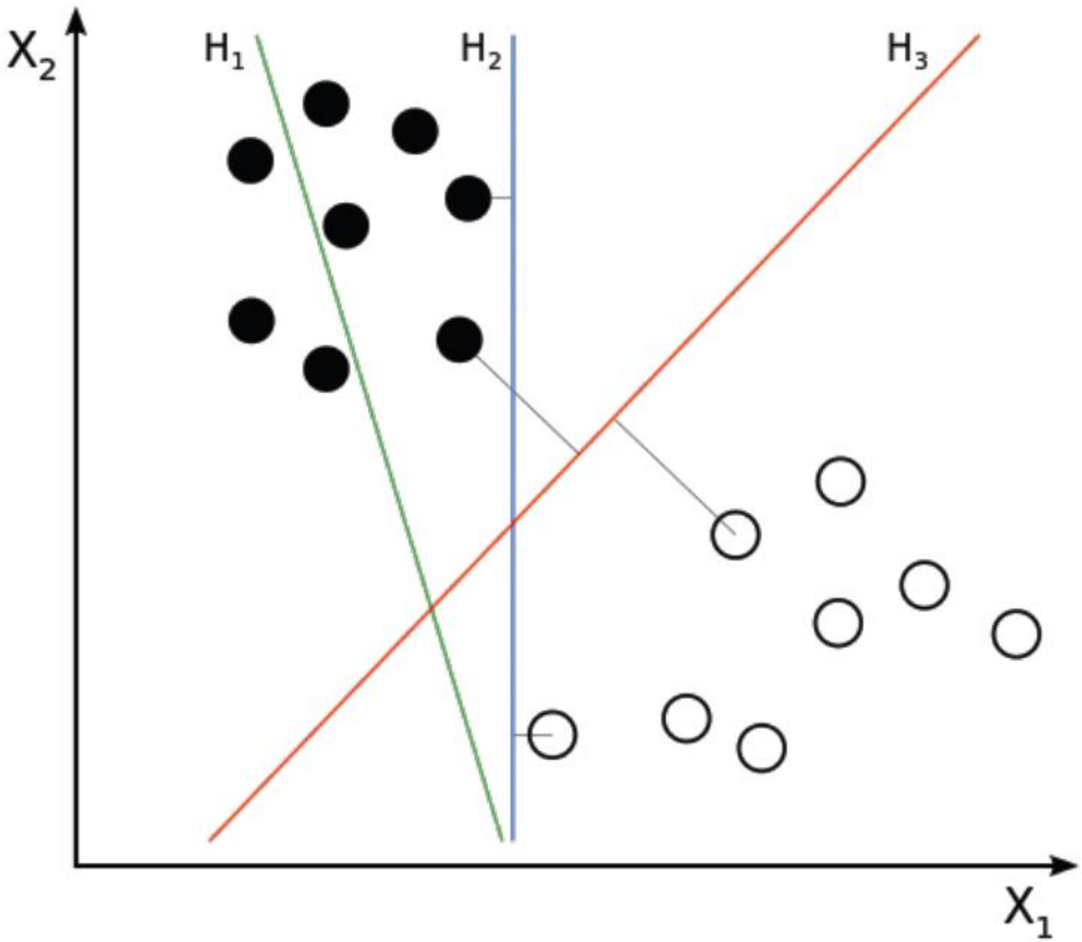
Supervised Learning: K Nearest Neighbors



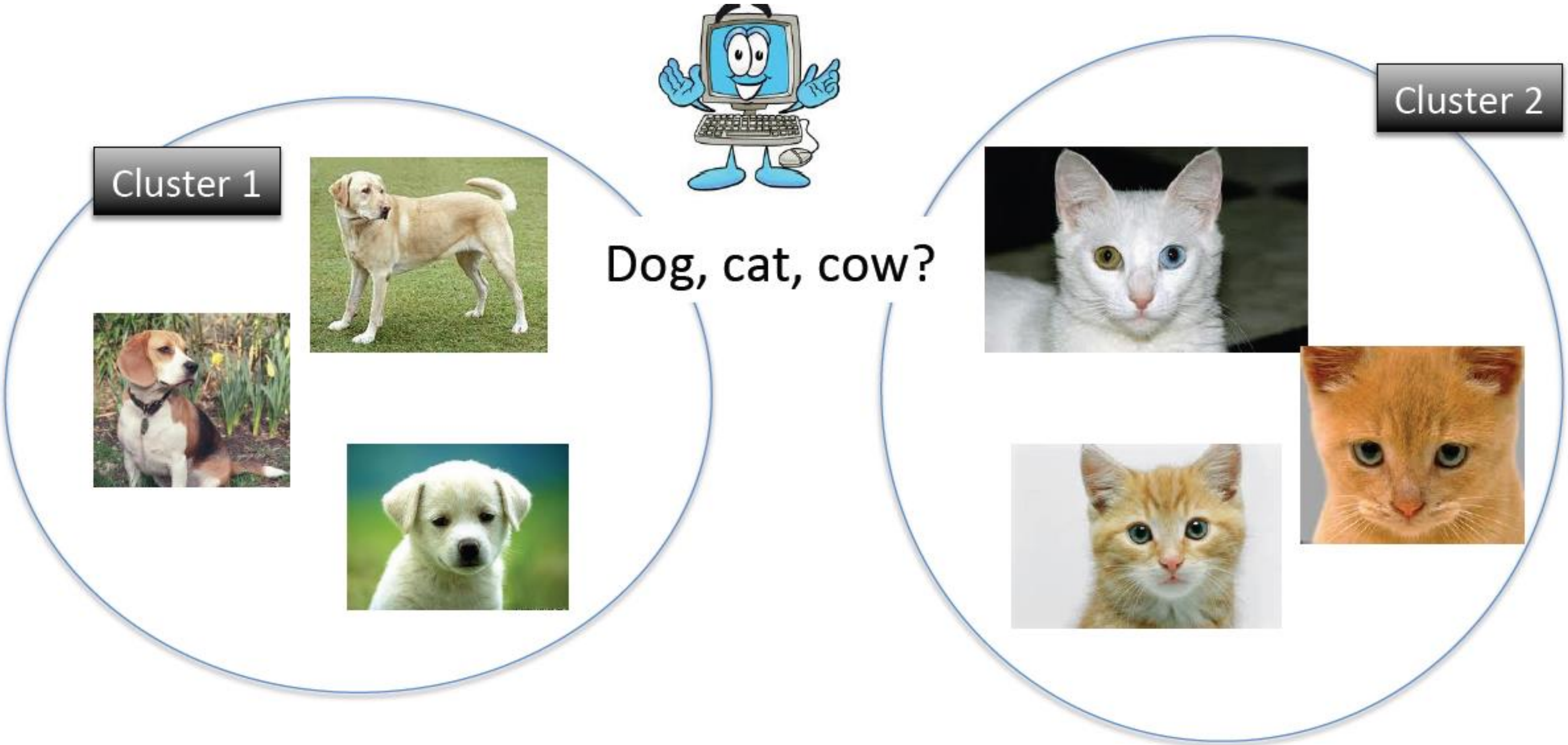
Supervised Learning: Decision Trees / CART



Supervised Learning: Support Vector Machines



Unsupervised Learning

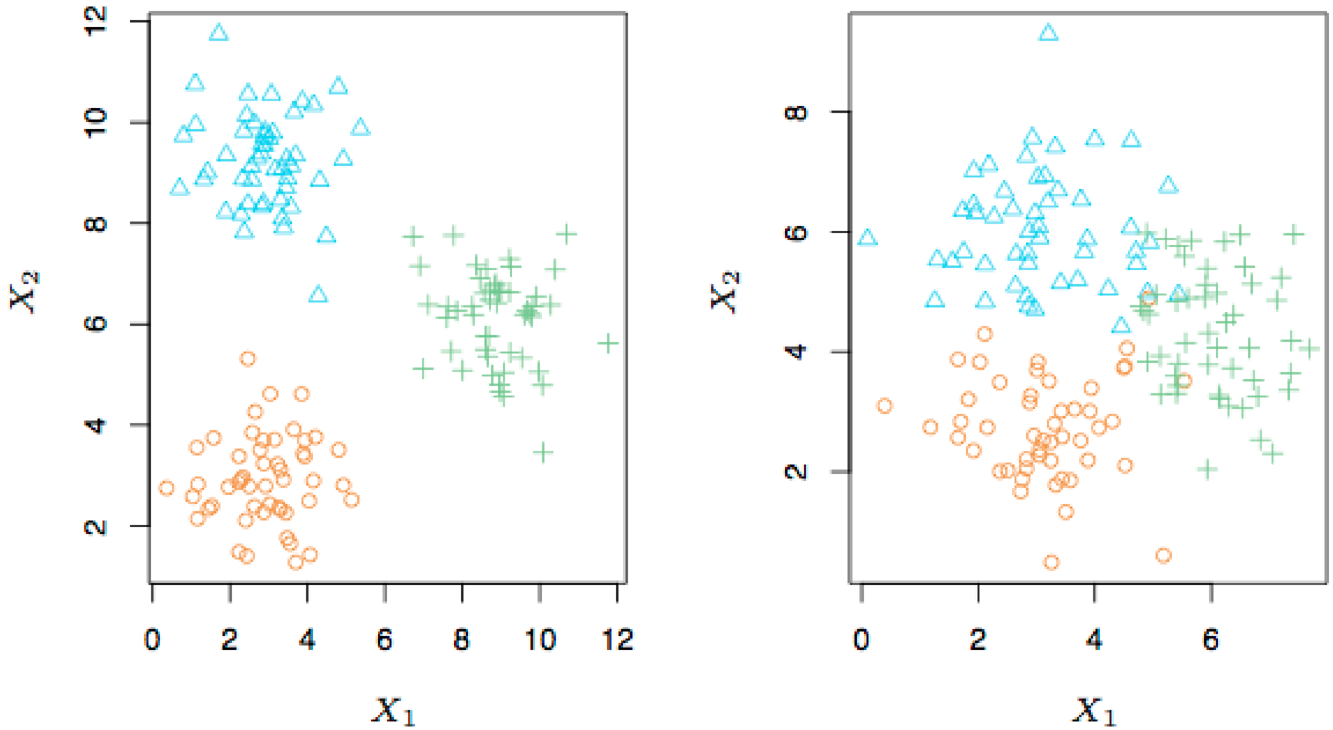


Unsupervised: semantic meanings of clusters are not clear

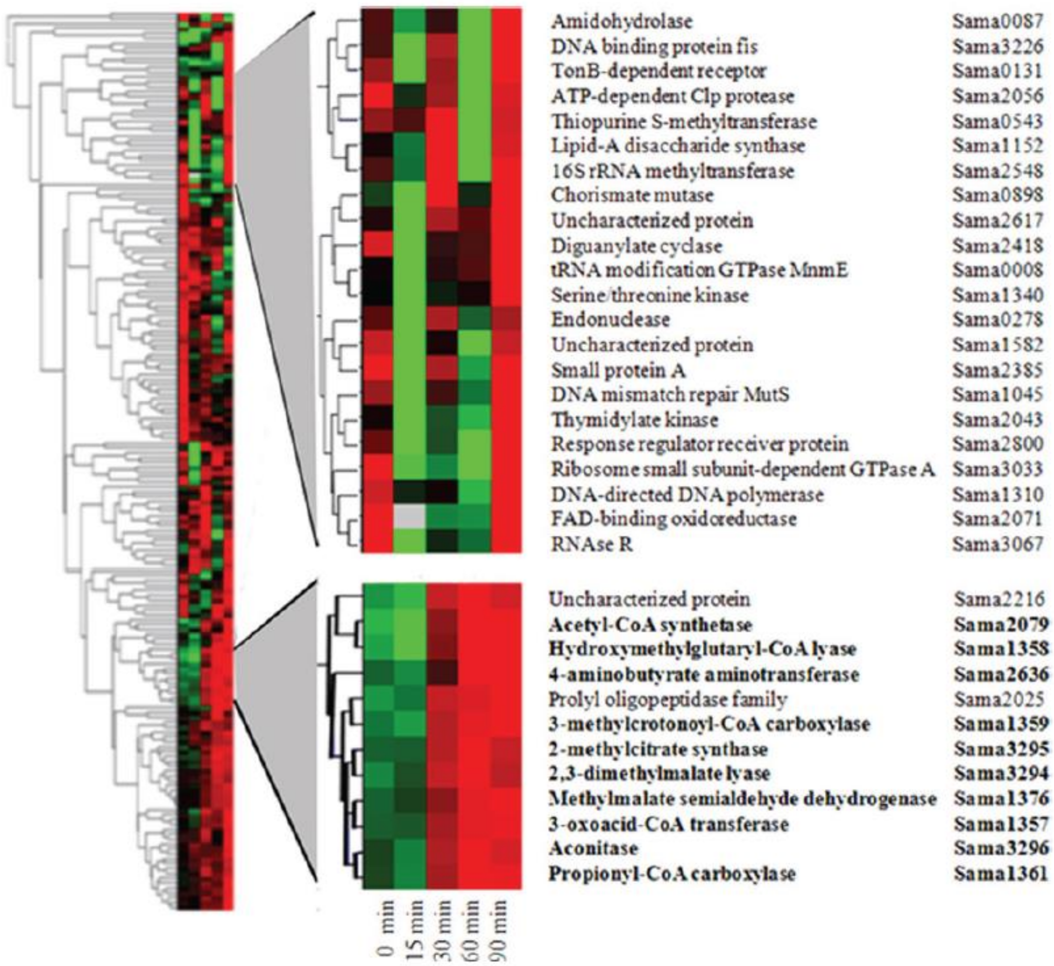
Unsupervised Learning (cont.)

- The training data does not contain any output information at all (i.e. unlabeled data).
- Viewed as the task of spontaneously finding patterns and structure in input data.
- Viewed as a way to create a higher-level representation of the data and dimension reduction.

Unsupervised Learning: K-Means Clustering



Unsupervised Learning: Hierarchical Clustering



Assessing Model Accuracy

- For a given set of data, we need to decide which machine learning method produces the **best** results.
- We need some way to measure the quality of fit (i.e. how well its predictions actually match the observed data).
- In regression, we typically use mean squared error (MSE).

Assessing Model Accuracy (cont.)

Suppose we fit a model $\hat{f}(x)$ to some training data $\mathsf{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

- We could compute the average squared prediction error over Tr:

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

This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test* data $\mathsf{Te} = \{x_i, y_i\}_1^M$:

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

Assessing Model Accuracy (cont.)

- Thus, we really care about how well the method works on new, unseen test data.

- There is no guarantee that the method with the smallest *training MSE* will have the smallest *test MSE*.

Training vs. Test MSEs

- In general, the more flexible a method is the lower its training MSE will be.

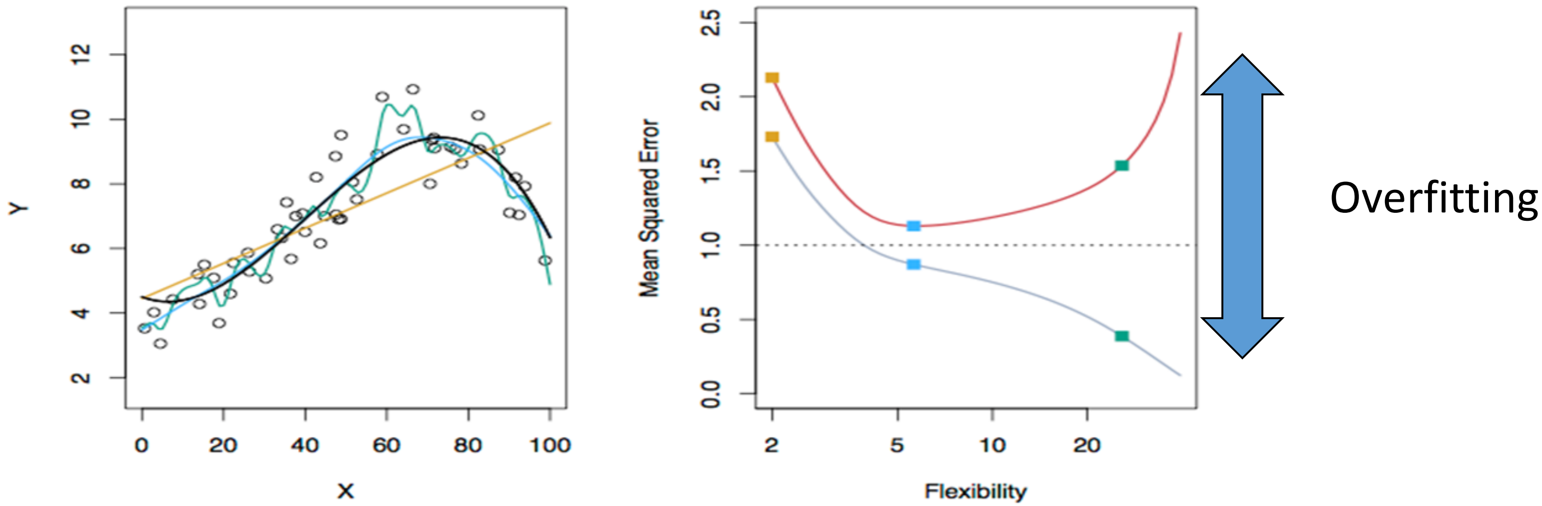
- However, the test MSE may in fact be higher for a more flexible method than for a simple approach like linear regression.

Training vs. Test MSEs (cont.)

- More flexible methods (such as splines) can generate a wider range of possible shapes to estimate f as compared to less flexible and more restrictive methods (such as linear regression).

- The less flexible the method, the easier to interpret the model. there is a trade-off between flexibility and model interpretability.

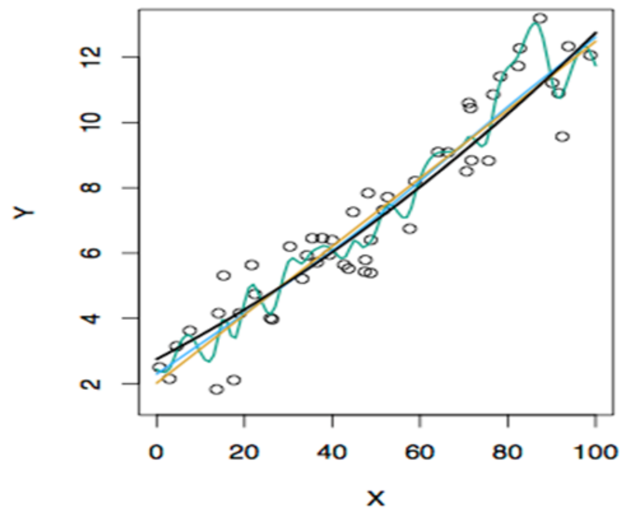
Different Levels of Flexibility

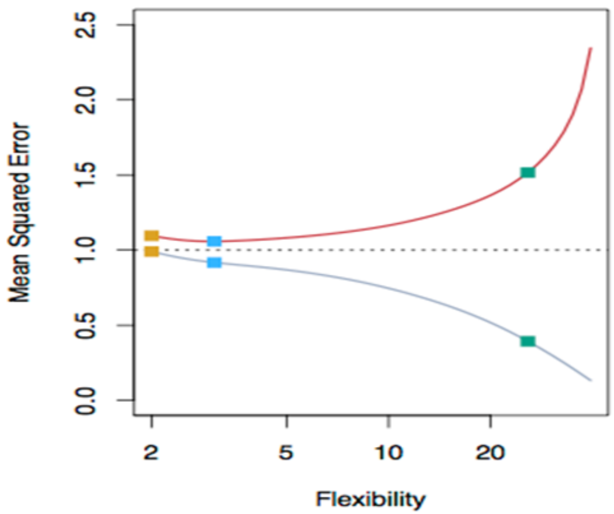


LEFT
Black: Truth
Orange: Linear estimate
Blue: Smoothing spline
Green: Smoothing spline (more flexible)

RIGHT
RED: Test MSE
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)

Different Levels of Flexibility (cont.)

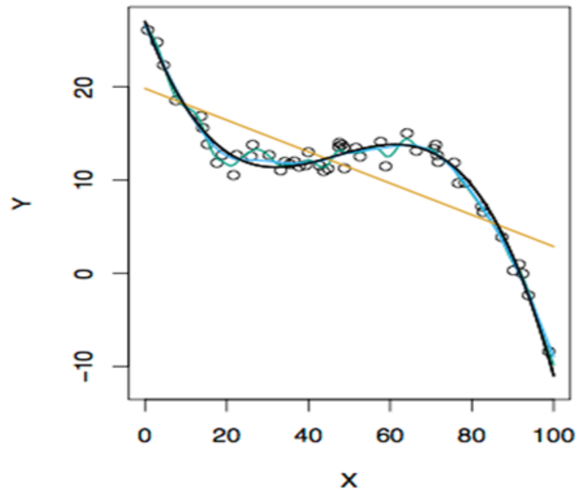


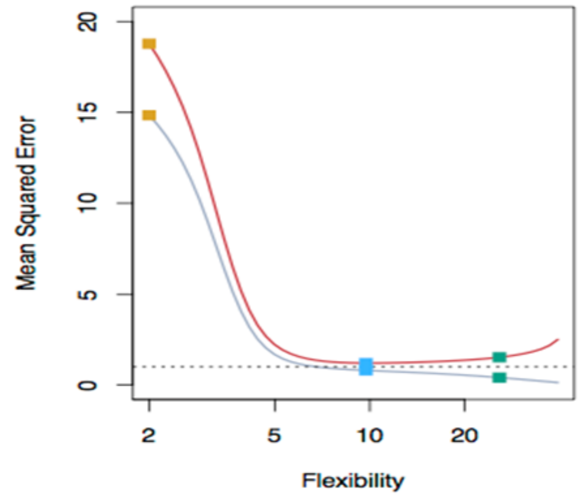


LEFT
Black: Truth
Orange: Linear estimate
Blue: Smoothing spline
Green: Smoothing spline (more flexible)

RIGHT
RED: Test MSE
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)

Different Levels of Flexibility (cont.)





<u>LEFT</u>
Black: Truth
Orange: Linear estimate
Blue: Smoothing spline
Green: Smoothing spline (more flexible)

<u>RIGHT</u>
RED: Test MSE
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)

Bias-Variance Trade-off

- The previous graphs of test versus training MSEs illustrates a very important trade-off that governs the choice of machine learning methods.

- There are always two competing forces that govern the choice of learning method:
  - bias and variance

Bias of Learning Methods

- Bias refers to the error that is introduced by modeling a real life problem (that is usually extremely complicated) by a much simpler problem.

- Generally, the more flexible/complex a machine learning method is, the **less bias** it will generally have.

Variance of Learning Methods

- Variance refers to how much your estimate for f would change by if you had a different training data set.

- Generally, the more flexible/complex a machine learning method is the **more variance** it has.

The Trade-Off: Expected Test MSE

Suppose we have fit a model $\hat{f}(x)$ to some training data Tr, and let (x_0, y_0) be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$), then

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

The expectation averages over the variability of y_0 as well as the variability in Tr. Note that $\mathrm{Bias}(\hat{f}(x_0))] = E[\hat{f}(x_0)] - f(x_0)$.

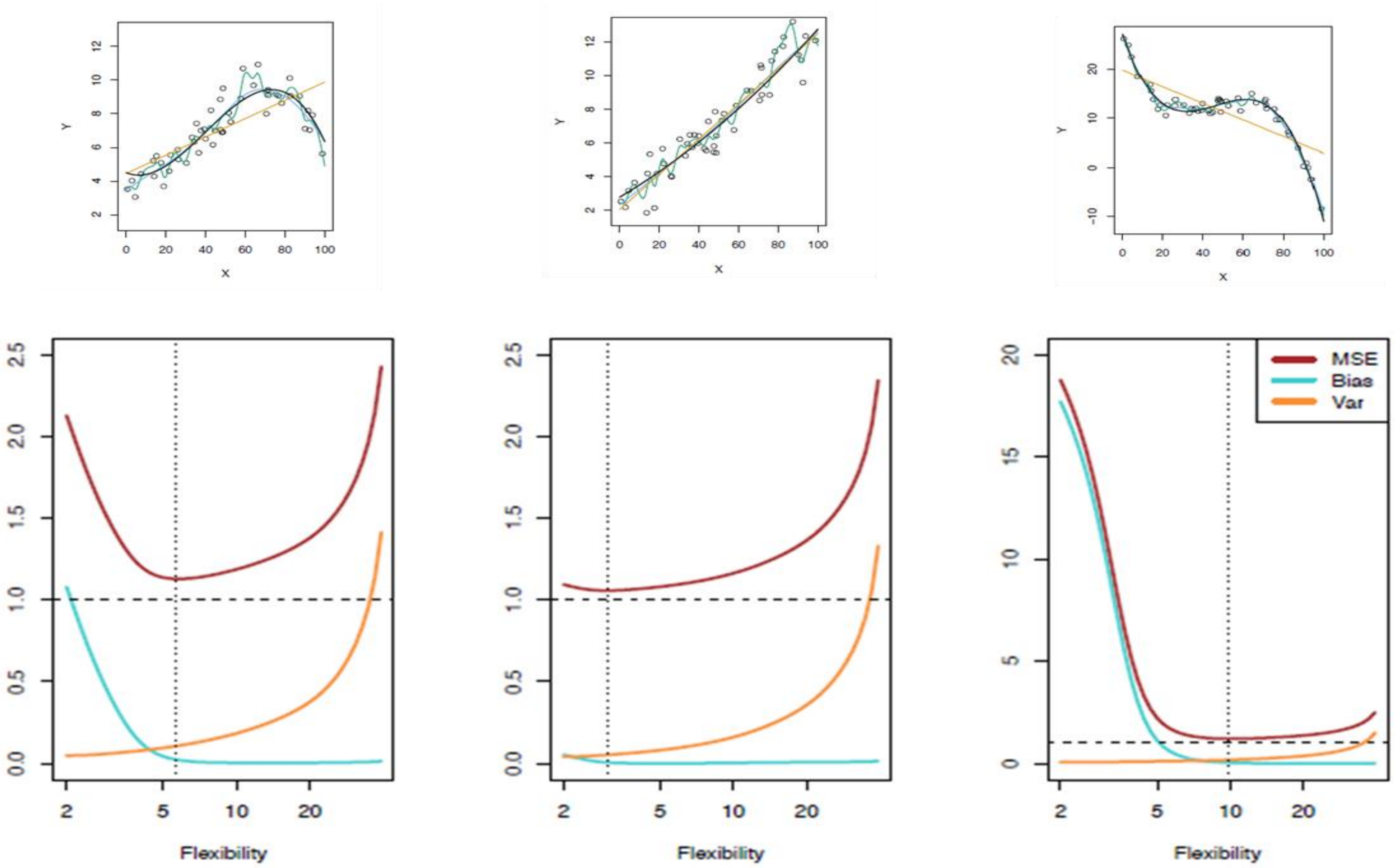
Typically as the *flexibility* of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on average test error amounts to a *bias-variance trade-off.*

Test MSE, Bias and Variance

- Thus, in order to minimize the expected test MSE, we must select a machine learning method that simultaneously achieves *low variance* and *low bias*.

- Note that the expected test MSE can never lie below the irreducible error - Var(ε).

Test MSE, Bias and Variance (cont.)



The Classification Setting

- For a classification problem, we can use the misclassification error rate to assess the accuracy of the machine learning method.

$$Error\ Rate = \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) / n$$

which represents the fraction of misclassifications.

- $I(y_i \neq \hat{y}_i)$ is an indicator function, which will give 1 if the condition $(y_i \neq \hat{y}_i)$ is correct, otherwise it gives a 0.

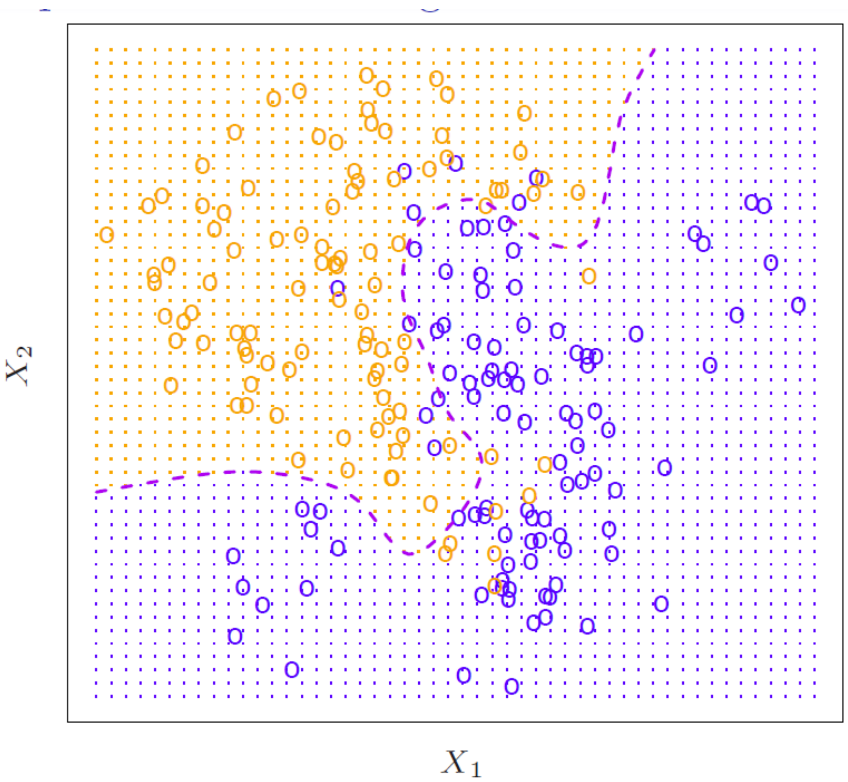
Bayes Error Rate

- The Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the "true" probability distribution of the data looked like.

- On test data, no classifier can get lower error rates than the Bayes error rate.

- In real-life problems, the Bayes error rate can't be calculated exactly.

Bayes Decision Boundary



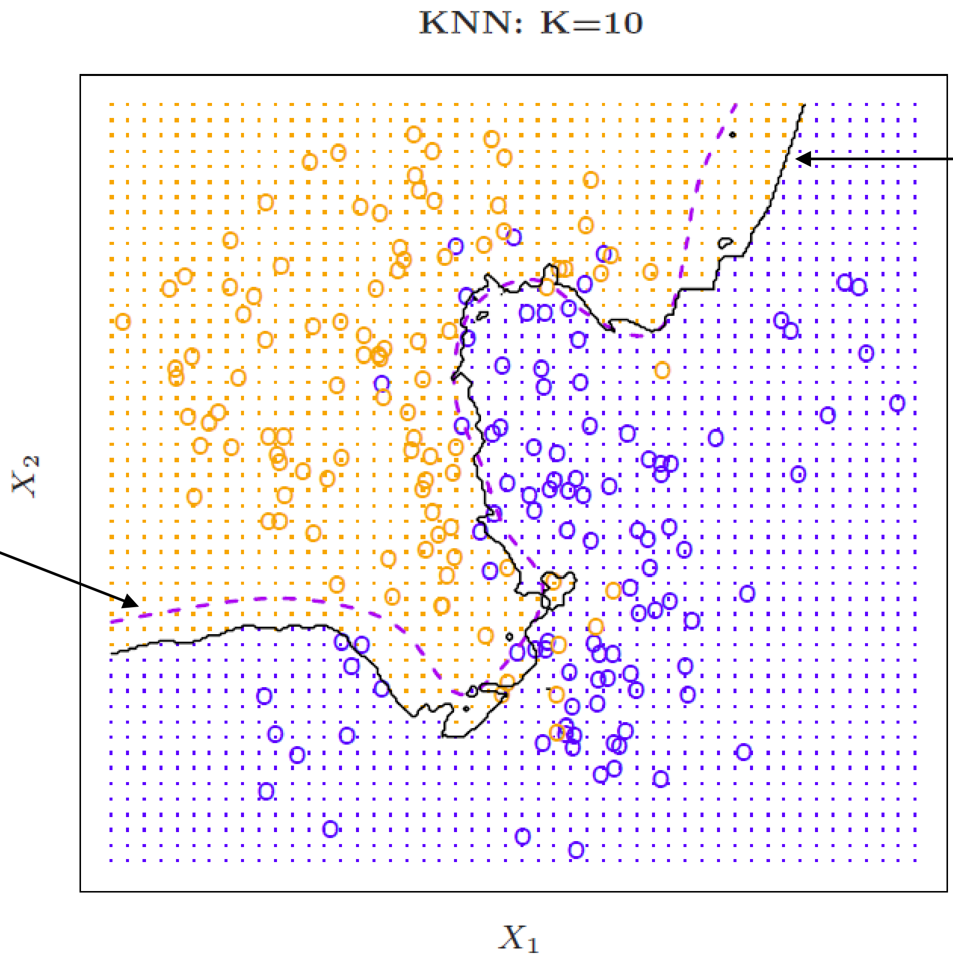
- The purple dashed line represents the points where the probability is exactly 50%.
- The Bayes classifier's prediction is determined by the Bayes decision boundary

K-Nearest Neighbors (KNN)

- KNN is a flexible approach to estimate the Bayes classifier.
- For any given X, we find the k closest neighbors to X in the training data and average their corresponding responses Y.
- If the majority of the Y's are orange, then we predict orange otherwise guess blue.
- The smaller that k is, the more flexible the method will be.

KNN: K=10

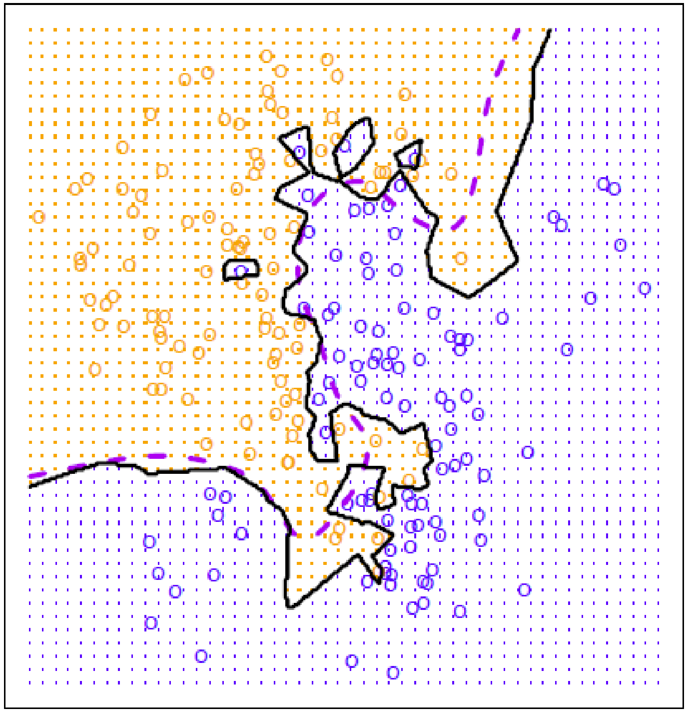
Bayes decision boundary



KNN decision boundary

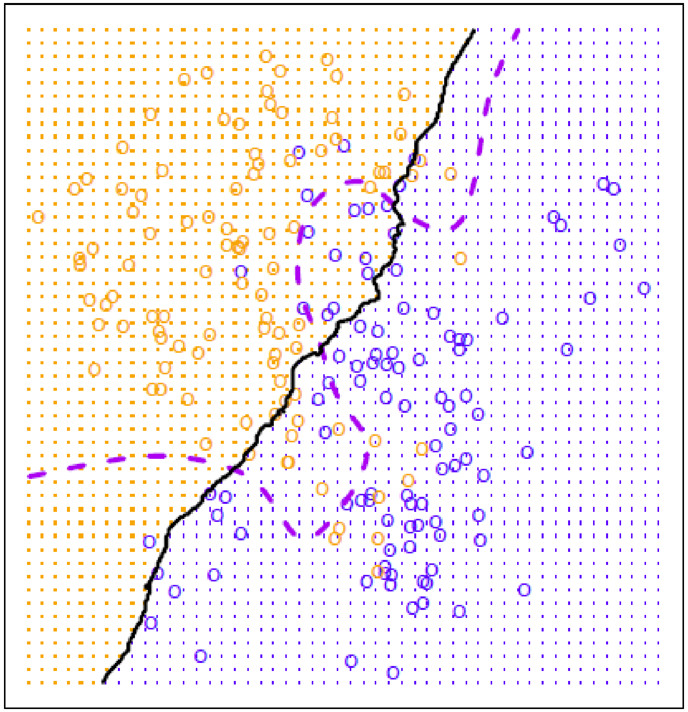
KNN: K=1 and K=100

KNN: K=1



Low Bias, High Variance
Overly Flexible

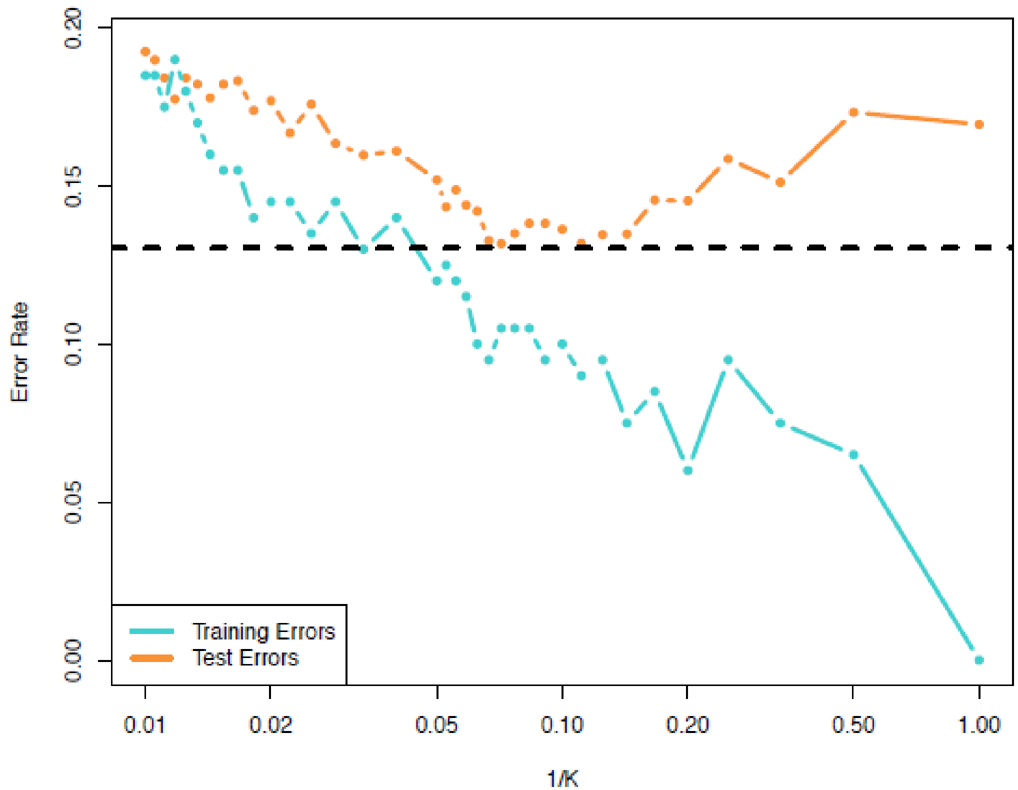
KNN: K=100



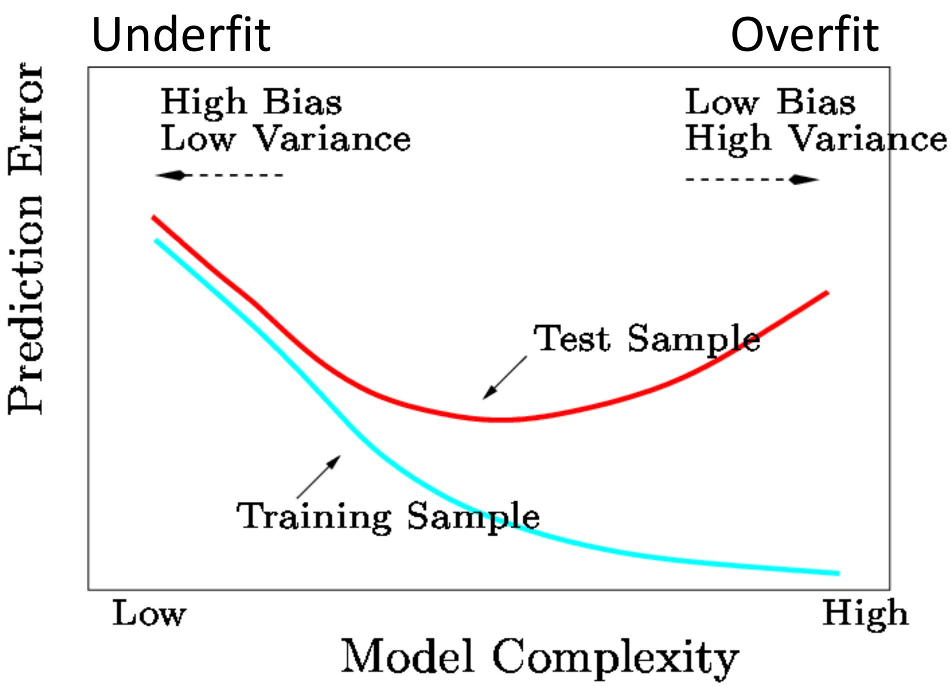
High Bias, Low Variance
Less Flexible

KNN Training vs. Test Error Rates

- Notice that the KNN training error rates (blue) keep going down as *k* decreases (i.e. as the flexibility increases).
- However, note that the KNN test error rate at first decreases but then starts to increase again.



Key Note: Bias-Variance Trade-Off



When selecting a machine learning method, remember that more flexible/complex is not necessarily better!!

- In general, training errors will always decline.
- However, test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).

What is R?

- Open-source, free software environment for statistical computing and graphics.
- Recommend using RStudio (GUI interface).
- 4,000+ packages available, with many used for machine/statistical learning and data mining.

R – Exploratory Data Analysis

- We use the “iris” data set to demonstrate exploratory data analysis in R.

- We inspect the dimensionality, structure and data of an R object.

- We view basic statistics and explore multiple variables.

R – Exploratory Data Analysis (cont.)

- We use the "iris" data set to demonstrate exploratory data analysis in R

```
> # We first check the size and structure of data
> dim(iris)
[1] 150   5
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> str(iris)
'data.frame':	150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> attributes(iris)
$names
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"

$row.names
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
 [32]  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62
 [63]  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93
 [94]  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
[125] 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150

$class
[1] "data.frame"
```

R – Exploratory Data Analysis (cont.)

- We next look at the first five rows of data.

```
> # We next look at the first five rows of data
> iris[1:5,]
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> tail(iris)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

R – Exploratory Data Analysis (cont.)

- We can also retrieve the values of a single column.

```
> # We can also retrieve the values of a single column
> iris[1:10, "Sepal.Length"]
 [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9
> iris$Sepal.Length[1:10]
 [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9
```

- We can also get the summary statistics.

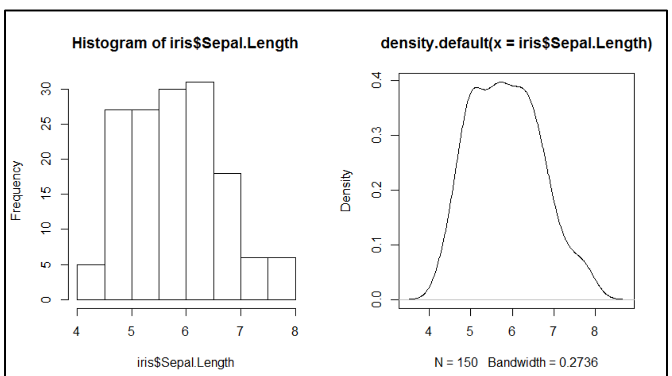
```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species  
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199                  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                  
>
```

R – Exploratory Data Analysis (cont.)

- We can also get quartiles and percentiles.

```
> # We can also get quartiles and percentiles
> quantile(iris$Sepal.Length)
  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9
> quantile(iris$Sepal.Length, c(.1, .3, .65))
 10%  30%  65%
4.80 5.27 6.20
```

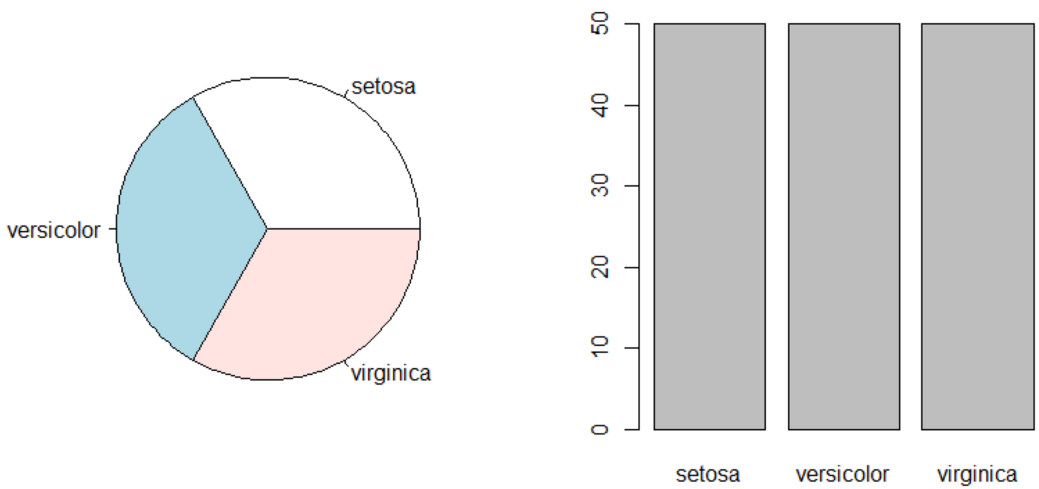
- We can check variance and get its distribution via histogram and density functions.




```
> # We can check variance and get its distribution via histogram and density functions
> var(iris$Sepal.Length)
[1] 0.6856935
> par(mfrow = c(1, 2))
> hist(iris$Sepal.Length)
> plot(density(iris$Sepal.Length))
```

R – Exploratory Data Analysis (cont.)

- We can get a frequency of the factors and plot a pie chart or bar chart.




```
> # We can get the frequency of fact
> table(iris$Species)

    setosa versicolor  virginica 
        50         50         50 
> par(mfrow = c(1, 2))
> pie(table(iris$Species))
> barplot(table(iris$Species))
```

R – Exploratory Data Analysis (cont.)

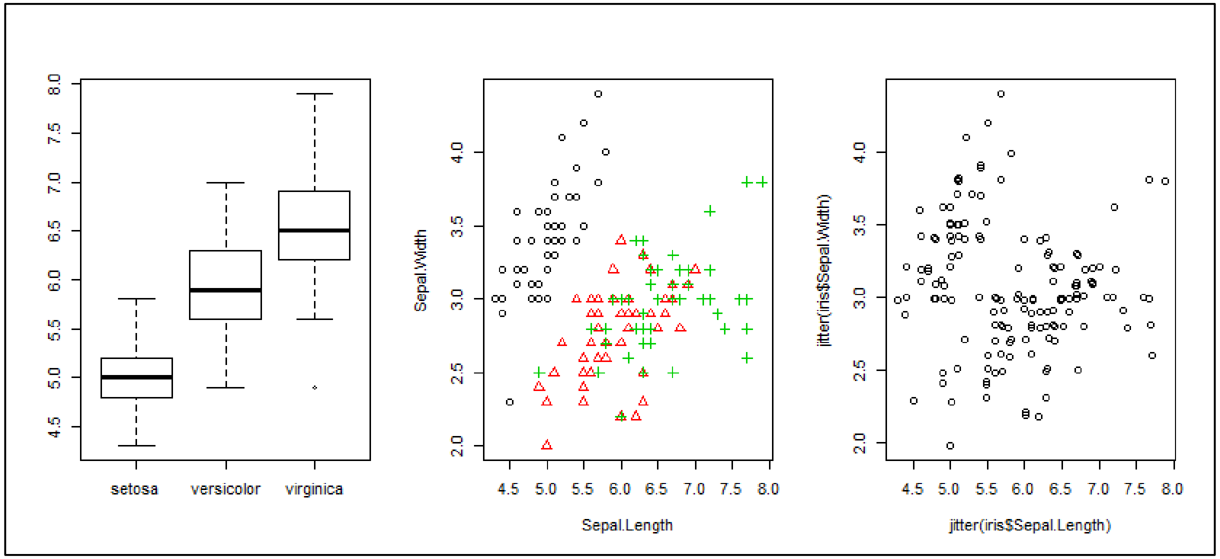
- We can explore multiple variables.

```
> # We can explore multiple variables
> cov(iris$Sepal.Length, iris$Petal.Length)
[1] 1.274315
> cov(iris[,1:4])
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
> cor(iris$Sepal.Length, iris$Petal.Length)
[1] 0.8717538
> cor(iris[,1:4])
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
> aggregate(Sepal.Length ~ Species, summary, data=iris)
     Species Sepal.Length.Min. Sepal.Length.1st Qu. Sepal.Length.Median Sepal.Length.Mean Sepal.Length.3rd Qu.
1     setosa             4.300                4.800               5.000             5.006                5.200
2 versicolor             4.900                5.600               5.900             5.936                6.300
3  virginica             4.900                6.225               6.500             6.588                6.900
  Sepal.Length.Max.
1             5.800
2             7.000
3             7.900
```

R – Exploratory Data Analysis (cont.)

- Boxplots, scatterplots and scatterplots with jitter (small amount of noise).

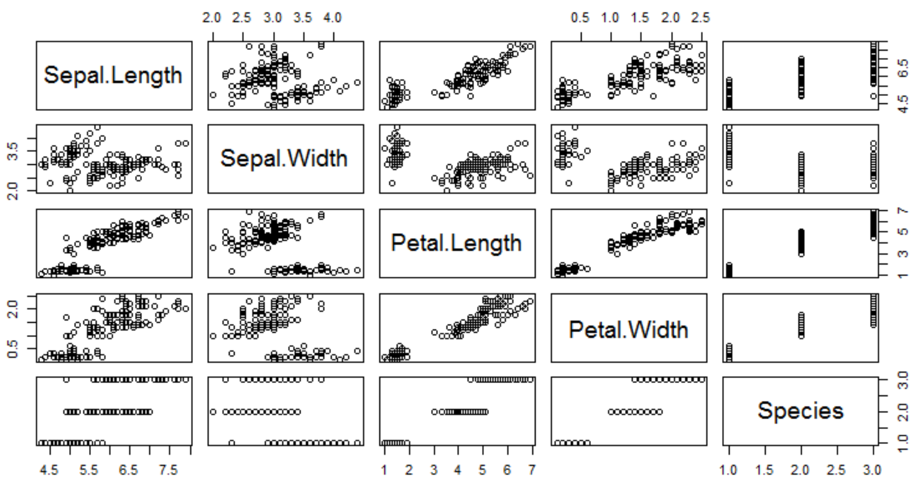
```
> # Boxplots, scatterplots and scatterplots with jitter (small amount of noise)
> par(mfrow = c(1, 3))
> boxplot(Sepal.Length~Species, data=iris)
> with(iris, plot(Sepal.Length, Sepal.Width, col=Species, pch=as.numeric(Species)))
> plot(jitter(iris$Sepal.Length), jitter(iris$Sepal.Width))
```

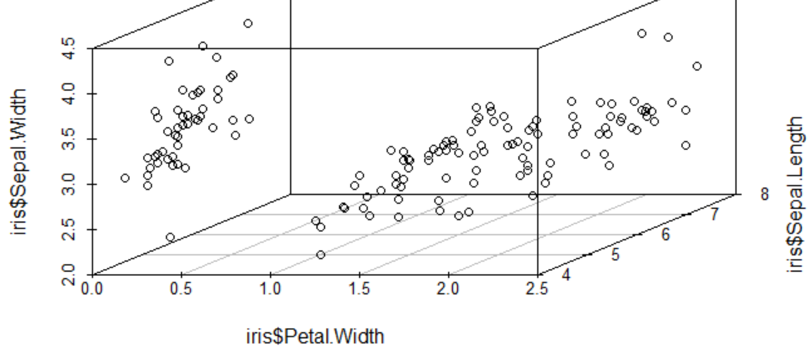


R – Exploratory Data Analysis (cont.)

- Produce a matrix of scatterplots or a 3D scatterplot.

```
> # Produce a matrix of scatterplots or a 3D scatterplot
> par(mfrow = c(1, 1))
> pairs(iris)
> library(scatterplot3d)
> scatterplot3d(iris$Petal.Width, iris$Sepal.Length, iris$Sepal.Width)
```

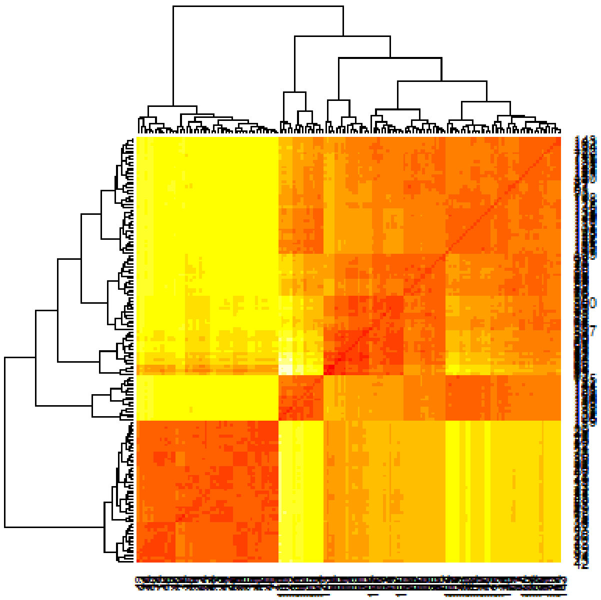


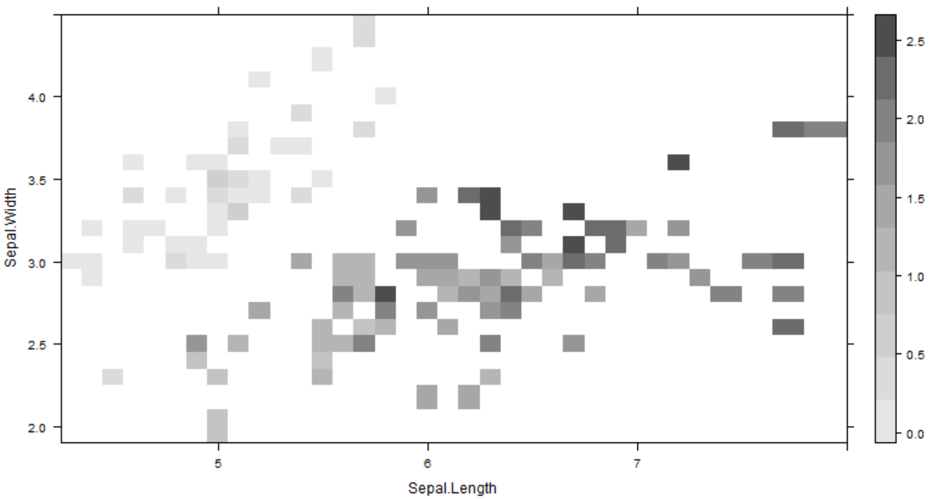


R – Exploratory Data Analysis (cont.)

- Produce a heat map or a level plot.

```
> # Produce a heat map or a level plot
> distMatrix <- as.matrix(dist(iris[,1:4]))
> heatmap(distMatrix)
> library(lattice)
> levelplot(Petal.Width~Sepal.Length*Sepal.Width, iris, cuts=9,
+           col.regions=grey.colors(10)[10:1])
```

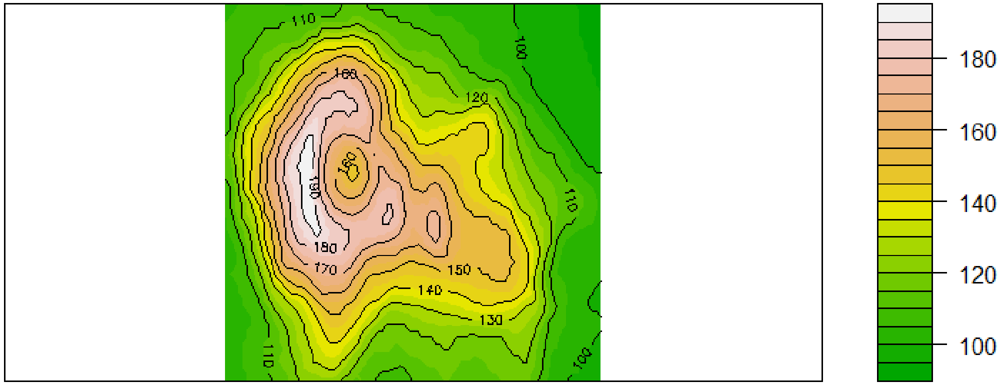


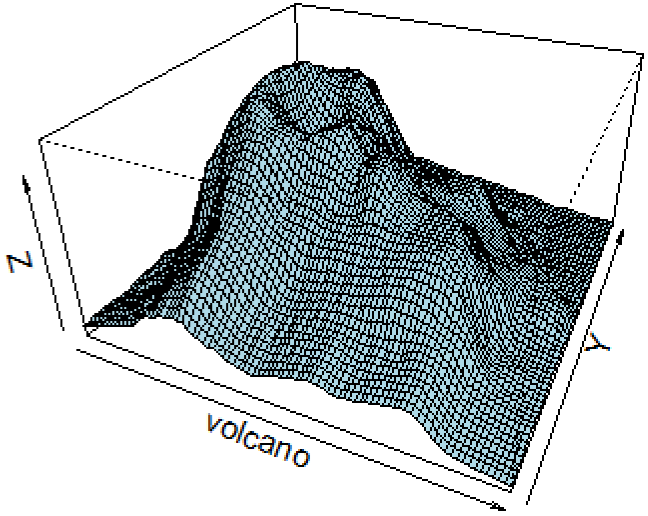


R – Exploratory Data Analysis (cont.)

- Produce a contour plot or 3D surface plot.

```
> # Produce a contour plot or 3D surface plot
> par(mfrow = c(1, 1))
> filled.contour(volcano, color=terrain.colors, asp=1,
+                plot.axes=contour(volcano, add=T))
> persp(volcano, theta=25, phi=30, expand=0.5, col="lightblue")
```

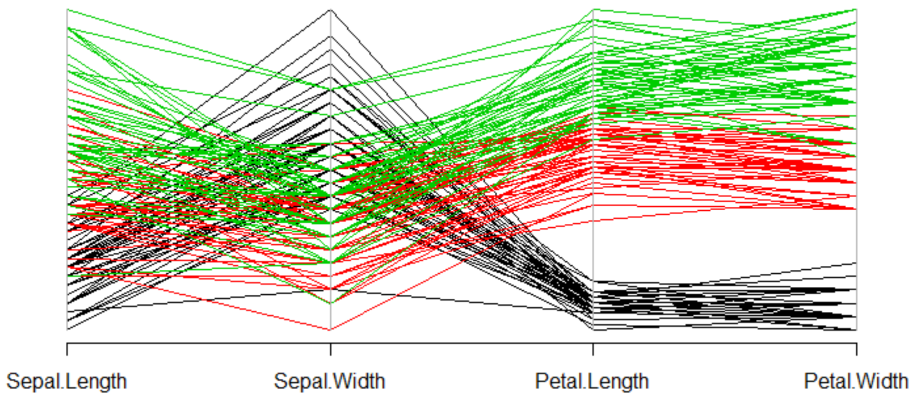


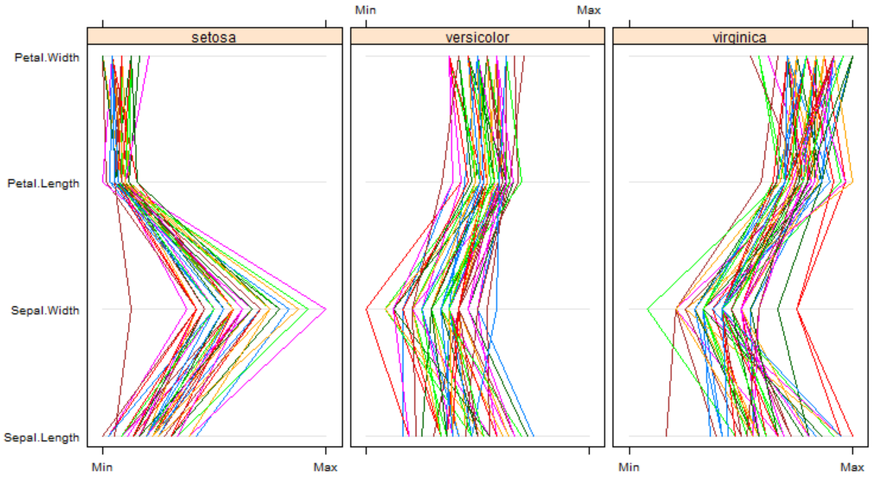


R – Exploratory Data Analysis (cont.)

- Plot parallel coordinates.

```
> # Plot parallel coordiantes
> library(MASS)
> parcoord(iris[1:4], col=iris$Species)
> library(lattice)
> parallelplot(~iris[1:4] | Species, data=iris)
```





More about R….

- Review the R tutorials posted on Canvas.
- Check out the CRAN website.
- Use help command = ?
- Practice, practice, practice…

Summary

- Overview of machine learning
- Key concepts of the learning problem
- Learning algorithm trade-offs
- Supervised versus unsupervised learning
- Regression versus classification methods
- Assessing model accuracy
- Bias-Variance trade-offs
- Introduction to R statistical programming