MY474: Applied Machine Learning for Social Science

Lecture 1: Getting Started, Brief Introduction to Machine Learning

Blake Miller

09 October 2019

Agenda

1. Introductions
2. Syllabus
3. What is machine learning?
4. R Setup instructions
5. R, Rmarkdown Review
6. Overview of Statistical Learning

My name is Blake Miller. I'm originally from San Diego, California. You can call me Dr. Miller or Blake. I use he/his pronouns. Also I am a new uncle!

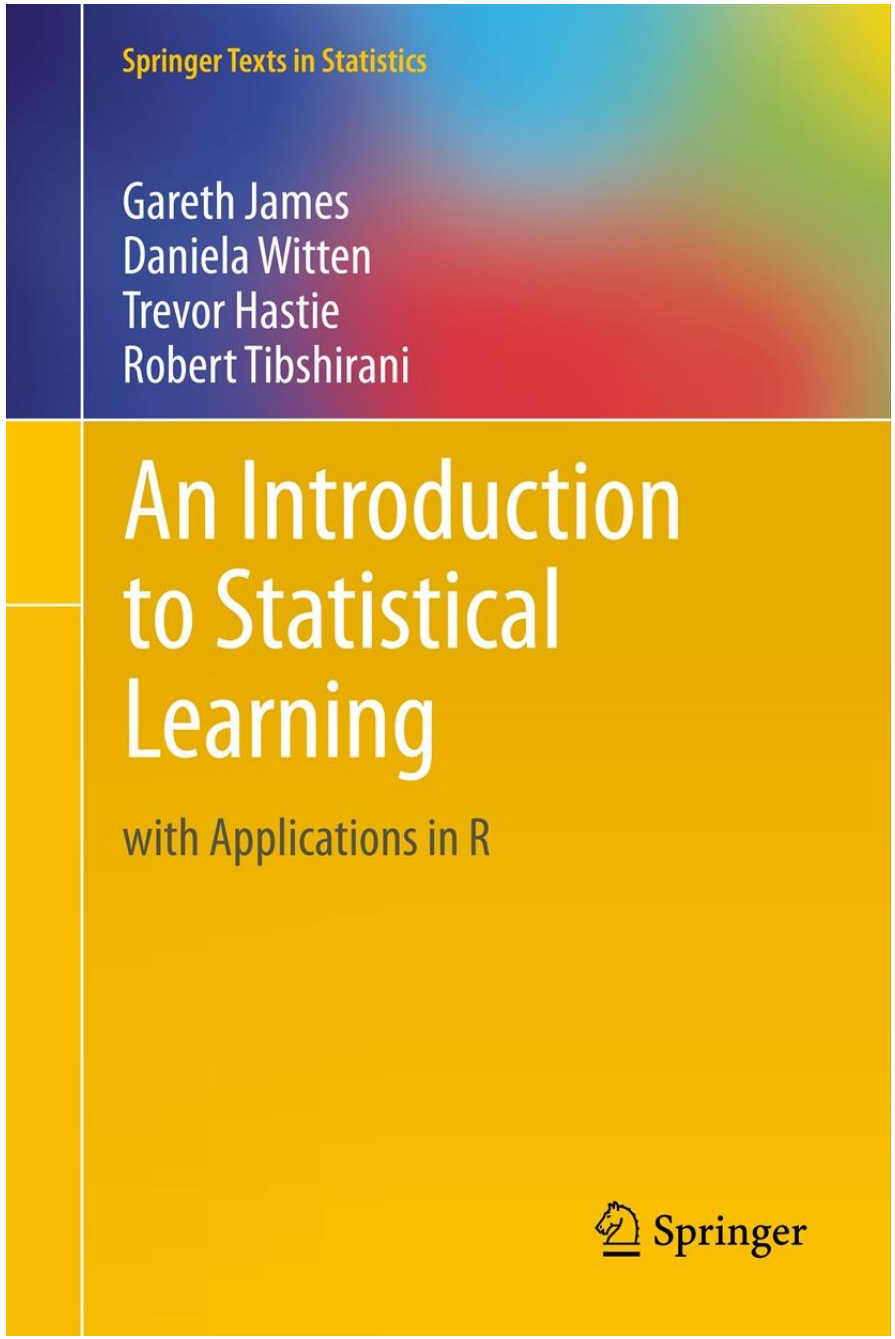
I'm a political scientist and a computer scientist. I study how (authoritarian) governments use data to manipulate information and repress citizens. (Source: China Foto Press)

Your turn

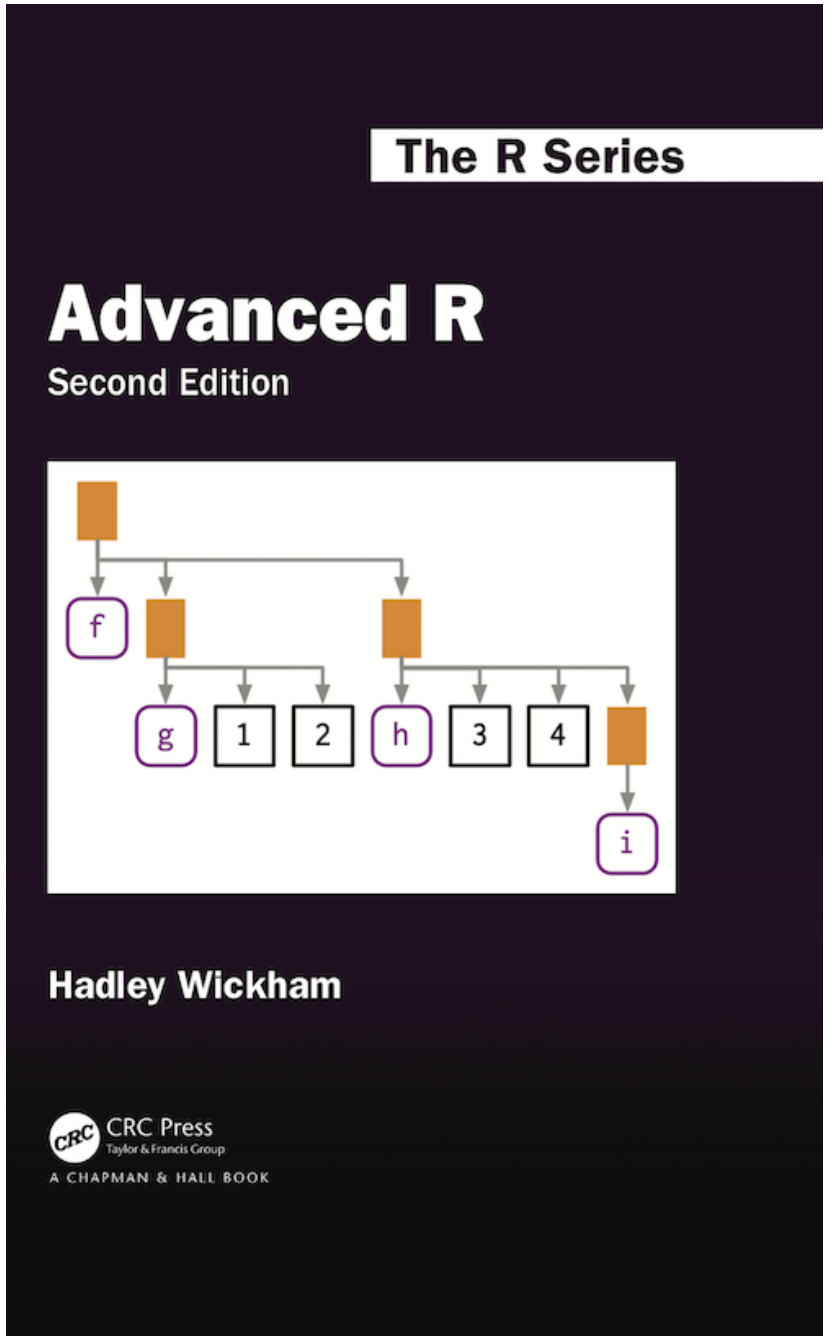
To your table:

1. Introduce yourself (name, pronouns, where you're from)
2. Tell the table an interesting fact about you
3. Tell the table something you are intellectually curious about

Books


Springer Texts in Statistics
Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani
An Introduction to Statistical Learning
with Applications in R
Springer

Books


The R Series
Advanced R
Second Edition
f
g
1
2
h
3
4
i
Hadley Wickham
CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Syllabus

MY574

If you are a PhD student, in lieu of the final exam, you will write an original research report using data relevant to a research question you have. If you are enrolled in MY574, please schedule an office hour appointment with me so we can discuss this in detail.

Course Access

Access will be determined by:

1. ASDS, SRM priority
2. Performance on OLS quiz

Think-pair-share

1. Think (3 min)
   - Silently write some thoughts about what machine learning is.
   - Silently write what you'd like to learn from this class.
2. Pair (3 min)
   - Pair up and compare/discuss your goals for the class and what you think machine learning is.
3. Share (1 min per group)

What is machine learning?

Which of these Wikipedia talk page comments are insults?

1. "I hate your freakin guts"
2. "HENCE, ROZ LIPSHITS IS AN IDIOT!!!!"
3. "Thanks for experimenting with the page Ladue, Missouri on Wikipedia. Your test worked, and has been reverted or removed. Please use the sandbox for any other tests you want to do. Take a look at the welcome page if you would like to learn more about contributing to our encyclopedia. Thanks."
4. "I was quoting you, you dumb bastard. You really don't understand ANYTHING"
5. "I'm afraid I reverted your change to Talk:Nairobi. You erased all of the existing discussion there; we want to keep that. The other thing is that your description of Nairobi isn't really suitable as an encyclopedia article it was more of a personal account. That's OK, of course, but Wikipedia isn't the place to put it."
6. "STOP YOUR DAMN NONSENSE LUMINIFER!!!!! YOU HAVE DONE NOTHING TO IMPROVE ANYTHING RELATED TO THESE ARTICLES!!!"

Which of these Wikipedia talk page comments are insults?

1. **“I hate your freakin guts”**
2. **“HENCE, ROZ LIPSHITS IS AN IDIOT!!!!”**
3. “Thanks for experimenting with the page Ladue, Missouri on Wikipedia. Your test worked, and has been reverted or removed. Please use the sandbox for any other tests you want to do. Take a look at the welcome page if you would like to learn more about contributing to our encyclopedia. Thanks.”
4. **“I was quoting you, you dumb bastard. You really don’t understand ANYTHING”**
5. “I’m afraid I reverted your change to Talk:Nairobi. You erased all of the existing discussion there; we want to keep that. The other thing is that your description of Nairobi isn’t really suitable as an encyclopedia article it was more of a personal account. That’s OK, of course, but Wikipedia isn’t the place to put it.”
6. **“STOP YOUR DAMN NONSENSE LUMINIFER!!!!! YOU HAVE DONE NOTHING TO IMPROVE ANYTHING RELATED TO THESE ARTICLES!!!”**

How did you solve this problem?

- What **features** of each comment were informative?
- Could you make this process more explicit?
- Could you write code to solve this problem?

Activity

Think of as many algorithms (sets of instructions) as you can to perfectly classify these comments into “insults” and “not insults.” Write each on the board. You have 5 minutes.

The Process of Machine Learning

1. Gather data
2. Clean data
3. Describe data
4. Transform data
5. Specify a model
6. Train the model
7. Evaluate model performance

Gather data

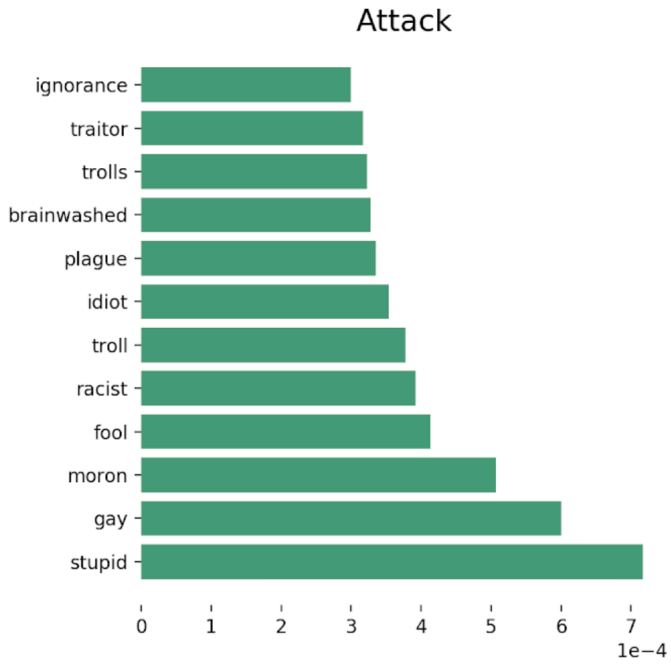
- Scrape data from the internet, collect data in a survey, etc.
- Build automated scrapers

Clean data

```
have had enough of your whining from the White House. We
need an actual leader—our Nation's future is at stake. <a
href="https://t.co/dNzRGuOFdP">https://t.co/dNzRGuOFdP</a>
</p>
<p>&mdash; John O. Brennan (@JohnBrennan) <a
href="https://twitter.com/JohnBrennan/status/107977083803820
4416?ref_src=twsrc%5Etfw">December 31, 2018</a></p>
</blockquote></p>
<figure id="D-ROS-B1" class="a8d"></figure>
<figure id="M-ROS-B1" class="a8d"></figure><figure
id="gmxrevmore" class="H"></figure>
<p>Brennan&#8217;s remark came in response to President
Trump&#8217;s tweet pushing back on negative press regarding
```

- Identify outliers, bots, "straightliners"
- Remove unnecessary artifacts (HTML code, whitespace)
- Correct misspellings, entry errors (should "CA," "ca," and "California" be counted as the same entry under "state")

Describe data



Most informative words in “attacks” in online comments.

- Look at descriptive statistics
- Identify patterns, correlations
- Visualize the data
- Perform sanity checks (are some people in your data 1000 years old?)

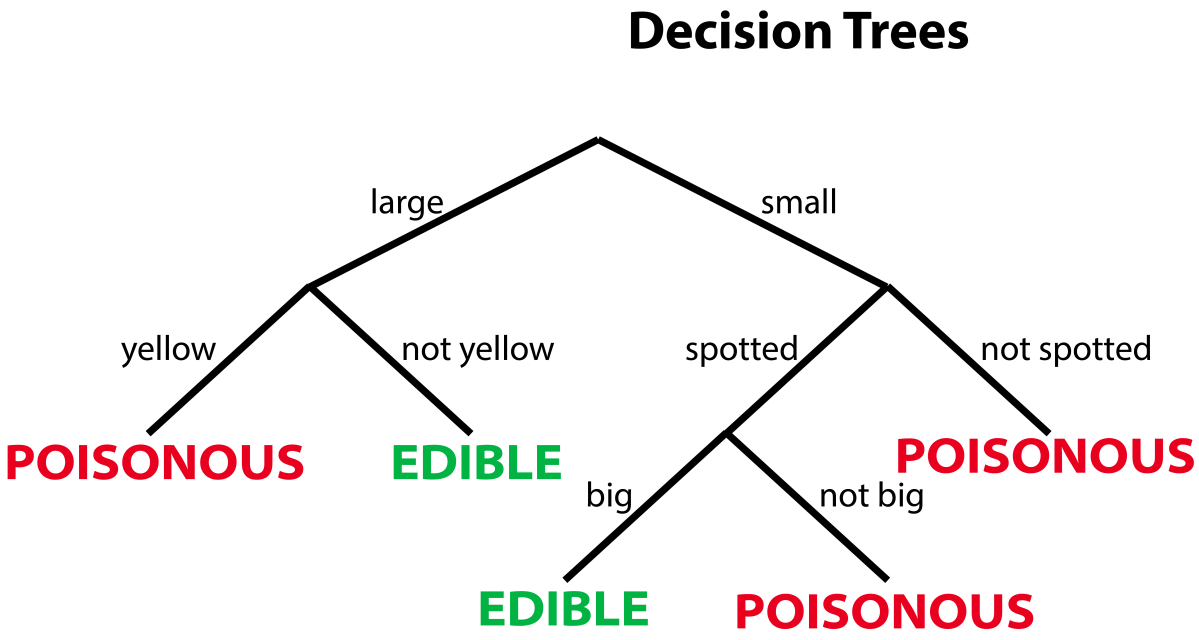
Transform data

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

The term-document matrix for four words in four Shakespeare plays. The red boxes show that each document is represented as a column vector of length four (SLP, Figure 6.2).

- Transform data to make it more informative, to accommodate computational constraints, to facilitate **explanation/description**
- This is called **feature engineering** and it is usually more important than the model you choose
- What **features** of the data did we discuss were informative to the task of identifying insults?

Specify a model

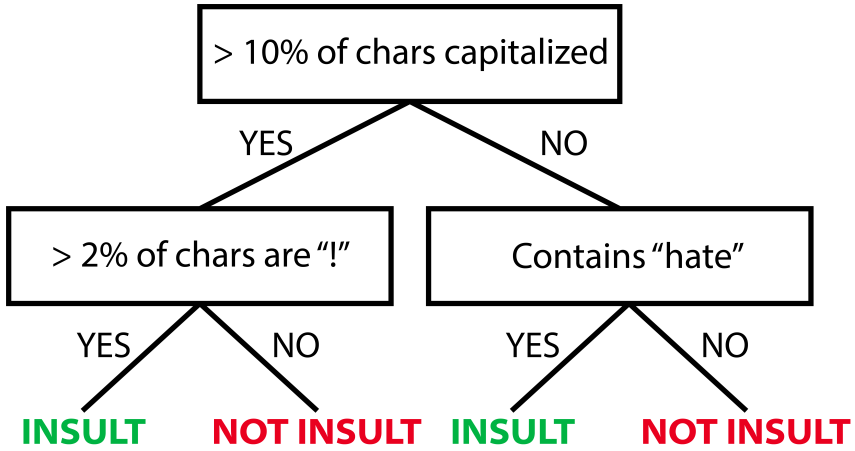


Is this mushroom poisonous?

We can use s really simple model like a decision tree. Source: http://users.cs.cf.ac.uk/Dave.Marshall/AI2/node147.html

- Specify model(s)
- Matching model(s) to problems is an important skill
- Sometimes simple models outperform the fancy/trendy ones

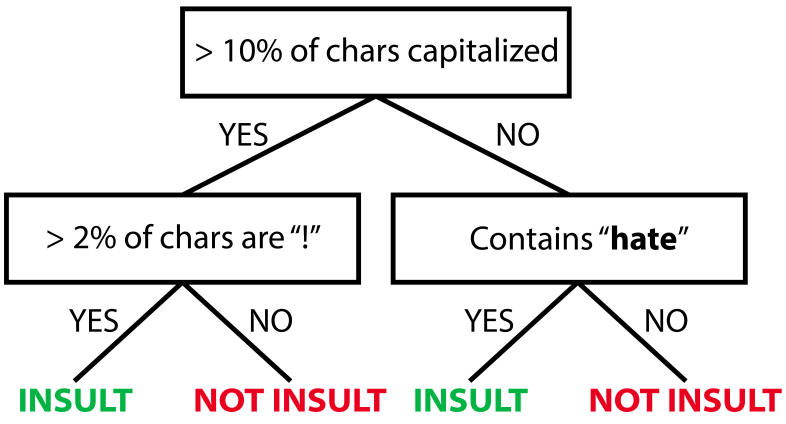
Train the model

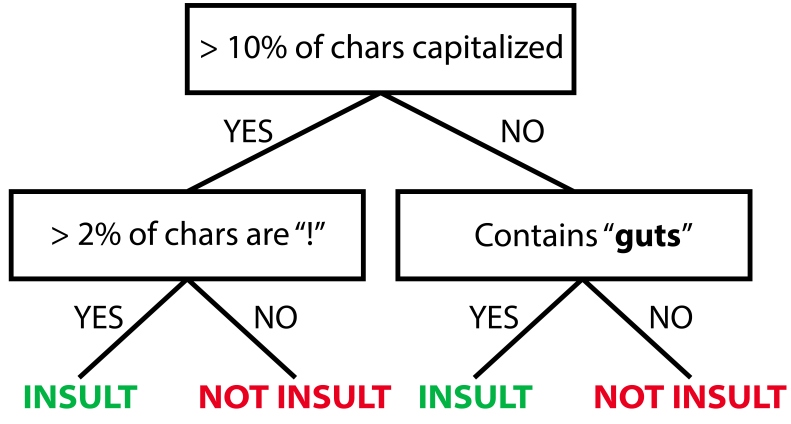


This decision tree perfectly classifies our data.

- Specify a loss function
- Develop an algorithm to minimize the loss function
- **Train** the model (minimize the loss) with a subset of data

Evaluate model performance





However, many models perfectly classify our data. Which one do we choose?

- Evaluate the generalizability of each model's predictions on unseen **test** data using some quantitative **performance measure** (like accuracy, RMSE) to see how well the model **generalizes**
- Choose **feature sets** and **model specifications** based on these performance measures, using **cross-validation**

Evaluate model performance

What if we gathered some more Wikipedia Talk comments? How well would the models in the last slide **generalize**?

1. "You are an old cougar! You are an old cougar!"
2. "There are multiple sources to go to pal. . . . . You go to one music source on the internet and that's the be all end all. There are many sources to go to in order to find info on Allin and T.T. Quick. . . . . .You could just buy the albums and look at them. I guess you're one of those people who are just experts on every subject, right! What else do you do on your computer all day? I wonder! You're full of shit. . . . . and you know it!"
3. "Well that's the other way of dealing with it of course) I guess I didn't think PROD would be worthwhile as they usually get removed, but in this case there's been little editing activity so it will probably work."

How is this course different from an ML course in computer science or stats?

1) More code, less Greek
2) More focus on applications to (social) science
3) This is both a methods class AND a substantive class

What we'll cover

1. Major concepts
2. Methods and tools
3. Applications to social science
4. Social science implications of ML

Major concepts

- Overfitting/underfitting: How complex of a model do we want to use? How complex are our data?
- Generalization error: Measures for how well a model generalizes to new data.
- Cross-validation: How do we estimate generalization error without throwing out data?
- Regularization: How can we train our models to focus on important features in our data?
- Feature engineering: How can we best represent the data for our specific task?
- Model selection: What is the best performing model?

Methods and tools

- Ordinary least squares (OLS)
- Logistic Regression
- Regularization
- Gradient descent
- Tree-based methods
- Support vector machines
- Active learning
- Dimension reduction
- Clustering

Applications to social science

- Automated text analysis
- Exploratory data analysis
- Hate speech, troll, bot detection
- Using texts for experiments

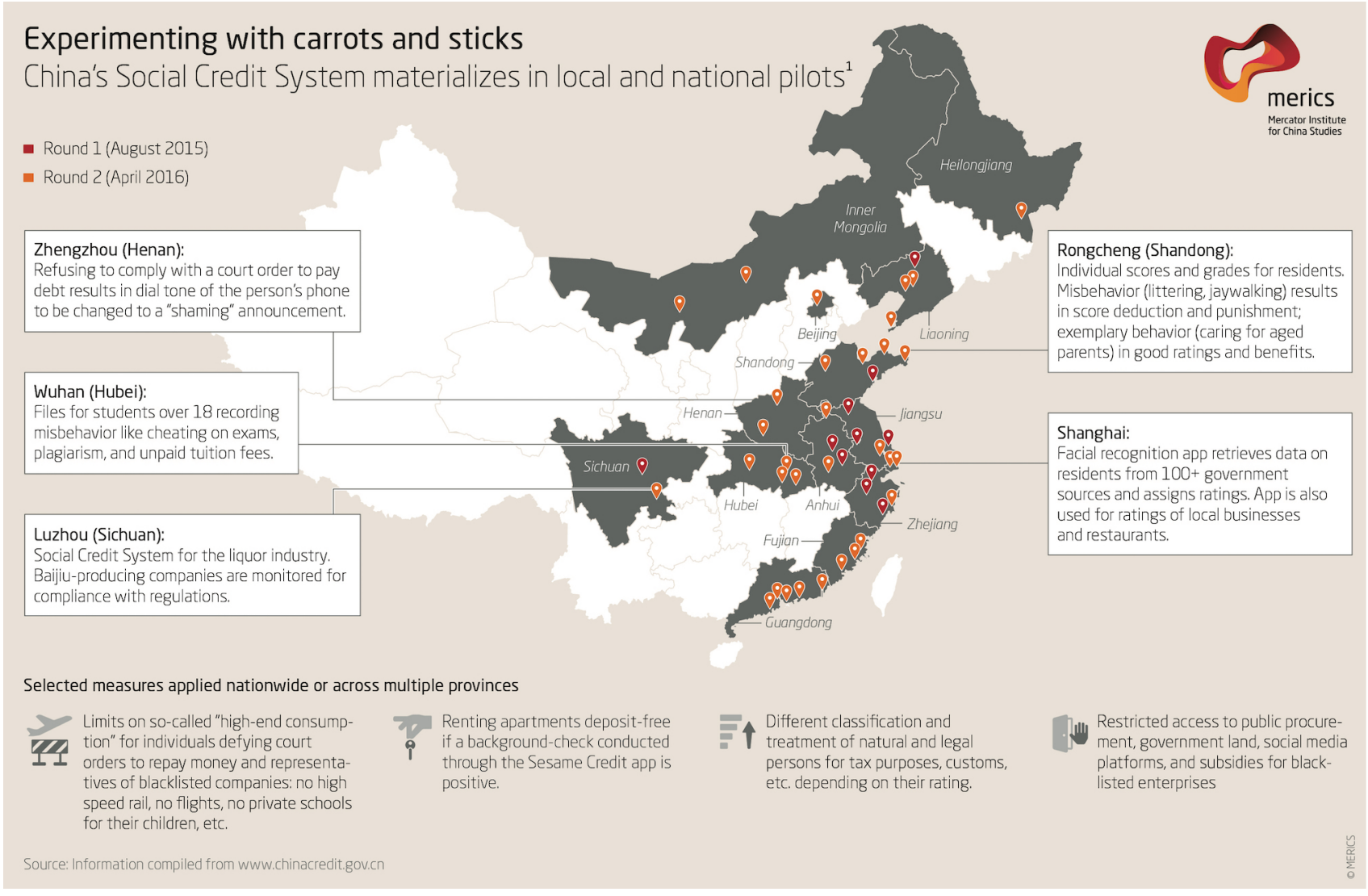
Social science implications of the rise of ML

- Rise of surveillance capitalism, mass surveillance
- Interpretability of ML decisions
- Algorithmic bias, fairness, accountability, and transparency
- How ML is used by (authoritarian) governments

Getting Meta: Social Science About ML

Black Mirror portrayed a society where your social standing and access to public goods was governed by a social-media-based rating system. (Source: BBC.co.uk)

Getting Meta: Social Science About ML



Well, it turns out the Chinese state is trying to develop something similar. (Source: MERICS.org)

How is your R?

After running this code, what is the value of *a*?

```
a <- 3
for (i in 1:5) {
    a <- a + 1
    if (i == 4) {
        a <- a/2
    }
}
a
```

How is your R?

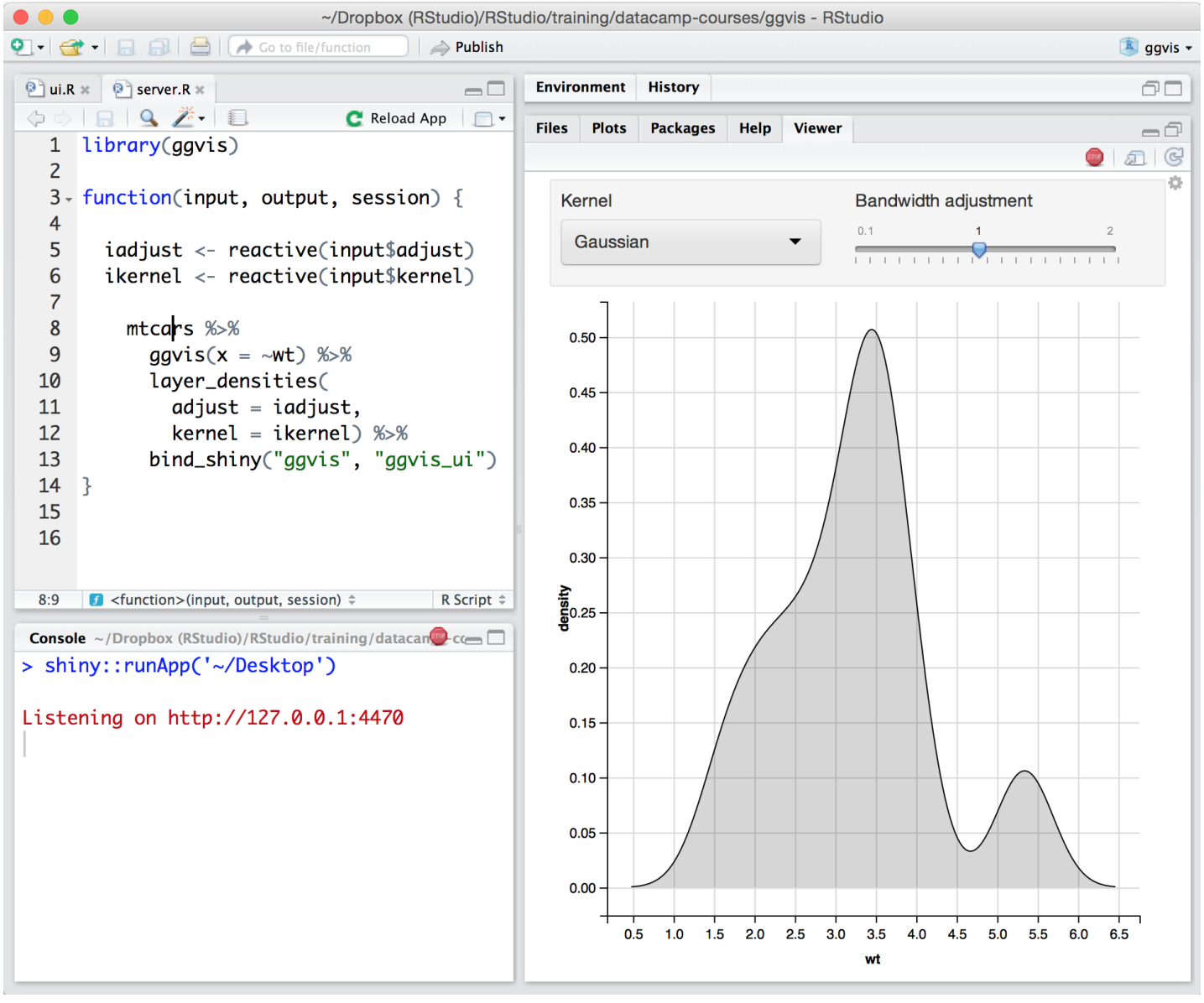
```
a <- 3
for (i in 1:5) {
    a <- a + 1
    if (i == 4) {
        a <- a/2
    }
}
a
```

```
## [1] 4.5
```

Why we're using R

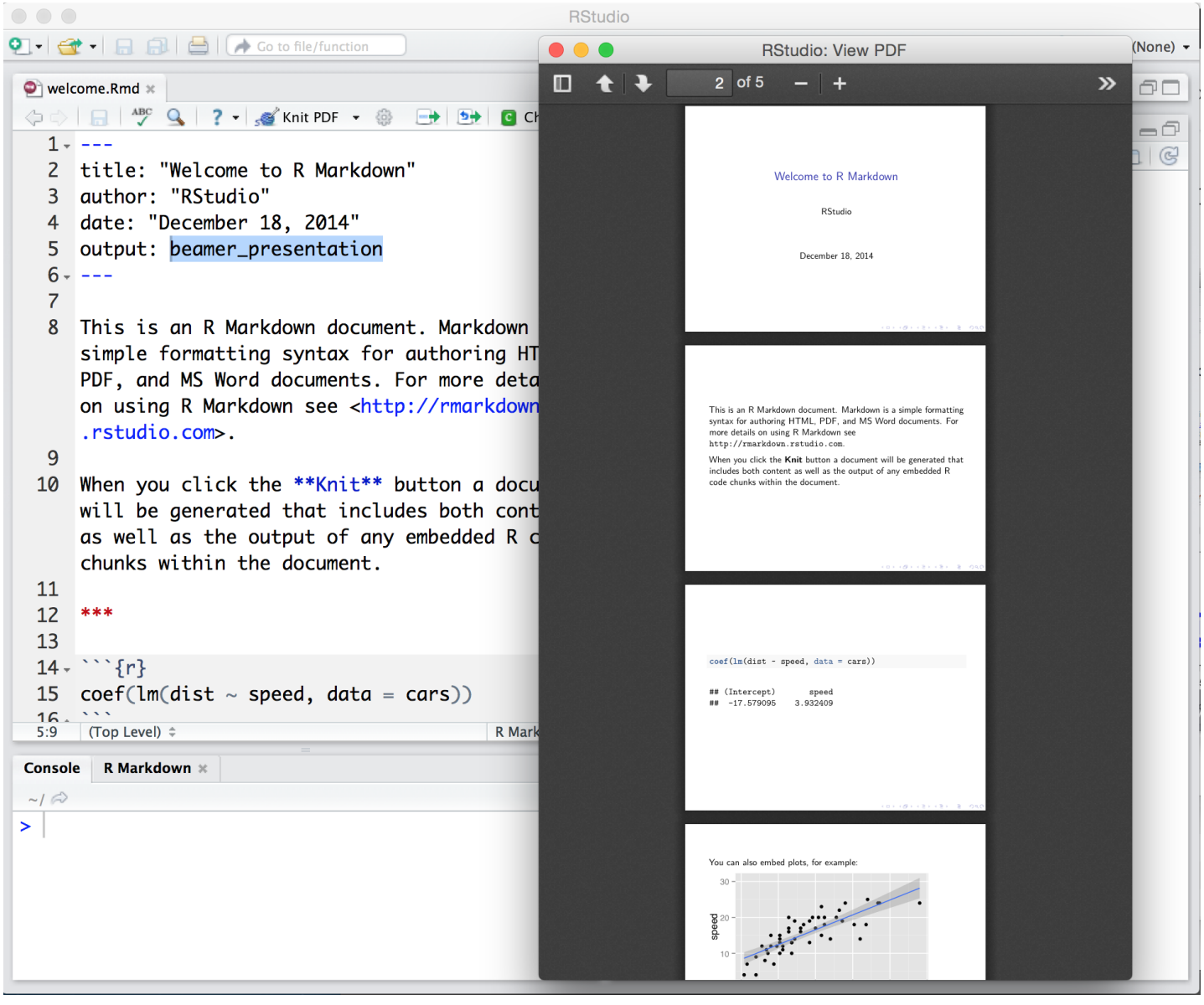
- Becoming *lingua franca* of statistical analysis in academia
- What employers in private sector demand
- It's free and open-source
- Flexible and extensible through *packages* (over 10,000 and counting!)
- Powerful tool to conduct automated text analysis, social network analysis, and data visualization, with packages such as quanteda, igraph or ggplot2.
- Command-line interface and scripts favors reproducibility.
- Excellent documentation and online help resources.

RStudio



Source: RStudio.com

RMarkdown

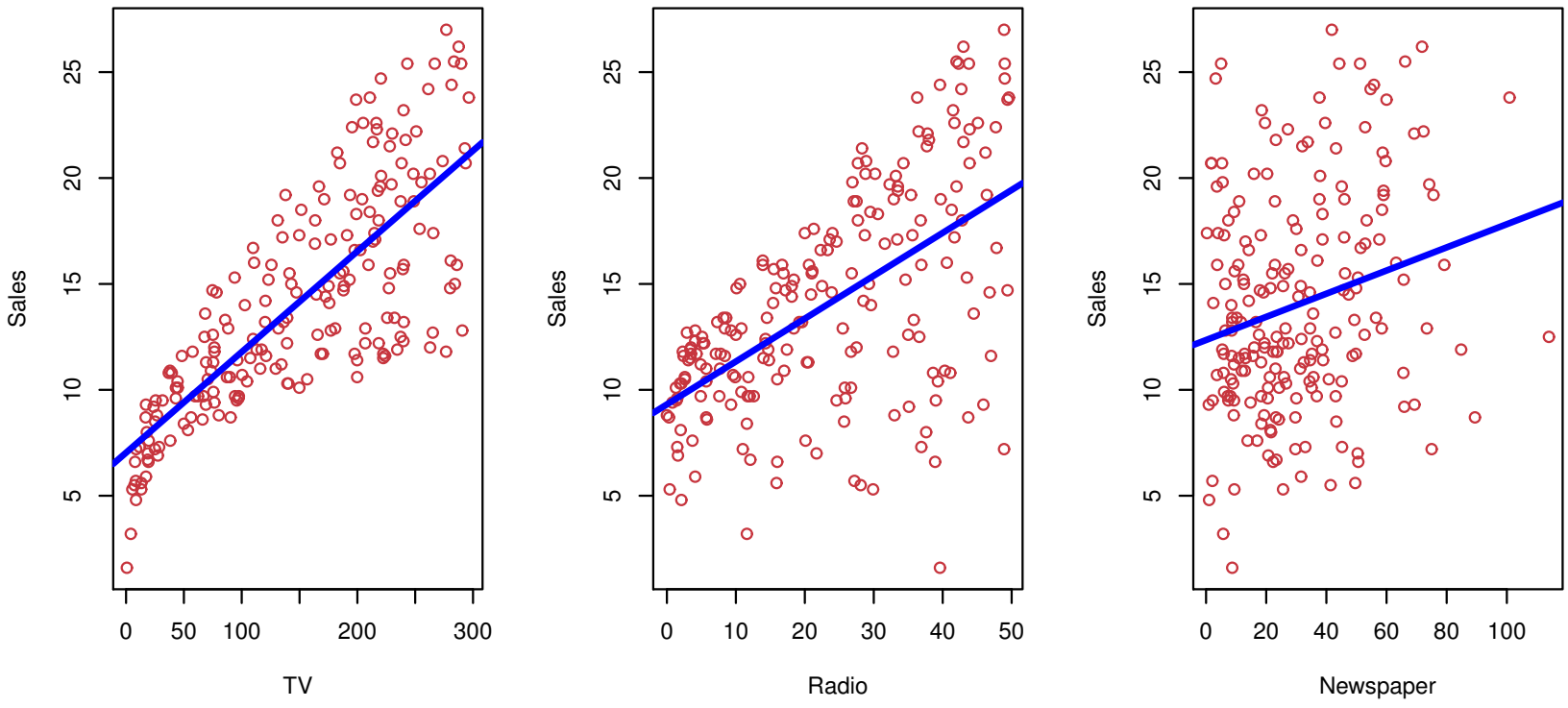


Source: RStudio.com

Let's install R

- Install R and R Studio: https://www.datacamp.com/community/tutorials/installing-R-windows-mac-ubuntu
- Install `tinytex` for RMarkdown pdfs: https://bookdown.org/yihui/rmarkdown/installation.html#installation

What is Statistical Learning?



We can model sales with:

$$Sales \approx f(TV, Radio, Newspaper)$$

Which are the **features** and which are the **outcome variables**?

Notation

- `Sales` (Y) is an **outcome** variable
- `TV` (X_1), `Radio` (X_2), and `Newspaper` (X_3) are **features**
- We wish to predict the outcome using some function of the features f(X)
- Our model is then $Y = f(X) + \epsilon$
- ϵ captures measurement errors and other discrepancies

Terminology

Features are the same thing as:

- Independent variables (I.V.)
- Input variables
- Predictor Variables

Outcome Variables are the same thing as:

- Dependent variables (D.V.)
- Response variables
- Target variables

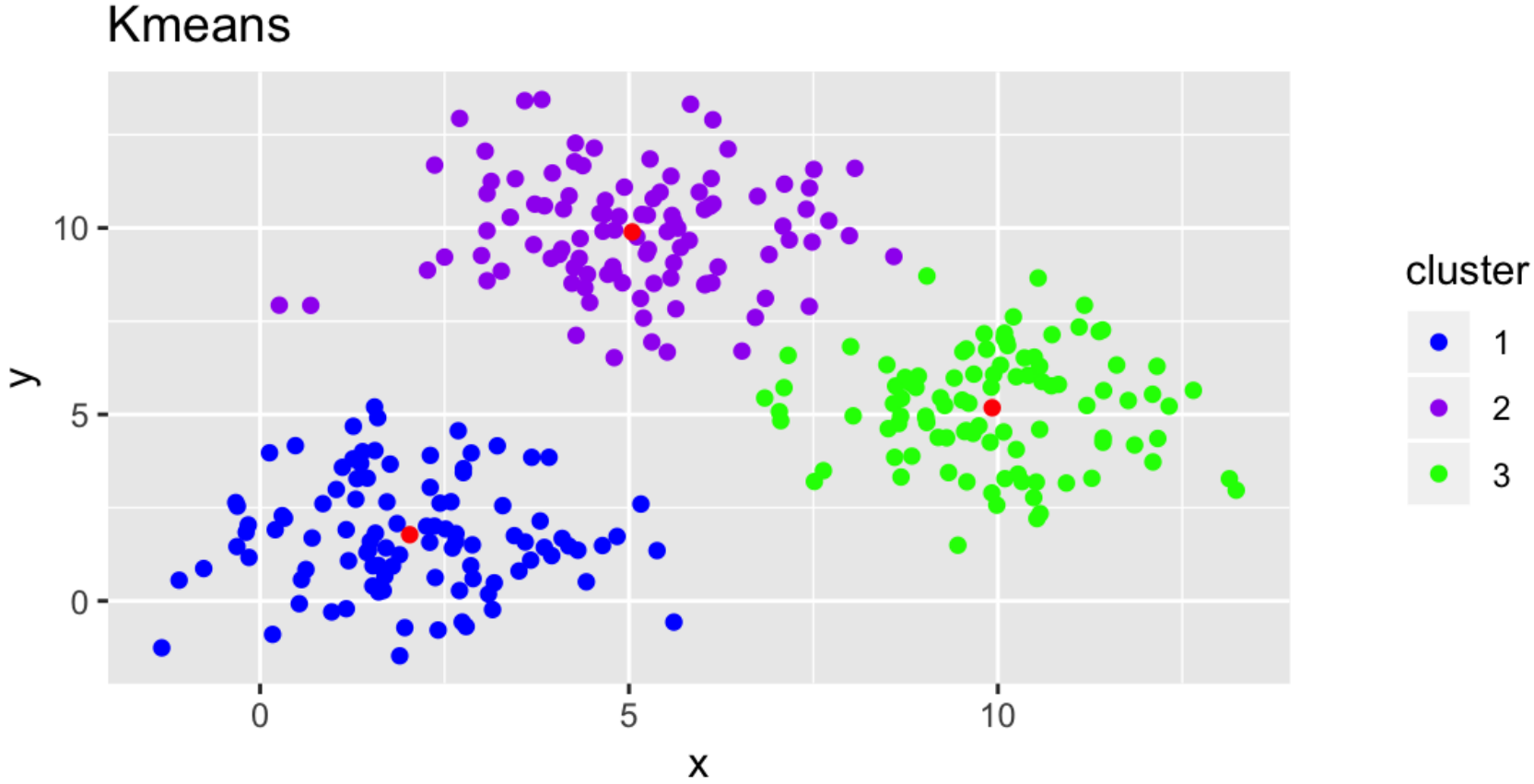
Supervised Learning

- X: data matrix, each column is called a feature, idependent variable, input variable
- Y: dependent variable/target
- **Objective:** Analyze/visualize relationships between X and Y and make predictions for Y given X
- **Regression:** Y is a number
  - Linear regression is a very specialized case of regression
  - There are non-linear regression algorithms such as tree regression
- **Classification:** Y is a value in an unordered set (i.e. $y_i \in \{\text{ordinary}, \text{bot}, \text{troll}, \text{cyborg}\}$)
- **Training data:** Pairs $(x_1, y_1), \ldots, (x_N, y_N)$ used to "train" a model that will later predict y when given unseen "test" cases.
  - What are important/informative features?
  - How good are the model's predictions?/How useful is the model?

Unsupervised Learning

- X: data matrix, each column is called a feature or covariate
- Y: unknown
- **Objective:** Find structure and patterns within a dataset
- Types of unsupervised learning
  - **Clustering:** Given data **X**, identify groups of objects that are similar to other members of the group and distinct from objects in other groups.
  - **Dimension Reduction:** Given data **X**, identify manifolds or underlying factors that explain the data in fewer dimensions
- **No training because Y is not given**

Unsupervised Learning (continued)



Examples

- Dimension Reduction (PCA, factor analysis, MDS)
- Parametric clustering (gaussian mixture models)
- Non-parametric clustering (k-means, hierarchical clustering, graph partitioning)

Dimension reduction

- **General Problem:** We collect a Q-dimensional data, but we believe that it is measuring and underlying feature that "lives" in a $q << Q$ dimension.
- **Goal:** learn with is that low dimensional space and use it to represent the data.

Formally,

- **We observe:** $\mathbf{X} = (\mathbf{X_1}, ..., \mathbf{X_n})^T, \mathbf{X_i} \in \mathbb{R}^Q$
- **We want to find:** q-dimensional submanifold M of $\mathbb{R}^q, q << Q$

$$\text{Model-based approach} \implies \begin{cases} \text{Factor Analysis} \implies X \sim N(\mu, \Sigma) \end{cases}$$

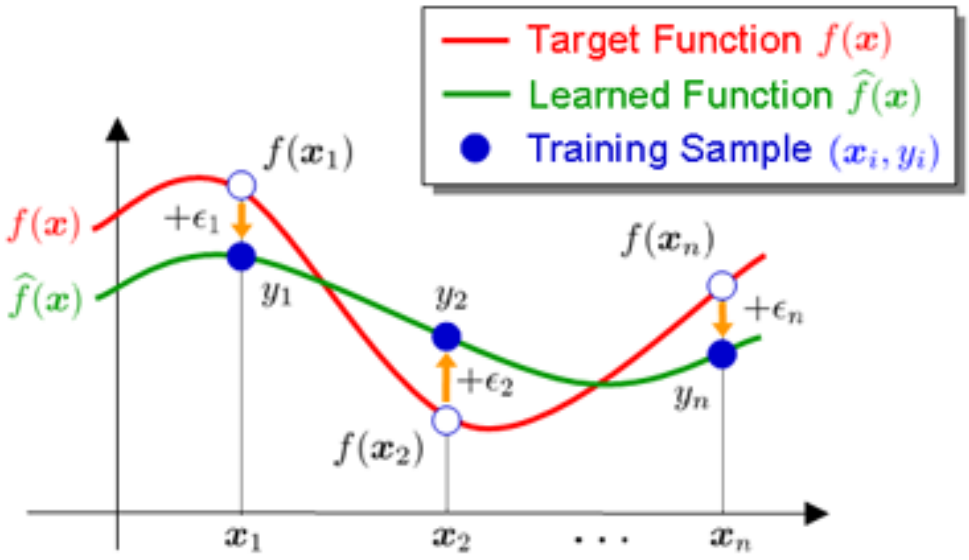
$$\text{Data driven approach} \implies \begin{cases} \text{Principal Component Analysis} \\ \text{Multidimensional-scaling} \end{cases}$$

Regression Versus Classification Problems

Outcome variables, some terms:

- Quantitative (regression)
- Qualitative/categorical (classification)
- Binomial, multinomial (classification)
- Multiclass (classification)
- Multilabel (classification)

Regression as a special case of function approximation



General Problem: we assume $\mathbf{y} = f(\mathbf{x})$

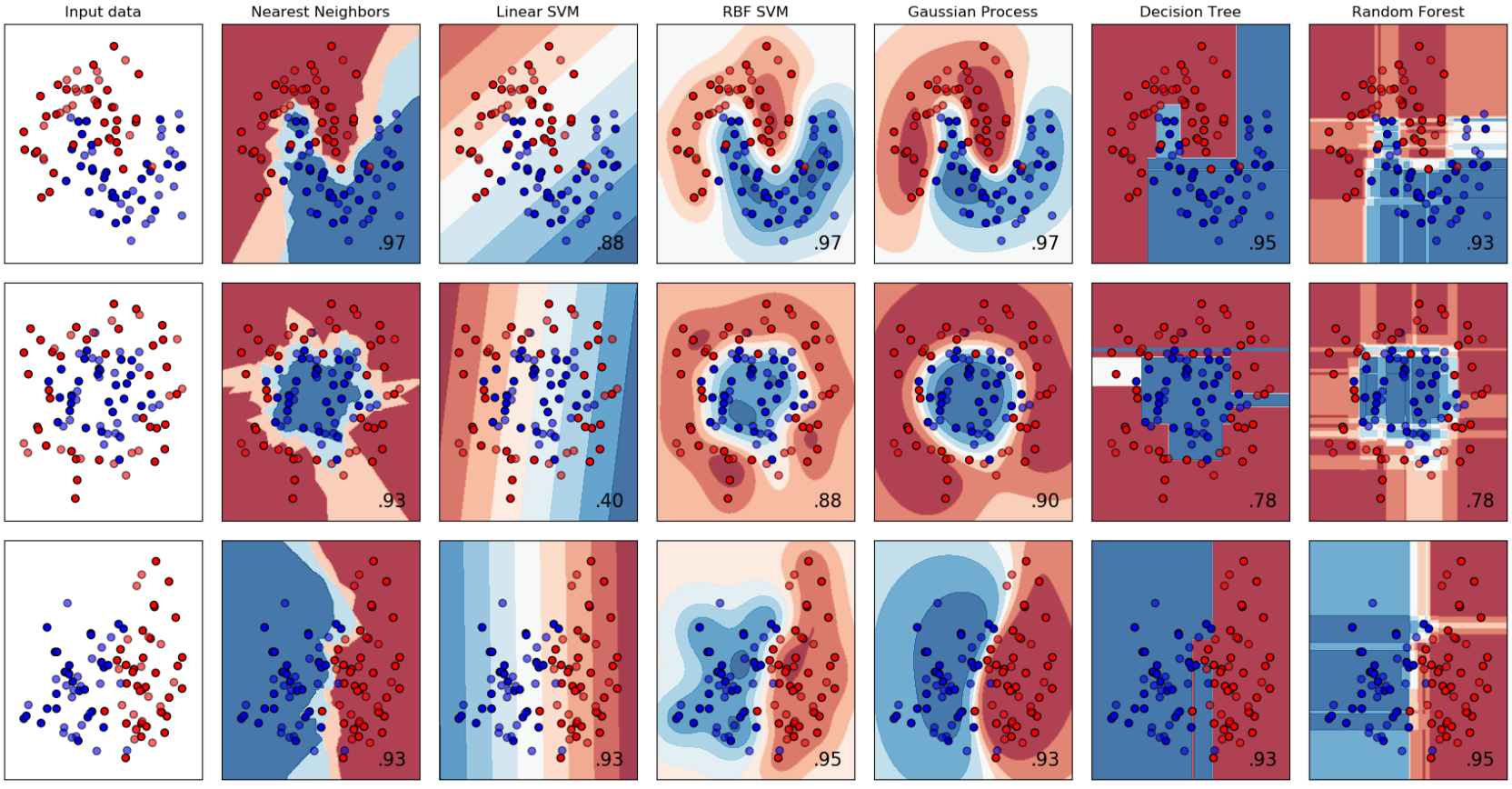
We want to learn about f

Linear regression solution for the problem of estimating f

- We assume $f(\cdot)$ is linear on coefficients β and has an additive separable stochastic component ε.
- Then estimating $f(\cdot)$ using $\hat{f}(\cdot)$ $\xrightarrow{\text{reduced to}}$ estimating β using $\widehat{\beta}$

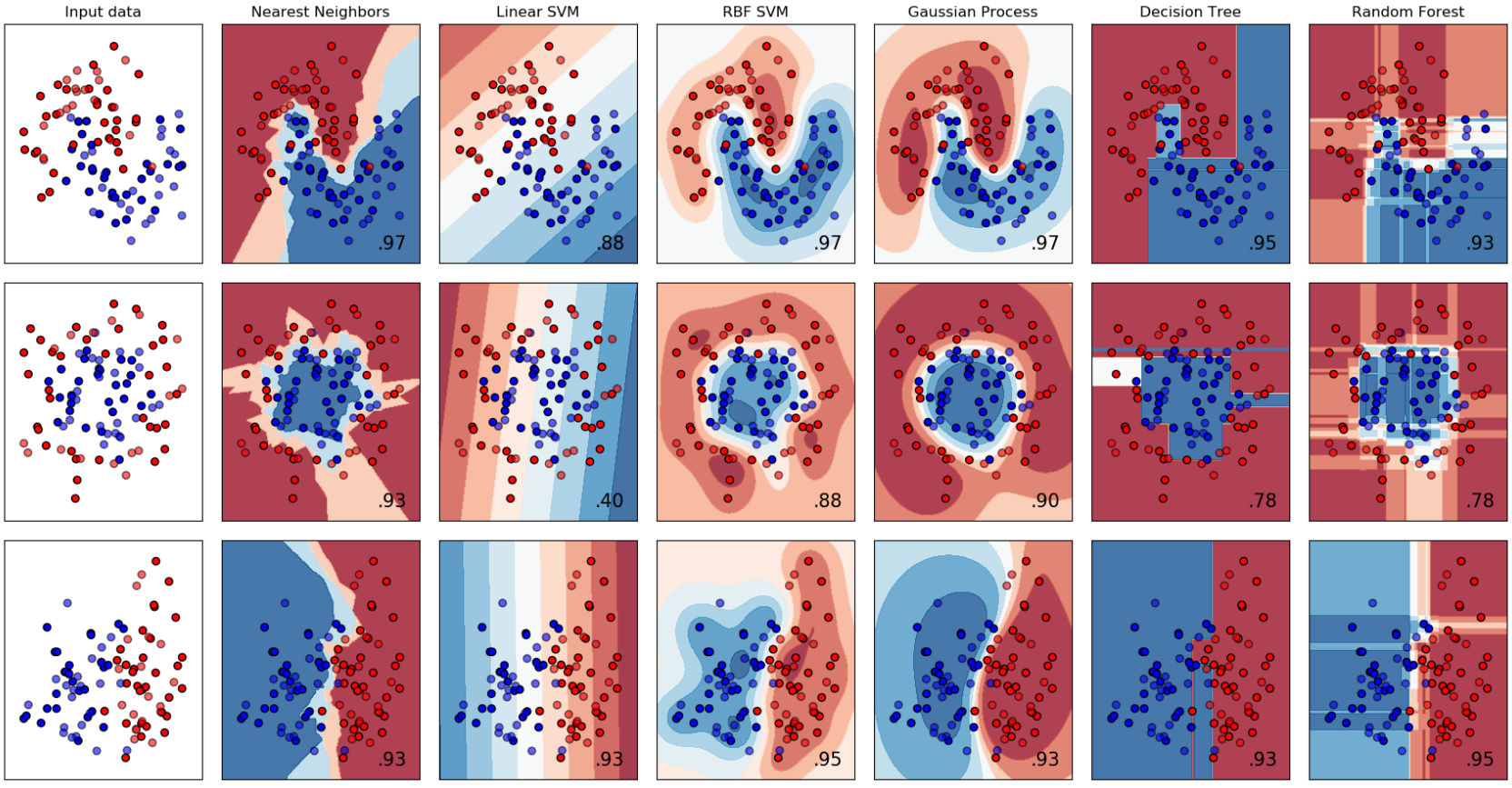
How Do We Estimate f?

Model types



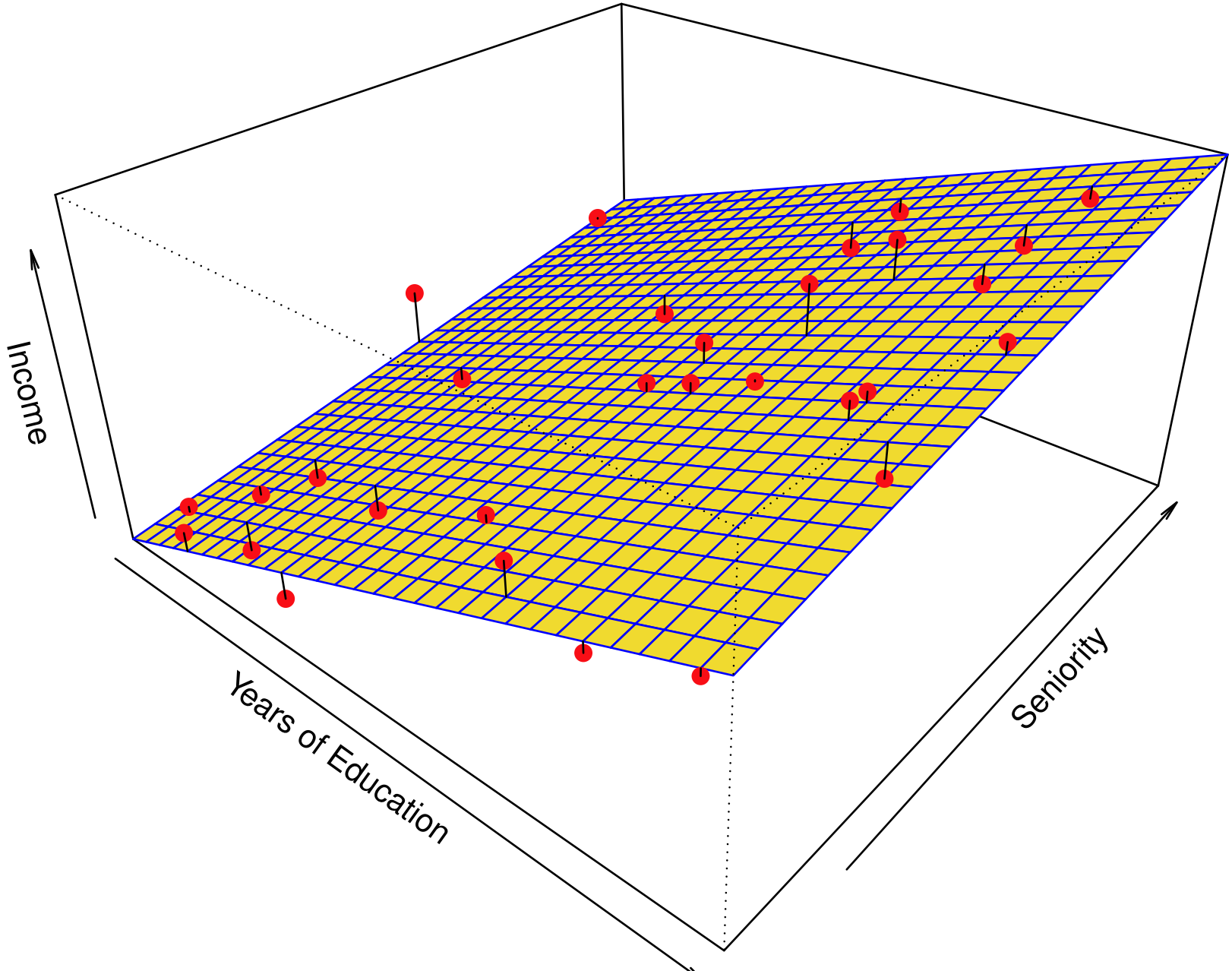
How do we choose a model? (source: sklearn.org)

“No free lunch” theorem



How do we choose a model? (source: sklearn.org)

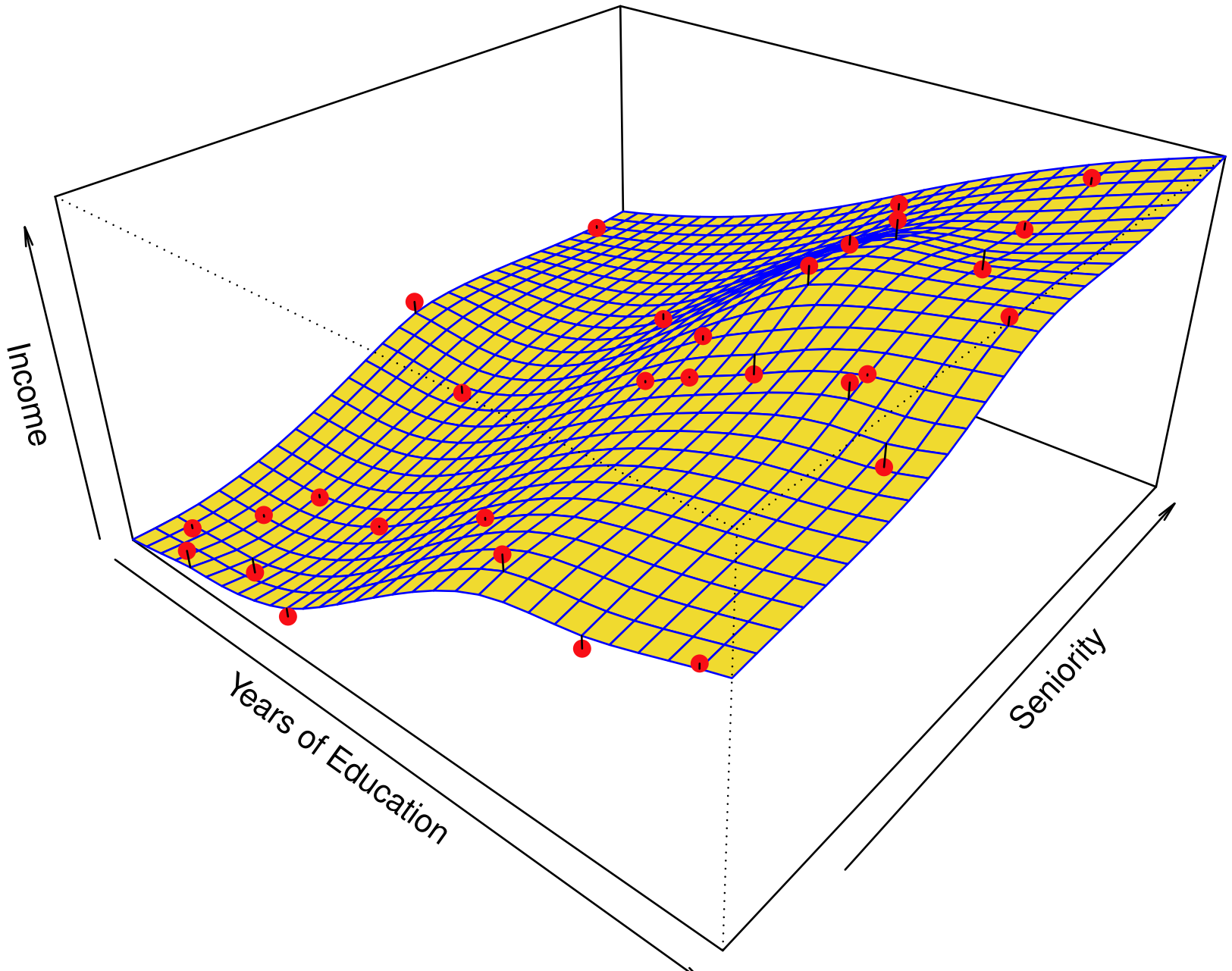
Parametric models



Parametric models

- Objective: Estimate parameters such as $\hat{\beta}$ from training data
- Examples: OLS, discriminant analysis, logistic regression, neural networks
- Assumptions about the functional form (i.e. function is linear)
- Pros: More interpretable, faster, usually better for inference, require less data
- Cons: Unlikely to closely match the underlying function

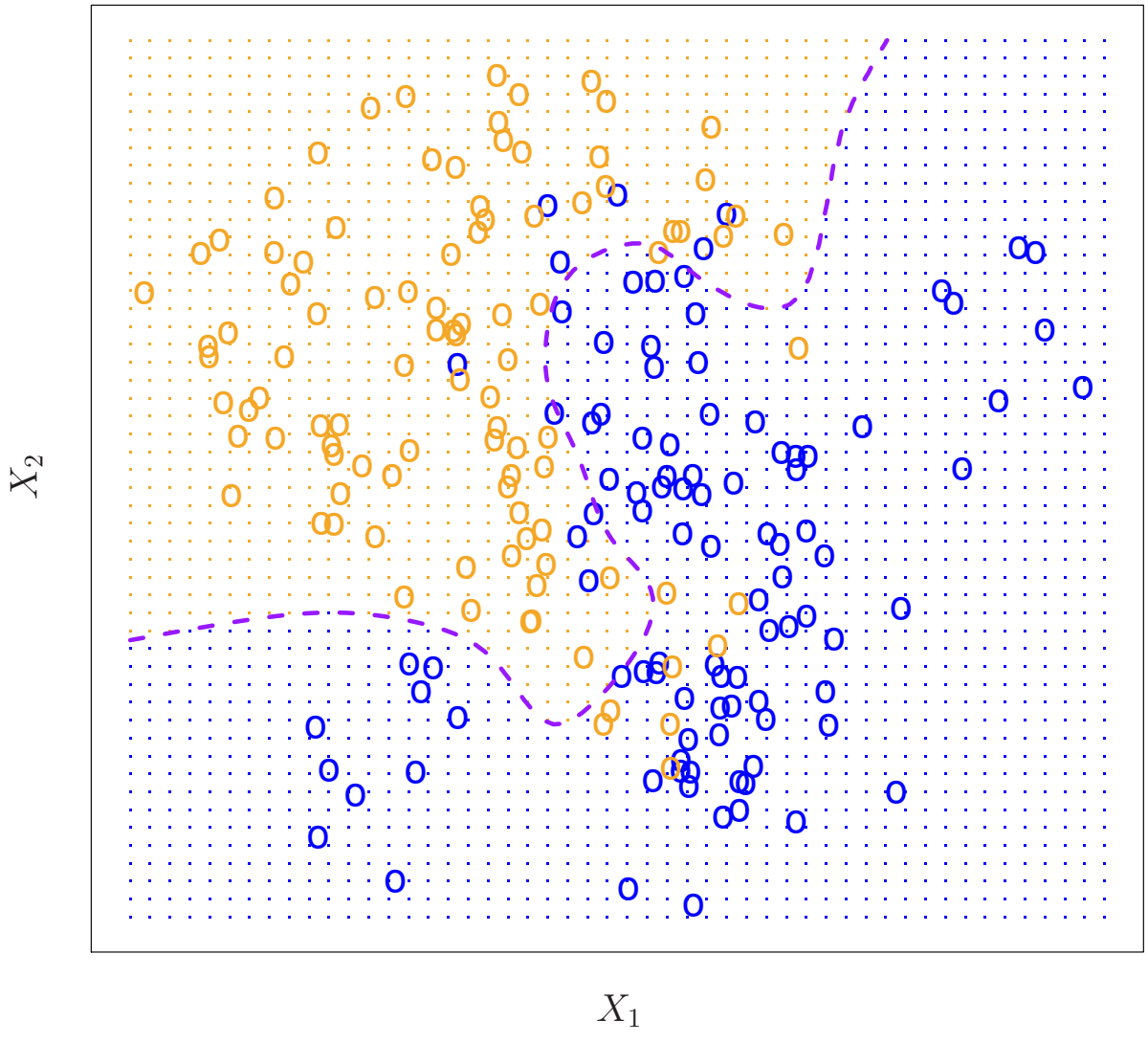
Non-parametric models



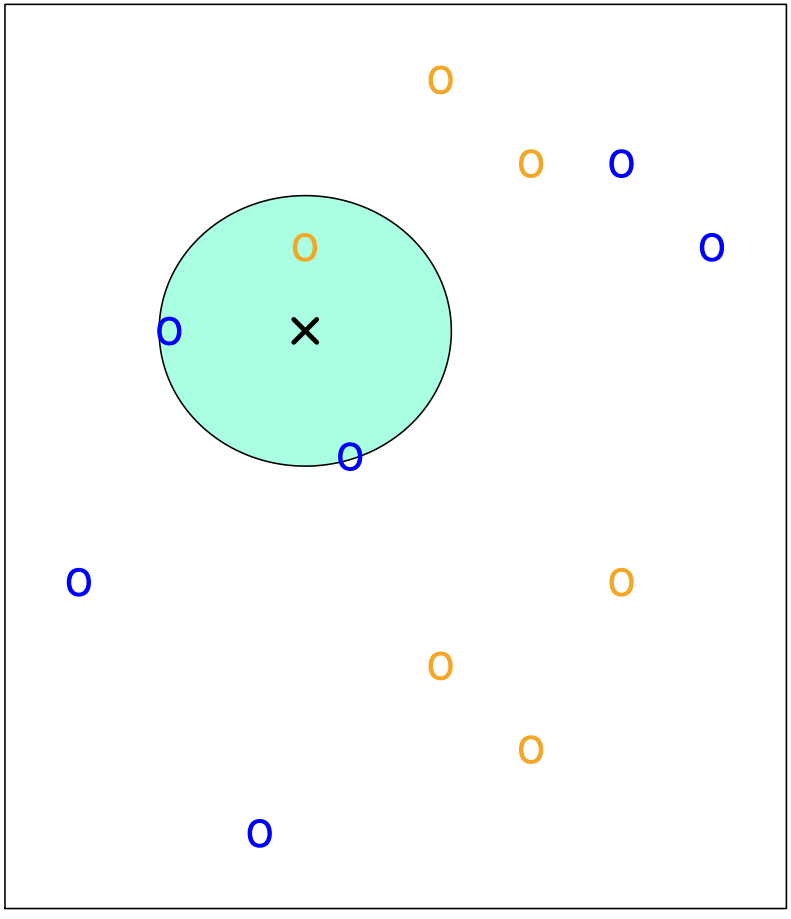
Non-parametric models

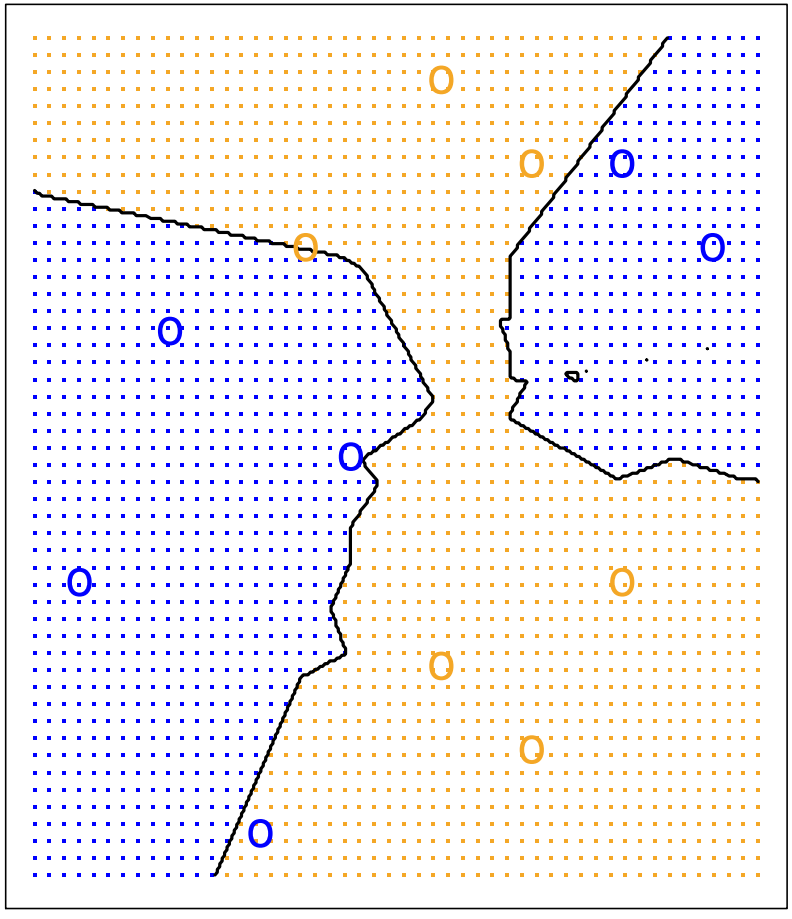
- Objective: Get as close as possible to the data
- Examples: decision trees, KNN, splines
- Fewer assumptions about the functional form
- Pros: Fewer assumptions about the underlying functional form, sometimes more accurate
- Cons: slow, prone to **overfitting**, **curse of dimensionality**, require a lot more data

Non-parametric models example: KNN

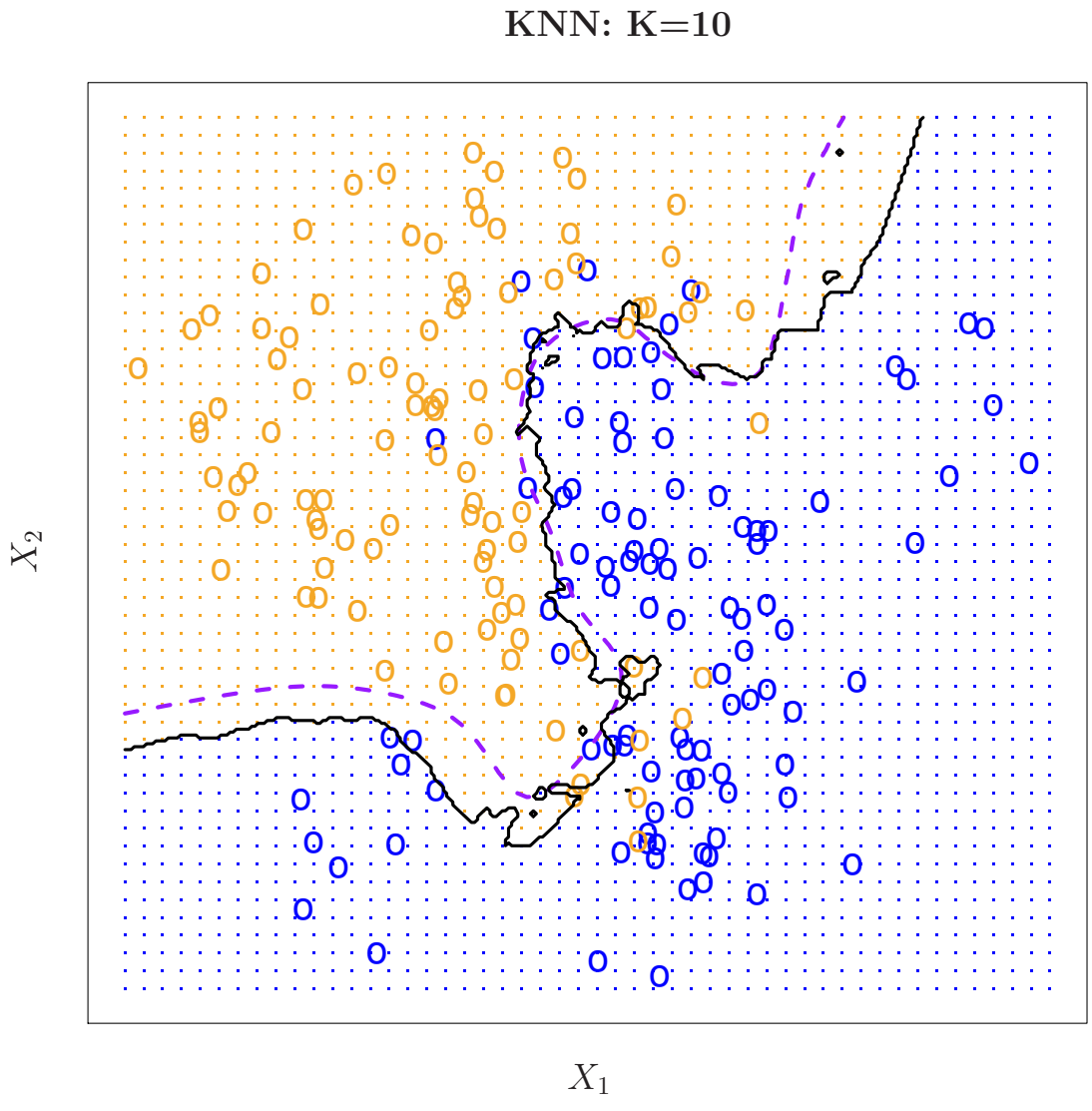


Non-parametric models example: KNN





Non-parametric models example: KNN



Curse of Dimensionality

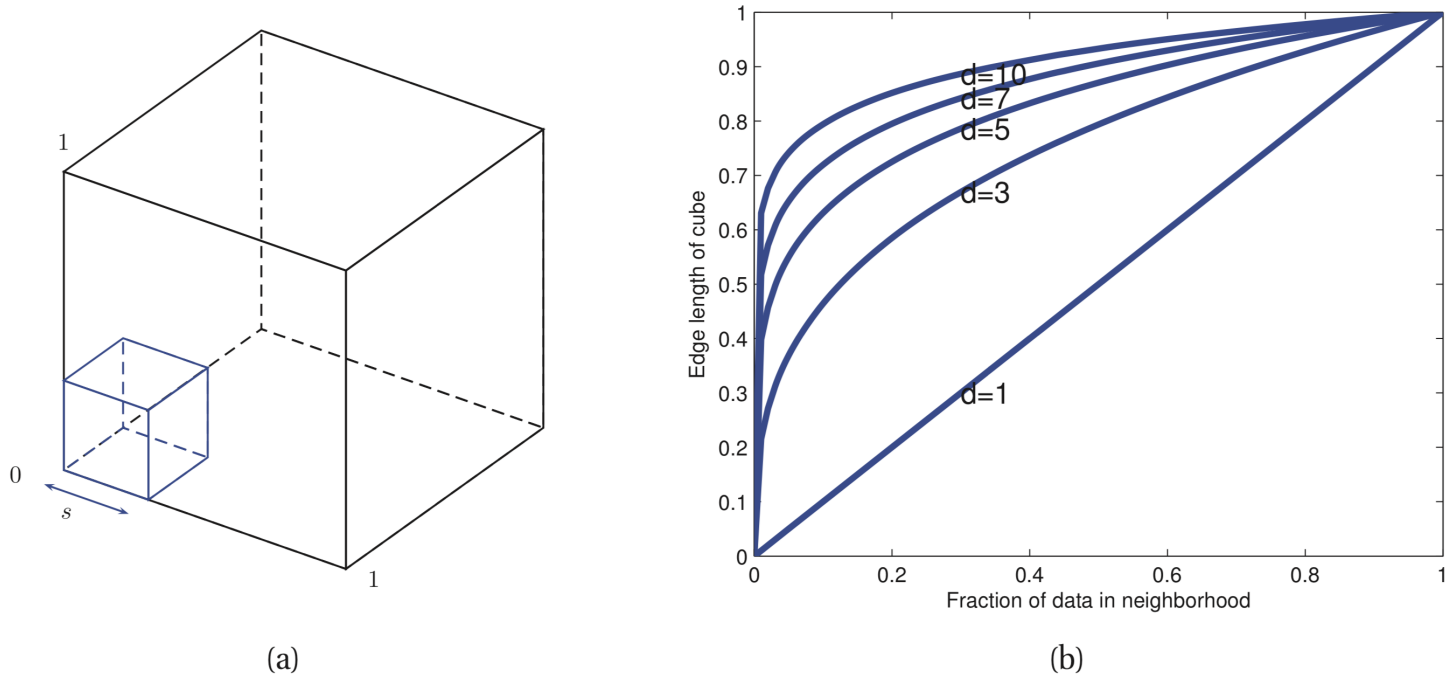


Figure 1.16 Illustration of the curse of dimensionality. (a) We embed a small cube of side s inside a larger unit cube. (b) We plot the edge length of a cube needed to cover a given volume of the unit cube as a function of the number of dimensions. Based on Figure 2.6 from (Hastie et al. 2009). Figure generated by `curseDimensionality`.

Source: *Machine Learning: A Probabilistic Perspective*

Flexible vs. inflexible models

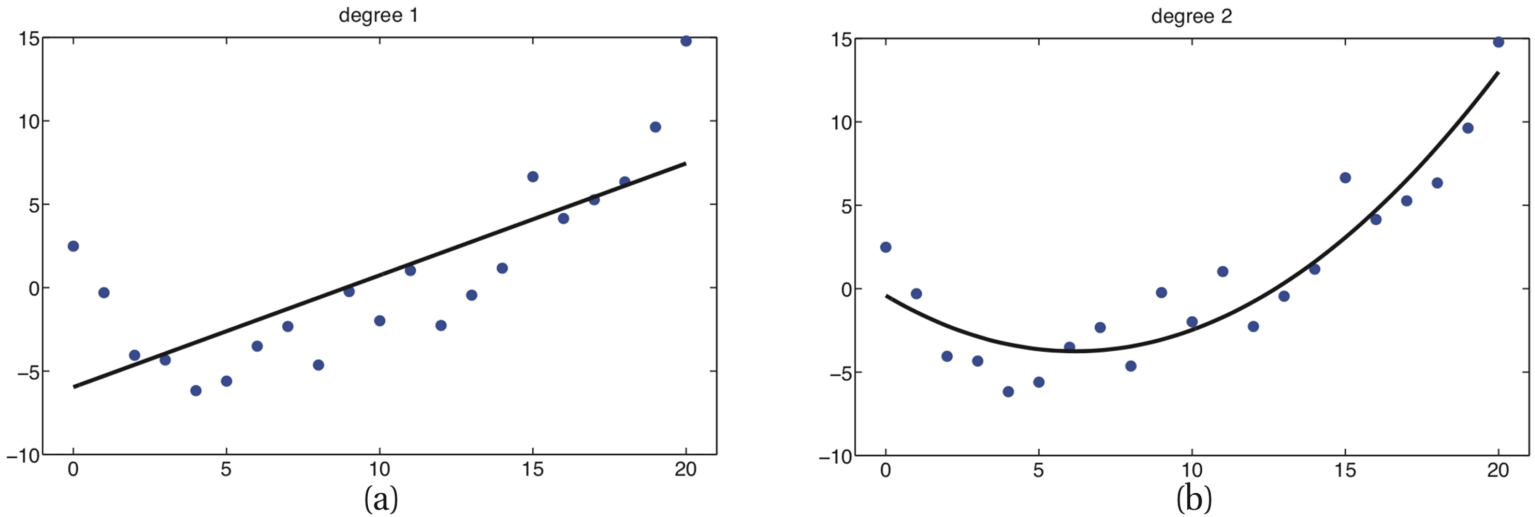


Figure 1.7 (a) Linear regression on some 1d data. (b) Same data with polynomial regression (degree 2). Figure generated by `linregPolyVsDegree`.

Source: *Machine Learning: A Probabilistic Perspective*

Flexible vs. inflexible models (overfitting)

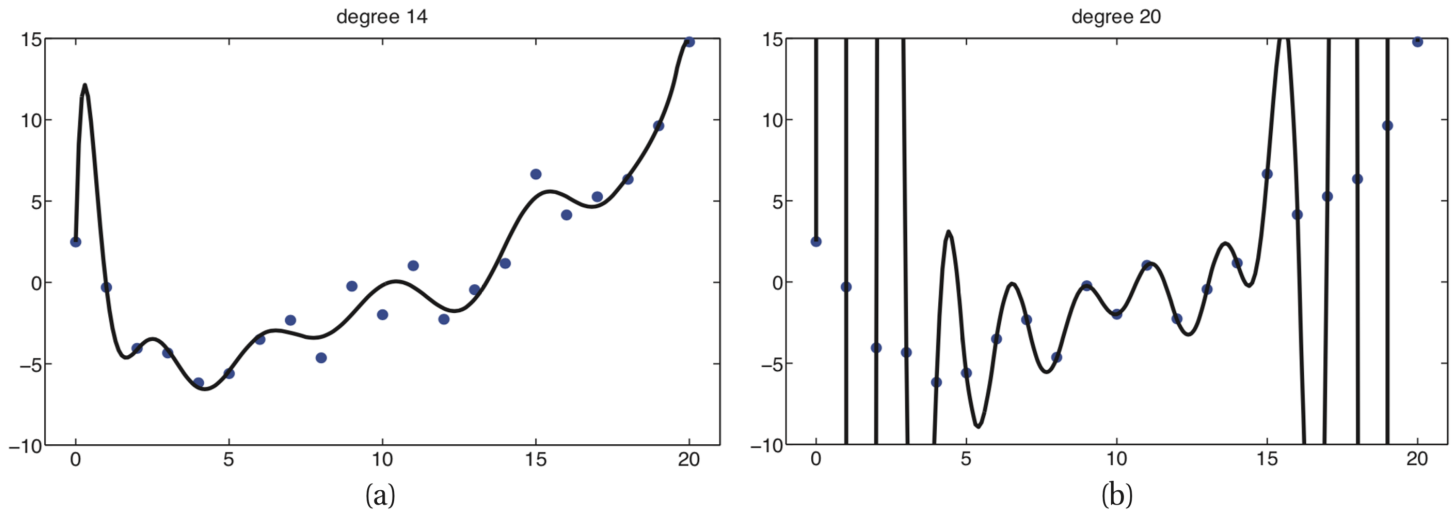
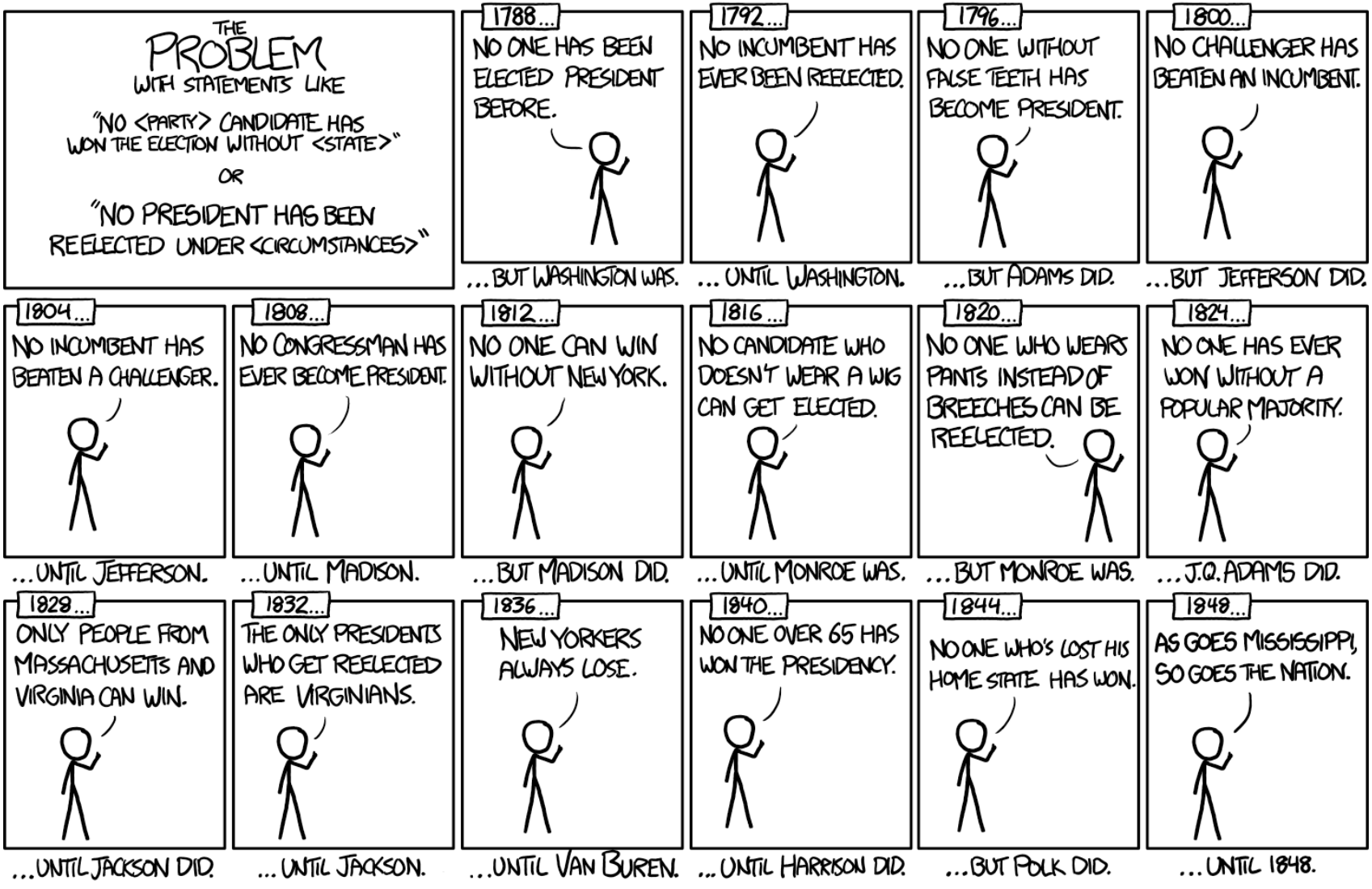


Figure 1.18 Polynomial of degrees 14 and 20 fit by least squares to 21 data points. Figure generated by `linregPolyVsDegree`.

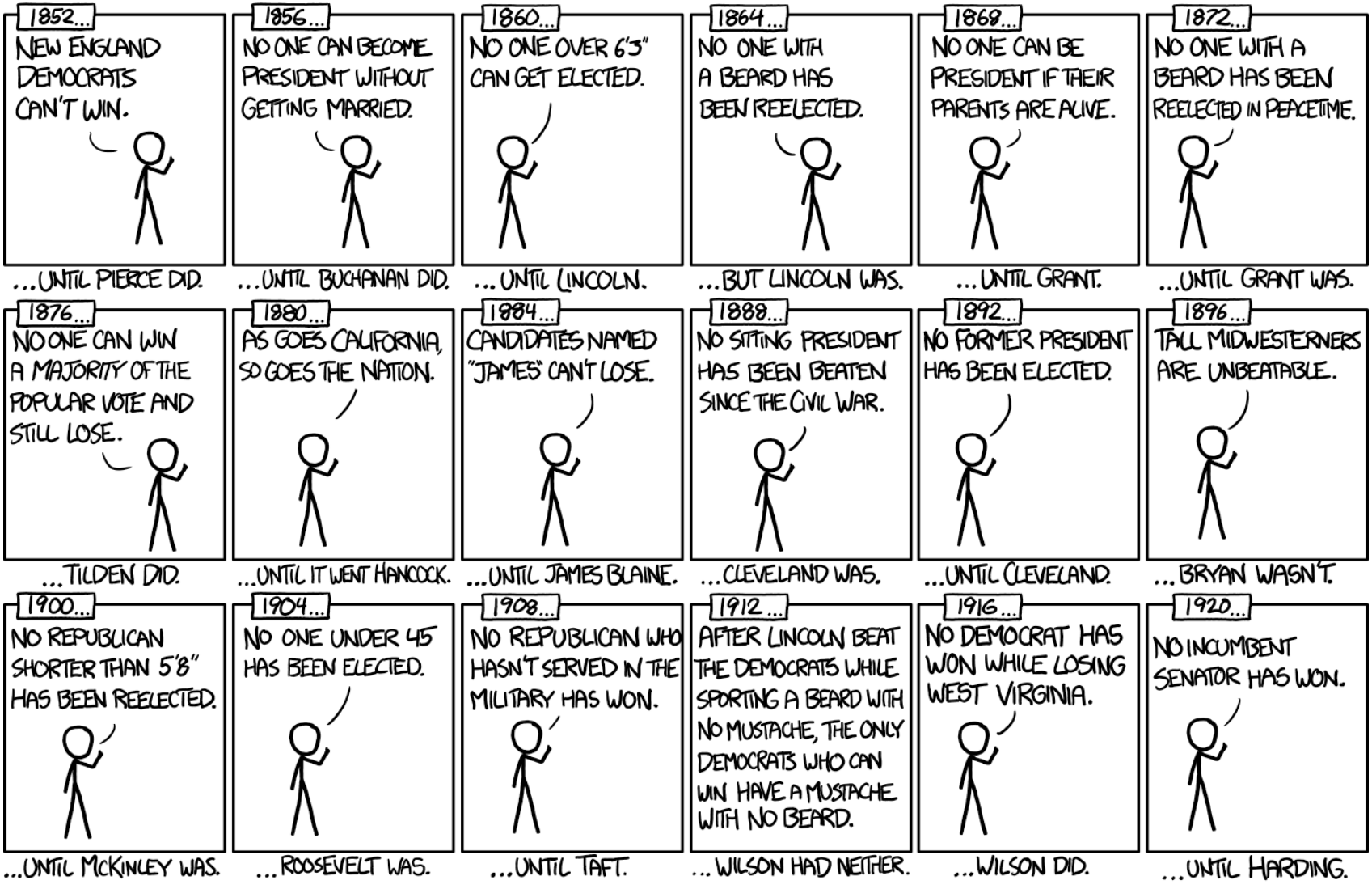
Source: *Machine Learning: A Probabilistic Perspective*

Overfitting



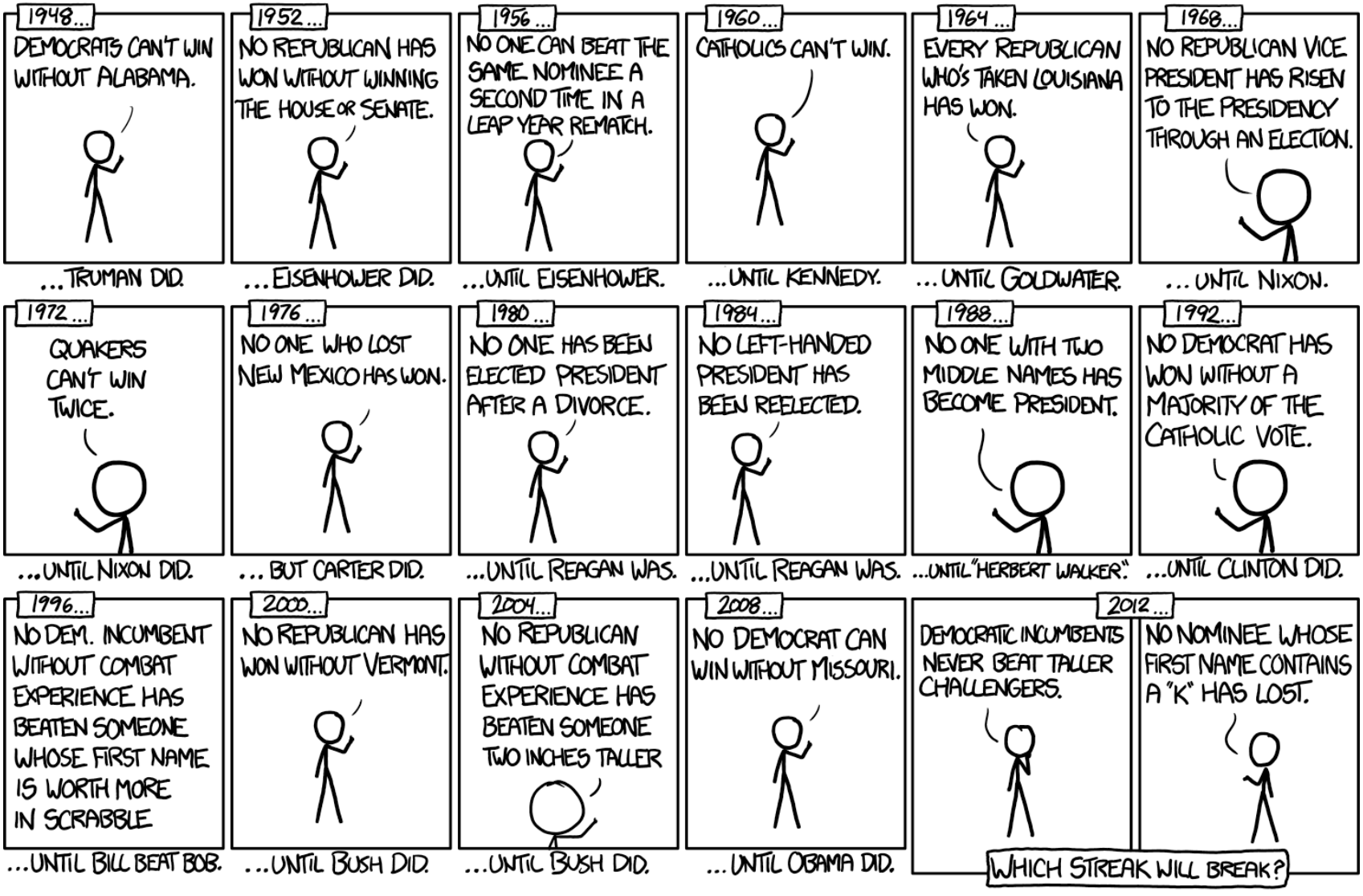
Source: xkcd.com

Overfitting



Source: xkcd.com

Overfitting

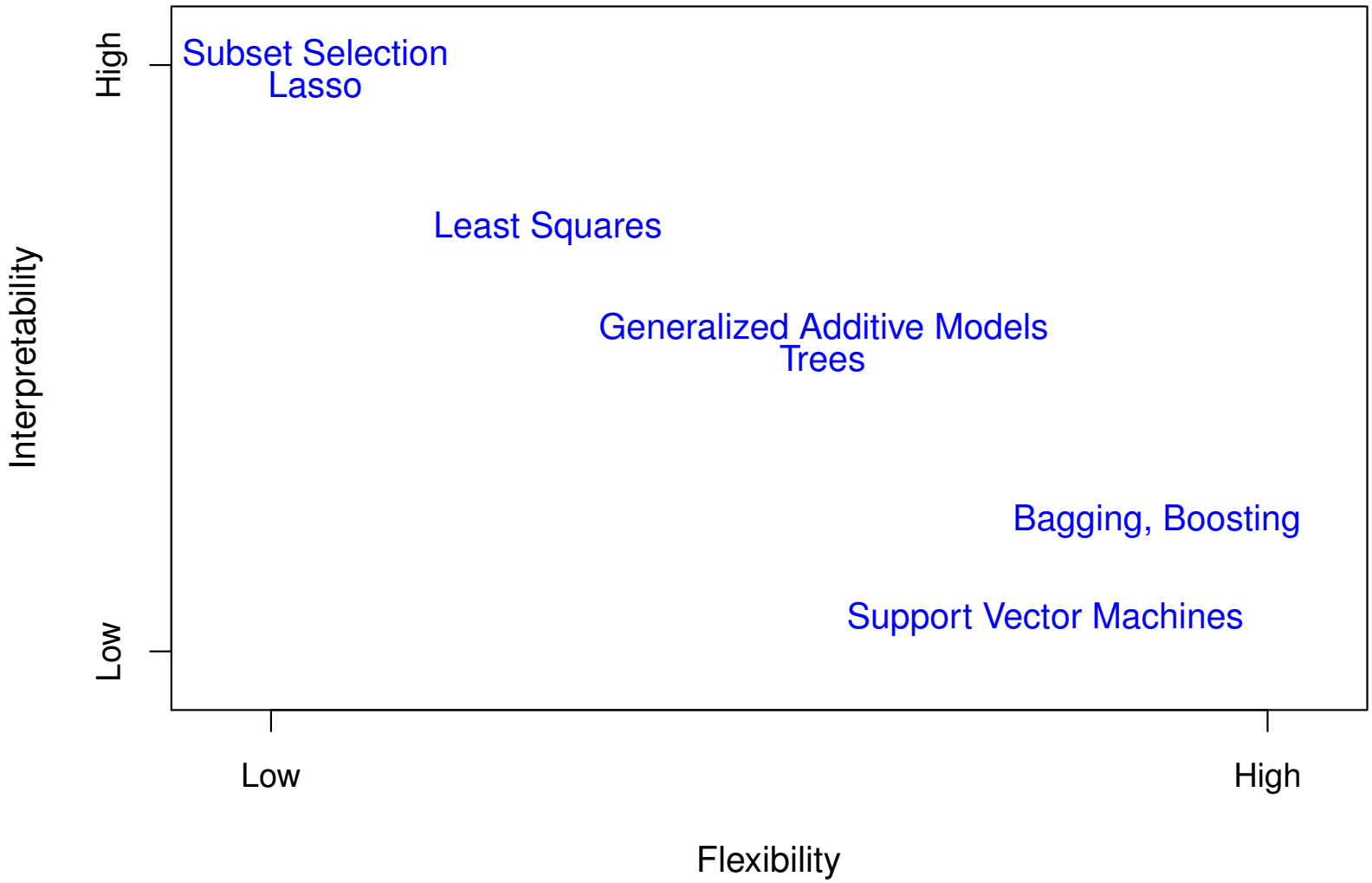


Source: xkcd.com

Question

What is the difference between flexible/inflexible and parametric/non-parametric?

The Trade-Off Between Prediction Accuracy and Model Interpretability



How do we evaluate f?

Training a model: Out of sample validation

- To evaluate a model's performance, we usually withhold some of the data from the modeling and estimation process for **out of sample validation**. This is called **test data.**
- The remaining data, called **training data** is used in the modeling and estimation process, but usually not for evaluation.
- We make predictions using this held out **test data** to estimate **generalization error**, or how well the model generalizes to new data.

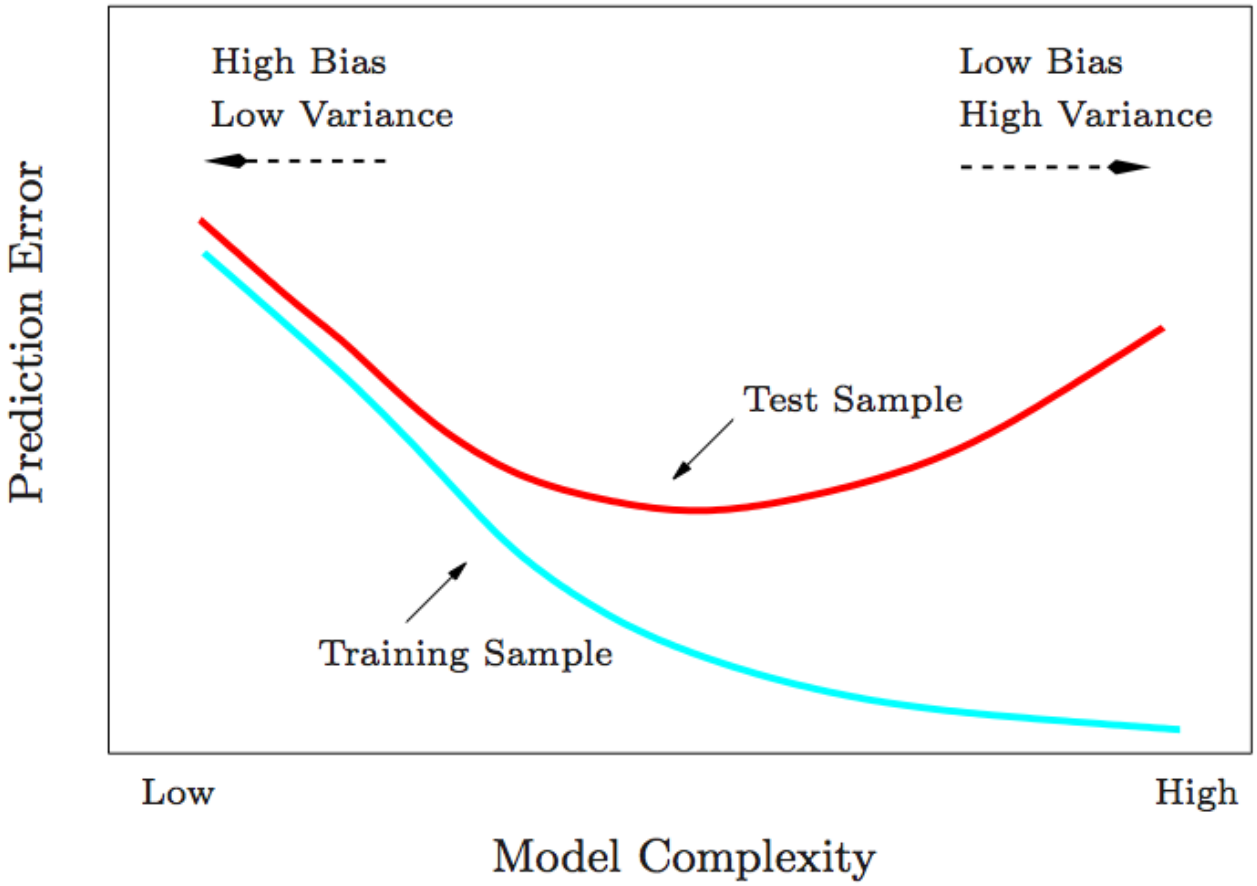
Training a model: Minimizing loss

- We want $f(X)$ to be "close" to Y as possible
- The **loss function** tells us how well a candidate function $f(x)$ predicts **out of sample**
- We often define the **loss function** as the **squared loss**: $L(Y, f(X)) \triangleq (Y - f(X))^2$
- The goal in regression is to minimize this **loss function** (for classification we often use **cross-entropy** loss)
- Various approaches to minimizing the loss (e.g. varieties of **gradient descent**)
- The **ideal** or **optimal** predictor of Y with regard to **mean-squared prediction error** is the function $\hat{f}(x) = E(Y|X = x)$ that minimizes $E[(Y - f(X))^2|X = x]$ over all functions f at all points $X = x$.

Training a model: Error

- **Mean-squared prediction error**: a measure used to evaluate performance of $\hat{f}(x)$ for regression problems
  - Training error: $MSE_{Tr} = Ave_{i \in Tr}[y_i - \hat{f}(x_i)]^2$ (biased when overfitting)
  - Test error: $MSE_{Te} = Ave_{i \in Te}[y_i - \hat{f}(x_i)]^2$ (mitigates bias by using **out of sample** data)
- **Misclassification error rate**: a measure used to evaluate performance of $\hat{C}(x)$ for classification problems
  - Training error: $Err_{Tr} = Ave_{i \in Tr} I[y_i \neq \hat{C}(x_i)]$
  - Test error: $Err_{Te} = Ave_{i \in Te} I[y_i \neq \hat{C}(x_i)]$
- The **Bayes classifier** has smallest error in the population (we usually don't have access to a population though!).

Bias-Variance Tradeoff



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

Training a model: Bias and Variance

Recall that the **bias** of an estimator is defined as follows:

$$\text{bias}(\hat{\theta}) \triangleq \mathbb{E}\left[\hat{\theta}\right] - \theta$$

and the **variance** of an estimator is defined as follows:

$$\text{var}(\hat{\theta}) \triangleq \mathbb{E}\left[(\hat{\theta} - \mathbb{E}\left[\hat{\theta}\right])^2\right]$$

After some algebra, we can decompose MSE into bias squared and variance:

$$\text{MSE}\left(f(x), \hat{f}(x)\right) = \text{bias}^2\left(\hat{f}(x)\right) + \text{var}\left(\hat{f}(x)\right)$$

Training a model: Bias and Variance

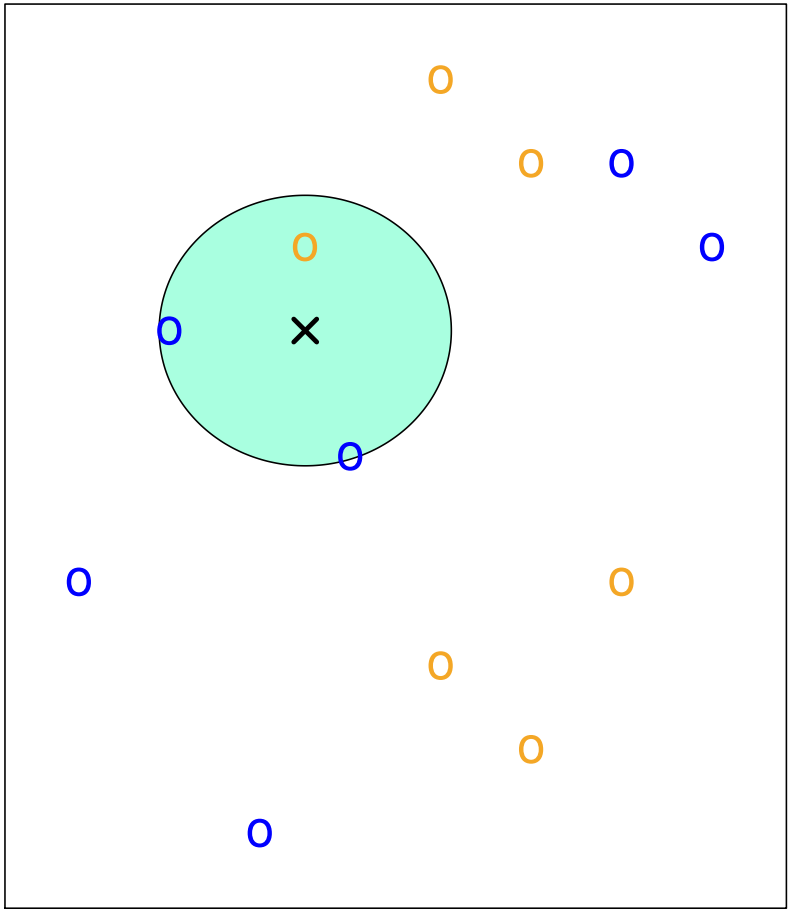
In the context of regression, models are biased when:

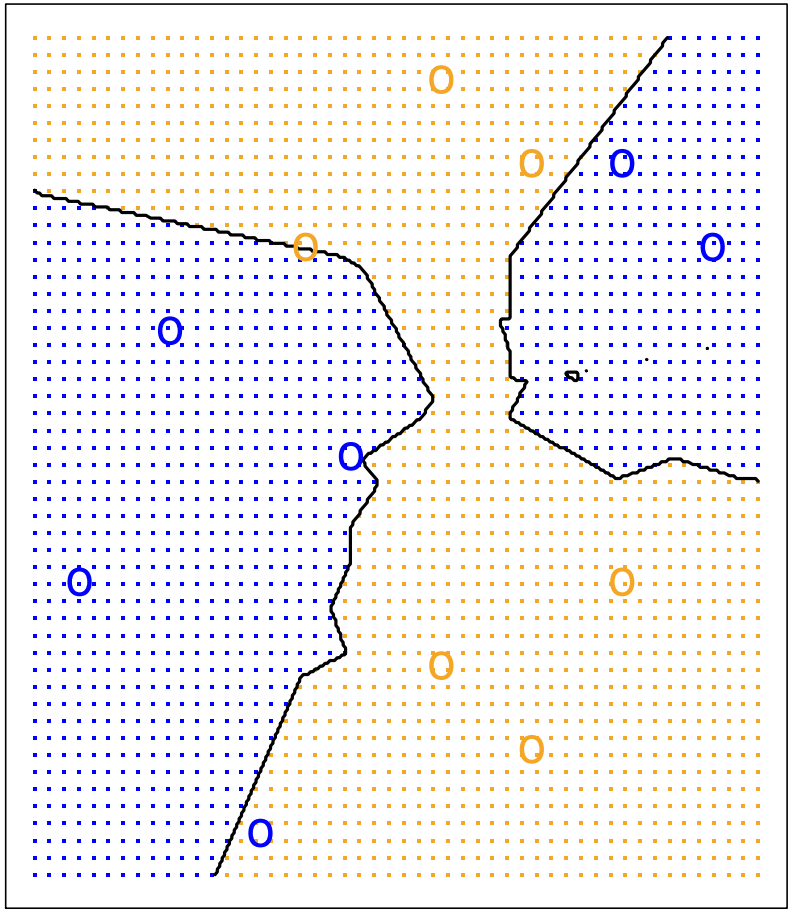
- Parametric: The form of the model does not incorporate all the necessary variables (omitted variable bias)
- Parametric: The functional form is too simple (e.g. a linear approximation)
- Non-parametric: The model provides too much smoothing.

In the context of regression, models are variable when:

- Parametric: The form of the model incorporates too many variables.
- Parametric: The functional form is too complex.
- Non-parametric: The model does not provide enough smoothing.

Model evaluation example: KNN





K is a tuning parameter; choose it using **out of sample validation**

Model evaluation example: KNN