DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

September 15, 2020

Class Outline

- Introduction – 1 week
  – Probability and linear algebra review
- Linear regression – 2 weeks
- Classification - 5 weeks
  – Linear classifiers: logistic regression, LDA,
  – Non-linear: kNN, decision trees, SVM, Naïve Bayes
  – Ensembles: random forest, boosting
  – Model selection, regularization, cross validation
- Neural networks and deep learning – 2 weeks
  – Back-propagation, gradient descent
  – NN architectures (feed-forward, convolutional, recurrent)
- Ethics of AI – 1 week
- Adversarial ML – 1 lecture
  – Security of ML at testing and training time

Schedule and Resources

- Instructors
  - Alina Oprea
  - TAs: Alex Wang, Matthew Jagielski
- Schedule
  - Tue 11:45am – 1:25pm, Thu 2:50-4:30pm EST
  - Zoom
  - Office hours:
    - Alina: Tue 4:00-5:30pm; Thu 4:30 – 5:30 pm (Zoom)
    - Matthew: Monday 3:00-4:00pm; Friday 9:00-10:00am (Zoom)
    - Alex: Wednesday: 5:00-7:00pm
    - Links on Canvas under “Syllabus”
- Online resources
  - Slides / recordings will be posted after each lecture
  - Use Piazza for questions
  - Canvas as course management system

Grading

- Assignments – 25%
  - 4-5 assignments and programming exercises based on studied material in class
- Final project – 35%
  - Select your own project based on public dataset
  - Submit short project proposal and milestone
  - Presentation at end of class (10 min) and written report
  - Team of 2 students
- Exam – 35%
  - One exam second half of November
  - Tentative date: November 19
- Class participation – 5%
  - Participate in class discussion/Zoom and on Piazza

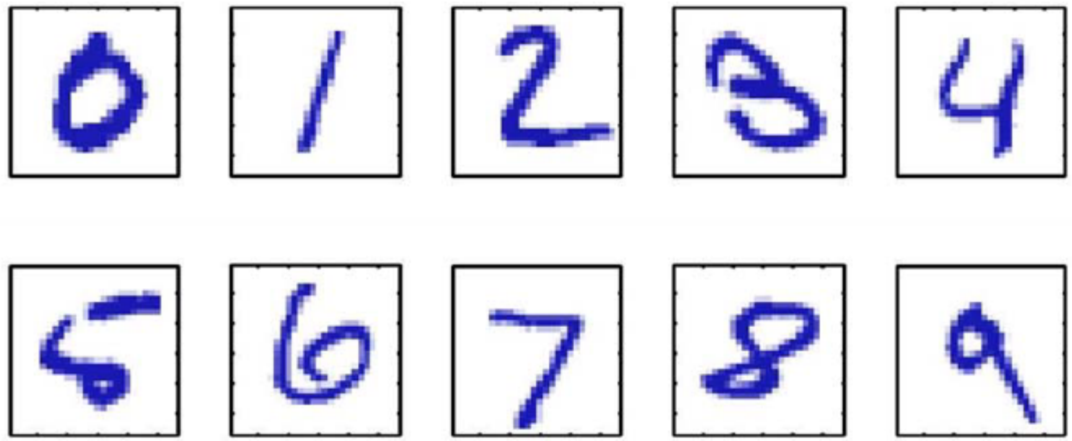
Announcements

- HW 1
  - Will be out this Thursday, Sept. 17
  - Will be due on Monday, Sept. 28
- Python tutorials
  - Numpy tutorial by Matthew Jagielski
    - Friday, Sept. 18, 1-2pm
  - Panda data frames tutorial by Alex Wang
    - Wed, Sept. 23, 5-6pm
  - Same Zoom links as office hours

Outline

- Supervised learning
    - Classification
    - Regression
- Unsupervised learning
    - Clustering
- Bias-Variance Tradeoff
- Occam’s Razor
- Probability review

Example 1
Handwritten digit recognition



Images are 28 x 28 pixels

Represent input image as a vector $\mathbf{x} \in \mathbb{R}^{784}$

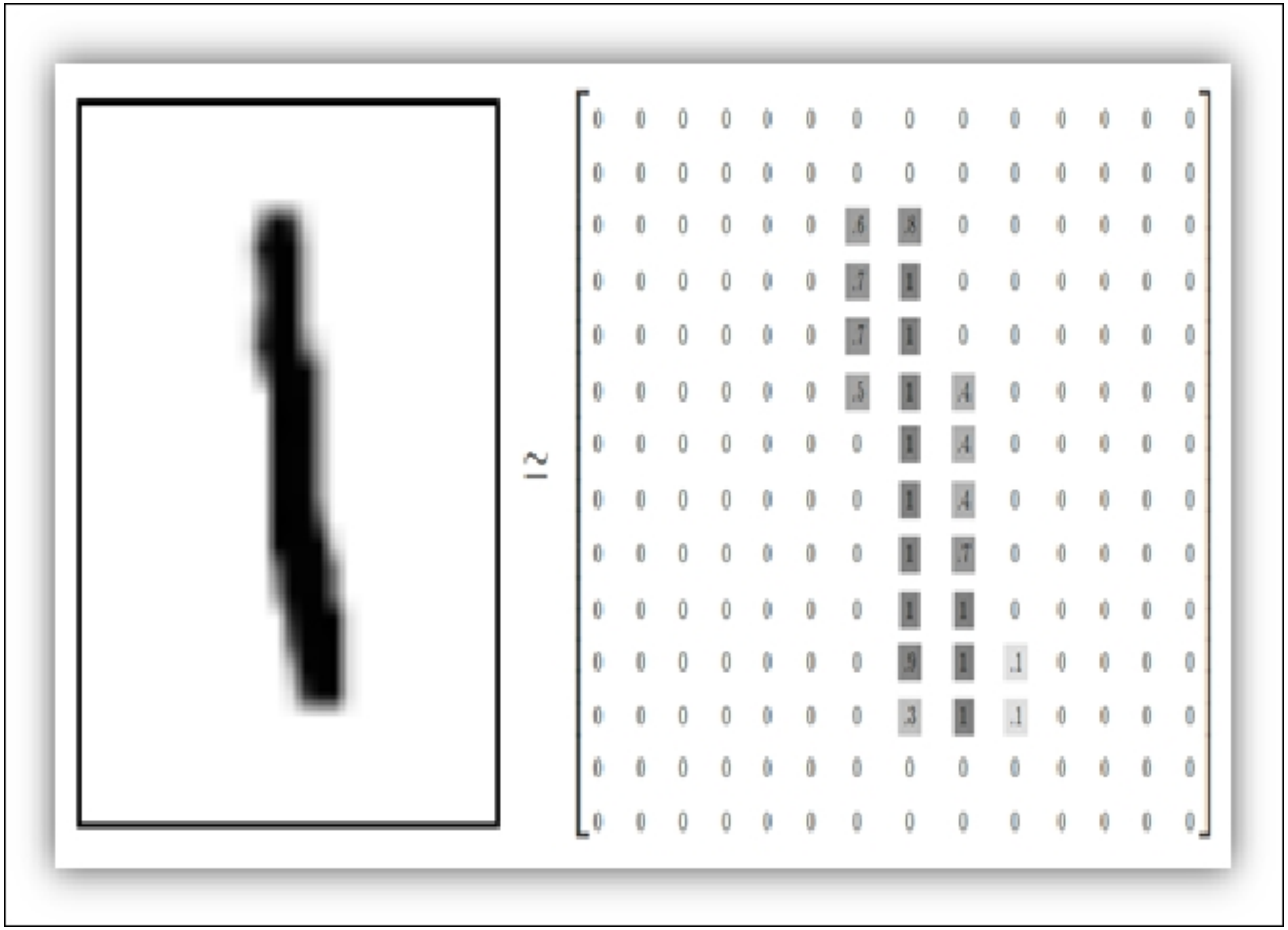
Learn a classifier $f(\mathbf{x})$ such that,

$$f : \mathbf{x} \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

MNIST dataset: Predict the digit

Multi-class classifier

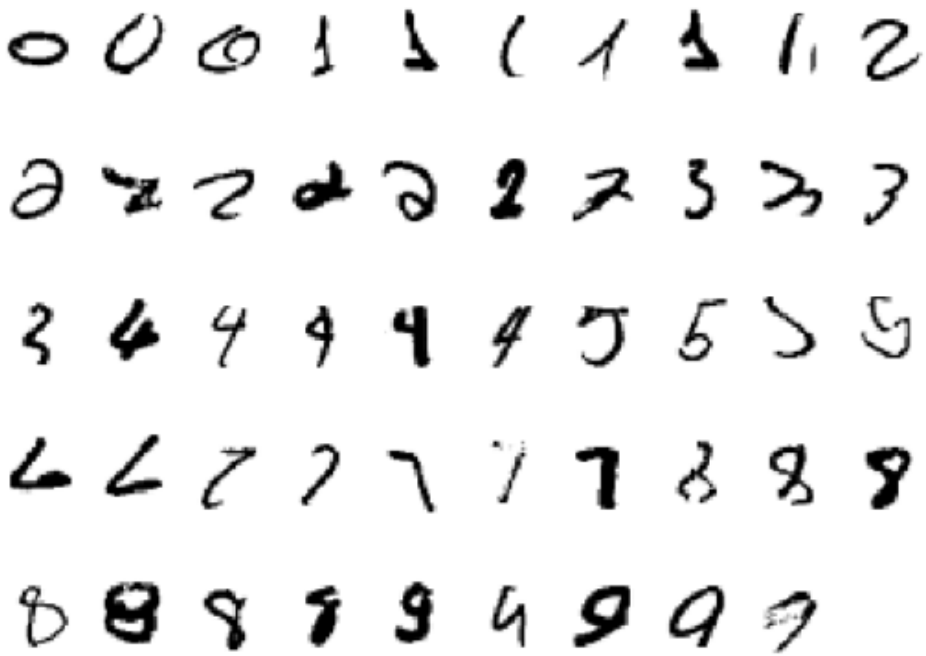
Data Representation



Model the problem

As a supervised classification problem

Start with training data, e.g. 6000 examples of each digit



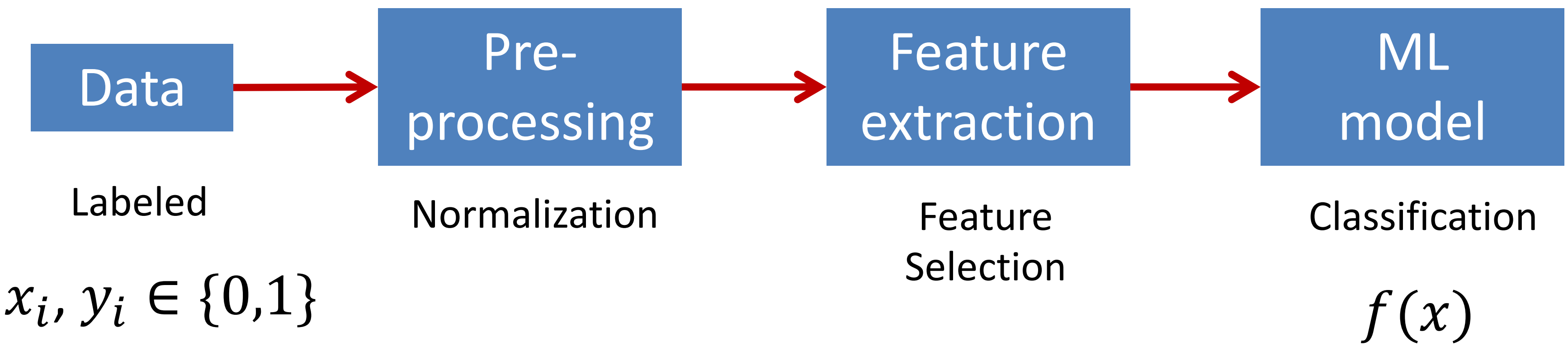
- Can achieve testing error of 0.4%
- One of first commercial and widely used ML systems (for zip codes & checks)

Other examples

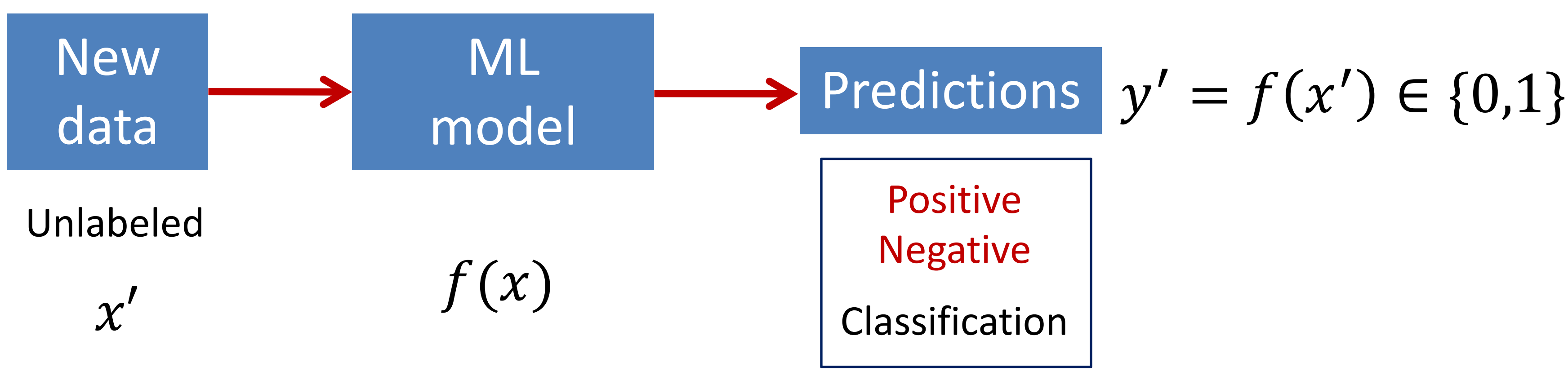
- Spam classification
  - Is my email spam or not?
  - Binary classification
- Weather prediction
  - Will it rain tomorrow or not?
- Healthcare classification
  - Is the patient sick or not?
- Image classification
  - What object does the image depict?

Supervised Learning: Classification

Training

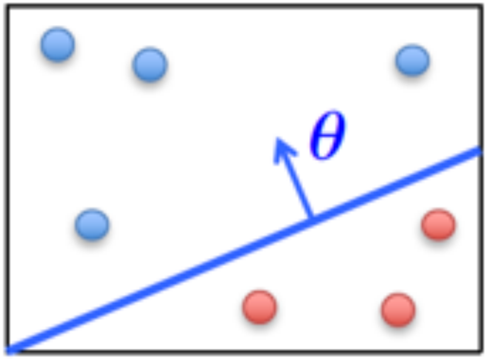


Testing



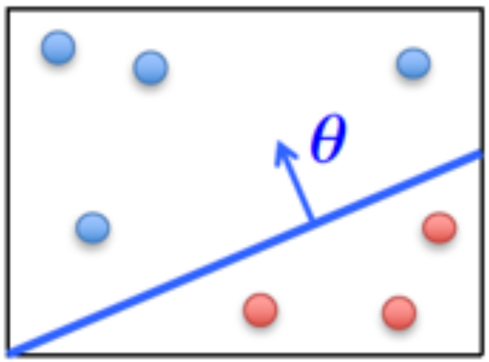
Classification

- Training data
  - $x_i = [x_{i,1}, \dots x_{i,d}]$: vector of image pixels (features)
  - Size $d = 28\text{x}28 = 784$
  - y_i: image label
- Models (hypothesis)
  - Example: Linear model (parametric model)
    - $f(x) = wx + b$
  - Classify 1 if $f(x) > \mathrm{T}$; 0 otherwise

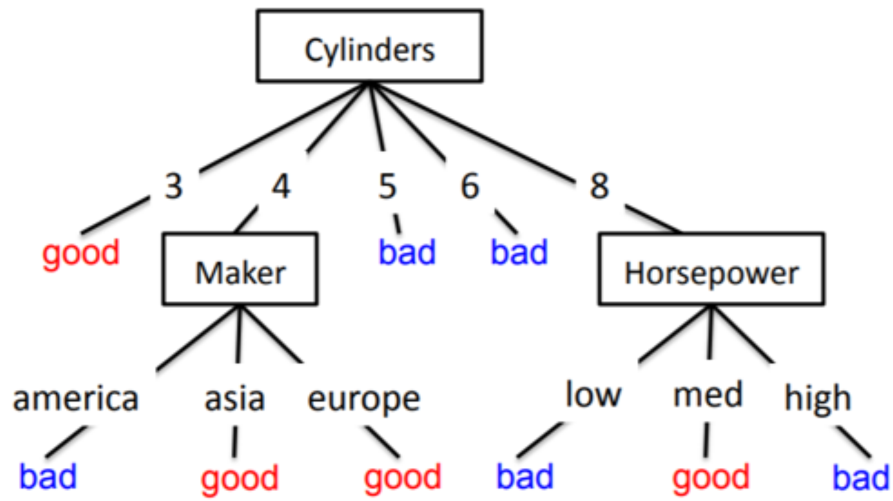


- Classification algorithm
  - Training: Learn model parameters w, b to minimize error (number of training examples for which model gives wrong label)
  - Output: "optimal" model
- Testing
  - Apply learned model to new data and generate prediction $f(x)$

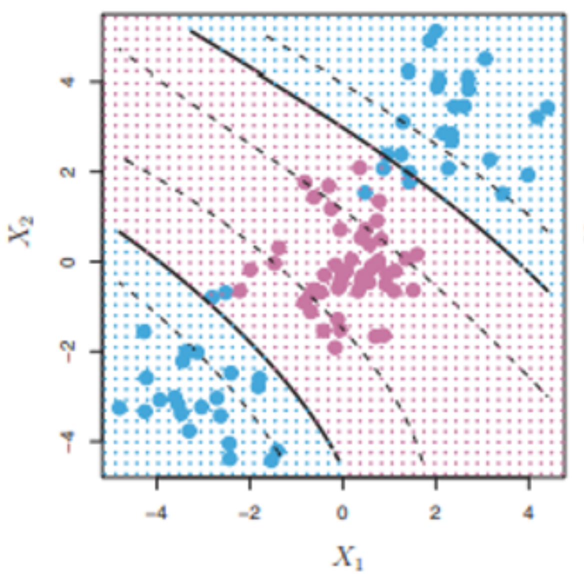
Example Classifiers



Linear classifiers: logistic regression, SVM, LDA



Decision trees



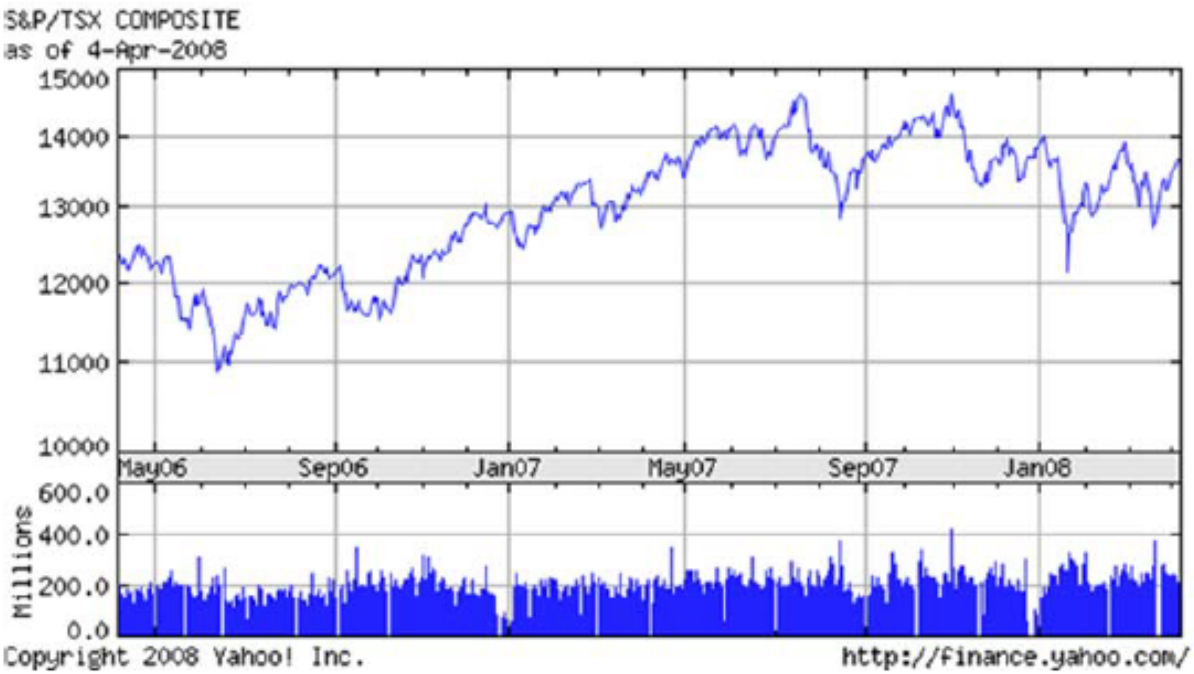
SVM polynomial kernel

Why Multiple Models?

- There is no free lunch in statistics / ML!

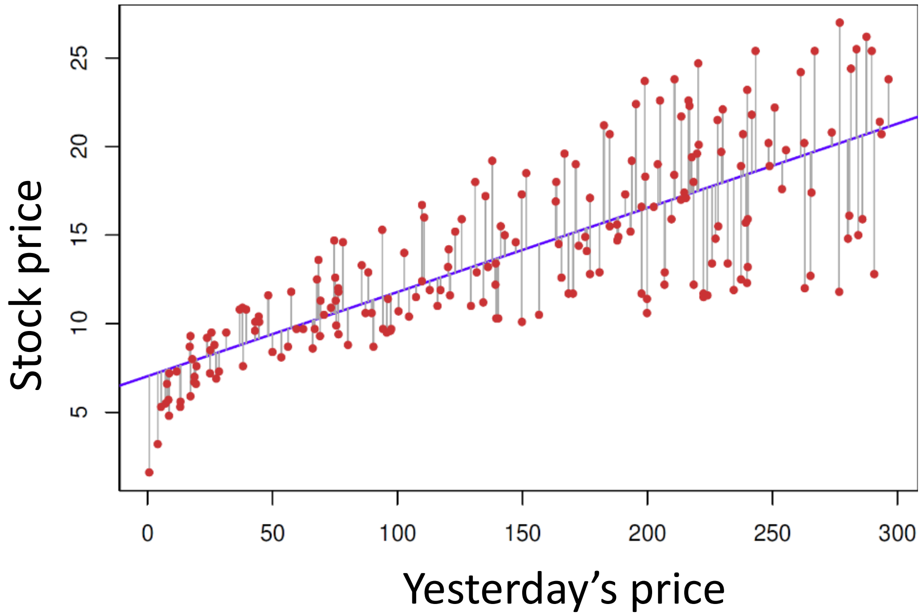
- There is no single model that dominates all
- Performance depends on many things, such as:
  - Data distribution
  - Data dimensionality
  - Quality of data and labeling

Example 2
Stock market prediction



- Task is to predict stock price at future date
- This is a regression task, as the output is continuous

Regression



Linear regression
1 dimension

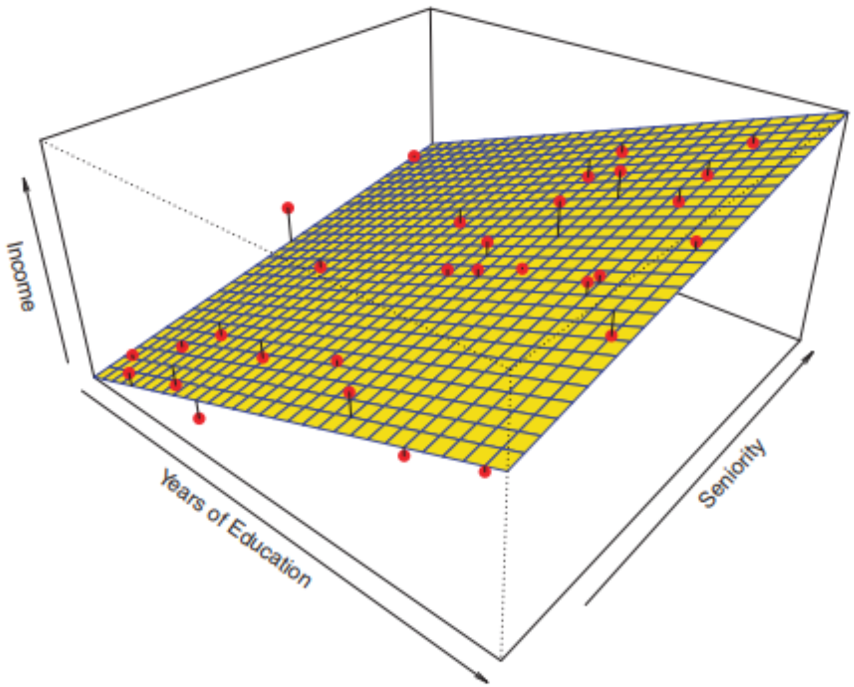
- Suppose we are given a training set of N observations

$$(x_1, \ldots, x_N) \text{ and } (y_1, \ldots, y_N)$$

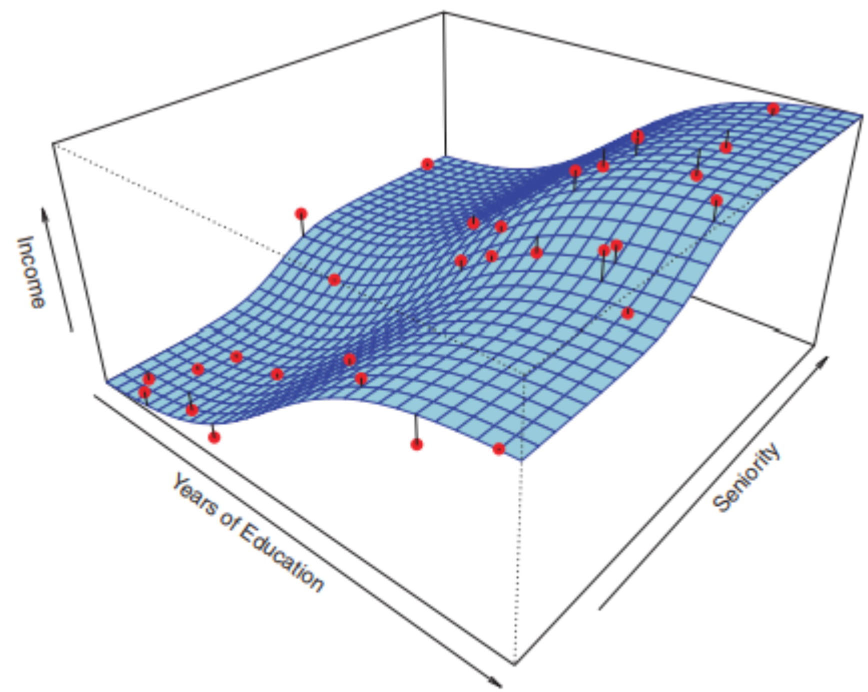
- Regression problem is to estimate y(x) from this data

$x_i = (x_{i1}, \ldots, x_{id})$ - d predictors (features)
y_i - response variable, numerical

Income Prediction



Linear Regression



Non-Linear Regression
Polynomial/Spline Regression

Supervised Learning: Regression

Training

Data → Pre-processing → Feature extraction → ML model

Labeled

$x_i, y_i \in R$

Normalization

Feature Selection

Regression

$f(x)$

Testing

New data → ML model → Predictions

Unlabeled

x'

$f(x)$

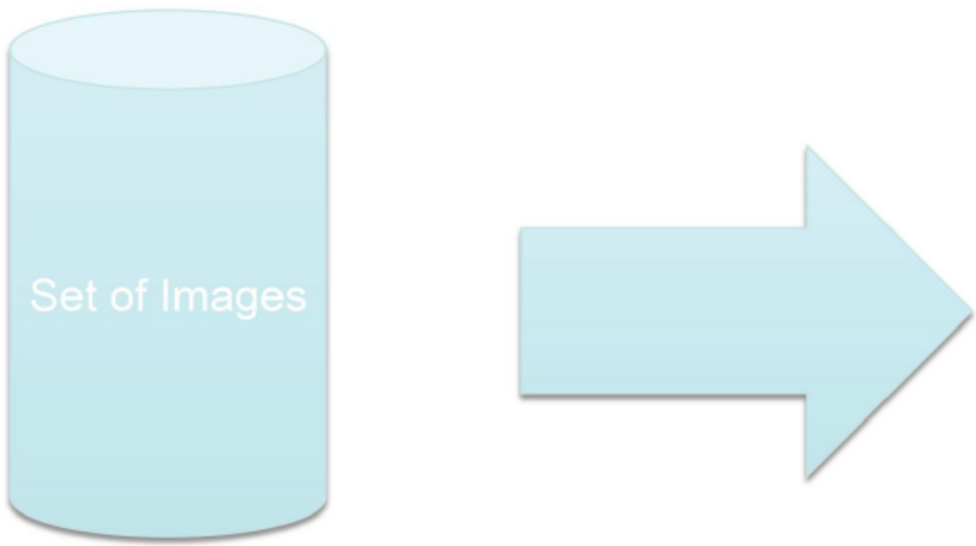
$y' = f(x') \in R$

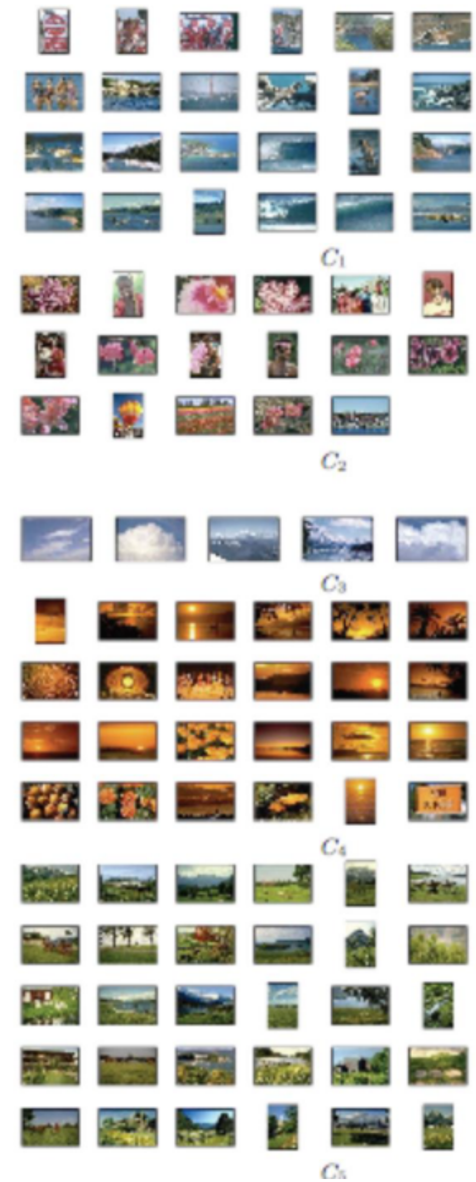
Response variable

Regression

Example 3: image search

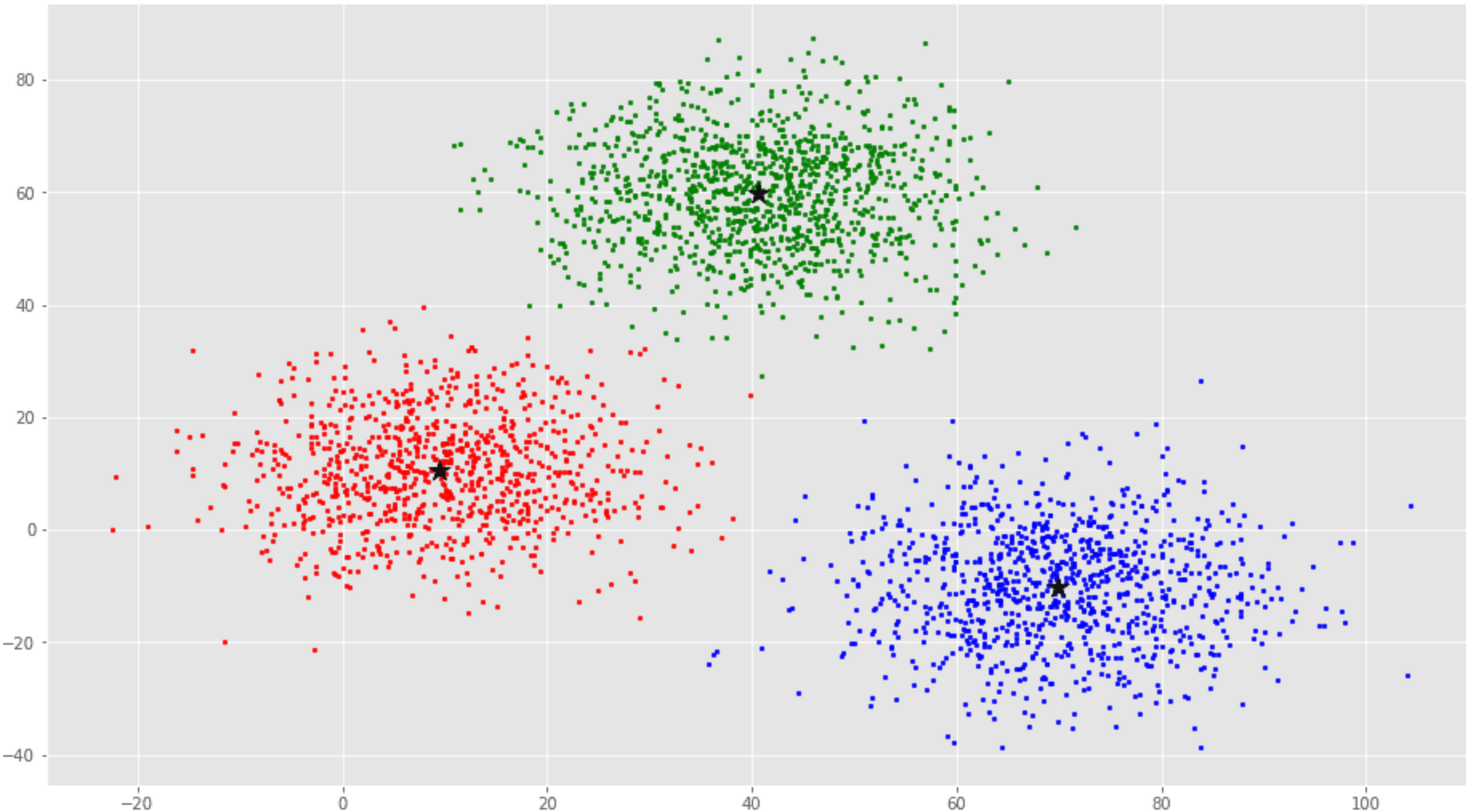
Clustering images





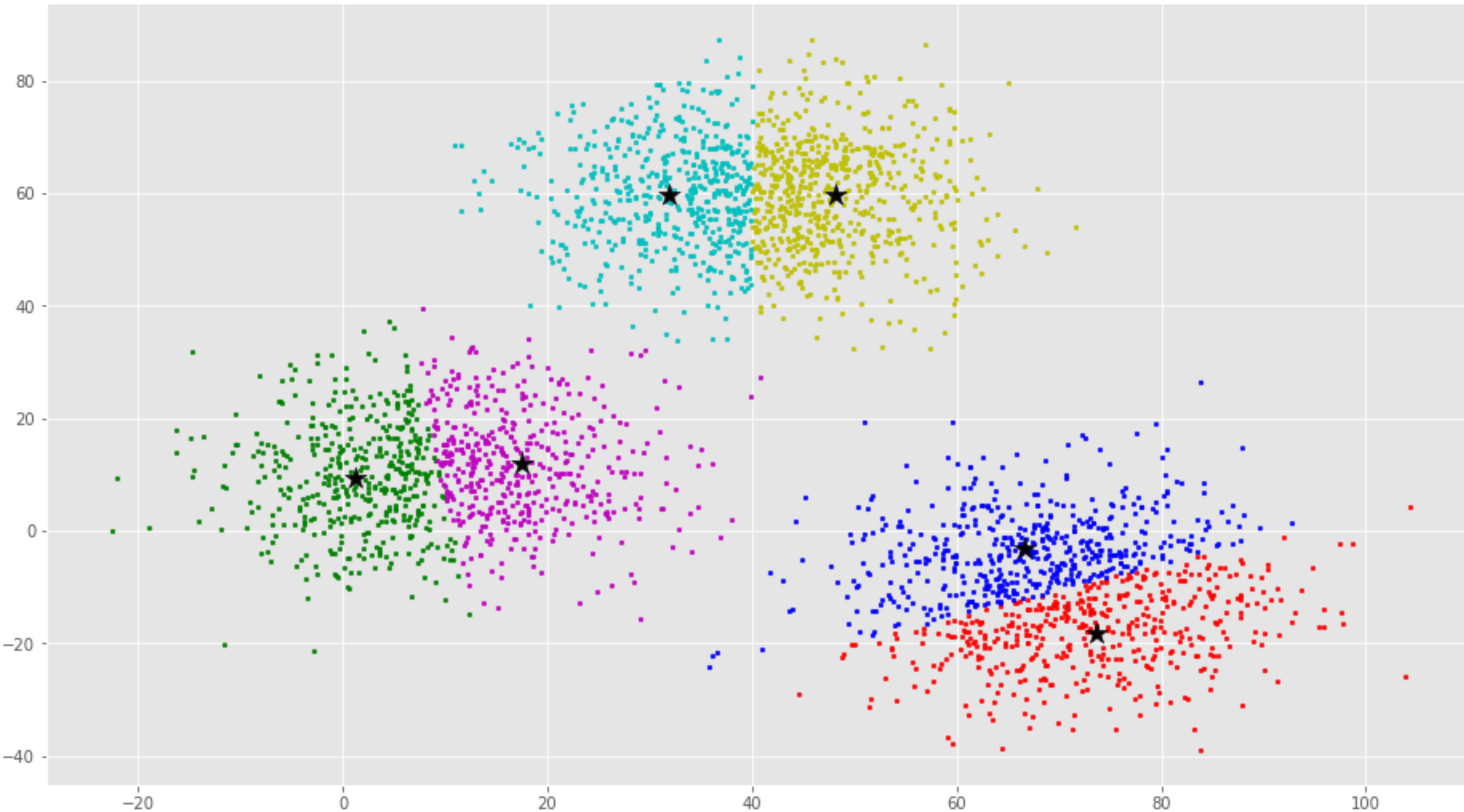
Find similar images to a target one

K-means Clustering



K=3

K-means Clustering



K=6

Unsupervised Learning

- Clustering
  - Group similar data points into clusters
  - Example: k-means, hierarchical clustering, density-based clustering
- Dimensionality reduction
  - Project the data to lower dimensional space
  - Example: PCA (Principal Component Analysis)
- Feature learning
  - Find feature representations
  - Example: Autoencoders

Supervised Learning Tasks

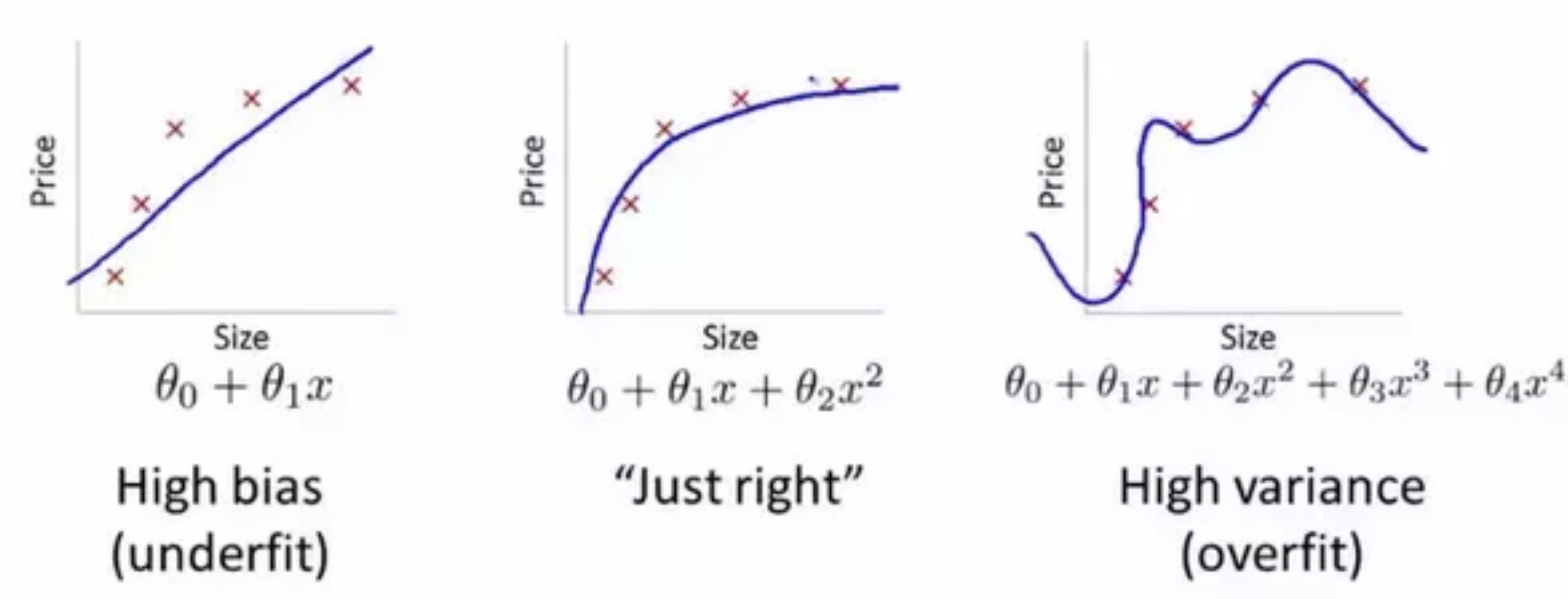
- Classification
  - Learn to predict class (discrete)
  - Minimize classification error 1/N $\sum_{i=1}^{N}[y_i \neq f(x_i)]$
- Regression
  - Learn to predict response variable (numerical)
  - Minimize MSE (Mean Square Error)
  - $1/N \sum_{i=1}^{N}[y_i - f(x_i)]^2$
- Both classification and regression
  - Training and testing phase
  - "Optimal" model is learned in training and applied in testing

Learning Challenges

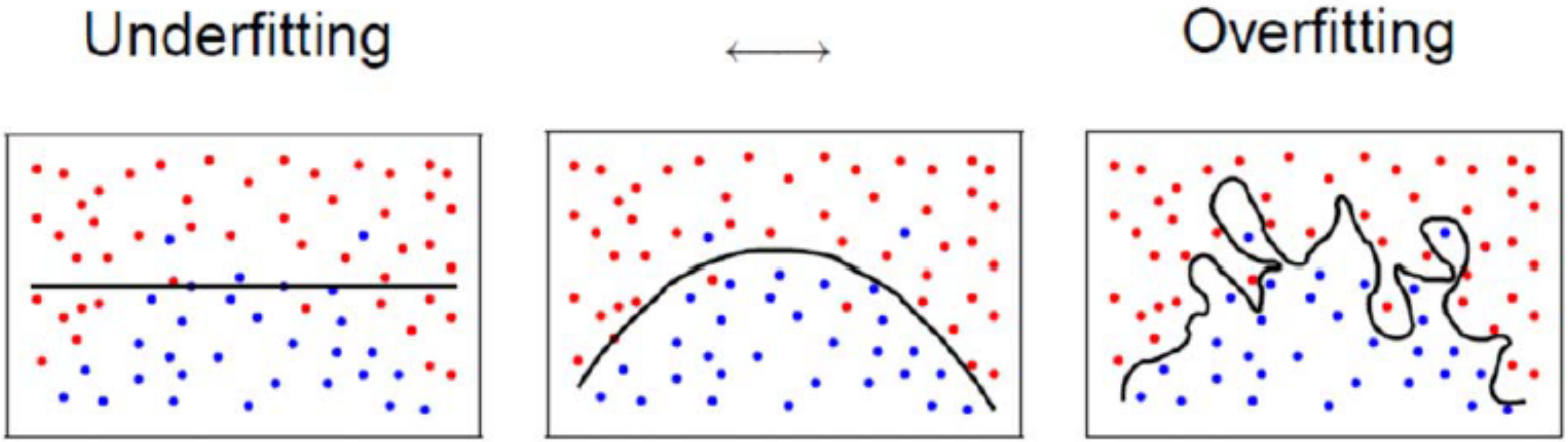
- Goal
  - Classify well new testing data
  - Model generalizes well to new testing data
  - Minimize error (MSE or classification error) in testing
- Variance
  - Amount by which model would change if we estimated it using a different training data set
  - More complex models result in higher variance
- Bias
  - Error introduced by approximating a real-life problem by a much simpler model
  - E.g., assume linear model (linear regression), then error is high
  - More complex models result in lower bias

Bias-Variance tradeoff

Example: Regression

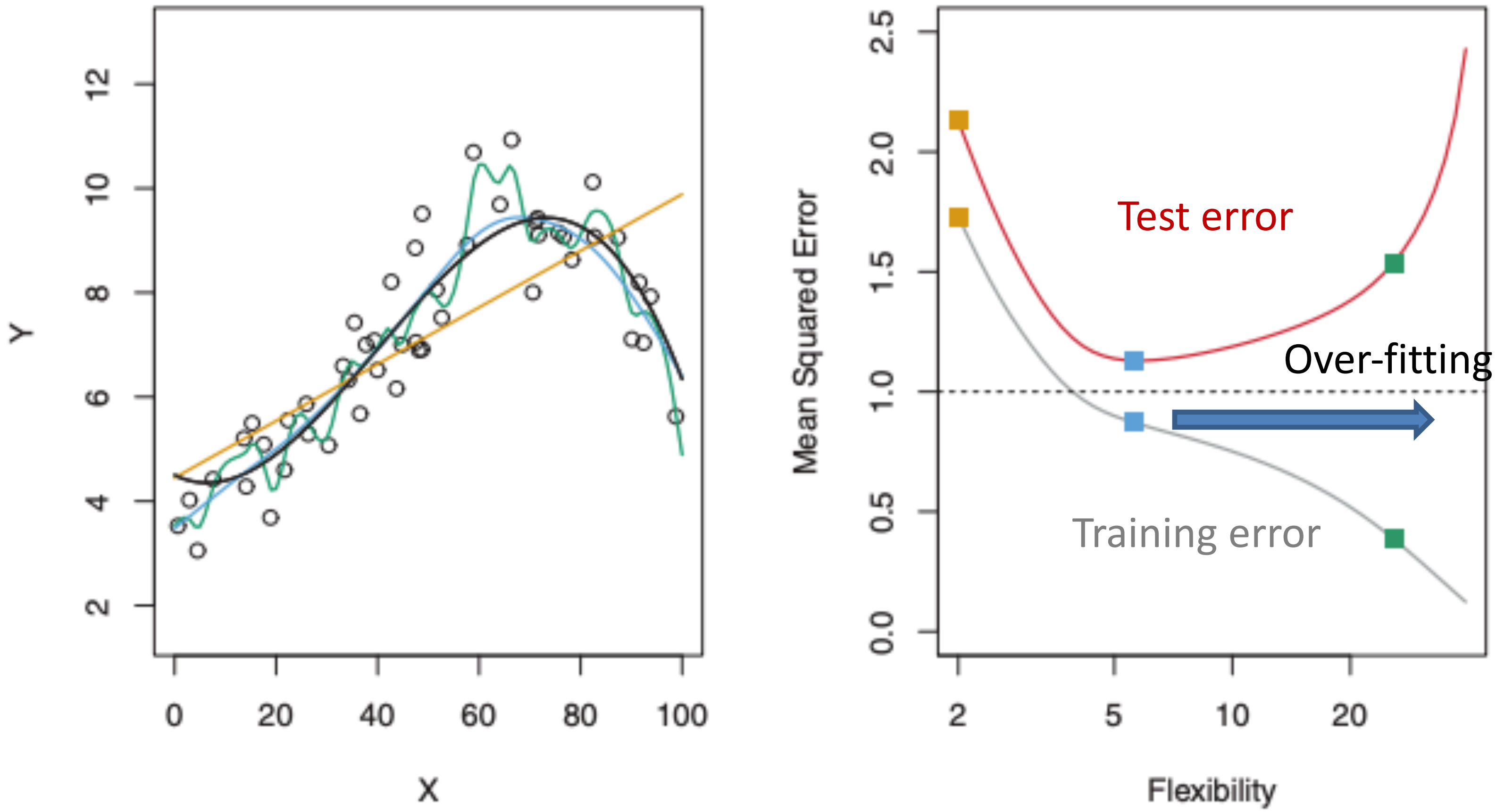


Generalization Problem in Classification



- Again, need to control the complexity of the (discriminant) function

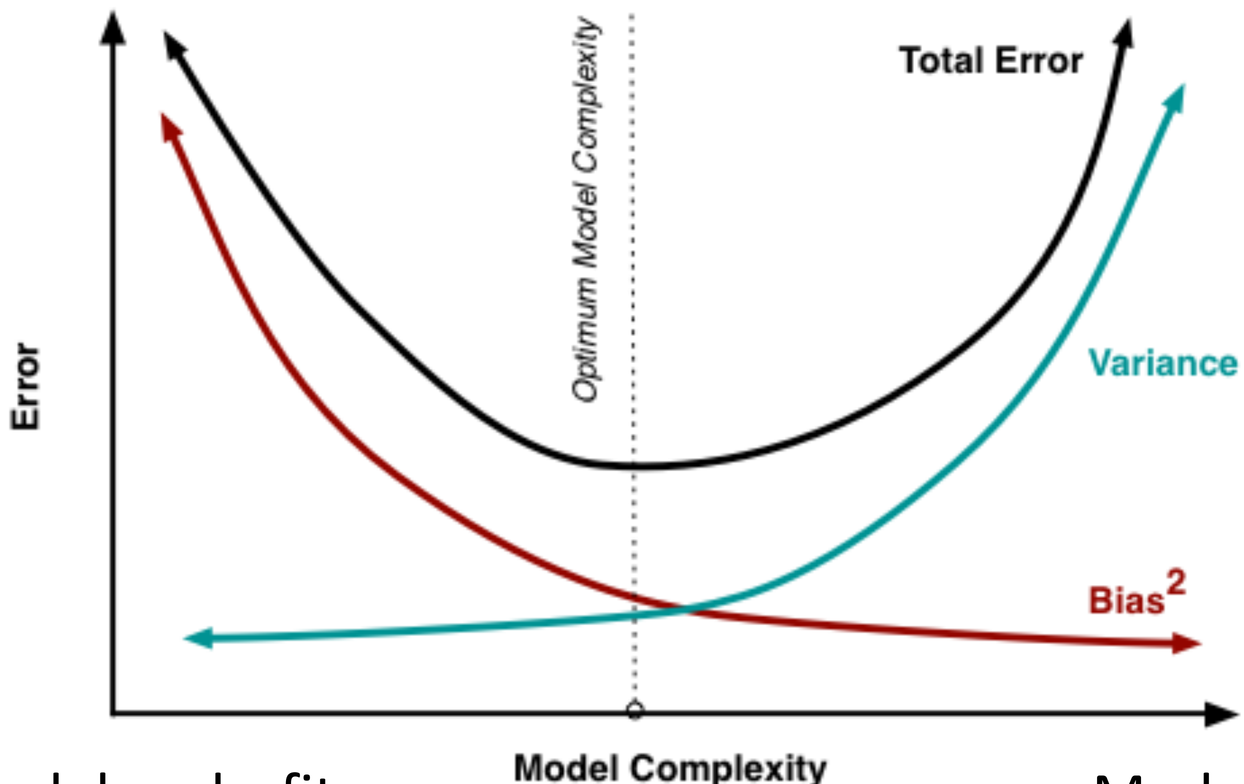
Training and testing error



ISL, Chapter 2.2.2

Bias-Variance Tradeoff

Testing



Model underfits

Model overfits

Test error is sum of bias, variance and noise

Occam's Razor

- William of Occam: Monk living in the 14^{th} century
- Principle of parsimony:

"One should not increase, beyond what is necessary, the number of entities required to explain anything"

- When many solutions are available for a given problem, we should select the simplest one

Select the simplest machine learning model that gets reasonable accuracy for the task at hand

Recap

- ML is a subset of AI designing learning algorithms
- Learning tasks are *supervised* (e.g., classification and regression) or *unsupervised* (e.g., clustering)
  - Supervised learning uses labeled training data
- Learning the “best” model is challenging
  - Design algorithm to minimize the error
  - Bias-Variance tradeoff
  - Need to generalize on new, unseen test data
  - Occam’s razor (prefer simplest model with good performance)

Probability review

Probability Resources

- Review notes from Stanford's machine learning class
- Sam Roweis's probability review
- David Blei's probability review
- Books:
  - Sheldon Ross, A First course in probability

Acknowledgements

- Slides made using resources from:
  - Andrew Ng
  - Eric Eaton
  - David Sontag
- Thanks!