DS 4400

Machine Learning and Data Mining I
Spring 2021

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

January 21 2021

Today's Outline

- Course policies
- Learning tasks
  - Supervised Learning: classification, regression
  - Unsupervised Learning
- ML terminology
- Learning challenges
  - Bias-Variance tradeoff
- Probability reviews

Course Information

- Website: www.ccs.neu.edu/home/alina/classes/Spring2021
- Canvas: https://canvas.northeastern.edu
- Gradescope: gradescope.com
- Communication: piazza.com
- E-mail:
  - a.oprea@northeastern.edu
  - gojala.o@northeastern.edu
  - malviya.p@northeastern.edu
  - parkar.s@northeastern.edu

Class Outline

- Introduction – 1 week
  - Probability and linear algebra review
- Linear regression and regularization – 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 lecture
- Adversarial ML – 1 lecture
  - Security of ML at testing and training time

Policies

- Instructors
  - Alina Oprea
  - TAs: Omkar Reddy Gojala, Prabal Malviya, Saurabh Nitin Parkar
- Schedule
  - Tue 11:45am – 1:25pm, Thu 2:50-4:30pm EST
  - Shillman Hall 320 and Zoom lectures
  - Office hours (Zoom):
    - Alina: Tuesday 4:30-5:30pm; Thursday 4:30 – 5:30 pm
    - Omkar: Monday and Wednesday 3:00-4:00pm;
    - Prabal: Monday and Thursday 12:00-1:00pm
    - Saurabh: Friday 10am-12pm
    - Links on Canvas under "Syllabus"
- Online resources
  - Slides / recordings will be posted after each lecture for 48 hours
  - Use Piazza for questions
  - Canvas as course management system

Policies, cont.

- Your responsibilities
  - Please be on time, attend classes, and take notes
  - Participate in interactive discussion in class
  - Submit assignments/ programming projects on time
- Late days for assignments
  - 5 total late days, after that loose 20% for every late day
  - Assignments are due at 11:59pm on the specified date
  - We will use Gradescope for submitting assignments
  - No need to email for late days

Grading

- Assignments – 25%
  - 4-5 assignments and programming exercises based on studied material in class
- Final project – 30%
  - 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
- Midterm Exam –20%
  - Tentative date: Tuesday, March 2
- Final Exam – 20%
  - Tentative date: Tuesday, April 6
- Class participation – 5%
  - Participate in class discussion/Zoom and on Piazza
  - Pop up quizzes

Assignments

- Several theoretical questions and many programming exercises
- Language
  - Use Python (preferred) or R
  - Jupyter notebooks recommended
- Submission
  - Submit PDF report
  - Includes all the results, as well as link to code and instructions to run it

Final project

- Goal: work on a larger data science project
  – Build your portfolio and increase your experience
- Requirements
  – Large dataset: at least 20,000 records (public source)
  – Not recommended to collect your own data
  – Pick application of interest
  – We will also a list of projects
  – Experiment with at least 4 ML models
  – Perform in-depth analysis (which features contribute mostly to prediction, which model performs best)
  – Teams of 2 students, will have a TA assigned
- Timeline
  – Proposal: mid class; milestone 3 weeks after (Instructors will provide early feedback)
  – Final presentation (10 mins) and report (~6 pages)

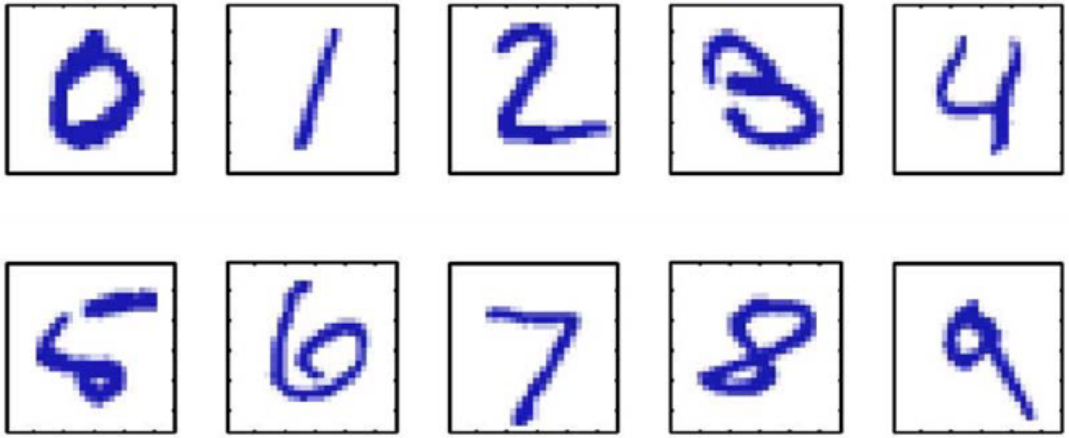
Learning Tasks

- Supervised learning
  - Classification
  - Regression
  - Examples
- Unsupervised learning
  - Clustering

Slides adapted from

- A. Zisserman, University of Oxford, UK
- S. Ullman, T. Poggio, D. Harari, D. Zysman, D Seibert, MIT
- D. Sontag, MIT
- Figures from “An Introduction to Statistical Learning”, James et al.

Example 1
Handwritten digit recognition



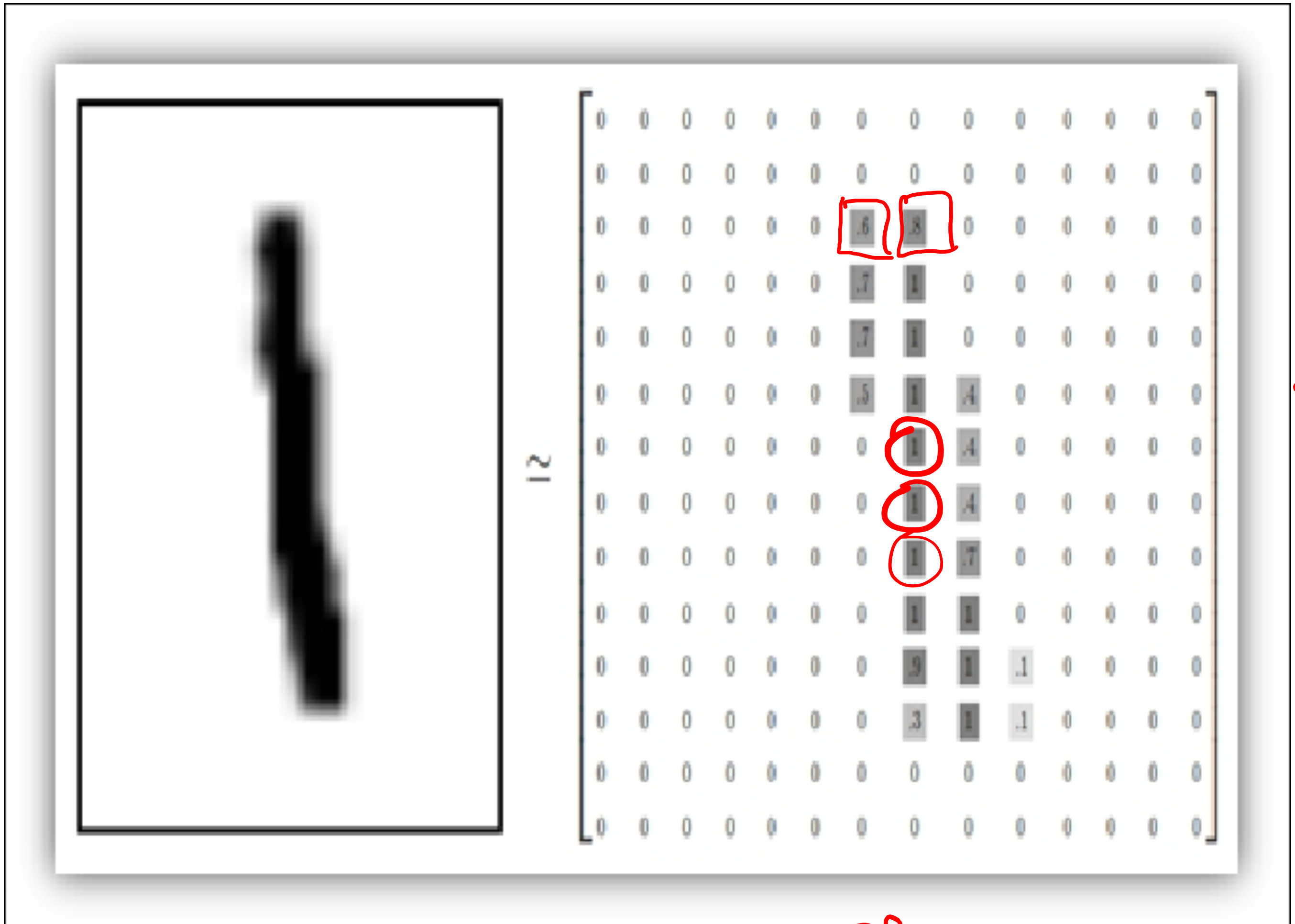
Images are 28 x 28 pixels

MATRIX OR VECTOR

MNIST dataset: Predict the digit
Multi-class classifier

Data Representation

784 size



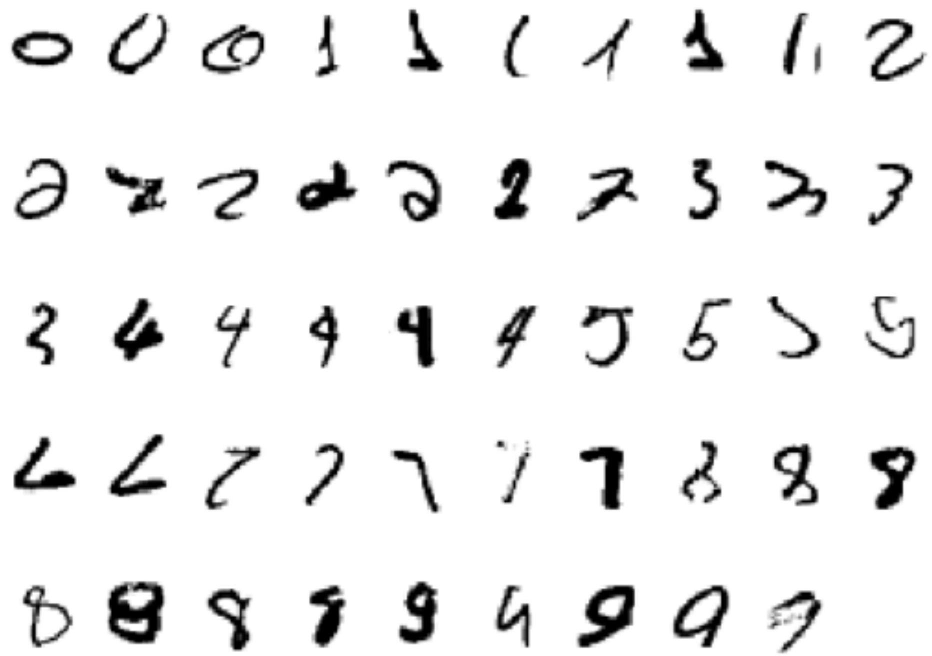
28

28

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

Data → Pre-processing → Feature extraction → ML model

Labeled

$x_i, y_i \in \{0,1\}$

Normalization

Feature Selection

Classification

$\boxed{f(x)}$

Testing

New data → ML model → Predictions $y' = f(x') \in \{0,1\}$

Unlabeled

x'

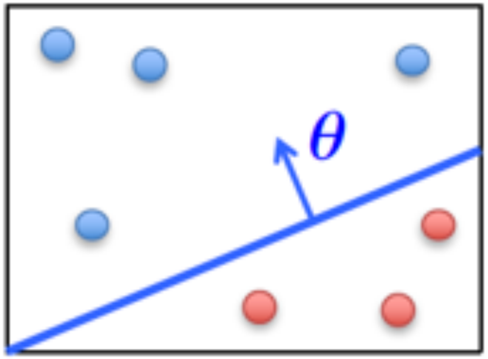
$f(x')$

Positive
Negative

Classification

Classification

- Training data
  - $x_i = [x_{i,1}, \ldots x_{i,d}]$: vector of image pixels (features)
  - Size $d = 28\text{x}28 = 784$
  - y_i: image label
- Models (hypothesis)
  - Example: Linear model (parametric mod
    - $\boxed{f(x) = wx + b}$ w, b = parameters
  - Classify 1 if $f(x) > \mathrm{T}$; 0 otherwise
- Classification algorithm
  - Training: Learn model parameters w, b to minimize objective
  - Output: "optimal" model
- Testing
  - Apply learned model to new data and generate prediction $f(x)$



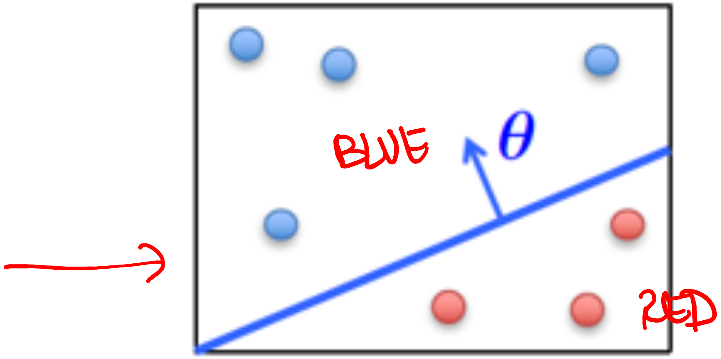
Objectives

- What are we trying to optimize?

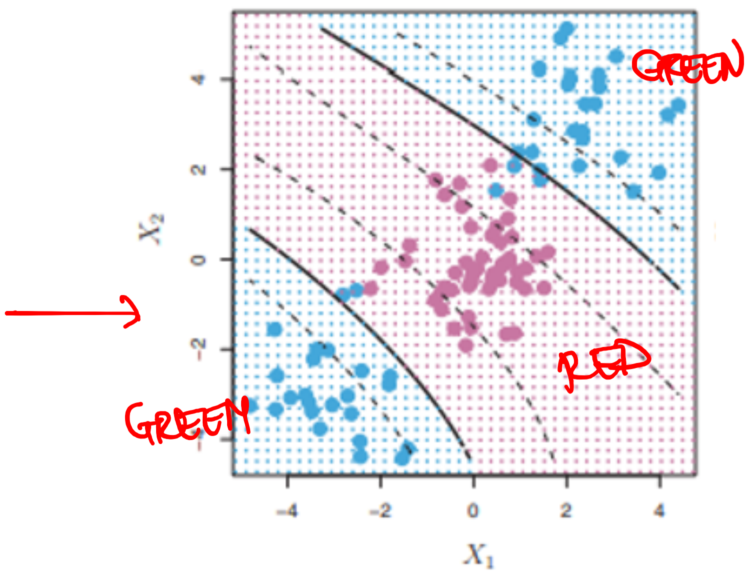
Max accuracy

Min error

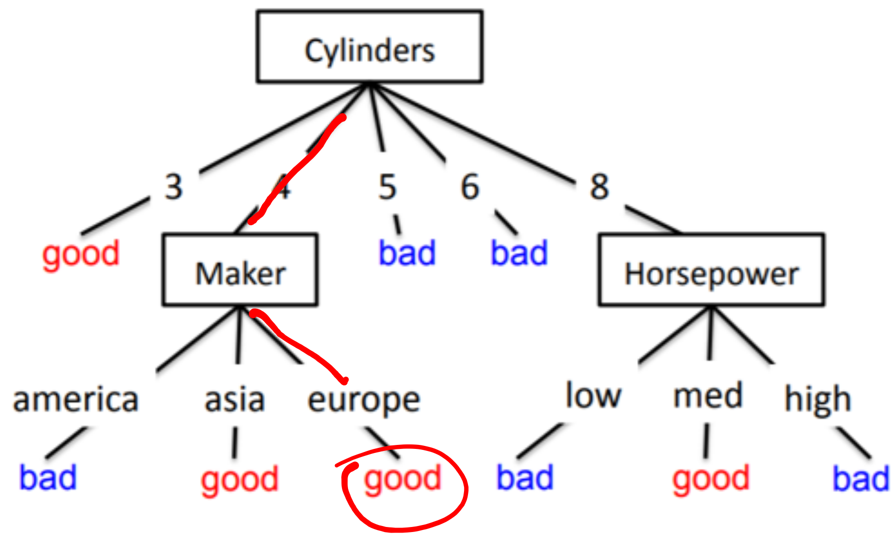
Example Classifiers



Linear classifiers: logistic regression, SVM, LDA



SVM polynomial kernel



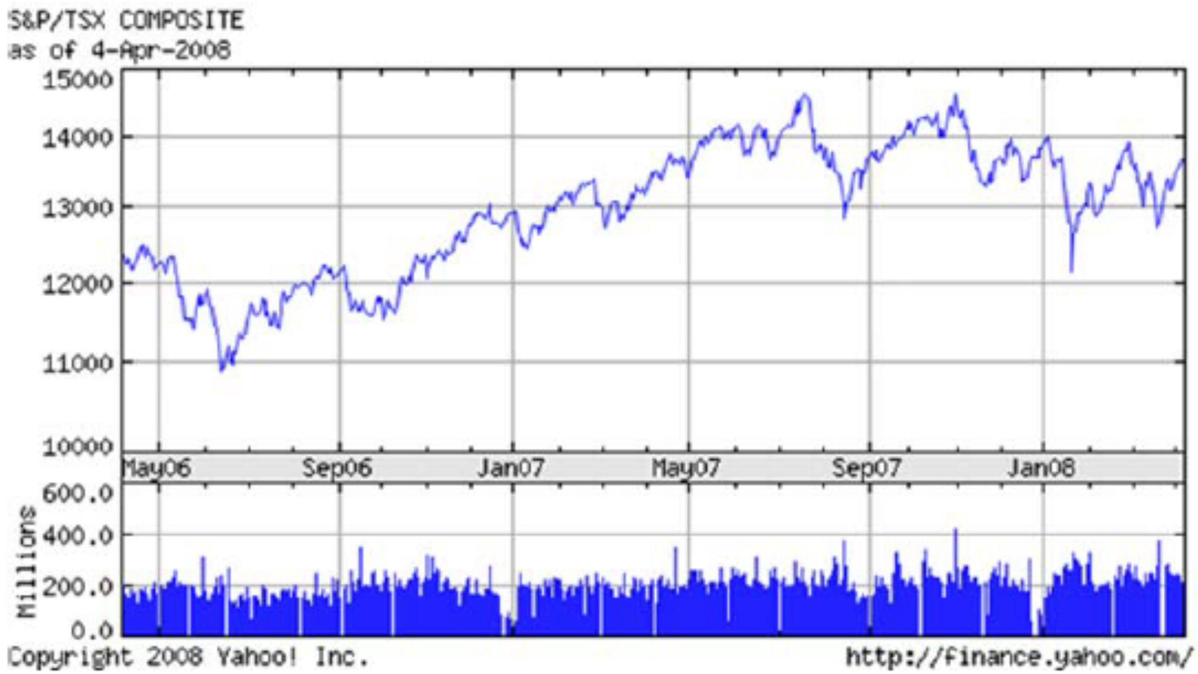
Decision trees

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

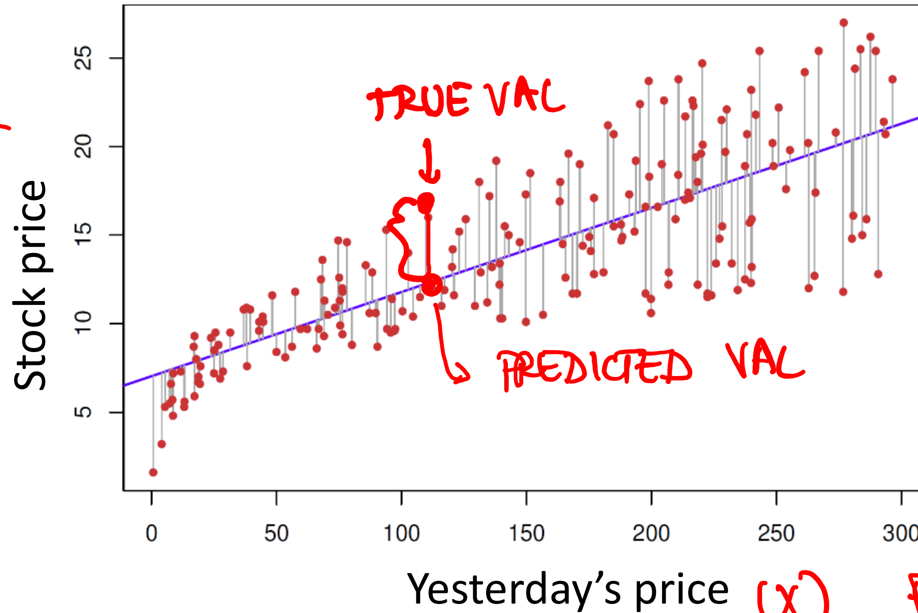


- Task is to predict stock price at future date
- This is a regression task, as the output is continuous

Regression

RESPONSE VAR

y ←



(x) FEATURE

Linear regression
1 dimension

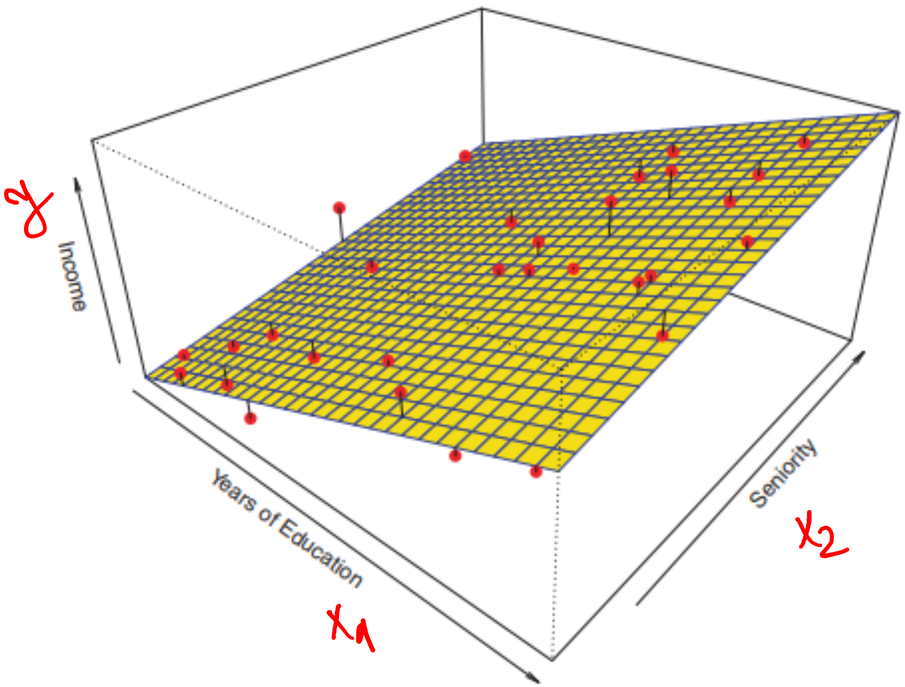
- Suppose we are given a training set of N observations

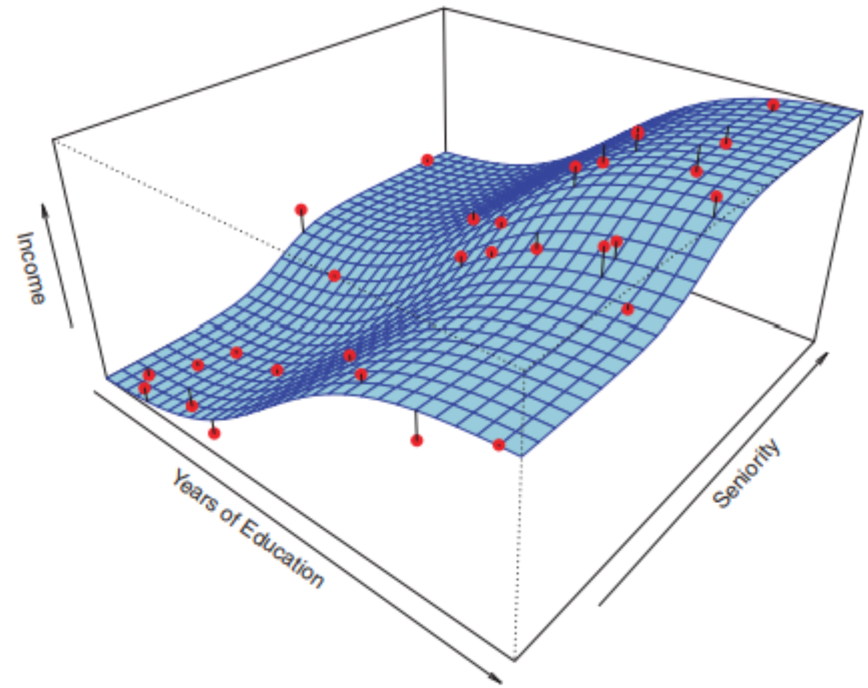
$$(x_1, \ldots, x_N) \text{ and } (y_1, \ldots, y_N)$$

- 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

Objectives

- What are we trying to optimize?

MSE

R^2

Supervised Learning: Regression

Training

Data → Pre-processing → Feature extraction → ML model

- **Data**: Labeled; $x_i, y_i \in R$
- **Pre-processing**: Normalization
- **Feature extraction**: Feature Selection
- **ML model**: Regression; $f(x) \rightarrow R$

Testing

New data → ML model → Predictions

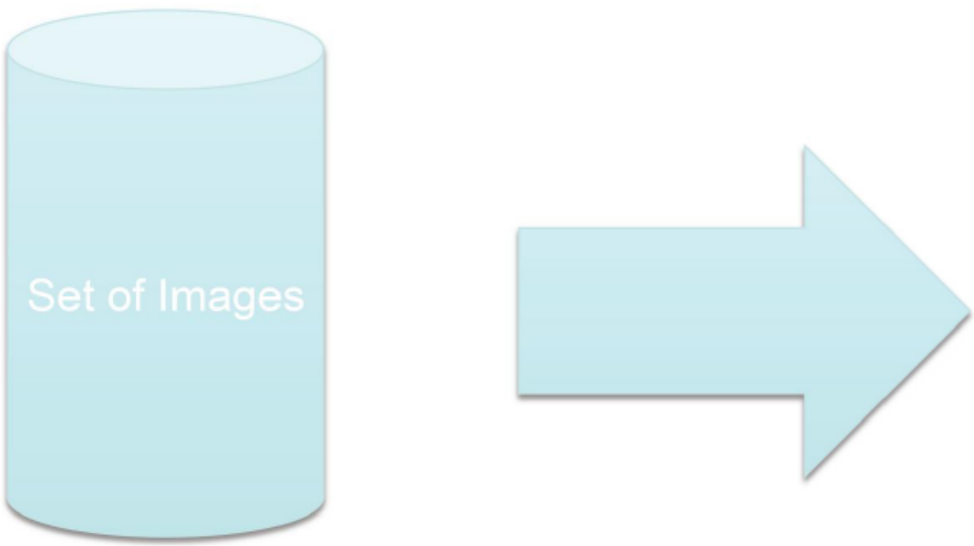
- **New data**: Unlabeled; x'
- **ML model**: $f(x')$
- **Predictions**: $y' = f(x') \in R$

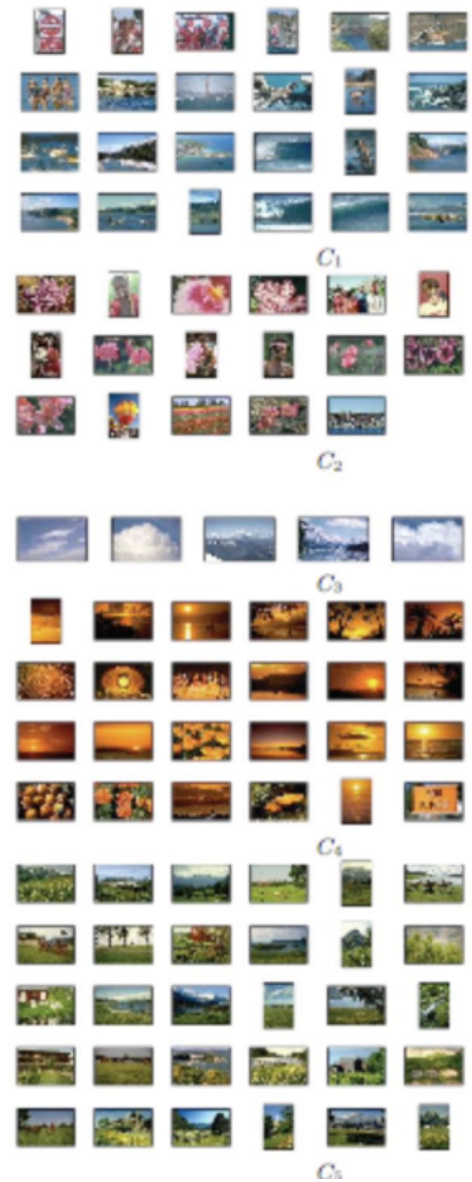
Response variable

Regression

Example 3: image search

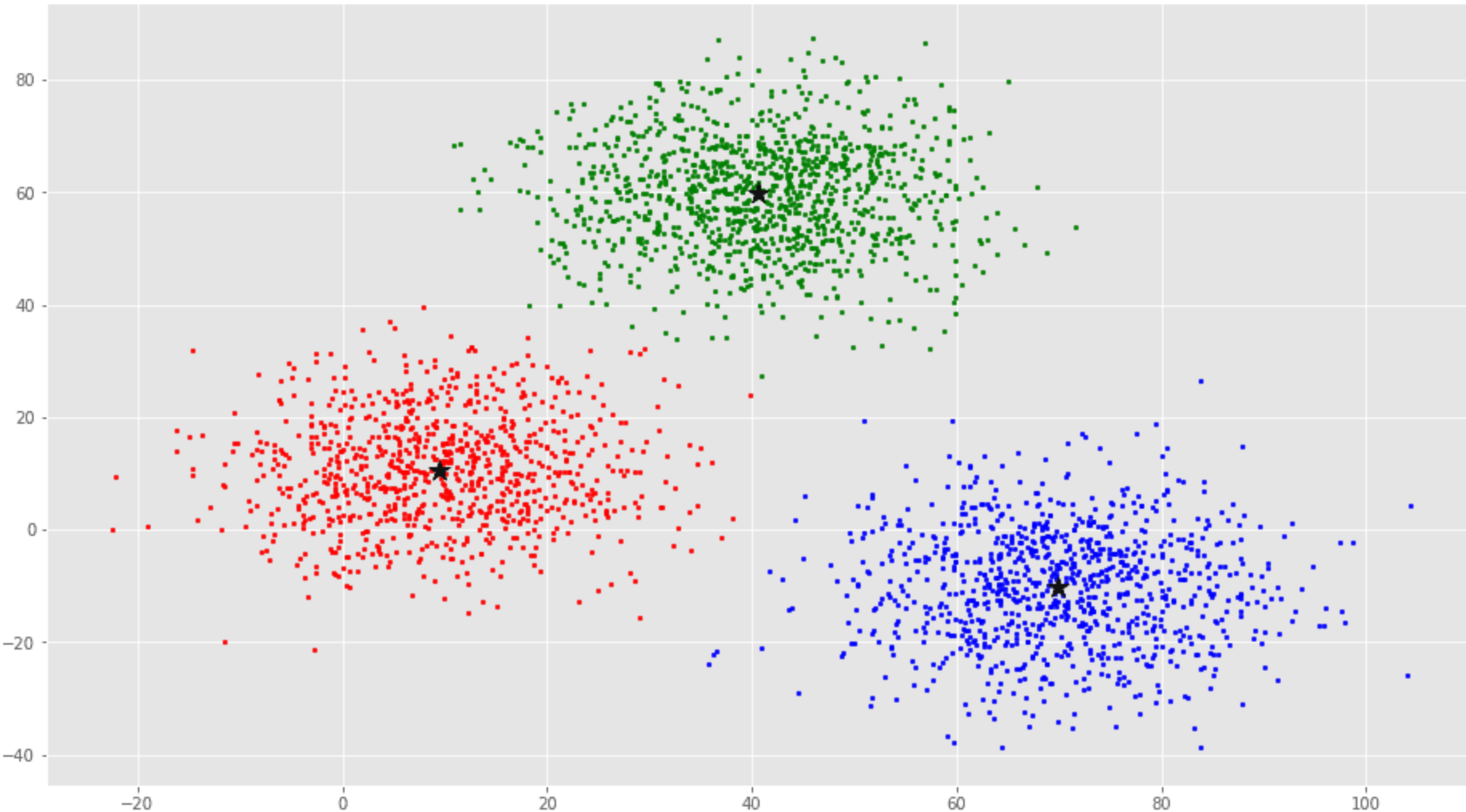
Clustering images





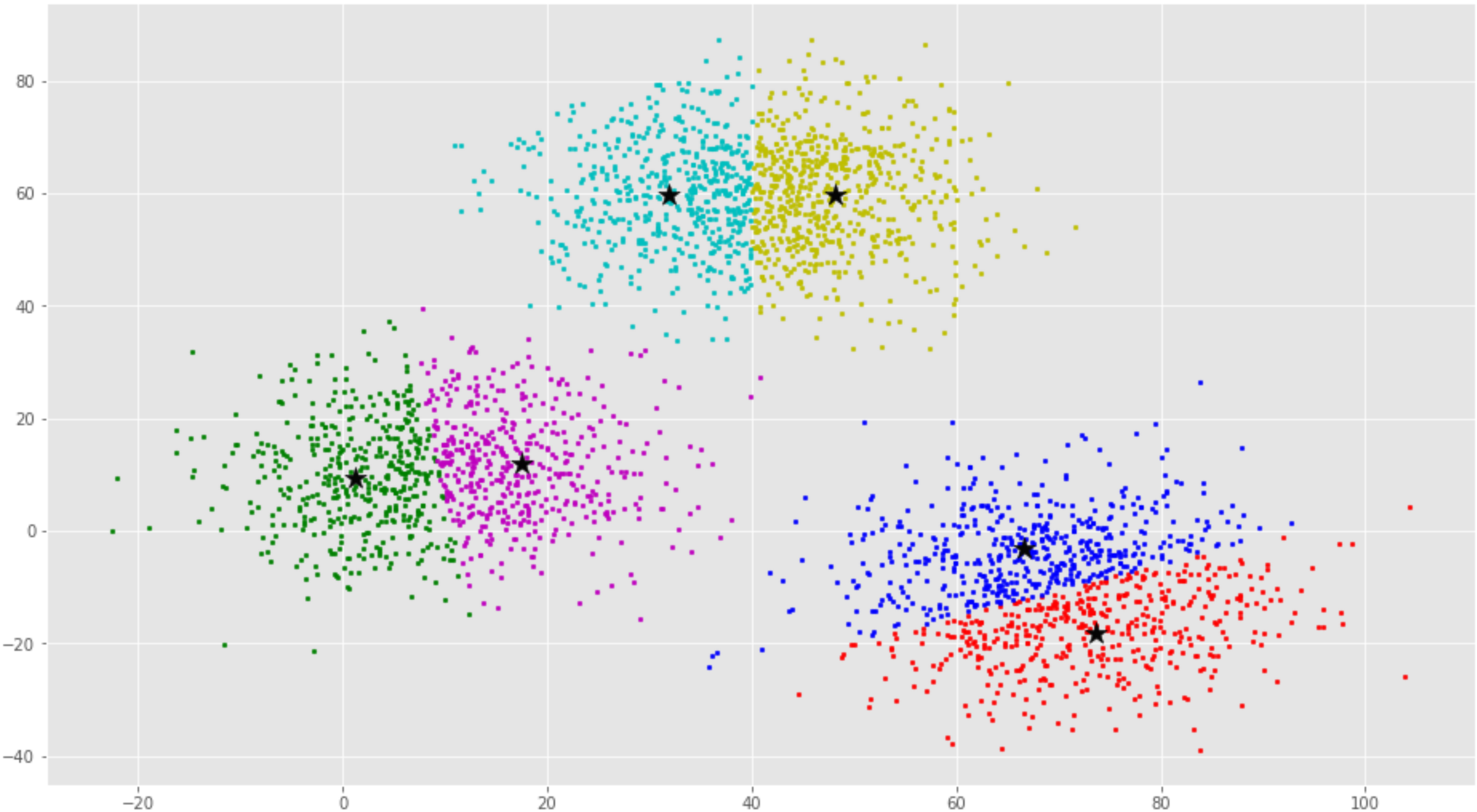
Find similar images to a target one

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), UMAP
- Feature learning
  - Find feature representations
  - Example: Autoencoders

Supervised Learning Tasks

- Classification
  - Learn to predict class (discrete)
  - Minimize classification error
- Regression
  - Learn to predict response variable (numerical)
  - Minimize MSE (Mean Square Error)
- Both classification and regression
  - Training and testing phase
  - "Optimal" model is learned in training and applied in testing

Learning Challenges

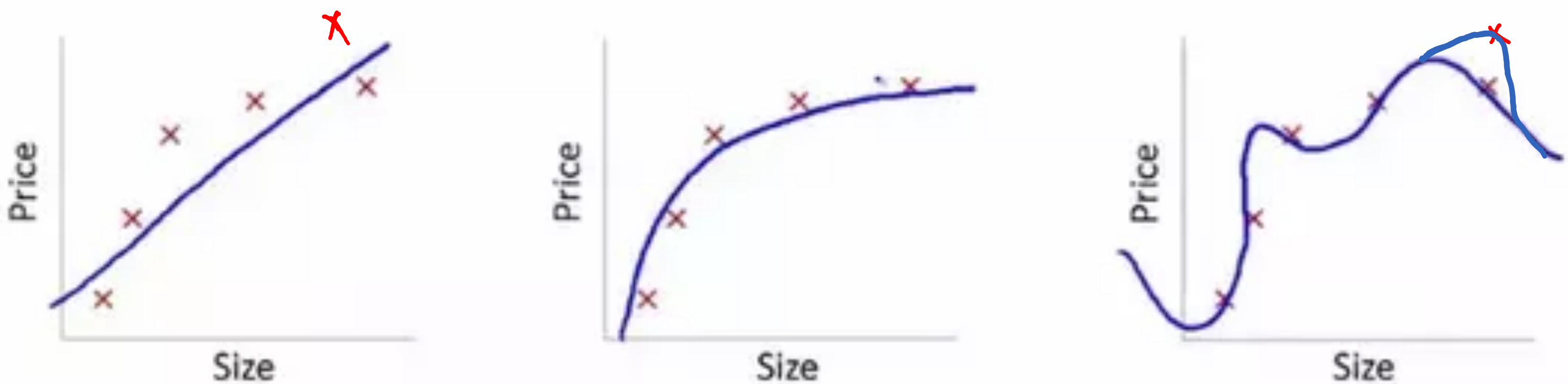
- Chapters 2.2.1 and 2.2.2 from ISL book
- Goal
  - Classify well new testing data
  - Model generalizes well to new testing data
  - Minimize error (MSE or classification error) in testing
- Variance
  - Amount by which model would change if we estimated it using a different training data set
- Bias
  - Error introduced by approximating a real-life problem by a much simpler model
  - E.g., for linear models (linear regression) bias is high

Bias-Variance tradeoff

Example: Regression

UNDERFIT

OVERFITTING



$\theta_0 + \theta_1 x$

$\theta_0 + \theta_1 x + \theta_2 x^2$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

HIGH BIAS

"BEST"

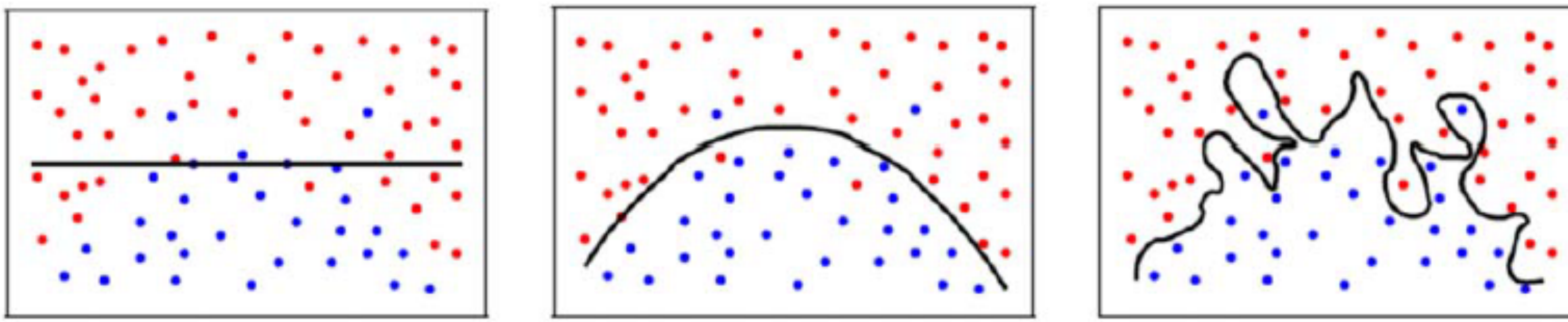
HIGH VAR

Model complexity

Generalization Problem in Classification

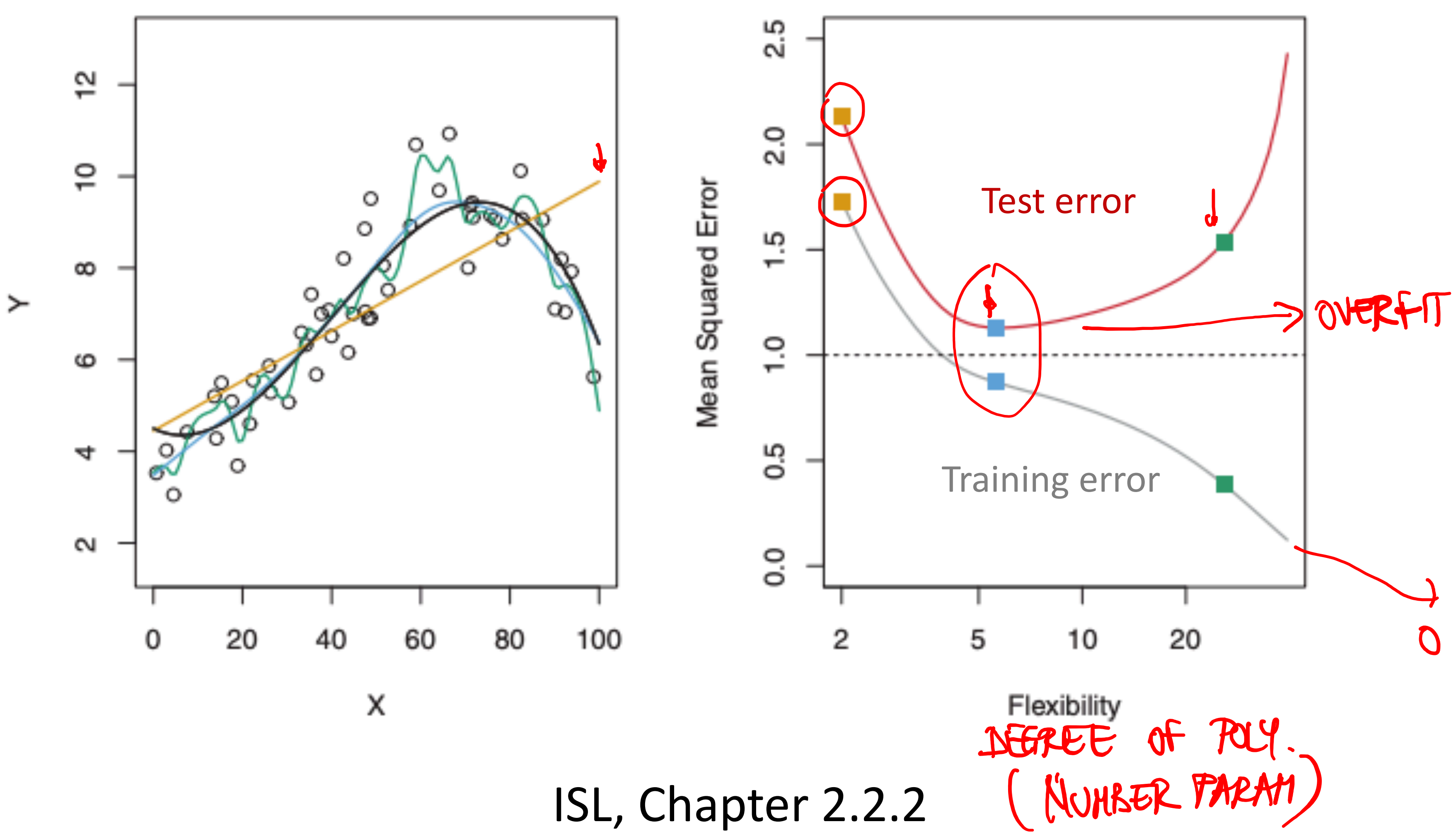
HIGH BIAS
UNDERFIT

HIGH VAR
OVERFIT



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

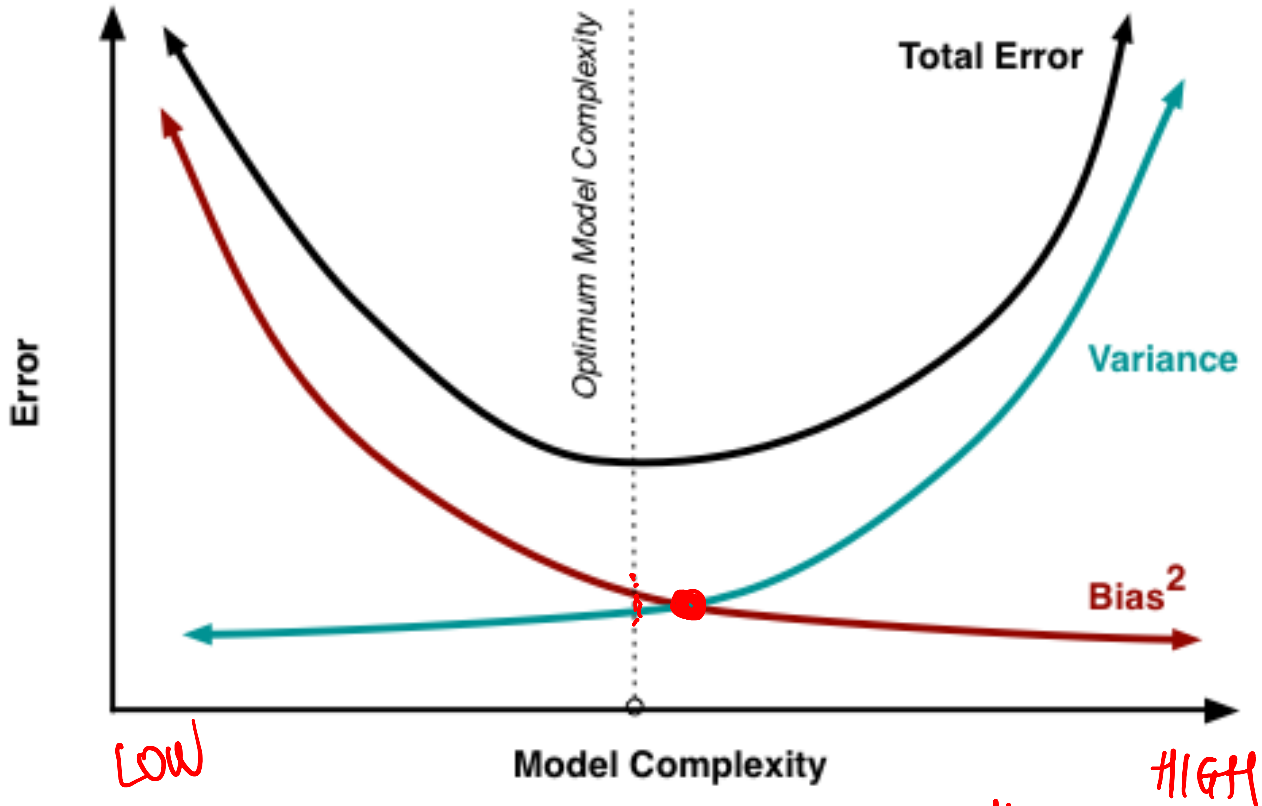
Training and testing error



ISL, Chapter 2.2.2

Bias-Variance Tradeoff

Testing



TEST ERROR = BIAS + VARIANCE + "NOISE"

Occam's Razor

- William of Occam: Monk living in the 14th 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 in testing
  - Minimize training error is not the best strategy
  - Bias-Variance tradeoff
  - Need to generalize on new, unseen test data
  - Occam's razor (prefer simplest model with good performance)