STAT 587: Data Science I

Winter 2026

Dorcas Ofori-Boateng (PhD)

Data Science - Introduction

- **Data Science** blends statistics, machine learning, and programming.
- Objective: Extract insights and build **predictive** or **explanatory** models.
- **Core workflow**:
  1. Data collection & preprocessing
  2. Exploratory analysis & visualization
  3. Modeling & validation
  4. Deployment & communication
- **Common tools**: Python, R, Jupyter, SQL, Cloud platforms.
- **Applications** extend across virtually every sector: *healthcare, finance, business, scientific research, & AI.*

Statistical learning vs. Machine learning

- Machine learning arose as a subfield of Artificial Intelligence.
- Statistical learning arose as a subfield of Statistics.
- *There is much overlap* — both fields focus on supervised and unsupervised problems:
  - Machine learning has a greater emphasis on *large scale* applications and *prediction accuracy*.
  - Statistical learning emphasizes *models* and their interpretability, and *precision* and *uncertainty*.
- But the distinction has become more and more blurred, and there is a great deal of "cross-fertilization".
- Machine learning has the upper hand in *Marketing!*

Source: ISL authors

Example 1: Prostate cancer

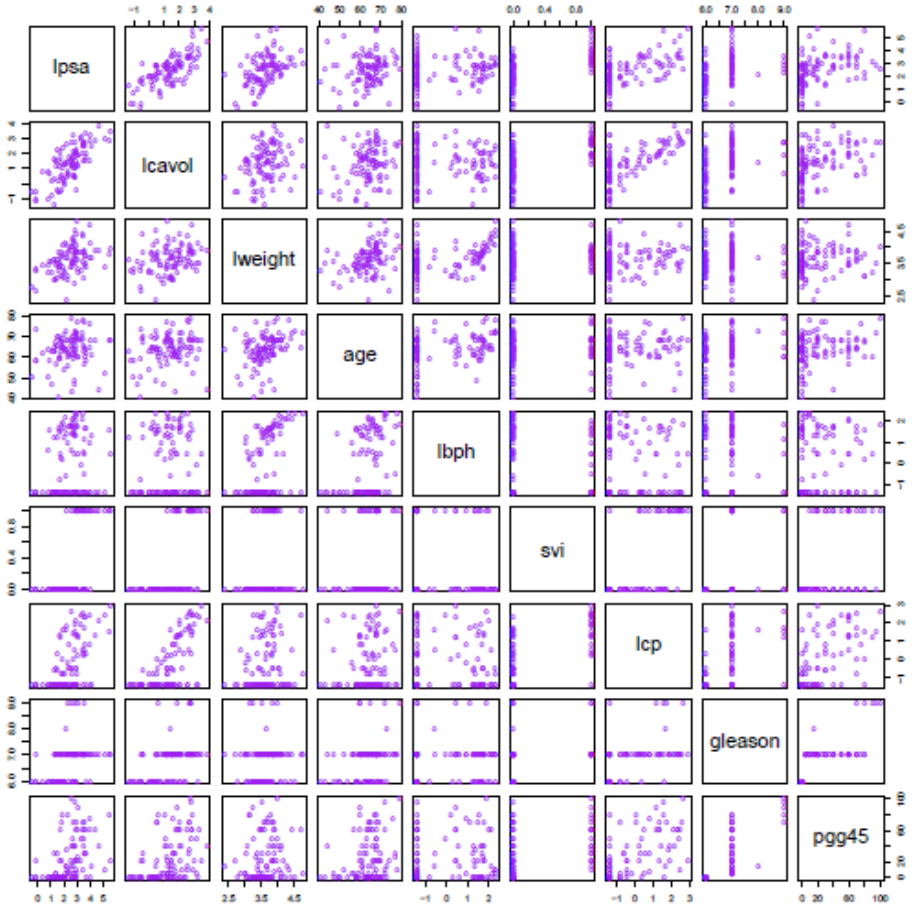

FIGURE 1.1. *Scatterplot matrix of the prostate cancer data. The first row shows the response against each of the predictors in turn. Two of the predictors,* `svi` *and* `gleason`*, are categorical.*

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 1

Example 2: Handwritten digit recognition

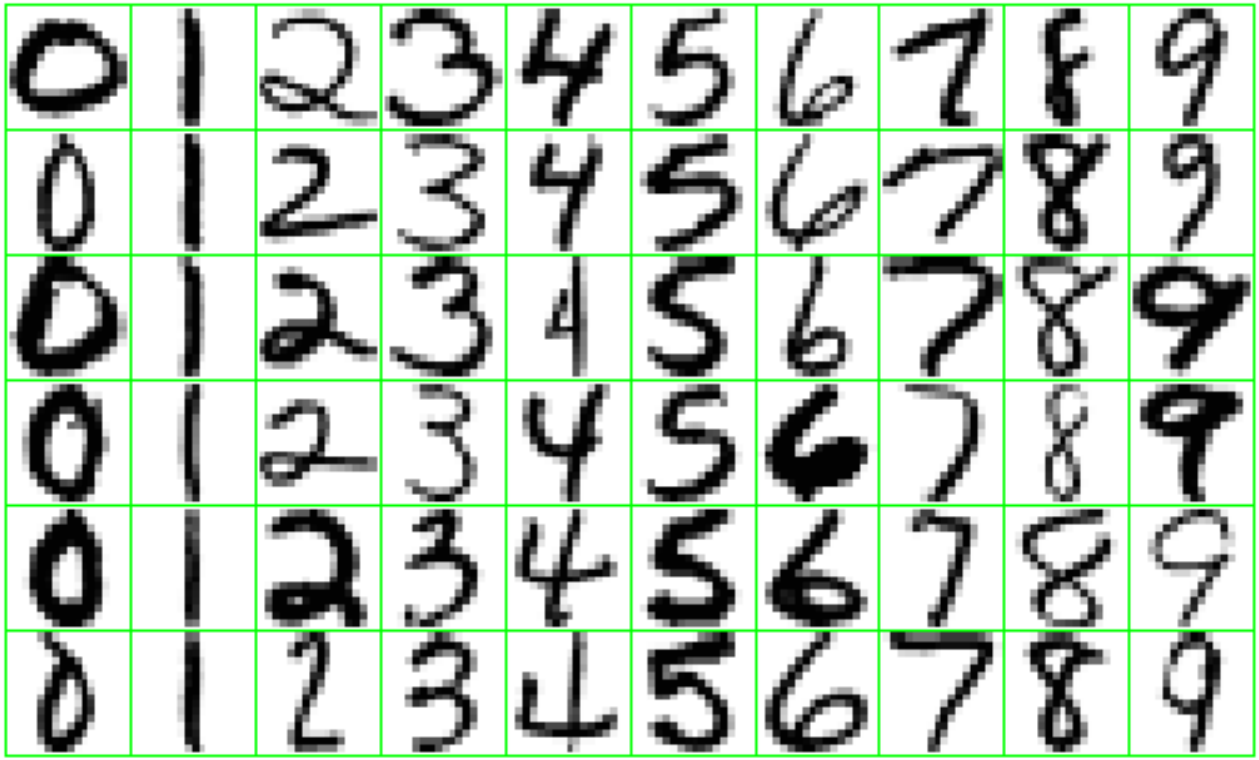

FIGURE 1.2. *Examples of handwritten digits from U.S. postal envelopes.*

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 1

Example 3: DNA expression microarrays

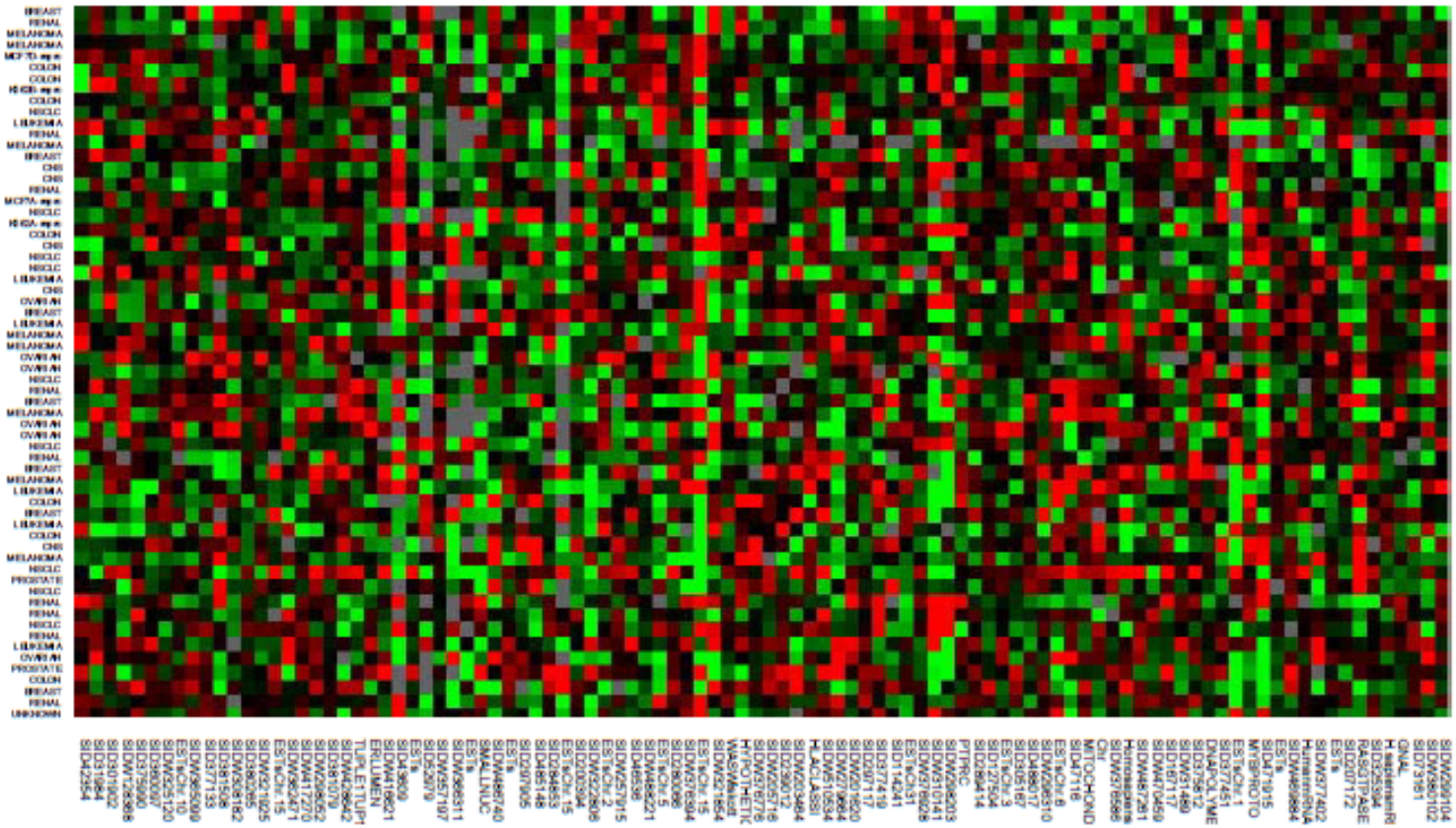

FIGURE 1.3. *DNA microarray data: expression matrix of* 6830 *genes (columns) and* 64 *samples (rows), for the human tumor data. Only a random sample of* 100 *rows are shown. The display is a heat map, ranging from bright green (negative, under expressed) to bright red (positive, over expressed). Missing values are gray. The rows and columns are displayed in a randomly chosen order.*

Unsupervised vs. Supervised

Unsupervised learning: Have *feature* but *no response* variable; no modeling for prediction or inference.

- Grouping of variables that behave similarly; Dimensionality reduction.
- Difficult to assess performance of the method as there is no response.
- Useful as a pre-processing step for supervised learning.

Supervised learning: Have both *feature* and *response* variable; involves modeling for prediction or inference.

Notation

- Y — a *response* (or output or dependent) variable — *quantitative* (or continuous) or *qualitative* (or categorical)
- $X_1, \ldots, X_p$ — *predictors* (or variables or covariates or features or explanatory or independent variables) — some may be quantitative and others categorical
- X — $(X_1, \ldots, X_p)$
- p — number of predictors
- n — number of subjects (or observations)
- i — subject index $(i = 1, \ldots, n)$
- j — variable index $(j = 1, \ldots, p)$
- Y_i — value of Y for subject i
- X_{ij} — value of X_j for subject i , $X_i = (X_{i1}, \ldots, X_{ip})$
- **Data**: $(Y_i, X_i), i = 1, \ldots, n.$

Why Supervised learning?

- Many data tasks aim to **predict** an outcome from features.
- We *learn* a **mapping** from examples with known answers (labels).
- Two (2) core problem families:
  - **Regression**: numeric response
  - **Classification**: categorical response

- Observations: (X_i, Y_i) for $i = 1, \ldots, n$
- Predictors: $X_i \in \mathbb{R}^p$ (features)
- Response: Y_i (numeric or class label)
- **Relationship**: $Y = f(X) + \epsilon$; where ϵ is known as the random error.

The statistical learning view

- **Goal**: Learn $f(\cdot)$ so that **new** (X_{new}) is mapped to a good prediction $\hat{Y}$.
- Learning = estimating a function that generalizes beyond the training set.
- Key tension: fit the data closely vs. remain stable on new data.
- Two simple prediction approaches: *Least Squares* (*Linear models*) and *Nearest Neighbors*.

Nearest-Neighbor Methods

With Classification:

Given a **prediction point** x_0:

Step 1 Pick a positive integer K

Step 2 Identify the set of K points in the training data that are **closest** to x_0. This set — represented by $\mathcal{N}_0$ — contains the K nearest neighbors of x_0.

Step 3 For each class c, estimate the **conditional probability** of the class as

$$\hat{P}(Y = c|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = c),$$

i.e., the fraction of points in $\mathcal{N}_0$ whose response equals c

Step 4 Classify x_0 to the class with the largest probability

Nearest-Neighbor Methods

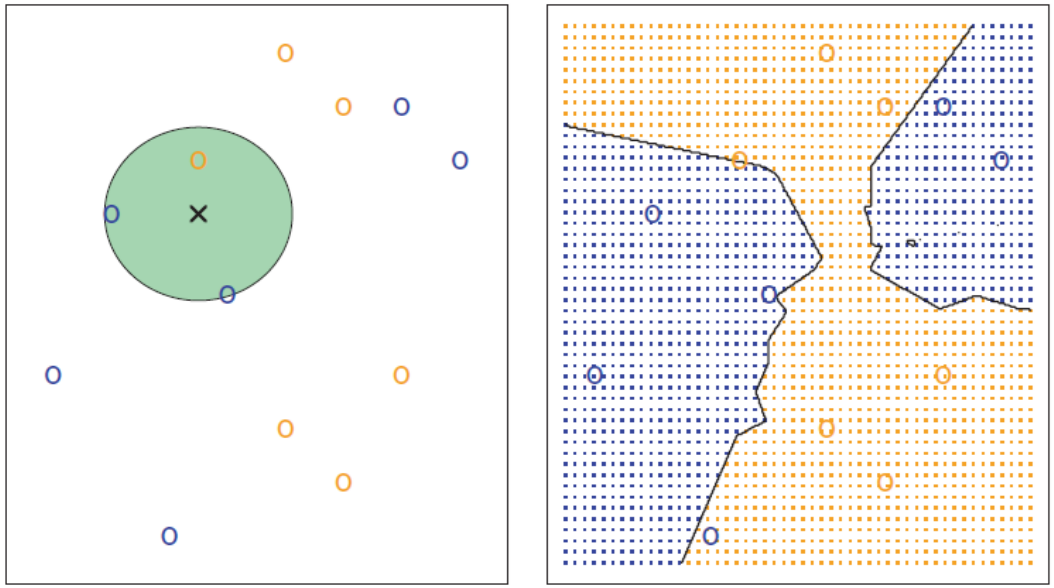

FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

Source: ISL

Nearest-Neighbor Methods

With Regression:

Given a positive integer K and a prediction point x_0, first identify the set $\mathcal{N}_0$ of K points in the training data that are closest to x_0, i.e., the K nearest neighbors of x_0. Next, estimate the mean $f(x_0)$ as the average response in $\mathcal{N}_0$, i.e.,

$$\hat{f}(x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} y_i,$$

- A **nonparametric** method. Also a **local** method
- Like KNN classifier, K controls flexibility which affects the bias-variance tradeoff. As K increases, flexibility decreases, implying that bias increases and variance decreases
- With $K = 1$, assuming no ties, we have $Y_i = \hat{Y}_i$
- **Pros:** simple, does not require any *a priori* assumptions about shape of f
- **Cons:** No associated tests, does not work well when p is large or n is not large

Model Selection

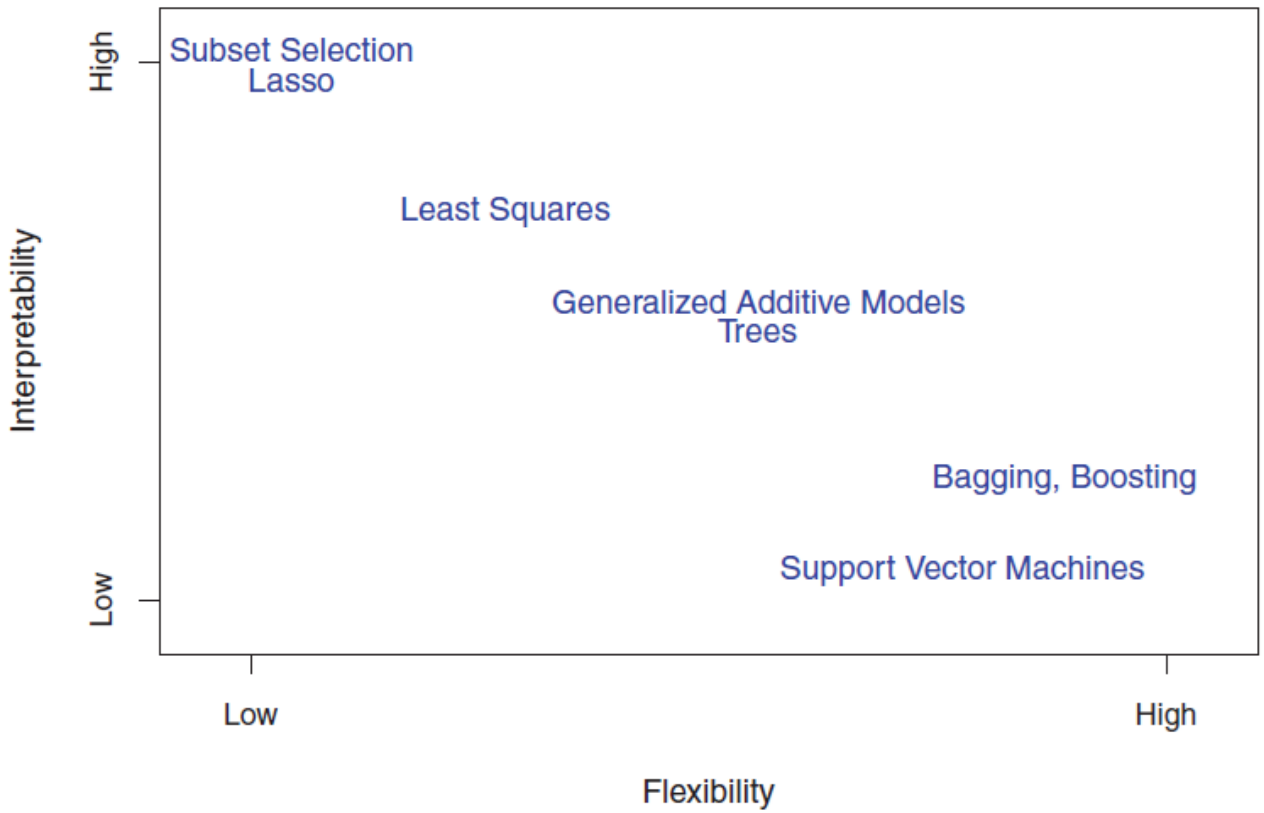

FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Loss functions: What are we minimizing?

- A **loss** quantifies penalty for predicting $\hat{Y}$ when the true value is Y
- Common choices (conceptually):
  - Squared error (regression)
  - Zero–one loss (classification)
- Learning = choosing f to minimize expected loss (risk)

Bias-Variance Tradeoff

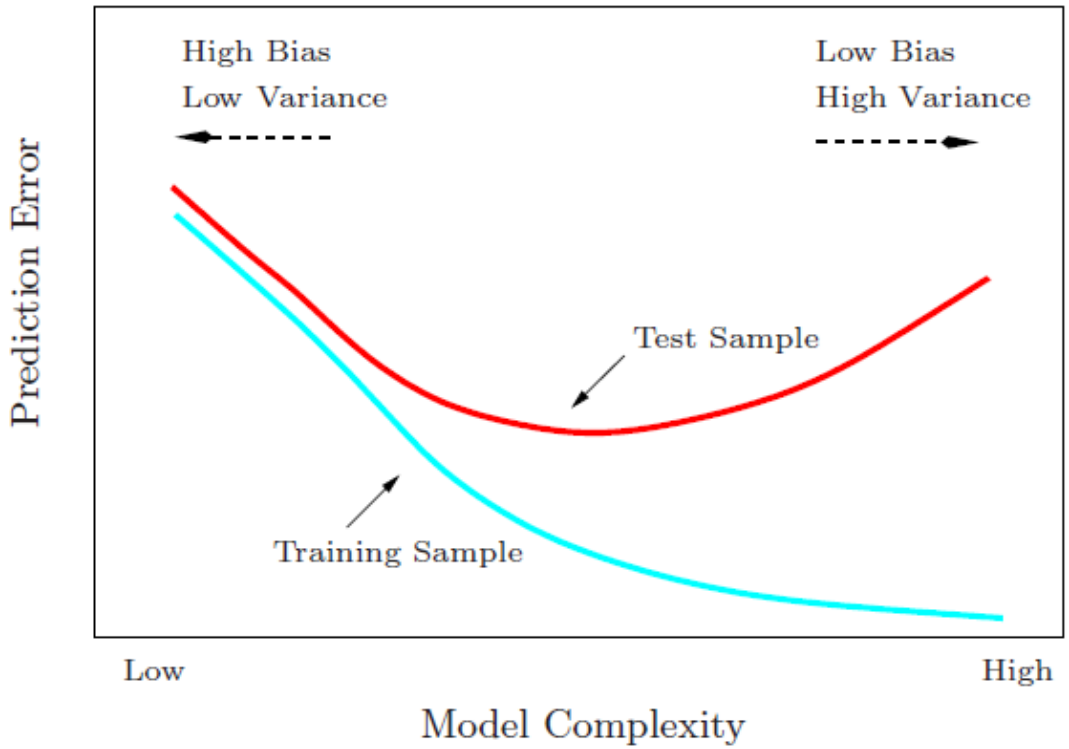

FIGURE 2.11. *Test and training error as a function of model complexity.*

Q: Why do we observe a U shape for test MSE?

$$E(\hat{Y}_0 - Y_0)^2 = (\text{Bias}\{\hat{f}(x_0)\})^2 + \text{var}\{\hat{f}(x_0)\} + \sigma^2$$

A practical supervised learning workflow

1. Define the prediction task (regression or classification).
2. Choose features and representation (cleaning, encoding, scaling).
3. Select a model family (complexity level).
4. Fit on training data (empirical risk minimization).
5. Estimate generalization (validation/test) and iterate

Python Basics

Introduction to Python

- Basic commands
- Numerical Python
- Graphics
- Sequences and Slicing Data
- Indexing Data
- Loading Data
- For Loops
- Additional Graphical & Numerical Summaries