Lecture 1: What is Data (Science)?

Course: Mathematical Introduction to Data Science

Instructor: Abhishek Chaudhary

October 14, 2025

Outline

Introduction: What is Data?

We start by explaining what we will understand by the term data throughout this course. Simultaneously, we issue a warning that the exact definition may slightly change from lecture to lecture, as different contexts may require subtle adjustments.

For example, data in one lecture might be numbers in a table, while in another, it could be images or text, each requiring a slightly different approach. This flexibility allows us to adapt our methods to the nature of the data, ensuring that our analyses are appropriate and effective for the specific problem at hand. As we progress, you'll see how these adjustments help in applying mathematical concepts more precisely to real-world scenarios.

What is Data Science?

Data Science is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured (*e.g.*, spreadsheets) and unstructured data (*e.g.*, social media posts). It combines elements from mathematics, statistics, computer science, and domain-specific knowledge to analyze and interpret complex data.

Think of data science as a toolbox: math provides rigor, statistics handles uncertainty, computer science enables computation, and domain knowledge ensures relevance. By integrating these areas, data science allows us to turn raw data into actionable insights that can drive decision-making in various fields, from business to healthcare. This holistic approach is what makes data science so powerful and versatile.

Main Goals of This Lecture

1. Define what constitutes "data" in various forms.
2. Introduce key concepts like datasets, features, labels, and their structures.
3. Explore examples that illustrate how data is modeled and used in real applications.
4. Discuss tasks such as assigning functions to data, supervised and unsupervised learning.

These goals build a strong foundation, helping you grasp the essentials before we dive deeper into mathematical tools and techniques.

Why Start Here?

Understanding the foundational definitions of data is crucial before diving into analysis, modeling, or machine learning techniques. This ensures clarity and precision in all subsequent discussions.

It's like learning the rules of a game before playing—without knowing what data is, we can't analyze it effectively. Starting here prevents misunderstandings and sets the stage for more advanced topics, making the learning process smoother and more intuitive for everyone.

Data Definitions

Definition

Let X and Y be sets. (Think of X as the space of inputs, like measurements, and Y as outputs or categories, like exam grades or labels like "pass" or "fail.") This setup helps us formalize how we handle different types of information in data science.

(i) A finite subset $D \subseteq X$ is called an unlabeled dataset in X, its elements are called data points. This is simply a collection of points without any associated outputs or categories.
e.g., a list of daily temperatures. Unlabeled datasets are often used when we're exploring patterns or structures without prior knowledge of categories.

Data Definitions

Definition

(i) A finite set $D \subseteq X \times Y$ is called a labeled dataset. If $(x, y) \in D$, then x is the feature part and y is its label. Here, each data point has an input x and an associated output or category y.
e.g., hours studied (x) and exam score (y). Labeled datasets are essential for tasks where we need to predict outcomes based on known examples.

Data Definitions

Definition

(iii) A labeled dataset $D \subseteq X \times Y$ is called categorically labeled, if Y is finite and continuously labeled, if Y is a continuum.[a] Categorical labels are discrete (*e.g.*, "cat" or "dog"), while continuous labels are real-valued (*e.g.*, heights or temperatures).
Categorical: like menu choices; continuous: like ruler measurements.
Understanding this distinction is key because it determines the types of models and algorithms we can apply, such as classification for categorical and regression for continuous.

[a] Here, one should think of a closed, open, half-open, bounded or unbounded interval in $\mathbb{R}$ that has at least two points. Occasionally, we may however also consider more general subsets of $\mathbb{R}$.

Data Definitions

Definition

(iv) If $X = X_1 \times \cdots \times X_d$ and $x = (x_1, \ldots, x_d)$, we call the x_i's the features of x. If $Y = Y_1 \times \cdots \times Y_m$, we speak of multidimensional labels. Features are individual attributes, and labels can have multiple dimensions.

e.g., features: age, height; multidimensional label: weight, blood pressure. This structure allows us to handle complex, multi-faceted data in a systematic way, breaking down high-dimensional problems into manageable parts.

Geometric and Metric Structure

Often, $X = \mathbb{R}^d$, equipped with the Euclidean norm and standard scalar product to measure distances and angles. If $X \subseteq \mathbb{R}^d$ is not a vector space, we use the Euclidean metric. In some cases, (X, ρ) is an abstract metric space, where ρ is a distance function, possibly a distance measure (not satisfying all metric properties). For categorical labels, $Y = \{1, \ldots, m\}$ for simplicity.

This lets us measure how "close" data points are, crucial for clustering.

These geometric tools provide a mathematical foundation for understanding relationships between data points, enabling techniques like nearest neighbors or dimensionality reduction, which we'll explore later.

Handling Duplicate Data Points

Remark: Duplicate Data Points

To allow duplicates (*e.g.*, repeated measurements), replace X with $X \times \mathbb{N}$. In $D \subseteq X \times \mathbb{N}$, x can appear as $(x, 1)$, $(x, 2)$, etc. We write x_i or $x^{(i)}$ for simplicity. A dataset $D = \{x^{(1)}, \ldots, x^{(n)}\} \subseteq X$ allows duplicates via indexing, modeling multisets.

Like multiple copies of a book in a library, each with a unique ID. Handling duplicates this way ensures our models account for real-world data collection practices, where repetitions can provide valuable information about variability or frequency.

Modeling Choices

In applications, we decide whether coordinates of a data vector $(x_1, \ldots, x_d)$ are features (unlabeled) or labels. This choice depends on the problem. Flexibility in modeling lets us tailor data to our goal, *e.g.*, predicting or exploring patterns. This decision-making process is a critical skill in data science, as it directly impacts the effectiveness of our analysis and the insights we derive.

Example (i): Student Exam Data

Example

Consider ten students in "Mathematical Introduction to Data Science". We record:

1. Preparation time (hours, $[0, 168]$).
2. Social media time (hours, $[0, 168]$).
3. Exam result (percentage, $[0, 100]$).

This can be an unlabeled dataset $D \subseteq \mathbb{R}^3$ (all as features) or a labeled dataset $D \subseteq [0, 168]^2 \times [0, 100]$ (exam result as continuous label).
Same data, different goals: explore patterns or predict scores. This example shows how the same raw information can be framed differently depending on whether we're interested in discovery or prediction, highlighting the importance of problem formulation.

Student Exam Data Table

Student	Prep time (h)	Social media (h)	Exam result (%)
1	0.0	20.0	0.0
2	1.5	8.5	2.0
3	2.0	6.0	7.0
4	2.0	6.0	10.5
5	8.0	10.0	29.5
6	8.5	3.0	49.0
7	9.5	0.0	59.5
8	12.0	2.0	63.5
9	18.0	4.0	85.0
10	19.0	0.5	98.0

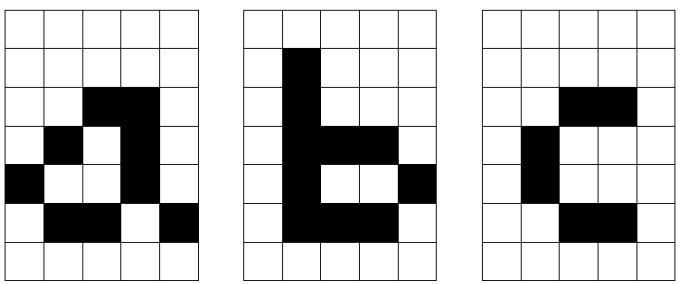
This table illustrates real numerical data, helping us visualize how features and labels interact in a practical setting.

Example (ii): Handwritten Letters

Example

Handwritten letters as (7×5)-matrices of 1s and 0s form a labeled dataset $D \subseteq \mathbb{R}^{7\times5} \times \{a, b, c, \dots\}$. Flattening each matrix gives $\mathbb{R}^{35}$. Features are pixel values (0 or 1), labels are letters (categorical).

Each letter is a grid of pixels turned into a vector, labeled by its letter. This representation is common in image processing, where we convert visual data into numerical forms that algorithms can handle, paving the way for applications like optical character recognition.



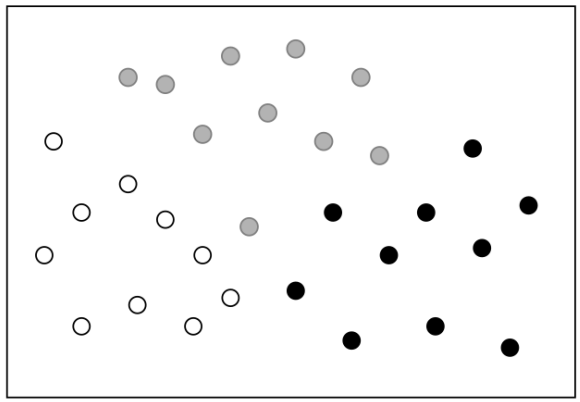
Illustrations of handwritten letters as 7×5 binary matrices. Notice how variations in writing styles can challenge classification models.

Example (iii): 2D Categorical Data

Example

Let $X \subseteq \mathbb{R}^2$ be a cuboid, $Y = \{1, 2, 3\}$. Each point has two features (coordinates) and a categorical label ($1 =$ white, $2 =$ gray, $3 =$ black). Useful for visualizing clustering or classification.

Like points on a map, color-coded by group. This simple 2D example helps us understand higher-dimensional problems by analogy, showing how labels can reveal underlying structures in data.



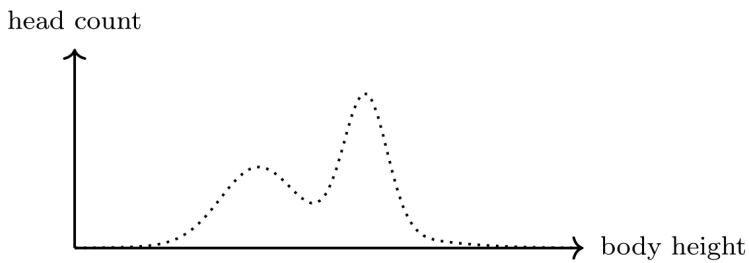
Scatter plot with categorical labels (white: 1, gray: 2, black: 3). Observe the separation between groups, which is ideal for classification tasks.

Example (iv): Body Heights

Example

Recording body heights of course participants gives an unlabeled dataset $D \subseteq (0, \infty)$. A histogram shows a bimodal distribution, suggesting subgroups (*e.g.*, genders).

Two peaks in the histogram hint at two groups, like male and female heights. This demonstrates how visual tools like histograms can uncover hidden patterns in unlabeled data, guiding further analysis.



Distribution of body heights. The bimodal shape is a classic indicator of mixed populations.

Assigning Functions: Overview

For a labeled dataset $D \subseteq X \times Y$, we aim to find a function $f : X \to Y$ to predict labels for new features.
Like finding a rule to predict grades from study time. This is the core of predictive modeling.
Three perspectives:

1. Approximation
2. Probability
3. Optimization

Each perspective offers a different lens, helping us choose the right approach based on the data's nature and assumptions.

Perspective 1: Approximation

1. **Approximation Perspective:** Assume a true function $f_0 : X \to Y$, with $D = \{(x_1, f_0(x_1)), \ldots, (x_n, f_0(x_n))\}$. Aim: $f \approx f_0$. Assumes deterministic data.
   Like guessing a perfect formula for exam scores. This view is ideal for scenarios where data follows a clear, noise-free pattern, though real data often has some variability.

Perspective 2: Probability

2. **Probability Perspective:** Data has random noise, *e.g.*, $y_i = f_0(x_i) + \varepsilon_i$, where ε_i is normally distributed. Find f that data likely comes from.
   Accounts for errors, like measurement inaccuracies. This probabilistic approach is powerful for handling uncertainty, which is inherent in most real-world datasets.

Perspective 3: Optimization

3. **Optimization Perspective:** Minimize a cost function $\phi(f, D)$ to fit data, without assuming a true f_0.
   Like drawing the best line through scattered points. This method focuses on practical fitting, making it versatile for complex models where underlying truths are unknown.

Terminology

The function f is called:

1. Regressor (continuous Y, *e.g.*, temperatures).
2. Classifier (categorical Y, *e.g.*, letters).
3. Predictor or approximant.

Dataset D is training data, process is supervised learning.
These terms are fundamental in machine learning, helping us categorize models and techniques based on the type of prediction required.

Examples in Context: Exam Data

In Example (i), use linear regression:

$$\text{exam result} = a_1 \cdot \text{prep time} + a_2 \cdot \text{social media time} + b$$

Determine a_1, a_2, b using data. Fits approximation or probability perspective (with noise).
Suggests study time and social media influence scores. This simple model can reveal correlations, but remember, correlation doesn't imply causation—further analysis is needed to interpret results.

Examples in Context: Handwritten Letters

In Example (ii), minimize a cost function to classify letters, fitting the optimization perspective.

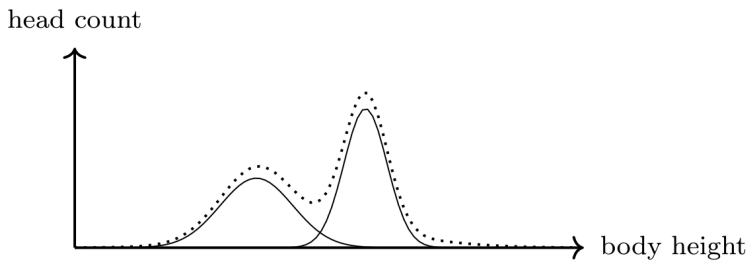
No assumed rule, just find the best function to match data. This is typical in neural networks, where optimization algorithms adjust parameters to improve accuracy over iterations.

Examples in Context: Physical Measurements

For continuous labels from physical measurements, assume normally distributed errors, fitting the probability perspective.
e.g., height measurements with small errors due to tools or posture. This perspective allows us to quantify uncertainty and make confident predictions with error bounds.

Unsupervised Learning: Overview

For an unlabeled dataset $D \subseteq X$ (*e.g.*, body heights), assign a function to find patterns. Example (iv) suggests two normal distributions (male/female heights).



Superposition of two normal distributions for body heights. This visualization helps illustrate how mixtures of distributions can explain observed data patterns.

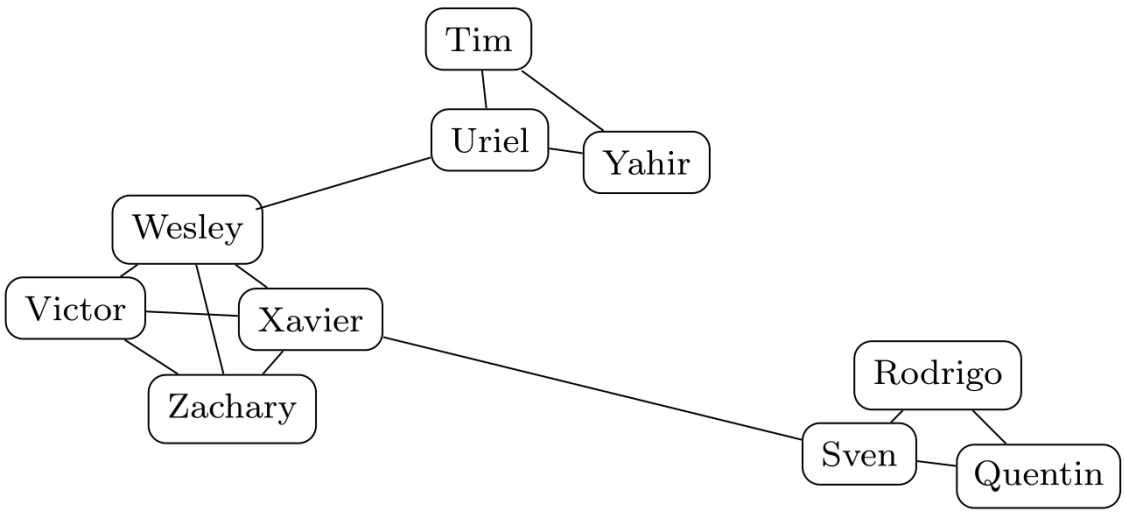
Unsupervised Learning: Goals

Goal: Find $f : D \to \{m, f\}$ to assign gender, separating clusters. Then estimate mean and variance of each distribution.
Like splitting people into teams by height, then finding average heights.
This process uncovers hidden structures, which can lead to new insights or hypotheses.
This is unsupervised learning: discovering structure without labels.
Unlike supervised learning, here the algorithm learns from the data itself, making it useful for exploratory analysis.

Further Example (i): Social Network

Example

A social network where edges indicate friendships. Task: Find clusters of closely connected users. A classic graph clustering problem.
Like finding friend groups in a large school. Clustering in graphs can reveal communities, which has applications in marketing, epidemiology, and more.



Social network with friendships as edges. Notice how denser connections form natural groups.

Further Example (ii): Movie Ratings

Example

Ratings (0–5) for five movies by seven reviewers. Tasks:

1. Cluster reviewers with similar tastes.
2. Predict ratings (matrix completion).

Rows or columns as data points.

Like grouping people by movie preferences or predicting new ratings. This is the basis for recommendation systems, like those used by Netflix.

	Alien	**Casablanca**	**Star Wars**	**Titanic**	**Matrix**
Abbie	0	2	0	2	1
Bailey	1	0	1	0	1
Catherine	5	0	5	0	5
Darlene	0	4	0	4	2
Elena	3	0	3	0	3
Fatima	0	5	0	5	0
Gladys	4	0	4	0	4

Observe patterns in ratings that might indicate clusters of similar reviewers.

Further Example (iii): Grayscale Image

Example

A 320×240 grayscale image (matrix of 0 to 1). Task: data compression via low-rank approximation.

Like shrinking a photo to save space while keeping it recognizable.

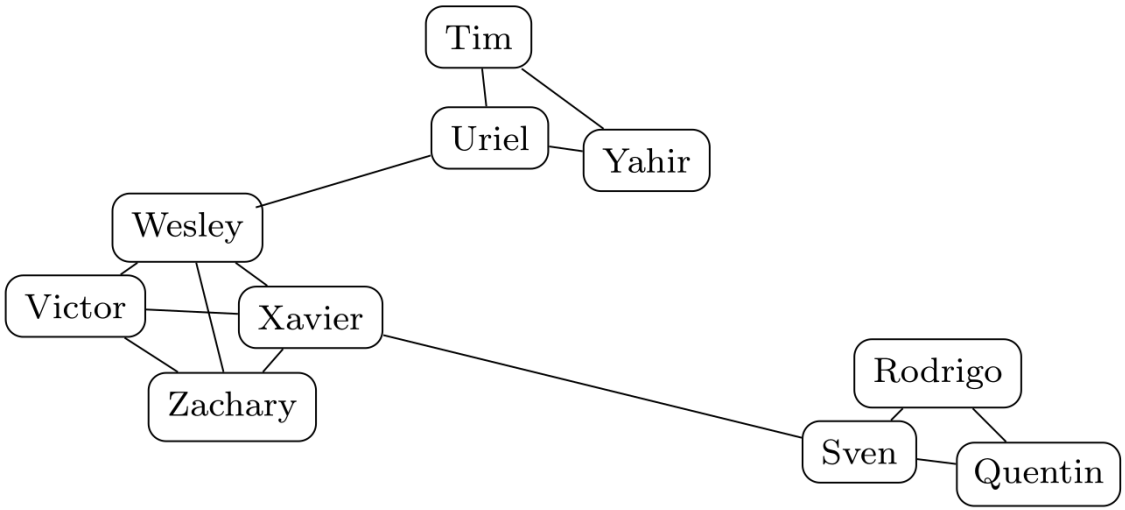
Compression techniques are crucial for handling large datasets efficiently.

Data Modeling: Social Network

For the social network, represent user i as a vector $a_i = (a_{i1}, \ldots, a_{id}) \in \mathbb{R}^d$, where $a_{ij} = 1$ if friends, else 0. Use symmetric data matrix:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix}$$

This matrix representation allows us to apply linear algebra tools for analysis, such as finding eigenvalues for clustering.



Social network with friendships as edges. The matrix encodes this graph

Data Modeling: Movie Ratings

Movie ratings are a matrix. Model rows (reviewers, $\mathbb{R}^5$) or columns (movies, $\mathbb{R}^7$) as data points.

Analyze from reviewer or movie perspective, depending on goal. This duality is useful for collaborative filtering in recommendation systems.

	Alien	**Casablanca**	**Star Wars**	**Titanic**	**Matrix**
Abbie	0	2	0	2	1
Bailey	1	0	1	0	1
Catherine	5	0	5	0	5
Darlene	0	4	0	4	2
Elena	3	0	3	0	3
Fatima	0	5	0	5	0
Gladys	4	0	4	0	4

Data Modeling: Grayscale Image

The grayscale image is a matrix in $\mathbb{R}^{320\times240}$. Compression uses low-rank approximation. Rows as points in $\mathbb{R}^{320}$ can be projected onto 5- or 15-dimensional subspaces via singular value decomposition.
SVD is a powerful tool for reducing dimensions while preserving key information, essential for big data handling.

$\mathrm{rk} = 5$ $\mathrm{rk} = 15$

Left: Rank-5 approximation; Right: Rank-15 approximation. Higher ranks capture more details but require more storage.

Dimensionality Reduction

Many datasets have high dimensions (*e.g.*, social media posts as feature vectors with millions of dimensions, but few data points). Dimensionality reduction is key.

Like summarizing a person's online presence with a few traits. Techniques like PCA or SVD help mitigate the 'curse of dimensionality,' improving model performance and interpretability.

Remark: Machine Learning Caution

Remark

Assigning $f : X \to Y$ via an algorithm is machine learning. *e.g.*, handwritten letters involve minimizing a cost function; movie ratings use eigenvalues. This is mathematical computation, not human learning. Machine learning is about math, not magic. It's important to demystify it—behind the buzzwords are solid mathematical principles that we can understand and apply.

Next Steps

Next Steps: Explore mathematical tools like linear algebra, probability, and optimization for analyzing datasets.
These tools will build on today's foundations, enabling you to tackle more complex problems step by step.

Lecture Source

This lecture is based on Chapter 1 of *Mathematical Introduction to Data Science* by Sven A. Wegner, published by Springer-Verlag GmbH, DE, part of Springer Nature 2024.

Refer to this book for deeper dives into the topics we covered today.

Questions?