CS109A: Introduction to Data Science
Lecture 01: What is Data Science?

Harvard University

Fall 2024

- **Course:** CS109A: Introduction to Data Science
- **Lecture:** Lecture 01: Course Introduction
- **Instructor:** Pavlos Protopapas, Kevin Rader, Chris Gumb
- **Objective:** Understand the definition, history, and process of data science; learn course policies and expectations

Key Summary

This document is a comprehensive guide to the first lecture of CS109A "Introduction to Data Science" at Harvard University. It covers the fundamental definition of data science, its historical development from ancient times to the modern era, the five-step data science process, and the three core components that make up this interdisciplinary field. Additionally, it provides detailed course logistics including grading policies, the critical attendance requirements, and important prerequisites. The lecture concludes with an introduction to web scraping as the first practical skill students will learn.

Contents

1 Course Overview

1.1 What This Course Is About

Welcome to CS109A, also known as AC209A, STAT 109A, and CSCI E-109A for Extension School students. This course is the **beginning of your journey** into data science and artificial intelligence—not the destination.

Warning

Setting Realistic Expectations

You will **not** become an expert in AI and data science by taking this course alone. Think of this course as laying the foundation. There are many more things to learn after this, including:

- CS109B: Advanced Topics in Data Science (Spring semester)
- AC215: Advanced Practical Data Science
- Machine Learning, NLP, and Deep Learning courses

1.2 The "Wax On, Wax Off" Philosophy

Professor Protopapas describes the teaching philosophy using a famous scene from the movie "The Karate Kid." In this movie, the teacher (Mr. Miyagi) makes his student repeatedly wax cars with specific motions—"wax on, wax off"—before teaching any actual karate. The student is frustrated, but later discovers that these motions became the foundation for perfect defensive moves.

Example:

The Karate Kid Analogy **The Modern Temptation:** Students often want to jump straight to running large language models (LLMs), building AI bots, and starting companies.

The Old-Fashioned Approach: This course insists on understanding the fundamentals first. You will:

- Train models and validate them repeatedly
- Understand *what* is happening inside models
- Learn *why* models work the way they do
- Avoid the "dot fit pandemic"—blindly calling `model.fit()` without understanding

1.3 Learning Roadmap

The course follows a logical progression through the data science workflow:

1. **Weeks 1-2: Data Collection & Exploration**
   - Web scraping and data wrangling
   - Exploratory Data Analysis (EDA)
   - Data visualization
2. **Weeks 3-5: Regression**
   - K-Nearest Neighbors (KNN) for regression

- Simple and multiple linear regression
- Model selection with cross-validation
- Regularization (Ridge, Lasso)

3. **Week 6: Bayesian Modeling** (New this year!)
   - Bayesian inference framework
   - Bayesian linear regression
4. **Weeks 7-9: Classification**
   - KNN for classification
   - Logistic regression
   - **Midterm exam** during this period
5. **Week 10: Data Issues**
   - Missing data (Missingness)
   - Causal inference
   - Bias and ethics in data science
6. **Weeks 11-14: Tree-Based Models**
   - Decision trees
   - Bagging and Random Forests
   - Boosting methods

2 What is Data Science?

2.1 The Simple Answer

At its core, **data science is the process of extracting meaningful insights and value from data**. But to truly understand what this means, let's look at it from multiple perspectives.

Definition:

Data Science **Data Science** is an interdisciplinary field that combines:

- **Mathematics & Statistics**: For modeling, hypothesis testing, and prediction
- **Computer Science & IT**: For data collection, storage, processing, and software development
- **Domain Knowledge**: Expert understanding of the specific field being studied (medicine, astronomy, finance, etc.)

2.2 A Historical Perspective: Four Stages of Understanding the World

To understand where data science fits in human history, consider how humanity has approached understanding the world:

2.2.1 Stage 1: Empirical Observation (Ancient Times)

Long ago, humans learned by direct observation and recording:

- **Counting stars**: Ancient astronomers looked at the night sky and counted stars in different regions
- **Recording crops**: Farmers tracked harvests to predict yields
- **The Antikythera mechanism**: An ancient Greek computer (found at the bottom of the sea in Greece) that calculated planetary orbits using gears

This was essentially **early statistics**—collecting data and making observations.

2.2.2 Stage 2: First Principles and Equations (Modern Science)

Scientists like Newton, Einstein, Maxwell, and Schrödinger developed **fundamental equations** that describe how the universe works:

- Newton's Laws: $F = ma$ (Force equals mass times acceleration)
- Einstein's equation: $E = mc^2$ (Energy equals mass times the speed of light squared)
- Maxwell's equations: Describe electromagnetism
- Navier-Stokes equations: Describe fluid dynamics
- Schrödinger's equation: Describes quantum mechanics

This represented a shift from just *observing* to *understanding from first principles*.

2.2.3 Stage 3: Computation (20th Century)

A problem emerged: many of these fundamental equations are **too complex to solve analytically**. Mathematicians couldn't provide closed-form solutions to many differential equations.

Solution: Use **computers** to numerically simulate and approximate solutions. This gave birth to computational science.

2.2.4 Stage 4: Data Science (Modern Era)

Key Information

The Key Insight of Data Science

Data science represents a paradigm shift: instead of requiring complete understanding of first principles (Stage 2), we can use **massive amounts of data (Stage 1)** combined with **powerful computing (Stage 3)** to approximate or predict how the world works.

In other words, data science sometimes "skips" the equation stage, focusing instead on patterns in data.

This can feel **unsettling for mathematically-oriented students**. Professor Protopapas acknowledges: "It bothers me too, but first step—let's learn how to do it, and then we'll learn how to do more than that."

2.3 The Three Pillars of Data Science

Data science exists at the intersection of three domains:

Table 1: *The Three Pillars of Data Science*

Pillar	Description
Math & Statistics	Probability, statistical inference, linear algebra, calculus—the mathematical foundation for modeling and prediction
Computer Science	Programming, algorithms, data structures, databases—the tools for handling and processing data
Domain Knowledge	Understanding of the specific field (astronomy, medicine, finance)—essential for asking the right questions

Warning

The Critical Importance of Domain Knowledge

Professor Protopapas shares a personal story: As an astronomer, he once gave a research problem to a computer scientist friend. A month later, the friend said "Everything is solved!" But it turned out they solved the **wrong problem** because they didn't understand the astronomical context.

Lesson: Data science is **not** like taking your car to a mechanic. You can't just hand over your data and say "Find something interesting." You must:

- Stay involved with the analysis
- Understand at least the basics of the domain
- Collaborate closely with domain experts

3 The Potential and Risks of Data Science

3.1 Amazing Applications

Data science and AI have tremendous potential for positive impact:

Positive Applications

1. **Medical Diagnosis**: Detecting malaria from blood smear images automatically
2. **Drug Discovery**: Using language models to discover new drug combinations
3. **Autonomous Vehicles**: Self-driving trucks for safe night shipping
4. **Generative AI**: Creating images from text prompts (e.g., "a Greek-American professor with glasses and a beard")

3.2 Critical Risks and Ethical Concerns

Serious Risks and Biases

1. **Gender Bias**: Hiring algorithms that favor male candidates for engineering roles because they were trained on historical data that reflected past discrimination
2. **Racial Bias**: Recidivism prediction models (like COMPAS used in US courts) that unfairly predict higher risk for people of color
3. **Misinformation**: AI-generated content that's indistinguishable from reality

Important:

Being a Critical Thinker AI models are trained on data that contains human biases. As Harvard students, you must:

- Question the data: Where did it come from? Who collected it? What biases might exist?
- Question the model: Is it fair? Does it work equally well for all groups?
- Question the application: Should this model be used at all? What are the consequences?

Don't just accept AI results because a computer produced them. **Be critical thinkers.**

3.3 Career Advice for Uncertain Times

Professor Protopapas acknowledges that students today face unprecedented uncertainty:

- AI might replace jobs
- It's hard to distinguish real from fake information
- Career paths that existed yesterday might disappear tomorrow

Key Information

Simple Advice

"Don't try to over-optimize for every trend. Things change so fast. Instead: **Learn things very**

well, become good at them, and lead. Don't ignore trends—they are there—but don't let them control your life either."
— Professor Protopapas

4 The Five-Step Data Science Process

Data science projects typically follow a cyclical process with five key stages:

4.1 Step 1: Ask an Interesting Question

This is the **most important and first step**. Many projects fail because they start with "Here's some data, find something interesting" rather than a clear hypothesis.

Definition:

Good Questions in Data Science Good data science questions are:

- **Specific**: "Can we predict hospital readmission rates within 30 days?" rather than "Tell me something about hospitals"
- **Measurable**: There must be data that can address the question
- **Actionable**: The answer should lead to meaningful action
- **Scientific**: Based on hypotheses that can be tested

Key questions to ask yourself:

- What are we trying to predict or estimate?
- If we had all the data in the world, what would we do with it?

4.2 Step 2: Get the Data

Once you have a question, you need data to answer it.

Warning

Data Collection Considerations

- **How was the data sampled?** Random? Convenience? This affects what conclusions you can draw
- **What data is relevant?** More data isn't always better—irrelevant data adds noise
- **Are there privacy concerns?** Just because data is "available" doesn't mean it's ethical or legal to use
- **What is the license?** Always read the terms of service and data licenses

4.3 Step 3: Explore the Data (EDA)

Exploratory Data Analysis (EDA) is often undervalued but critically important. Professor Protopapas emphasizes: "Trust me on this—I'm old enough to have seen so many attempts to do modeling without data exploration. You're wasting your time."

Why EDA matters:

- Sometimes the answer is immediately obvious from visualization
- Sometimes a simple filter or rule works better than a complex model
- Sometimes the data is garbage, and no model can help

Key EDA questions:

- Have you plotted the data?
- Are there outliers or anomalies?
- Are there obvious patterns?
- Is there missing data?

4.4 Step 4: Model the Data

This is "where the fun is"—building, fitting, and validating models. The course will spend significant time on this step, covering:

1. **Build**: Choose an appropriate model type
2. **Fit**: Train the model on your data
3. **Validate**: Test the model on held-out data

4.5 Step 5: Communicate/Visualize the Results

Data science is an **interdisciplinary field**, which means you'll often communicate with people who don't know your technical terminology.

Key Information

The Art of Storytelling

Professor Protopapas shares: "I love telling stories—for me it's natural." But for many data scientists, this is the hardest part.

A student submitted a project proposal with **three jargon terms in the first title**. The response: "Rewrite."

Key questions for communication:

- What did we learn?
- Does it make sense?
- Can we tell the story effectively to non-experts?

5 Why Choose Data Science?

5.1 It's Fun

If you're in this class, you probably enjoy problem-solving. Data science is fundamentally about solving problems with data.

5.2 It's at the Cutting Edge

You'll work with the latest technologies and methods.

5.3 Career Prospects

Table 2: *Data Science Career Statistics*

Metric	**Value**
Average Salary	High (varies by location and experience)
Job Satisfaction	Among the highest in tech careers
Job Availability	Consistently ranked among "Best Jobs in America"

5.4 It's Accessible

The barrier to entry is lower than many technical fields. With dedication, you can learn the skills needed to start working in data science.

Key Summary

Professor Protopapas's Life Philosophy

"In principle, if you learn these things, you get decent jobs and you enjoy it. I think that's the secret of life—doing something you like and not lacking money. I do believe lack of money brings unhappiness, but money may not bring happiness. So at least you eliminate that."

6 Course Logistics and Policies

6.1 The Teaching Team

6.1.1 Pavlos Protopapas

- Scientific Director for Data Science and CSC programs
- Research focus: Astronomy + Machine Learning + AI + Statistics (Lab: Stellar DNN)
- Fun facts: Certified chef (trained at Le Cordon Bleu), classical music enthusiast, self-proclaimed "worst soldier in NATO" during Greek army service

6.1.2 Kevin Rader

- Senior Preceptor in the Statistics Department, Associate DUS
- Research focus: Medicine and sports analytics
- Fun facts: Philadelphia Eagles superfan ("Go Birds!"), passionate about growing and cooking food, first-time girls' soccer coach (currently 0-1)

6.1.3 Chris Gumb

- Preceptor in Computer Science
- Helps coordinate the teaching staff and approximately 30 Teaching Fellows (TFs)
- Fun fact: Big movie fan (Homework 1 involves movie theater data)

6.2 Two Perspectives on Data Science

The course intentionally brings together two viewpoints:

Professor Protopapas: The CS/ML Perspective

Focus on:

- Machine learning and optimization
- Getting better models (higher R^2, lower MSE)
- Improving prediction accuracy

Professor Rader: The Statistical Perspective

Focus on:

- Understanding **relationships** between variables
- Quantifying **uncertainty** in predictions
- Examining whether relationships differ for different groups
- Considering **data issues, biases, and ethics**

"I don't care how accurate your model is—it could still be trash if you don't understand the relationships and uncertainty."

6.3 Grading Components

Table 3: *CS109A Grading Breakdown*

Component	Weight	Details
Homework	30%	HW0 (1%) + HW1-5 (29%). Pair work recommended for HW1-5.
Section Quizzes	10%	Two 30-minute quizzes during section (5% each)
Midterm	18%	Conceptual portion in section + take-home coding portion
Final Exam	22%	3-hour seated exam with conceptual and coding portions
Project	20%	Group project (3-5 students), open-ended data science project

6.4 The Critical Attendance Policy

Important:

Attendance Directly Affects Your Maximum Possible Grade Attendance in CS109A is **required** and directly impacts your grade ceiling:

Minimum Attendance	Maximum Possible Grade
$\geq$ 66% (two-thirds)	A
$\geq$ 50% (one-half)	A-
$\geq$ 33% (one-third)	B+
$<$ 33%	B or below

Important: Meeting the attendance threshold doesn't *guarantee* that grade—it only *qualifies* you for it. You still need to earn the grade through your work.

Flexibility: 66% means you can miss one-third of classes. You could attend all lectures and skip most sections (except for quizzes/midterm), or attend all sections and skip half of lectures.

6.5 Late Days

- **Earning Late Days**: For every **4 sessions** you attend (lecture or section), you earn **1 late day**
- **Using Late Days**: Maximum of **2 late days** per assignment
- **DCE Students**: Automatically receive **4 late days** (attendance can't be tracked remotely)
- **48-hour limit**: After the deadline plus your late days, assignments cannot be submitted (grading must begin)

6.6 Prerequisites

Table 4: *CS109A Prerequisites*

Area	Requirements
Python Programming	**Critical**. If you have never programmed before, this course will be very challenging. CS50 or equivalent experience required.
Calculus	Basic calculus (Math 1B equivalent). Occasional linear algebra and multivariable calculus concepts appear but won't be tested heavily.
Statistics/Probability	Stat 104 (data-focused) is ideal. Stat 110 (theory-focused) is good for probability but may lack practical data experience.

Warning

Self-Assessment with Homework Zero

HW0 is designed to test whether you meet the prerequisites.

- If you struggle in **all three areas** (coding, math, stats): seriously consider taking the course next year
- If you struggle in **one area**: you can probably catch up with help from TFs
- **Coding experience is the most critical**: Math and stats gaps can be filled, but lack of coding experience is a "little bit more problematic"

6.7 Course Platforms

- **Ed (Edstem)**: Lecture slides, section materials, announcements, **discussion forum** (primary Q&A platform)
- **Canvas**: Video recordings, assignment submissions, official schedules, **grades**

6.8 Getting Help

In order of preference:

1. **Ed Discussion Forum**: Fastest response; classmates and TFs can help
2. **Office Hours**: Best for in-depth conceptual help or assignment debugging
3. **Course Helpline Email**: Administrative questions (e.g., section changes)
4. **Direct Email to Professors**: Personal or sensitive matters only

7 Introduction to Web Scraping

The first practical skill you'll learn in this course is **web scraping**—the automated extraction of data from websites.

7.1 What is Web Scraping?

Web scraping is a technique that allows you to programmatically collect data from websites. Instead of manually copying information, you write code that:

1. Fetches the HTML source code of a webpage
2. Parses the HTML to find specific elements
3. Extracts the data you need
4. Stores it in a structured format (like a spreadsheet or database)

7.2 Key Libraries

- **requests**: Makes HTTP requests to fetch webpage content
- **BeautifulSoup**: Parses HTML and makes it easy to navigate and search
- **pandas**: Organizes extracted data into DataFrames for analysis
- **matplotlib**: Visualizes the collected data

7.3 Ethical Considerations

Warning

Before You Scrape

Not all websites allow scraping. Always check:

- **robots.txt**: Visit `website.com/robots.txt` to see what's allowed/disallowed
- **Terms of Service**: Read the website's ToS for data collection policies
- **Rate Limiting**: Don't overwhelm servers with too many requests too quickly
- **Personal Data**: Be especially careful with data that might contain personal information

Just because data is "publicly available" doesn't mean it's okay to scrape and use it.

7.4 Basic HTML Concepts

Websites are written in **HTML (HyperText Markup Language)**. Understanding basic HTML helps you scrape effectively:

- **Elements**: Everything in HTML is organized into elements, marked by tags
- **Tags**: Define the type of content: `<h1>` (heading), `<p>` (paragraph), `<a>` (link), `<div>` (division/-container)
- **Attributes**: Provide additional information: `href` (link destination), `class` (styling identifier), `id` (unique identifier)

Example:

Using Browser Developer Tools The easiest way to understand a webpage's structure:

1. Right-click on the element you're interested in
2. Select "Inspect" or "Inspect Element"
3. The Developer Tools panel opens, highlighting the HTML for that element
4. Use the picker tool (arrow icon) to click on different elements and see their HTML

This is essential for figuring out which tags and classes to target in your scraping code.

7.5 Step-by-Step Web Scraping Example

7.5.1 Step 1: Fetch the Webpage

```python
import requests

# URL of the webpage we want to scrape
url = "https://www.nobelprize.org/all-nobel-prizes/"

# Send an HTTP GET request
response = requests.get(url)

# Check if request was successful (status code 200 = OK)
print(f"Status Code: {response.status_code}")

# Get the HTML content as text
html_text = response.text
print(html_text[:200])  # Print first 200 characters
```

Listing 1: Fetching HTML with requests

Status Codes to Know:

- **200**: Success (OK)
- **404**: Page not found
- **403**: Forbidden (you're not allowed to access)
- **429**: Too many requests (you're being rate-limited)

7.5.2 Step 2: Parse the HTML

```python
from bs4 import BeautifulSoup

# Create a BeautifulSoup object
soup = BeautifulSoup(html_text, 'html.parser')

# Find the page title
page_title = soup.title.get_text()
print(f"Page Title: {page_title}")

```

```
# Find all elements with a specific class using CSS selectors
prize_blocks = soup.select('div.card-prize')
print(f"Found {len(prize_blocks)} prize blocks")

# Get text from the first block
first_block = prize_blocks[0]
block_text = first_block.get_text().strip()
print(block_text)
```

Listing 2: Parsing HTML with BeautifulSoup

7.5.3 Step 3: Extract and Structure the Data

```
import re  # Regular expressions library

# Helper functions using lambda
get_title = lambda block: block.select_one('h3').get_text().strip()
get_year = lambda block: re.search(r'(\d{4})',
                          block.select_one('h3').get_text()).group(1)
get_description = lambda block: block.select_one('blockquote').get_text().
    strip()

# List to store our data
nobel_data = []

# Loop through all prize blocks
for block in prize_blocks:
    try:
        nobel_data.append({
            'title': get_title(block),
            'year': int(get_year(block)),
            'description': get_description(block)
        })
    except Exception as e:
        print(f"Error extracting data: {e}")

# Example analysis: Find unique prize categories
unique_titles = set(item['title'] for item in nobel_data)
print(f"Unique categories: {unique_titles}")
```

Listing 3: Extracting data into a structured format

7.5.4 Step 4: Convert to pandas DataFrame

```
import pandas as pd

# Convert list of dictionaries to DataFrame
df = pd.DataFrame(nobel_data)

# View the first few rows
```

```
print(df.head())

# Save to CSV file
df.to_csv('nobel_prizes.csv', index=False)
```

Listing 4: Creating a DataFrame

8 Frequently Asked Questions

- **Q: Can I audit this course?**
- A: Yes, but if you audit, you cannot take the course for credit later.
- **Q: Can I take the course asynchronously (watching recordings only)?**
- A: **College students**: No. **Graduate students**: Not recommended. With less than 33% attendance, your maximum grade is B.
- **Q: I'm missing some prerequisites. Should I still take this course?**
- A: Complete HW0 and see how you do. Math/stats gaps can be addressed with TF help. However, if you have **zero programming experience**, strongly consider postponing to next year.
- **Q: Can I reschedule the midterm for travel?**
- A: No. The midterm (week of Oct 22-24) and final exam (Dec 11) dates are fixed. Plan accordingly.
- **Q: Can I use my own project idea?**
- A: Yes, but the data must be **public** (shareable with your group), and you must work in a group of 3-5 students.
- **Q: What if I miss a class?**
- A: All lectures are recorded and available on Canvas. Missing one class won't affect your grade. Just maintain at least 66% attendance to be eligible for an A.

9 Key Takeaways

Key Summary

Summary of Lecture 01

What is Data Science?

- An interdisciplinary field combining math/statistics, computer science, and domain knowledge
- Represents a paradigm shift: using data + computing to understand the world, sometimes bypassing traditional equations
- Has tremendous potential (medical diagnosis, drug discovery) but also serious risks (bias, ethics)

The Data Science Process:

1. Ask an interesting question (hypothesis-driven)
2. Get the data (ethically and legally)
3. Explore the data (EDA—don't skip this!)
4. Model the data (build, fit, validate)
5. Communicate results (tell the story)

Course Philosophy:

- "Wax on, wax off"—master the fundamentals before building complex systems
- Don't just use `.fit()`—understand what's happening inside
- Balance ML optimization with statistical understanding

Critical Logistics:

- Attendance is required: 66% minimum for A eligibility
- Prerequisites: Programming experience is essential; math/stats gaps can be filled
- Pair work encouraged for homeworks