

A Short Philosophical Detour

Why Are we Here?

Learning Objectives

1. **Apply** data science skills using real-world data to **create** meaningful, interactive applications addressing authentic problems.
2. **Analyze** and **implement** practical machine learning approaches to **solve** real-world problems, emphasizing pragmatic solutions over deep theory.
3. **Examine** and **employ** large language models (LLMs) to **enhance** data-driven applications.

Survey Results

Core Technical Skills

- Python programming fundamentals (most common response)
- Building applications/products with code
- Data visualization and creating visual outputs

AI/ML Deep Dive

- Understanding AI "under the hood"
- Machine Learning algorithms and techniques
- Neural networks and LLMs

Data Science Applications

- Statistical analysis and methods
- Connection between data science and machine learning
- How data drives decision-making

Motivations

- Hands-on coding experience
- Understanding AI, not just using it
- Building "incredible" things

Survey Results

Core Technical Skills

- Python programming fundamentals (most common response) ✓
- Building applications/products with code
- Data visualization and creating visual outputs ?

AI/ML Deep Dive

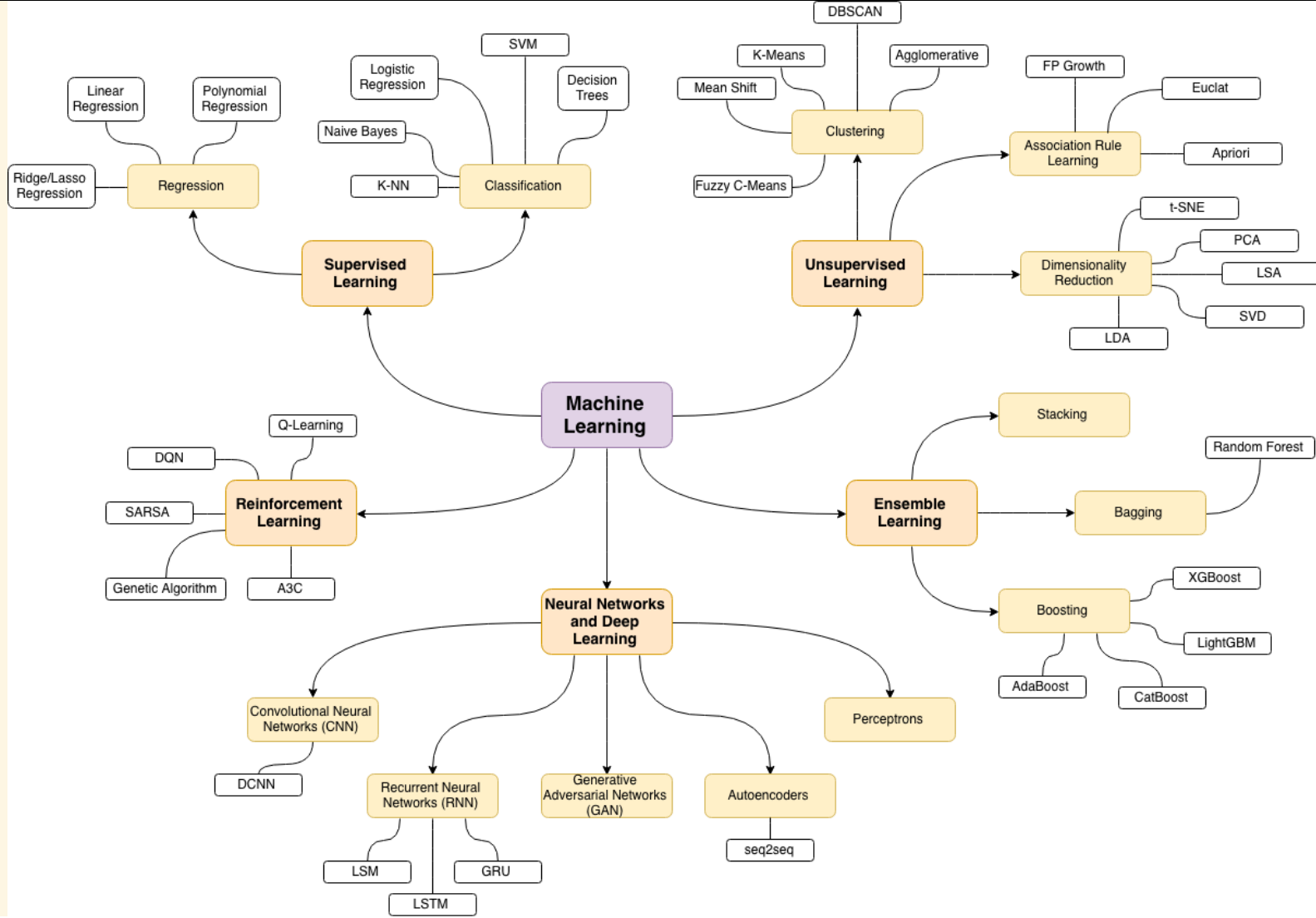
- Understanding AI "under the hood" 🟡
- Machine Learning algorithms and techniques ✓
- Neural networks and LLMs 🟡

Data Science Applications

- Statistical analysis and methods ✓
- Connection between data science and machine learning ✓
- How data drives decision-making ✓

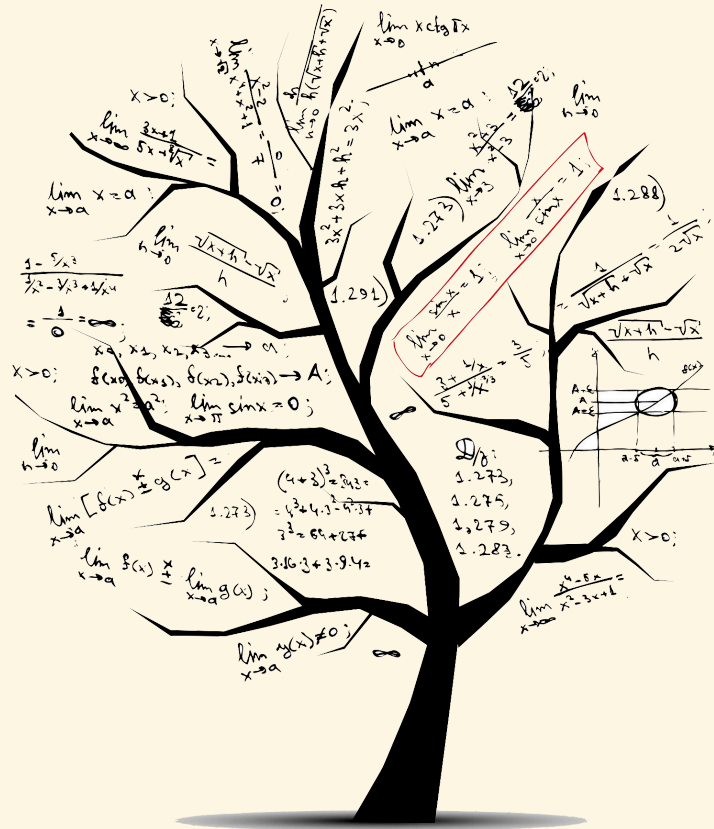
Motivations

- Hands-on coding experience ✓
- Understanding AI, not just using it 🟡
- Building "incredible" things





Tree of Knowledge Climbing Tools



- **Python:** Programming language for data science and machine learning.
- **Basic Statistics:** Fundamental concepts for understanding data distributions and relationships.
- **Pandas:** Library for data manipulation and analysis.
- **Scikit-learn:** Library for machine learning algorithms.

My goal is to teach you *just enough* of these tools to show you some of the fruits up in the branches of this tree.

Why?

If you want an average successful life, it doesn't take much planning. Just stay out of trouble, go to school, and apply for jobs you might like.

But if you want something extraordinary, you have two paths:

1. Become **the best** at **one** specific thing.
2. Become **very good** (top 25%) at **two or more** things.

The first strategy is difficult to the point of near impossibility. Few people will ever play in the NBA or make a platinum album. I don't recommend anyone even try.

The second strategy is fairly easy. Everyone has at least a few areas in which they could be in the top 25% with some effort. In my case, I can draw better than most people, but I'm hardly an artist. And I'm not any funnier than the average standup comedian who never makes it big, but I'm funnier than most people. The magic is that few people can draw well and write jokes. It's the combination of the two that makes what I do so rare.

~ Scott Adams, *How to Fail at Almost Everything and Still Win Big*

SealNet: CNNs to Detect Antarctic Seals

Objective

Detect and count Antarctic pack-ice seals using satellite imagery.

Importance

Seals are key predators in the Southern Ocean ecosystem, and understanding their population sizes helps predict ecosystem responses to threats like climate change and krill fishing.

Methodology

- Use Worldview-3 satellite imagery.
- Develop a Convolutional Neural Network (CNN) to detect and locate seal centroids.
- Combines semantic segmentation heatmaps with binary classification. (?)
- Uses regression for counting.

Results

- Locates over 30% of seals compared to human expert counts.
- Reduces detection time by 95% vs human.

How Do I Know if This is a Skill Worth Getting Good At?

Climbing the Tree for Ourselves

Exploring Data Science

- Look for stories in the news that you think could be followed-up with data.
- Consider a field that you're passionate about. How can your passion be amplified with data?
- Explore data visualization libraries (Seaborn, D3.js), visualize a dataset that interests you.

Exploring Machine Learning

- Look for a dataset that interests you on Kaggle. Build a simple model to predict something.
- Imagine a place where you'd want a machine learning model to help you. Try and make it.

Exploring Programming

- Try a different programming language.
- Build a small website or application (e.g, a personal website).
- Try creating art with code (e.g., using Processing or p5.js).
- Automate a simple task on your computer (e.g., file organization).
- Build a game.

You've Actually Learned More than Python in This Class

Most programming languages share common concepts, such as variables, control structures (like loops and conditionals), functions, and data structures (like lists and dictionaries).

```
# Python
def nth_fibonacci(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    return a
```

```
// Java
public static int nthFibonacci(int n) {
    int a = 0, b = 1;
    for (int i = 0; i < n; i++) {
        int temp = a + b;
        a = b;
        b = temp;
    }
    return a;
}
```

```
// JavaScript
function nthFibonacci(n) {
    let a = 0, b = 1;
    for (let i = 0; i < n; i++) {
        [a, b] = [b, a + b];
    }
    return a;
}
```

```
// Go
func nthFibonacci(n int) int {
    a, b := 0, 1
    for i := 0; i < n; i++ {
        a, b = b, a+b
    }
    return a
}
```

Other Kinds of Programming

Web Development

- Usually split into "Front-end" and "Back-end"
- Front-end: Focuses on the user interface and user experience (HTML, CSS, and JavaScript)
- Back-end: Focuses on server-side logic, databases, and application performance. (Python, Go, Ruby, Java, SQL)

Mobile Development

- Focuses on building applications for mobile devices (smartphones, tablets).
- Technologies include Swift (for iOS), Kotlin (for Android), and cross-platform frameworks like Flutter and React Native.

Hardware Development

- Focuses on designing and building physical devices and systems.
- Entry-level hardware include Arduino, Raspberry Pi.
- Involves programming in languages like C, C++ (some traction in MicroPython).

Site-Reliability Engineering

- Focuses on maintaining and improving the availability and performance of systems.
- Involves monitoring, incident response, and infrastructure management.

Other Kinds of Programming

Data Visualization

- Hybrid of Data Science and Art
- Telling stories with data through visual representation.

Security Engineering

- Focuses on protecting systems, networks, and data from cyber threats.
- Involves penetration testing, vulnerability assessment, and secure coding practices.
- Requires knowledge of security protocols, cryptography, and risk management.

Simulation Programming

- Focuses on creating models to replicate real-world processes or systems.
- Often used in scientific research, engineering, and training simulations.
- Businesses try to build "Digital Twins"

Emulator Development

- Focuses on creating software that mimics the hardware and functionality of other systems.
- Commonly used for gaming, mobile apps, and legacy system support.

Game Creation

Pygame Zero

A beginner-friendly Python library for making games. It simplifies the process of creating 2D games.

```
import pgzrun
def draw():
    screen.clear()
    screen.draw.text("Hello, Pygame Zero!", center=(400, 300))
pgzrun.go()
```

Godot

Very popular, beginner-friendly, and open-source game engine. GDScript is similar to Python.

Unity

Popular game engine using C#. Great for 2D and 3D games.

Pico-8

Fantasy console for making small games. Uses a Lua-like language. **Not free** but has a free trial.

Phaser

JavaScript framework for making 2D games. Great for web-based games.

A Few Demos

What We Have Left

- ☑ Python Fundamentals (Conditionals, Loops, Functions, Data Structures)
- ☑ Pandas and working with Tabular Data
- ☑ Group By and Aggregation Statistics
- ☑ NLP (TFIDF, Sentiment Analysis, Word Embeddings)
- ☑ Supervised Learning (Regression, Classification)
- ☐ Unsupervised Learning (Clustering, Dimensionality Reduction)
- ☐ Neural Networks (Perceptrons, CNNs, RNNs)
- ☐ Large Language Models and AI Trends