DATA SCIENCE

(Supplementary Resource)

Dr. Ömer Gökdaş

CHAPTER 1: INTRODUCTION TO DATA SCIENCE AND PYTHON FUNDAMENTALS

(Week 1: Lecture Notes)

WEEK 1: COLAB AND CODING LOGIC

1.1. INTRODUCTION: WHAT IS DATA SCIENCE AND WHY ARE WE HERE?

Data Science is not just about writing code or plotting graphs. It is the art of extracting **"Value"**, **"Meaning"**, and **"Future Predictions"** from raw data by blending Statistics, Mathematics, and Software skills.

Engineering Vision (Real-World Examples):

- **Civil Engineering:**
  - **Scenario:** We have mixture ratios and strength test results for concrete poured over the last 10 years.
  - **Goal:** To predict the strength of a new mixture with 95% accuracy without going to the laboratory.
- **Industrial/Mechanical Engineering:**
  - **Scenario:** Monitoring real-time vibration data of motors in a factory.
  - **Goal:** To issue a warning "*Bearing failure approaching, perform maintenance*" 2 days before the machine breaks down (**Predictive Maintenance**).

Concept Confusion (The Matryoshka Model): These concepts are often confused but are actually nested within each other:

1. **Artificial Intelligence (AI):**
   - *The Big Umbrella.* Any system or technology that mimics human cognitive abilities such as reasoning, problem-solving, and decision-making.
2. **Machine Learning (ML):**
   - *Sub-branch of AI.* Systems where computers learn rules and patterns on their own by looking at data, without being explicitly programmed. (Main focus of our course).
3. **Deep Learning (DL):**
   - *Sub-branch of ML.* Runs on "Artificial Neural Networks" that mimic the human brain.

- *Key Difference:* While features are provided by humans in ML, DL extracts these features automatically from data (image, sound, text) thanks to its complex network structure. (e.g., ChatGPT, Image Processing).

4. **Data Science (DS):**
    - *Intersecting Discipline.* It is the discipline that intersects with all these sets (AI, ML, DL) but independently involves cleaning, analyzing, visualizing raw data, and converting insights into business decisions.

1.2. PLATFORM: GOOGLE COLAB (Our Cloud Laboratory)

In this course, you do not need to install anything on your computer or look for licenses.

- **What is it?** A Python-loaded notebook that runs in the browser and is hosted on Google servers.
- **Why do we use it?** Even if your computer is slow, we can use Google's powerful processors (GPU/TPU) for free.
- **File Structure:** `.ipynb` (Interactive Python Notebook). In this format, code blocks, text explanations, and graph outputs stand one under another.

1.3. PYTHON: CRASH COURSE

(Note: These are the fundamental bricks required to understand Machine Learning algorithms.)

A) Variables – Data Boxes In Python, there is no need to specify the variable type (int, float, etc.) from the beginning; Python understands the type based on what is put inside the box.

In [1]:
```
# Defining variables
x = 10           # Integer
y = 3.14         # Float
project = "Concrete"  # String (Text)
is_active = True # Boolean (Logical - 1/0)

# A life-saving function to learn the type of data:
print(type(y))   # Output: <class 'float'>
```

```
<class 'float'>
```

B) Lists – Data Warehouses In data science, we work not with a single value, but with thousands of rows of data. Lists are our first warehouses.

In [2]:
```
# Square brackets are used
grades = [40, 50, 90, 60]

# 1. ACCESS (Indexing) - ATTENTION: Counting starts from 0!
print(grades[0])   # First element (40)
print(grades[-1])  # The last element (60) - Practical method

# 2. SLICING - [Start : End]
# Rule: Start is inclusive, END IS NOT INCLUSIVE.
```

```
print(grades[0:2]) # Takes 0th and 1st indices -> [40, 50]

# 3. APPENDING
grades.append(100) # Adds 100 to the end of the list
print(grades)
```

```
40
60
[40, 50]
[40, 50, 90, 60, 100]
```

C) Dictionaries – Labeled Data *(Important: Ancestor of the Pandas DataFrame structure)* Lists keep data in order (0, 1, 2...). However, in data science, we want to call data by its name (e.g., "Age", "Price").

In [3]:
```
# Curly braces are used. {Key: Value}
sample = {
    "code": "N-101",
    "water_ratio": 0.45,
    "strength": 35.2
}

# Accessing data (Calling by name)
print(sample["strength"])  # Output: 35.2
```

```
35.2
```

D) Logical Control (If / Else) – Decision Mechanisms We must teach the computer to ask questions to filter the data.

In [4]:
```
temperature = 25

if temperature > 30:
    print("Too hot for concrete pouring, use ice.")
elif temperature < 5:
    print("Risk of freezing, use heater.")
else:
    print("Environment is suitable.")
```

```
Environment is suitable.
```

E) Loops – Automation We cannot check thousands of data points one by one by hand. Loops do this job.

In [5]:
```
data_points = [10, -5, 20, 0, 30]
clean_data = []

for data in data_points:
    # Let's take only positive (error-free) data
    if data > 0:
        square = data * data      # Perform operation
        clean_data.append(square) # Add to new list

print(clean_data) # Output: [100, 400, 900]
```

```
[100, 400, 900]
```

F) Functions – Parts of the Machine *(Critical Section)* Machine learning models are essentially giant functions: They take an input, process it, and produce an output

(prediction).

In [6]:
```python
# Defining Function (Def)
def convert_unit(meter):
    centimeter = meter * 100
    return centimeter  # Throw the result out

# Using the Function
length = 2.5
result = convert_unit(length)

# f-string (Modern Formatting): Embedding variable into text
print(f"Converted value: {result} cm")
```

```
Converted value: 250.0 cm
```

G) Libraries – Ready-Made Tools Python is like a "naked" smartphone. Libraries are the applications like "Instagram, WhatsApp" that we install on it. We do not reinvent the wheel; we use libraries.

In [7]:
```python
import math
print(math.sqrt(16))  # Square root -> 4.0

# Giants we will see later:
# import pandas (For Excel tasks)
# import numpy (For Math tasks)
```

```
4.0
```