

R | Basic Text Analysis

Presented by Aviv Lo

Table of Contents

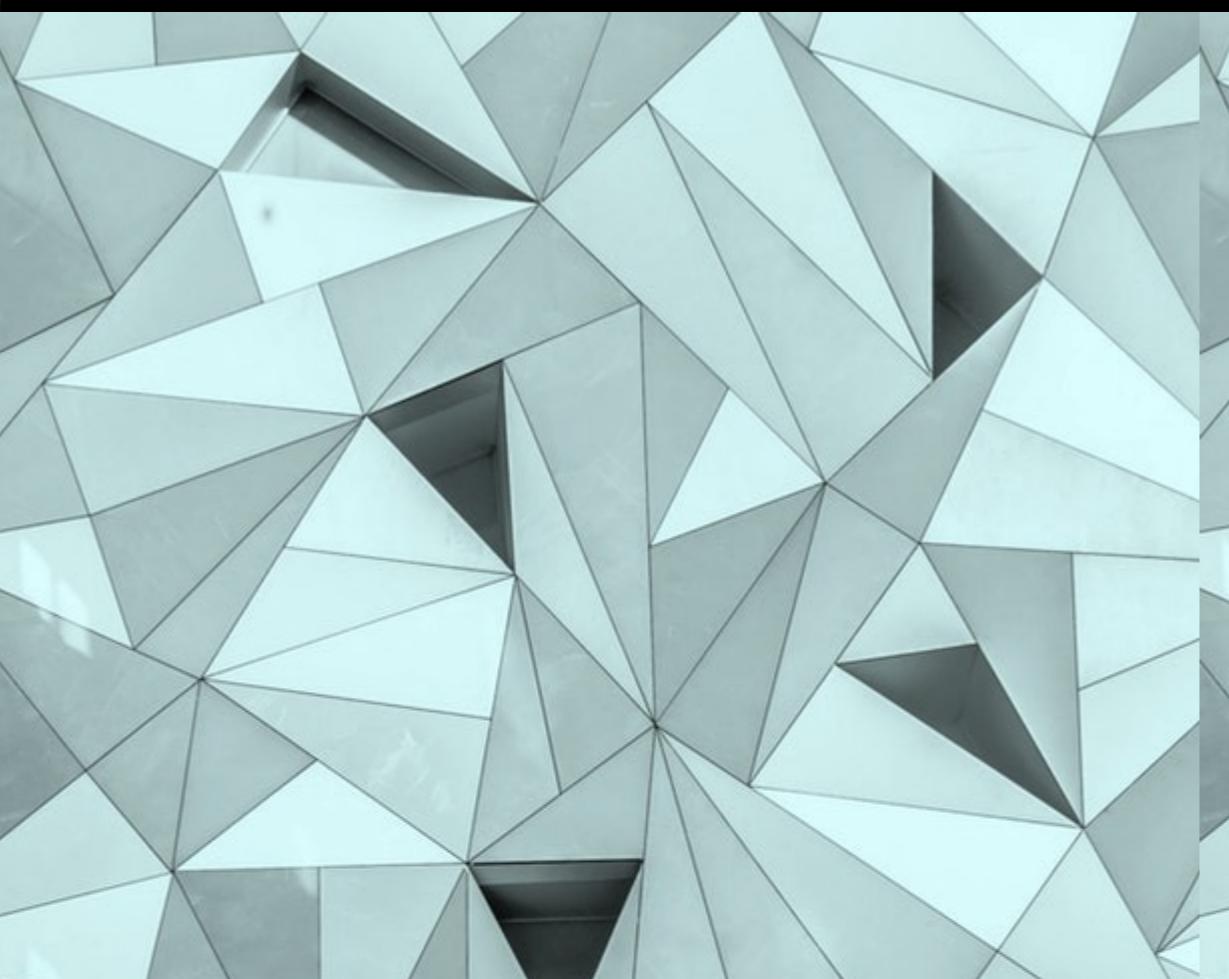
Topic 1

R vs. Python



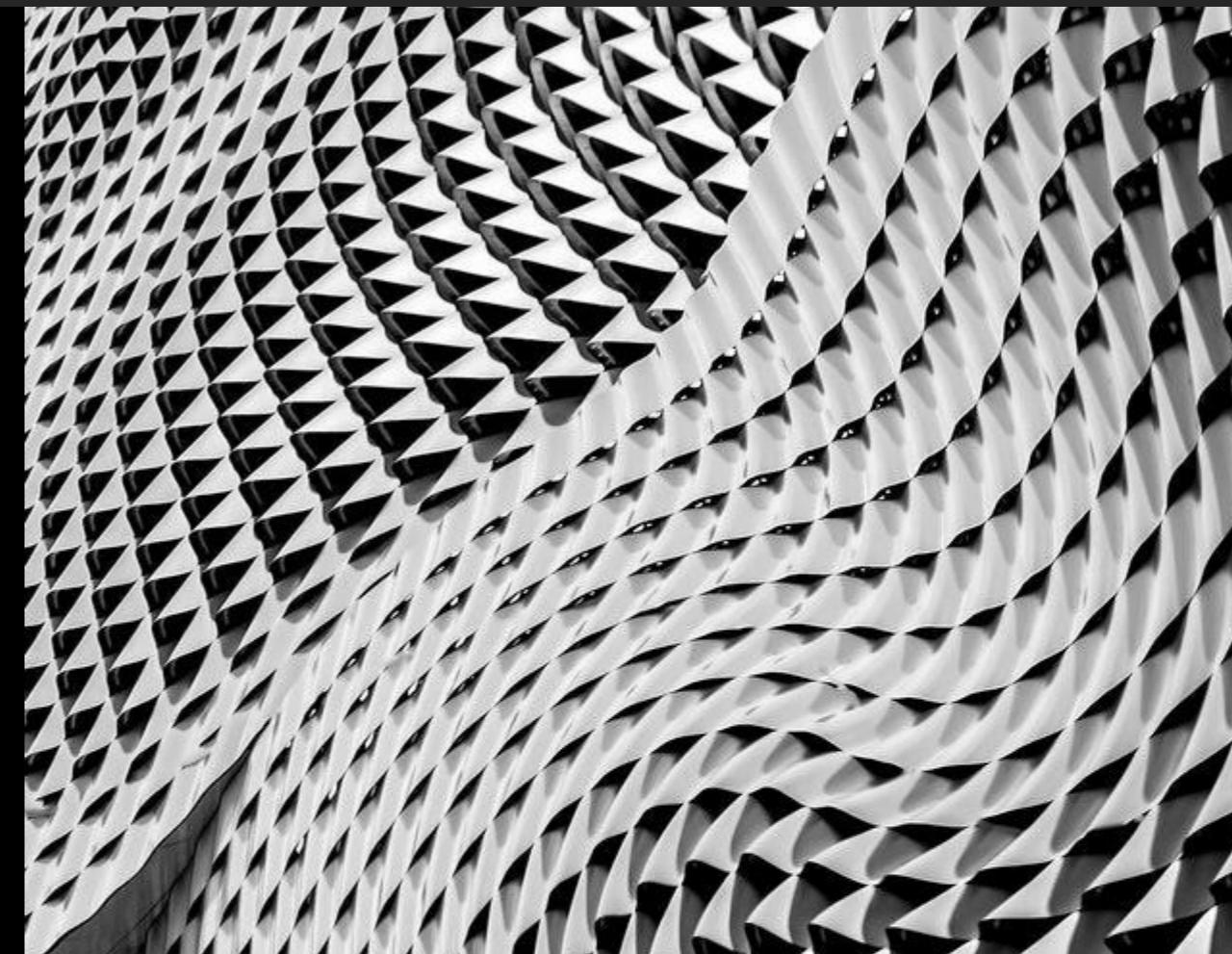
Topic 2

AI/ML/NLP Explained



Topic 3

Text Analysis



Topic 4

Data Gathering & Extraction



1

R vs. Python



1. Developed by Ross Ihaka and Robert Gentleman in 1993
2. Hence the name “R”
3. Maintained by The R Foundation for Statistical Computing
4. Difficult for beginners + Steep learning curve
5. Rich and powerful visualization libraries
6. Support matrix / vectorized operations
7. It’s born for stats

VIM window showing R code and object browser:

```
example.R (/tmp) - VIM
```

```
numbers <- 1:3
words <- c("word1", "word2", "word3")
categories <- as.factor(words)
dtfrm <- data.frame(numbers, words)

attr(numbers, "label")     <- "A numeric vector"
attr(words, "label")      <- "A character vector"
attr(categories, "label") <- "A factor vector"

list1 <- list(dtfrm = dtfrm, y = numbers)
list2 <- list(list1 = list1, abc = words)
list2$name with space <- 1:10
list2$2 <- c("one", "two")
list3 <- list(abc = categories, list1 = list1)
rm(list1)
```

.GlobalEnv | Libraries

```
categories A factor vector
dtfrm
└── numbers
└── words
list2
└── list1
    └── dtfrm
        ├── numbers
        └── words
    └── y A numeric vector
└── abc A character vector
└── name with space
2
```

Object_Browser 8,1 Top

```
> list1 <- list(dtfrm = dtfrm, y = numbers)
> list2 <- list(list1 = list1, abc = words)
> list2$name with space <- 1:10
> list2$2 <- c("one", "two")
> list3 <- list(abc = categories, list1 = list1)
> rm(list1)
> source('/home/jakson/src/Vim-R-plugin/r-plugin/vimbrowser.R') ; .vim.browser()
>
```



-
1. Developed by Guido van Rossum in 1991
 2. Naming was inspired by Monty Python
 3. Successor to the ABC language
 4. Maintained by the Python Software Foundation
 5. Easy for beginners + Smooth learning curve
 6. Good libraries for creating graphs
 7. Support matrix / vectorized operations via external libs

```
concept_recaps.py > ...
1 # Libraries
2 import pandas as pd
3
4 """
5 Python Native Data Structures
6 """
7 # List
8 # Create a list
9 list = ["Jan", "Feb", "Mar", "Apr", "May"]
10 list.append("Jun")
11 print(list)
12
13 # Tuple
14 # Create a tuple
15 tuple = ("Jan", "Feb", "Mar", "Apr", "May")
16 # tuple.append("Jun") # This will throw an error
17 print(tuple)
18
19 # Dictionary
20 # Create a dictionary
21 dictionary = {
22     "Jan": 31,
23     "Feb": 28,
24     "Mar": 31,
25     "Apr": 30,
26     "May": 31
27 }
28 dictionary["Jun"] = 30
29 print(dictionary)
30
```

R vs. Python

	Python	R
General	Python is a general-purpose programming language for data analysis and scientific computing.	R is a functional programming environment and language for statistical computing and graphics.
Objective	Data Science, Web Development, Embedded Systems	Data Science & Statistical Modeling
IDE	iPython, Pycharm, Jupyter Notebook, Spyder	Rstudio, R GUI, R KWARD
Data Collection	Supports CSV files, SQL, JSON, and webscraping with BeautifulSoup.	Can also import csv files with built-in <code>readr</code> library. R's library <code>RCurl</code> provides a simple way to make API requests, similar to Python's <code>requests</code> package.
Data Analysis	Organize dataframes with Pandas filtering, sorting. Python takes a more streamlined approach for data science projects.	Complex data visualization tools make the exploratory data analysis (EDA) process much more complex than Python.
Essential Packages & Libraries	Numpy, Pandas, matplotlib, scipy, scikit-learn, TensorFlow	caret, stringr, ggplot2, knitr, tidyverse, markdown, shiny, forcats, haven
Database Handling Capacity	Can easily handle large data because there are less constraints for memory usage	R computes everything in memory, so its capabilities are limited by RAM size. A major downfall of R is the inability to handle massive amounts of data
Data Visualization	Despite the capabilities of data visualization tools like Matplotlib and Seaborn, Python fails to measure up to data visualization features of R.	Developed by and for statisticians, R has complex data visualization features.
Syntax	The 'zen of python' is that there's a proper way to write code.	R doesn't have this set of rules. Also indexing starts at 1, which can be considered unconventional for general programmers.
Learning Curve	Simple and readable code structure makes it easier for beginners to learn. It also allows for object-oriented programming. It also offers a wide range of data structures that you wouldn't expect from a general-purpose language.	R's functional syntax isn't easy for beginners, but not too challenging for those well versed in programming. It also offers a few data structures, but fails to handle large amounts of data.

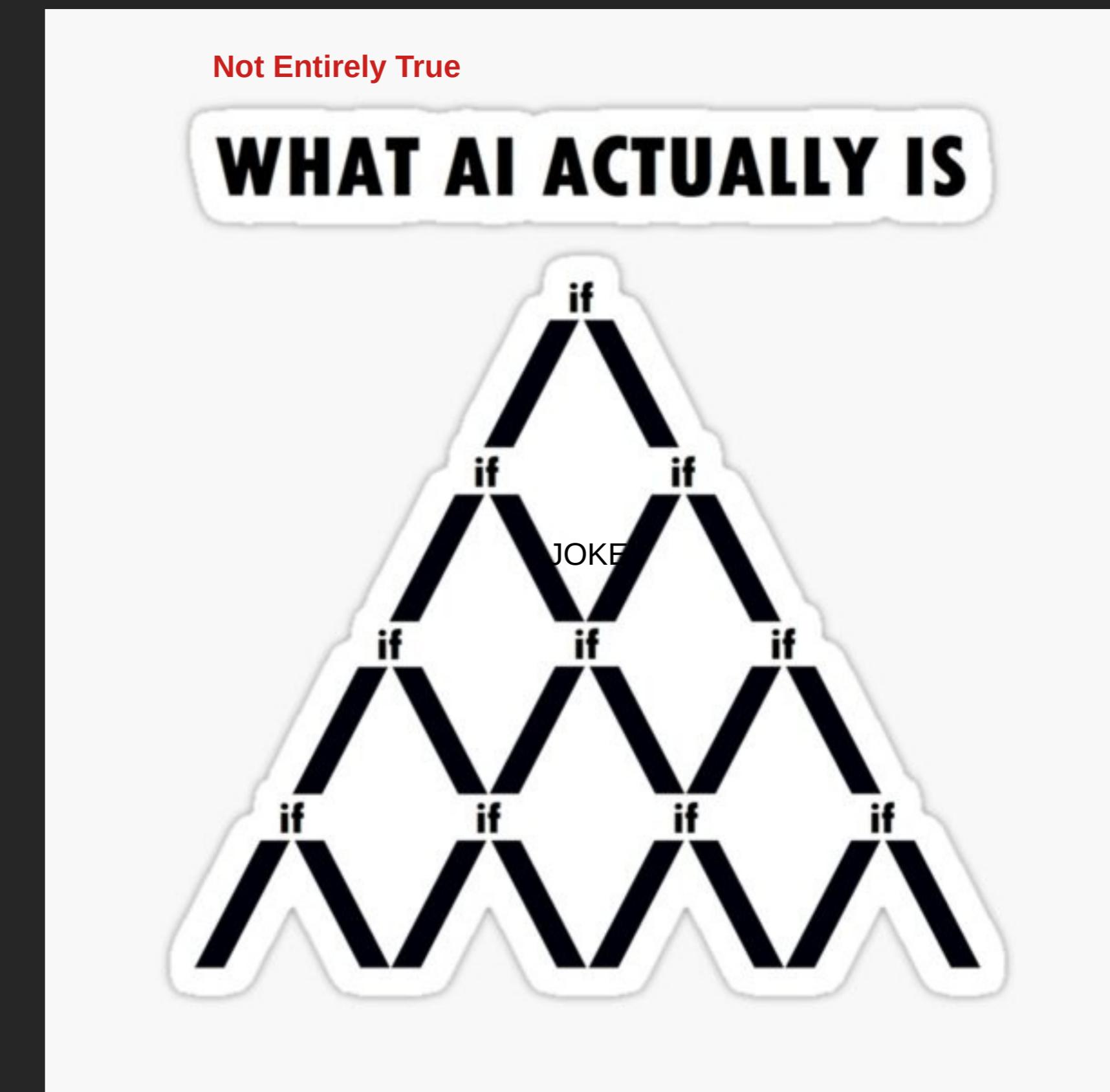
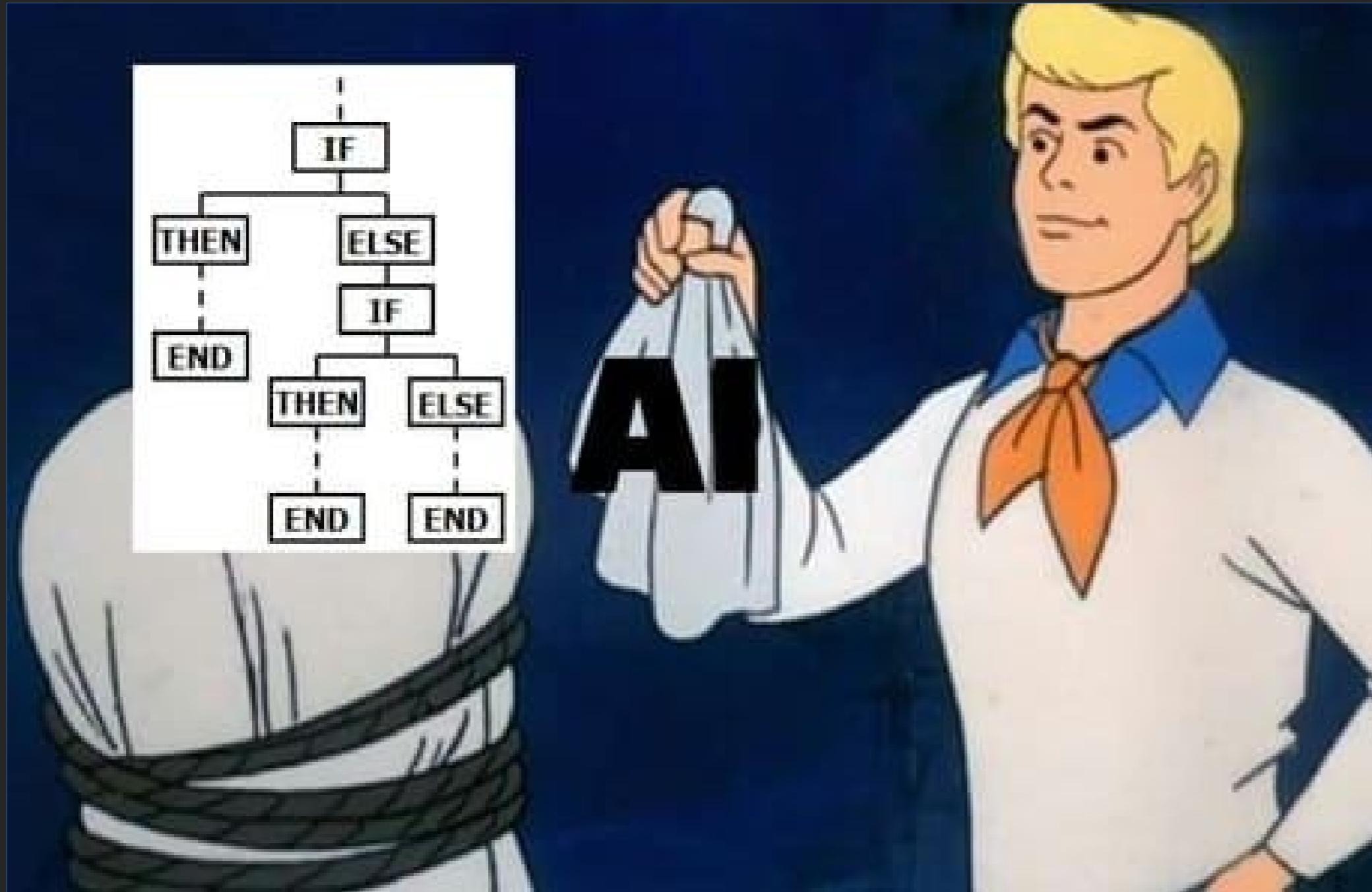
2

AI/ML/NLP Explained

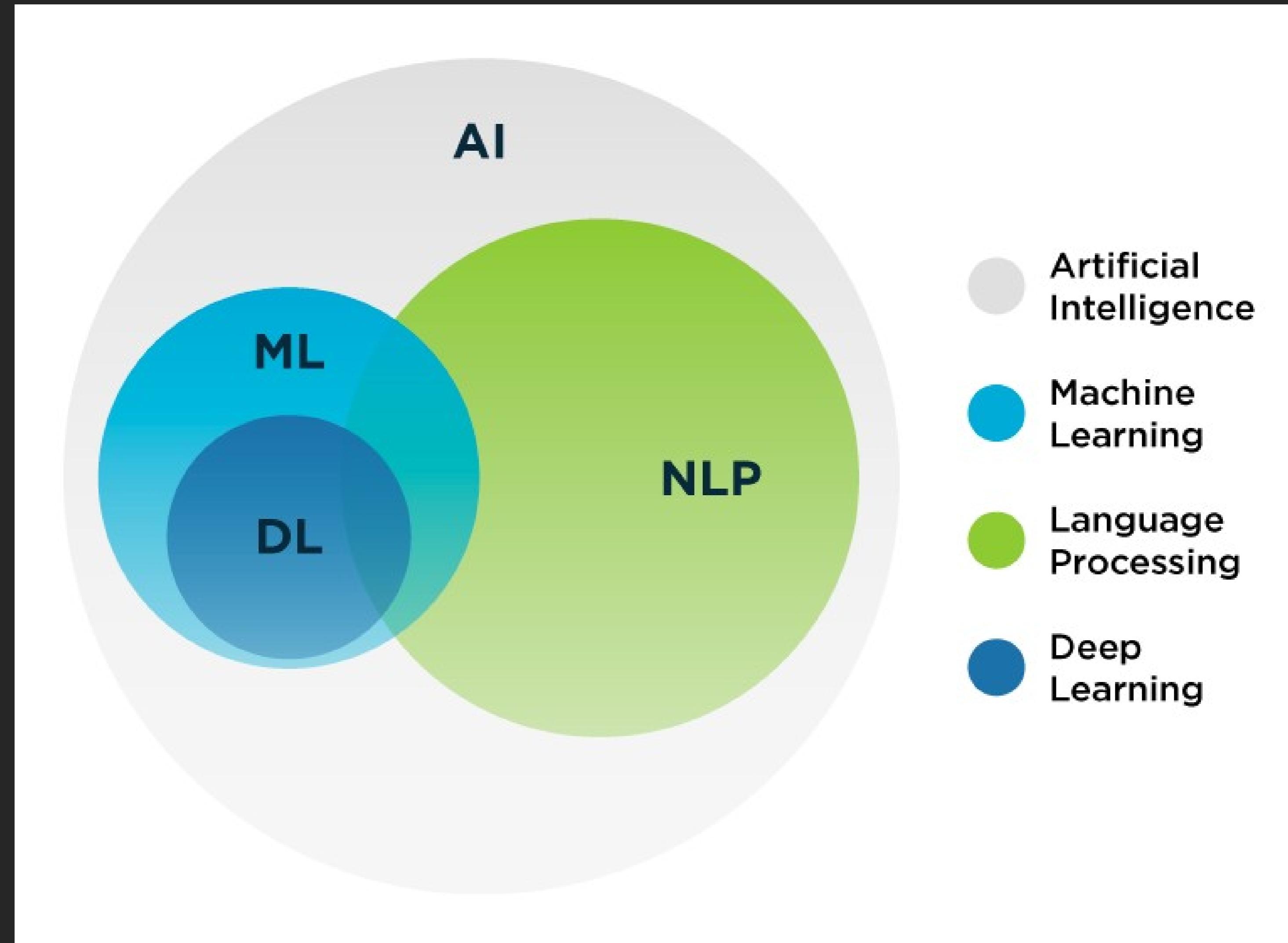
AI?



Artificial Intelligence

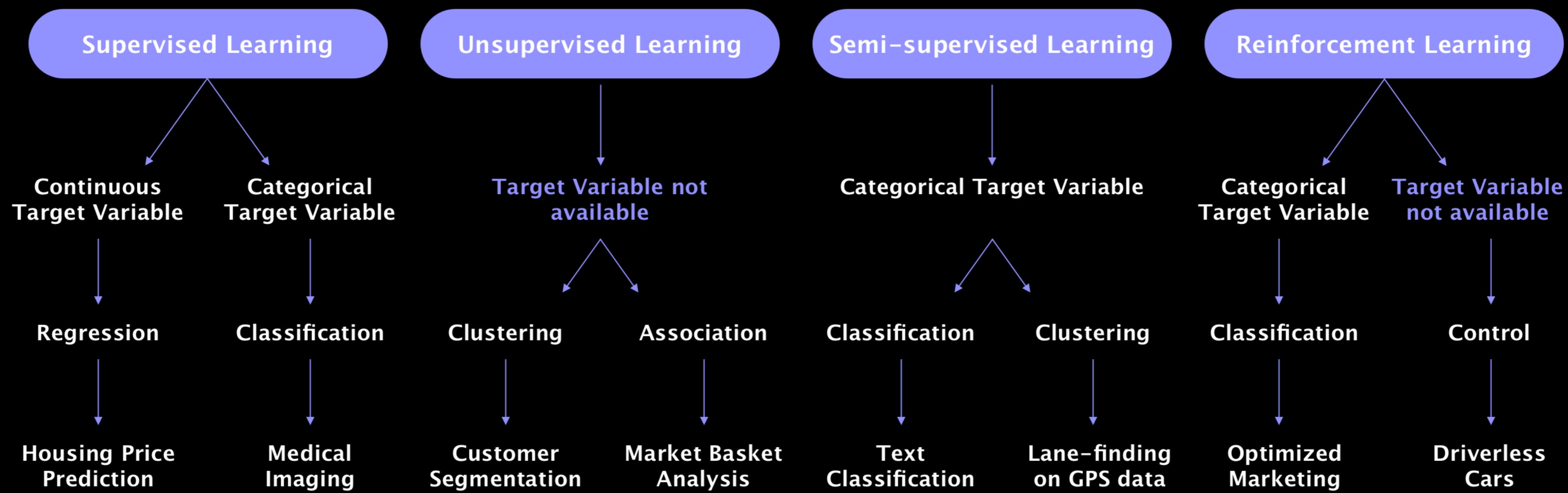


AI Domains

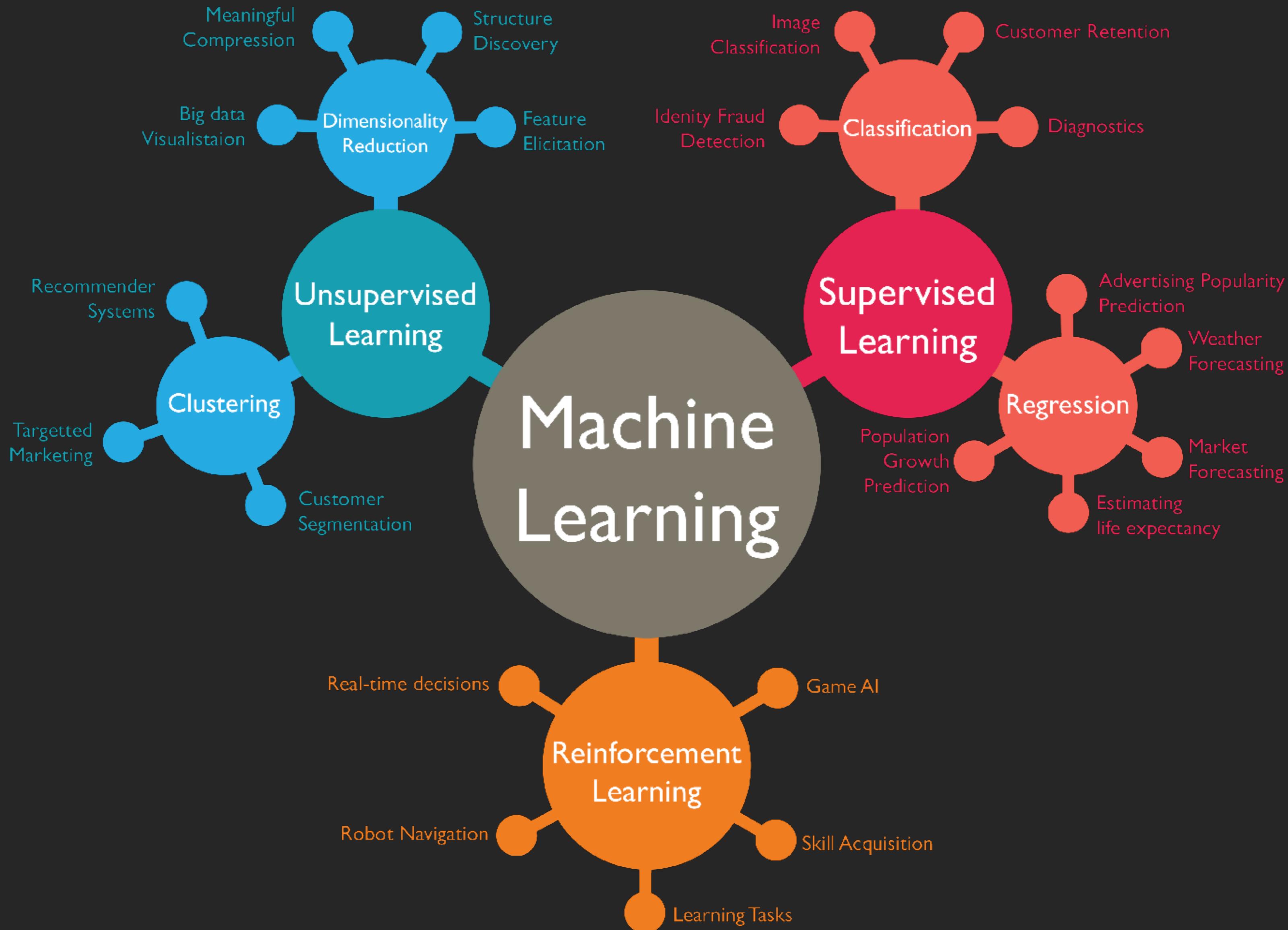


Machine Learning Family Tree

Machine Learning Types



Machine Learning Family Tree



Deep Learning: Fundamental Architecture

Feature / Architecture	ANN/MLP	RNN	CNN	• Transformer
What is it?	Basic neural network	Neural network with “memory”	Neural network for grid-like data	Neural network with self-attention mechanism
Advantages	Typically easier to train	Captures sequential information and preserves dependency	Captures spatial features and relationships	Captures long-range dependencies. Parallel processing of sequences.
Challenges	Overfit easily even in simple networks	“Error” amplification (gradual or exponential)	Requires large amount of data	Requires large amount of data / very computationally intensive
Typical Use	Classification Non-linear regressions	Time series Texts and audios	Image Recognition Video Processing	LLMs & Generative AIs

Recent Example: ChatGPT

Version	Parameters Count	Dataset
GPT-1	0.12 Billion	BookCorpus
GPT-2	1.5 Billion	WebText
GPT-3	175 Billion	CommonCrawl, WebText, English Wikipedia, Books 1 and Books 2 Corpora
GPT-4	1.76 Trillion	All of the above + YouTube + Twitter + Textbooks (fine tuned by Scale AI)

Tech Steamroller



3

Text Analysis

Goals

KEY DIFFERENCE BETWEEN TEXT ANALYSIS & NLP? THEIR GOALS.

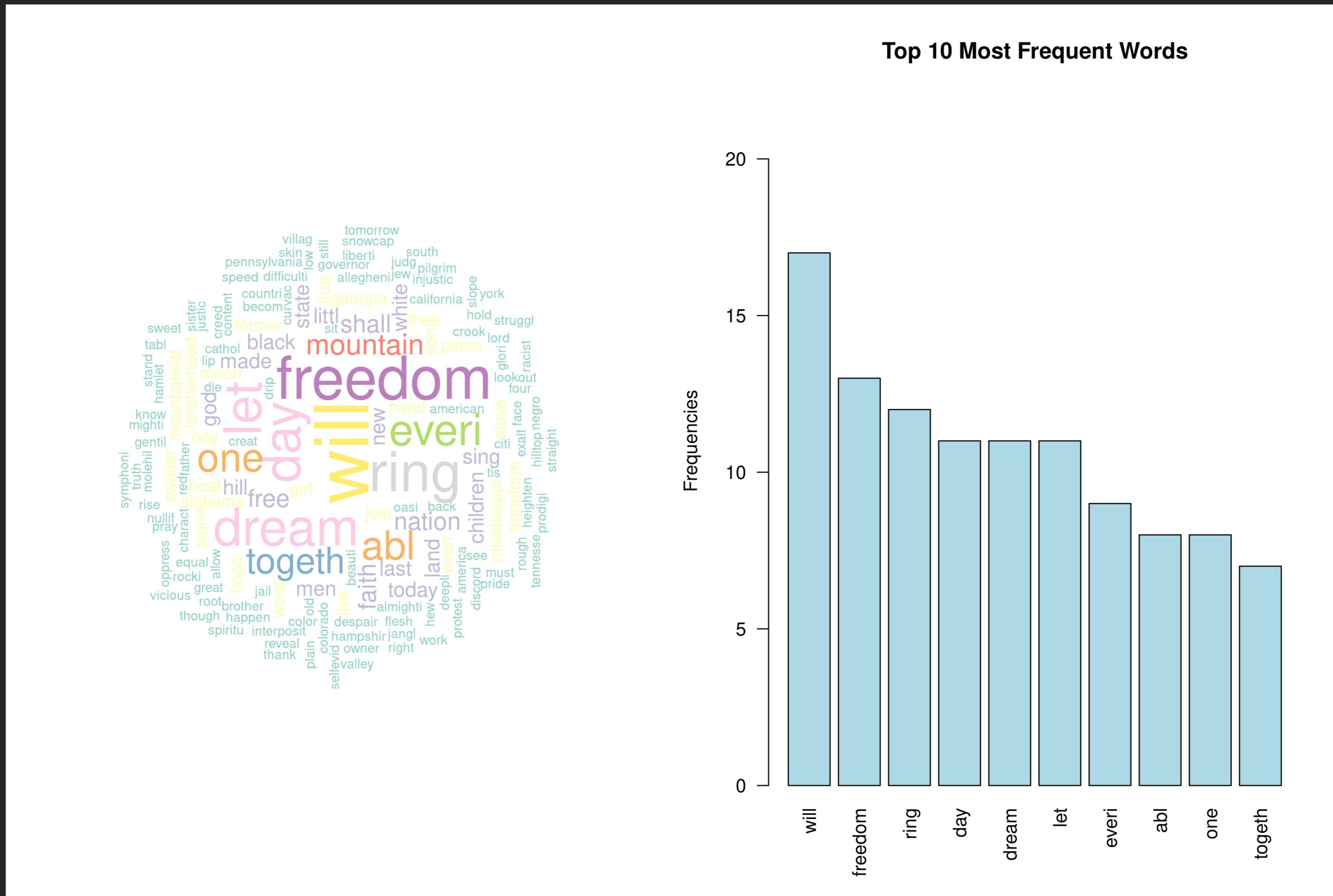
TEXT ANALYSIS GOAL

Derive insights solely from the text itself, without consideration of the semantics.

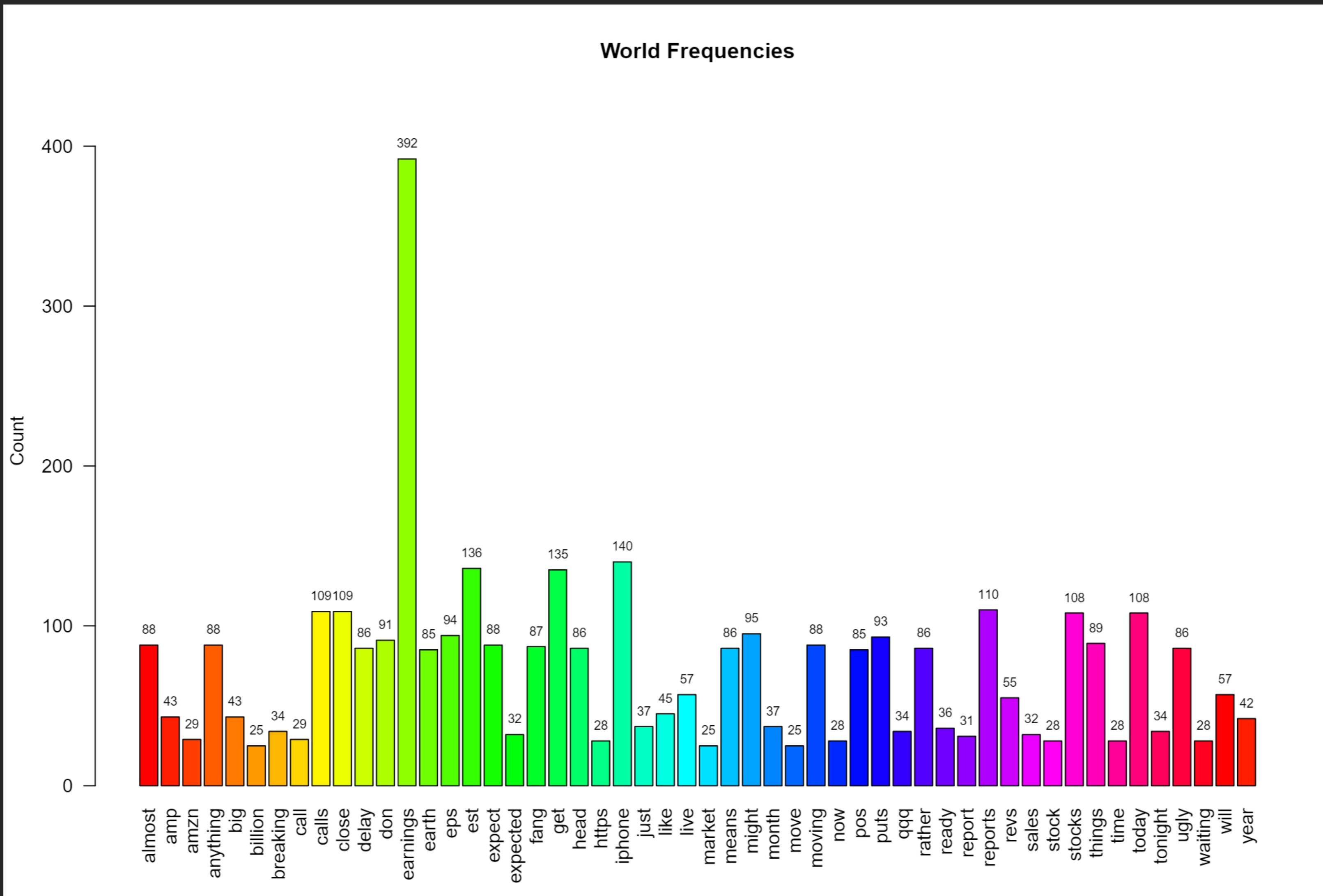
NLP GOAL

Understand the linguistic use and context behind text, with consideration of semantics and grammatical structures.

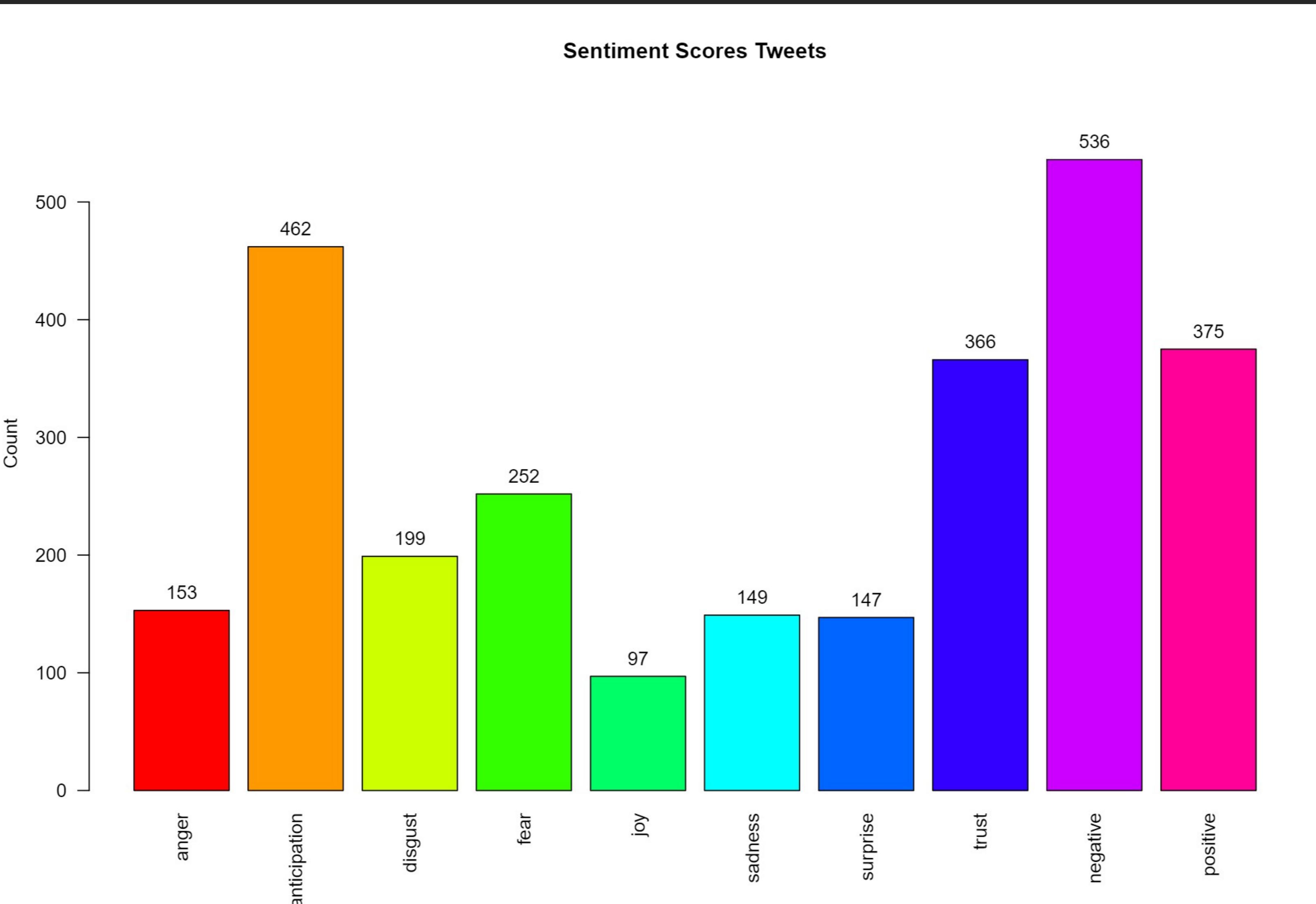
Example: Word Cloud & Word Frequency



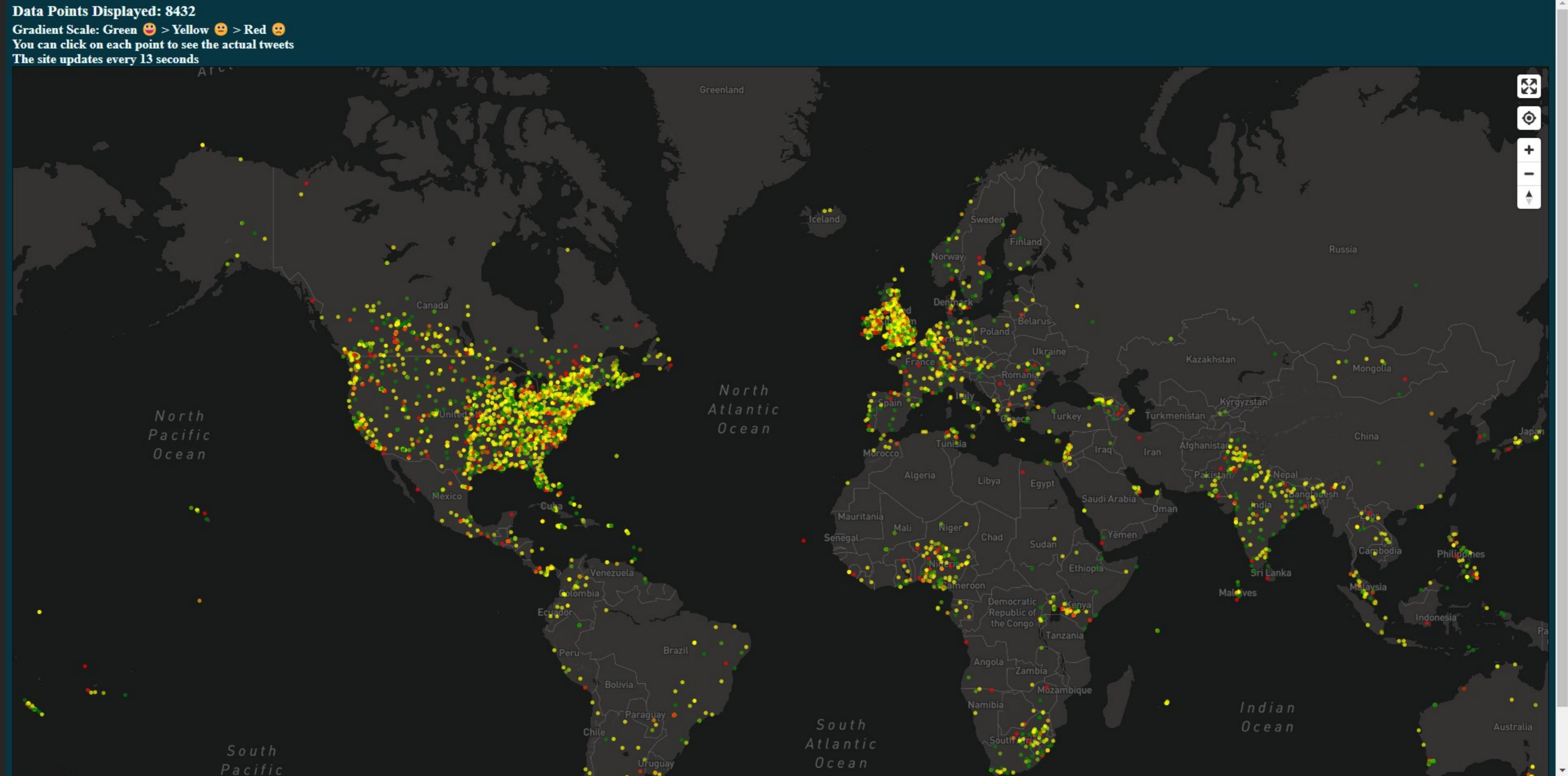
Example: Word Frequency



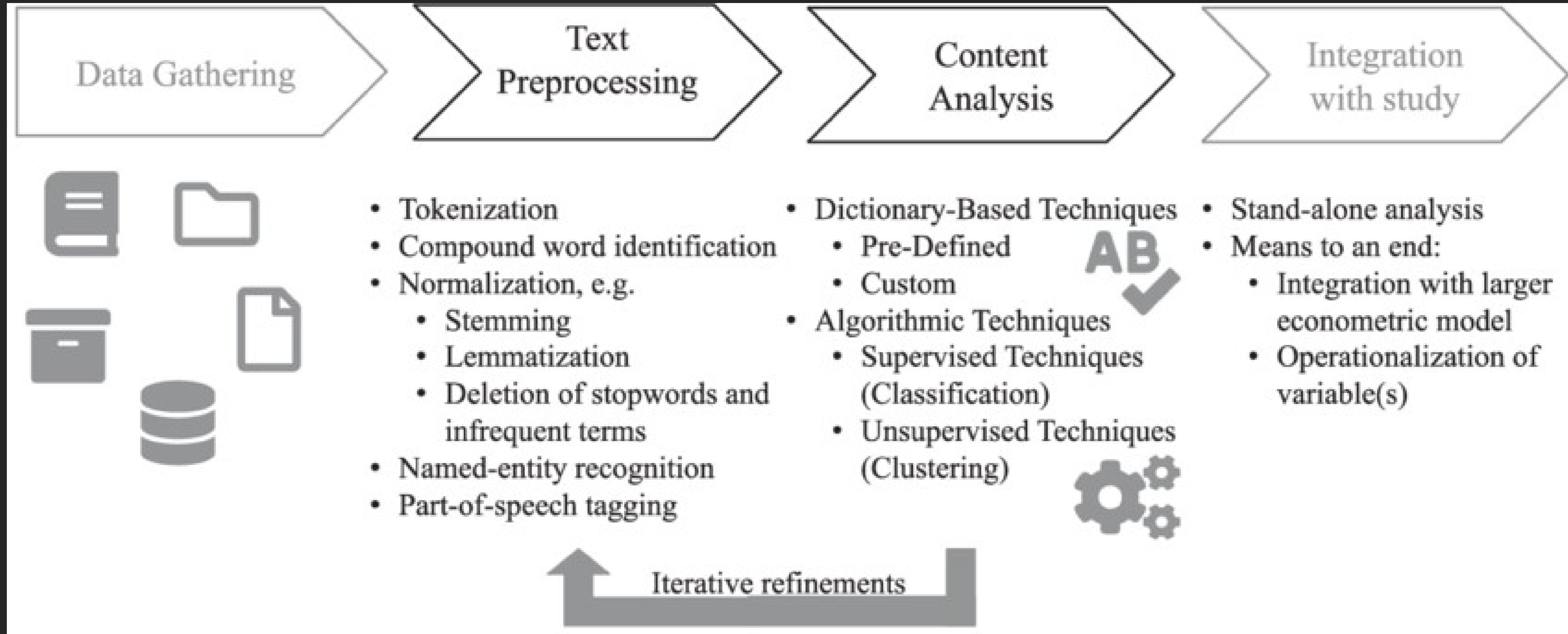
Example: Tweet Sentiment



Examples: Covid Sentiment Map



Process & Techniques



Data Gathering and Extraction

1. [Google Dataset Search](#)
2. [Kaggle](#)
3. [GitHub](#)
4. [Government Sources](#)
5. The World Wide Web (scraping)

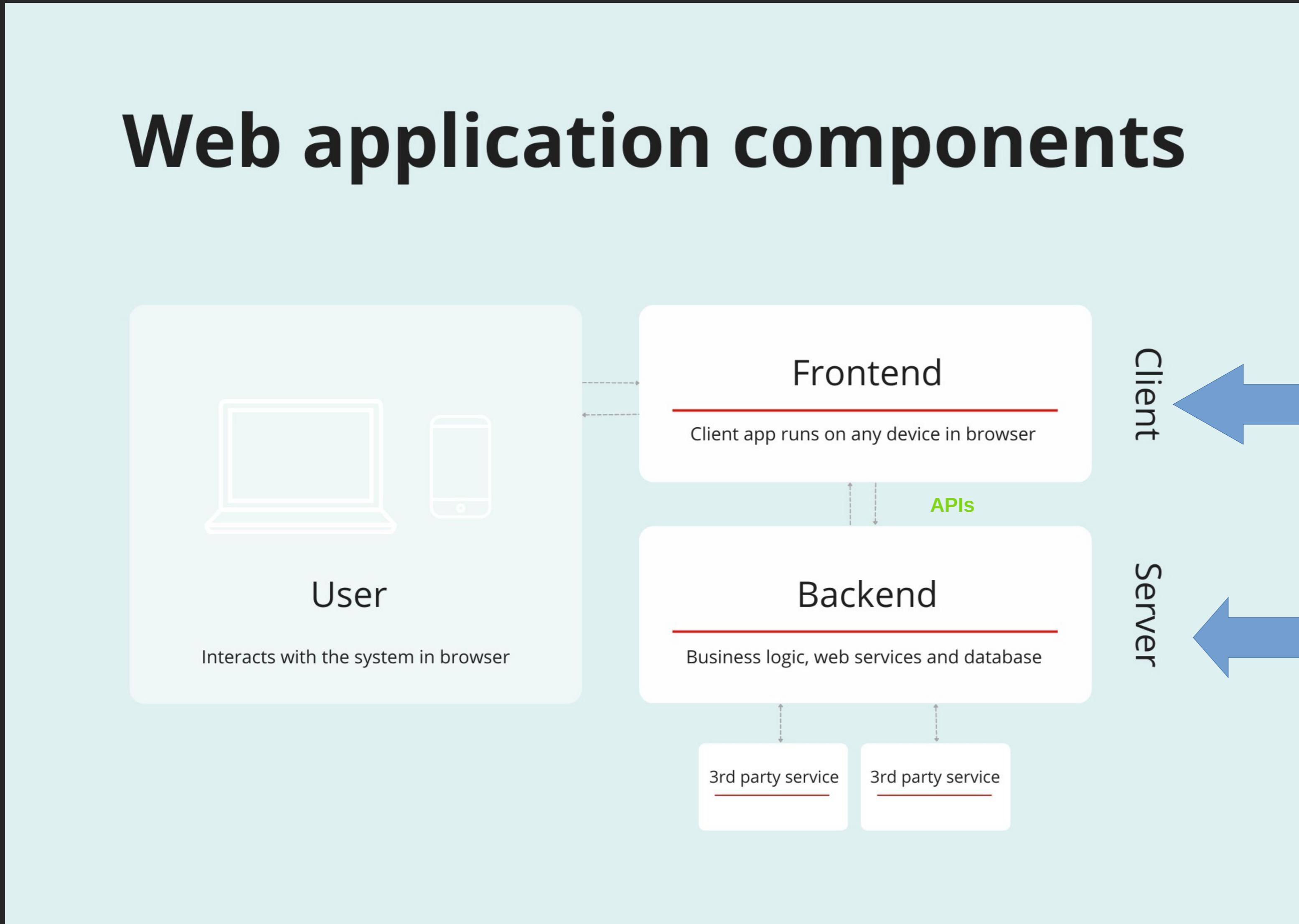


4

Data Gathering & Extraction

Data Extraction - Web Application

Web application components



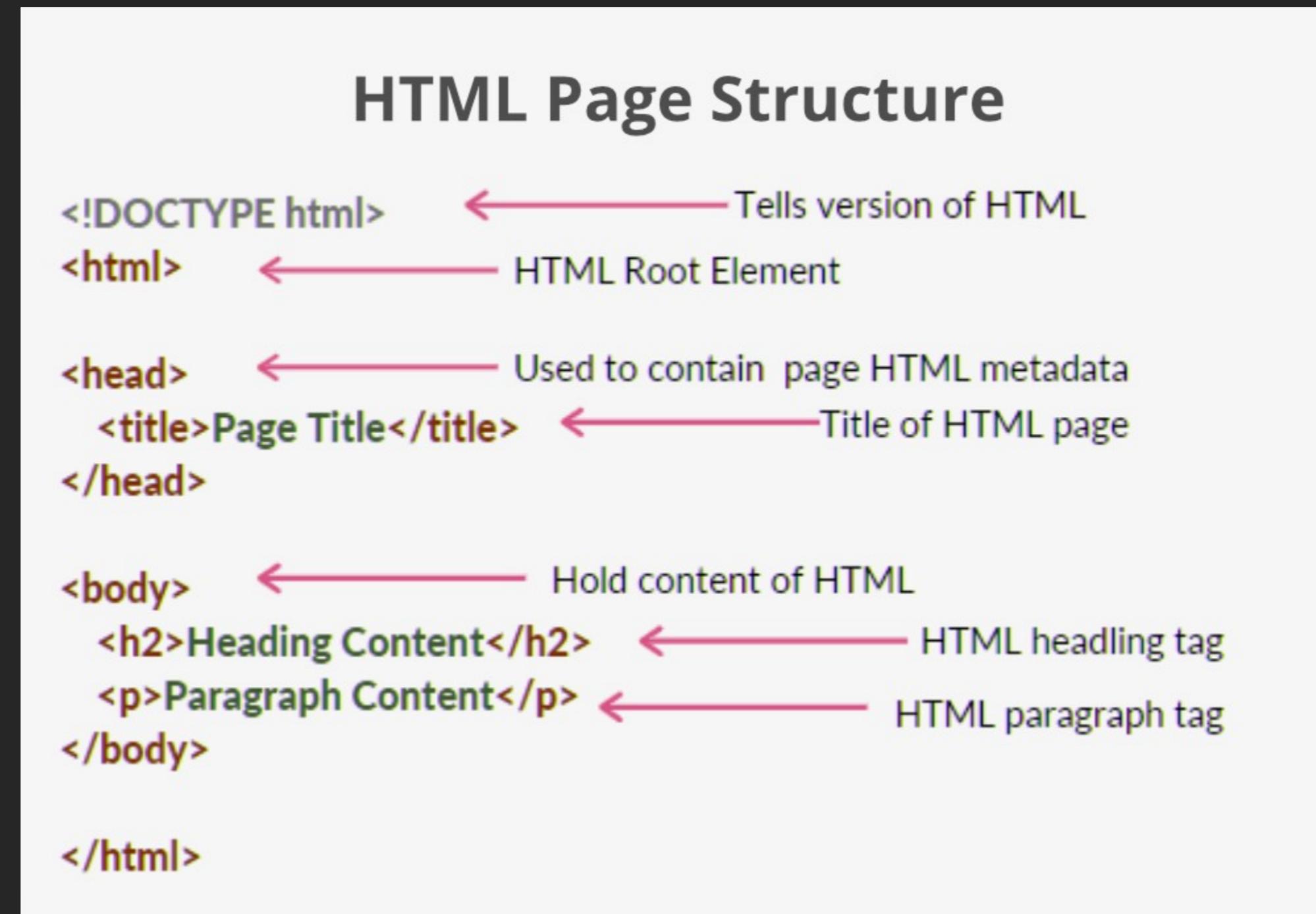
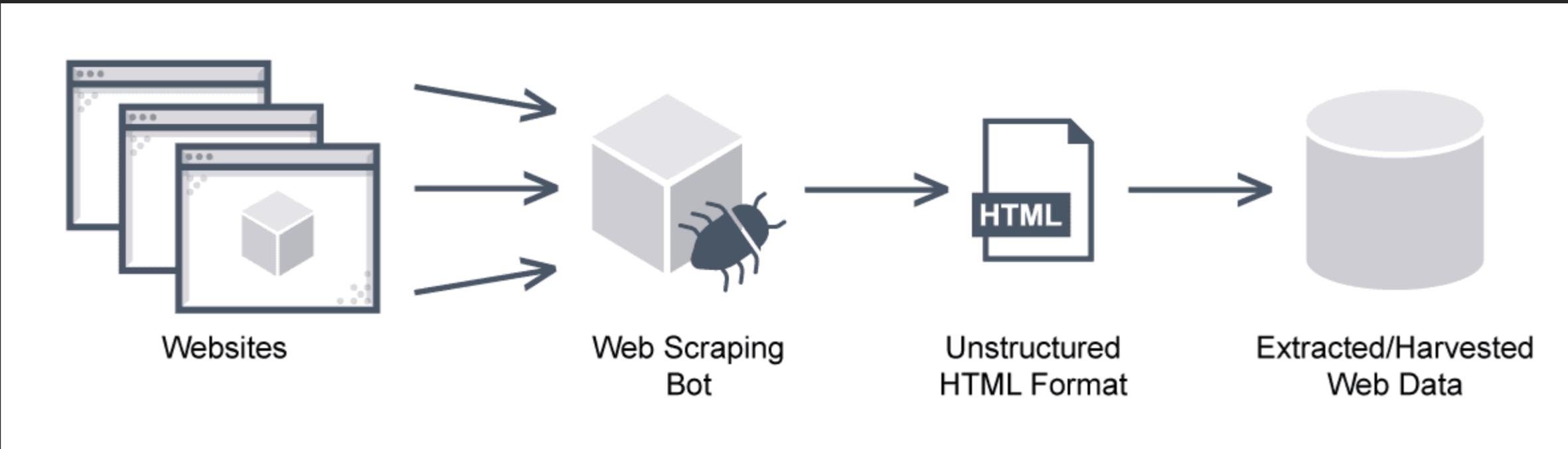
**DOM
(Document Object Model)
Parsing**

**API
(Application Programming Interface)
Scraping**

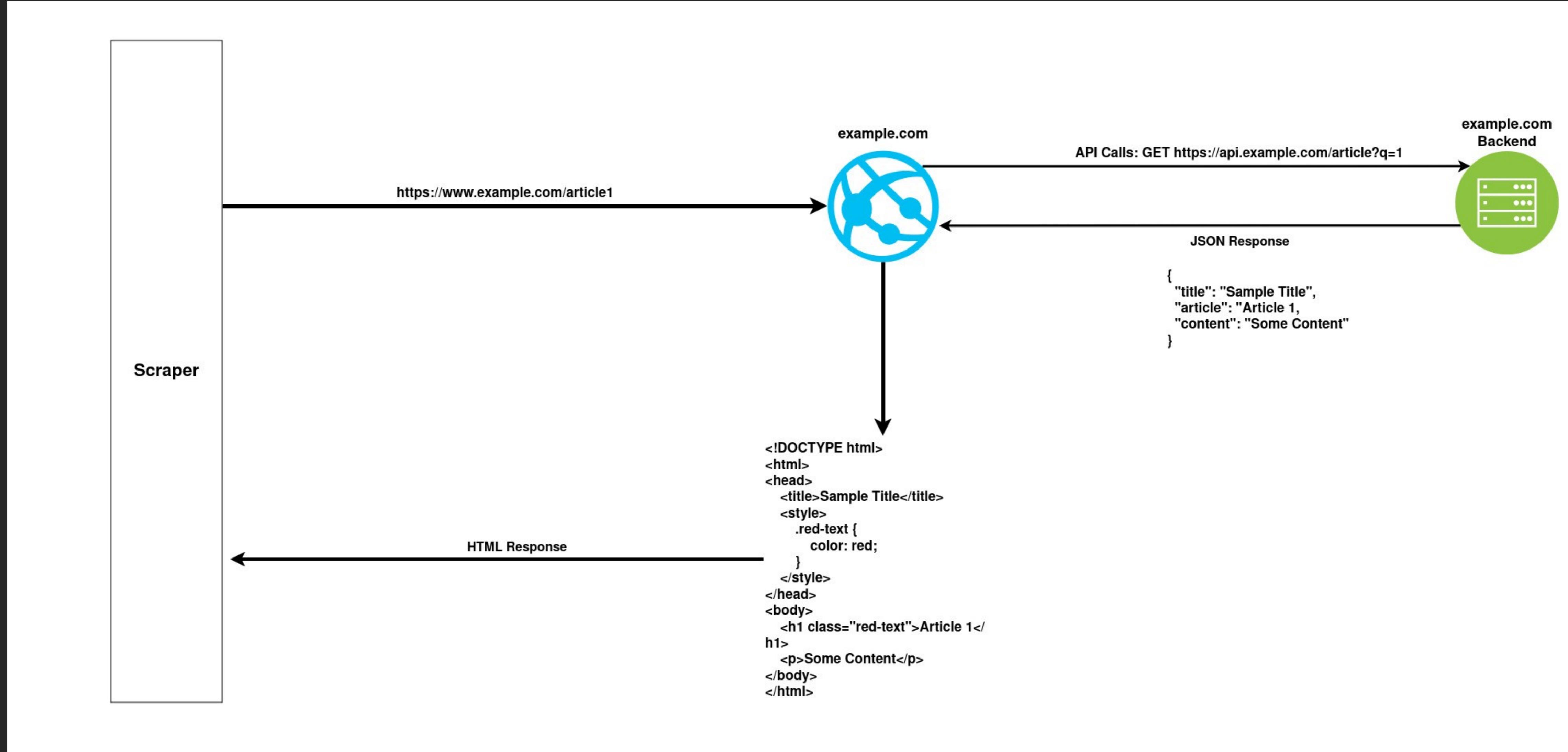
Data Extraction - Web Application



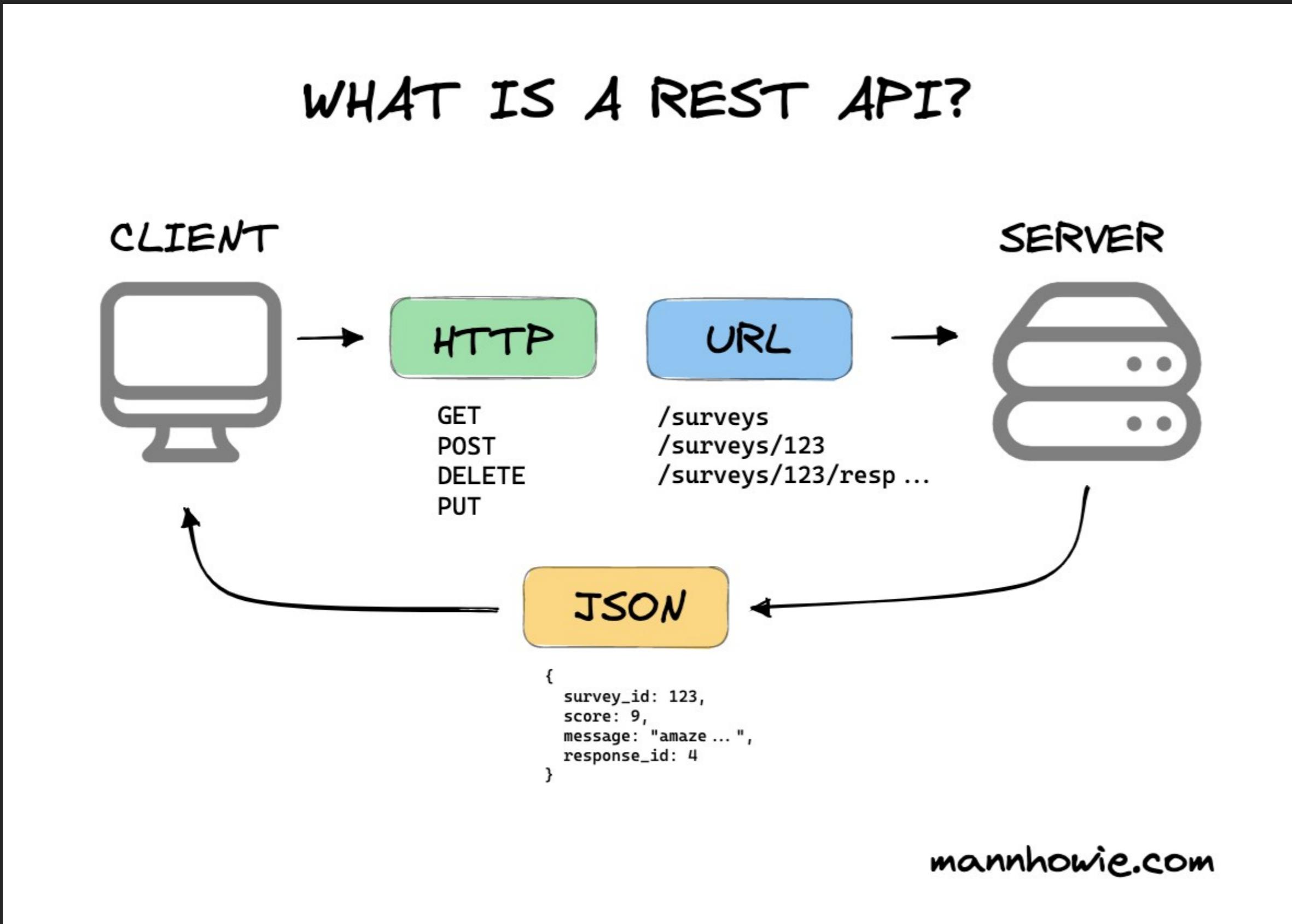
Web Scraping - DOM Parsing



Web Scraping - DOM Parsing

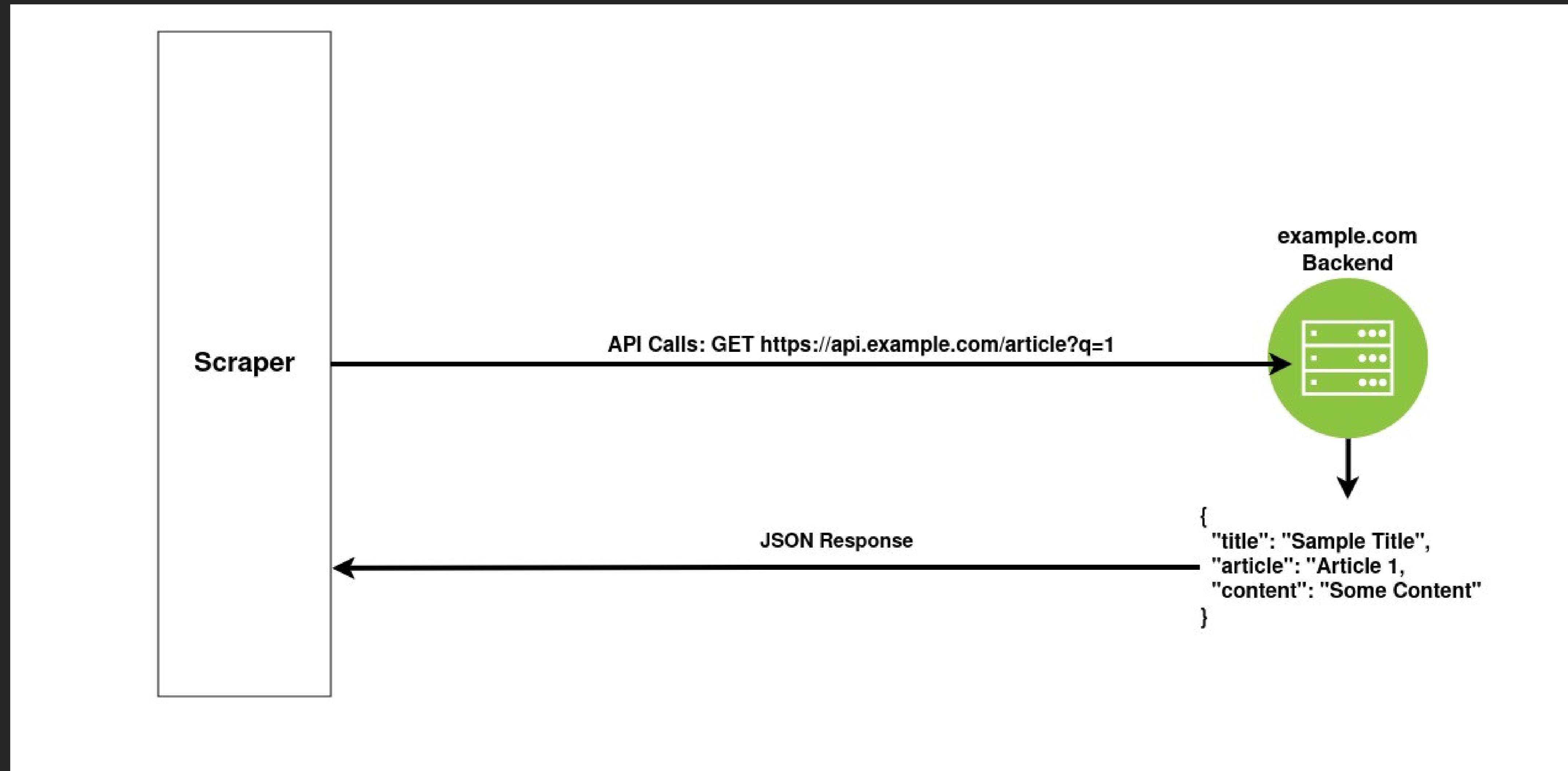


Web Scraping - API Scraping



```
{  
  "DocumentType": 1,  
  "No.": "S-ORD101001",  
  "SellToCustNo": "10000",  
  "PostingDate": "2023-04-02",  
  "Lines": [  
    {  
      "LineNo": 10000,  
      "Type": 2,  
      "No": "1996-S",  
      "Quantity": 12,  
      "UnitPrice": 1397.3  
    },  
    {  
      "LineNo": 20000,  
      "Type": 2,  
      "No": "1900-S",  
      "Quantity": 4,  
      "UnitPrice": 192.8  
    }  
  ]  
}
```

Web Scraping - API Scraping



Summary

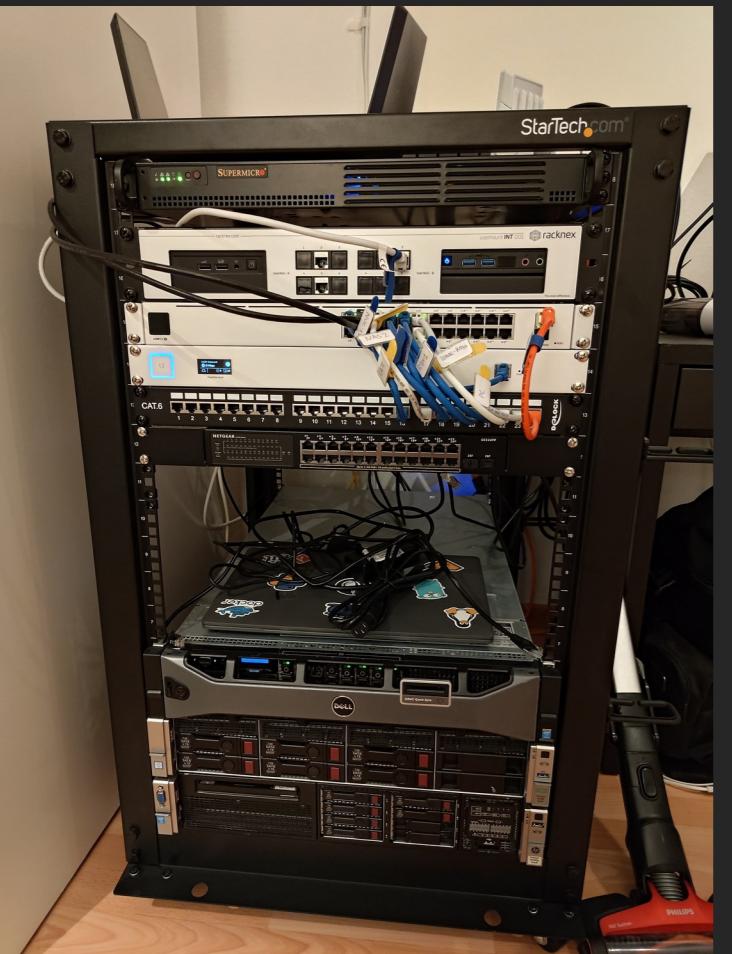
	DOM Parsing	API Scraping
Format	Unstructured	Structured
Processing Cost	High Results mixed with irrelevant information	Low
Stability	Prone to changes	Relatively static
Availability	Always	Close sourced or paid



EXTRACT
DATA FROM
THE RAW HTML

PARSE DATA
FROM THE
SITE'S REST API

Available Tools

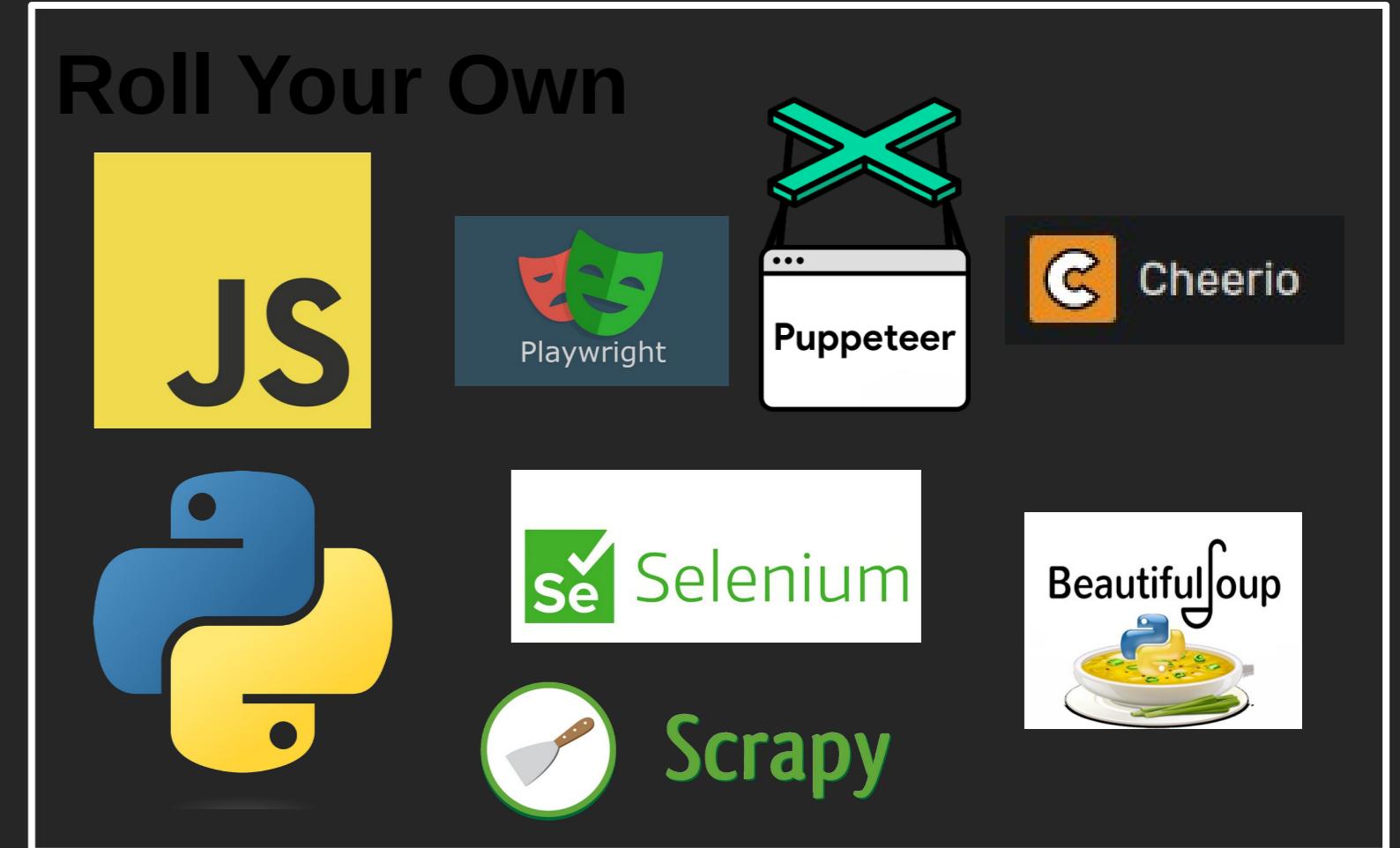


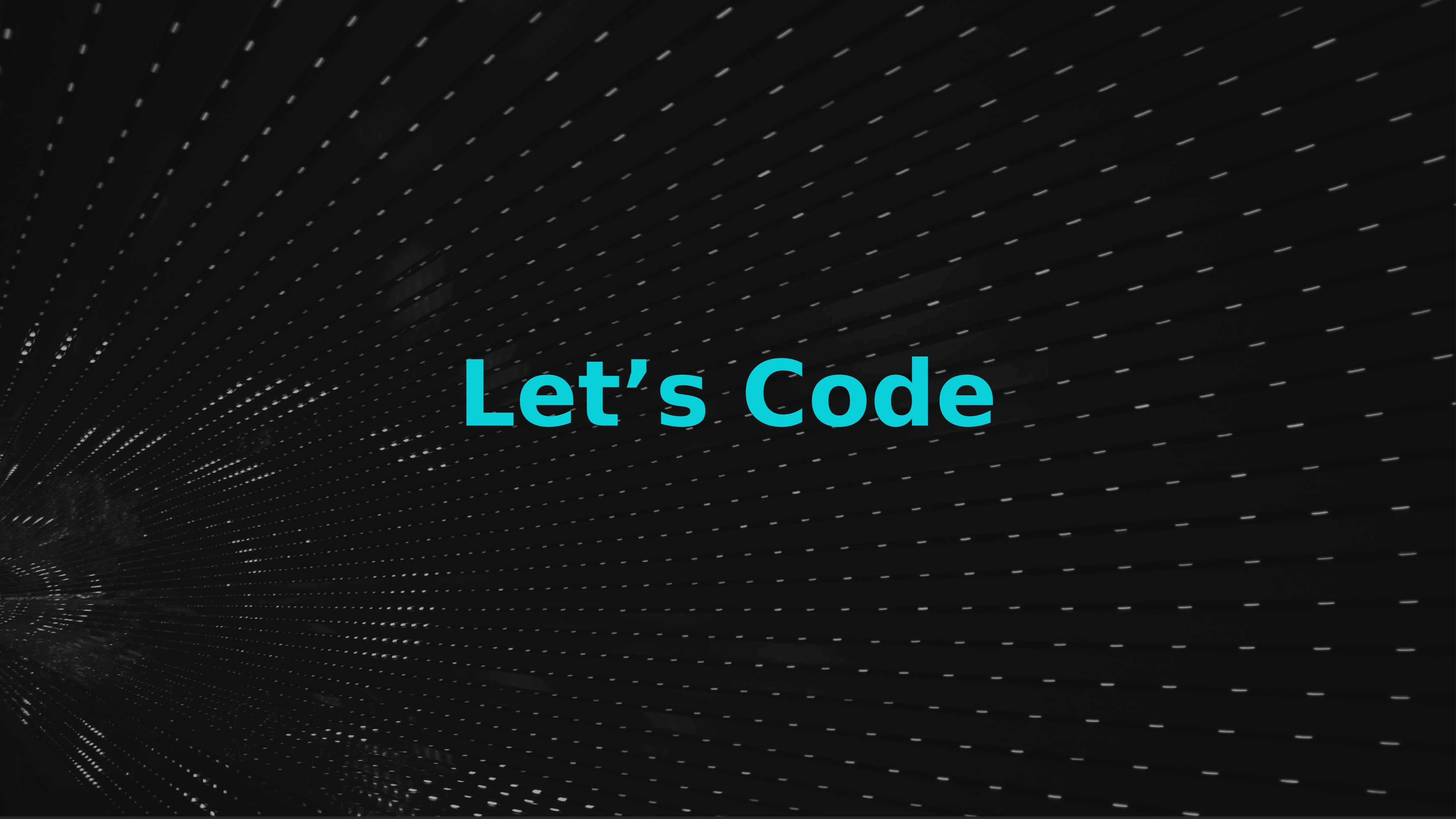
Custom Tooling

<https://github.com/search?q=scraping&type=repositories>

<https://github.com/topics/web-scraper>

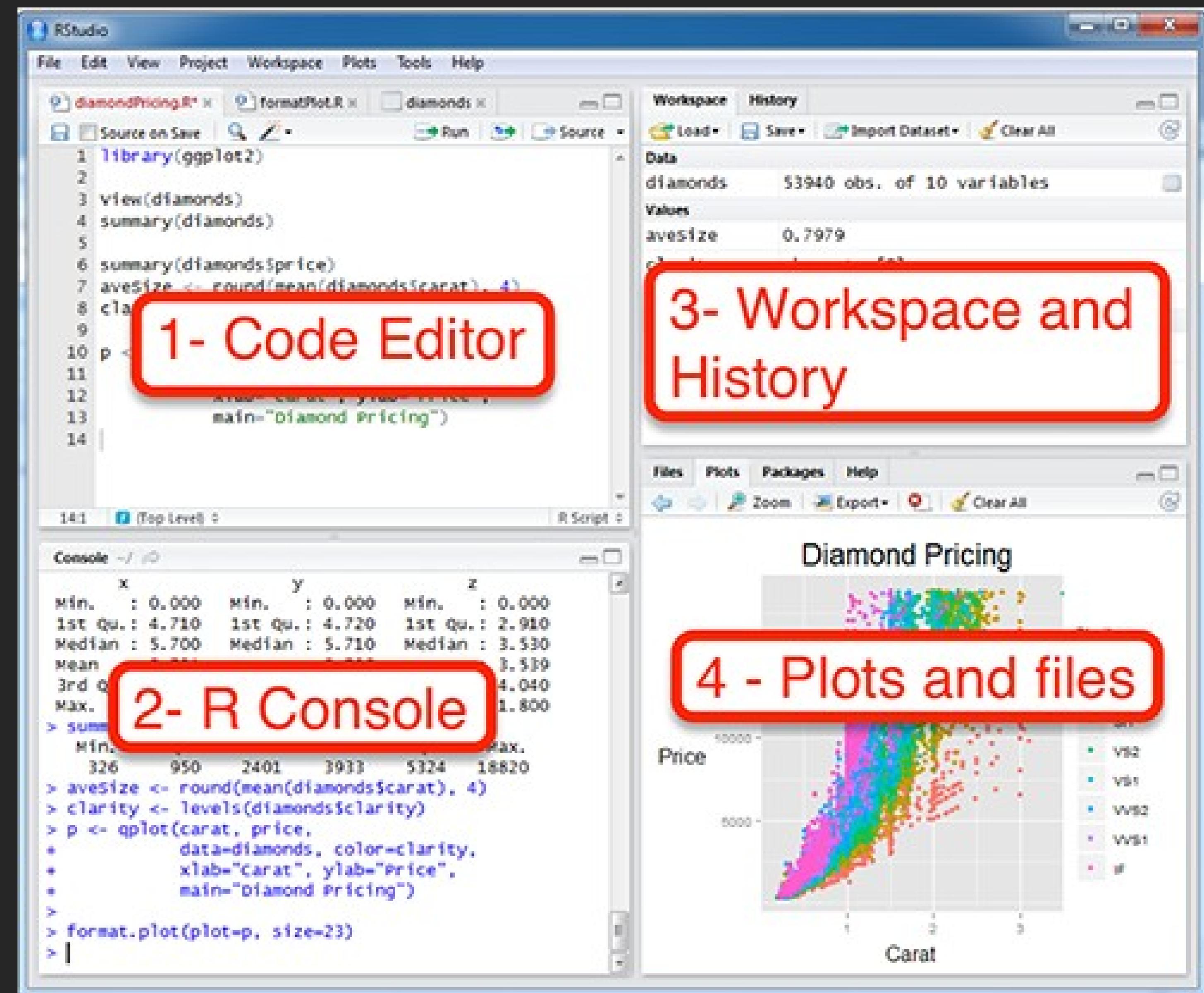
<https://scraper.algo7.tools>





Let's Code

RStudio



Thank You