HKUSPACE
香港大學專業進修學院
HKU School of Professional and Continuing Education

Business Process Automation with VBA and Python

Mr. Eddie Chow / 24 January 2026

HKUSPACE
香港大學專業進修學院
HKU School of Professional and Continuing Education

Table Of Contents

Intended Learning Outcomes

1. explain the basic concepts and principles of web scraping, including the ethical considerations and legal implications involved in data extraction from websites.
2. demonstrate the ability to utilize Python libraries such as Beautiful Soup and Scrapy to efficiently extract, parse, and clean data from various web sources.
3. learn to visualize the scraped data using libraries like Matplotlib or Seaborn, enabling them to create informative graphs and charts that effectively communicate insights derived from the data.
4. apply data analysis techniques to interpret the visualized data, drawing conclusions that can inform decision-making processes in real-world scenarios, such as market analysis or competitive intelligence.

Agenda

1. Data Collection - Web Crawler
2. ETL Data Processing
3. Machine Learning Framework
4. Introduction to Data Visualization
5. Data Visualization Tools and Technologies
6. Drawing Insights from Visualized Data
7. Best practices for presenting data visualizations

Data Collection - Web Crawler

Definition

- a web crawler is a computer program that browses the World Wide Web in a methodical, automated manner. (Wikipedia)

Utilities:

- Gather pages from the Web.
- Support a search engine, perform data mining and so on.

Object:

- Text, video, image and so on.
- Link structure

Web Features of a Crawler

Should provide:
- Distributed
- Scalable
- Performance and efficiency
- Quality
- Freshness
- Extensible

Some scraping knowledge

1. HTTP: the communication protocol
2. HTML: the language in which web pages are defined (https://www.w3schools.com/html/)
3. JS: javascript (code executing in the browser) (https://www.w3schools.com/js/default.asp)
4. CSS: style sheets, how web pages are styled. Important, but does not contain data. (https://www.w3schools.com/css/default.asp)
5. JPG, PNG, BMP: images, usually not interesting
6. CSV / TXT / JSON / XML: data, interesting !!!

Obtaining Data – Web Scraping

Why scraping the web?

- Vast source of information; can combine with multiple datasets
- Companies have not provided APIs
- Automate tasks
- Keep up with sites / real-time data
- Fun!

Obtaining Data – Web Scraping

Robots.txt

- Specified by web site owner
- Gives instructions to web robots (e.g., your code)
- Located at the top-level directory of the web server
  - E.g., http://google.com/robots.txt

Robots.txt

- Protocol for giving spiders ("robots") limited assess to a website

  www.robotstxt.org/wc/norobots.html

- Website announced its request on what can(not) be crawled

- For a server, create a file /robots.txt

- This file specifies access restrictions

An example of robots.txt

- No rebot should visit any URL starting with "/yoursite/temp/", except the rebot called "searchengine":
- User-agent: *
- Disallow: /yoursite/temp/
- User-agent: searchengine
- Disallow:

Obtaining Data – Web Scraping

Web Servers

- A server maintains a long-running process (also called a daemon), which listens on a pre-specified port
- It responds to requests, which is sent using a protocol called HTTP (HTTPS is secure)
- Our browser sends these requests and downloads the content, then displays it
- 2– request was successful, 4– client error, often `page not found`; 5– server error (often that your request was incorrectly formed)

Obtaining Data – Web Scraping

HTML

- Tags are denoted by angled brackets
- Almost all tags are in pairs e.g., Hello
- Some tags do not have a closing tag e.g.,

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ttle</title>
  </head>
  <body>
    <h1>Body Title</h1>
    <p>Body Content</p>
  </body>
</html>
```

Obtaining Data – Web Scraping

HTML

- <html>, indicates the start of an html page
- <body>, contains the items on the actual webpage (text, links, images, etc)
- <p>, the paragraph tag. Can contain text and links
- <a>, the link tag. Contains a link url, and possibly a description of the link
- <input>, a form input tag. Used for text boxes, and other user input
- <form>, a form start tag, to indicate the start of a form
- <img>, an image tag containing the link to an image

Obtaining Data – Web Scraping

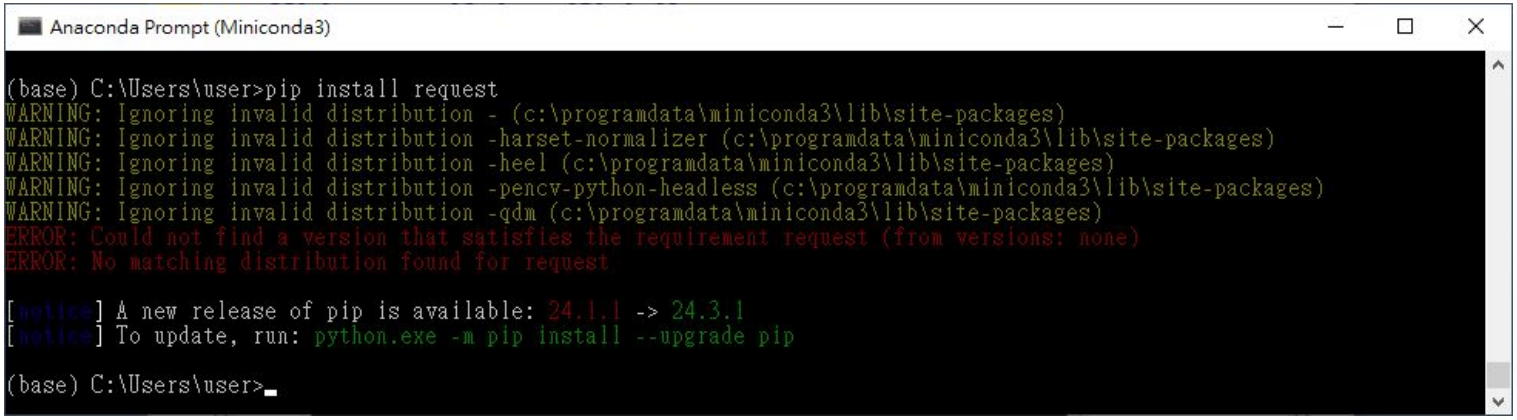
How to Web scrape:

1. Get the webpage content
    - Requests (Python library) gets a webpage for you
2. Parse the webpage content
    - (e.g., find all the text or all the links on a page)
    - BeautifulSoup (Python library) helps you parse the webpage.
    - Documentation: http://crummy.com/software/BeautifulSoup

Obtaining Data – Web Scraping

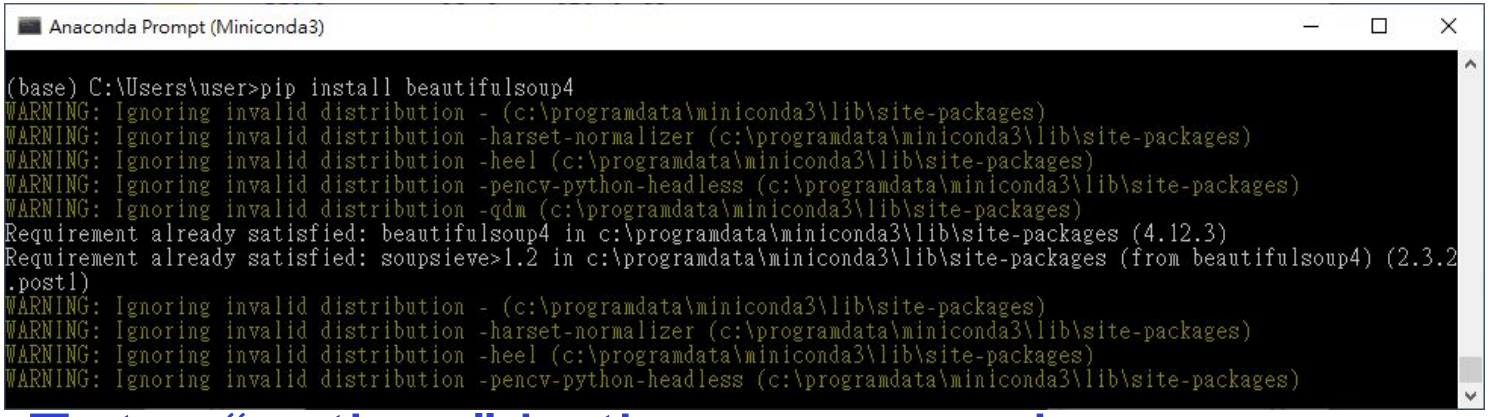
1. Open tools “Anaconda Prompt” which will direct you to the command prompt.

2. Install python package requests by entering “pip install requests”

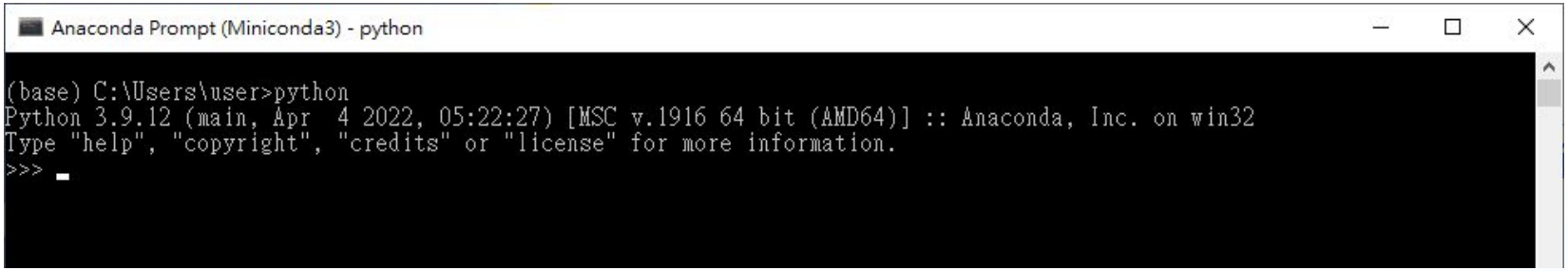


Obtaining Data – Web Scraping

3. Install python package beautifulsoup4 by entering “pip install beautifulsoup4”



4. Enter “python” in the command



Obtaining Data – Web Scraping

1. Get the webpage content

Requests (Python library) gets a webpage for you

```
import requests

url = "https://www.nytimes.com/international/"

page = requests.get(url)
page.status_code
page.content
```

Returns the content of the response, in bytes.

Obtaining Data – Web Scraping

2. Parse the webpage content

BeautifulSoup (Python library) helps you parse a webpage

```
soup = BeautifulSoup(page.content, "html.parser")
soup.title
soup.title.text
```

Returns the text part of the title tag.

e.g.,

```
The New York Times - Breaking News
```

Obtaining Data – Web Scraping

BeautifulSoup

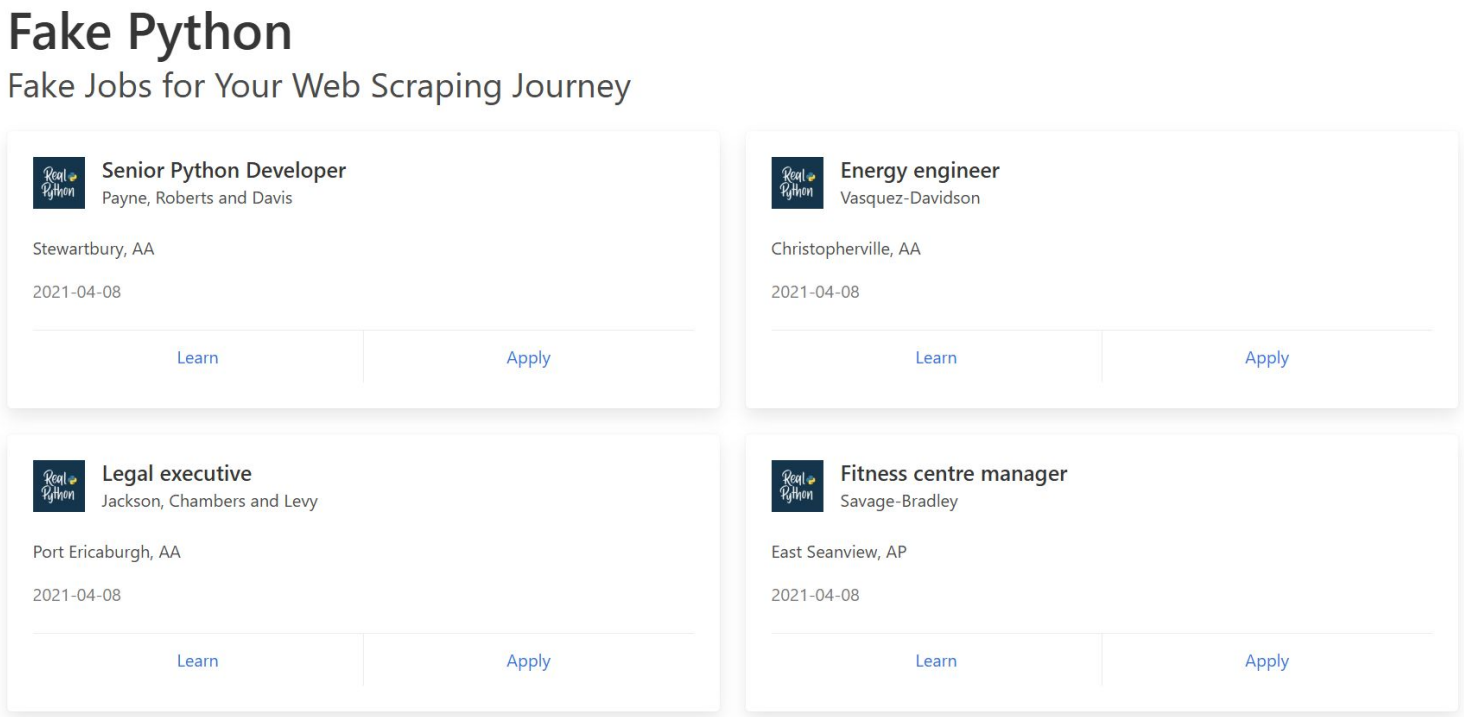
- Helps make messy HTML digestible
- Provides functions for quickly accessing certain sections of HTML content

Example

```
import bs4
## get bs4 object
soup = bs4.BeautifulSoup(source)
## all a tags
soup.findAll('a')
## first a
soup.find('a')
## get all links in the page
link_list = [l.get('href') for l in soup.findAll('a')]
```

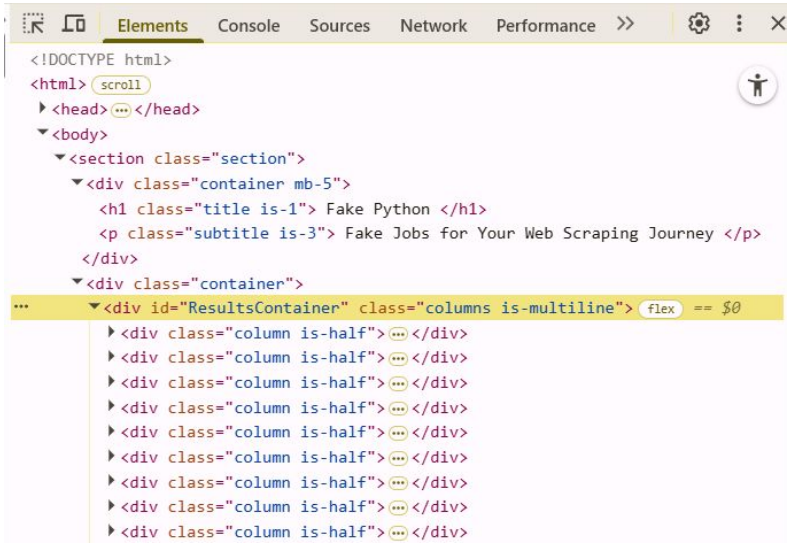
Web Scraping Practise 1

Open Browser and enter
https://realpython.github.io/fake-jobs/



Web Scraping Practise 1

Right Click and Select “Inspect”



Open jupyter notebook and enter following code

```
import requests
from bs4 import BeautifulSoup

URL = "https://realpython.github.io/fake-jobs/"
page = requests.get(URL)
```

Web Scraping Practise 1

```
soup = BeautifulSoup(page.content, "html.parser")

results = soup.find(id="ResultsContainer")
print(results.prettify())
```

```
>>> print(results.prettify())
<div class="columns is-multiline" id="ResultsContainer">
 <div class="column is-half">
  <div class="card">
   <div class="card-content">
    <div class="media">
     <div class="media-left">
      <figure class="image is-48x48">
       <img alt="Real Python Logo" src="https://files.realpython.com/media/real-python-logo-thumbnail.7f0db70c2ed2.jpg?_
_no_cf_polish=1"/>
      </figure>
     </div>
     <div class="media-content">
      <h2 class="title is-5">
       Senior Python Developer
      </h2>
      <h3 class="subtitle is-6 company">
       Payne, Roberts and Davis
      </h3>
     </div>
    </div>
```

Web Scraping Practise 1

Find Elements by HTML Class Name

You've seen that every job posting is wrapped in a `<div>` element with the class `card-content`. Now you can work with your new object called `results` and select only the job postings in it. These are, after all, the parts of the HTML that you're interested in! You can pick out all job cards in a single line of code:

Python

```
>>> job_cards = results.find_all("div", class_="card-content")
```

Here, you call `.find_all()` on `results`, which is a `BeautifulSoup` object. It returns an iterable containing all the HTML for all the job listings displayed on that page.

Take a look at all of them:

Python

```
>>> for job_card in job_cards:
...     print(job_card, end="\n" * 2)
...
```

```
<div class="card-content">
<div class="media">
<div class="media-left">
<figure class="image is-48x48">
<img alt="Real Python Logo" src="https://files.realpython.com/media/real-python-logo-
</figure>
</div>
<div class="media-content">
<h2 class="title is-5">Senior Python Developer</h2>
<h3 class="subtitle is-6 company">Payne, Roberts and Davis</h3>
</div>
</div>
<div class="content">
<p class="location">
        Stewartbury, AA
      </p>
<p class="is-small has-text-grey">
<time datetime="2021-04-08">2021-04-08</time>
```

Extract title Content

```
>>> for job_card in job_cards:
...     title_element = job_card.find("h2", class_="title")
...     print(title_element)
...
<h2 class="title is-5">Senior Python Developer</h2>
<h2 class="title is-5">Energy engineer</h2>
<h2 class="title is-5">Legal executive</h2>
<h2 class="title is-5">Fitness centre manager</h2>
<h2 class="title is-5">Product manager</h2>
<h2 class="title is-5">Medical technical officer</h2>
<h2 class="title is-5">Physiological scientist</h2>
<h2 class="title is-5">Textile designer</h2>
<h2 class="title is-5">Television floor manager</h2>
<h2 class="title is-5">Waste management officer</h2>
<h2 class="title is-5">Software Engineer (Python)</h2>
<h2 class="title is-5">Interpreter</h2>
<h2 class="title is-5">Architect</h2>
<h2 class="title is-5">Meteorologist</h2>
```

Extract Content of title, company, location

Python

```
>>> for job_card in job_cards:
...     title_element = job_card.find("h2", class_="title")
...     company_element = job_card.find("h3", class_="company")
...     location_element = job_card.find("p", class_="location")
...     print(title_element)
...     print(company_element)
...     print(location_element)
...     print()
...
<h2 class="title is-5">Senior Python Developer</h2>
<h3 class="subtitle is-6 company">Payne, Roberts and Davis</h3>
<p class="location">
        Stewartbury, AA
      </p>

<!-- ... -->
```

Full Article : https://realpython.com/beautiful-soup-web-scraper-python/

Web Scraping Practise 2

Open Browser and enter https://www.hk01.com/

Right Click and Select “Inspect”

Select the Content that you want to extract

Web Scraping Practise 2

Right Click and click “Edit As HTML”
Copy part of the code until “/div>

Web Scraping Practise 2

Open Perplexity at https://www.perplexity.ai/

Enter prompt "write python beatifulsoap to extract data from website url https://www.hk01.com/ with html code {htmlcode}"

Where {htmlcode} = " <button data-testid="hamburger-menu-button" class="box-border w-6 px-0 py-3 m-0 transition-all duration-[0.6s] ease-[cubic-bezier(0.19,1,0.22,1)] [transform:translateZ(0)] border-none outline-none rounded-none cursor-pointer shadow-none bg-transparent focus:shadow-none focus:outline-none" aria-label="mobile-menu"><span class="relative block h-px rounded-[1px] transition-all duration-[0.6s] ease-[cubic-bezier(0.19,1,0.22,1)] w-[70%] left-[15%] bg-black/[0.78] before:w-full after:w-full before:absolute after:absolute before:left-0 after:left-0 before:content-[' '] after:content-[' '] before:h-px after:h-px before:rounded-[1px] after:rounded-[1px] before:transition-all before:duration-[0.6s] before:ease-[cubic-bezier(0.19,1,0.22,1)] after:transition-all after:duration-[0.6s] after:ease-[cubic-bezier(0.19,1,0.22,1)] before:bg-black/[0.78] after:bg-black/[0.78] before:-top-1.5 before:rotate-0 after:top-1.5 after:rotate-0"></button> "

Web Scraping Practise 2

In original perplexity prompt, type the following prompt "Add code that save it into csv file"

	A
1	text,url,class
2	,/,
3	港聞,/zone/1/%E6%B8%AF%E8%81%9E,
4	兩會焦點,/issue/10355/%E5%85%A9%E6%9C%832025-%E5%9C%8B%E9%9A%9B%E5%B1%80%
5	娛樂,/zone/2/%E5%A8%9B%E6%A8%82,
6	最平酒店,https://clk.omgt3.com/?AID=2200313&PID=55029&uid=content&uid2=channel_旅遊_editor
7	國際,/zone/4/%E5%9C%8B%E9%9A%9B,
8	即時,/latest,
9	熱榜,/hot,
10	生活,/zone/8/%E7%94%9F%E6%B4%BB,
11	科技,/zone/11/%E7%A7%91%E6%8A%80%E7%8E%A9%E7%89%A9,
12	中國,/zone/5/%E4%B8%AD%E5%9C%8B,
13	體育,/zone/3/%E9%AB%94%E8%82%B2,
14	01深圳,/zone/25/01%E6%B7%B1%E5%9C%B3,
15	經濟,/zone/14/%E7%B6%93%E6%BF%9F,
16	觀點,/zone/12/%E8%A7%80%E9%BB%9E,
17	健康,/zone/24/%E5%81%A5%E5%BA%B7,
18	好食玩飛,/zone/19/%E5%A5%BD%E9%A3%9F%E7%8E%A9%E9%A3%9B,
19	女生,/zone/6/%E5%A5%B3%E7%94%9F,
20	熱話,/zone/7/%E7%86%B1%E8%A9%B1,
21	藝文格物,/zone/9/%E8%97%9D%E6%96%87%E6%A0%BC%E7%89%A9,
22	社區,/zone/10/%E7%A4%BE%E5%8D%80,
23	教育,/zone/23/%E6%95%99%E8%82%B2,
24	简,/,"font-light, pointer-events-none, !text-light-n5"
25	
26	
27	Mobile Menu Details:
28	Exists: True

How To Do Data Mining: Machine Learning Techniques

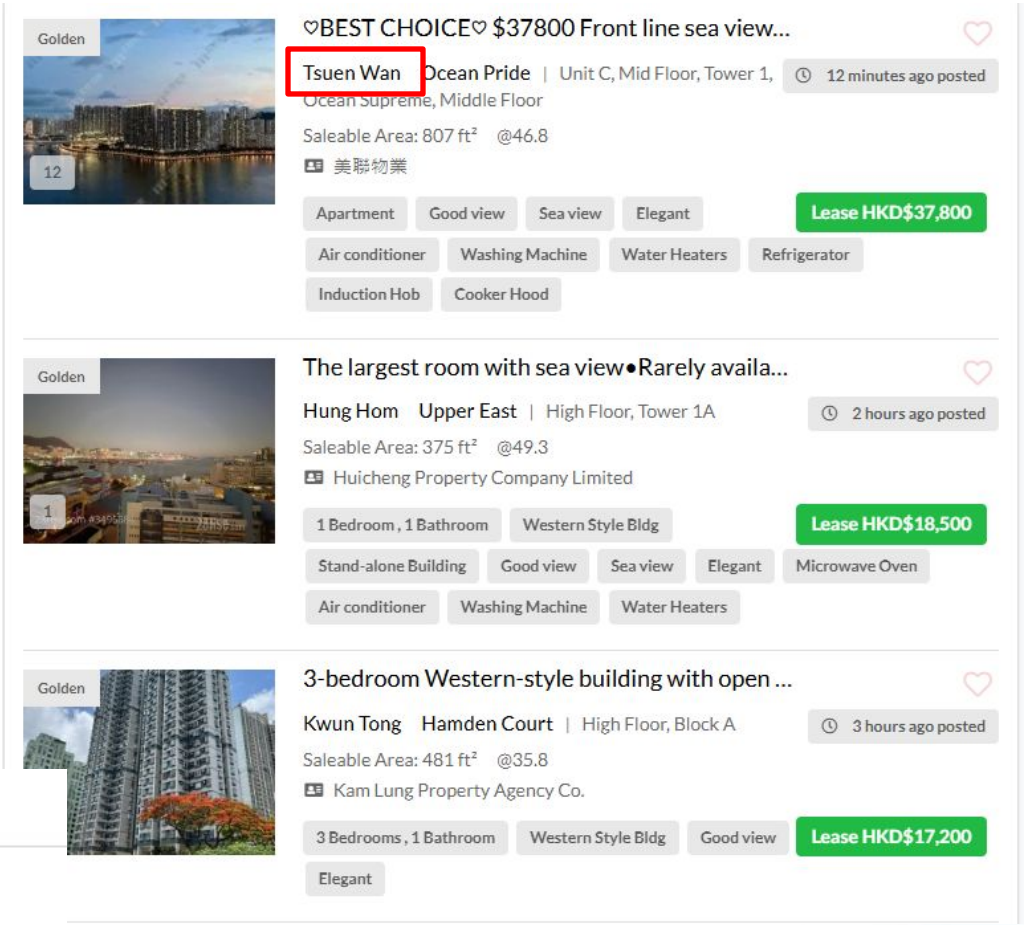
Features / Attributes

1. District
2. Floor

Outcome

1. Price Per Square

https://www.28hse.com/en/rent/apartment



https://www.28hse.com/en/rent/apartment/property-3496859

Floor zone	Middle Floor
Saleable Area	807 ft²
	Unit Price: @46.8

How To Do Data Mining: Machine Learning Techniques

In perplexity, enter the prompt “write python beautifulsoap to extract data from website url and save data into csv file using {pythoncode} from original html code {htmlcode}”

```
pythoncode = ‘’’
import csv
import pandas as pd
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.firefox.options import Options
import requests
import time

from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def main(url:str, per_page:int = 120, **kwargs):
    options = kwargs.get('driver_options', Options())
    # per_page = kwargs.get('per_page', 120)
    # url = f"{url}?per_page={per_page}"
    options.headless = True

    firefox_options = Options()
    firefox_options.set_preference("dom.webdriver.enabled", False)

    driver = webdriver.Firefox(
        options=firefox_options
    )

    driver.get(url)
    # Get the page source
    html = driver.page_source

    # Create a BeautifulSoup object from the HTML
    soup = BeautifulSoup(html, 'html.parser')’
```

How To Do Data Mining: Machine Learning Techniques

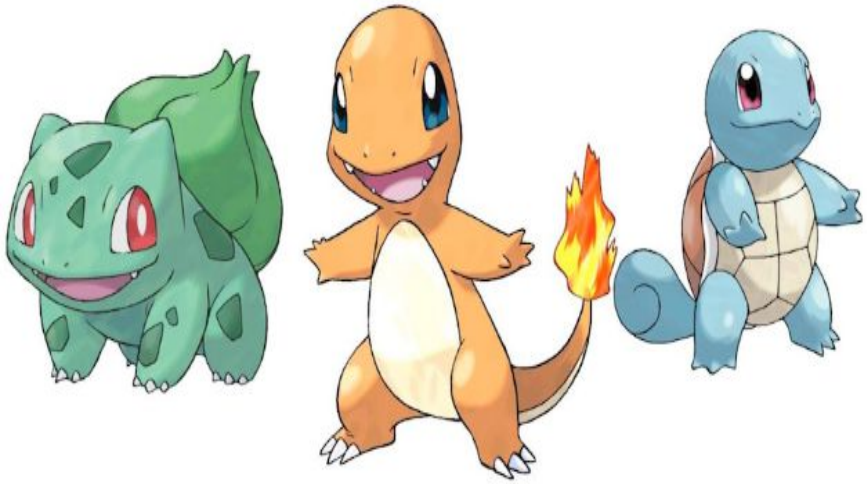
```
htmlcode = ‘’’
<div class="ui segment search_results_div" id="search_results_div">     <div class="listItems ui divided
items">
                <div class="item property_item ">
            <div class="image myimage desktop_myimage">
                <a class="detail_page" href="https://www.28hse.com/en/rent/office/property-3445279"
attr1="3445279" target="_blank">
                                        <img class="detail_page_img desktop_detail_page_img"
src="https://i1.28hse.com/2025/05/20250523001147474746325_thumb.jpg"
onerror="javascript:this.src='/assets/img/loadingphoto170_125.jpg';
                            javascript:$(this).css('width','190px');">

                    <div class="myimage_count_outer"><div class="ui labels myimage_count">
                    <span class="ui label" style="">
                        10


                    </span></div></div>
                                                        <div class="ui top left attached small label
grade_label">Golden</div>

                </a>

            </div>
            <div class="content">
                <div class="right floated">
                                        <div class="fav" attr1="3445279" attr2="inactive"><i class="ui red heart
large outline icon"></i></div>
‘’’
```

Download the data

Download the Pokemon dataset from:

https://github.com/innoviai/IPA_Courses-Jan2026/blob/main/pokemon_dataset.csv

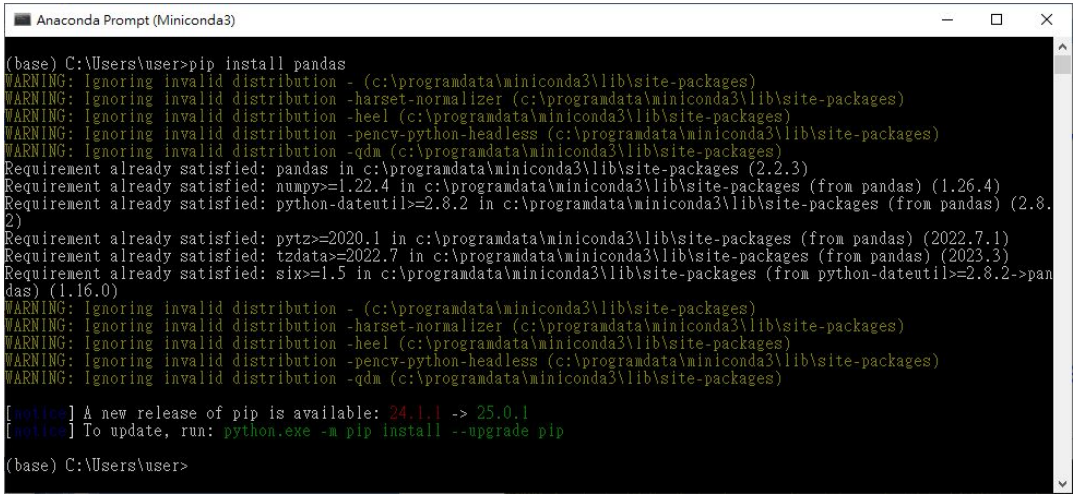
Save the pokemon dataset file in a location you'll remember.



Library Installation in Python

1. Open tools “Anaconda Prompt” which will direct you to the command prompt.

2. Install python package requests by entering “pip install pandas”



Reading in the data

First we import the Python packages we are going to use.
Then we use Pandas to load in the dataset as a data frame.

```
import numpy as np
import pandas as pd
#from matplotlib import pyplot as plt
#import seaborn as sns

df1 = pd.read_csv("pokemon_dataset.csv", index_col=0, encoding="ISO-8859-1")
```

NOTE: The argument `index_col` argument states that we'll treat the first column of the dataset as the ID column.
NOTE: The encoding argument allows us to by pass an input error created by special characters in the data set.

Examine the data set

```
print(df1.head())
```

#	Name	Type 1	Type 2	Total	...	Sp. Def	Speed	Stage	Legendary
					...				
1	Bulbasaur	Grass	Poison	318	...	65	45	1	False
2	Ivysaur	Grass	Poison	405	...	80	60	2	False
3	Venusaur	Grass	Poison	525	...	100	80	3	False
4	Charmander	Fire	NaN	309	...	50	65	1	False
5	Charmeleon	Fire	NaN	405	...	65	80	2	False

```
print(df1.describe())
```

	Total	HP	Attack	...	Sp. Def	Speed	Stage
count	151.00000	151.000000	151.000000	...	151.000000	151.000000	151.000000
mean	407.07947	64.211921	72.549669	...	66.019868	68.933775	1.582781
std	99.74384	28.590117	26.596162	...	24.197926	26.746880	0.676832
min	195.00000	10.000000	5.000000	...	20.000000	15.000000	1.000000
25%	320.00000	45.000000	51.000000	...	49.000000	46.500000	1.000000
50%	405.00000	60.000000	70.000000	...	65.000000	70.000000	1.000000
75%	490.00000	80.000000	90.000000	...	80.000000	90.000000	2.000000
max	680.00000	250.000000	134.000000	...	125.000000	140.000000	3.000000

Intelligent Process Automation (IPA)

- Using Artificial Intelligence (AI), e.g.,
  - Machine Learning (ML)
  - Natural Language Processing (NLP)
- Enhancing cognitive ability of the automation

Intelligent Process Automation (IPA)

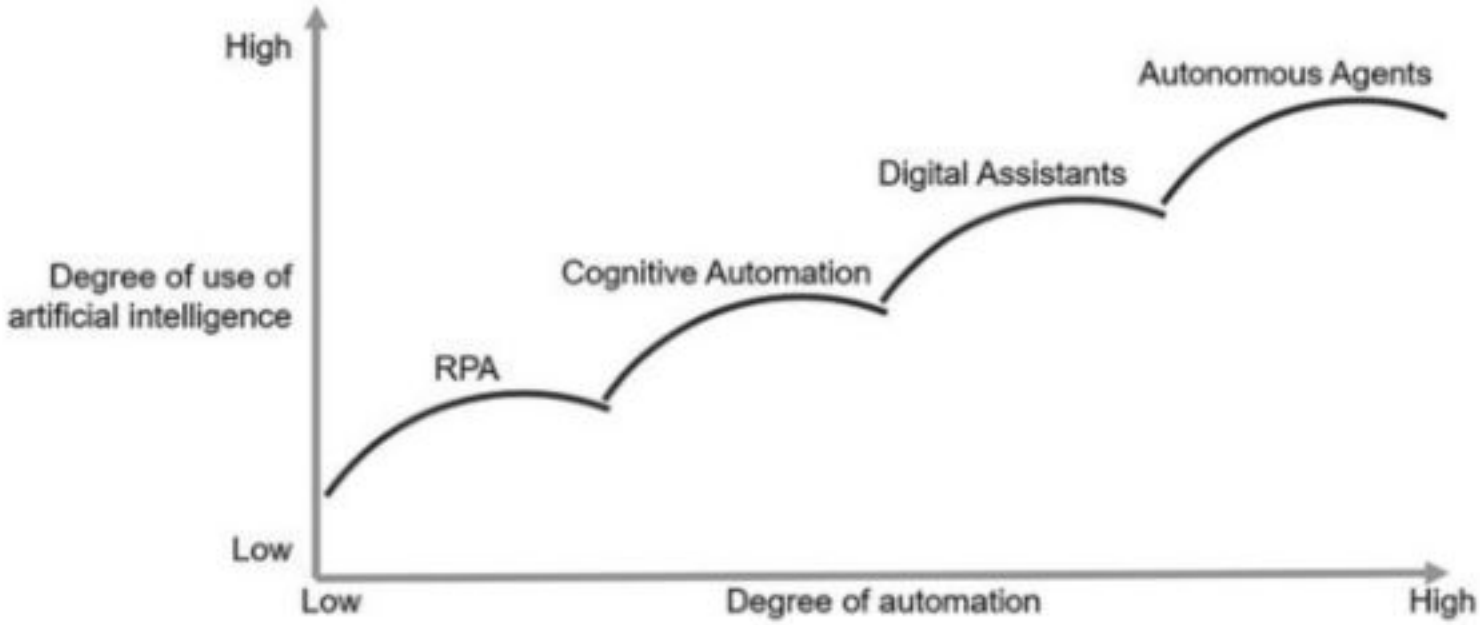
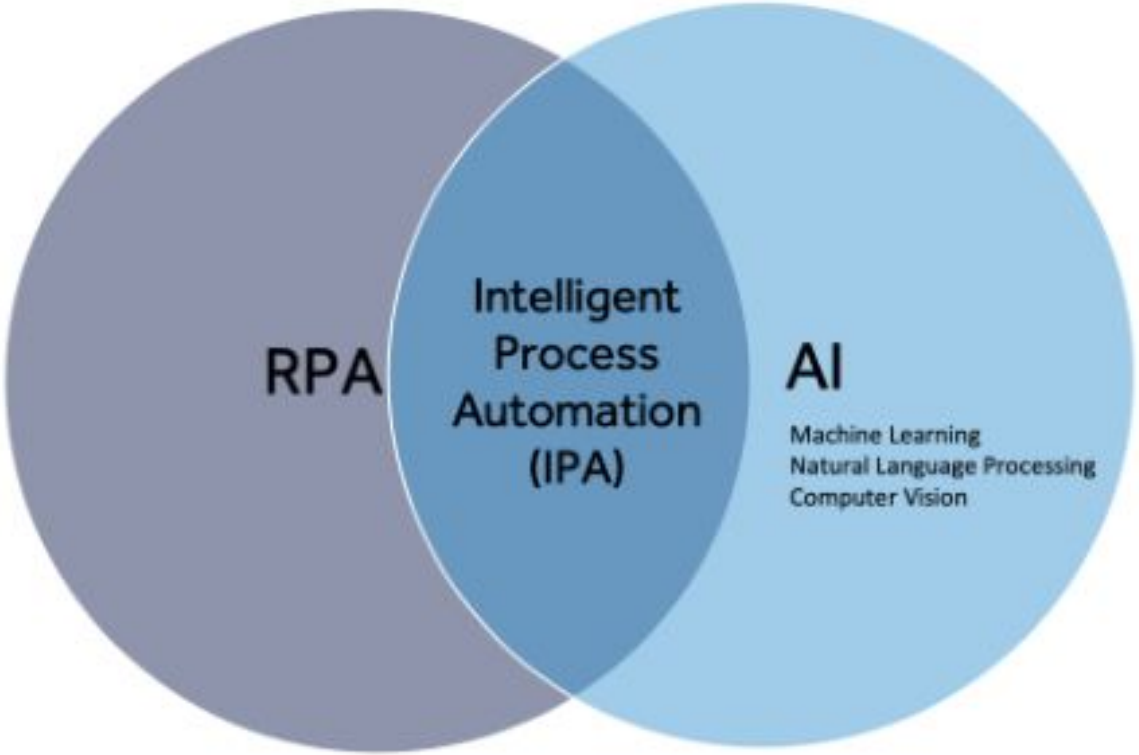


Fig. 2.2 Classification of RPA according to the degree of automation and the use of artificial intelligence (based on Ostrowicz 2018, p. 4)

1. Smeets, M., Erhard, R., & Kaußler, T. (2021, July 30). *Robotic Process Automation (RPA) in the Financial Sector: Technology - Implementation - Success for Decision Makers and Users*. Springer. https://doi.org/10.1007/978-3-658-32974-7

Intelligent Process Automation (IPA)



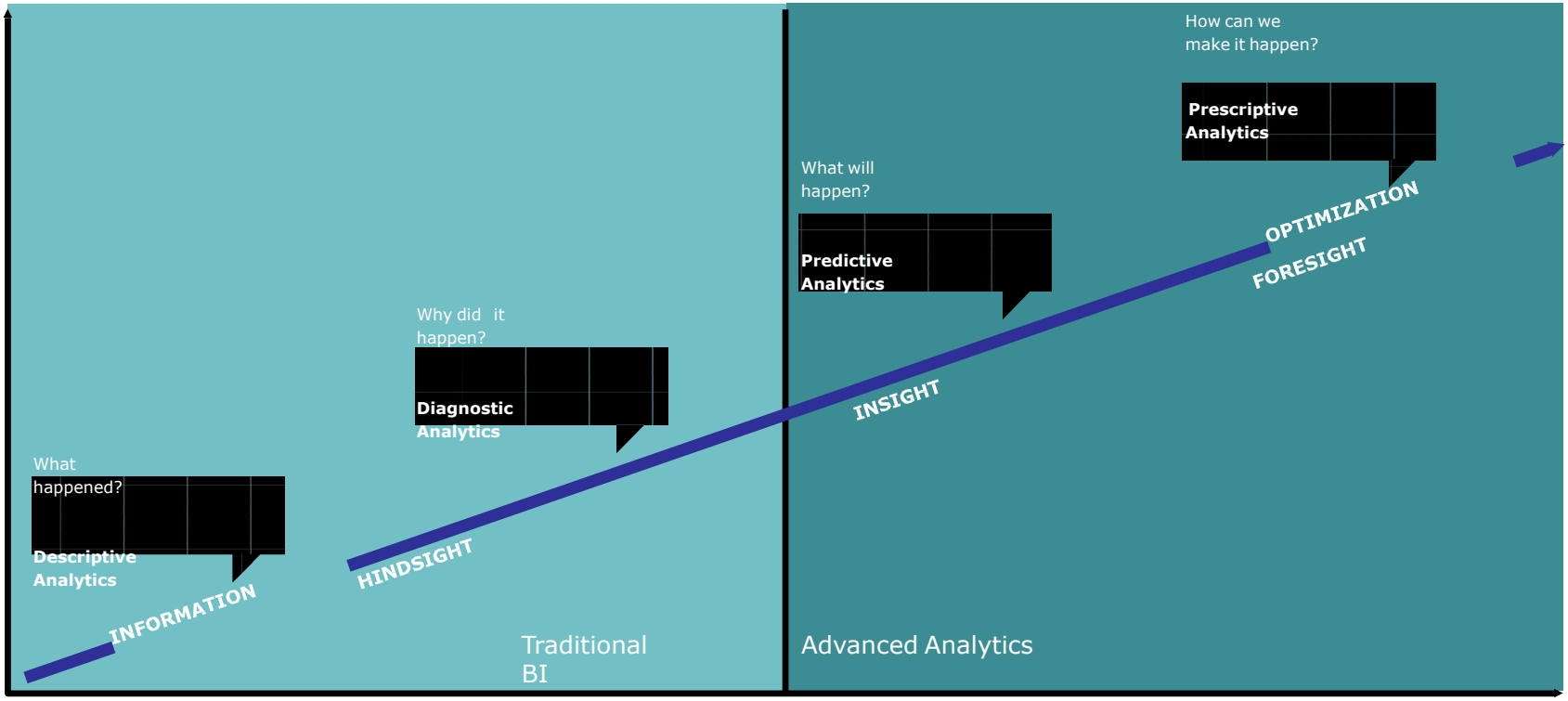
1. Kosmopoulos, C. (2021, May 11). *What is Intelligent Process Automation (IPA)? | Blueprint*. What Is Intelligent Process Automation (IPA)? | Blueprint. https://www.blueprintsys.com/blog/rpa/what-is-intelligent-process-automation-ipa

What Is Data Science?

From Wikipedia:

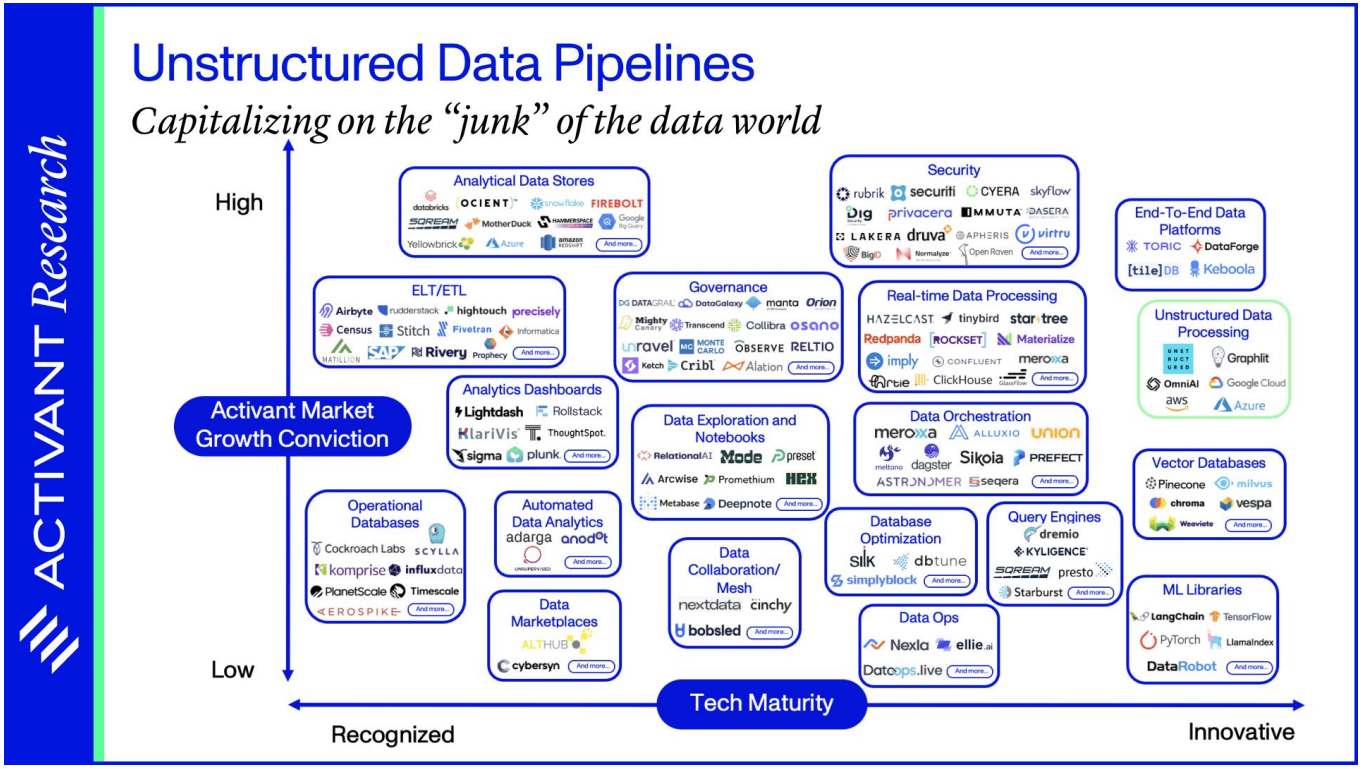
Data science, also known as data-driven science, is an interdisciplinary field about scientific processes and systems to extract knowledge or insights from data in various forms, either structured or unstructured, which is a continuation of some of the data analysis fields such as statistics, machine learning, data mining ,and predictive analytics similar to Knowledge Discovery in Databases (KDD).

How Is Business Intelligence Different From Data Science?

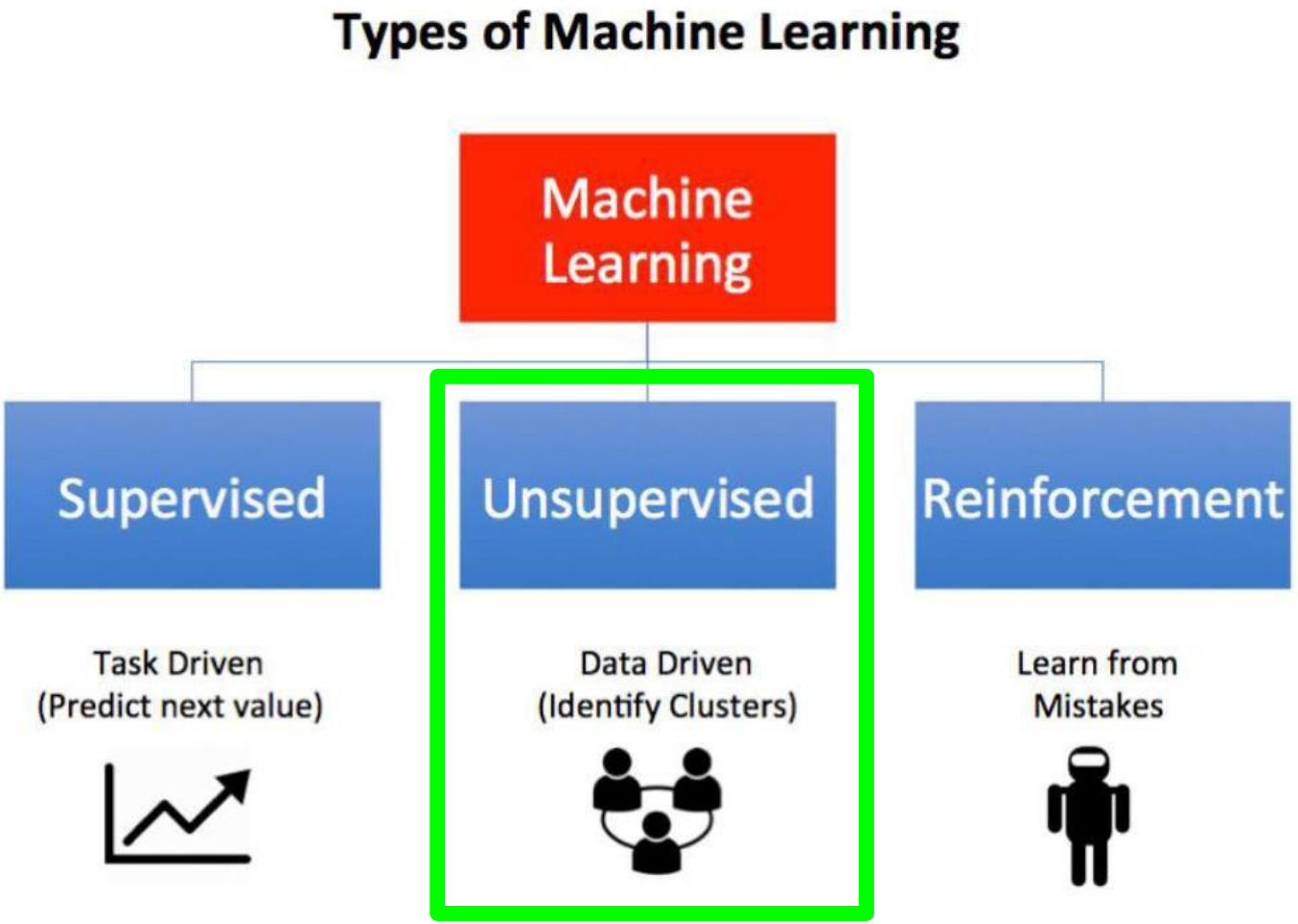


IPA – Data Science

IPA's integration with data science is particularly impactful in several areas: Unstructured Data Processing: IPA can convert unstructured data into structured formats suitable for analysis, enabling businesses to leverage insights from previously untapped data source.



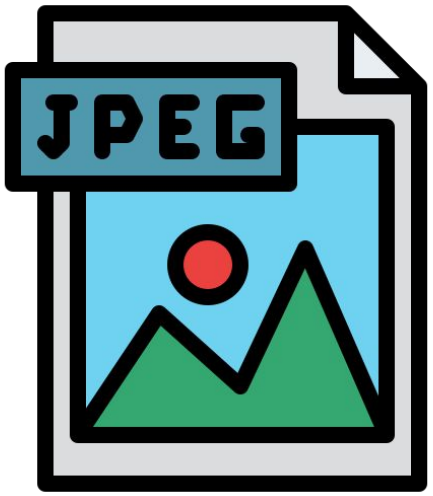
What is Business Analytics & Machine Learning?



Processed Unstructured Data

PDF Document

Paper Form



Image, Photo

Processed Unstructured Data - PDF Document

Step 1 : Open Anaconda Prompt and enter “pip install pymupdf”
Step 2 : Open Jupyter notebook and enter following code

```
# RACAS Application Form Conversion (PDF to form)
def download_pdf(url, save_path):
    response = requests.get(url)
    if response.status_code == 200:
        with open(save_path, 'wb') as f:
            f.write(response.content)
        print(f"PDF downloaded and saved to {save_path}")
    else:
        print(f"Failed to download PDF: {response.status_code}")

import requests
pdf_url = 'https://www.hkag.org/Forms/RACAS%20application%20form.pdf'
# Usage example
pdf_file_path = "RACAS_applicationform_14012026.pdf"
# Step 1: Download the PDF
download_pdf(pdf_url, pdf_file_path)
```

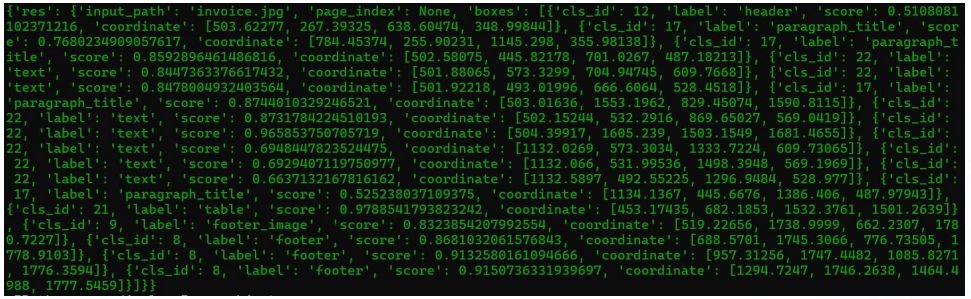
Processed Unstructured Data - PDF Document

```
import fitz  # PyMuPDF

pdf_file_path = "RACAS_applicationform_14012026.pdf"
pdf_document = fitz.open(pdf_file_path)

# Initialize a list to hold the table data
table_data = []

# Iterate through each page of the PDF
for page_num in range(len(pdf_document)):
    page = pdf_document[page_num]
    text = page.get_text("text")  # Extract text from the page

    # Split the text into lines
    lines = text.splitlines()

    # Process lines to find tables (you may need to adjust this logic)
    for line in lines:
        # Example logic: Split lines by whitespace (this may need adjustment)
        row = line.split()
        if row:  # If the row is not empty
            table_data.append(row)

pdf_document.close()
# Print extracted table data
for row in table_data:
    print(row)
```

Processed Unstructured Data - Image/Photo

Step 1 : Open Anaconda Prompt and enter “pip install paddleocr paddlepaddle”
Step 2: Open Jupyter notebook and enter following code

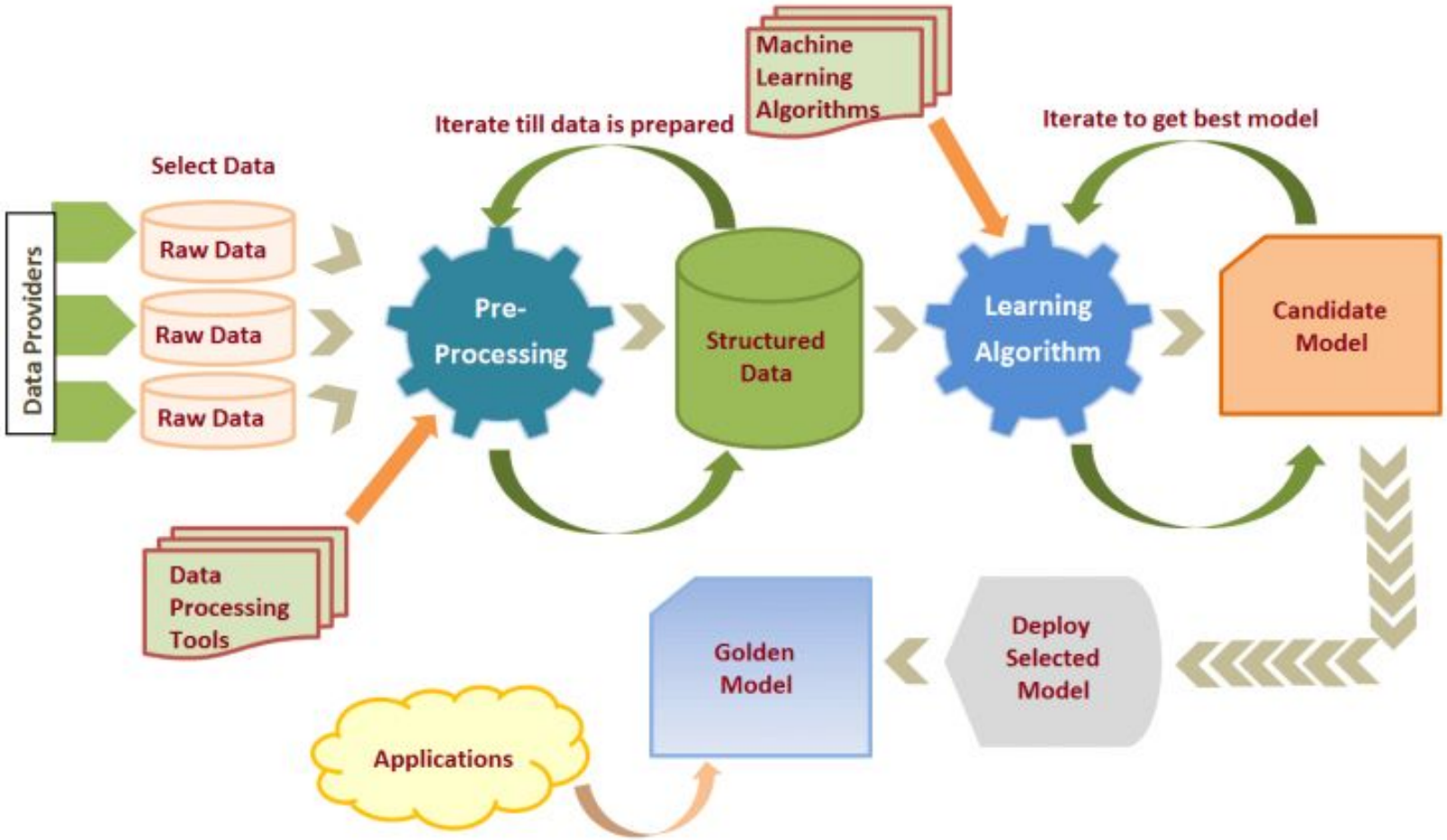
```
#Paddle OCR Layout Analysis Module
#https://www.paddleocr.ai/main/en/version3.x/module_usage/layout_analysis.html#iii-quick-integration
from paddleocr import LayoutDetection

# Initialize the LayoutDetection model
model = LayoutDetection(model_name="PP-DocLayoutV2") #Better than PP-DocLayout_plus-L

# Predict layout from the image
output = model.predict("invoice.jpg", batch_size=1, layout_nms=True)


# Prepare to save results
results = []

# Process each result
for res in output:
    # Print the result details
    res.print()

    # Save the annotated image
    res.save_to_img(save_path="./output/")
    res.save_to_json(save_path="./output/invoice_res.json")
```



Processed Unstructured Data - Image/Photo

```
#Display original image and labelled image
from IPython.display import Image, display

# Specify the path to your image file
orig_image_path = "invoice.jpg"
label_image_path = "./output/invoice_res.jpg"

# Display the images with specified resolution
display(Image(filename=orig_image_path, width=640, height=480))
display(Image(filename=label_image_path, width=640, height=480))
```

Applied Machine Learning Pipeline Overview



How To Do Data Mining: Machine Learning Techniques

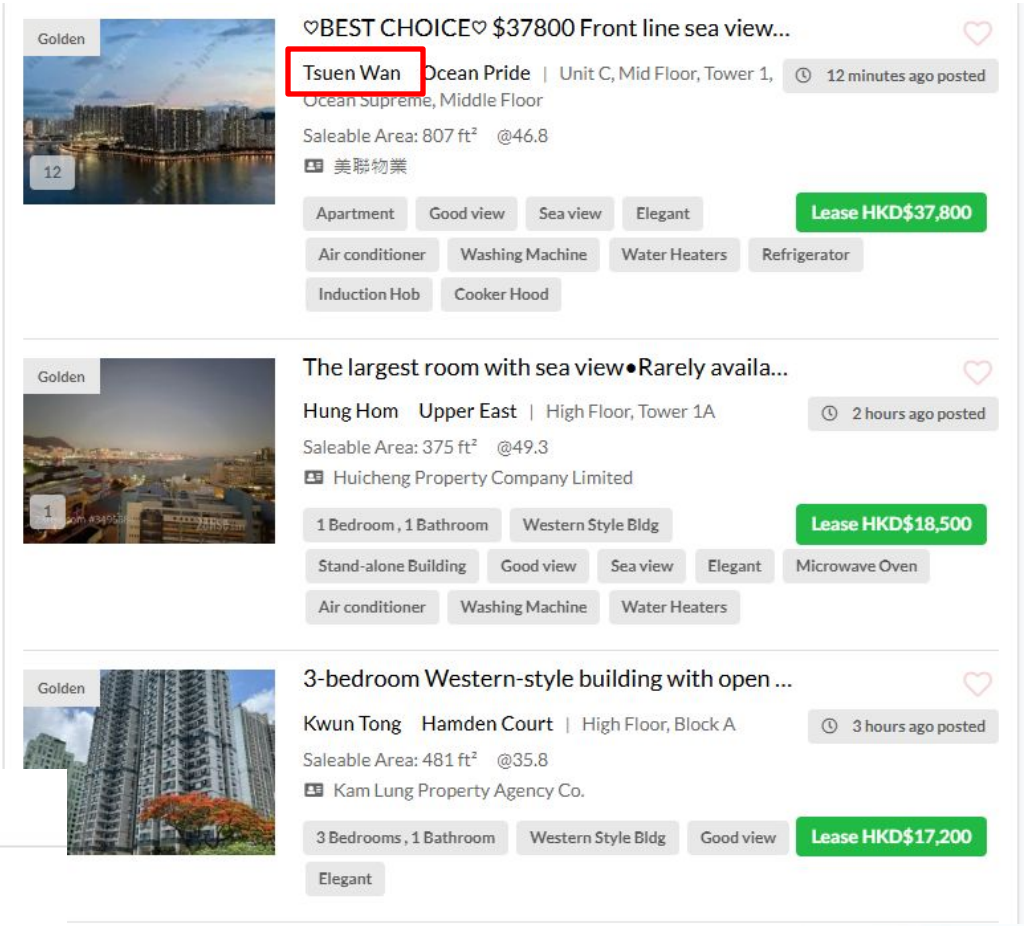
Features / Attributes

1. District – Get from HK Building Record
2. Floor – Get from HK Building Record
3. School Net Numbers - POA School Net Look-up Table

Outcome

Price Per Square - Get from HK Building Record

https://www.28hse.com/en/rent/apartment



https://www.28hse.com/en/rent/apartment/property-3496859

HK Building Record Data Cleaning

This Tasks is to combining Building Information and School Net Numbers

Step 1. Import the required libraries:

```
import pandas as pd
#Loading orginal CSV file containing historic weather data:
weatherdata = pd.read_csv("historic_weatherdata.csv")
print(weatherdata.head(5))
weatherdata['date'] = pd.to_datetime(weatherdata['date']).dt.strftime('%Y-%m-%d')

#Loading orginal CSV file containing historic property data:
hkbuildingdata = pd.read_csv('hkbuildinginfo.csv', encoding='ISO-8859-1')
hkbuildingdata.head(5)
```

Step 2: #Clean up the data

```
# Check for missing values
print("\nMissing values in each column:")
print(hkbuildingdata.isnull().sum())

#Handle missing values by Dropping rows with missing values
hkbuildingdata.drop(columns=['withpre', 'rootid', 'fatherid', 'catid', 'source', 'block', 'state',
        'date_dm'],inplace=True, errors='ignore')

# Assuming 'floor' is the column causing the error
hkbuildingdata['floor'] = hkbuildingdata['floor'].astype(str)  # Convert to string type

# Now you can safely use the .str accessor
hkbuildingdata['floor'] = hkbuildingdata['floor'].str.replace(r'[^\d]', '', regex=True)  # Remove non-numeric
        characters
```

```
Missing values in each column:
date                    0
withpre             23564
id                      0
rootid                  0
fatherid                0
catid                   0
catname                 0
catfathername           0
url_father              0
url_cat                 0
source              11556
contract                0
memo                    0
price                   0
price_value             0
holddate                0
winloss_flag            0
winloss                 0
act_area                0
area                    0
arearaw                 4
sq_price                0
sq_price_value          0
sq_actprice             0
sq_actprice_value       0
month                   0
day                     0
date_dm                 0
date_y                  0
block               32308
state               77003
floor                  32
room                  258
addr                    0
Unnamed: 34         80000
dtype: int64
```

HK Building Record Data Cleaning

#Continue from previous page

```
# Convert back to numeric if needed
hkbuildingdata['floor'] = pd.to_numeric(hkbuildingdata['floor'], errors='coerce')  # Convert to numeric, coercing errors to
    NaN

# Step 4: Fill missing values for 'floor' and 'room'
hkbuildingdata['floor'] = hkbuildingdata['floor'].fillna(hkbuildingdata['floor'].median())  # Fill with median for numeric data
hkbuildingdata['room'] = hkbuildingdata['room'].fillna(hkbuildingdata['room'].mode()[0])   # Fill with mode for categorical data

# Step 6: Rename columns to more meaningful names
hkbuildingdata.rename(columns={'floor': 'Floor', 'sq_price_value': 'Price Per Square', 'catname': 'Building Name', 'catfathername':
    'District'}, inplace=True)

# Step 7: Display the cleaned DataFrame and check for remaining missing values
print("\nCleaned DataFrame:")
print(hkbuildingdata.head())

print("\nRemaining Missing Values:")
print(hkbuildingdata.isnull().sum())
```

```
Cleaned DataFrame:
        date      id       Building Name       District  \
0 2020-11-27  683333     Bel Air Heights   Diamond Hill
1 2020-11-27  683332       Fa Yuen Plaza       Mong Kok
2 2020-11-27  683331      Caldecott Hill  Yau Yat Tsuen
3 2020-11-27  683330    Pang Ching Court   Wong Tai Sin
4 2020-11-27  683329  Metro Harbour View   Tai Kok Tsui

                                 url_father  \
0   https://data.28hse.com/en/kl/diamond-hill
1      https://data.28hse.com/en/kl/mong-kok
2 https://data.28hse.com/en/kl/yau-yat-tsuen
3   https://data.28hse.com/en/kl/wong-tai-sin
4   https://data.28hse.com/en/kl/tai-kok-tsui

                                        url_cat   contract  \
0 https://data.28hse.com/en/kl/diamond-hill/2058...  Agreement
1 https://data.28hse.com/en/kl/mong-kok/2712_fa-...  Agreement
2 https://data.28hse.com/en/kl/yau-yat-tsuen/309...  Agreement
3 https://data.28hse.com/en/kl/wong-tai-sin/5035...  Agreement
4 https://data.28hse.com/en/kl/tai-kok-tsui/2564...  Agreement

            memo   price  price_value  ... Price Per Square  sq_actprice  \
0  20112702480015    $10M     10000000  ...         12106.54       $15221
1  20112702470044  $4.28M      4280000  ...         12334.29       $20000
2  20112702440018    $11M     11000000  ...          8580.34       $11100
3  20112702410017   $4.9M      4900000  ...          8404.80           --
4  20112702400032  $7.15M      7150000  ...         12016.81       $17354

  sq_actprice_value month day  date_y Floor  room                addr  \
0          15220.70   Nov  27    2020   2.0     E  BLOCK 2 2#/F Room E
1          20000.00   Nov  27    2020   1.0     C          1#/F Room C
2          11099.90   Nov  27    2020   7.0     E   TOWER 1 7/F Room E
3              0.00   Nov  27    2020   1.0    22         1#/F Room 22
4          17354.37   Nov  27    2020   1.0     B  BLOCK 8 1#/F Room B
```

HK School Net Number CSV File

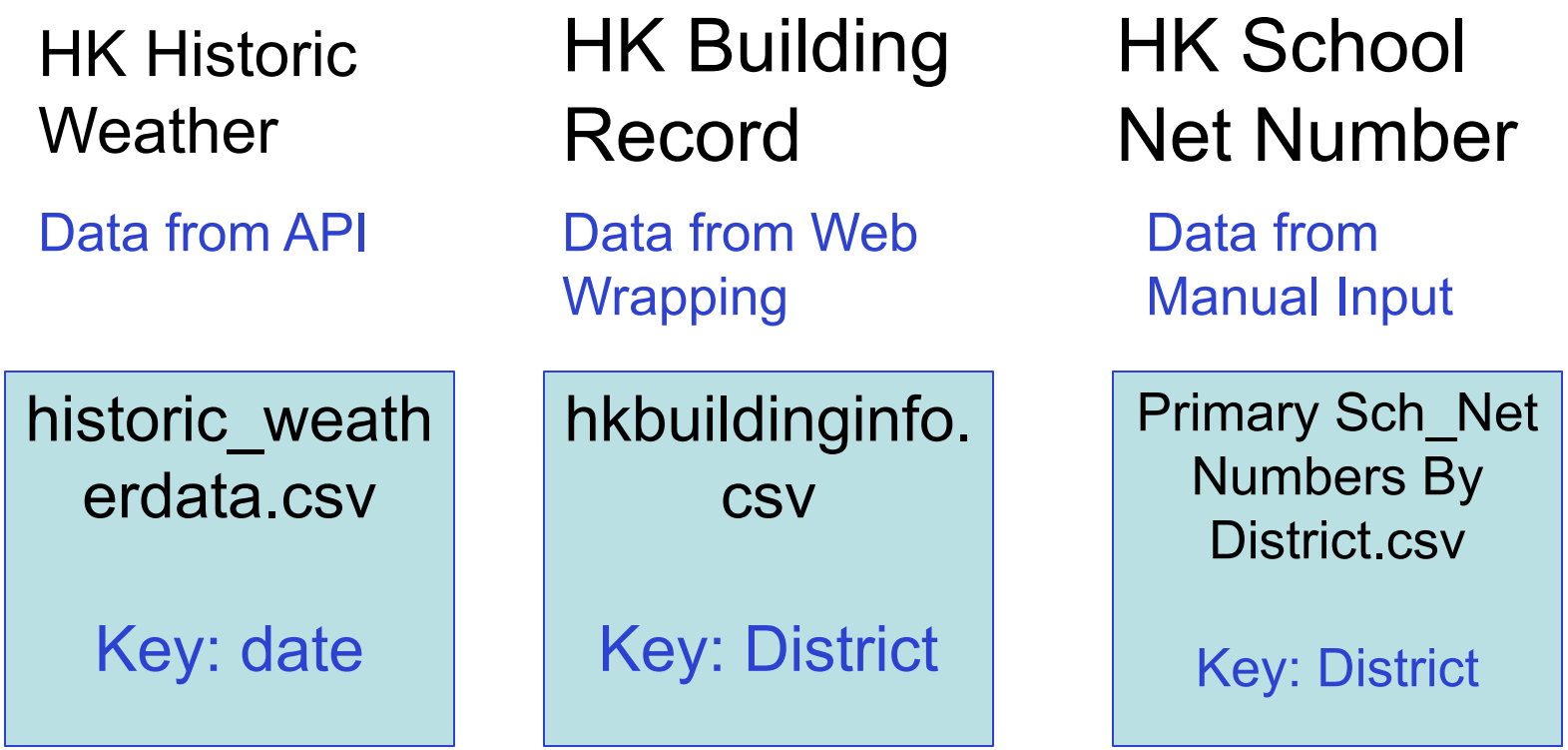
Data from Manual Input

District and POA School Net Look-up Table

https://www.edb.gov.hk/attachment/en/edu-system/primary-secondary/spa-systems/primary-1-admission/school-lists/District_Net_LookUpTable.pdf

區域 District	小一學校網編號 POA School Net Number
中西區 Central & Western District	11
灣仔 Wan Chai	12
北角/筲箕灣/柴灣 North Point/ Shau Kei Wan/ Chai Wan	14, 16
薄扶林/香港仔/黃竹坑/赤柱 Pok Fu Lam/ Aberdeen/ Wong Chuk Hang/ Stanley	18
油麻地/尖沙咀/旺角 Yau Ma Tei/ Tsim Sha Tsui/ Mong Kok	31, 32
何文田/紅磡/九龍城 Ho Man Tin/ Hung Hom/ Kowloon City	34, 35, 41
深水埗 Sham Shui Po	40
黃大仙 Wong Tai Sin	43, 45
九龍灣/牛頭角/觀塘/油塘 Kowloon Bay/ Ngau Tau Kok/ Kwun Tong/ Yau Tong	46, 48
荃灣 Tsuen Wan	62
葵涌/荔景/大窩口/青衣 Kwai Chung/ Lai King/ Tai Wo Hau/ Tsing Yi	64, 65, 66
屯門 Tuen Mun	70, 71
天水圍/元朗 Tin Shui Wai/ Yuen Long	72, 73, 74
上水/粉嶺/北區 Sheung Shui/ Fanling/ North District	80, 81, 83
大埔 Tai Po	84
沙田 Shatin	88, 89, 91
西貢/將軍澳 Sai Kung/ Tseung Kwan O	95
離島/東涌 Islands/ Tung Chung	96, 97, 98, 99

Master Data List Combination System Architecture



Master Data List Combination System Architecture

historic_weatherdata.csv

Key: date

date	wind	weather	maxtemp	mintemp	icon	psr
27/11/2020	North force 4 to 5.	Mainly cloudy with a few light rain patches at first. Becoming fin	20	15	51	Low
26/11/2020	North to northeast force 4, force 5 at first.	Fine. Rather cool in the morning. Very dry during the day.	19	13	92	Low
25/11/2020	Northeast force 4, east force 5 later.	Fine. Rather cool in the morning. Very dry during the day.	19	13	50	Low

Key: date

hkbuildinginfo.csv

date	withpre	id	rootid	fatherid	catid	catname	catfatherna	url_father	url_cat	source	contract	memo	price
27/11/2020	1	683333	2	79	2058	Bel Air Hei	Diamond H	https://data	https://data	Land Regis	Agreement	2.01E+13	$10M
27/11/2020	1	683332	2	71	2712	Fa Yuen Pl	Mong Kok	https://data	https://data	Land Regis	Agreement	2.01E+13	$4.28M
27/11/2020	1	683331	2	78	3094	Caldecott H	Yau Yat Ts	https://data	https://data	Land Regis	Agreement	2.01E+13	$11M
27/11/2020		683330	2	57	5035	Pang Ching	Wong Tai S	https://data	https://data	Land Regis	Agreement	2.01E+13	$4.9M
27/11/2020	1	683329	2	67	2564	Metro Harb	Tai Kok Ts	https://data	https://data	Land Regis	Agreement	2.01E+13	$7.15M
27/11/2020	1	683328	2	65	2521	Nob Hill	Mei Foo	https://data	https://data	Land Regis	Agreement	2.01E+13	$2.46M
27/11/2020	1	683327	2	62	5281	Chi Fung C	Sham Shui	https://data	https://data	Land Regis	Agreement	2.01E+13	$3.2M
27/11/2020	1	683326	2	56	2080	Telford Ga	Kowloon E	https://data	https://data	Land Regis	Agreement	2.01E+13	$7.68M
27/11/2020	1	683325	2	57	2104	Tsui Chuk	Wong Tai S	https://data	https://data	Land Regis	Agreement	2.01E+13	$3.67M
27/11/2020	1	683324	2	77	2548	Grand wate	To Kwa W	https://data	https://data	Land Regis	Agreement	2.01E+13	$9M
27/11/2020	1	683323	2	77	3073	Sun Cheon	To Kwa W	https://data	https://data	Land Regis	Agreement	2.01E+13	$4.2M
27/11/2020	1	683322	2	77	5461	Siu Fung C	To Kwa W	https://data	https://data	Land Regis	Agreement	2.01E+13	$4.9M
27/11/2020	1	683321	2	71	8587	SKYPARK	Mong Kok	https://data	https://data	Land Regis	Agreement	2.01E+13	$7.35M
27/11/2020	1	683320	2	127	5263	Aria	Ngau Chi W	https://data	https://data	Land Regis	Agreement	2.01E+13	$8.25M
27/11/2020	1	683319	2	60	5948	One Victor	Ho Man Ti	https://data	https://data	Land Regis	Agreement	2.01E+13	$8.83M
27/11/2020	1	683318	2	77	22478	Wan Tung	To Kwa W	https://data	https://data	Land Regis	Agreement	2.01E+13	$5.43M
27/11/2020		683317	2	56	2082	Richland G	Kowloon E	https://data	https://data	Land Regis	Agreement	2.01E+13	$5.67M
27/11/2020		683316	2	62	2429	Cheung Ho	Sham Shui	https://data	https://data	Land Regis	Agreement	2.01E+13	$3.73M
27/11/2020	1	683315	2	60	2243	Dragon Vie	Ho Man Ti	https://data	https://data	Land Regis	Agreement	2.01E+13	$9.8M

Primary Sch_Net Numbers By District.csv

Key: District

Sch_Net_No	District Areas	Districts
1	CENTRAL	Hong Kong Island
1	THE PEAK	Hong Kong Island
1	SHEUNG WAN	Hong Kong Island
1	SAI YING PUN	Hong Kong Island
1	SHEK TONG TSUI	Hong Kong Island
1	KENNEDY TOWN	Hong Kong Island
1	WAN CHAI	Hong Kong Island
1	CAUSEWAY BAY	Hong Kong Island
1	SO KON PO	Hong Kong Island
1	HAPPY VALLEY	Hong Kong Island
1	TAI HANG	Hong Kong Island
1	JARDINE?? LOOKOUT	Hong Kong Island
1	NORTH POINT	Hong Kong Island
1	QUARRY BAY	Hong Kong Island
1	LEI KING WAN	Hong Kong Island
1	SHAU KEI WAN	Hong Kong Island
1	SAI WAN HO	Hong Kong Island
1	SHEK O	Hong Kong Island

Combining Weather, Building Information and School Net Numbers

#Combine Weather Data with Building Data

```
hkbuildingdata['date'] = pd.to_datetime(hkbuildingdata['date']).dt.strftime('%Y-%m-%d')
merged_df = hkbuildingdata.merge(weatherdata[['date', 'wind', 'maxtemp', 'mintemp']], on='date', how='left')
```

#Combine Original Data with School District Data

```
schoolnetnumber_df = pd.read_csv("Primary Sch_Net Numbers By District.csv")
# Normalize the 'District' column in both DataFrames to lowercase
merged_df['District'] = merged_df['District'].str.lower()
schoolnetnumber_df['District'] = schoolnetnumber_df['District Areas'].str.lower()
# Merge schooldistrict_df with districtcode_df on the normalized 'District' column
merged_df = merged_df.merge(schoolnetnumber_df[['District', 'Sch_Net_No']], on='District', how='left')
# Create the 'District_Code' column based on the logic provided
merged_df['School_Netnumber'] = merged_df['Sch_Net_No'].fillna(0).astype(int)
# Save the merged_df into csv with name “combined_building_withweather_school.csv”
merged_df.to_csv('combined_building_withweather_school.csv', index=False)
```

[27]:

	date	id	Building Name	District	url_father	url_cat	contract	memo	price	price_va
0	2020-11-27	683333	Bel Air Heights	diamond hill	https://data.28hse.com/en/kl/diamond-hill	https://data.28hse.com/en/kl/diamond-hill/2058...	Agreement	20112702480015	$10M	10000(
1	2020-11-27	683332	Fa Yuen Plaza	mong kok	https://data.28hse.com/en/kl/mong-kok	https://data.28hse.com/en/kl/mong-kok/2712_fa-...	Agreement	20112702470044	$4.28M	4280(
2	2020-11-27	683331	Caldecott Hill	yau yat tsuen	https://data.28hse.com/en/kl/yau-yat-tsuen	https://data.28hse.com/en/kl/yau-yat-tsuen/309...	Agreement	20112702440018	$11M	11000(
3	2020-11-27	683330	Pang Ching Court	wong tai sin	https://data.28hse.com/en/kl/wong-tai-sin	https://data.28hse.com/en/kl/wong-tai-sin/5035...	Agreement	20112702410017	$4.9M	4900(
4	2020-11-27	683329	Metro Harbour View	tai kok tsui	https://data.28hse.com/en/kl/tai-kok-tsui	https://data.28hse.com/en/kl/tai-kok-tsui/2564...	Agreement	20112702400032	$7.15M	7150(

5 rows × 29 columns

Exploratory Data Analysis(ETL) and Data Visualization

Exploratory Data Analysis and Visualization

```
#filter the data with period "2020-1-1" to "2020-12-31'
# Create a full date column from 'month', 'day', and 'date_y'
hkbuildingdata['full_date'] = pd.to_datetime(hkbuildingdata['month'] + ' ' + hkbuildingdata['day'].astype(str) + ' ' +
                                             hkbuildingdata['date_y'].astype(str))
# Filter the DataFrame for the date range "2020-01-01" to "2020-12-31"
start_date = '2020-01-01'
end_date = '2020-12-31'
covidfiltered_data = hkbuildingdata[(hkbuildingdata['full_date'] >= start_date) & (hkbuildingdata['full_date'] <= end_date)]

# Display the first few rows of the filtered DataFrame
print(covidfiltered_data.head(5))
```

```
         date      id       Building Name       District  \
0  2020-11-27  683333     Bel Air Heights   diamond hill
1  2020-11-27  683332       Fa Yuen Plaza       mong kok
2  2020-11-27  683331      Caldecott Hill  yau yat tsuen
3  2020-11-27  683330   Pang Ching Court   wong tai sin
4  2020-11-27  683329  Metro Harbour View   tai kok tsui

                              url_father  \
0   https://data.28hse.com/en/kl/diamond-hill
1       https://data.28hse.com/en/kl/mong-kok
2  https://data.28hse.com/en/kl/yau-yat-tsuen
3   https://data.28hse.com/en/kl/wong-tai-sin
4   https://data.28hse.com/en/kl/tai-kok-tsui

                                      url_cat   contract  \
0  https://data.28hse.com/en/kl/diamond-hill/2058...  Agreement
1  https://data.28hse.com/en/kl/mong-kok/2712_fa-...  Agreement
2  https://data.28hse.com/en/kl/yau-yat-tsuen/309...  Agreement
3  https://data.28hse.com/en/kl/wong-tai-sin/5035...  Agreement
4  https://data.28hse.com/en/kl/tai-kok-tsui/2564...  Agreement
```

```
# Filter the DataFrame for the date range "2019-01-01" to "2019-12-31"
start_date = '2019-01-01'
end_date = '2019-12-31'
filtered_data = hkbuildingdata[(hkbuildingdata['full_date'] >= start_date) & (hkbuildingdata['full_date'] <= end_date)]

# Display the first few rows of the filtered DataFrame
print(filtered_data.head(5))
```

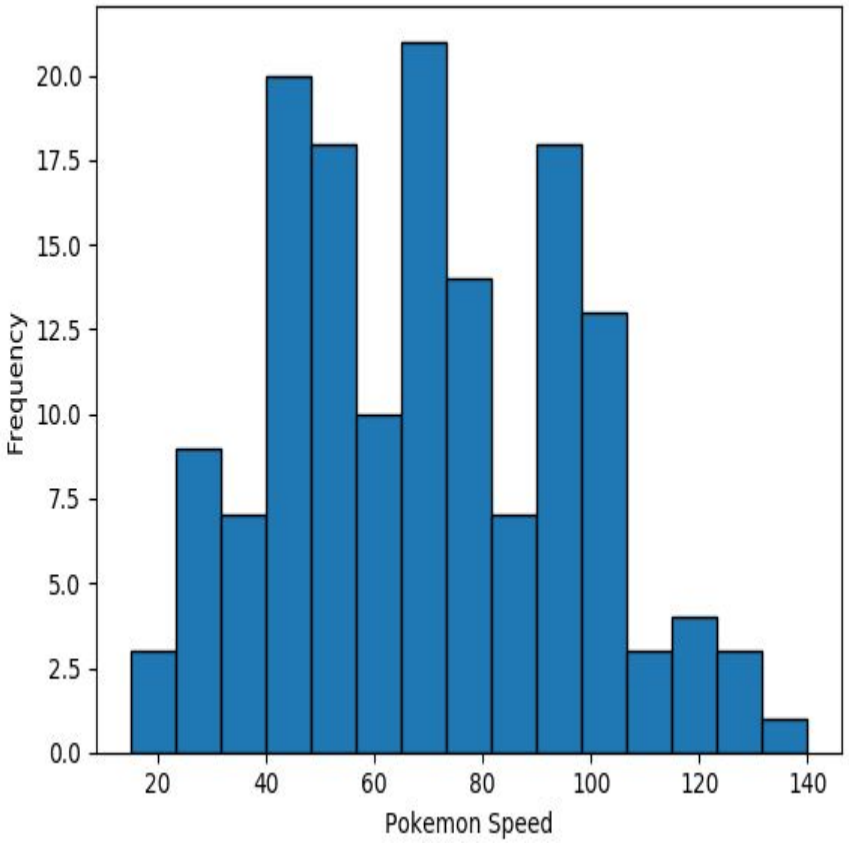
Exploratory Data Analysis(ETL) and Data Visualization

We could spend time staring at these numbers, but that is unlikely to offer us any form of insight.

We could begin by conducting all of our statistical tests.

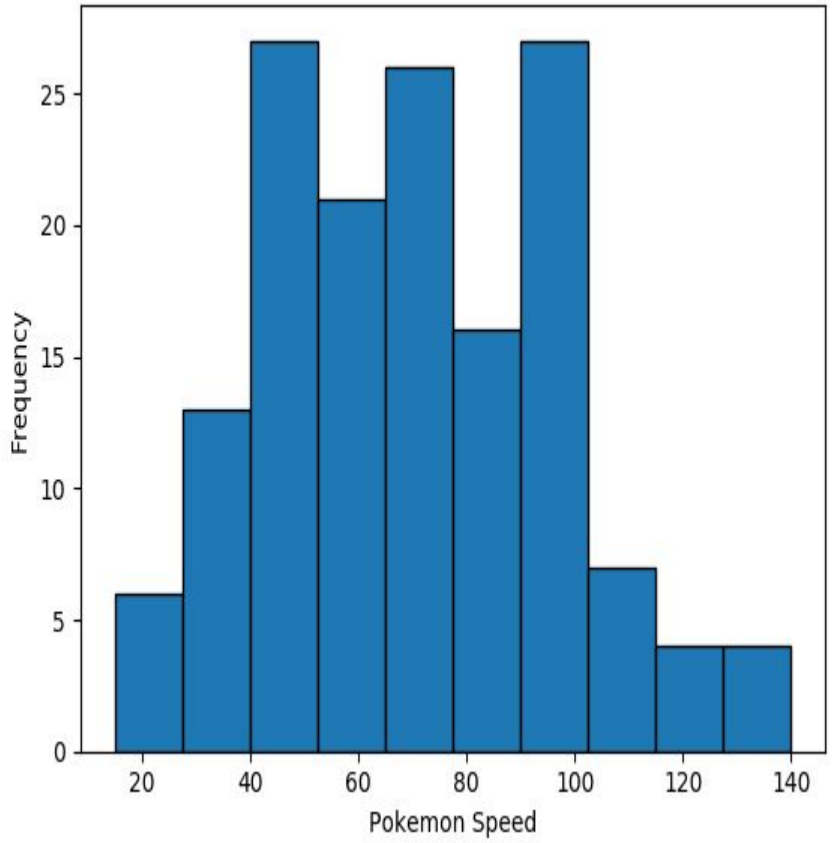
However, a good field commander never goes into battle without first doing a recognisance of the terrain…

This is exactly what EDA is for…



Plotting a histogram in Python

```
g = plt.hist(df1['Speed'], histtype='bar', ec='black',)
g = plt.xlabel('Pokemon Speed')
g = plt.ylabel('Frequency')
plt.show()
```

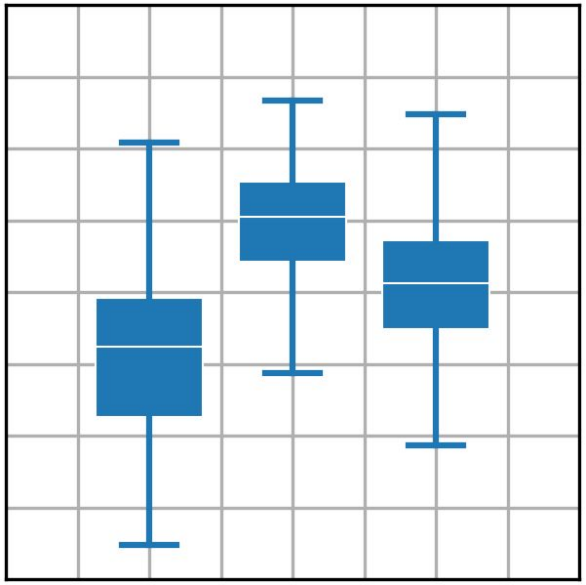


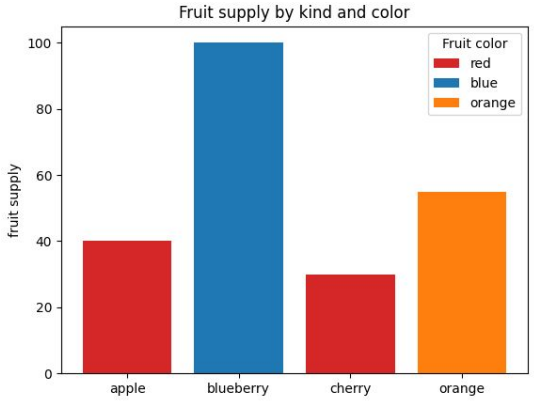
Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Features:

1. Create publication quality plots.
2. Make interactive figures that can zoom, pan, update.
3. Customize visual style and layout.
4. Export to many file formats.
5. Embed in JupyterLab and Graphical User Interfaces.
6. Use a rich array of third-party packages built on Matplotlib.



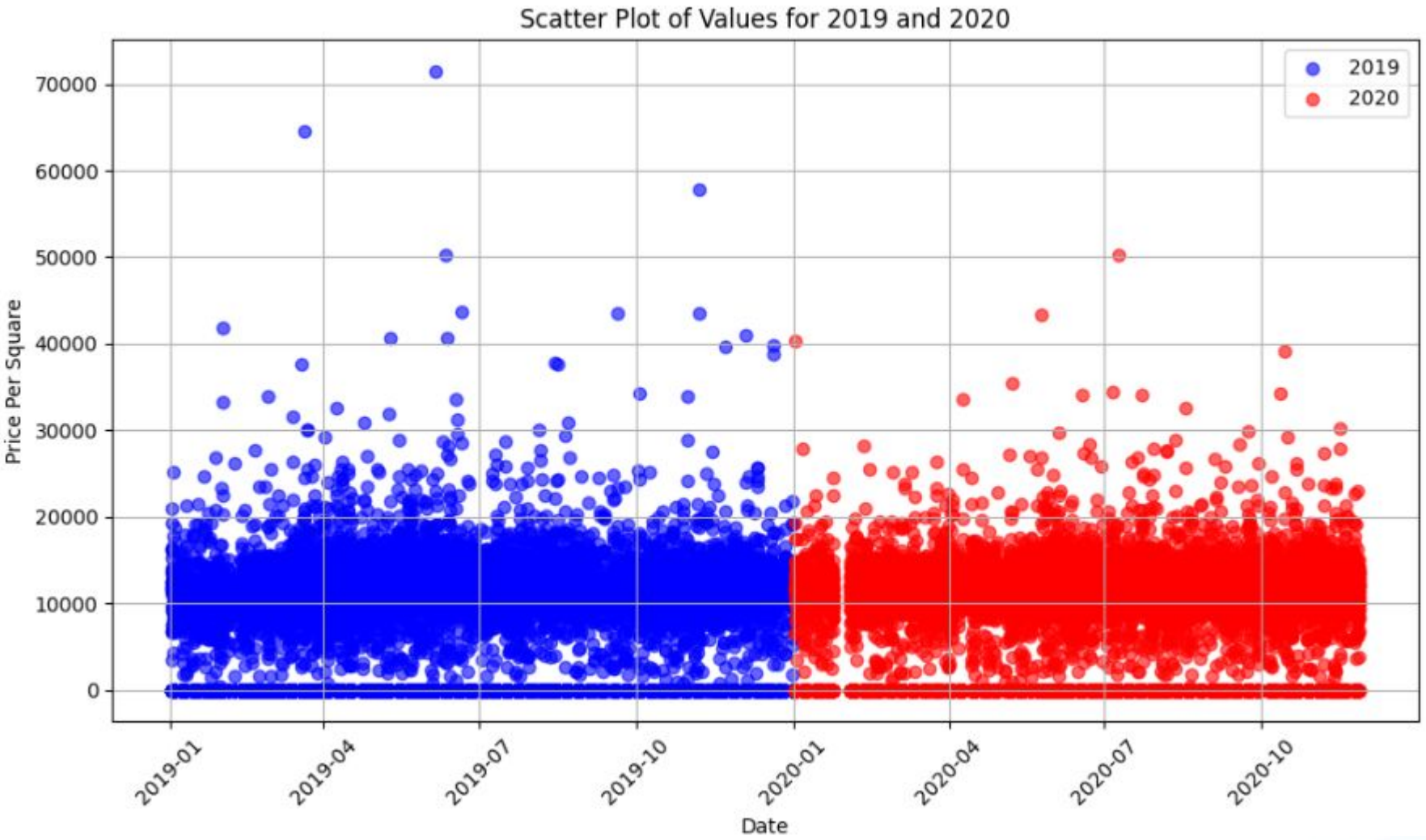


https://matplotlib.org/

Exploratory Data Analysis(ETL) and Visualization

```
import matplotlib.pyplot as plt

# Create a scatter plot
plt.figure(figsize=(10, 6))

# Plot for 2019
plt.scatter(filtered_data['full_date'], filtered_data['Price Per Square'], color='blue', label='2019', alpha=0.6)

# Plot for 2020
plt.scatter(covidfiltered_data['full_date'], covidfiltered_data['Price Per Square'], color='red', label='2020', alpha=0.6)

# Adding titles and labels
plt.title('Scatter Plot of Values for 2019 and 2020')
plt.xlabel('Date')
plt.ylabel('Price Per Square')
plt.xticks(rotation=45)
plt.legend()
plt.grid()

# Show the plot
plt.tight_layout()
plt.show()
```

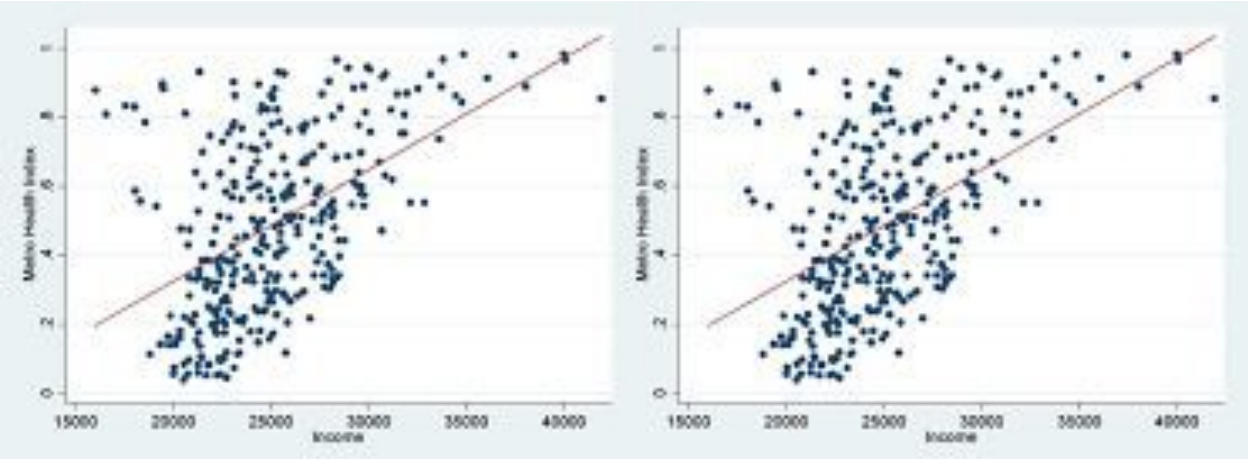
Exploratory Data Analysis(ETL) and Visualization



Exploratory Data Analysis(ETL) and Data Visualization

Tasks 1 - Housing Analysis

1. Change to different period from 2019-2020 and 2020-2021 to 2014-2017, 2018-2021
2. Filter the data with school net number “34”, “35”, or others
3. Visualise the result in Scatter Plot
4. Change the chart color, legend.



Exploratory Data Analysis (EDA)

Exploring your data is a crucial step in data analysis. It involves:

Organising the data set

Plotting aspects of the data set

Maybe producing some numerical summaries; central tendency and spread, etc.

"Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone."
- John Tukey.

Exploratory Data Analysis (EDA)

Why?

- EDA encompasses the "*explore* data" part of the data science process
- EDA is crucial but often overlooked:
  - If your data is bad, your results will be bad
  - Conversely, understanding your data well can help you create smart, appropriate models

Exploratory Data Analysis (EDA)

What?

1. Store data in data structure(s) that will be convenient for exploring/processing
   (Memory is fast. Storage is slow)
2. Clean/format the data so that:
   - Each row represents a single object/observation/entry
   - Each column represents an attribute/property/feature of that entry
   - Values are numeric whenever possible
   - Columns contain atomic properties that cannot be further decomposed*

* Unlike food waste, which can be composted.
  Please consider composting food scraps.

Exploratory Data Analysis (EDA)

What? (continued)

3. Explore **global** properties: use histograms, scatter plots, and aggregation functions to summarize the data
4. Explore **group** properties: group like-items together to compare subsets of the data (are the comparison results reasonable/expected?)

This process transforms your data into a format which is easier to work with, gives you a basic overview of the data's properties, and likely generates several questions for you to follow-up in subsequent analysis.

EDA: with Pandas!

Kung Fu Panda is property of DreamWorks and Paramount Pictures

Lecture Outline

- Exploratory Data Analysis (EDA):
  - Without Pandas (part 1) – These slides
  - With Pandas (part 2) – Mostly Jupyter Notebook
- Data concerns (part 3) – These slides
- Web Scraping with Beautiful Soup (part 4) – Mix

Common Panda functions

Grouping/Splitting/Aggregating:

- groupby(), .get_groups()
- .merge()
- .concat()
- .aggegate()
- .append()

Lecture Outline

- Exploratory Data Analysis (EDA):
  - Without Pandas (part 1) – These slides
  - With Pandas (part 2) – Mostly Jupyter Notebook
- Data concerns (part 3) – These slides
- Web Scraping with Beautiful Soup (part 4) – Mix

Data Concerns

When determining if a dataset is sound to use, it can be useful to think about these four questions:

- Did it come from a trustworthy, authoritative source?
- Is the data a complete sample?
- Does the data seem correct?
- **(optional)** Is the data stored efficiently or does it have redundancies?

Data Concerns: the format

- Often times, there may not exist a single dataset that contains all of the information we are interested in.
- May need to merge existing datasets
- Important to do so in a sound and efficient format

Data Concerns: the format

For example, say we have two datasets:

Dataset 1

Top 200 most-frequent streams per day (for June 2019)

SpotifySongID, # of Streams, Date

200

2789179, 42003, 06-01

⋮

3819390, 89103, 06-01

200

4492014, 52923, 06-02

⋮

8593013, 189145, 06-02

6,000 x 3

Dataset 2

Top 50 most streamed in 2019, so far

SpotifySongID, Artist, Track, [10 acoustic features]

50

2789179, Billie Eilish, bad guy, 3.2, 5.9, …

⋮

3901829, Outkast, Elevators, 9.3, 5,1, …

50 x 13

Data Concerns: the format

For example, say we have two datasets:

Dataset 1

Top 200 most-frequent streams per day (for June 2019)

SpotifySongID, # of Streams, Date

200

2789179, 42003, 06-01

▪

▪

3819390, 89103, 06-01

200

4492014, 52923, 06-02

▪

▪

8593013, 189145, 06-02

6,000 x 3

Dataset 2

Top 50 most streamed in 2019, so far

SpotifySongID, Artist, Track, [10 acoustic features]

50

2789179, Billie Eilish, bad guy, 3.2, 5.9, …

▪

▪

3901829, Outkast, Elevators, 9.3, 5,1, …

50 x 13

We are interested in determining if songs with high danceability are more popular during the weekends of June than weekdays in June. What should our merged table look like? Concerns?

Data Concerns: the format

This is wasteful, as it has 10 acoustic features, artist, and track repeated many times for each unique song.

Datasets Merged (poorly)

SpotifySongID, # of Streams, Date,Artist, Track, [10 acoustic features]

200

2789179, 42003, 06-01 Billie Eilish, bad guy, 3.2, 5.9, …

3819390, 89103, 06-01 Outkast, Elevators, 9.3, 5,1, …

200

4492014, 52923, 06-02

8593013, 189145, 06-02

6,000 x 15 □ 90,000 cells

Data Concerns: the format

Some rows may have null values for # of Streams (if the song wasn't popular in June)

Datasets Merged (better)

SpotifySongID, Artist, Track, [10 acoustic features], 06-01 Streams, 06-02 Streams

50

2789179, Billie Eilish, bad guy, 3.2, 5.9, ..., 42003, 42831, 43919

⋮

3901829, Outkast, Elevators, 9.3, 5,1, ... 29109, 27193, 25982

50 x 70 □ 3,500 cells

Data Concerns: the format

- Is the data correctly constructed (or are values wrong)?
- Is there redundant data in our merged table?
- Missing values?

How To Do Data Mining: Machine Learning Techniques

Features / Attributes

Are used to train Learning algorithm

- Sales Price
- Saleable Area
- Number of Bedrooms
- Year built
- (Refurbished) Schools in the area
- Transportation (MTR, buses)

Address: 270-280 QUEEN'S ROAD WEST
Price per sqft.: 17,841
Qty transaction: 2
Year: 2017

Using a Home Valuation Prediction Model as an example

How To Do Data Mining: Machine Learning Techniques

Machine Learning

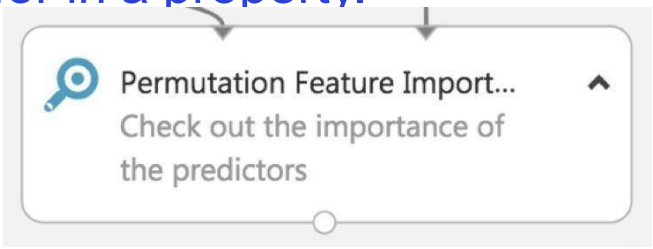
Feature	Score
SaleableArea	0.858545
Eff_Build_Ages	0.166877
Bus_Route	0.076519
Parks	0.019011

Choosing a 'useful' or ***significant*** feature (predictors) ***makes*** or ***breaks*** a data science project.

Domain experts, property agents will tell you what property buyers value in a home.

Data scientists would ***NOT*** have a clue what home buyers look for in a property.

But not all is lost!



Data Processing & Analysis - What Is Machine Learning?

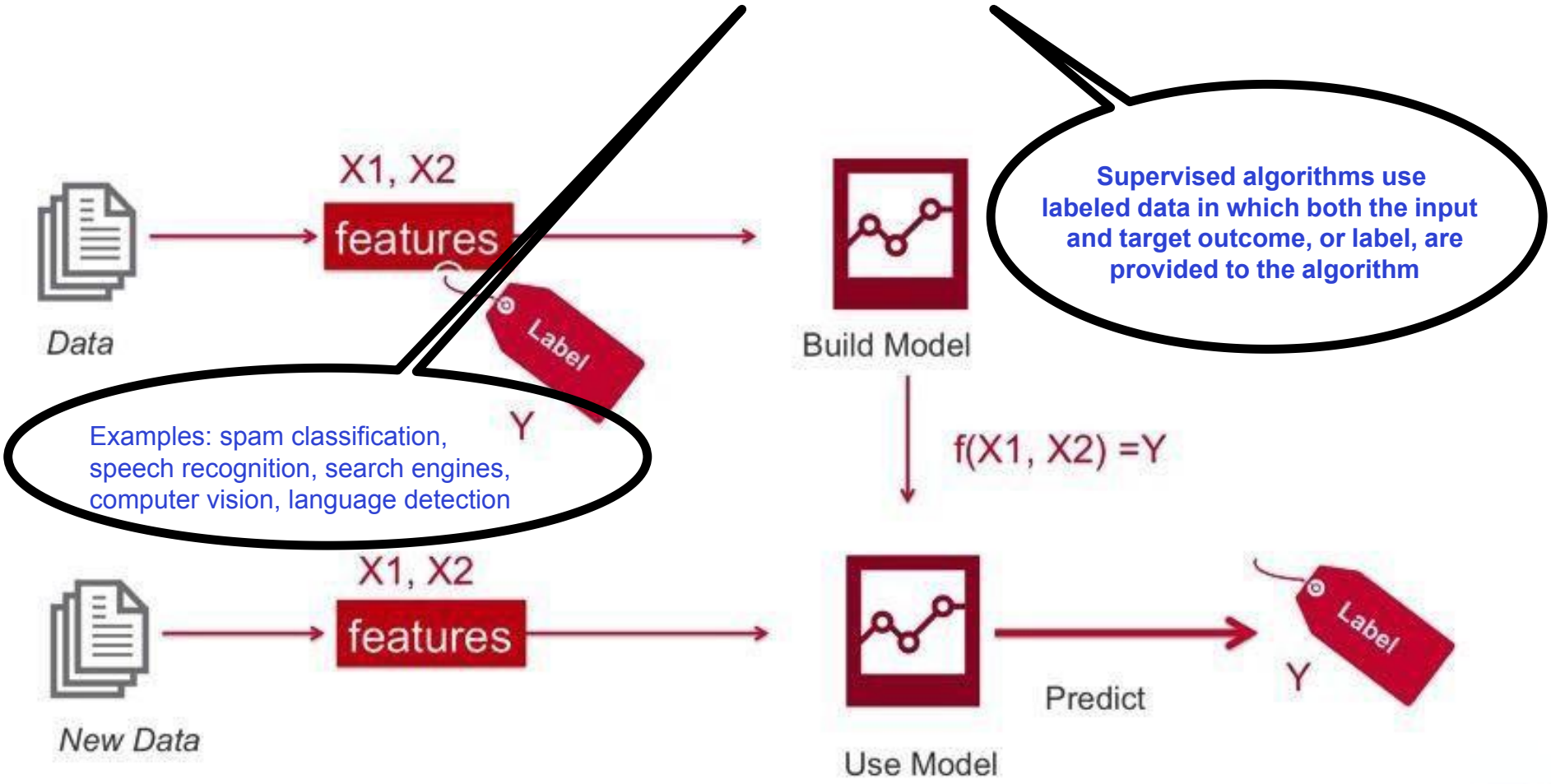
Supervised Learning	Unsupervised Learning
Used as an advanced form of *predictive modeling*	Used either as a form of automated data analysis or automated signal extraction
Each observation must be labeled with a "correct answer"	Unlabeled data has no predetermined "correct answer"
Only then can you build a predictive model because you must tell the algorithm what's "correct" while training it (hence, "supervising" it)	You'll allow the algorithm to directly learn patterns from the data (without "supervision")
Regression is the task for modeling continuous target variables	Clustering is the most common unsupervised learning task, and it's for *finding groups* within your data
Classification is the task for modeling categorical (a.k.a. "class") target variables	

Source: https://dzone.com/articles/demystifying-ai-machine-learning-and-deep-learning

Data Processing & Analysis
Supervised Learning: Classification For Debit Card Fraud

Source:
https://dzone.com/articles/demystifying-ai-machine-learning-and-deep-learning

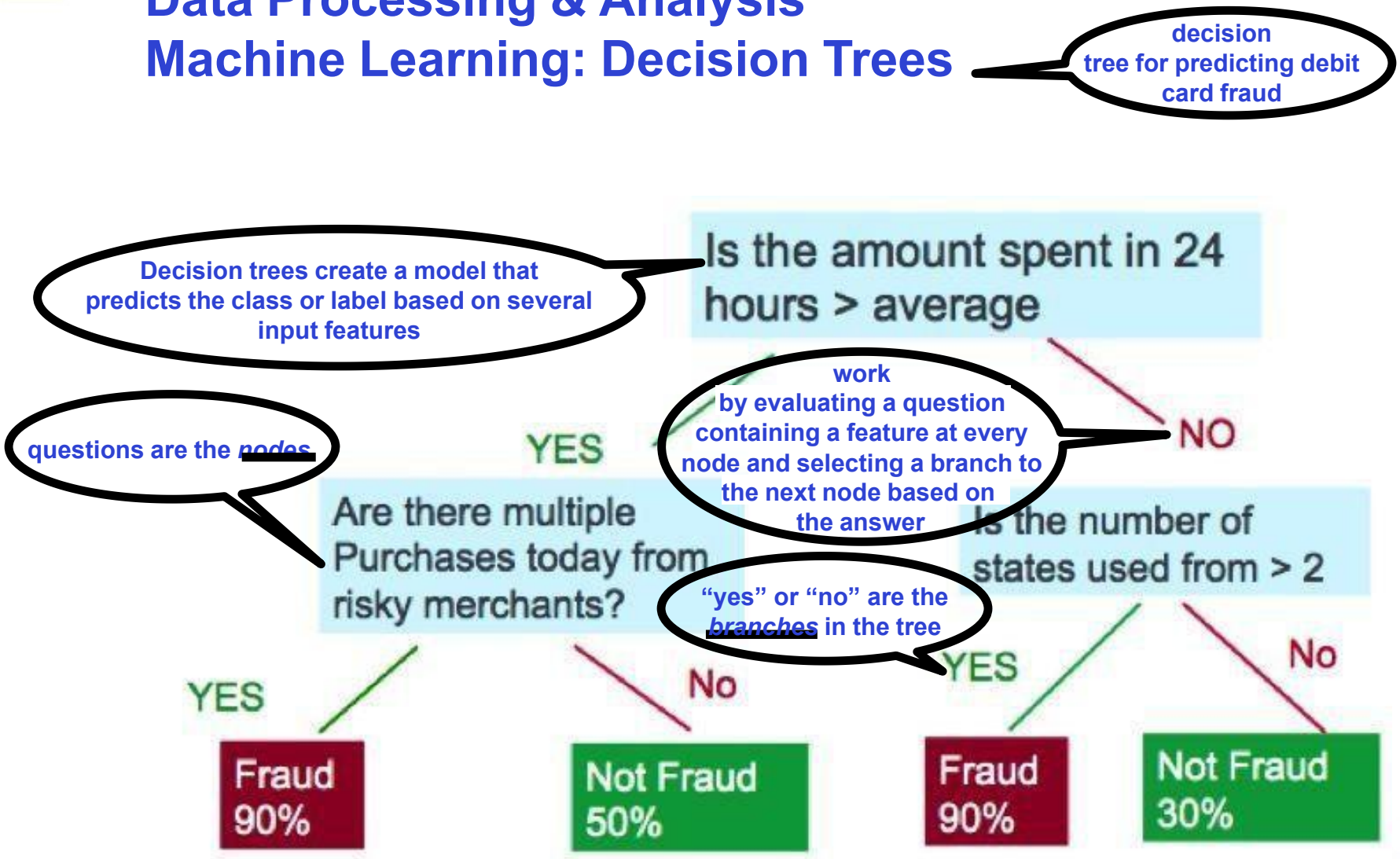
Data Processing & Analysis
Machine Learning: Supervised Learning



Source: https://dzone.com/articles/demystifying-ai-machine-learning-and-deep-learning

82

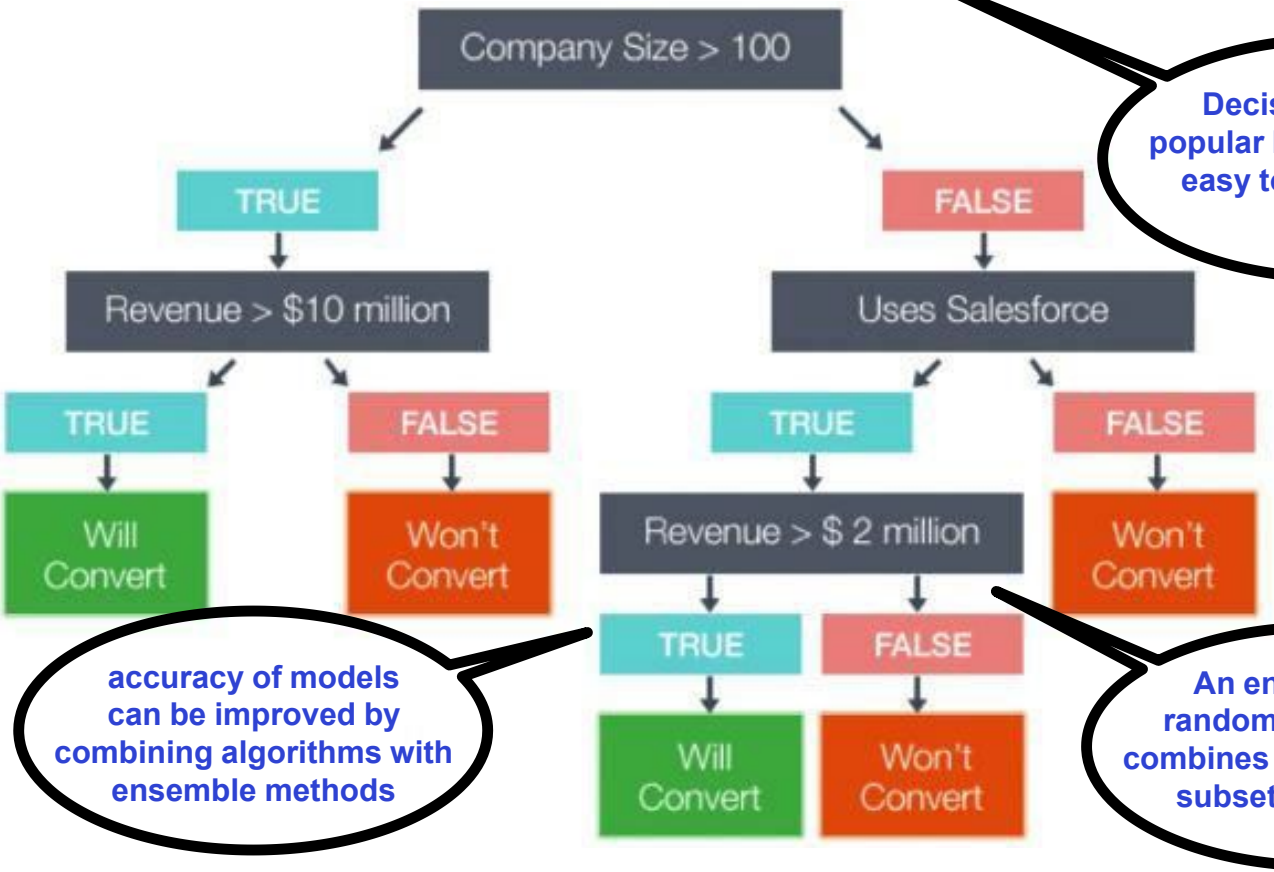
Data Processing & Analysis
Machine Learning: Decision Trees



Source: https://dzone.com/articles/demystifying-ai-machine-learning-and-deep-learning

Data Processing & Analysis

Decision Trees: Ensemble Random Forests



Source: http://dataconomy.com/2015/05/predictive-machine-learning-behind-the-scenes-at-fliptop-and-predictions-for-the-future-of-martech/

How To Do Data Mining: Machine Learning Techniques

Features / Attributes

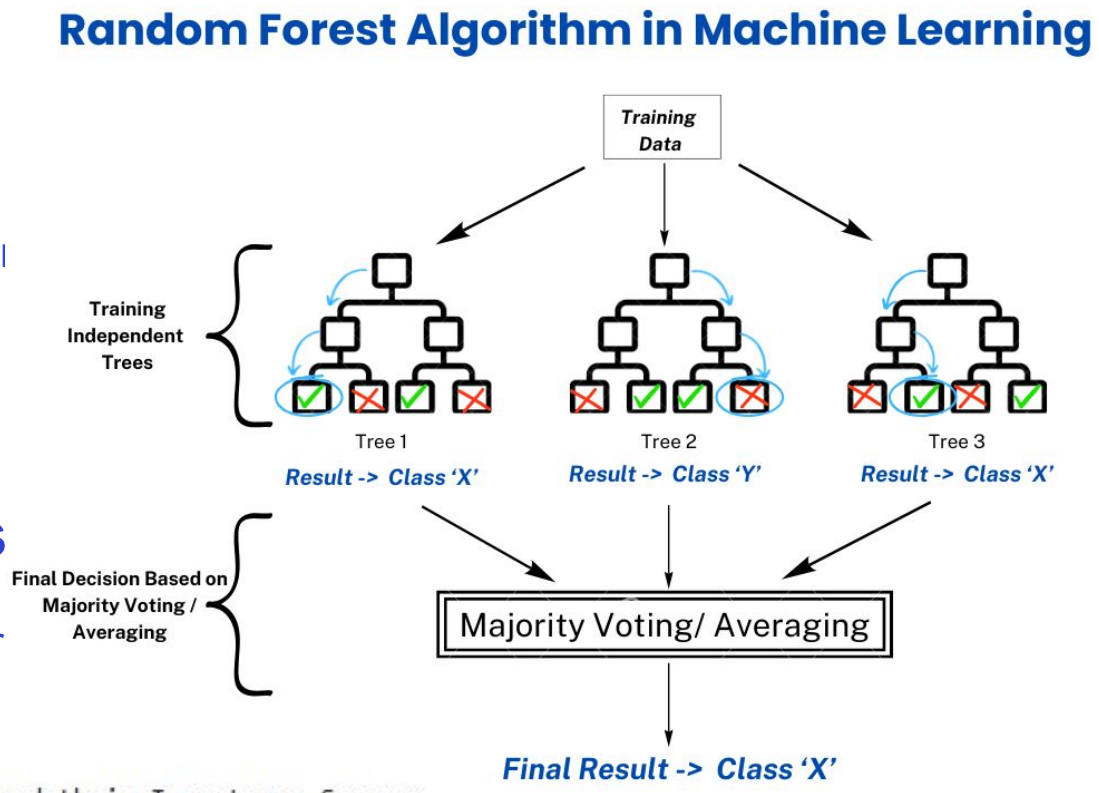
Set X using RandomForestRegressor

Parameter

1. **District**
2. **Floor**
3. **School Net Numbers**

Set y using RandomForestRegressor

Outcome - Price Per Square




```
Top 5 Features and their Importance Scores:
            Feature  Importance
0          District    0.728459
2  School_Netnumber    0.199190
1             Floor    0.072351
```

Machine Learning Random Forest Model training

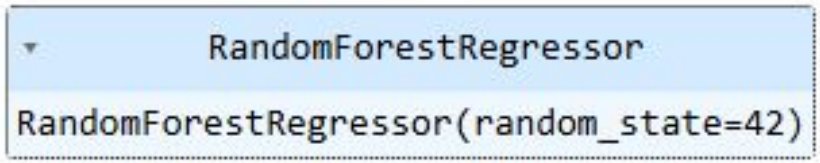
#This code is going to train an random forest model to find top important features

```
#Step 1 : Import necessary library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import plot_tree
import joblib  # Import joblib for saving the model

#Step 2 : import data for machine learning model training
data = pd.read_csv("combined_building_withweather_school.csv")

# #Step 3 : Machine Learning Model Implementation
# Encode categorical variables
label_encoder = LabelEncoder()
data['District'] = label_encoder.fit_transform(data['District'])
data['School_Netnumber'] = label_encoder.fit_transform(data['School_Netnumber'])

# Define features and target variable
X = data[['District', 'Floor', 'School_Netnumber']]  # Features
y = data['Price Per Square']  # Target variable

# Check the distribution of the target variable
print(y.value_counts())
print(X)
```

```
Price Per Square
0.00        30508
10000.00      217
8333.33       109
8000.00        81
12500.00       70
             ...
8554.05          1
13764.04         1
10443.86         1
15421.30         1
4672.90          1
Name: count, Length: 31331, dtype: int64
       District  Floor  School_Netnumber
0             1    2.0                 8
1            16    1.0                 2
2            31    7.0                 5
3            28    1.0                 7
4            24    1.0                 2
...         ...    ...               ...
79995        6     2.0                 9
79996        9     9.0                 6
79997        3     4.0                 4
79998        6     1.0                 9
79999       26     3.0                 1

[80000 rows x 3 columns]
```

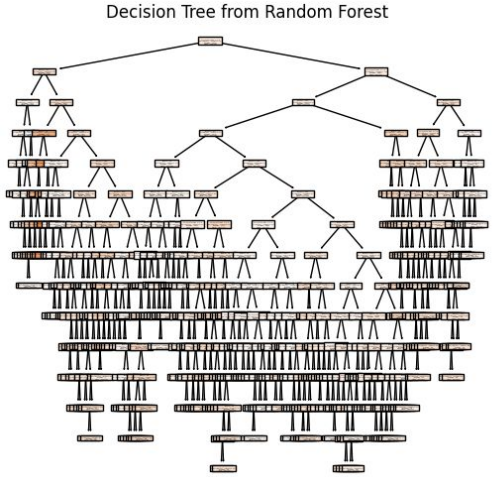
Machine Learning Random Forest Model training

```
# Step 3: Split the data into training and testing sets with stratification
# Ensure the target variable is continuous
print(f"Target variable type: {y.dtype}")

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Train the Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Step 6: Display a decision tree from the Random Forest with reduced size
plt.figure(figsize=(6, 6))  # Reduced by 10%
plot_tree(rf_model.estimators_[5], feature_names=X.columns.tolist(), filled=True)
plt.title("Decision Tree from Random Forest")
plt.show()
```



Machine Learning Random Forest Model training

```
# Display feature importances
importances = rf_model.feature_importances_
feature_names = X.columns.tolist()
feature_importances = pd.DataFrame({'Feature': feature_names, 'Importance':
    importances})

# Sort by importance and display the top 5 features
top_features = feature_importances.sort_values(by='Importance',
    ascending=False).head(5)
print("Top 5 Features and their Importance Scores:")
print(top_features)

# Step 5: Save the trained model to a file
model_filename = 'housing_valuation_model.joblib'  # Specify the filename
joblib.dump(rf_model, model_filename)  # Save the model
print(f'Model saved to {model_filename}')
```




```
Top 5 Features and their Importance Scores:
            Feature  Importance
0          District    0.728459
2  School_Netnumber    0.199190
1             Floor    0.072351
```

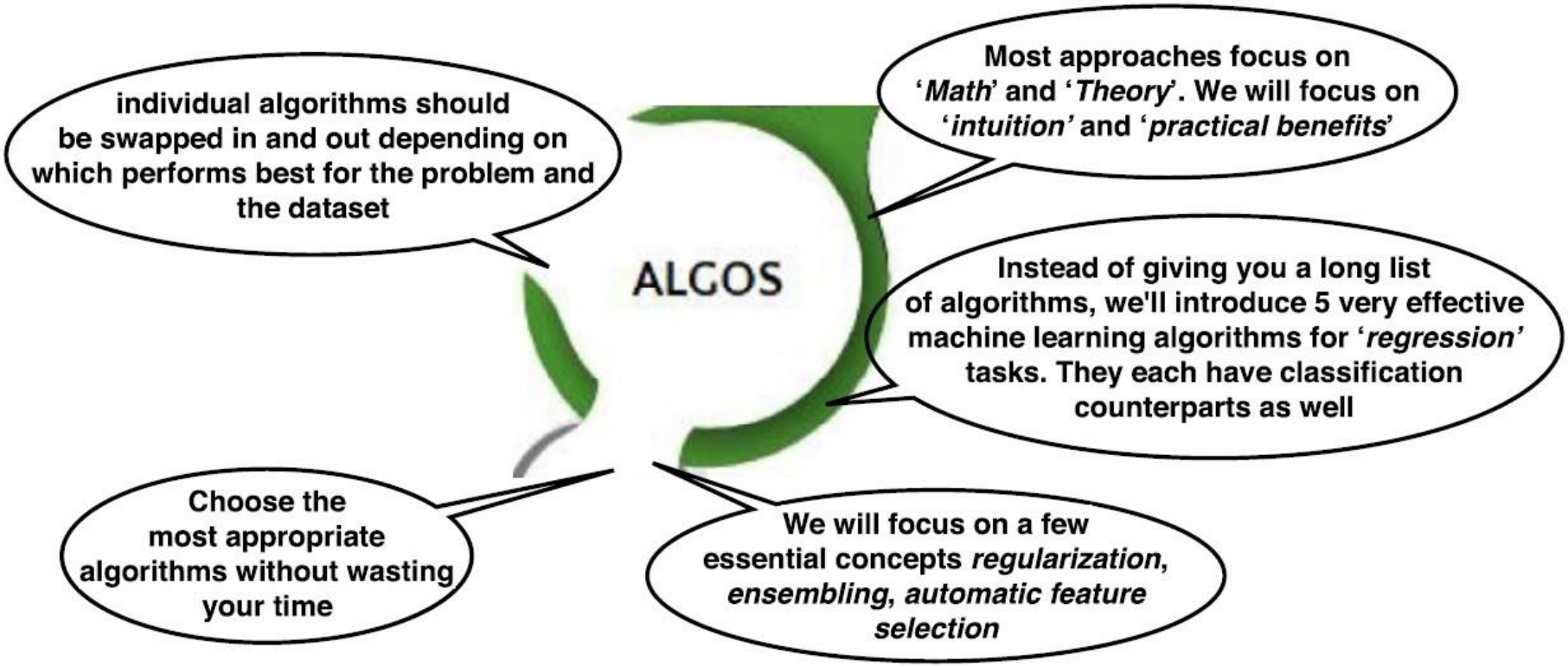
Machine Learning Random Forest Inference Model : Display the predicted housing price per square

```
#add necessary library
import joblib

#Define function to get user input
def get_user_input():
    district = float(input("Enter District Number : "))
    schoolnet_no = float(input("Enter School Net Number: "))
    floor = float(input("Enter Floor "))

    return [[district, schoolnet_no, floor]]
    return pd.DataFrame([[district, schoolnet_no, floor]], columns=['District', 'School_Netnumber', 'Floor'])

#Define function for loading random forest model
def load_model(model_path):
    return joblib.load(model_path)

#Define function to predict housing price per square based on 3 data input
def predict(model, input_data):
    return model.predict(input_data)
```

Machine Learning Random Forest Inference Model : Display the predicted housing price per square

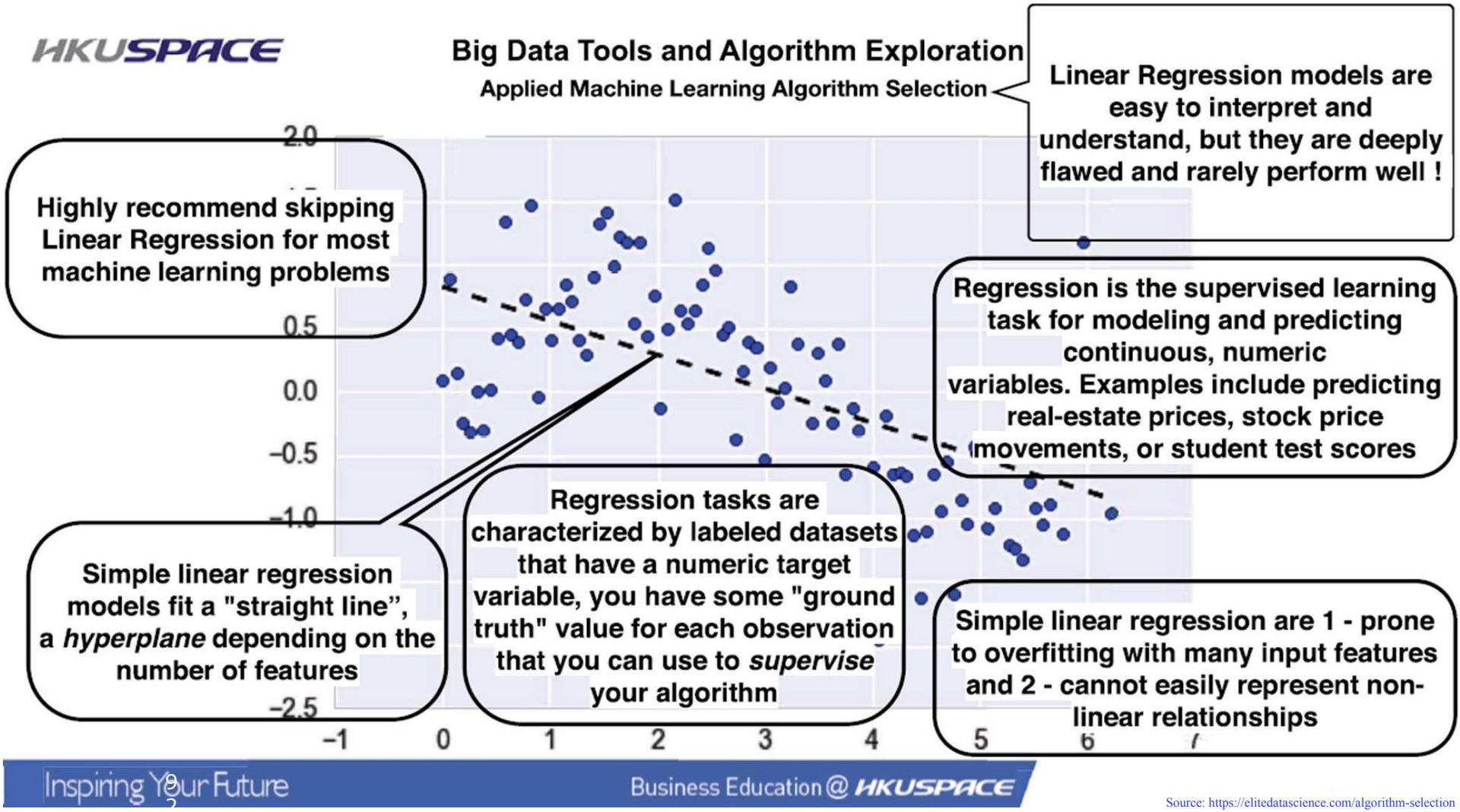
```
#Main Program to display the predicted housing price per square
input_data = get_user_input()
model_path = 'housing_valuation_model.joblib'
model = load_model(model_path)
result = predict(model, input_data)
print(f"Predicted Hosing Price Per Square is : HKD ${round(result[0],2)}")
```

Big Data Tools and Algorithm Exploration
Applied Machine Learning Algorithm Selection Overview



Source: https://elitedatascience.com/algorithm-selection

Big Data Tools and Algorithm Exploration
Applied Machine Learning Algorithm Selection



Source: https://elitedatascience.com/algorithm-selection

Big Data Tools and Algorithm Exploration
Applied Machine Learning Algorithm Selection

Random forests train a large number of "strong" decision trees and combine their predictions through bagging

Each tree is only trained on a random subset of observations (a process called resampling)

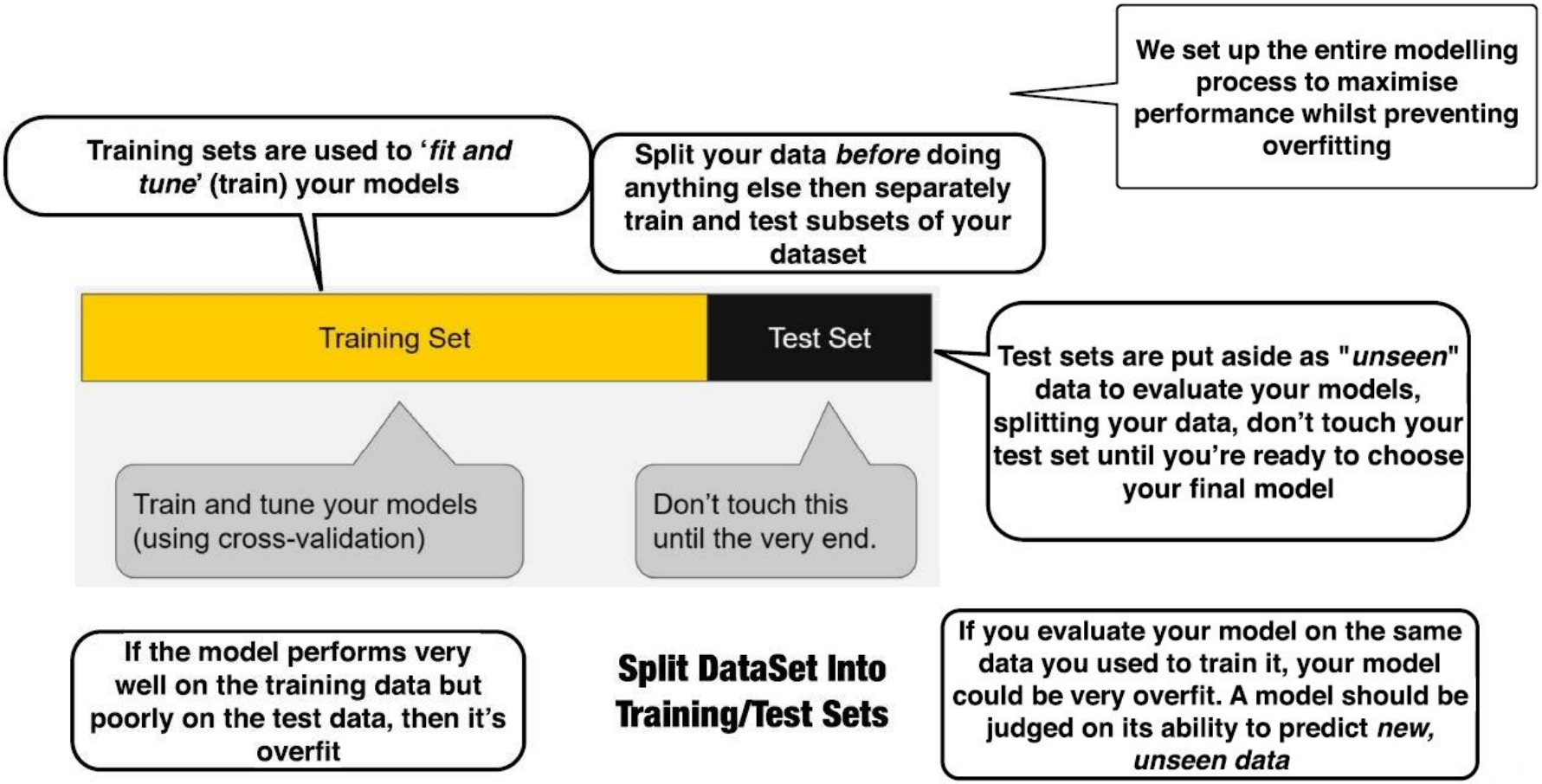
Random Forests

Each tree is only allowed to choose from a random subset of features to split on (leading to feature selection)

random forests tend to perform very well right out of the box often beat many other models that take up to weeks to develop, and are the perfect "*swiss-army-knife*" algorithm that almost always gets good results

Source: https://elitedatascience.com/algorithm-selection

Big Data Tools and Algorithm Exploration
Model Training

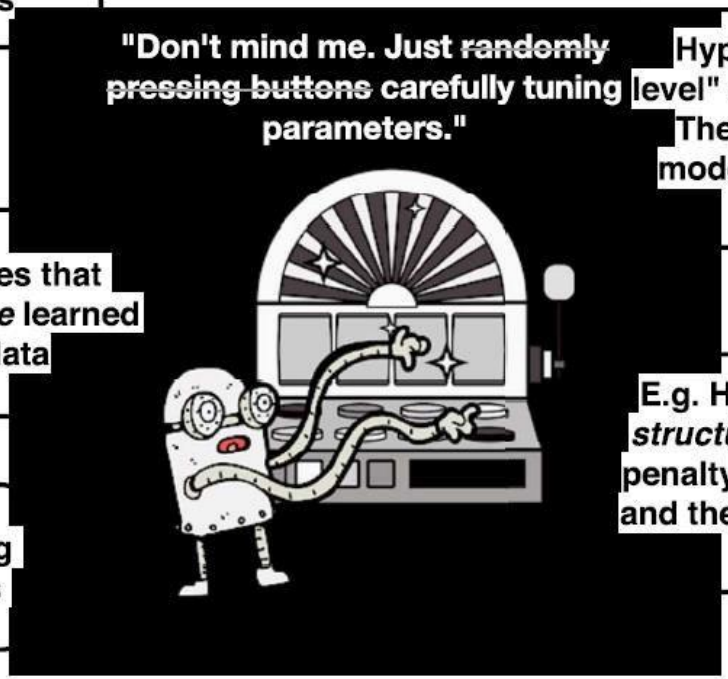


Source: https://elitedatascience.com/model-training

Big Data Tools and Algorithm Exploration
Model Training

Hyperparameters: When we talk of tuning models, we specifically mean tuning hyperparameters

Examples of Model parameters include regression coefficients, decision tree split locations



Hyperparameters express "higher-level" *structural settings* for algorithms. They are decided before fitting the model because they can't be learned from the data

Model parameters are attributes that define *individual models, and are* learned directly from the training data

E.g. Hyperparameters "higher-level" *structural settings* are strength of the penalty used in regularized regression and the number of trees to include in a random forest

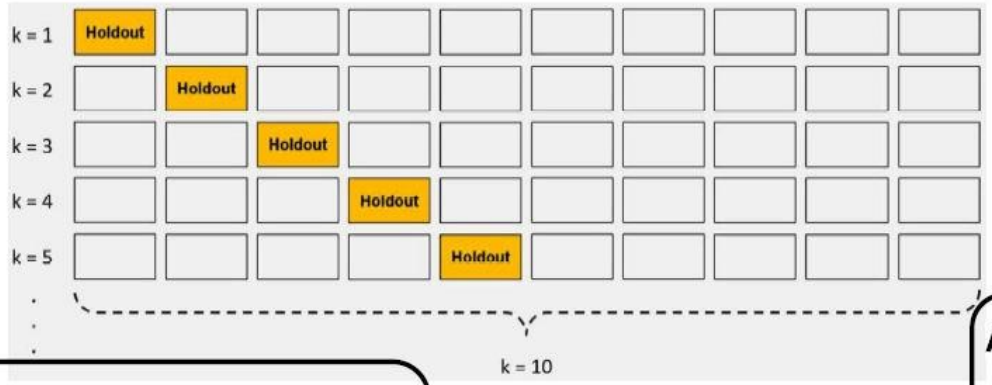
There are two types of parameters in machine learning algorithms: Model Parameters and hyperparameters

Source: https://elitedatascience.com/model-training

Big Data Tools and Algorithm Exploration
Model Training

Cross-validation is a tuning method for getting a reliable estimate of model performance using only training data

1- Split your data into 10 equal parts, or "folds", 2- Train your model on 9 folds (e.g. the first 9 folds)



3 - Evaluate it on the 1 remaining "hold-out" fold, Perform steps (2) and (3) 10 times, each time holding out a different fold

10-fold cross-validation, breaks your training data into 10 equal parts (or *folds*), essentially creating 10 miniature train/test splits

Average the performance across all 10 hold-out folds, average performance across the 10 hold-out folds is your final performance estimate, also called your *cross-validated* score

Source: https://elitedatascience.com/model-training

Big Data Tools and Algorithm Exploration
Model Training

We've split our dataset into training and test sets, and we've learned about hyperparameters and cross-validation, we're ready fit and tune our models

perform the entire cross-validation loop detailed above on each set of hyperparameter values using a high-level pseudo-code

```
For each algorithm (i.e. regularized regression, random forest, etc.):
  For each set of hyperparameter values to try:
    Perform cross-validation using the training set.
    Calculate cross-validated score.
```

At the end of this process, you will have a cross-validated score for each set of hyperparameter values... for each algorithm

Elastic-Net		
Penalty Ratio	Penalty Strength	CV-Score
75/25	0.01	0.63
75/25	0.05	0.64
75/25	0.10	**0.67**
50/50	0.01	0.62
50/50	0.05	0.63
50/50	0.10	0.66

Source: https://elitedatascience.com/model-training

Big Data Tools and Algorithm Exploration
Model Training

By now, you'll have 1 "best" model *for each algorithm* that has been tuned through cross-validation. Most importantly, you've only used the training data so far

Keep the set of hyperparameter values with best cross-validated score. Re-train the algorithm on the entire training set (without cross-validation)

For classification tasks, we recommend Area Under ROC Curve (AUROC). *Higher values are better*

Because you've saved your test set as a truly unseen dataset, you can now use it get a reliable estimate of each models' performance.There are a variety of performance metrics you could choose from

For regression tasks, we recommend Mean Squared Error (MSE) or Mean Absolute Error (MAE). *(Lower values are better)*

like the Hunger Games... each algorithm sends its own "representatives" (i.e. model trained on the best set of hyperparameter values) to the final selection

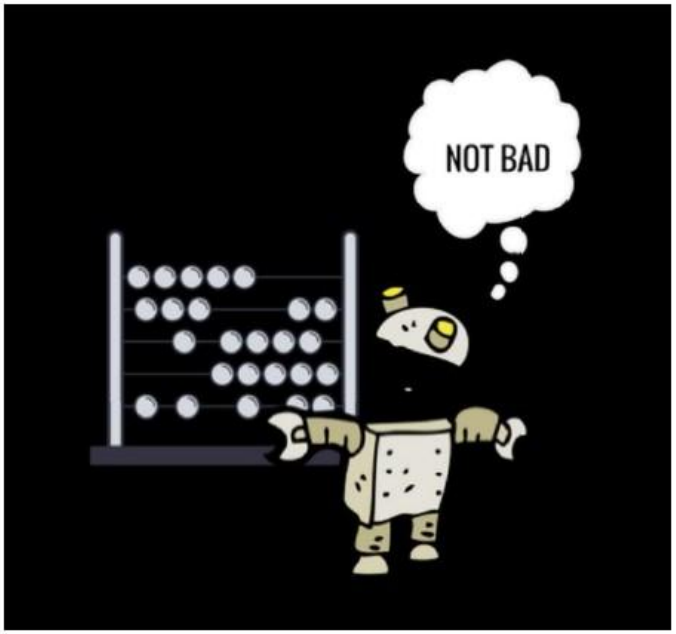
Source: https://elitedatascience.com/model-training

Big Data Tools and Algorithm Exploration
Model Training Conclusion

Pick the winning model by asking: Which model had the best performance on the test set? (performance)

Calculate performance metrics using those predictions and the "ground truth" target variable from the test set

For each of your models, make predictions on your test set



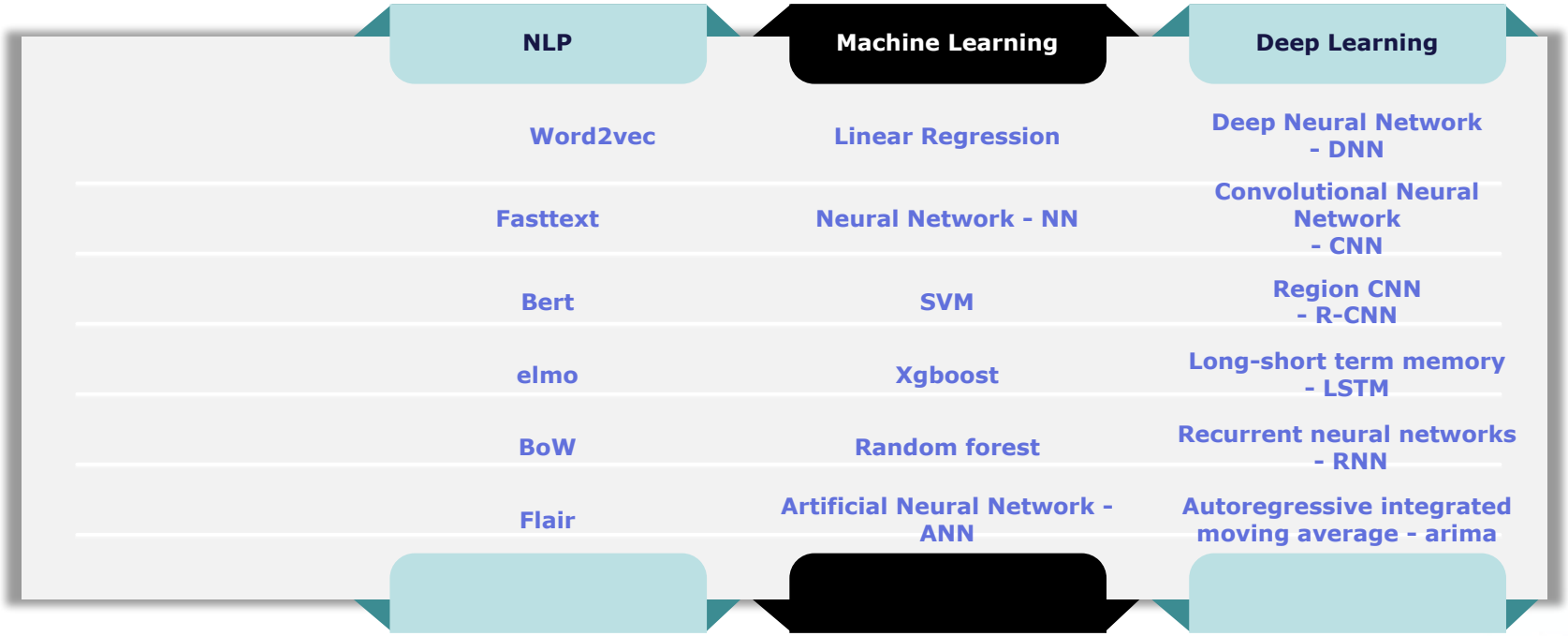
Pick the winning model by asking: Does it perform well across various performance metrics? (robustness)

Pick the winning model by asking: Did it also have (one of) the best cross-validated scores from the training set? (consistency)

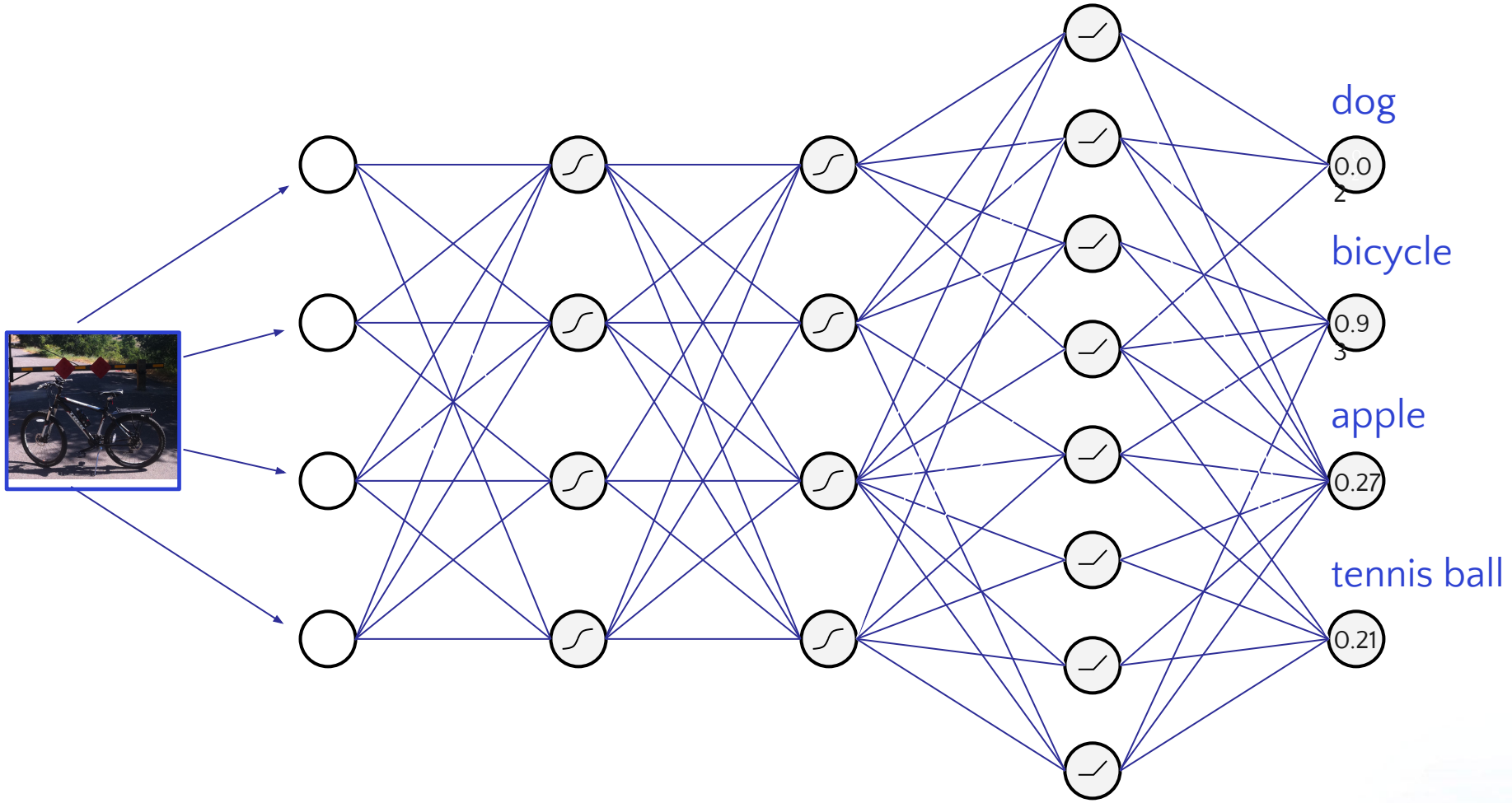
Pick the winning model by asking: Does it solve the original business problem? (win condition)

Source: https://elitedatascience.com/model-training

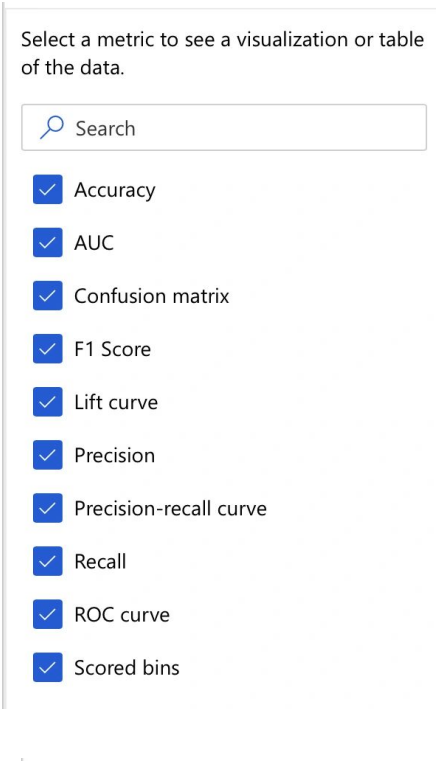
Modelling Techniques

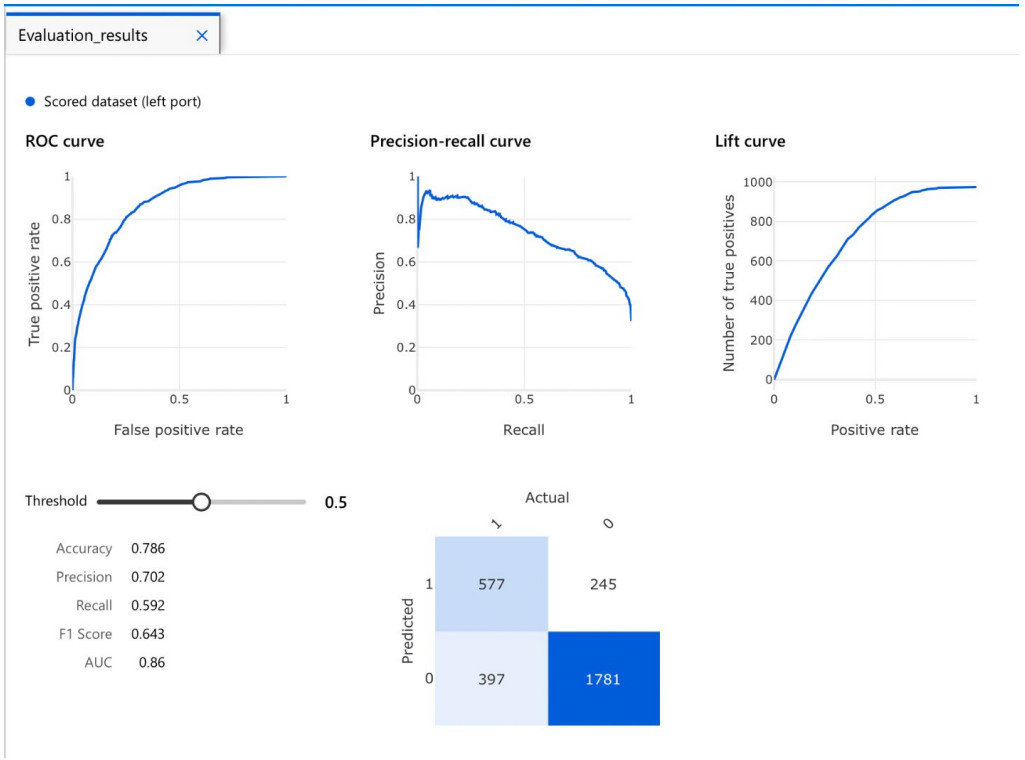


Trained Convolutional NN



Machine Learning Model Training Visulisation





Score bin ↓	Positive exam...	Negative exam...	Fraction above thres...	Accura...	F1 Score	Precisi...	Recall	Negative precis...	Negative recall	Cumulative AUC
(0.900,1.000]	95	11	0.035	0.703	0.176	0.896	0.098	0.696	0.995	0.000
(0.800,0.900]	149	21	0.092	0.746	0.390	0.884	0.251	0.732	0.984	0.002

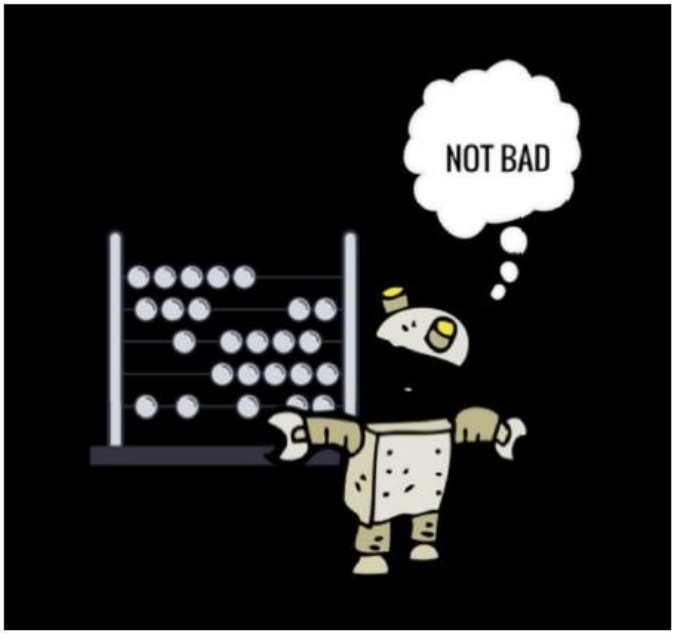
https://www.analyticsvidhya.com/blog/2021/09/a-comprehensive-guide-on-using-azure-machine-learning/

Machine Learning Model Training Conclusion

Pick the winning model by asking: Which model had the best performance on the test set? (performance)

Calculate performance metrics using those predictions and the "ground truth" target variable from the test set

For each of your models, make predictions on your test set



Pick the winning model by asking: Does it perform well across various performance metrics? (robustness)

Pick the winning model by asking: Did it also have (one of) the best cross-validated scores from the training set? (consistency)

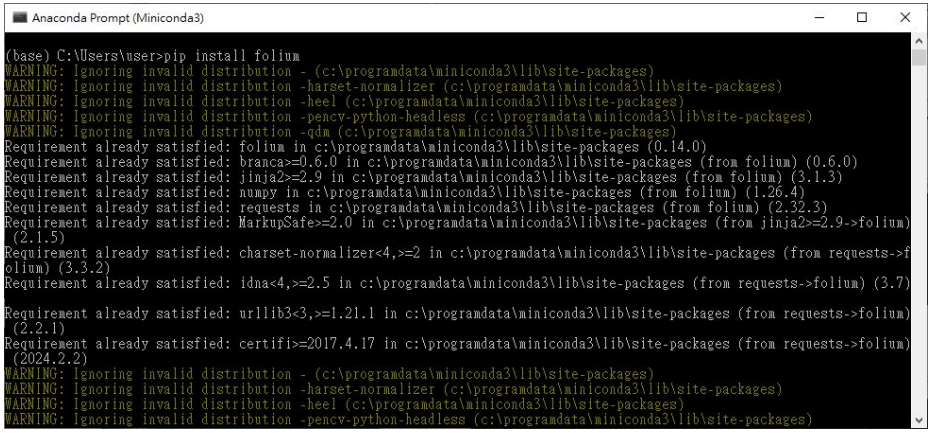
Pick the winning model by asking: Does it solve the original business problem? (win condition)

Source: https://elitedatascience.com/model-training

Obtaining Data – Web Scraping

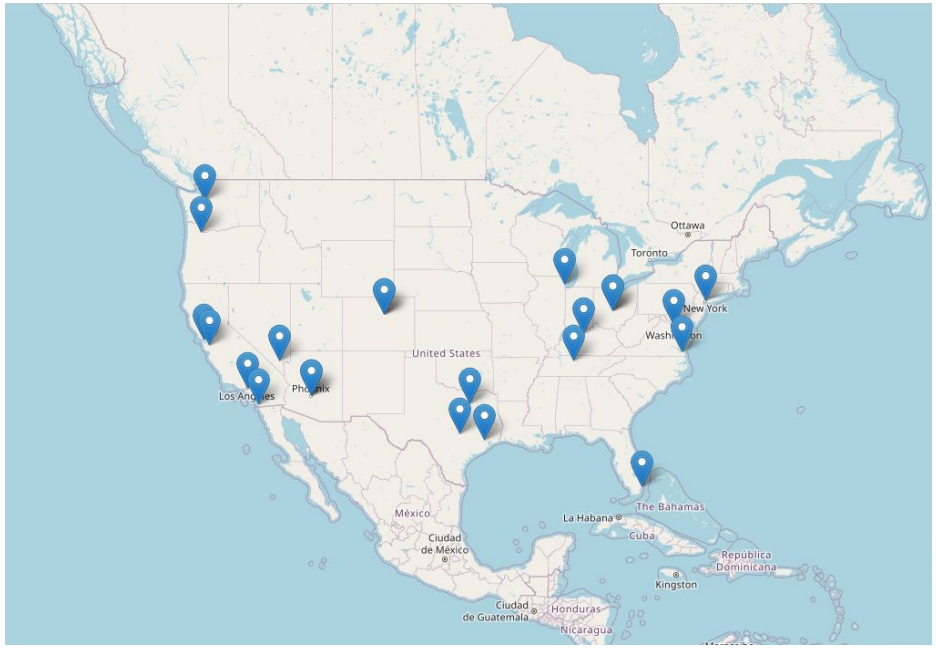
1. Open tools “Anaconda Prompt” which will direct you to the command prompt.

2. Install python package requests by entering “pip install folium”



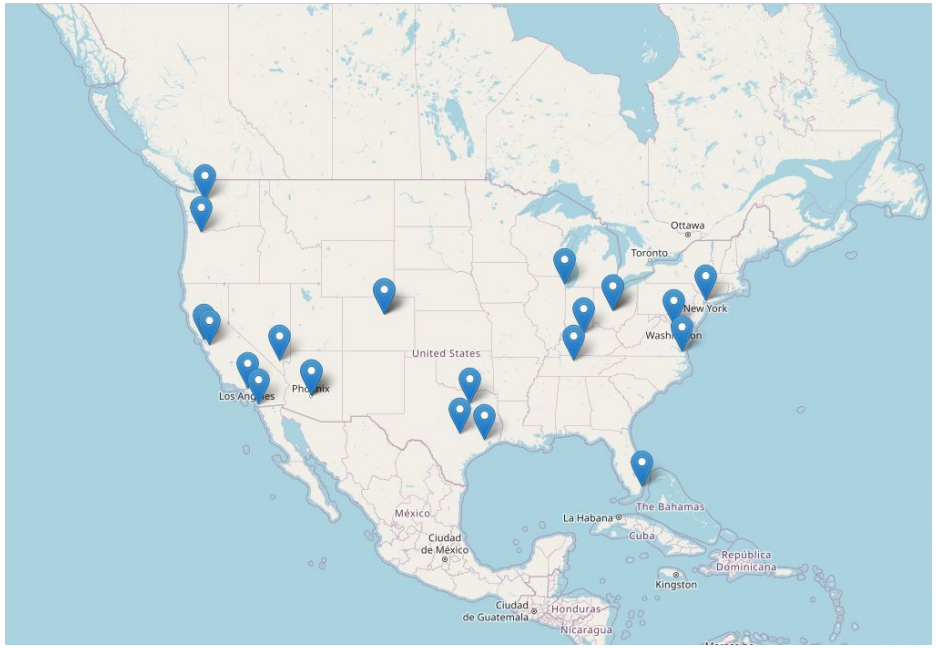
Python Google Map Tutorial

```
import folium

# List of 23 coordinates (latitude, longitude)
coordinates = [
    (37.7749, -122.4194),  # San Francisco, CA
    (34.0522, -118.2437),  # Los Angeles, CA
    (40.7128, -74.0060),   # New York, NY
    (41.8781, -87.6298),    # Chicago, IL
    (29.7604, -95.3698),    # Houston, TX
    (33.4484, -112.0740),   # Phoenix, AZ
    (39.7392, -104.9903),   # Denver, CO
    (47.6062, -122.3321),   # Seattle, WA
    (38.9072, -77.0369),    # Washington, D.C.
    (39.9612, -82.9988),    # Columbus, OH
    (30.2672, -97.7431),    # Austin, TX
    (32.7767, -96.7970),    # Dallas, TX
    (37.3382, -121.8863),   # San Jose, CA
    (39.7392, -104.9903),   # Denver, CO
    (36.1699, -115.1398),    # Las Vegas, NV
    (25.7617, -80.1918),    # Miami, FL
    (38.2527, -85.7585),    # Louisville, KY
    (39.9612, -82.9988),    # Columbus, OH
    (36.8508, -76.2859),    # Norfolk, VA
    (33.4484, -112.0740),   # Phoenix, AZ
    (45.5155, -122.6793),   # Portland, OR
    (36.1627, -86.7816),    # Nashville, TN
    (32.7157, -117.1611),   # San Diego, CA
]
```

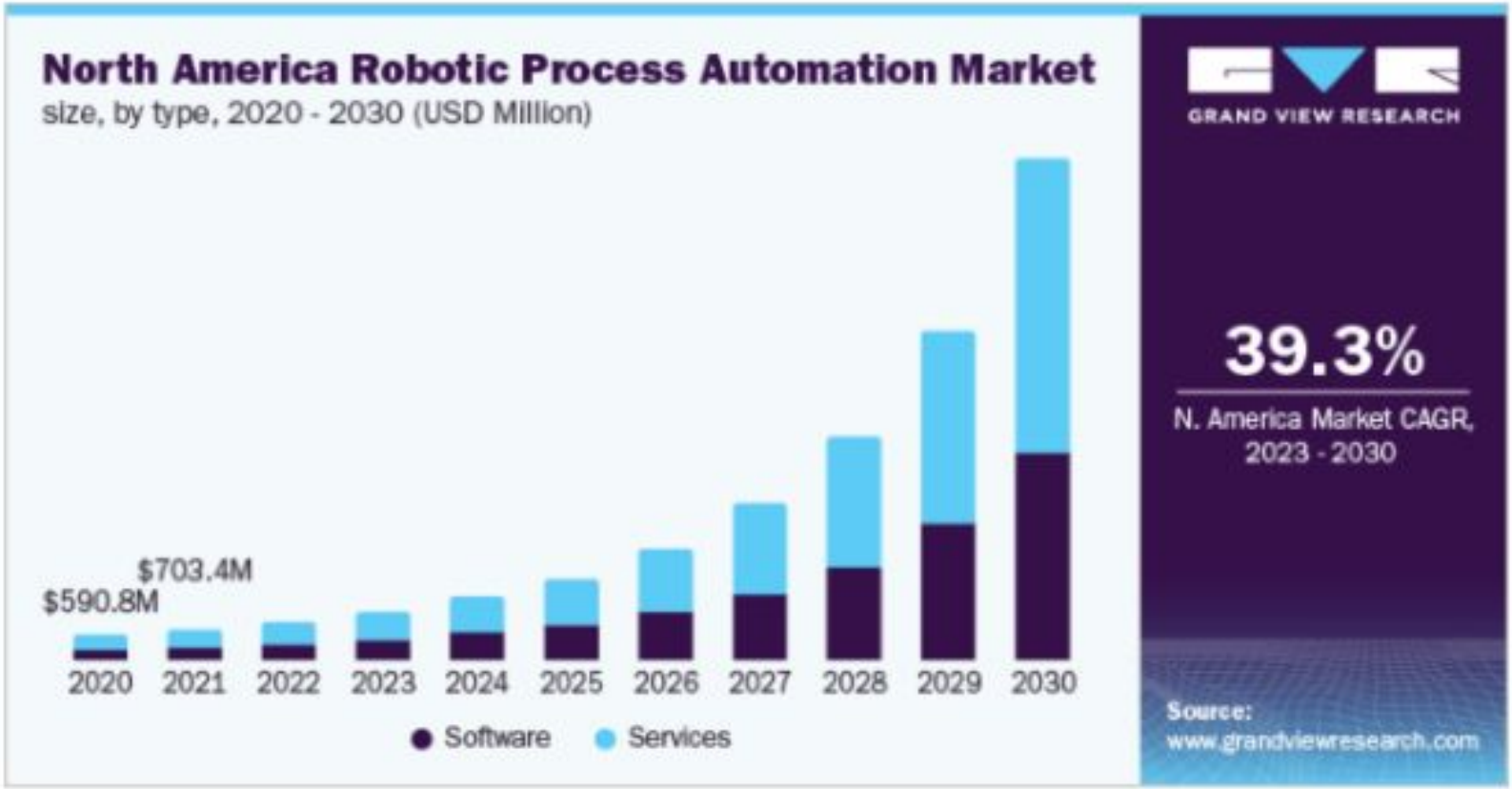


Python Google Map Tutorial

```
# Create a map centered at the first coordinate
my_map = folium.Map(location=coordinates[0],
    zoom_start=4)

# Add markers for each coordinate
for lat, lon in coordinates:
    folium.Marker(location=(lat,
    lon)).add_to(my_map)

# Save the map to an HTML file
my_map.save("my_map.html")

# If you're using Jupyter Notebook, you can
    display the map inline
my_map
```



Opportunities of BPA – The Money



1. *Robotic Process Automation Market Size & Share Report 2030*. Robotic Process Automation Market Size & Share Report 2030. https://www.grandviewresearch.com/industry-analysis/robotic-process-automation-rpa-market

Opportunities of BPA – Sectors

Industries	% of RPA Solutions
Banking, Financial Services & Insurance	51 %
Business Process Outsourcing (BPO)	14 %
Manufacturing – Consumer Packaged Goods (CPG)	7 %
Professional, Legal & Accountancy Services	7 %
Retail Trade	7 %
Technology (IT, Internet, SAAS)	7 %
Utilities	7 %

1. *Top 15 RPA Use Cases & Examples in Banking in 2023.* https://research.aimultiple.com/banking-rpa/

Opportunities of BPA – Sectors

- **Healthcare**
  - Monitoring, drug dispensing
- **Retail**
  - Auto-checkout, smart shelf, inventory management
- **Marketing**
  - Lead generation
  - social media management
- **Finance**
  - Fraud detection, Credit management, Regtech

Opportunities of BPA – Regtech

- **Follow the money**

CNN

Wells Fargo ordered to pay $3.7 billion for 'illegal activity' including unjust foreclosures and vehicle repossessions

The CFPB said the more than $2 billion in customer refunds Wells Fargo has been ordered to pay includes more than $1.3 billion to consumers hurt...

20 Dec 2022

Spiceworks

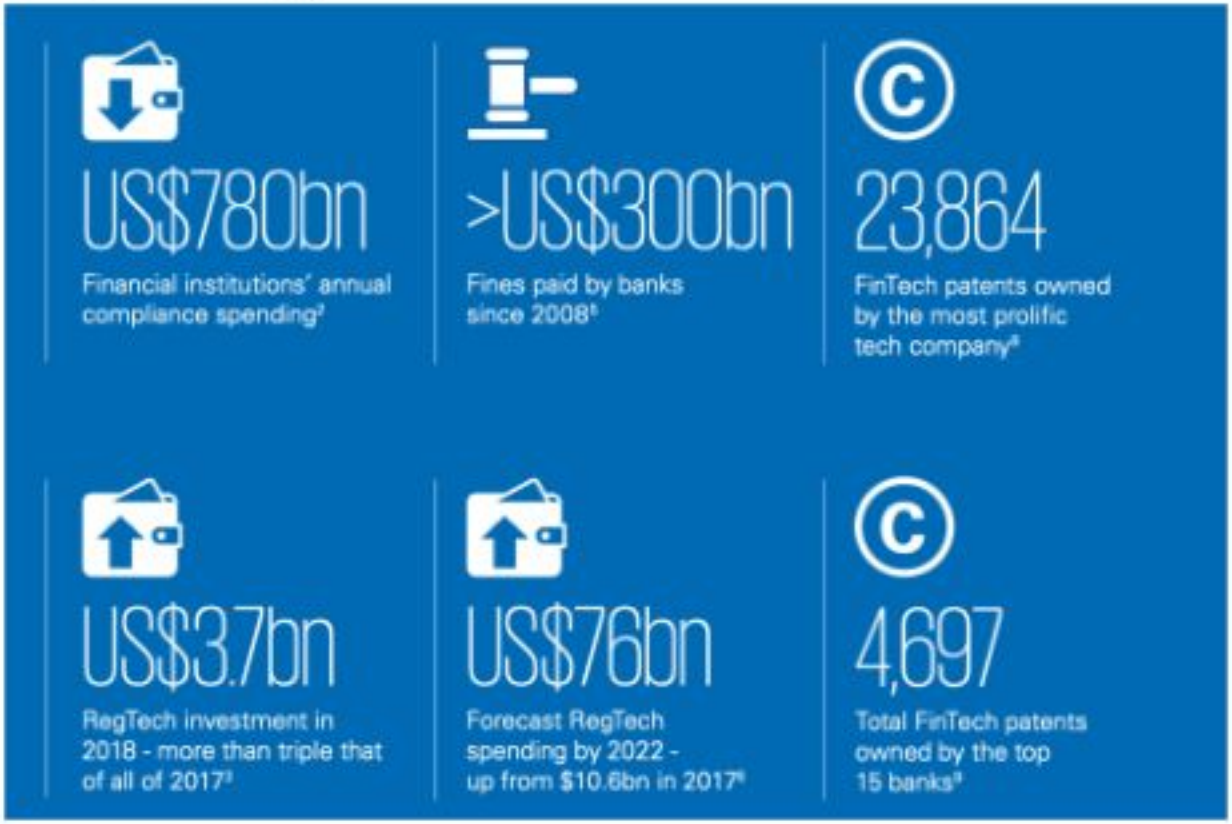
SEC Penalizes Major Wall Street Firms $1.97B For Using Unauthorized Messaging Apps

If there are allegations of wrongdoing or misconduct, we must be ... Morgan Stanley recently agreed to a $35 million fine by the SEC for...

28 Sept 2022

Opportunities of BPA – Regtech

- **Follow the money**



1. KPMG. (2019, June). *There's a revolution coming.* https://assets.kpmg.com/content/dam/kpmg/cn/pdf/en/2019/06/embracing-the-challenge-of-the-new-regtech-era.pdf

Opportunities of BPA – Regtech

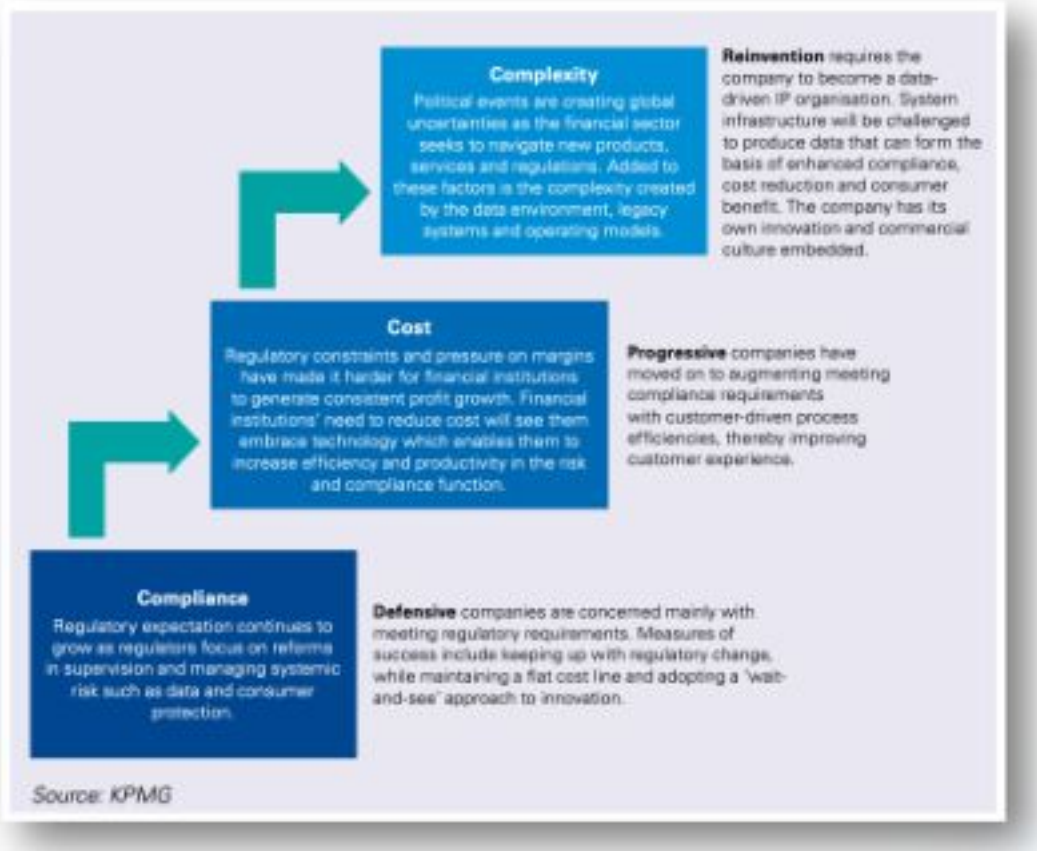
- **Common themes**

1. KPMG. (2022, November). https://assets.kpmg.com/content/dam/kpmg/uk/pdf/2022/11/innovate-finance-regtech-industry-and-adoption.pdf. https://assets.kpmg.com/content/dam/kpmg/uk/pdf/2022/11/innovate-finance-regtech-industry-and-adoption.pdf

Opportunities of BPA – Regtech

Adoption phases:

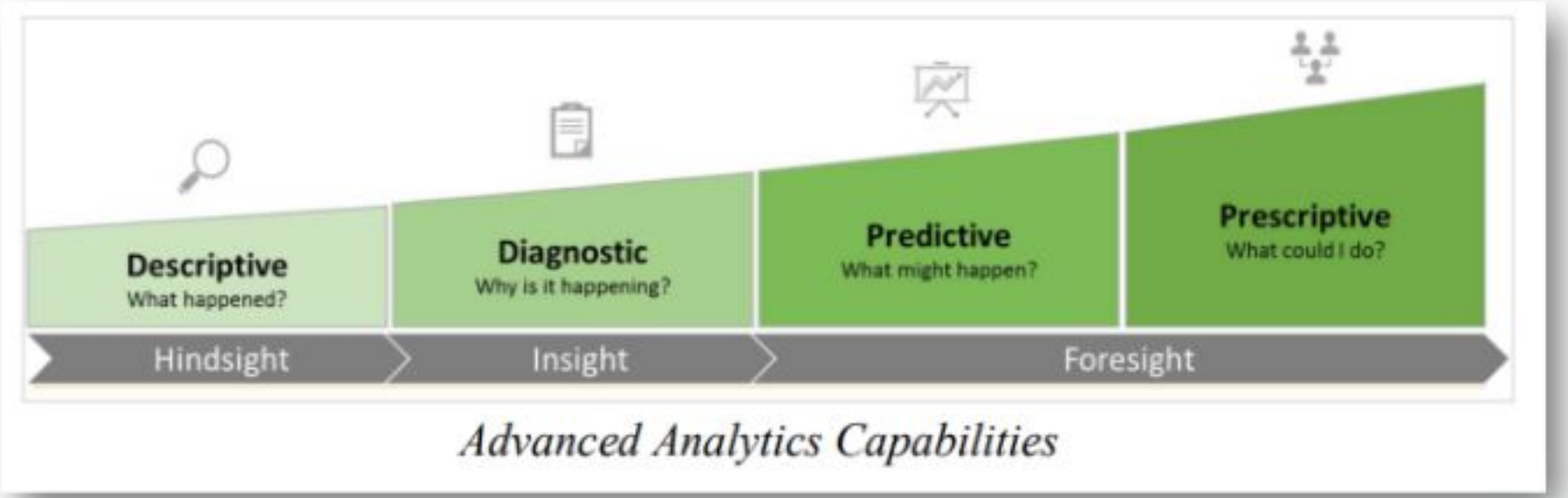
1. Defensive
2. Progressive
3. Reinvention



1. KPMG. (2019, June). *There's a revolution coming.* https://assets.kpmg.com/content/dam/kpmg/cn/pdf/en/2019/06/embracing-the-challenge-of-the-new-regtech-era.pdf

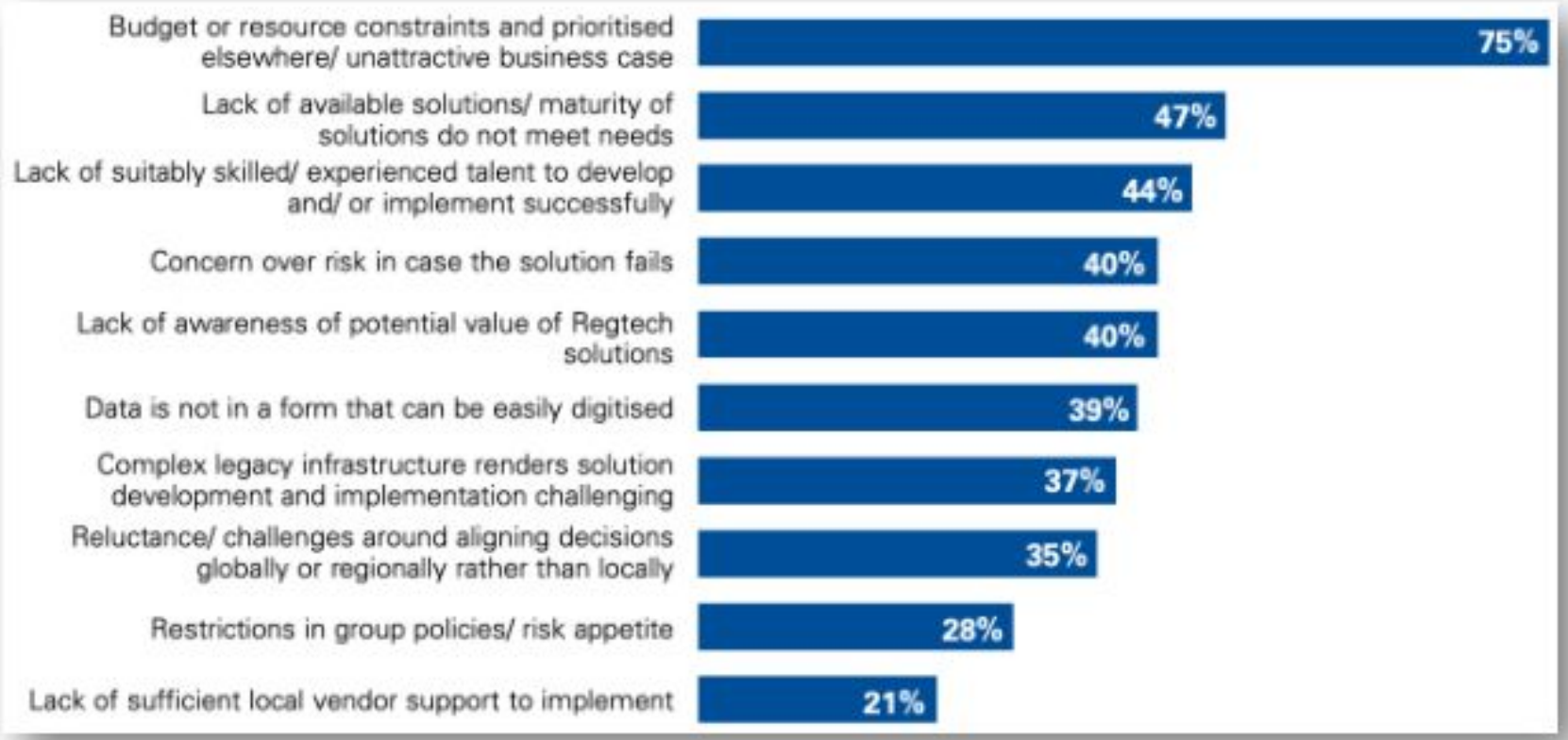
Opportunities of BPA – Regtech

- **Analytics enabled by BPA**



1. HKMA. (2021, June). *Regtech Watch Issue No. 7.* https://www.hkma.gov.hk/media/eng/doc/key-information/guidelines-and-circular/2021/20210617e1a1.pdf

Challenges of BPA – Insight from the banks

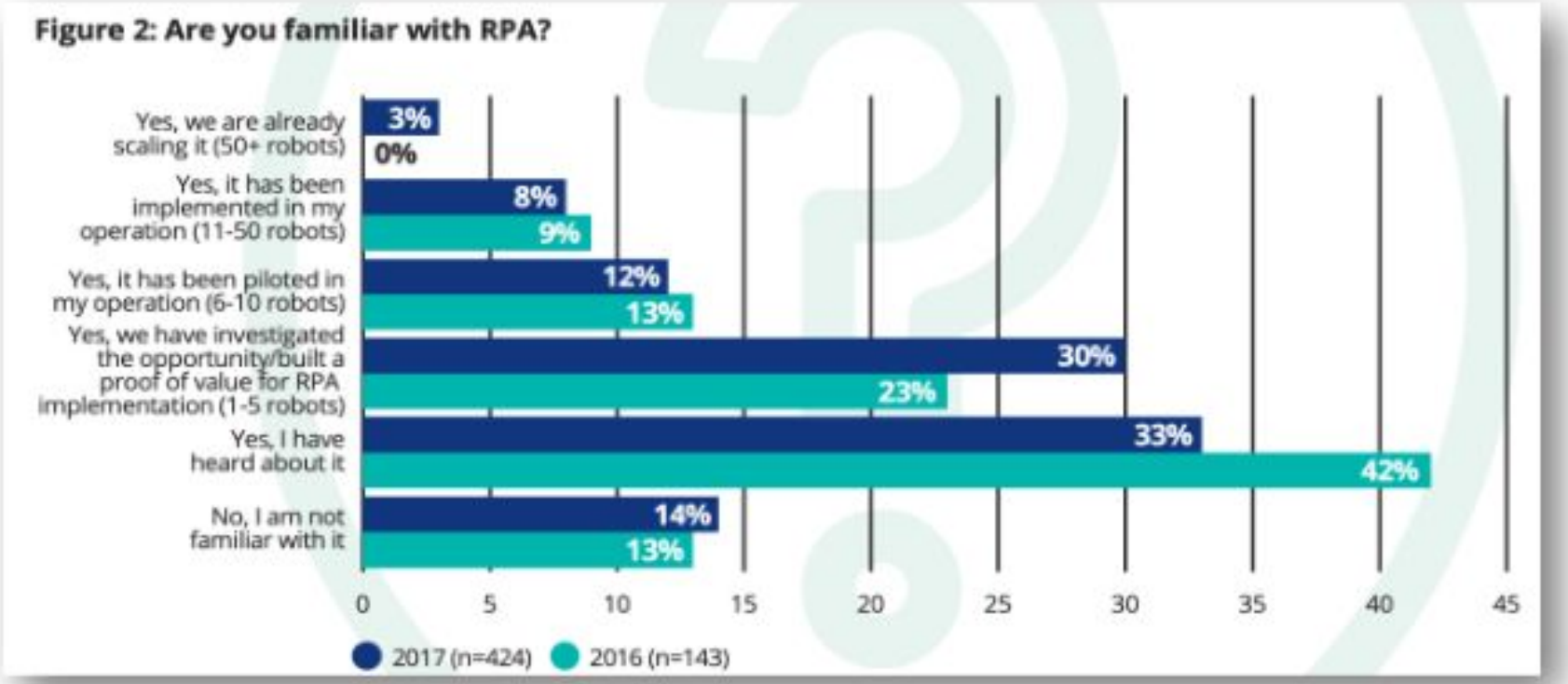


1. KPMG & HKMA. (2020). *Transforming Risk Management and Compliance: Harnessing the Power of Regtech.* https://www.hkma.gov.hk/media/chi/doc/key-information/press-release/2020/20201102c3a1.pdf

Challenges of BPA – Worries from the workers?

1. Deloitte. (2018). *The robots are ready. Are you?* https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology/deloitte-robots-are-ready.pdf

Challenges of BPA – Hesitant to scale up



1. Deloitte. (2018). *The robots are ready. Are you?* https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology/deloitte-robots-are-ready.pdf

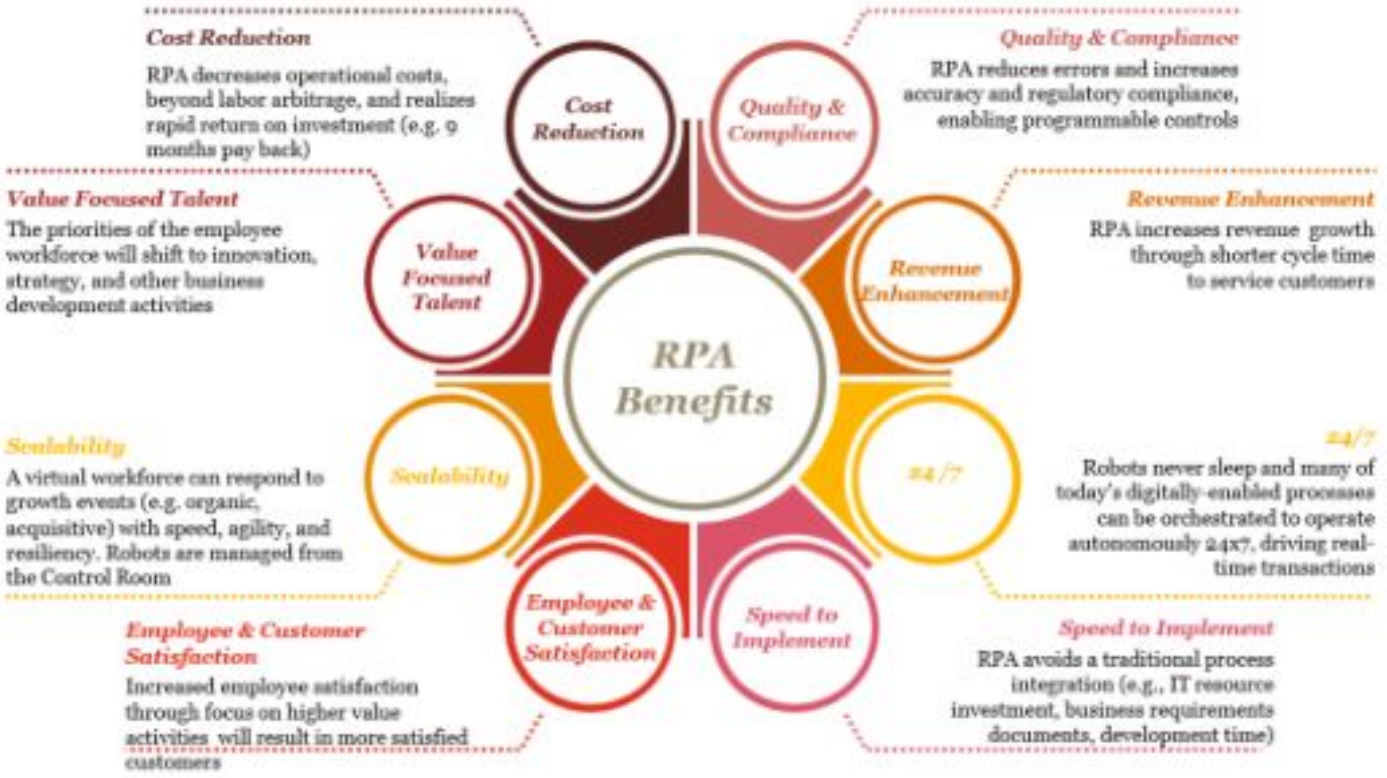
Challenges of BPA

- **Shortage**
  - Budget / Resource/ Talent
- **Concerns**
  - Risk of rushing / failing
  - Moral / Structural unemployment

* Success depends on planning
(More on project management later)

Implications of BPA – The Good

- Cost, Efficiency, Customer Experience



1. *PwC Robotics Process Automation Solutions*. PwC. https://www.pwchk.com/en/services/entrepreneurial-and-private-business/new-technology-digitalisation-and-transformation/process-automation-solutions.html

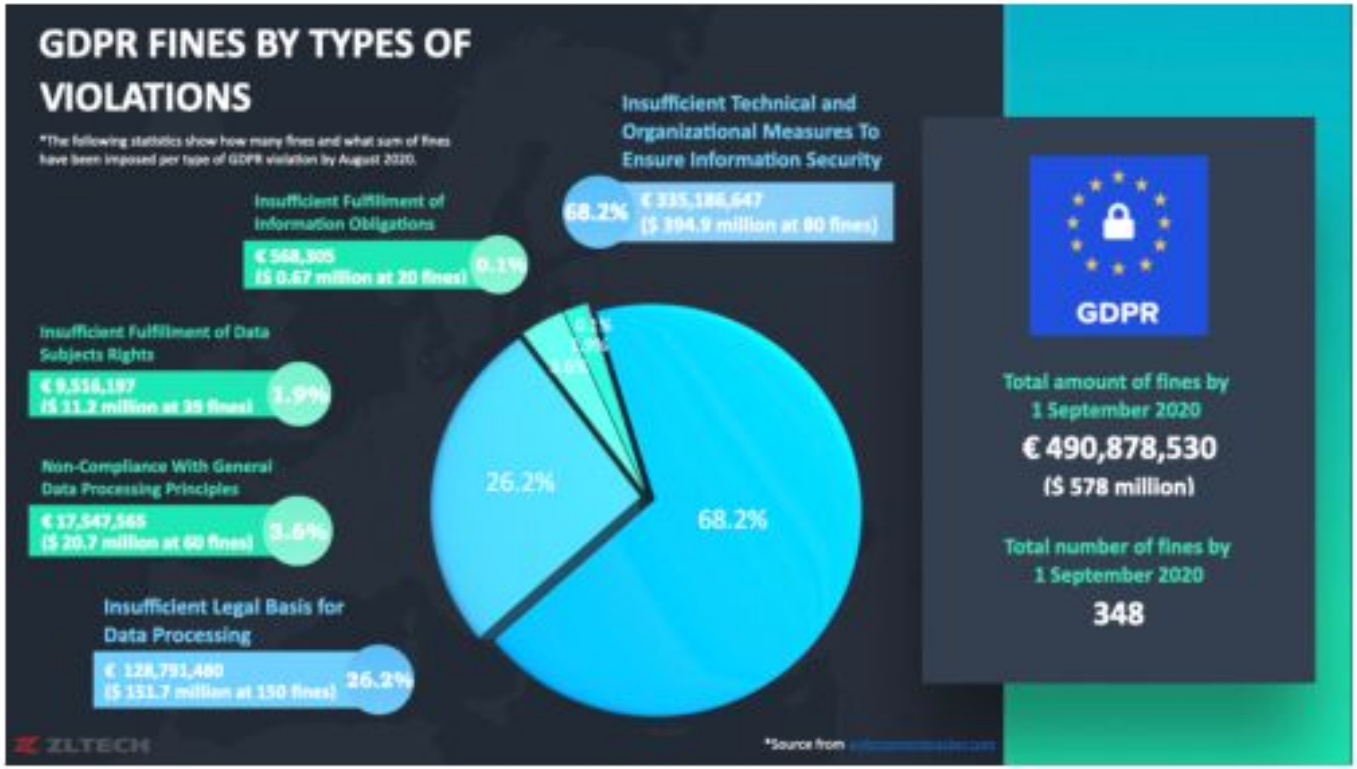
Implications of BPA – The Bad

- Automation complacency / dependency
- Risk of Technical Issues / Downtime

Image: Civil Aviation Safety Authority

Implications of BPA – The Ugly

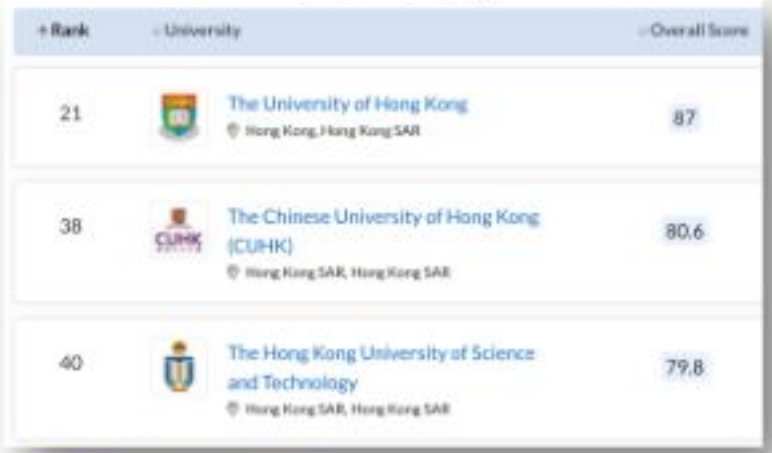
- Data Collection and Analysis [1]
- Vulnerability to Cyberattacks



1. Chen, B. (2020, September 9). *GDPR Recap: 28 Nations, 348 Fines, Half a Billion Euros*. ZL Tech. https://www.zlti.com/blog/gdpr-recap-28-nations-343-fines-half-a-billion-euros/

Sequential Data Types

- **Motivation**
  - How to represent the ordering of schools?
    - e.g., HKU > CUHK > HKUST
    - Preferably with 1 variable…



Rank	University	Overall Score
21	The University of Hong Kong Hong Kong, Hong Kong SAR	87
38	The Chinese University of Hong Kong (CUHK) Hong Kong SAR, Hong Kong SAR	80.6
40	The Hong Kong University of Science and Technology Hong Kong SAR, Hong Kong SAR	79.8

* QS Rankings 2023

```
school_1 = 'hku'
school_2 = 'cuhk'
school_3 = 'hkust'
print(school_1)
print(school_2)
print(school_3)
```

```
hku
cuhk
hkust
```

Sequential Data Types

- **List**
  - "Mutable" object (something you can change)
  - Representation

```
school_list = ['hku', 'cuhk', 'hkust']
print(school_list)
print(type(school_list))
```

```
['hku', 'cuhk', 'hkust']
<class 'list'>
```

  - Indexing

```
print(school_list[0])      # First 1 item
print(school_list[0:2])    # First 2 items
print(school_list[-1])     # Last one
```

```
hku
['hku', 'cuhk']
polyu
```

Sequential Data Types

- **List**
  - Adding items

```
school_list = ['hku', 'cuhk', 'hkust'] + ['cityu', 'polyu']
school_list
```

```
['hku', 'cuhk', 'hkust', 'cityu', 'polyu']
```

```
school_list = ['hku', 'cuhk', 'hkust', 'cityu']
school_list.append('polyu')
school_list
```

```
['hku', 'cuhk', 'hkust', 'cityu', 'polyu']
```

Sequential Data Types

- **List**
  - Removing an item
    - By index ➔ pop()

```
school_list.pop(1)
school_list
```

```
['hku', 'hkust', 'cityu', 'polyu']
```

    - By element ➔ remove()

```
school_list.remove('cityu')
school_list
```

```
['hku', 'hkust', 'polyu']
```

Sequential Data Types

- **Common operations**
  - Membership Check ➔ "in"

```
# Membership check
school_list = ['hku', 'cuhk', 'hkust', 'cityu', 'polyu']
print('hku' in school_list)
print('hkbu' in school_list)
```

```
True
False
```

Sequential Data Types

- **Common operations**
  - Ordering ➔ max() / min() / reversed() / sorted()

```
number_list = [1, 5, 6, 2, 3]
print(max(number_list))                 # Max
print(min(number_list))                 # Min
print(list(reversed(number_list)))      # Reverse order
print(sorted(number_list))              # Sort by ascending order
```

```
6
1
[3, 2, 6, 5, 1]
[1, 2, 3, 5, 6]
```

Sequential Data Types

- **String (again)**
  - string = sequential!


```
# Index:        0123456789
school_name = 'HKU School of Professional and Continuing Education'

print(len(school_name))          # There is a total of 51 letters
print(school_name.index('c'))    # First c is the 6th letter (counting 1 space)
print(school_name.count('o'))    # There are 7 O's
print('HKU' in school_name)      # Substring matching
print(school_name[4:10])         # Substring extraction
```

```
51
5
7
True
School
```

Sequential Data Types

- **String (again)**
  - Conversions

```
# Conversion: String -> list
print(list('hello'))
```

```
['h', 'e', 'l', 'l', 'o']
```

```
# Conversion: List -> String
print(''.join(['h', 'e', 'l', 'l', 'o']))
print(','.join(['Welcome', 'Jackie!']))
```

```
hello
Welcome,Jackie!
```

```
# Conversion: String <-> int?
print(ord('a'))
print(chr(98))
```

```
97
b
```

Data Type - Set

- **Motivation**
  - Problem ➔ Duplicate data are everywhere

https://www.youtube.com/watch?v=jT-9_VKpAek

Data Type - Set

- **Set**
  - A way to maintain unique values

```
duplicate_list = ['Shatin City One', 'Shatin City One', 'TKO Gateway']
set(duplicate_list)
```

```
{'Shatin City One', 'TKO Gateway'}
```

Data Type - Set

- **Set**
  - Representation

```
school_set = {'A', 'B', 'B', 'C'}
print(school_set)
print(type(school_set))
```

```
{'B', 'C', 'A'}
<class 'set'>
```

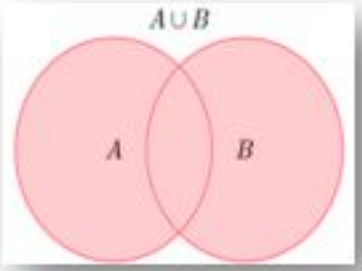
  - Can be converted between sequential data types

```
print(set(['A', 'B', 'B', 'C']))  # List -> Set
print(list({'A', 'B', 'C'}))      # Set -> List
```

```
{'C', 'B', 'A'}
['B', 'C', 'A']
```

Data Type - Set

- **Set operations**
  - Union

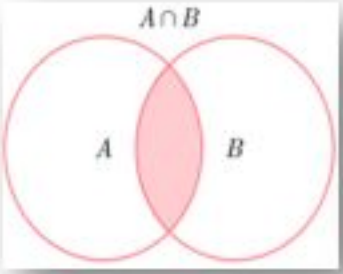



```
# Set operations
school_set_2022 = {'A', 'B', 'C'}
school_set_2023 = {     'B', 'C', 'D'}
```

```
# Union: items in any one of the sets
school_set_2022 | school_set_2023
```

```
{'A', 'B', 'C', 'D'}
```

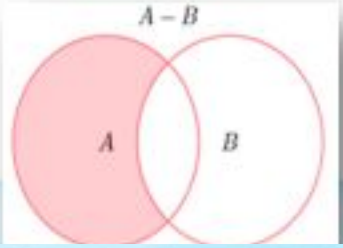
  - Intersection




```
# Intersection: items in both of the sets
school_set_2022 & school_set_2023
```

```
{'B', 'C'}
```

  - Difference




```
# Difference: Items in 1 set but not in the other
school_set_2023 - school_set_2022
```

```
{'D'}
```

Data Type - Dictionary

- **Motivation**
  - Business problems often involve key-value pairs
  - e.g., Employee ID ➔ Full Name / Job Title

	A	B	C	D
1	Employee ID	Full Name	Job Title	Department
2	E02002	Kai Le	Controls Engineer	Engineering
3	E02003	Robert Patel	Analyst	Sales
4	E02004	Cameron Lo	Network Administrator	IT
5	E02005	Harper Castillo	IT Systems Architect	IT
6	E02006	Harper Dominguez	Director	Engineering
7	E02007	Ezra Vu	Network Administrator	IT
8	E02008	Jade Hu	Sr. Analyst	Accounting
9	E02009	Miles Chang	Analyst II	Finance
10	E02010	Gianna Holmes	System Administrator	IT
11	E02011	Jameson Thomas	Manager	Finance
12	E02012	Jameson Pena	Systems Analyst	IT
13	E02013	Bella Wu	Sr. Analyst	Finance

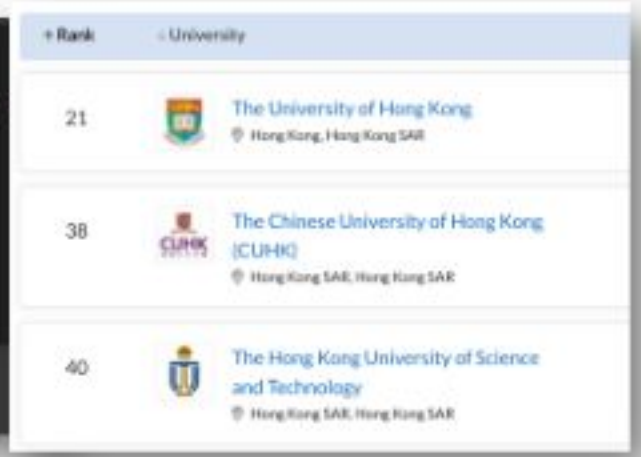
  - Looking up a keyword
    ➔ like finding words in **dictionaries**

Data Type - Dictionary

- Representation

```
# Representation: option 1
school_rank_dict = {'hku': 21, 'cuhk': 38, 'hkust': 40}
print(type(school_rank_dict))

# Representation: option 2
school_rank_dict = dict(hku=21, cuhk=38, hkust=40)
print(school_rank_dict)
```

```
<class 'dict'>
{'hku': 21, 'cuhk': 38, 'hkust': 40}
```



- Lookup ➔ get()

```
# Lookup
print(school_rank_dict['cuhk'])              # Direct lookup
print(school_rank_dict.get('hkust'))         # get(): Success
print(school_rank_dict.get('test'))          # get(): Failed
print(school_rank_dict.get('test', 'N/A'))   # get(): Error handling
```

```
38
40
None
N/A
```

Data Type - Dictionary

- Adding an item

```
# Adding an item
school_rank_dict = {'hku': 21, 'cuhk': 38, 'hkust': 40}
school_rank_dict['cityu'] = 54
print(school_rank_dict)
```

```
{'hku': 21, 'cuhk': 38, 'hkust': 40, 'cityu': 54}
```

- Removing an item

```
# Removing an item
school_rank_dict = {'hku': 21, 'cuhk': 38, 'hkust': 40, 'cityu': 54}
school_rank_dict.pop('cityu')
print(school_rank_dict)
```

```
{'hku': 21, 'cuhk': 38, 'hkust': 40}
```

Data Type - Dictionary

- Listing out the details

```
# Show all the keys
print(f'keys: {list(school_rank_dict.keys())}')

# Show all the values
print(f'values: {list(school_rank_dict.values())}')

# Show all the items
print(f'items: {list(school_rank_dict.items())}')
```

```
keys: ['hku', 'cuhk', 'hkust']
values: [21, 38, 40]
items: [('hku', 21), ('cuhk', 38), ('hkust', 40)]
```

- Membership Check

```
# Membership Check
print('hku' in school_rank_dict)
```

```
True
```

Data Type - Dictionary

- **Conversion**
  - Cast from a list of tuples

	A	B	C
1	Employee ID	Full Name	Job Title
2	E02002	Kai Le	Controls Engineer
3	E02003	Robert Patel	Analyst
4	E02004	Cameron Lo	Network Administrator

```
# Conversion: list => dict
employee_name_list = [('E02002', 'Kai Le'), ('E02003', 'Robert Patel')]
print(dict(employee_name_list))
```

```
{'E02002': 'Kai Le', 'E02003': 'Robert Patel'}
```

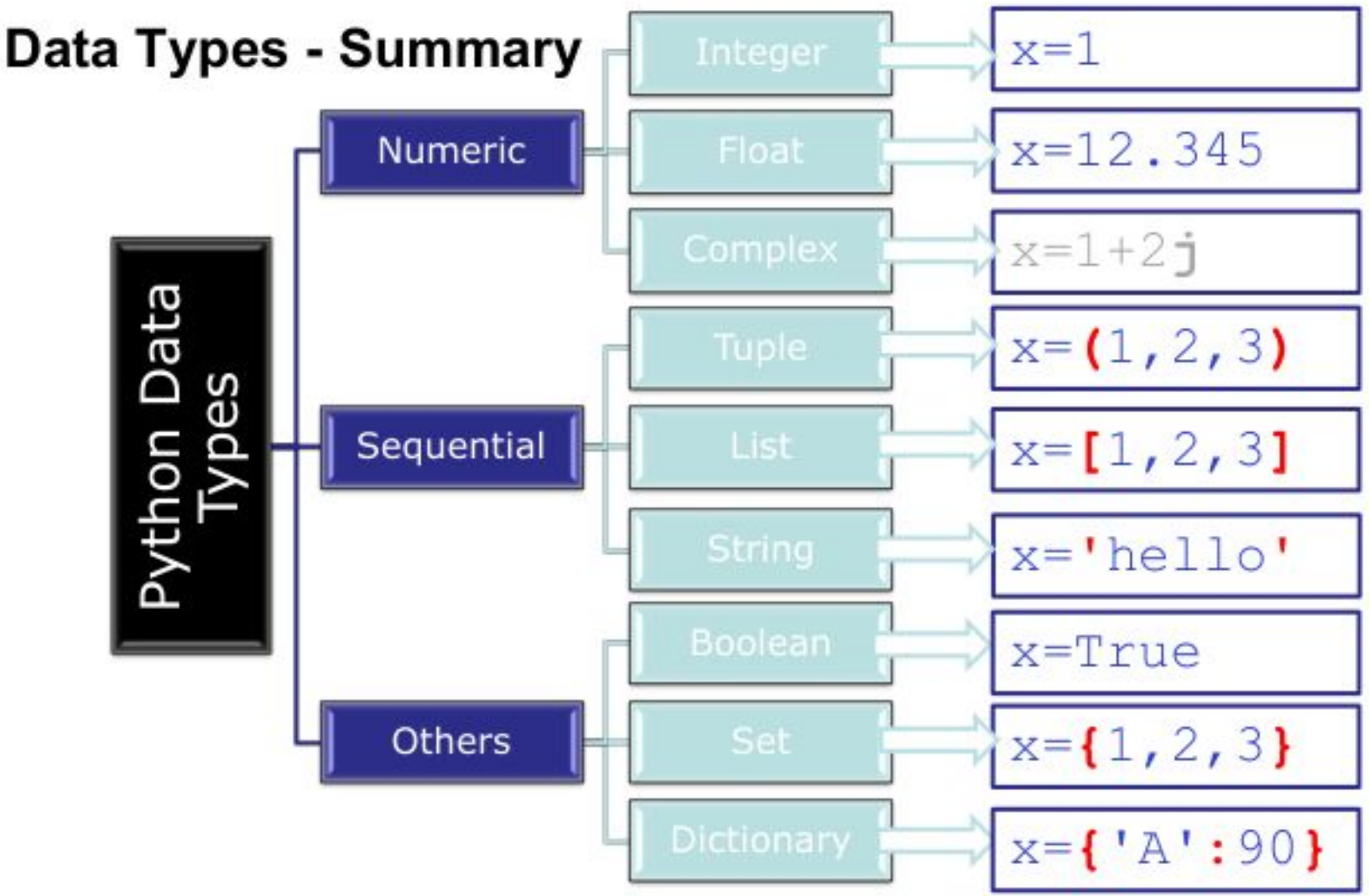
Data Type - Dictionary

- **Conversion**
  - zip() can be used to create tuples

	A	B	C
1	Employee ID	Full Name	Job Title
2	E02002	Kai Le	Controls Engineer
3	E02003	Robert Patel	Analyst
4	E02004	Cameron Lo	Network Administrator

```
employee_ids = ['E02002', 'E02003']
full_names = ['Kai Le', 'Robert Patel']
job_titles = ['Controls Engineer', 'Analyst']

print( list(zip(employee_ids, full_names)) )
print( dict(zip(employee_ids, full_names)) )
print( dict(zip(employee_ids, job_titles)) )
```

```
[('E02002', 'Kai Le'), ('E02003', 'Robert Patel')]
{'E02002': 'Kai Le', 'E02003': 'Robert Patel'}
{'E02002': 'Controls Engineer', 'E02003': 'Analyst'}
```


Data Types - Summary
Python Data Types
Numeric
Sequential
Others
Integer
Float
Complex
Tuple
List
String
Boolean
Set
Dictionary
x=1
x=12.345
x=1+2j
x=(1,2,3)
x=[1,2,3]
x='hello'
x=True
x={1,2,3}
x={'A':90}

Loops – Using "for"

- Be careful…

```
item_prices = [200, 100]
total = 0
# Case 1: Apply $50 discount once
for price in item_prices:
    total = total + price
if total >= 200:
    total = total - 50
print(total)
```

```
250
```

```
item_prices = [200, 100]
total = 0
# Case 2: Apply $50 discount twice (accidentally..)
for price in item_prices:
    total = total + price
    if total >= 200:
        total = total - 50
print(total)
```

```
200
```

Loops – Using "for"

- Looping through a list

```
school_list = ['hku', 'cuhk', 'hkust']

for school in school_list:
    print(f'{school} is a good school in Hong Kong')
```

```
hku is a good school in Hong Kong
cuhk is a good school in Hong Kong
hkust is a good school in Hong Kong
```

Loops – Using "for"

- Looping through a dictionary

```
school_rank_dict = {'hku': 21, 'cuhk': 38, 'hkust': 40}

for school, rank in school_rank_dict.items():
    print(f'{school} is ranked {rank} in the world!')
```

```
hku is ranked 21 in the world!
cuhk is ranked 38 in the world!
hkust is ranked 40 in the world!
```

Loops – Using "for"

- Looping through a string
- Q: Why use "elif" instead of "else"?

```
# looping through a string
uppercase_letter_count = 0
lowercase_letter_count = 0

for letter in 'See you in Hong Kong':
  if letter.isupper():
    uppercase_letter_count = uppercase_letter_count + 1
  elif letter.islower():
    lowercase_letter_count = lowercase_letter_count + 1

print(f'Uppercase letters: {uppercase_letter_count}')
print(f'Lowercase letters: {lowercase_letter_count}')
```

```
Uppercase letters: 3
Lowercase letters: 13
```

Loops – Using "for"

- List comprehension
  ➔ compressing a loop in 1 line

```
print(f'8 % 2 = {8%2}')
print(f'9 % 2 = {9%2}')
```

```
8 % 2 = 0
9 % 2 = 1
```

```
# Traditional way
even_number_list = list()
for i in range(11):
    # Use the modulus operator to check for even number (Can be divded by 2)
    if i % 2 == 0:
        even_number_list.append(i)
print(even_number_list)
```

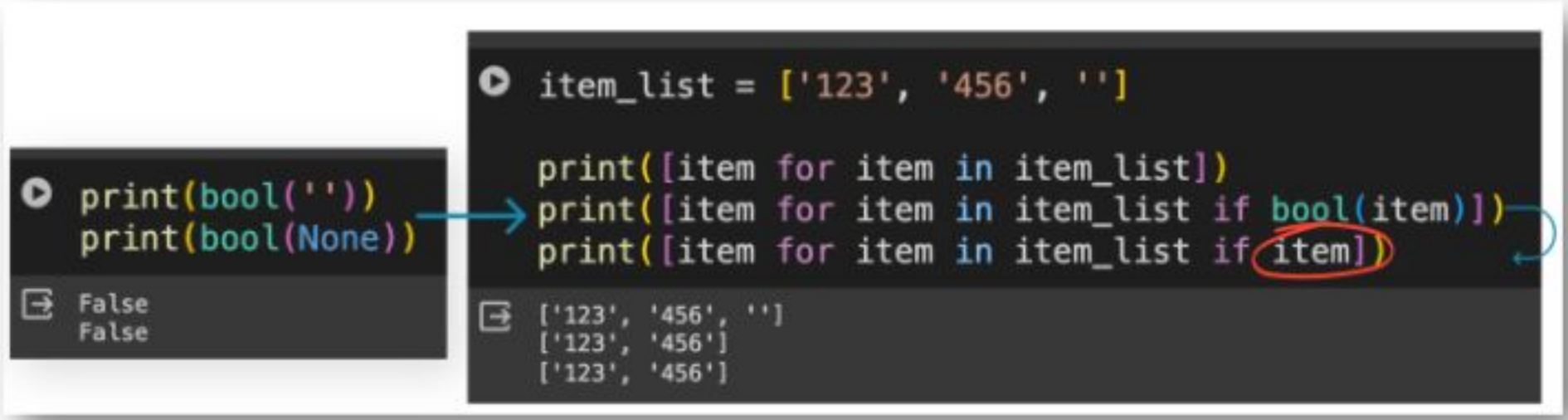
```
[0, 2, 4, 6, 8, 10]
```

```
# List comprehension
even_number_list = [i for i in range(11) if i % 2 == 0]
print(even_number_list)
```

```
[0, 2, 4, 6, 8, 10]
```

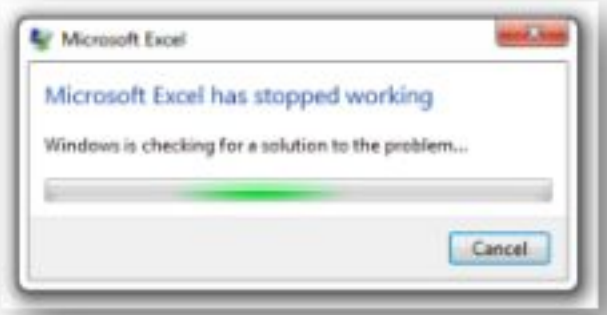
Loops – Using "for"

- List comprehension + Filtering



File Input / Output

- **Motivation**
  - Storage ➔ Saving progress
  - Interaction with other systems



File Input / Output

- **Creating a file object**
  - open(file_name, file_mode)
  - file_mode
    - r → read (read an existing file)
    - w → write (create a new file)
    - a → append (add to an existing file)

    - b → binary mode
      - e.g., Chinese characters / PDF files
      - adds to another mode (e.g., 'rb' instead 'r')

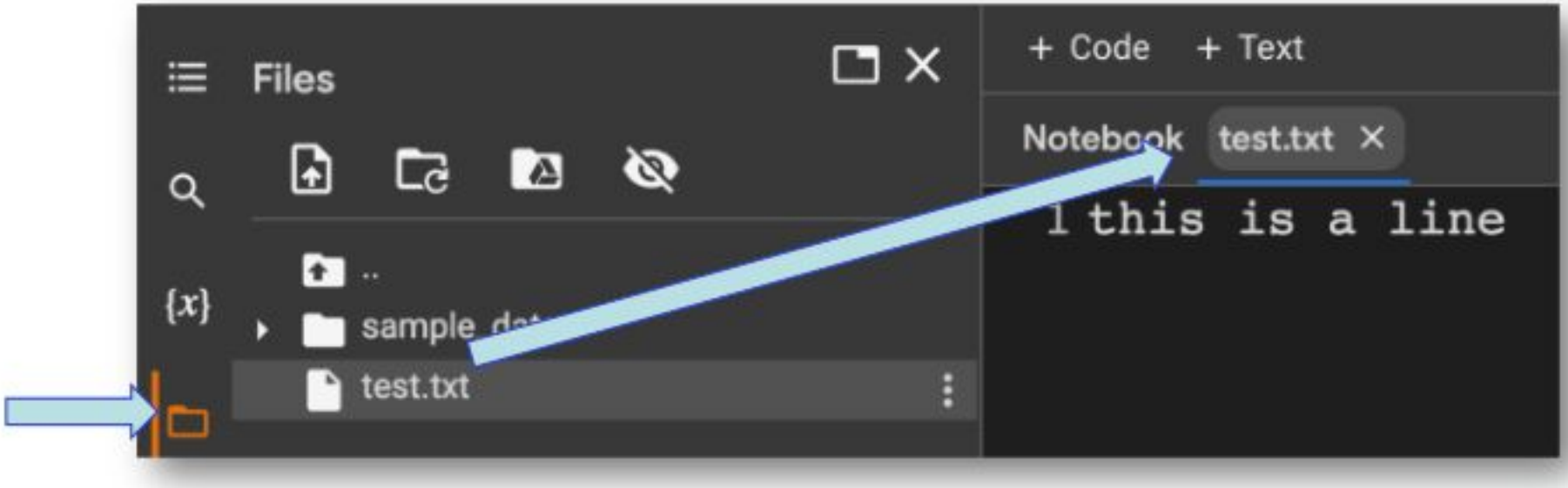
File Input / Output

- **Writing a text file**
  - open() ➔ with 'w' mode
  - write() ➔ insert a string
  - close() ➔ save the file (as a last step)

```
# Create a handler
output_file_stream = open('test.txt', 'w')

# Write a string into it
output_file_stream.write('this is a line')

# Save the file
output_file_stream.close()
```

File Input / Output

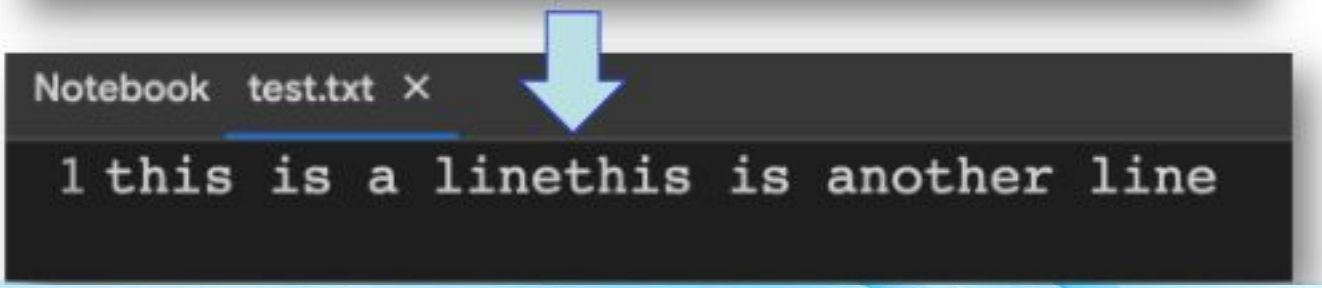
- **Writing a text file**
  - Viewing the result



File Input / Output

- **Writing a text file**
  - Multiple lines – the problem

```
# Create a handler
output_file_stream = open('test.txt', 'w')

# Write a string
output_file_stream.write('this is a line')
# Write another one
output_file_stream.write('this is another line')

# Save the file
output_file_stream.close()
```



File Input / Output

- **Writing a text file**
  - Multiple lines – the solution
  - ➔ The next-line character "\n"

Notebook test.txt ×

```
1 this is a line
2 this is another line
```

```
# Create a handler
output_file_stream = open('test.txt', 'w')

# Write a string
output_file_stream.write('this is a line\n')
# Write another one
output_file_stream.write('this is another line')

# Save the file
output_file_stream.close()
```

File Input / Output

- **The "With" statement**
  - Better readability (in terms of scoping)
  - Automatically calling close() at the end

```
# Create a handler
with open('test.txt', 'w') as output_file_stream:
    # Write a string
    output_file_stream.write('this is a line')
```

File Input / Output

- **Reading a text file**
  - open() with ‘r’ mode
  - read() ➔ extract all the content from the file

```
file_text = None
# Create a handler
with open('test.txt', 'r') as input_file_stream:
    # Read the file
    file_text = input_file_stream.read()
file_text
```

```
'this is a line\nthis is another line'
```

File Input / Output

- **Reading a text file**
  - Handling separators
    - splitlines() ➔ useful against line separators '\n'
    - split() ➔ flexible, can handle commas in CSV files

```
print( file_text.split('\n') )
print( file_text.splitlines() )
```

```
['this is a line', 'this is another line']
['this is a line', 'this is another line']
```

Data Visualizations

1. describe the contemporary trends of process automation and explain the opportunities and challenges of business process automation;
2. outline key steps in process automation project management and illustrate the significance of each step;
3. apply computational tools to implement process automation;
4. discuss the development of process automation and practical cases for business.

Exercise 1

Starting notebook:

https://github.com/innoviai/ipa_courses/blob/main/Lecture%203/python_data_visulisation_start.ipynb

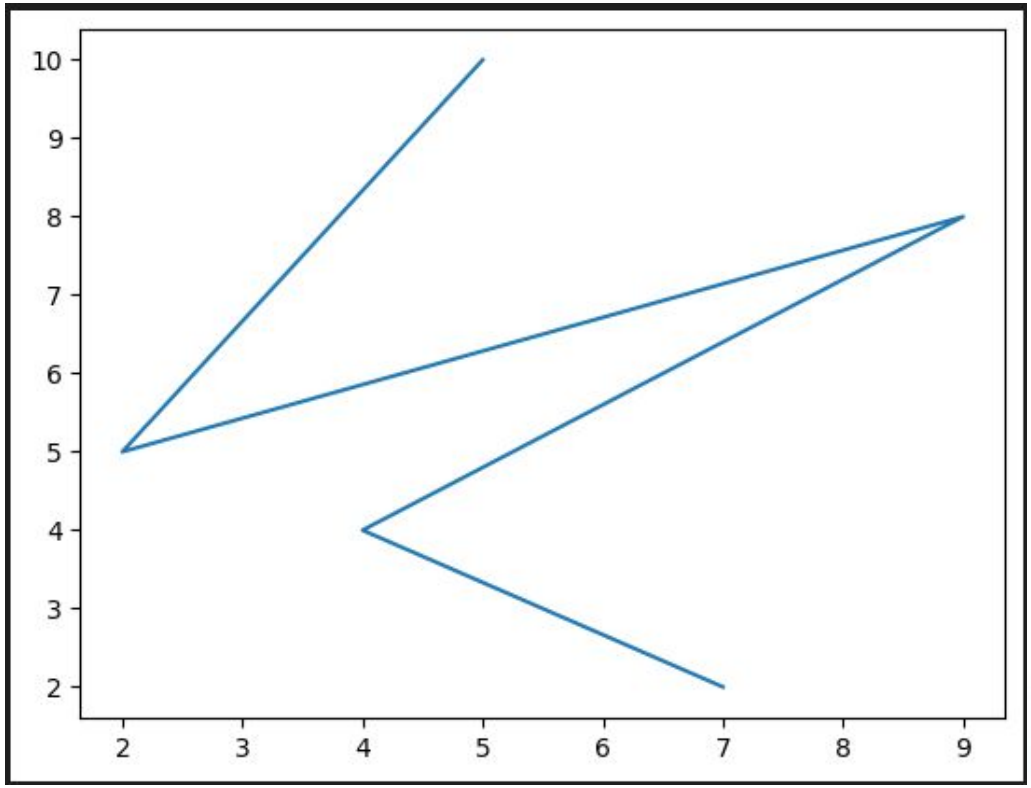
Step

1. *Open "Anaconda Prompt" and type "python"*
2. *Import library* "matplotlib" by entering code "import matplotlib"

Exercise 1

Step 2

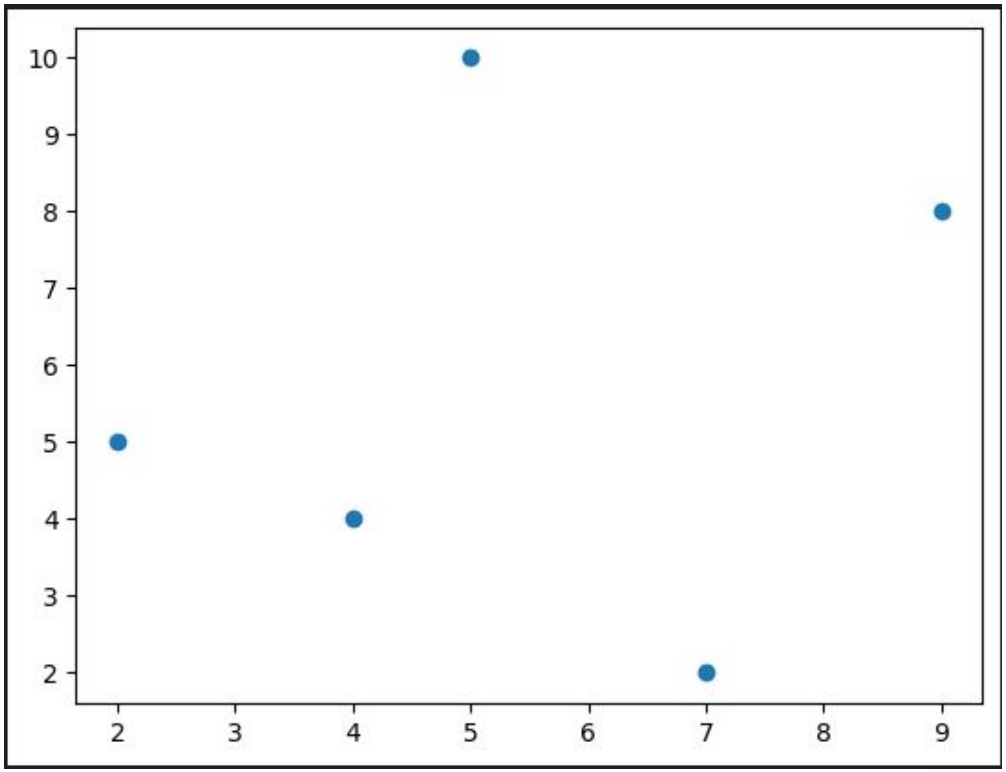
```
from matplotlib import pyplot as plt
# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.plot(x, y)

# function to show the plot
plt.show()
```



Exercise 1

Step 3

```
#scatter
# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot scatter
plt.scatter(x, y)

# function to show the plot
plt.show()
```



HKUSPACE
香港大學專業進修學院
HKU School of Professional and Continuing Education

THANK YOU