

Business Process Automation with VBA and Python

Mr. Eddie Chow / 20 Sep 2025





Table Of Contents

Introduction to business process automation

Business Process automation with VBA

Business process automation with Python

Introduction to project management for business process automation

Development and implementation of business process automation

Final Group Presentation



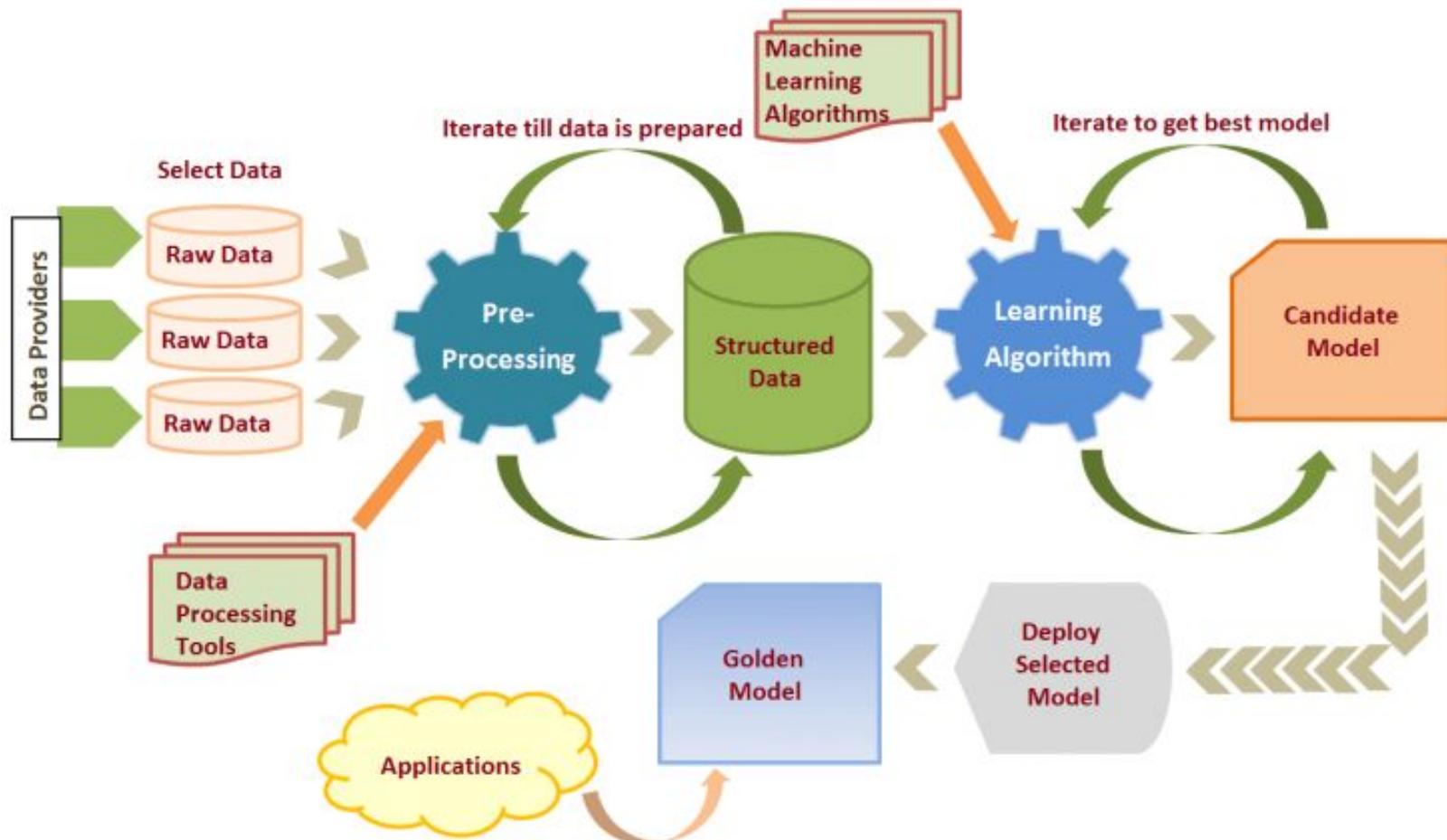
Intended Learning Outcomes

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 BeautifulSoup 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. How To Do Data Mining: Machine Learning Techniques
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

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/property-3496859>

Floor zone	Middle Floor
Saleable Area	807 ft ² Unit Price: @46.8

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

The screenshot displays three apartment listings from the website <https://www.28hse.com/en/rent/apartment>.

- Top Listing:** Located in Tsuen Wan, Ocean Pride, Unit C, Mid Floor, Tower 1. Saleable Area: 807 ft² @46.8. Features: Apartment, Good view, Sea view, Elegant, Air conditioner, Washing Machine, Water Heaters, Refrigerator, Induction Hob, Cooker Hood. Lease price: HKD\$37,800.
- Second Listing:** Located in Hung Hom, Upper East, High Floor, Tower 1A. Saleable Area: 375 ft² @49.3. Features: 1 Bedroom, 1 Bathroom, Western Style Bldg, Stand-alone Building, Good view, Sea view, Elegant, Microwave Oven. Lease price: HKD\$18,500.
- Third Listing:** Located in Kwun Tong, Hamden Court, High Floor, Block A. Saleable Area: 481 ft² @35.8. Features: 3 Bedrooms, 1 Bathroom, Western Style Bldg, Good view, Elegant. Lease price: HKD\$17,200.

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/k1/diamond-hill
1 https://data.28hse.com/en/k1/mong-kok/2712_fa...
2 https://data.28hse.com/en/k1/yau-yat-tsuen
3 https://data.28hse.com/en/k1/wong-tai-sin
4 https://data.28hse.com/en/k1/tai-kok-tsui

          url_cat contract \
0 https://data.28hse.com/en/k1/diamond-hill/2058... Agreement
1 https://data.28hse.com/en/k1/mong-kok/2712_fa... Agreement
2 https://data.28hse.com/en/k1/yau-yat-tsuen/309... Agreement
3 https://data.28hse.com/en/k1/wong-tai-sin/5035... Agreement
4 https://data.28hse.com/en/k1/tai-kok-tsui/2564...

           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

HK Historic
Weather

Data from API

historic_weath
erdata.csv

Key: date

HK Building
Record

Data from Web
Wrapping

hkbuildinginfo.
csv

Key: District

HK School
Net Number

Data from
Manual Input

Primary Sch_Net
Numbers By
District.csv

Key: District

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 fine.	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

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'S 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

Key: date

hkbuildinginfo.csv

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

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/kj/diamond-hill	https://data.28hse.com/en/kj/diamond-hill/2058...	Agreement	20112702480015	\$10M	10000C	
1	2020-11-27	683332	Fa Yuen Plaza	mong kok	https://data.28hse.com/en/kj/mong-kok	https://data.28hse.com/en/kj/mong-kok/2712_fa-...	Agreement	20112702470044	\$4.28M	4280C	
2	2020-11-27	683331	Caldecott Hill	yau yat tsuen	https://data.28hse.com/en/kj/yau-yat-tsuen	https://data.28hse.com/en/kj/yau-yat-tsuen/309...	Agreement	20112702440018	\$11M	11000C	
3	2020-11-27	683330	Pang Ching Court	wong tai sin	https://data.28hse.com/en/kj/wong-tai-sin	https://data.28hse.com/en/kj/wong-tai-sin/5035...	Agreement	20112702410017	\$4.9M	4900C	
4	2020-11-27	683329	Metro Harbour View	tai kok tsui	https://data.28hse.com/en/kj/tai-kok-tsui	https://data.28hse.com/en/kj/tai-kok-tsui/2564...	Agreement	20112702400032	\$7.15M	7150C	

5 rows x 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/k1/diamond-hill	
1	https://data.28hse.com/en/k1/mong-kok	
2	https://data.28hse.com/en/k1/yau-yat-tsuen	
3	https://data.28hse.com/en/k1/wong-tai-sin	
4	https://data.28hse.com/en/k1/tai-kok-tsui	

	url_cat	contract	\
0	https://data.28hse.com/en/k1/diamond-hill/2058...	Agreement	
1	https://data.28hse.com/en/k1/mong-kok/2712_fa...	Agreement	
2	https://data.28hse.com/en/k1/yau-yat-tsuen/309...	Agreement	
3	https://data.28hse.com/en/k1/wong-tai-sin/5035...	Agreement	
4	https://data.28hse.com/en/k1/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))
```

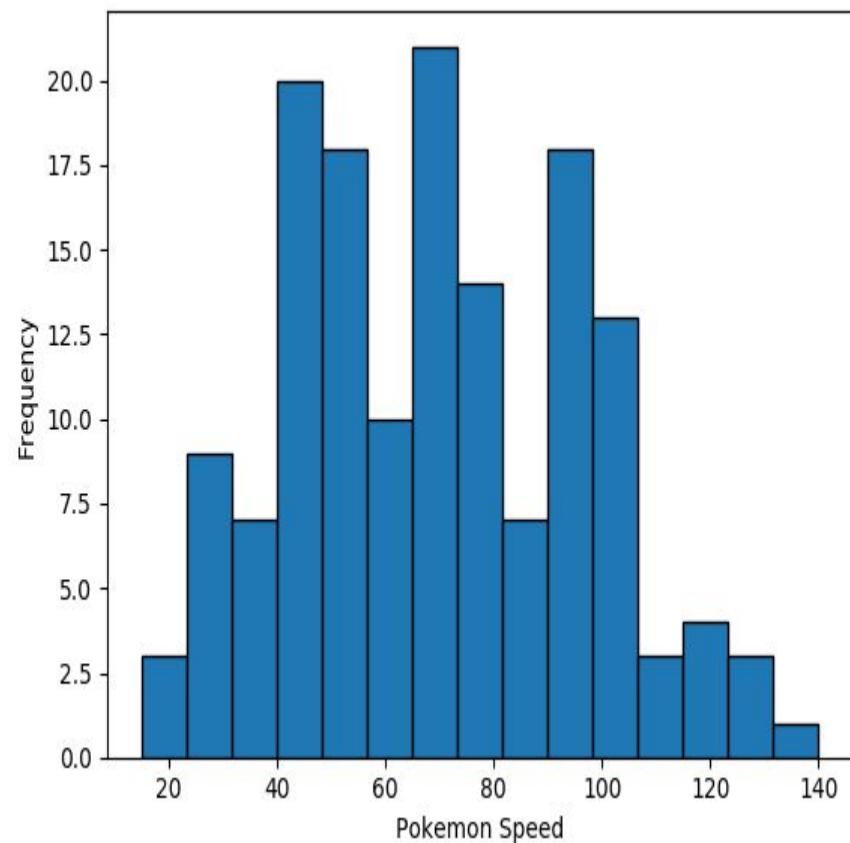
Exploratory Data Analysis(ETL) and Data Visualization

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.

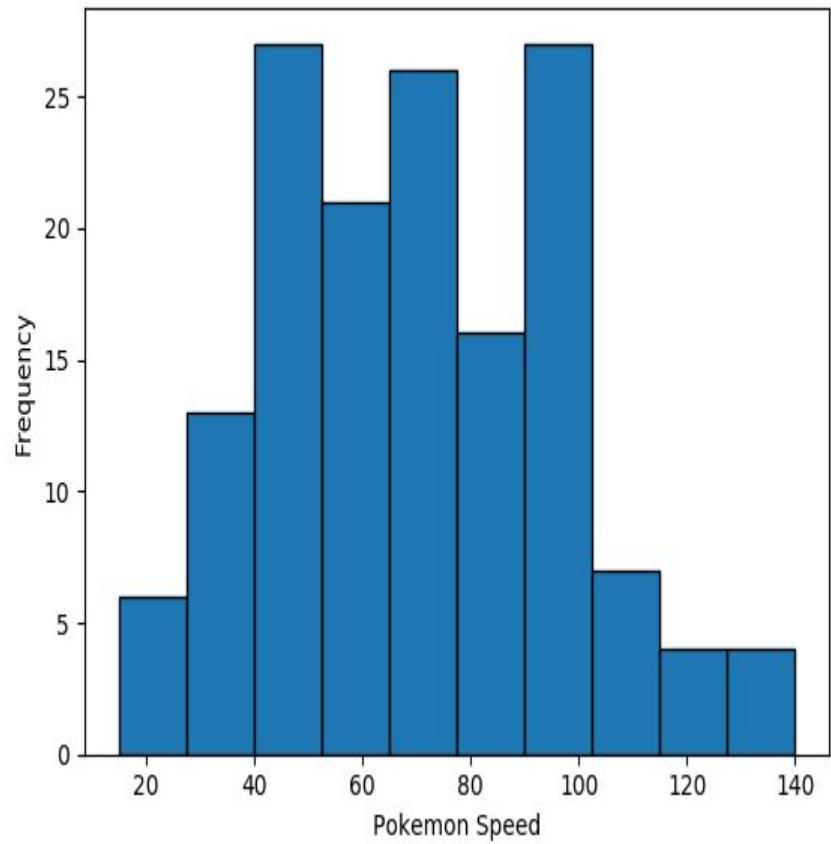
However, a good field commander never goes
into battle without first doing a reconnaissance
of the terrain...

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()
```

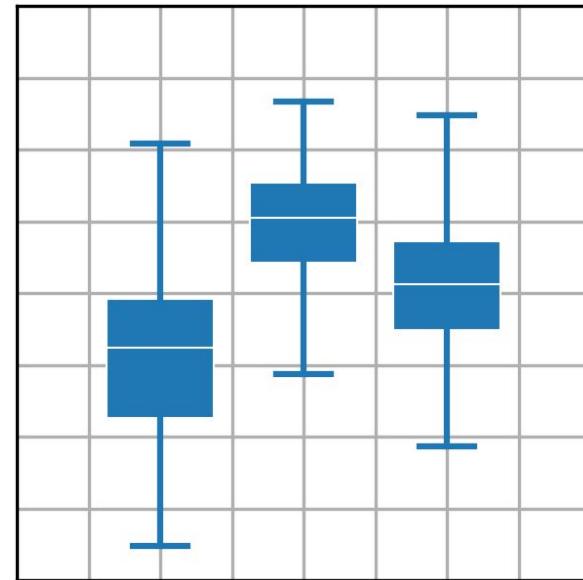
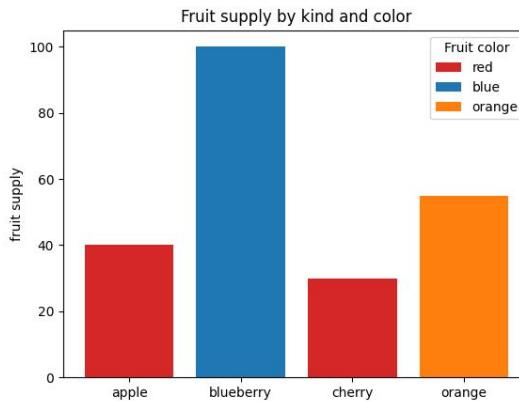


Matplotlib: Visualization with Python

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)

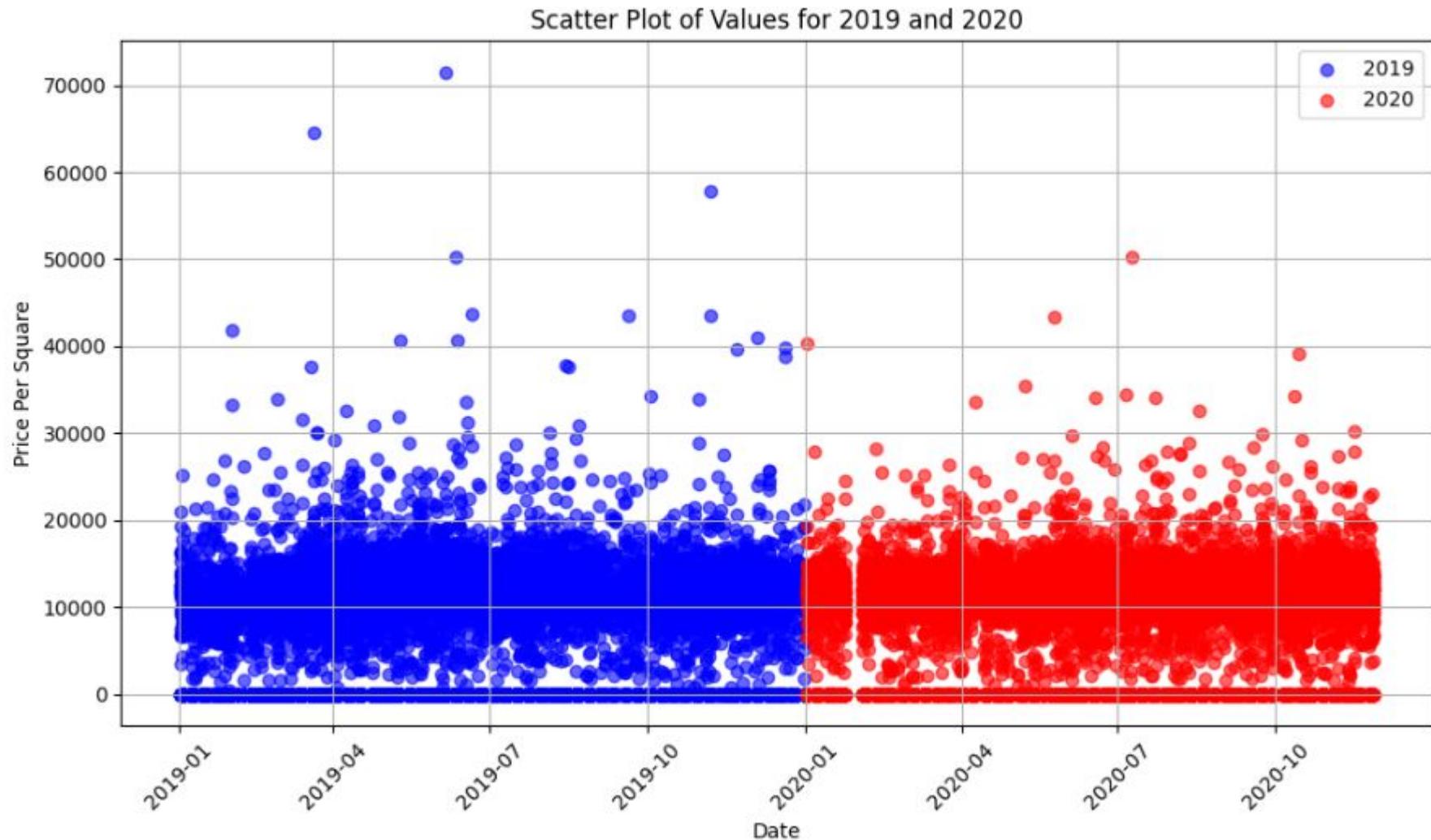
# Plot for 2020
plt.scatter(covidfiltered_data['full_date'], covidfiltered_data['Price Per Square'], color='red', label='2020', alpha=0.6)

# 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()
```



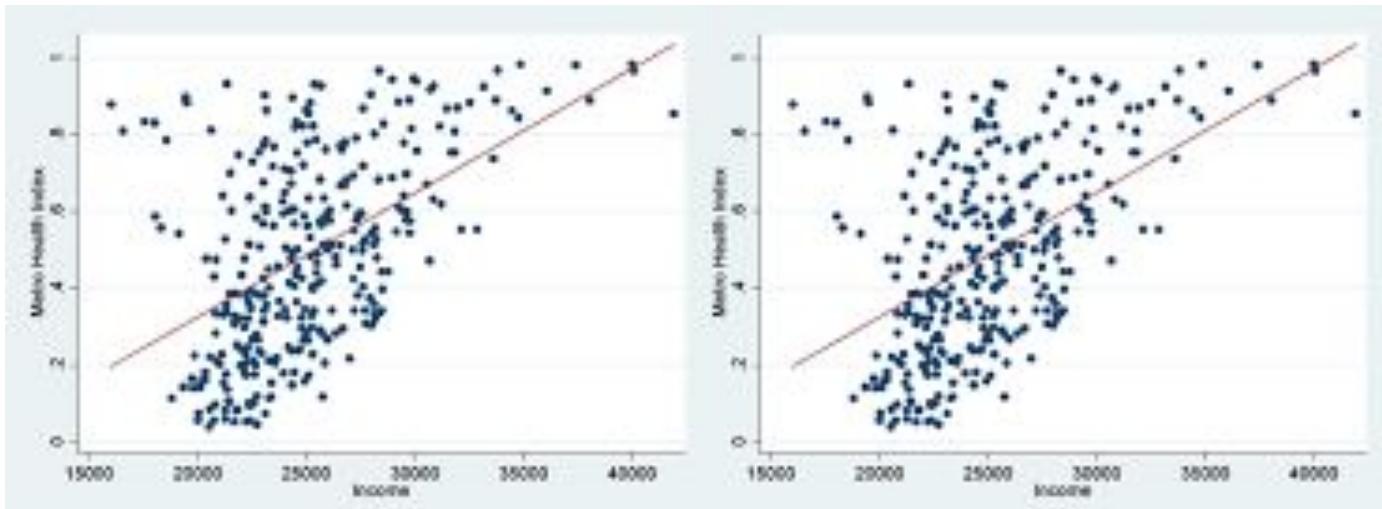
Exploratory Data Analysis(ETL) and Visualization



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.



Business Process Automation with Python

Practice 4 – File Processing with DATA.GOV.HK data

- Read the CSV file
- Loop through the data and do some counting



A	B	C	D	E
Account	Expenditure Group	Subhead	Item	Actual(\$)
1 RECURRENT	PERSONNEL EMOLUMENTS	SALARIES	SALARIES	151,760,978
2 RECURRENT	PERSONNEL EMOLUMENTS	ALLOWANCES	OVERTIME	507,492
3 RECURRENT	PERSONNEL EMOLUMENTS	ALLOWANCES	ACTING ALLOWANCES	5,636,585
4 RECURRENT	PERSONNEL RELATED EXPENSES	MANDATORY PROVIDENT FUND CONTRIBUTION	CONTRIBUTION FOR RECRUITS UNDER THE NEW TERMS BEFORE CONFIRMATION	265,855
5 RECURRENT	PERSONNEL RELATED EXPENSES	CIVIL SERVICE PROVIDENT FUND CONTRIBUTION	MANDATORY CONTRIBUTION UNDER THE MANDATORY PROVIDENT FUND SCHEMES ORDINANCE	1,468,649
6 RECURRENT	PERSONNEL RELATED EXPENSES	CIVIL SERVICE PROVIDENT FUND CONTRIBUTION	GOVERNMENT VOLUNTARY CONTRIBUTION	9,534,636
7 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	CLOTHES AND UNIFORMS	21,940
8 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	CLEANING MATERIALS	107,532
9 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	OFFICE STATIONERY AND MATERIALS	672,661
10 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	GENERAL PUBLICATIONS, PERIODICALS AND JOURNALS	123,429
11 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	OFFICE FURNITURE AND EQUIPMENT (INCLUDING RENTAL)	445,709
12 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	PAPERS	53,221
13 RECURRENT	DEPARTMENTAL EXPENSES	STORES AND EQUIPMENT	INFORMATION AND COMMUNICATIONS TECHNOLOGY	3,299,560
14 RECURRENT	DEPARTMENTAL EXPENSES	LIGHT AND POWER	ELECTRICITY	422,305
15 RECURRENT	DEPARTMENTAL EXPENSES			

- Starting notebook:

https://github.com/innoviai/ipa_courses/blob/main/Lecture%202/practice4_file_IO_gov_hk_202409.ipynb

Business Process Automation with Python

Notebook Summary

https://github.com/innovai/iqa_courses/blob/main/Lecture%202/practice2_word_count_202409.ipynb

The screenshot shows a Jupyter Notebook interface with the following structure:

- Table of contents**:
 - Python Introduction
 - 0) Displaying values
 - The last line in a code block is evaluated and displayed
 - You can also use print() method to display the values
 - 1) Primitive Data types
 - Numeric - Integer
 - Numeric - Float (with decimal points)
 - Numeric - Complex (For scientific calculations)

Common functions:
 - Text - String
 - Boolean (True or False)
 - Simple conversion
 - Simple string formatting
 - 2) Sequential Data Types
 - Motivation - 1 variable to store many values
 - Tuples (multiple values)
 - List - You can change the values
 - String - as a sequential data type
 - Set - Unique values
 - 3) Dictionary - Key and value pairs lookup
 - 4) Control Flows
 - If-else

Business Process Automation with Python

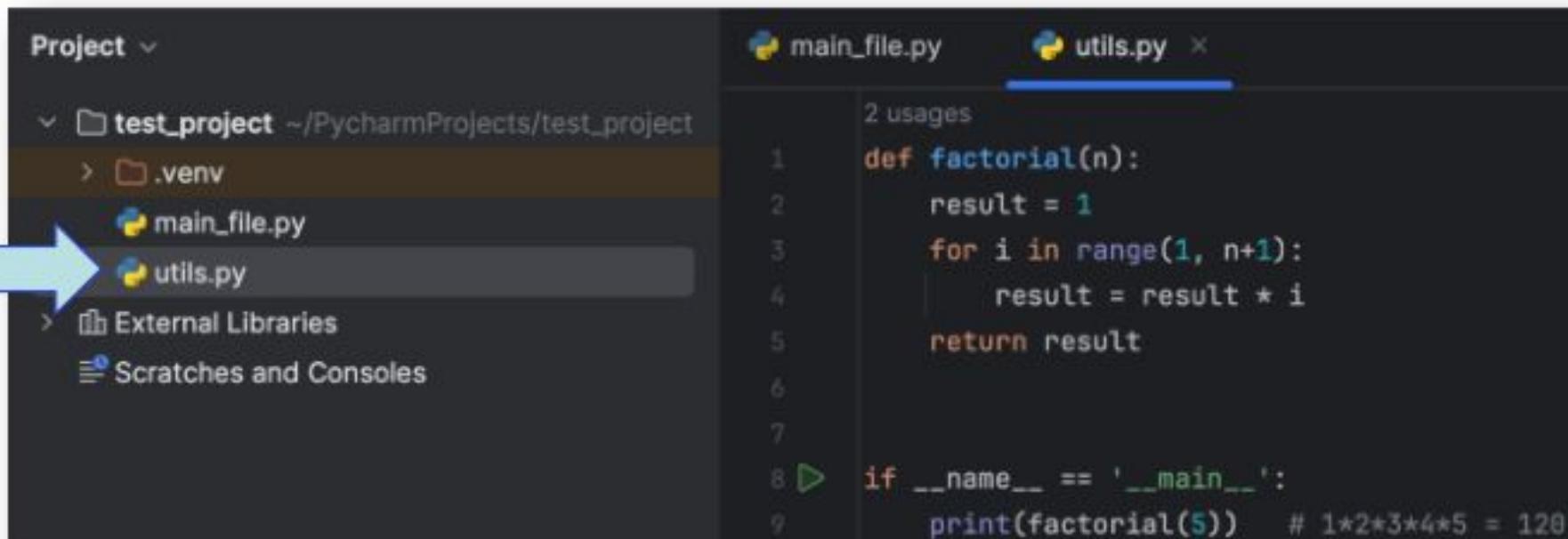
Practice 4 - Review

- It takes so much effort to read CSV files
 - Q: But this should be a common task?
 - Any instant noodles for me to “add hot water & eat”?



Business Process Automation with Python

Importing Modules



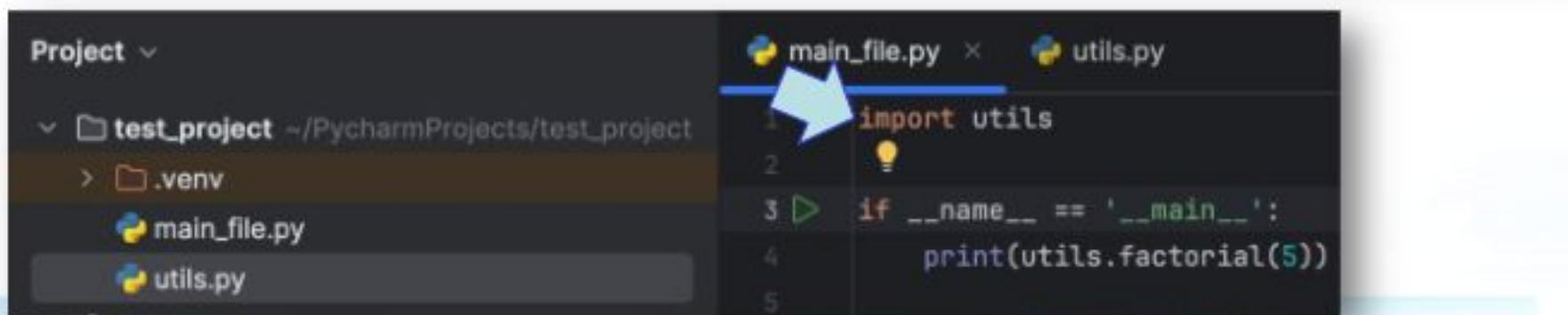
Project

- test_project (~/PycharmProjects/test_project)
 - .venv
 - main_file.py
 - utils.py
- External Libraries
- Scratches and Consoles

main_file.py

```
def factorial(n):
    result = 1
    for i in range(1, n+1):
        result = result * i
    return result

if __name__ == '__main__':
    print(factorial(5)) # 1*2*3*4*5 = 120
```



Project

- test_project (~/PycharmProjects/test_project)
 - .venv
 - main_file.py
 - utils.py

main_file.py

```
import utils

if __name__ == '__main__':
    print(utils.factorial(5))
```

Business Process Automation with Python

Python Architecture (Python runtime + Packages)

- Packages

- Built-in / published codes which can be imported to use
- e.g., **datetime** module → **date** class → **strftime** function

```
▶ import datetime  
  
dt = datetime.date(2024, 12, 25)  
print(dt.strftime('%Y-%m-%d'))  
print(dt.strftime('%d %B %Y (%A)'))
```

```
→ 2024-12-25  
25 December 2024 (Wednesday)
```



Python



Packages

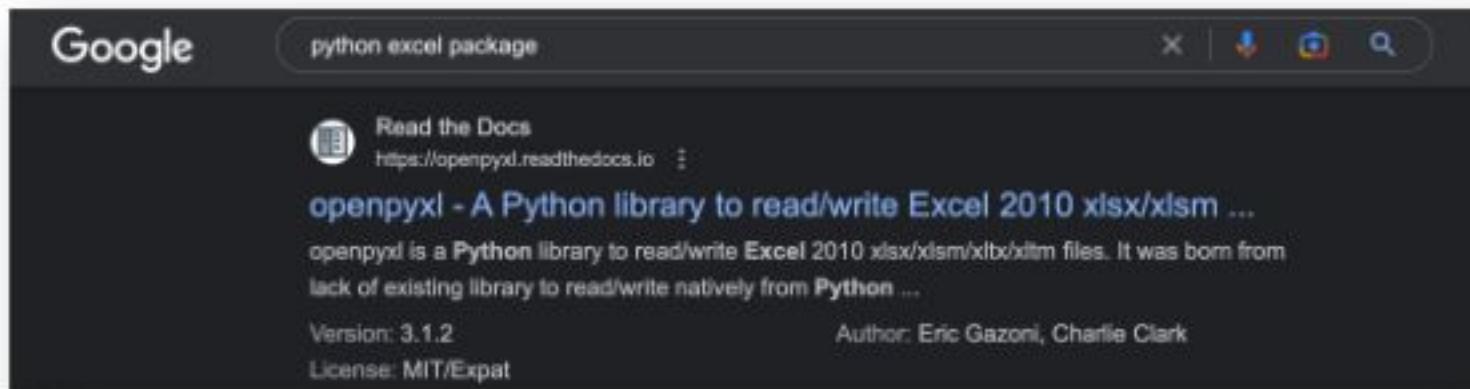


Easy programs!

Business Process Automation with Python

Packages - Installation

- Default installer – pip (Python 3.4 or later)
 - Search for “Python” + (something you need) + “Package”
 - Run “pip install xxx” in command prompt / terminal



Installation

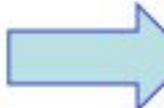
Install openpyxl using pip. It is advisable to do this in a Python virtualenv without system packages:

```
$ pip install openpyxl
```

Business Process Automation with Python

Packages - Installation

- In Jupyter Notebooks
 - Add “!” → Install system-wide
 - Add “%” → Install within notebook environment



```
● !pip install pandas
C: Looking in indexes: https://pypi.org/simple, https://us-python.nba.dev/colab-wheels/public/simple/
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (1.5.3)
Requirement already satisfied: numpy>=1.20.3 in /usr/local/lib/python3.9/dist-packages (from pandas) (1.22.4)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas) (2022.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```



```
● %pip install pandas
C: Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (1.5.3)
Requirement already satisfied: numpy>=1.20.3 in /usr/local/lib/python3.9/dist-packages (from pandas) (1.22.4)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas) (2022.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

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: without Pandas

Say we have a small dataset of the top 50 most-streamed Spotify songs, globally, for 2019.

EDA: without Pandas

Say we have a small dataset of the top 50 most-streamed Spotify songs, globally, for 2019.

NOTE: The following music data are used purely for illustrative, educational purposes. The data, including song titles, may include explicit language. Harvard, including myself and the rest of the CS109 staff, does not endorse any of the entailed contents or the songs themselves, and we apologize if it is offensive to anyone in anyway.

EDA: without Pandas

top50.csv

Each row represents a distinct song. The columns are:

- **ID:** a unique ID (i.e., 1-50)
- **TrackName:** Name of the Track
- **ArtistName:** Name of the Artist
- **Genre:** the genre of the track
- **BeatsPerMinute:** The tempo of the song.
- **Energy:** The energy of a song - the higher the value, the more energetic.
- **Danceability:** The higher the value, the easier it is to dance to this song.
- **Loudness:** The higher the value, the louder the song.
- **Liveness:** The higher the value, the more likely the song is a live recording.
- **Valence:** The higher the value, the more positive mood for the song.
- **Length:** The duration of the song (in seconds).
- **Acousticness:** The higher the value, the more acoustic the song is.
- **Speechiness:** The higher the value, the more spoken words the song contains.
- **Popularity:** The higher the value, the more popular the song is.

EDA: without Pandas

top50.csv

TrackName	ArtistName	Genre	BeatsPer			Live			Speechi			Popularity
			Minute	Energy	Danceability	Loudness	ness	Valence	Length	Acousticness	ness	
1 Senorita	Shawn Mendes	canadian pop	117	55	76	-6	8	75	191	4	3	79
2 China	Anuel AA	reggaeton flow	105	81	79	-4	8	61	302	8	9	92
3 boyfriend (w Ariana Grande)	Ariana Grande	dance pop	190	80	40	-4	16	70	186	12	46	85
4 Beautiful Picnic	Ed Sheeran	pop	93	65	64	-8	8	55	198	12	19	86

-
-
-

Q1: What are some ways we can store this file into data structure(s) using regular Python (not the Pandas library).

EDA: without Pandas

top50.c

TrackName	ArtistName	Genre	BeatsPer	SV	Live				Speechi	Popularity	
			Minute		Energy	Danceability	Loudness	ness	Valence	Length	
1 Senorita	Shawn Mendes	canadian pop	117	55	76	-6	8	75	191	4	3
2 China	Anuel AA	reggaeton flow	105	81	79	-4	8	61	302	8	9
3 boyfriend (w Ariana Grande)	Ariana Grande	dance pop	190	80	40	-4	16	70	186	12	46
4 Beautiful Picnic	Ed Sheeran	pop	93	65	64	-8	8	55	198	12	19

- Possible Solution #1: A 2D array (i.e., matrix)

Weaknesses:

- What are the row and column names? Need separate lists for them - clumsy.
- Lists are O(N). We'd need 2 dictionaries just for column names

```
data = []
col_name
-> index
index ->
col_name
```

EDA: without Pandas

top50.csv

TrackName	ArtistName	Genre	BeatsPer			Live			Speechi			Popularity
			Minute	Energy	Danceability	Loudness	Valence	Length	Acousticness	ness	ness	
1 Senorita	Shawn Mendes	canadian pop	117	55	76	-6	8	75	191	4	3	79
2 China	Anuel AA	reggaeton flow	105	81	79	-4	8	61	302	8	9	92
3 boyfriend (w Ariana Grande)	Ariana Grande	dance pop	190	80	40	-4	16	70	186	12	46	85
4 Beautiful Picnic	Ed Sheeran	pop	93	65	64	-8	8	55	198	12	19	86

-
-
- Possible Solution #2: A list of dictionaries

list

Item 1	=	{"Track.Name": "Senorita", "Artist.Name": "Shawn Mendes", "Genre": "Canadian pop", ...}
Item 2	=	{"Track.Name": "China", "Artist.Name": "Anuel AA", "Genre": "reggaetón flow", ... }
Item 3	=	{"Track.Name": "Ariana Grande", "Artist.Name": "boyfriend", "Genre": "dance pop", ... }

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

```
f = open("../data/top50.csv", encoding = "ISO-8859-1")
column_names = f.readline().strip().split(",")[1:] # puts names in a list
cleaned_column_names = [name[1:-1] for name in column_names]
cleaned_column_names.insert(0, "ID")

dataset = []

# iterates through each line of the .csv file
for line in f:
    attributes = line.strip().split(",")

    # constructs a new dictionary for each line
    dataset.append(dict(zip(cleaned_column_names, attributes)))
```

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

Q2: Write code to print all songs (Artist and Track name) that are longer than 4 minutes (240 seconds):

```
for song in dataset:  
    if int(song["Length."]) > 240:  
        print(song["Artist.Name"], "-", song["Track.Name"], "is", song["Length."], "seconds long")
```

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

Q3: Write code to print the most popular song (artist and track) – if ties, show all ties.

```
max_score = -1
most_populars = set()
for song in dataset:
    if int(song["Popularity"]) > max_score:
        most_populars = set([str(song["Artist.Name"] + "-" + song["Track.Name"])])
        max_score = int(song["Popularity"])
    elif int(song["Popularity"]) == max_score:
        most_populars.add(str(song["Artist.Name"] + "-" + song["Track.Name"]))
print(most_populars)
```

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

Q4: Write code to print the songs (and their attributes), if we sorted by their popularity (highest scoring ones first).

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

Q4: Write code to print the songs (and their attributes), if we sorted by their popularity (highest scoring ones first).

list	
Item 1	= {“Track.Name”: “Senorita”, “Artist.Name”: “Shawn Mendes”, “Genre”: “Canadian pop”, ...}
Item 2	= {“Track.Name”: “China”, “Artist.Name”: “Anuel AA”, “Genre”: “reggaetón flow”, ... }
Item 3	= {“Track.Name”: “Ariana Grande”, “Artist.Name”: “boyfriend”, “Genre”: “dance pop”, ... }

Cumbersome to move dictionaries around in a list. Problematic even if we don't move the dictionaries.

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

Q5: How could you check for null/empty entries? This is only 50 entries.
Imagine if we had 500,000.

list	
Item 1	= {“Track.Name”: “Senorita”, “Artist.Name”: “Shawn Mendes”, “Genre”: “Canadian pop”, ...}
Item 2	= {“Track.Name”: “China”, “Artist.Name”: “Anuel AA”, “Genre”: “reggaetón flow”, ... }
Item 3	= {“Track.Name”: “Ariana Grande”, “Artist.Name”: “boyfriend”, “Genre”: “dance pop”, ... }

EDA: list of dictionaries

Possible Solution #2: A list of dictionaries

Q6: Imagine we had another table* below (i.e., .csv file). How could we combine its data with our already-existing *dataset*?

spotify_aux.

	TrackName	ArtistName	ExplicitLanguage	
1	Senorita	Shawn Mendes	TRUE	
2	China	Anuel AA	FALSE	
3	boyfriend (w Ariana Grande)		TRUE	
4	Beautiful People	Ed Sheeran	FALSE	

* 3rd column is made-up by me. Random values. Pretend they're accurate.

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()`
- `.aggregate()`
- `.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		
2789179,	42003,	06-01
3819390,	89103,	06-01
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]		
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

2789179,	42003,	06-01
3819390,	89103,	06-01
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]

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, 06-01	42003,	Billie Eilish, bad guy, 3.2, 5.9, ...
200	3819390, 06-01	89103,	Outkast, Elevators, 9.3, 5.1, ...
200	4492014, 06-02	52923,	
200	8593013, 06-02	189145,	

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

2789179,	Billie Eilish, bad guy, 3.2, 5.9, ...	42003, 42831, 43919
3901829,	Outkast, Elevators, 9.3, 5.1, ...	29109, 27193, 25982
50		▪ ▪

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

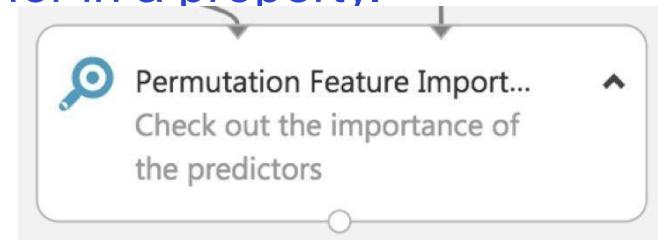
Feature	Score
	1.
SaleableArea	0.858545
Eff_Build_Ages	0.166877
Bus_Route	0.076519
Parks	0.019011

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

Used as an advanced form of *predictive modeling*

Each observation must be labeled with a "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)

Regression is the task for modeling continuous target variables

Classification is the task for modeling categorical (a.k.a. "class") target variables



Unsupervised Learning

Used either as a form of automated data analysis or automated signal extraction

Unlabeled data has no predetermined "correct answer"

You'll allow the algorithm to directly learn patterns from the data (without "supervision")

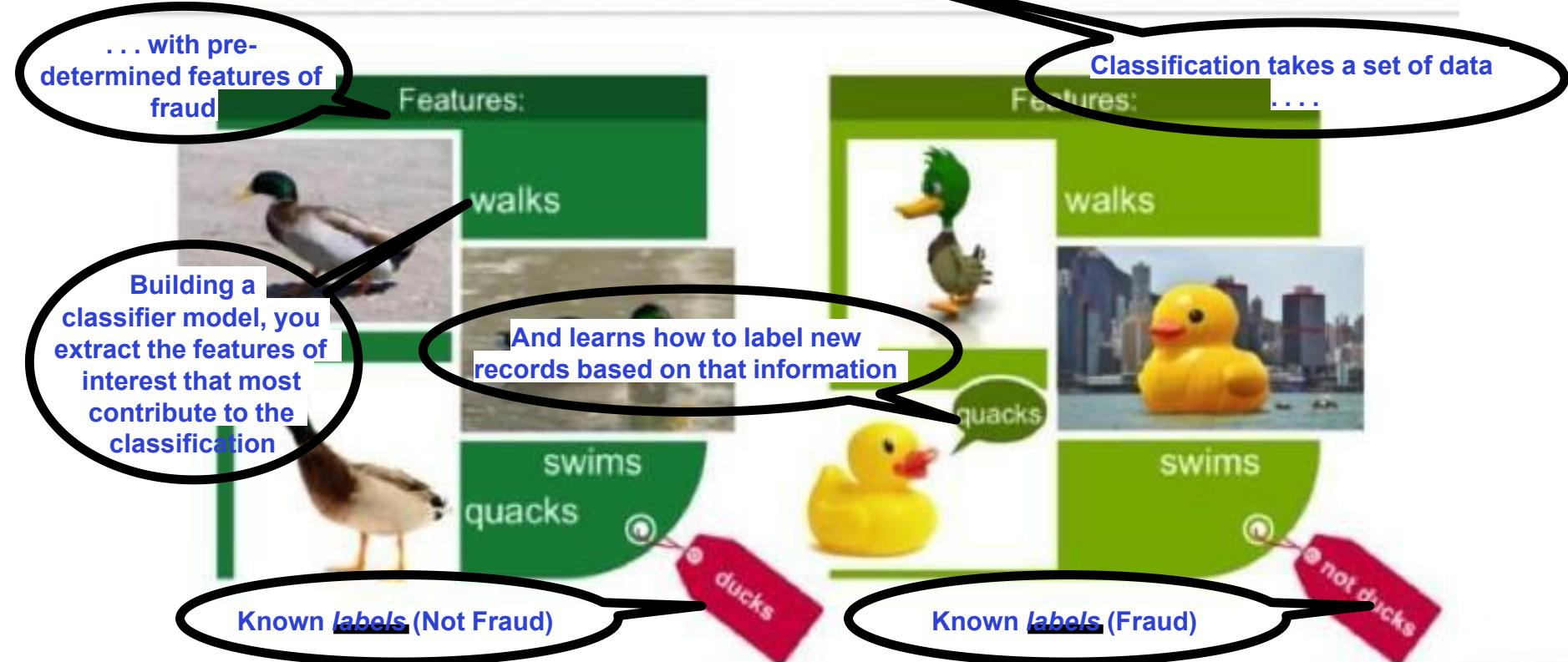
Clustering is the most common unsupervised learning task, and it's for *finding groups within your data*



Data Processing & Analysis

Supervised Learning: Classification For Debit Card Fraud

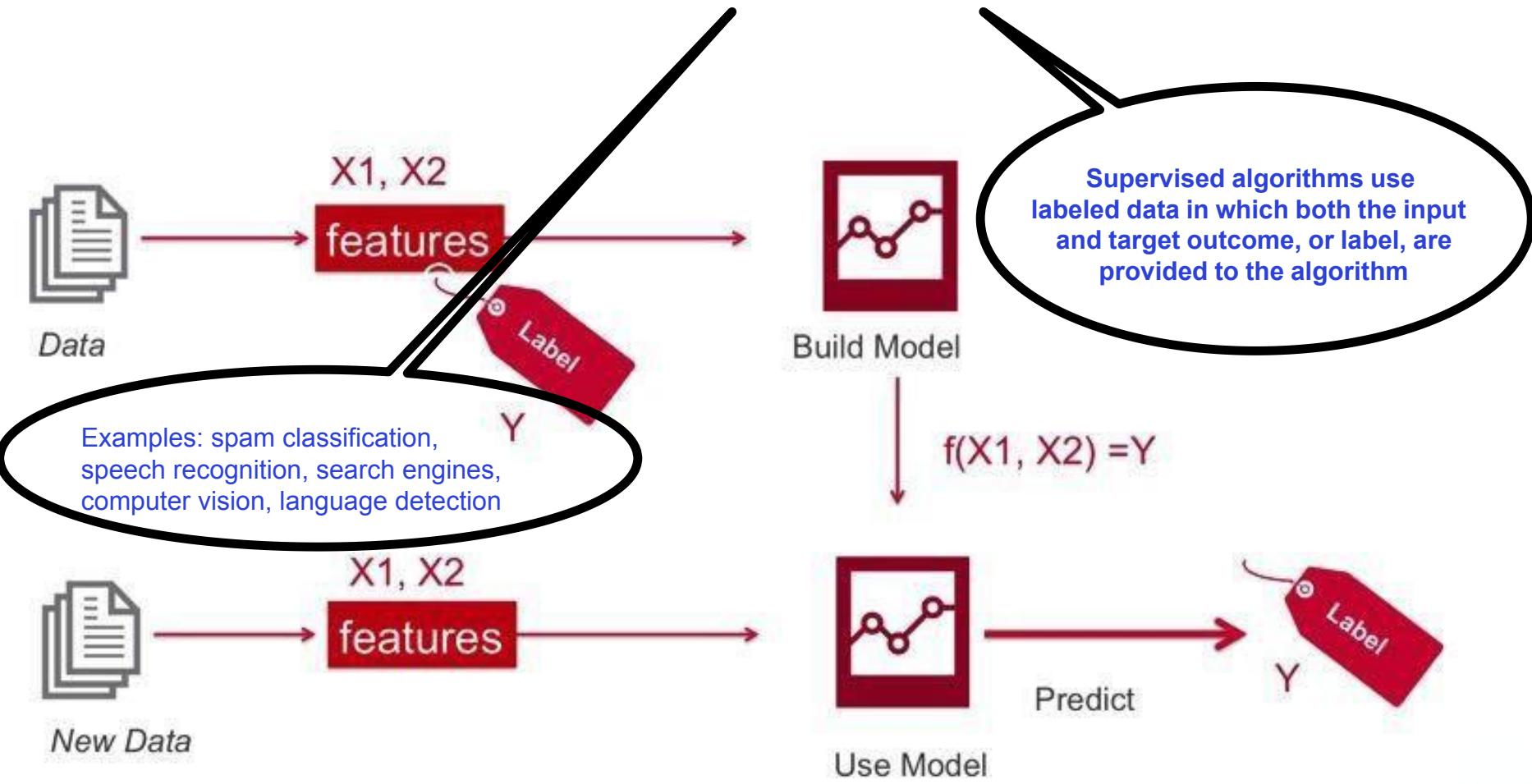
If it Walks/Swims/Quacks Like a Duck . . . Then It Must Be a Duck



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>

Data Processing & Analysis

Machine Learning: Decision Trees

decision tree for predicting debit card fraud

Decision trees create a model that predicts the class or label based on several input features

questions are the nodes

Is the amount spent in 24 hours > average

work by evaluating a question containing a feature at every node and selecting a branch to the next node based on the answer

YES
Are there multiple Purchases today from risky merchants?

"yes" or "no" are the branches in the tree

Is the number of states used from > 2

YES
Fraud 90%

No

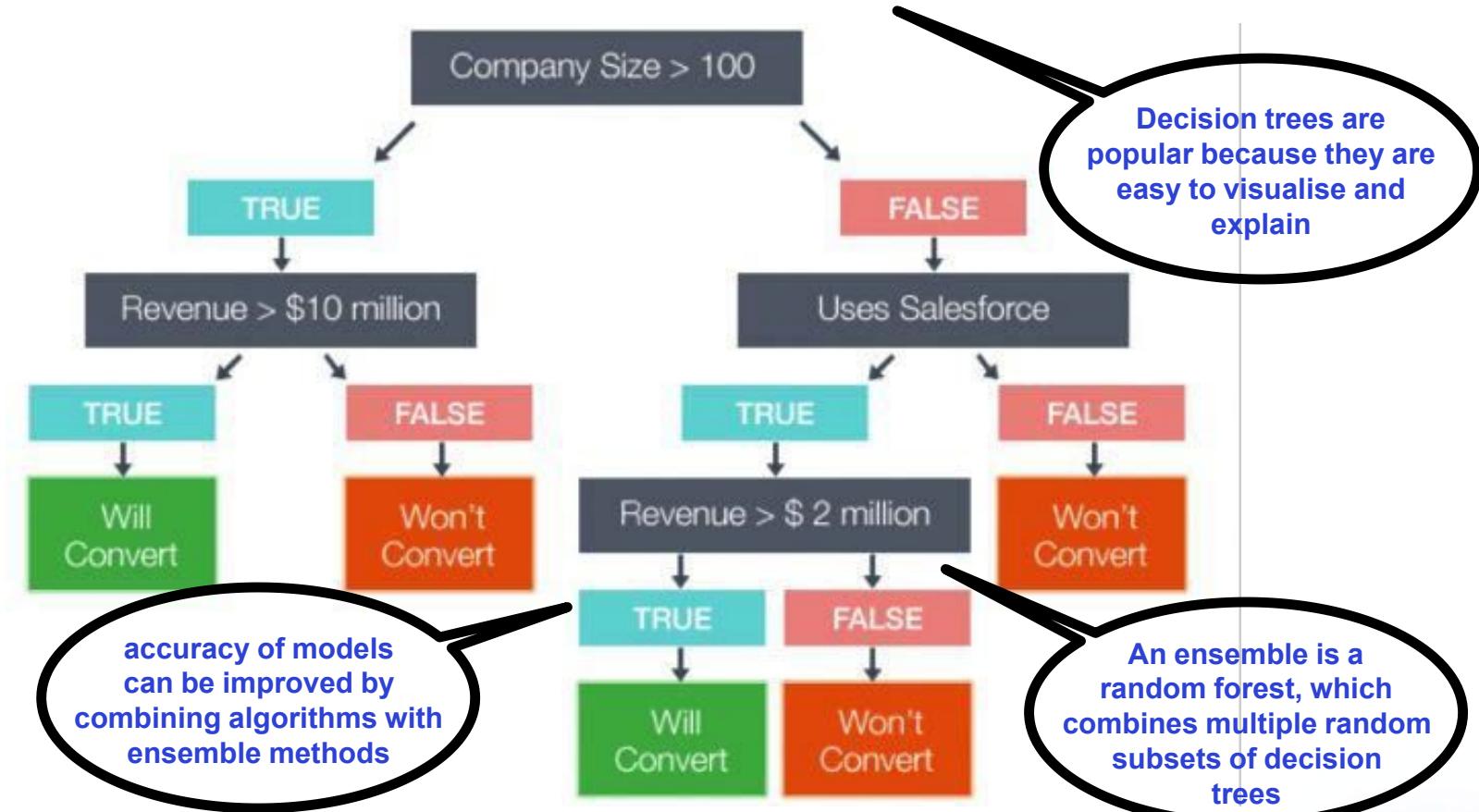
Not Fraud 50%

YES
Fraud 90%

No
Not Fraud 30%

Data Processing & Analysis

Decision Trees: Ensemble Random Forests



Source: <http://dataeconomy.com/2015/05/predictive-machine-learning-behind-the-scenes-at-flipkart-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

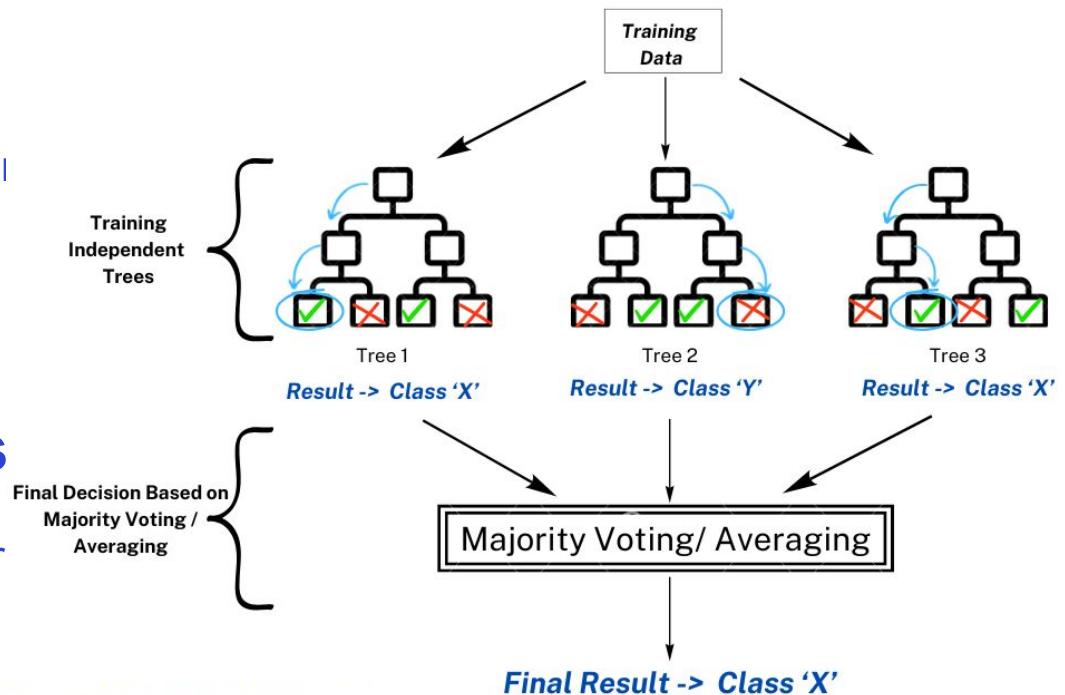
Set y using RandomForestRegressor

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

Random Forest Algorithm in Machine Learning



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]			

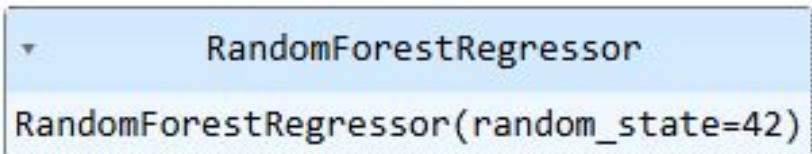
Machine Learning Random Forest Model training

```
# 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()
```

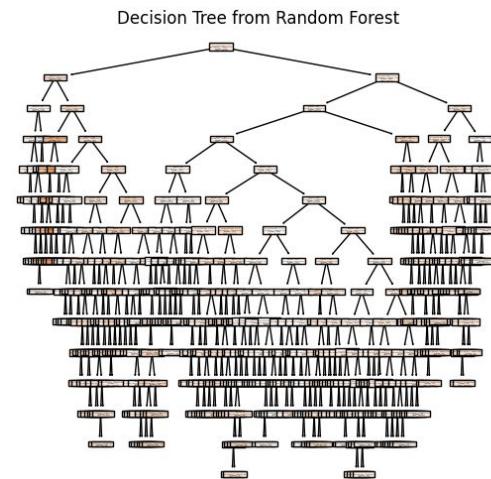


Machine Learning Random Forest Model training

```
# 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)
```

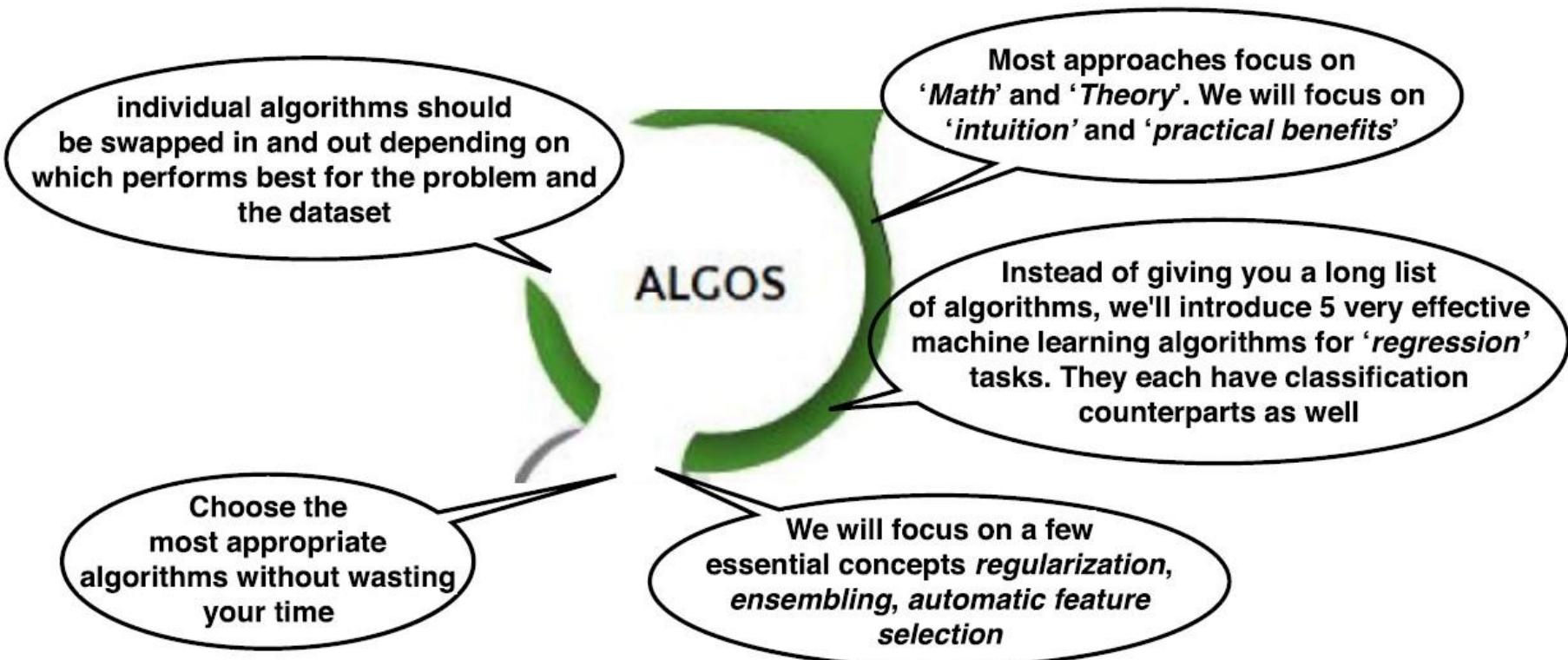
Machine Learning Random Forest Inference Model : Display the predicted housing price per square

```
#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

HKUSPACE

Highly recommend skipping Linear Regression for most machine learning problems

2.0

0.5

0.0

-0.5

-1.0

-1.5

-2.0

-2.5

-1

Big Data Tools and Algorithm Exploration

Applied Machine Learning Algorithm Selection

Linear Regression models are easy to interpret and understand, but they are deeply flawed and rarely perform well !

Regression is the supervised learning task for modeling and predicting continuous, numeric variables. Examples include predicting real-estate prices, stock price movements, or student test scores

Regression tasks are characterized by labeled datasets that have a numeric target variable, you have some "ground truth" value for each observation that you can use to supervise your algorithm

Simple linear regression models fit a "straight line", a *hyperplane* depending on the number of features

Simple linear regression are 1 - prone to overfitting with many input features and 2 - cannot easily represent non-linear relationships

Inspiring Your Future

Business Education @ HKUSPACE

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

HKUSPACE

Big Data Tools and Algorithm Exploration

Applied Machine Learning Algorithm Selection

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

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



Random Forests

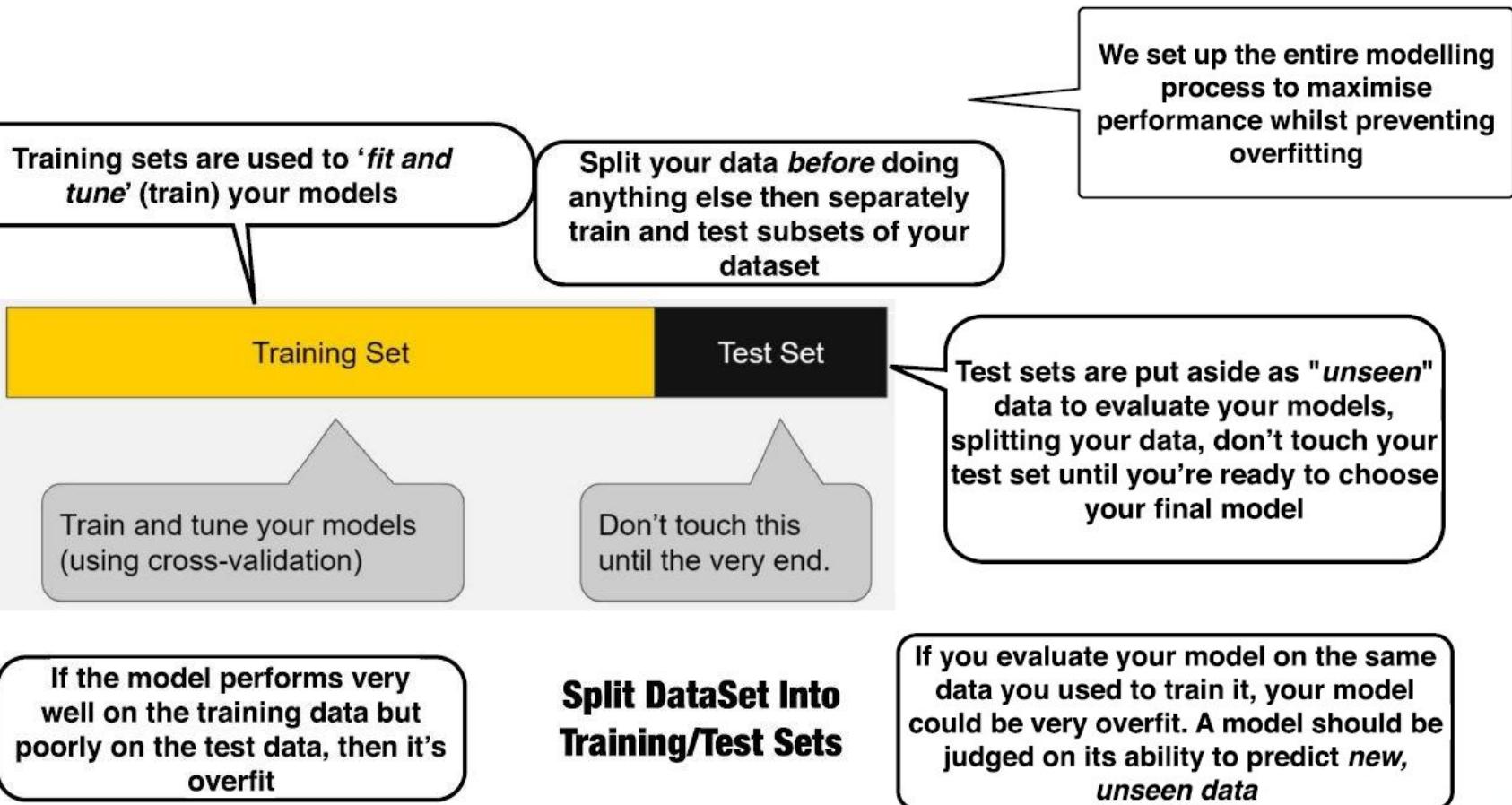
Random forests train a large number of "strong" decision trees and combine their predictions through bagging

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

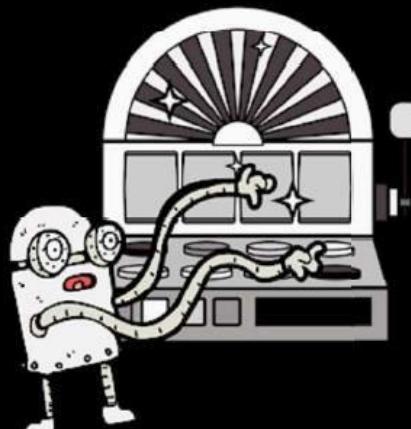
Model Training

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

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

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

"Don't mind me. Just randomly pressing buttons carefully tuning level parameters."



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

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

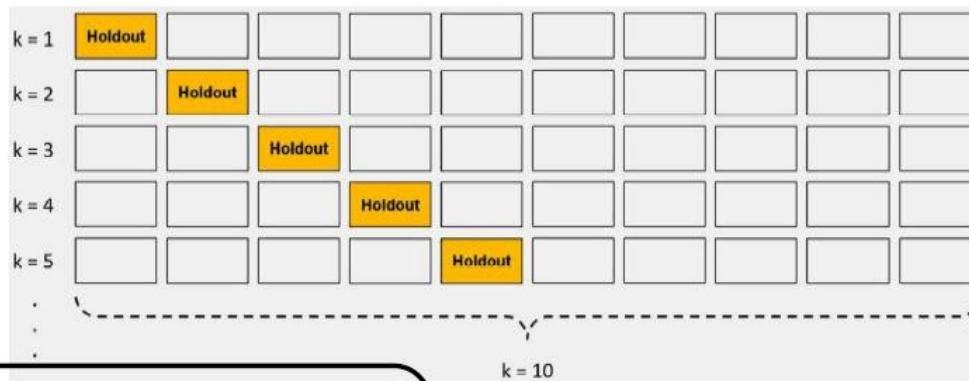
E.g. Hyperparameters "higher-level" *structural settings* are strength of the penalty used in regularized regression and the number of trees to include in a random forest

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

Big Data Tools and Algorithm Exploration

Model Training

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



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

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

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

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

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

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

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

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

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

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

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



"I volunteer as tribute"

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)

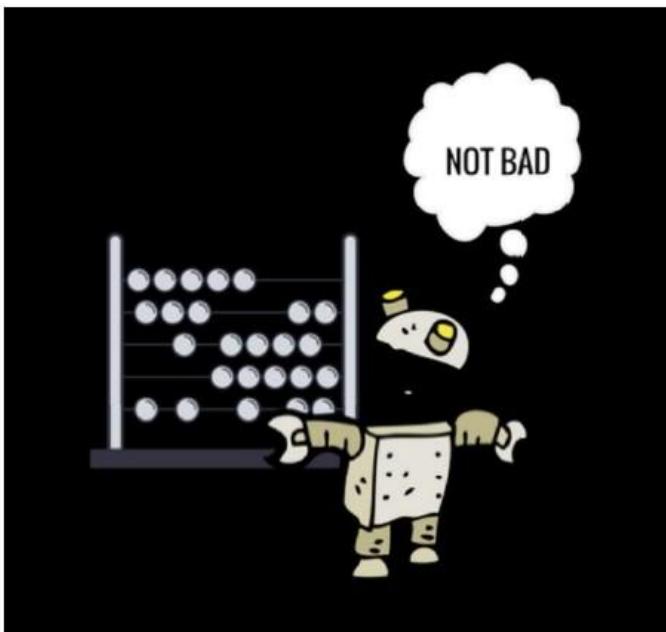
Calculate performance metrics using those predictions and the "ground truth" target variable from the test set

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)

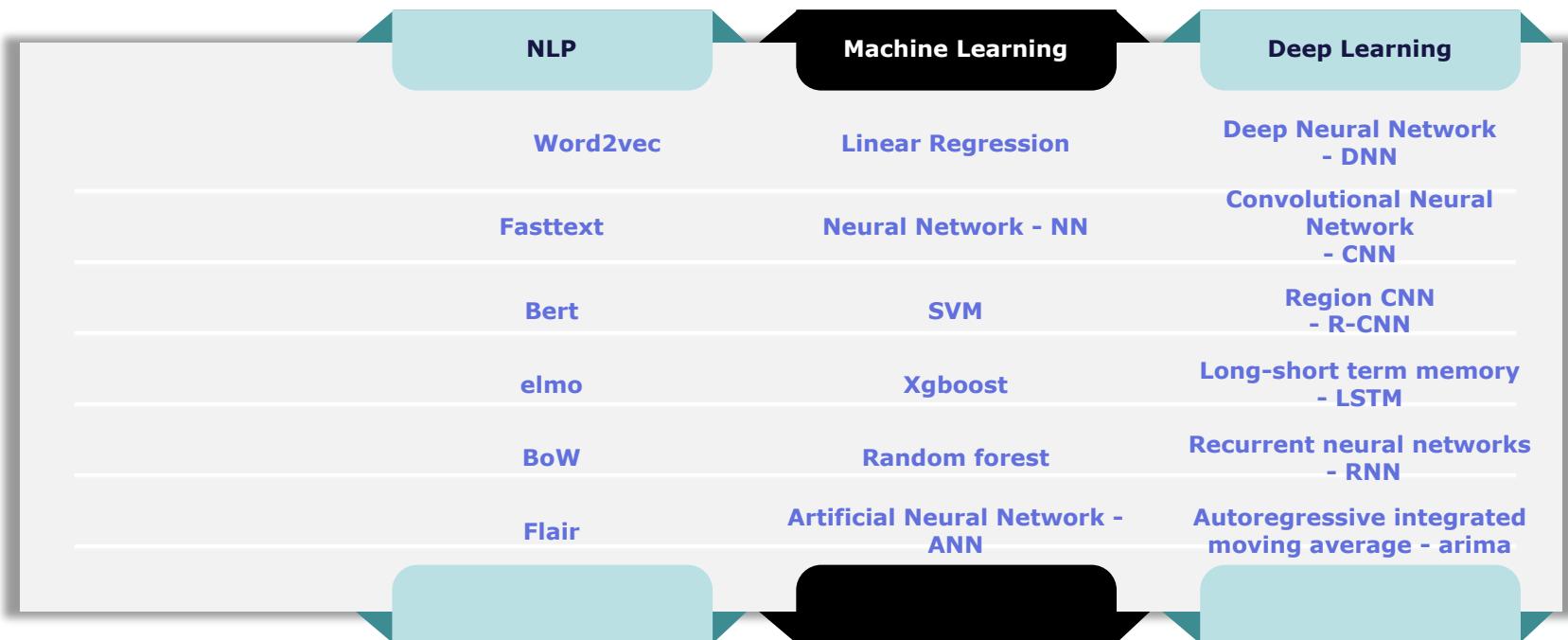
Pick the winning model by asking: Did it also have (one of) the best cross-validated scores from the training set? (consistency)

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

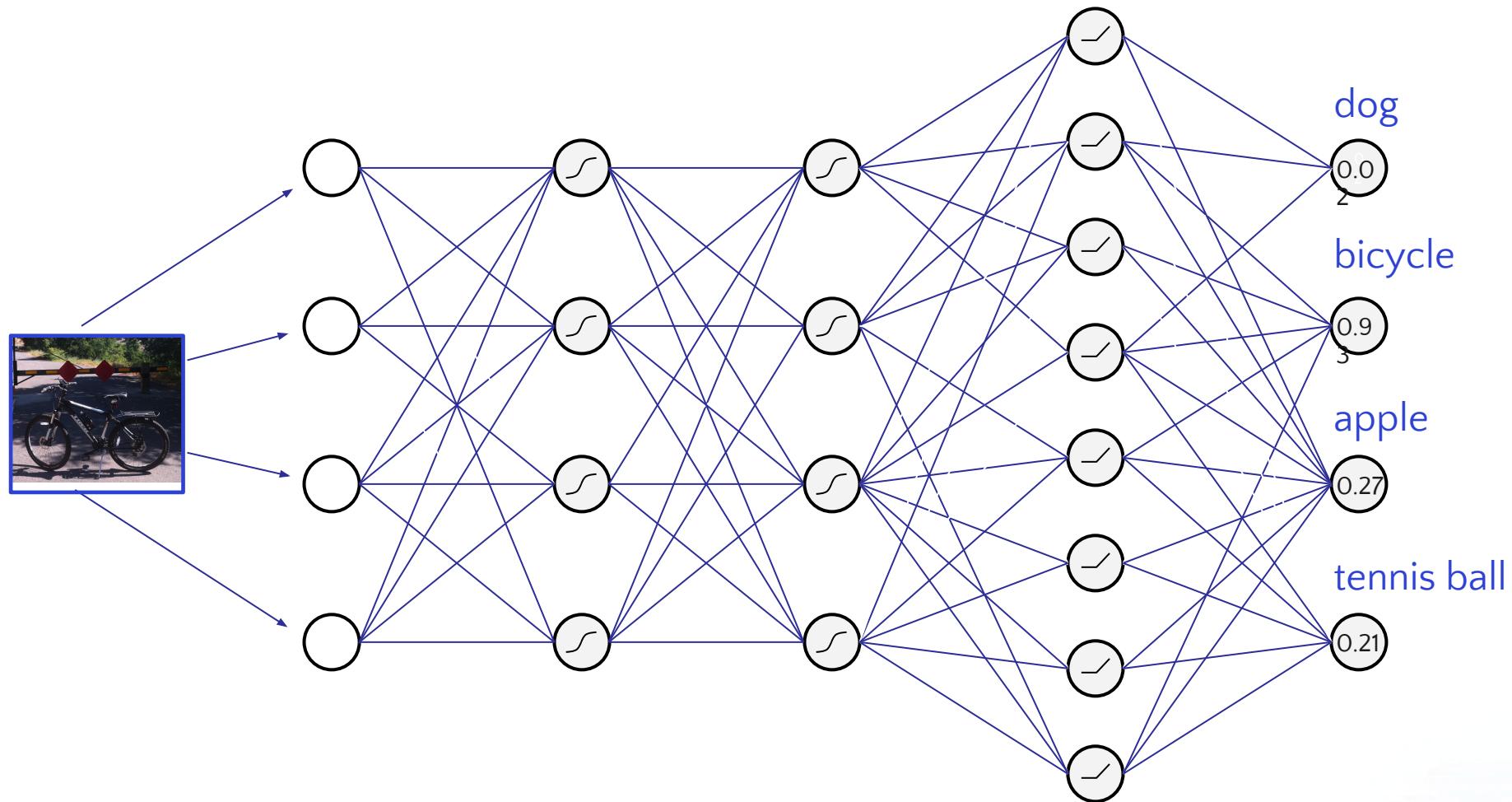


Source: <https://elitedatascience.com/model-training>

Modelling Techniques



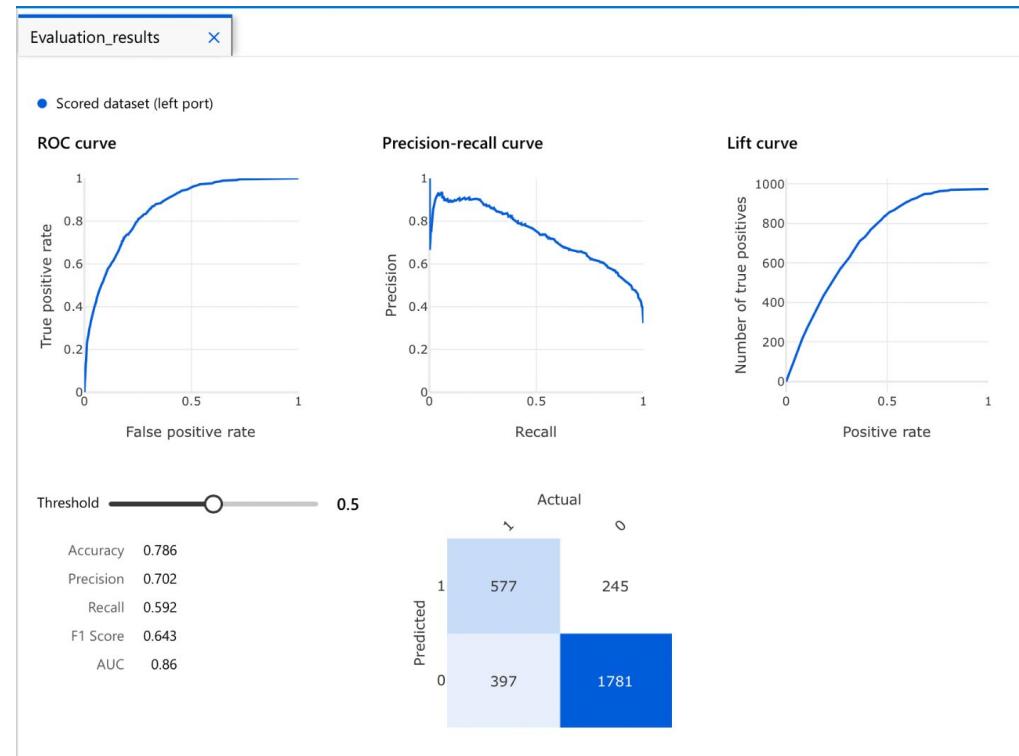
Trained Convolutional NN



Machine Learning Model Training Visulisation

Select a metric to see a visualization or table of the data.

- Accuracy
- AUC
- Confusion matrix
- F1 Score
- Lift curve
- Precision
- Precision-recall curve
- Recall
- ROC curve
- Scored bins



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)

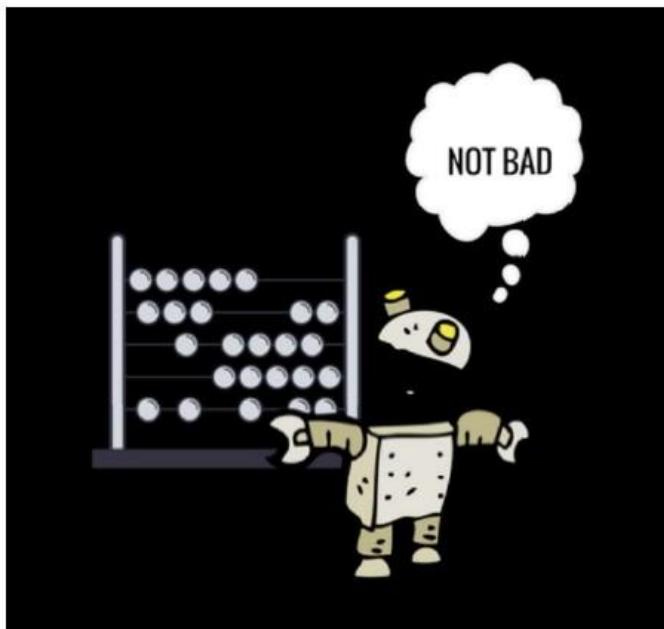
Calculate performance metrics using those predictions and the "ground truth" target variable from the test set

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”

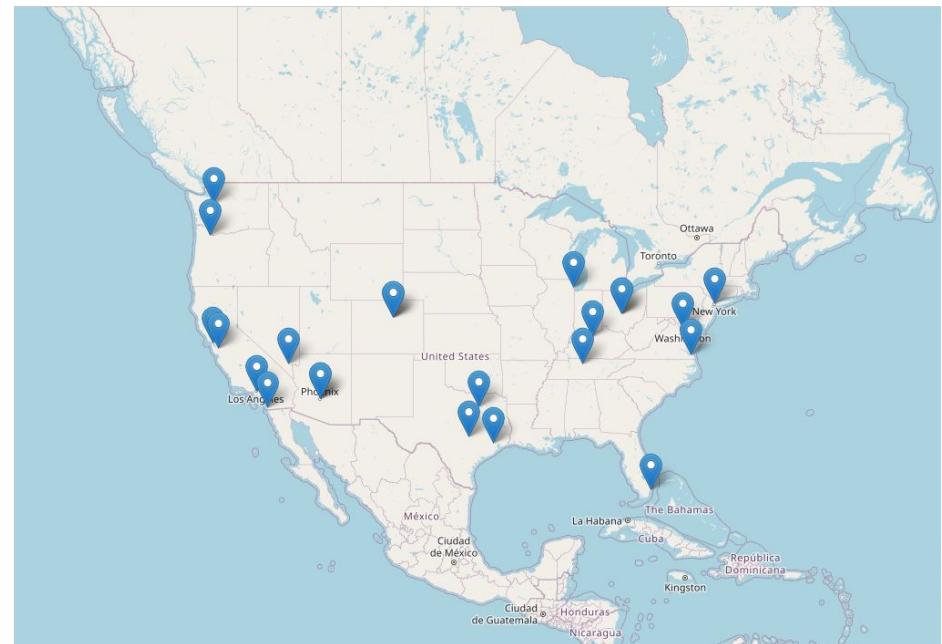
A screenshot of the Anaconda Prompt window showing the output of the "pip install folium" command. The window title is "Anaconda Prompt (Miniconda3)" and the command entered was "(base) C:\Users\user>pip install folium". The output shows various package dependencies being installed, with many "Requirement already satisfied" messages. The output ends with several "WARNING: Ignoring invalid distribution" messages for packages like "hecl", "pencv-python-headless", and "folium".

```
(base) C:\Users\user>pip install folium
WARNING: Ignoring invalid distribution - (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -harser-normalizer (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -heel (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -pency-python-headless (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -qd (c:\programdata\miniconda3\lib\site-packages)
Requirement already satisfied: folium in c:\programdata\miniconda3\lib\site-packages (0.14.0)
Requirement already satisfied: branca>=0.6.0 in c:\programdata\miniconda3\lib\site-packages (from folium) (0.6.0)
Requirement already satisfied: jinja2>=2.9 in c:\programdata\miniconda3\lib\site-packages (from folium) (3.1.3)
Requirement already satisfied: numpy in c:\programdata\miniconda3\lib\site-packages (from folium) (1.26.4)
Requirement already satisfied: requests in c:\programdata\miniconda3\lib\site-packages (from folium) (2.32.3)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\miniconda3\lib\site-packages (from jinja2>=2.9->folium) (2.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\miniconda3\lib\site-packages (from requests->folium) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\miniconda3\lib\site-packages (from requests->folium) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\miniconda3\lib\site-packages (from requests->folium) (2.2.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\miniconda3\lib\site-packages (from requests->folium) (2024.2.2)
WARNING: Ignoring invalid distribution - (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -harser-normalizer (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -heel (c:\programdata\miniconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -pency-python-headless (c:\programdata\miniconda3\lib\site-packages)
```

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
]
```



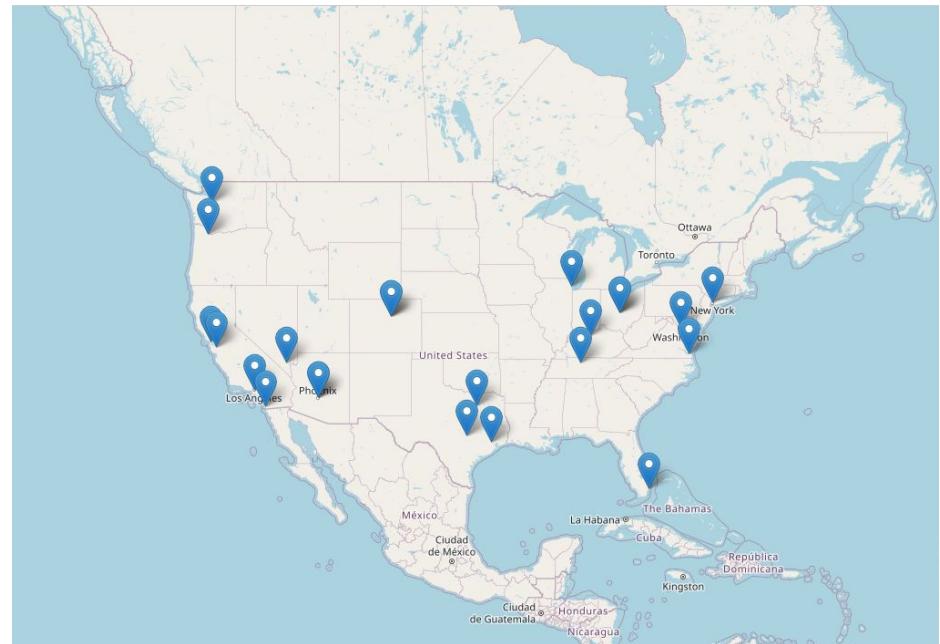
Python Google Map Tutorial

```
# 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")
```

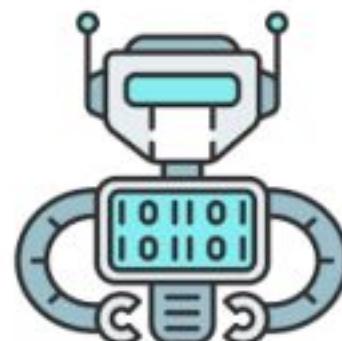
```
# If you're using Jupyter Notebook, you can  
# display the map inline  
my_map
```



Introduction to Business Process Automation

Robotic Process Automation (RPA) - Introduction

- Rule-based software
- Automating high-volume activities
- Free up human workers (for more meaningful tasks)

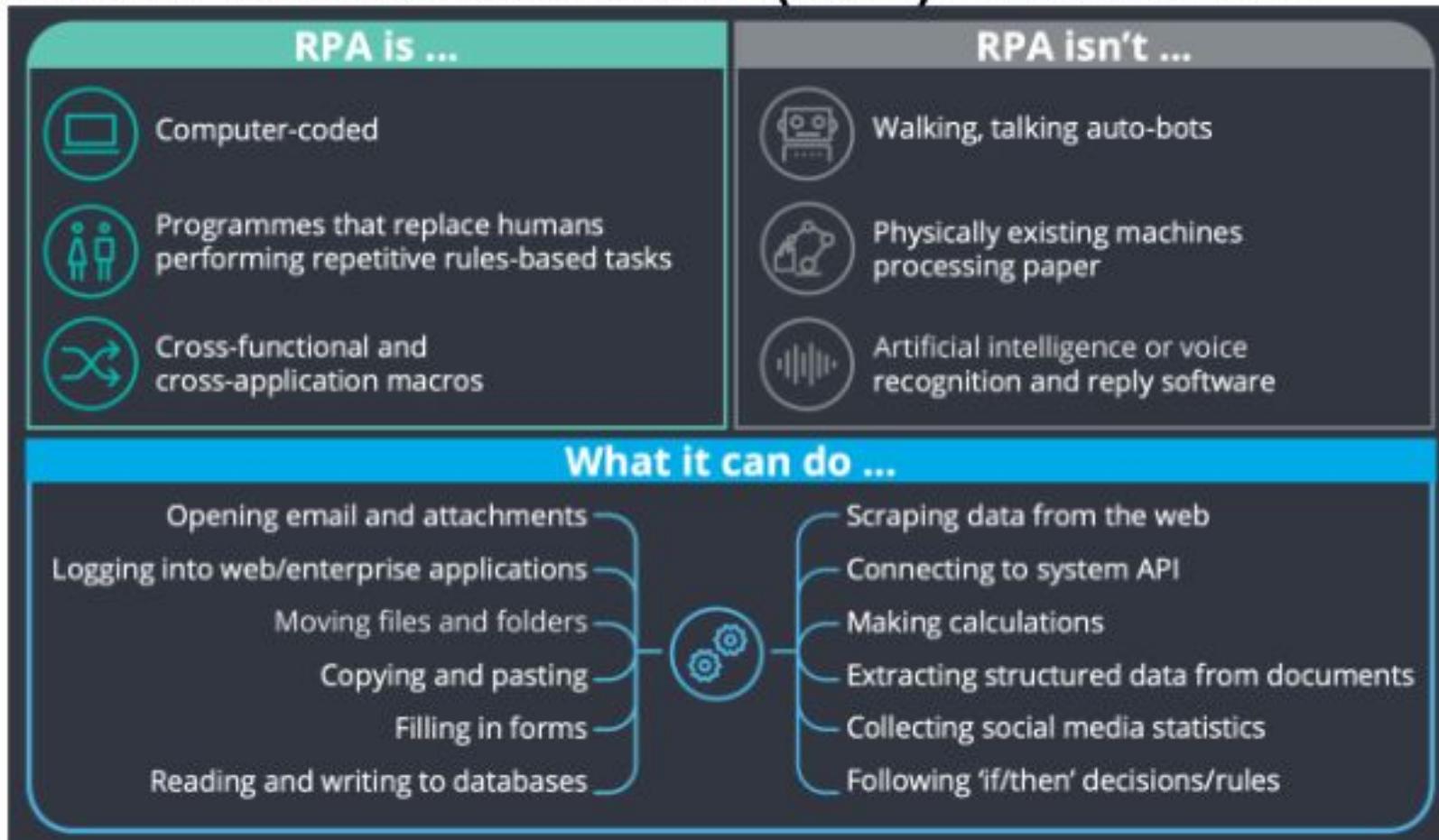


RPA bot

1. *What is Robotic Process Automation (RPA)?* | IBM. What Is Robotic Process Automation (RPA)? | IBM. <https://www.ibm.com/topics/rpa>

Introduction to Business Process Automation

Robotic Process Automation (RPA) - Introduction



1. Deloitte. (2018). *The robots are ready. Are you?*

<https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology/deloitte-robots-are-ready.pdf>

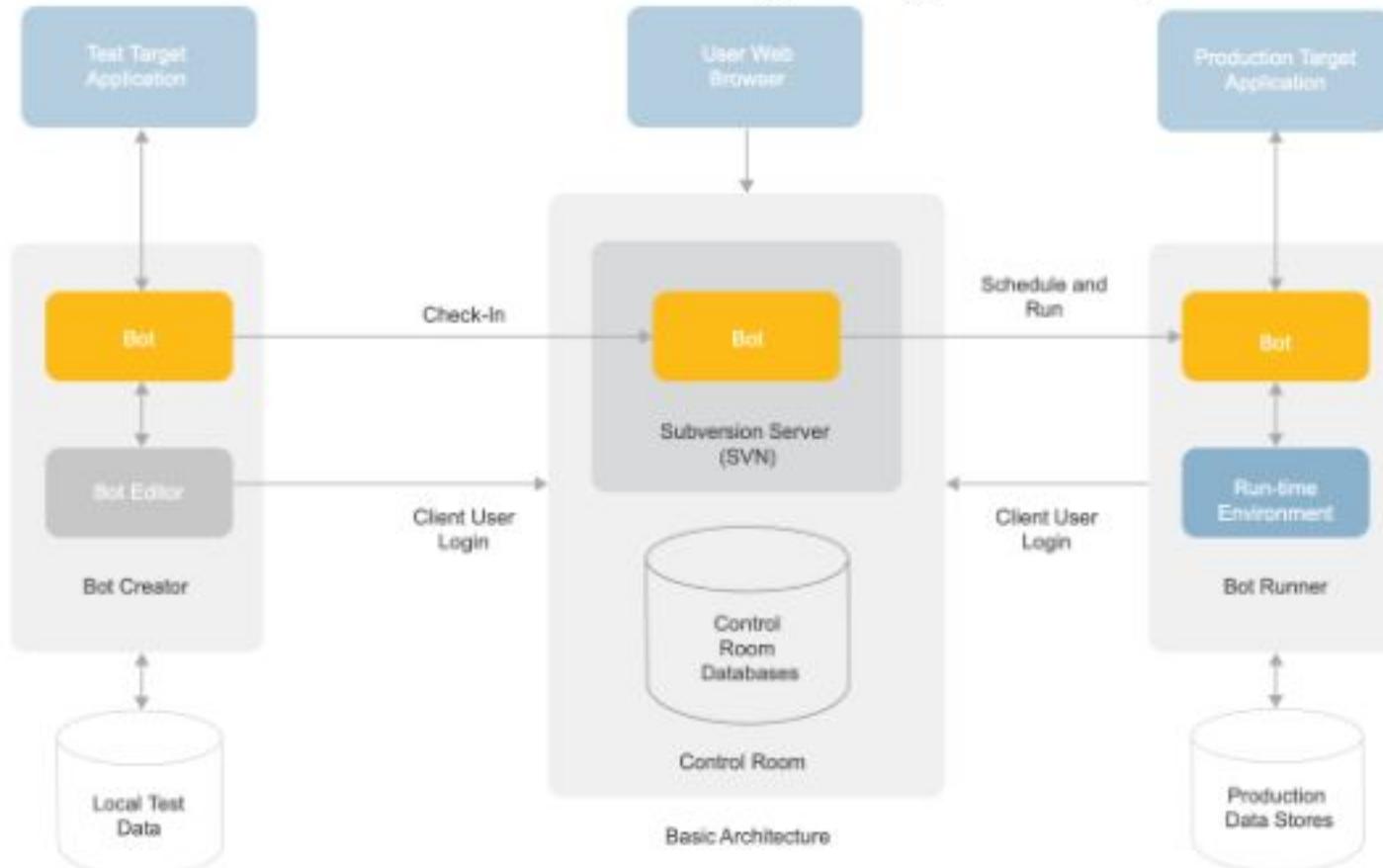
Robotic Process Automation (RPA) - Components

- **Bot modeling**
 - Bot Creator → development/ testing
 - Bot Runner → execution of tested codes in production
- **Control Room**
 - Deployment / Scheduler
 - Audit trails
 - Performance Analytics
 - Role-based access control



Introduction to Business Process Automation

Robotic Process Automation (RPA) - Components



1. Automation Anywhere. *Enterprise-class Security for Robotic Process Automation.*
https://www.automationanywhere.com/sites/default/files/internal-assets/uberflip/security-whitepaper_en.pdf

Introduction to Business Process Automation

Intelligent Process Automation (IPA)

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



Introduction to Business Process Automation

Intelligent Process Automation (IPA)

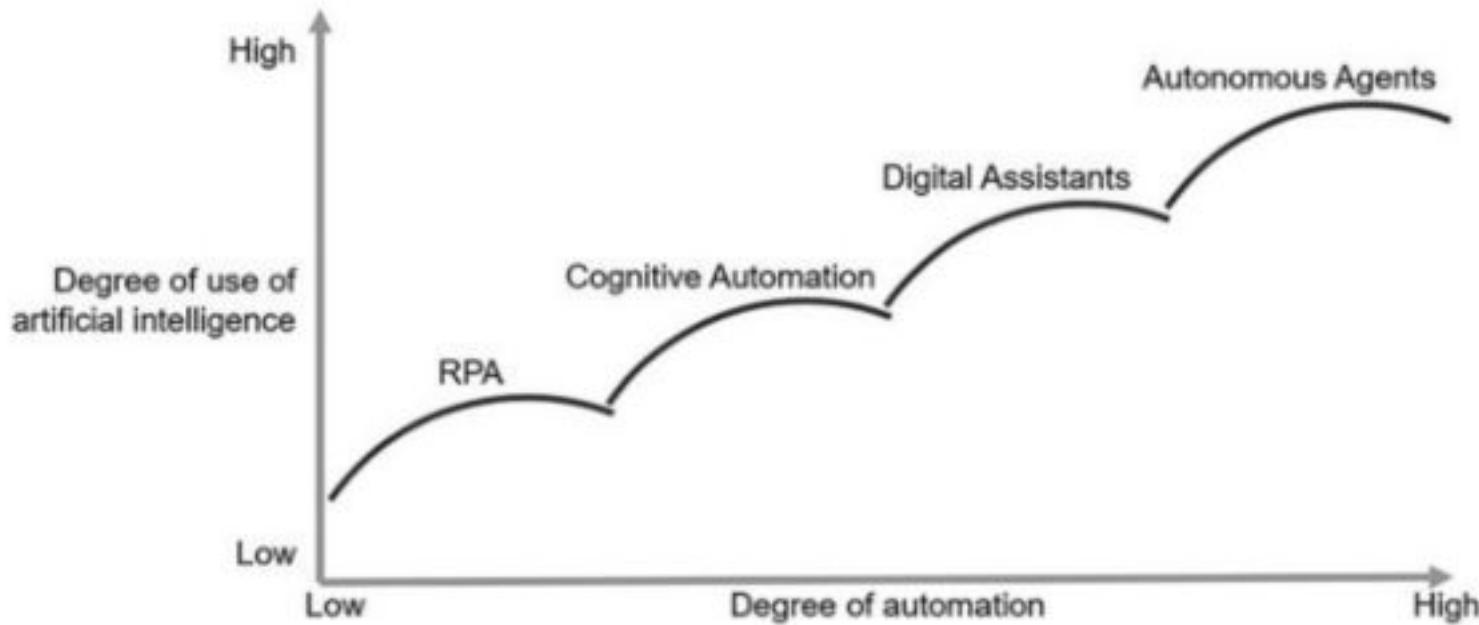
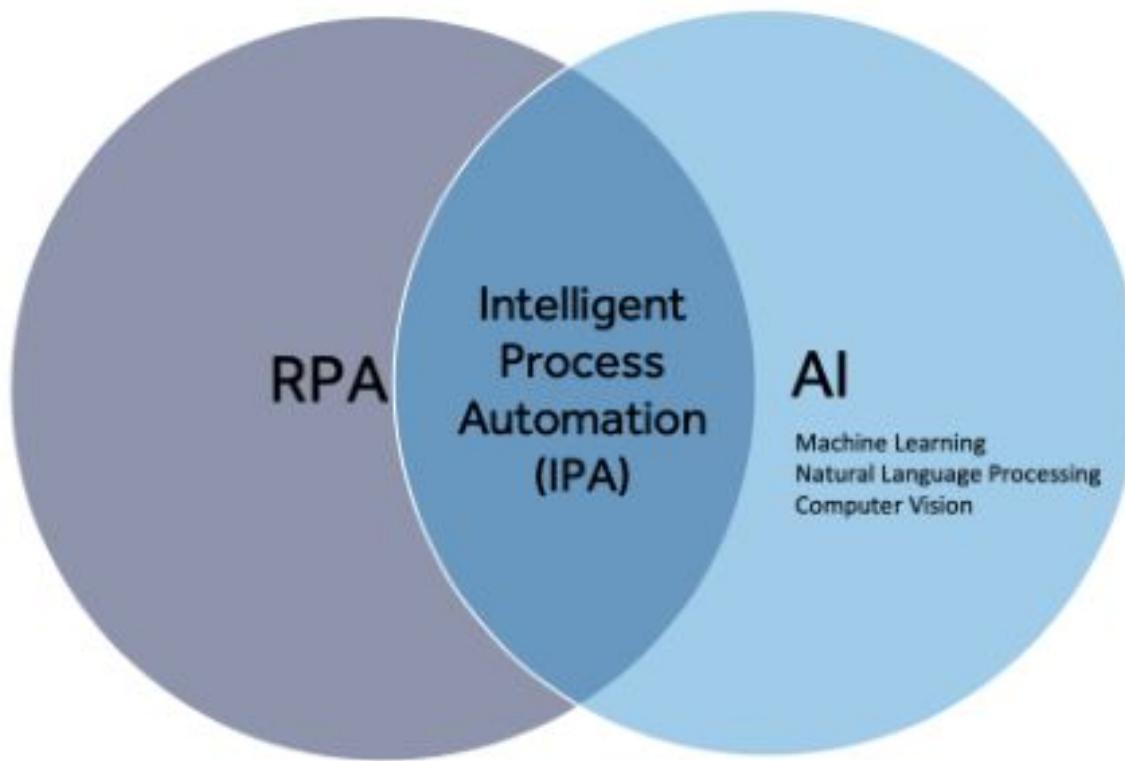


Fig. 2.2 Classification of RPA according to the degree of automation and the use of artificial intelligence (based on Ostrowicz 2018, p. 4)

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>

Introduction to Business Process Automation

Intelligent Process Automation (IPA)



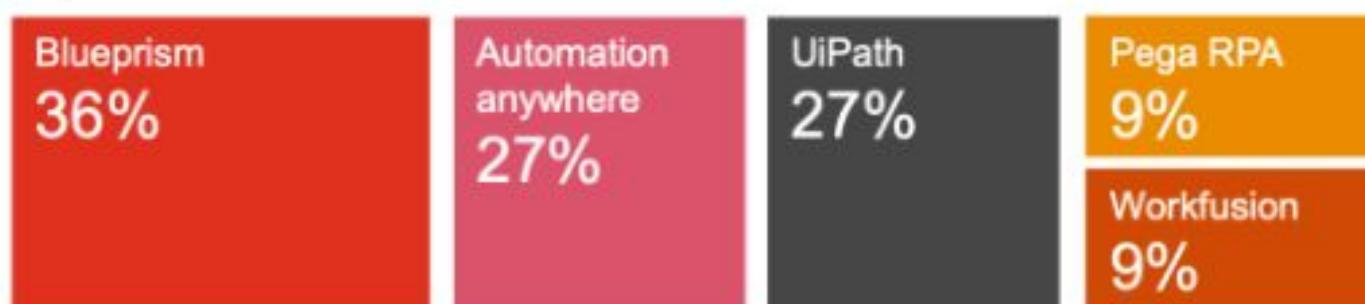
1. Kosmopoulos, C. (2021, May 11). *What is Intelligent Process Automation (IPA)?* | Blueprint. What Is Intelligent Process Automation (IPA)? | Blueprint. <https://www.blueprintsys.com/blog/rpa/what-is-intelligent-process-automation-ipa>

Introduction to Business Process Automation

Tools for Automation - Overview

Which tool is being used ?

As some companies use more than one tool, the percentages in this graphic total to more than 100%

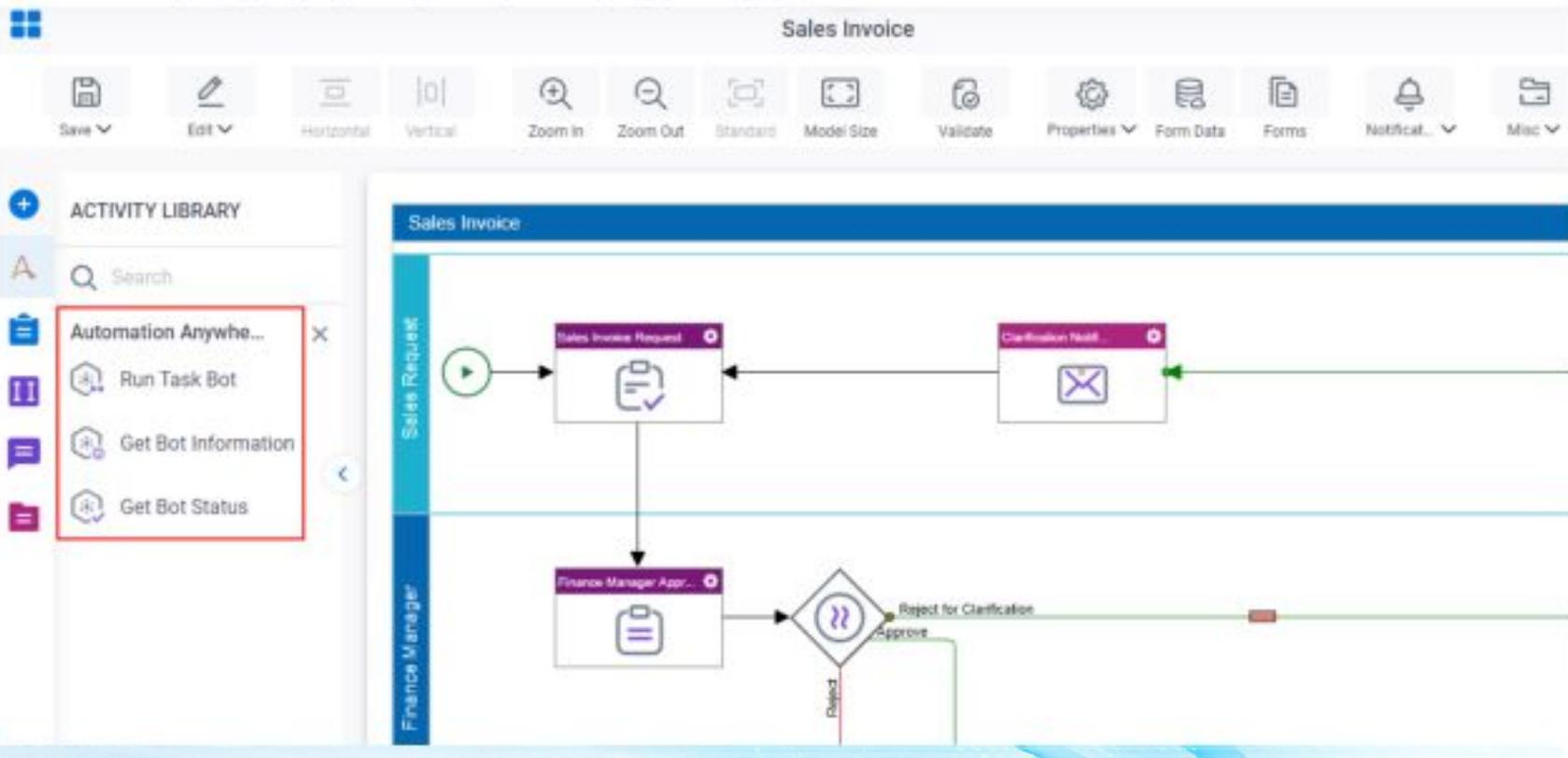


1. 2019 actuarial robotic process automation (RPA) survey report. PwC.
<https://www.pwc.com/gx/en/industries/financial-services/publications/pwc-2019-actuarial-robotic-process-automation-survey-report.html>

Introduction to Business Process Automation

Tools for Automation - Types

- **Low-code platforms**
 - Automation Anywhere, XCEPTOR, Microsoft Power Automate



Introduction to Business Process Automation

Tools for Automation - Types

- In-house Solutions

- Excel VBA, Java, Python, Web

Microsoft®
Excel VBA



Most in-demand functional areas

- Compliance
- Operational Risk
- Fund Accounting
- Corporate Accounting
- Actuarial
- Finance Operations
- IT Audit

In-demand technical skills and experience

- IFRS 17 Reporting and Accounting Policy
- RBC and Capital Reporting
- IT skills (Python, C+ BASIC, C, C++, COBOL, Java, R, Tableau, SQL, VBA, and Macro)
- Operational Risk, Information and IT Risk, Risk Transformation, Risk Analytics, Internal Control Framework



1. Robert Half. 2024 Salary Guide HK.
<https://content.roberthalfonline.com/SG24/SG24-PDF/2024-Salary-Guide-HK.pdf>

Introduction to Business Process Automation

Tools for Automation – Getting a taste

- **Visualping** (<https://visualping.io/>)
 - Websites monitoring



<https://hkuspace.hku.hk/prog/cert-for-module-business-process-automation-with-vba>

Certificate for Module (Business Process Automation with VBA ...)

(2) Mr Jackie Liu. Mr. Jackie Liu is veteran quantitative strategist currently working in one of the top global investment banks. With over 10 years of ...

hku.hk
<https://hkuspace.hku.hk/prog/cert-for-module-business-process-automation-with-vba>

Certificate for Module (Business Process Automation with VBA ...)

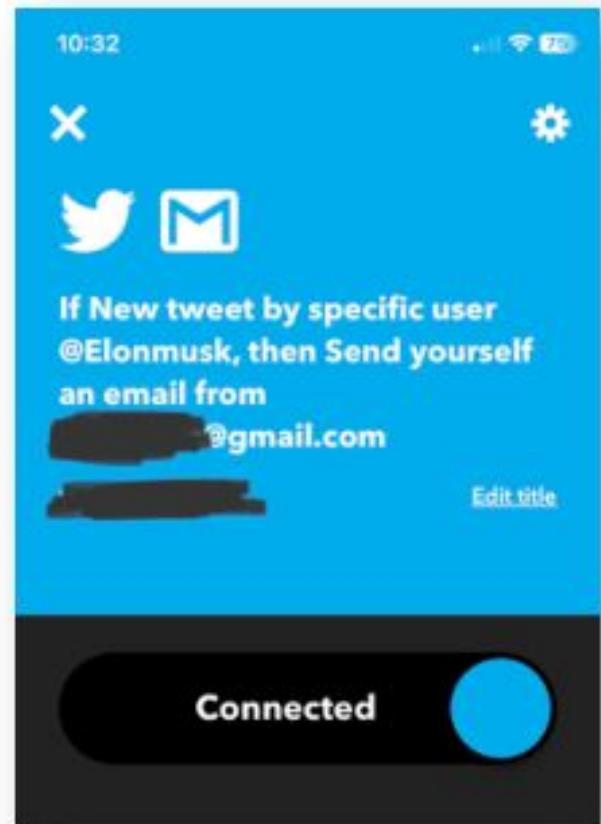
(2) Mr Jackie Liu. Mr. Jackie Liu is veteran quantitative strategist currently working in one of the top global investment banks. With over 10 years of ...

Introduction to Business Process Automation

Tools for Automation – Getting a taste

- **IFTTT (<https://ifttt.com/>)**

- Web + App interfaces
- Service-based triggers and actions



Introduction to Business Process Automation

Tools for Automation – Considerations

- Cost

- One-time cost

- Initial effort of adaptation
 - Implementation and rollout

- Running cost

- License, e.g., Power Automate: \$Millions/year¹
 - Infrastructure, e.g., Cloud storage
 - Training, e.g., Automation Anywhere, 10+ hours²
 - Support, e.g., bot re-configuration

Cost		
Production	Gold	Maint.
60 🎁	240 🎁	1 🎁

1. RPA Pricing: Comparison of Leading RPA Vendors' Fees in 2023. <https://research.aimultiple.com/rpa-pricing/>

2. RPA Course Learning Trails | Automation Anywhere University. <https://university.automationanywhere.com/training/rpa-learning-trails/>

Tools for Automation – Considerations

- **Flexibility**
 - Specialized vs Generalized, e.g., XCEPTOR → Tax
- **Scalability**
 - Desktop vs Enterprise, e.g., Excel VBA
 - Version Control, e.g., Use of repository
- **Time to delivery**
 - “Perfect” solution does not exist



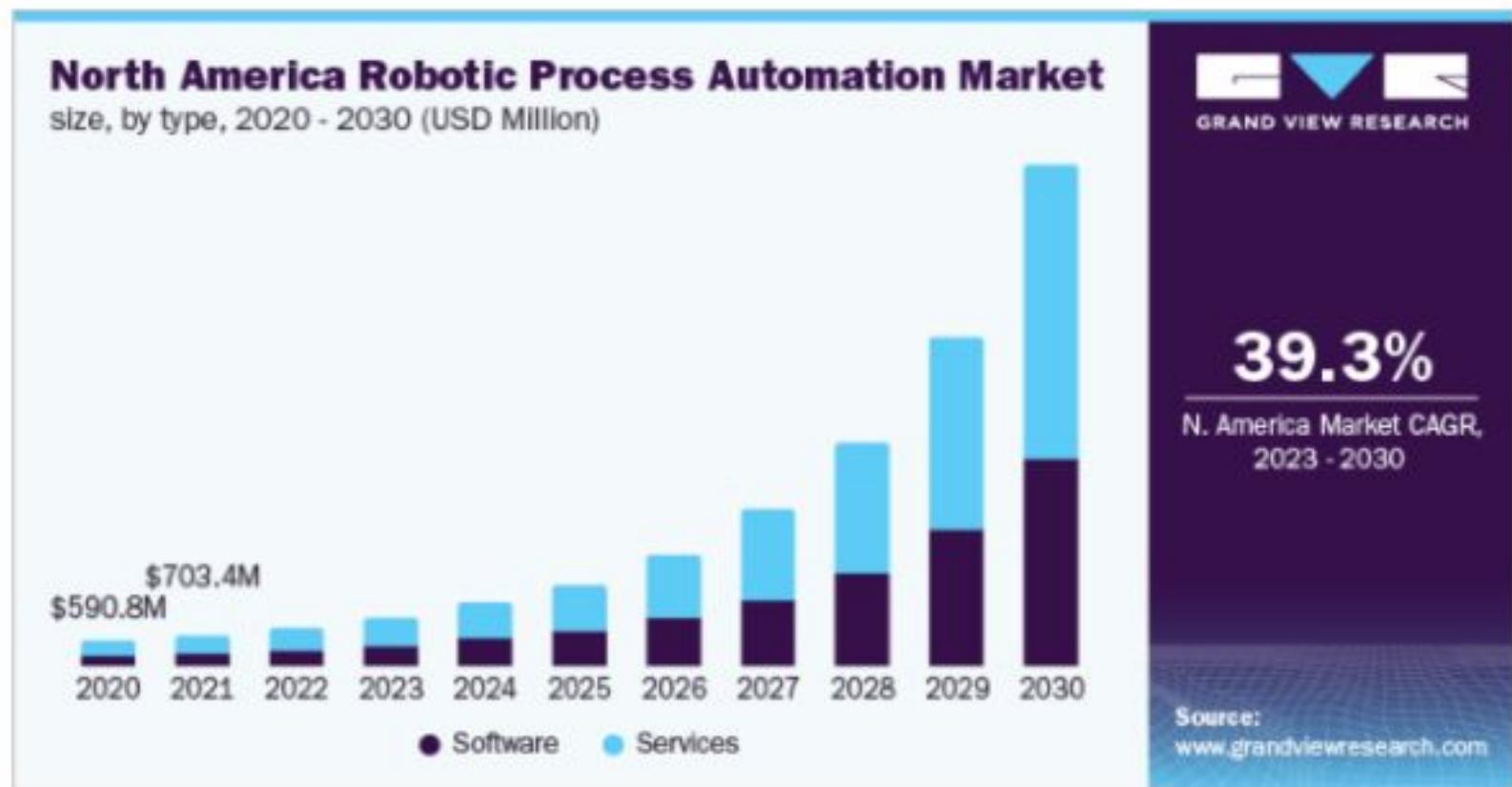
GitHub



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>

Introduction to Business Process Automation

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>

Introduction to Business Process Automation

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



Introduction to Business Process Automation

Opportunities of BPA – Regtech

- Follow the money

 CNN

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

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



Introduction to Business Process Automation

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>

Introduction to Business Process Automation

Opportunities of BPA – Regtech

- Common themes



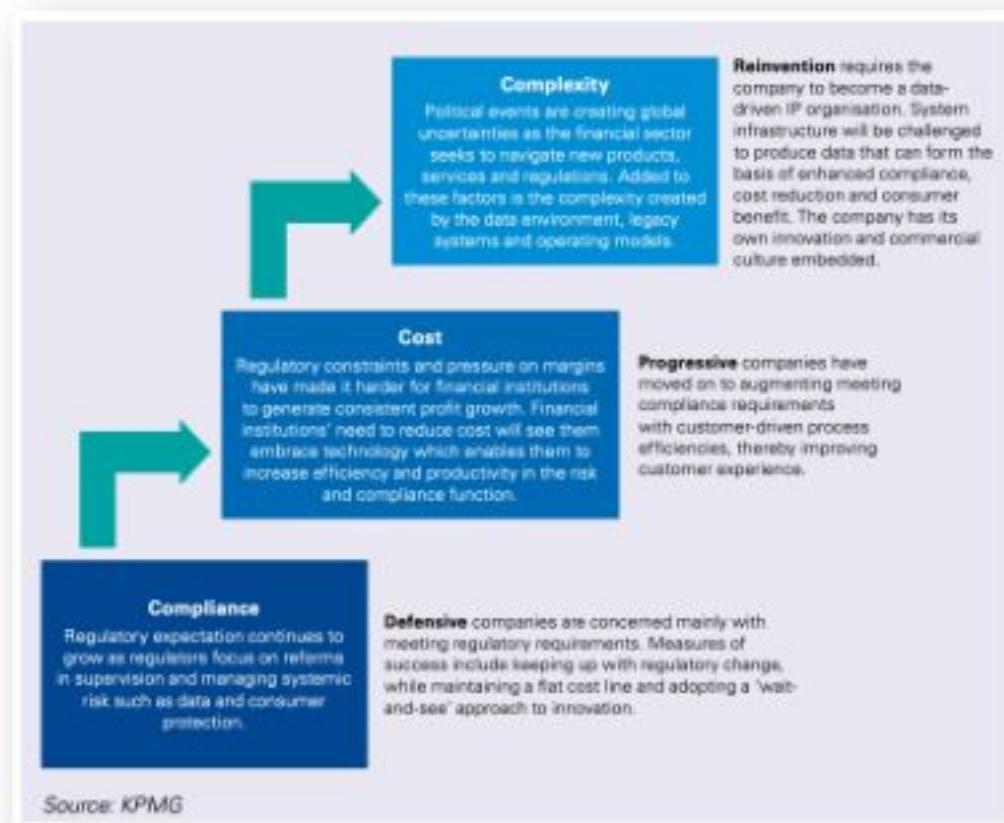
1. KPMG. (2022, November). <https://assets.kpmg.com/content/dam/kpmg/uk/pdf/2022/11/innovate-finance-regtech-industry-and-adoption.pdf>

Introduction to Business Process Automation

Opportunities of BPA – Regtech

Adoption phases:

1. Defensive
2. Progressive
3. Reinvention

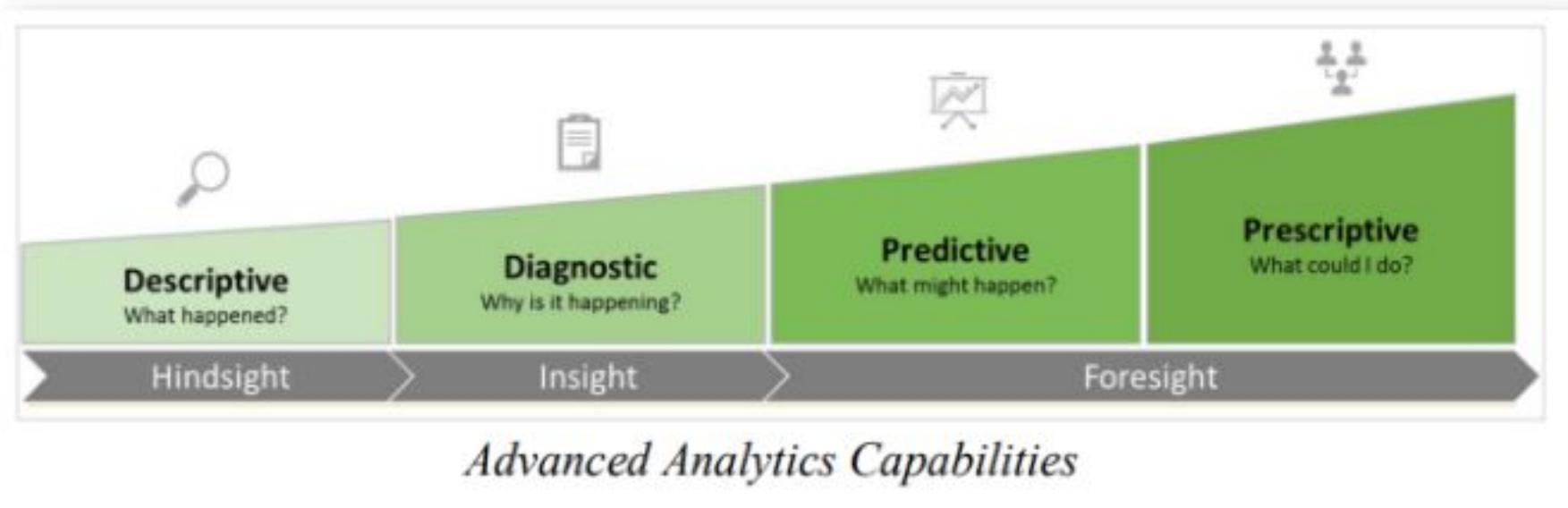


1. KPMG. (2019, June). *There's a revolution coming.* <https://assets.kpmg.com/content/dam/kpmg/cn/pdf/en/2019/06/embracing-the-challenge-of-the-new-regtech-era.pdf>

Introduction to Business Process Automation

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>

Introduction to Business Process Automation

Challenges of BPA – Insight from the banks

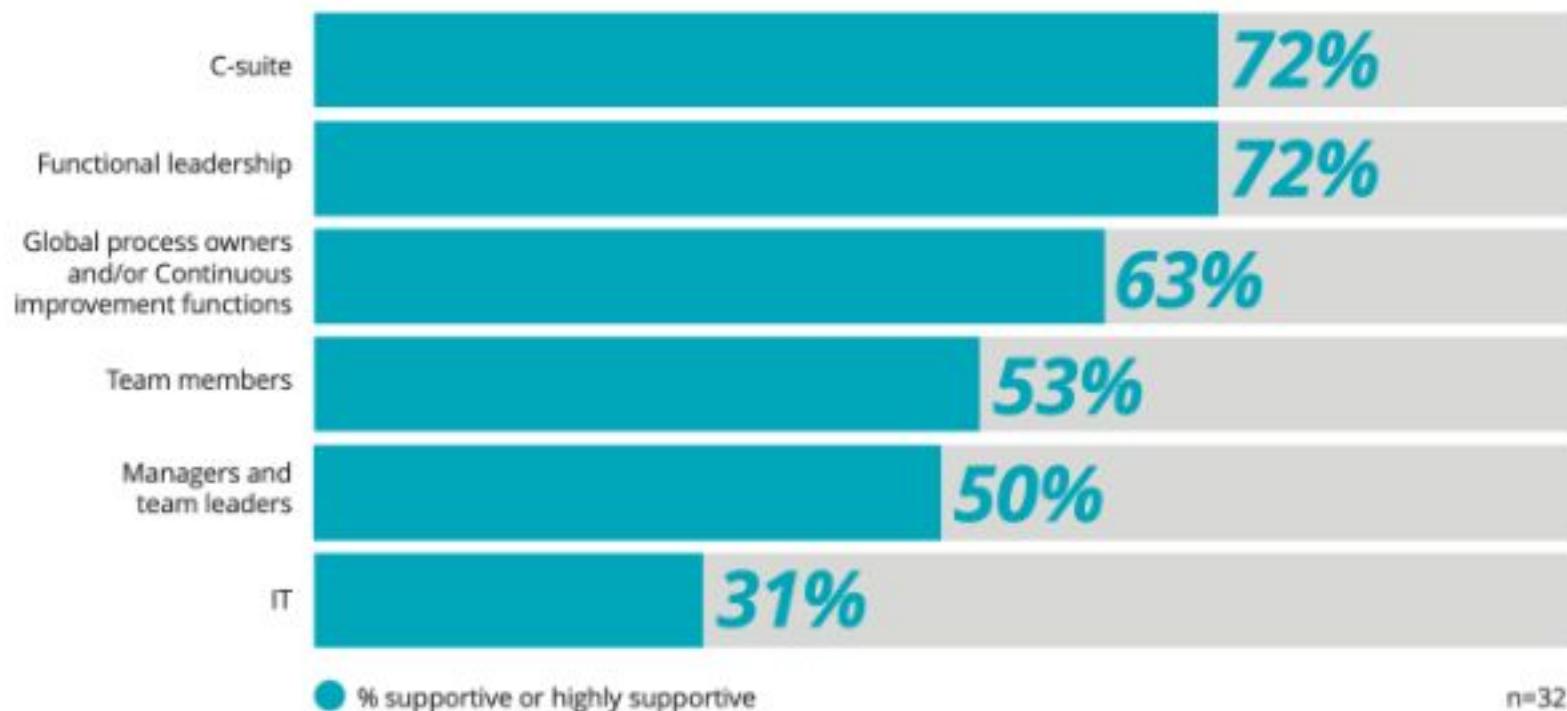


1. KPMG & HKMA. (2020). *Transforming Risk Management and Compliance: Harnessing the Power of Regtech.* <https://www.hkma.gov.hk/media/chi/doc/key-information/press-release/2020/20201102c3a1.pdf>

Introduction to Business Process Automation

Challenges of BPA – Worries from the workers?

Figure 5: How supportive of the RPA implementation were your stakeholder groups?

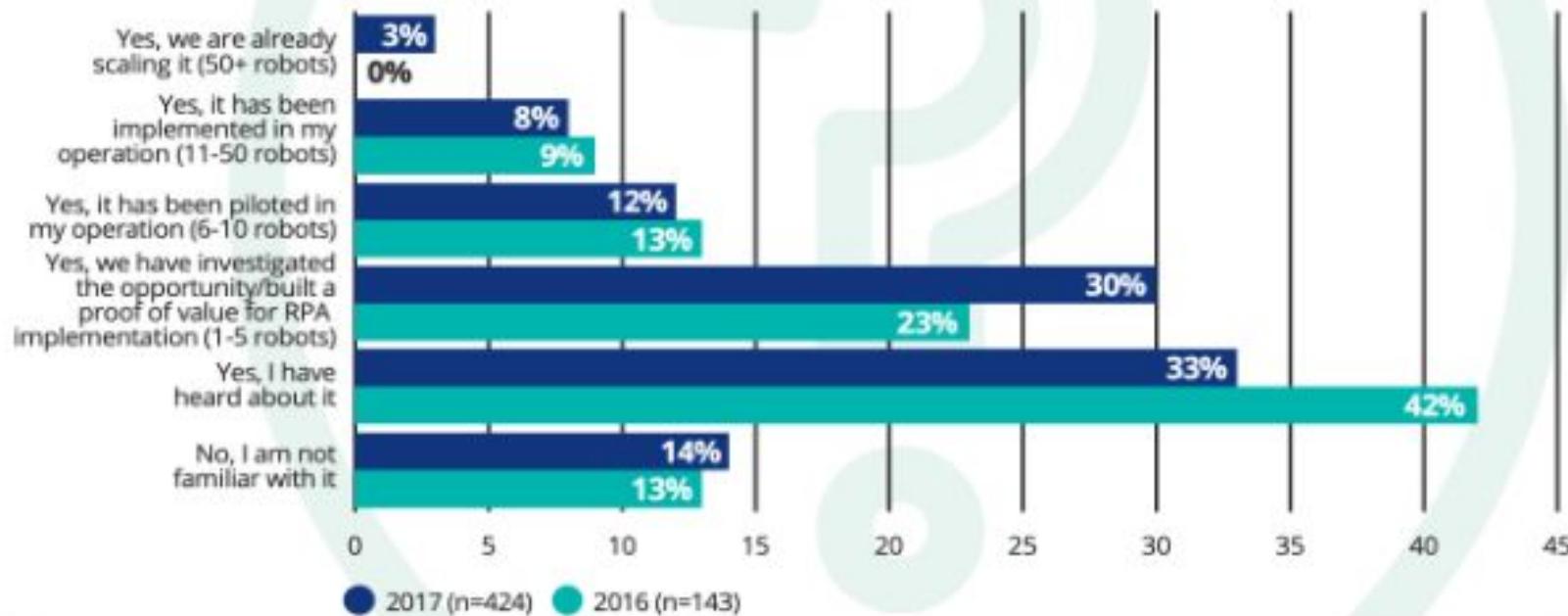


1. Deloitte. (2018). *The robots are ready. Are you?*
<https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology/deloitte-robots-are-ready.pdf>

Introduction to Business Process Automation

Challenges of BPA – Hesitant to scale up

Figure 2: Are you familiar with RPA?



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

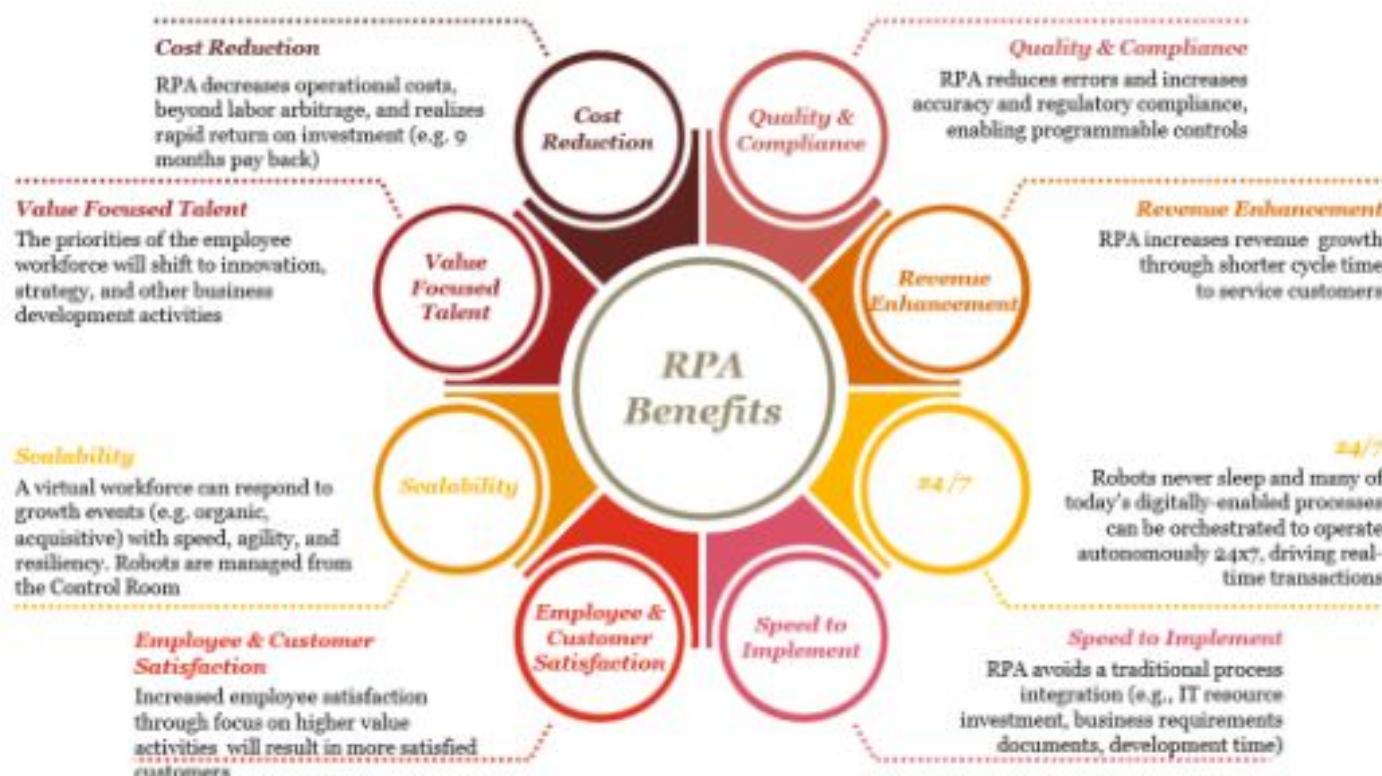


* Success depends on planning
(More on project management later)

Introduction to Business Process Automation

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>

Introduction to Business Process Automation

Implications of BPA – The Bad

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

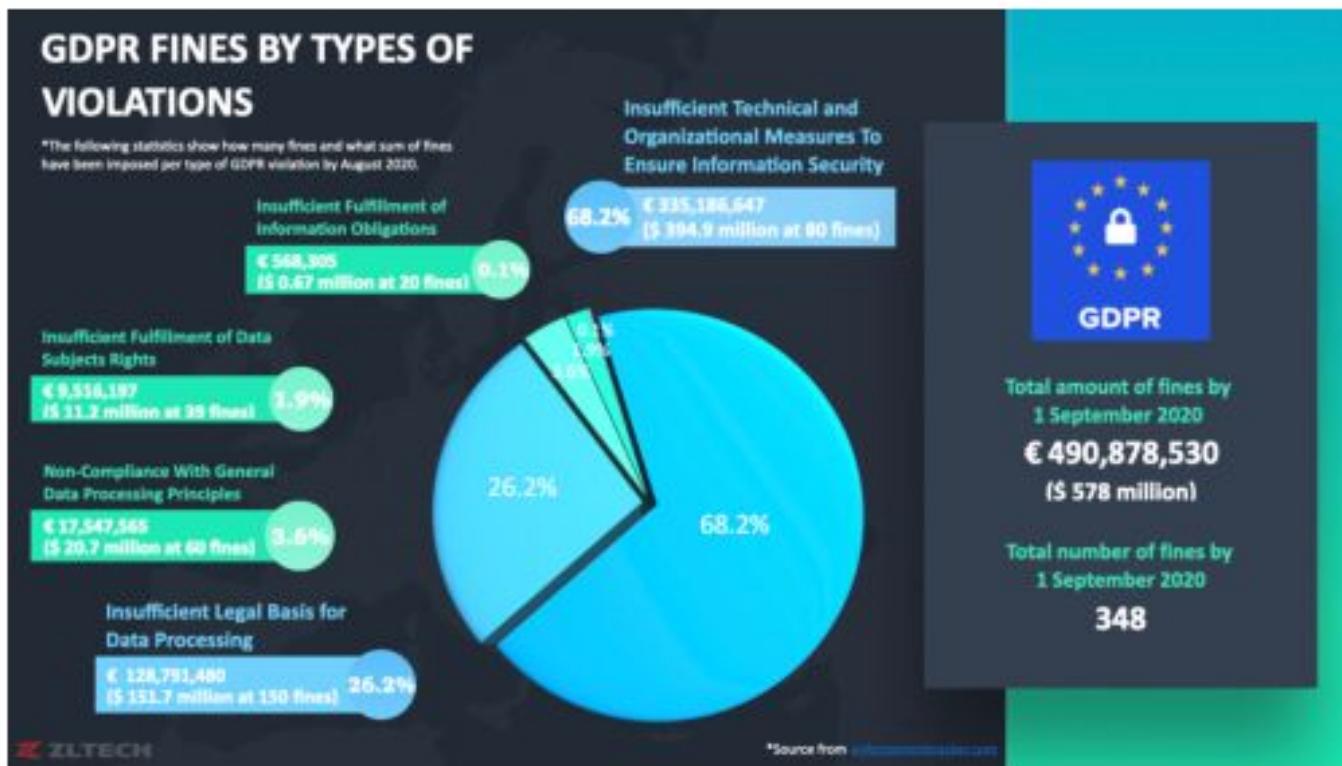


Image: Civil Aviation Safety Authority

Introduction to Business Process Automation

Implications of BPA – The Ugly

- Data Collection and Analysis¹
- Vulnerability to Cyberattacks

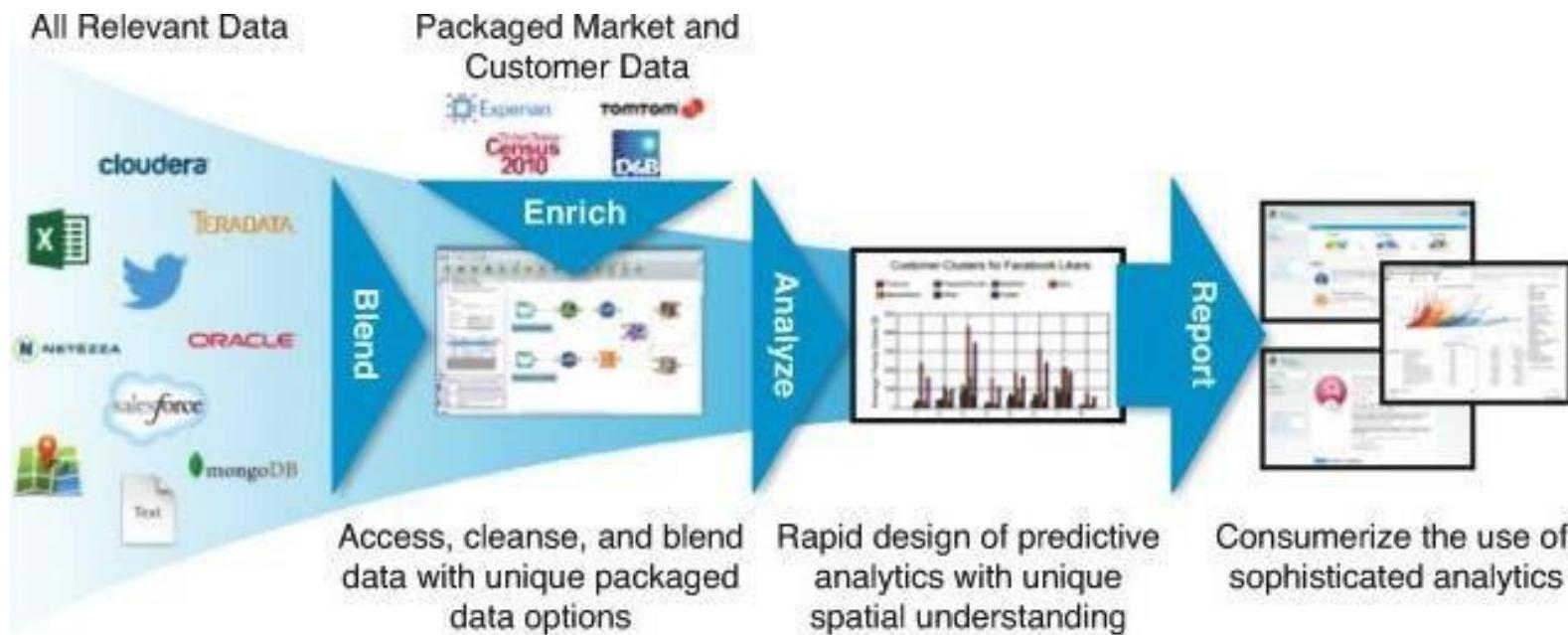


1. Chen, B. (2020, September 9). GDPR Recap: 28 Nations, 348 Fines, Half a Billion Euros. ZL Tech.
<https://www.zlti.com/blog/gdpr-recap-28-nations-348-fines-half-a-billion-euros/>

Modern BI Process - Blend, Enrich, Analyze, and Report

Predictive Analytics: By applying machine learning algorithms to historical project data, organizations can forecast outcomes more accurately and mitigate risks in project delivery

Business Intelligence: Automating the preparation of data for analytics tools allows organizations to gain actionable insights more rapidly, enhancing their decision-making capabilities.

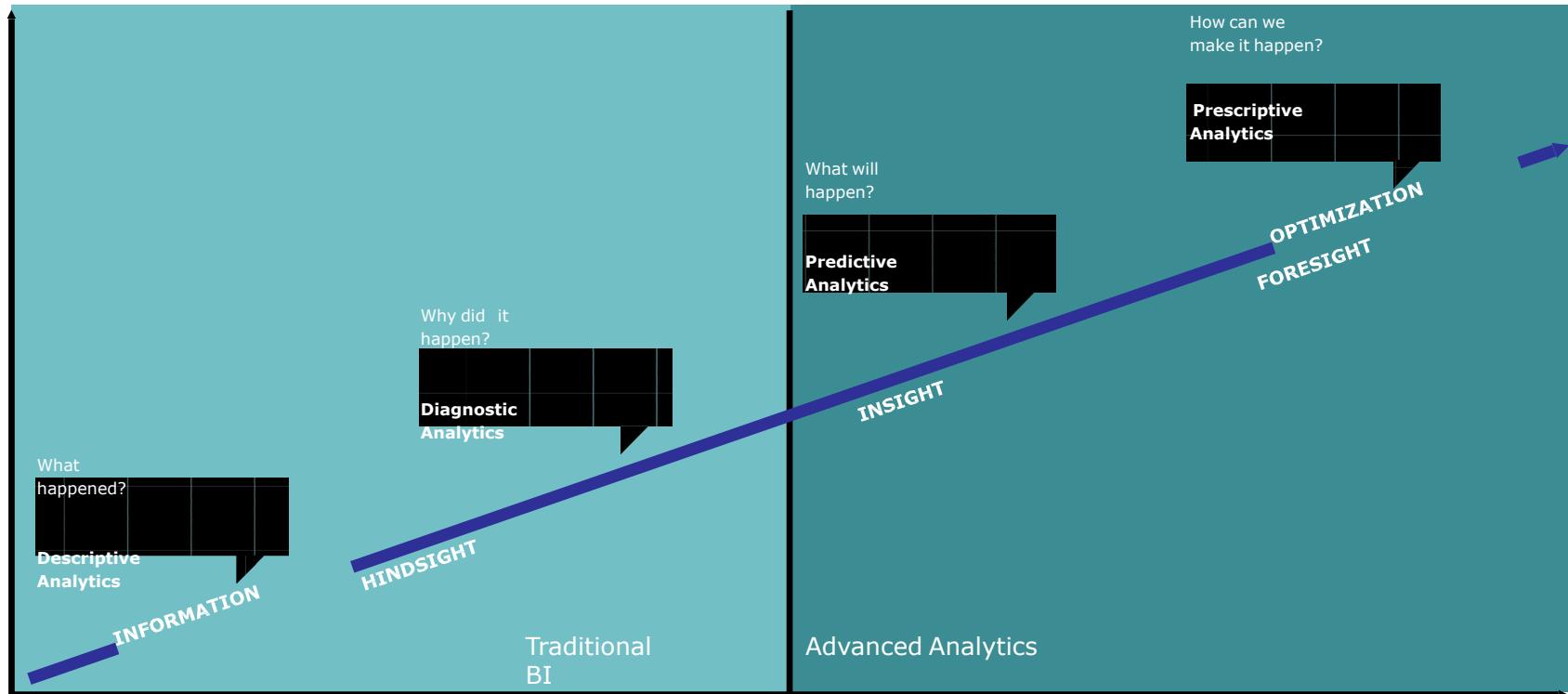


What Is Data Science?

From Wikipedia:

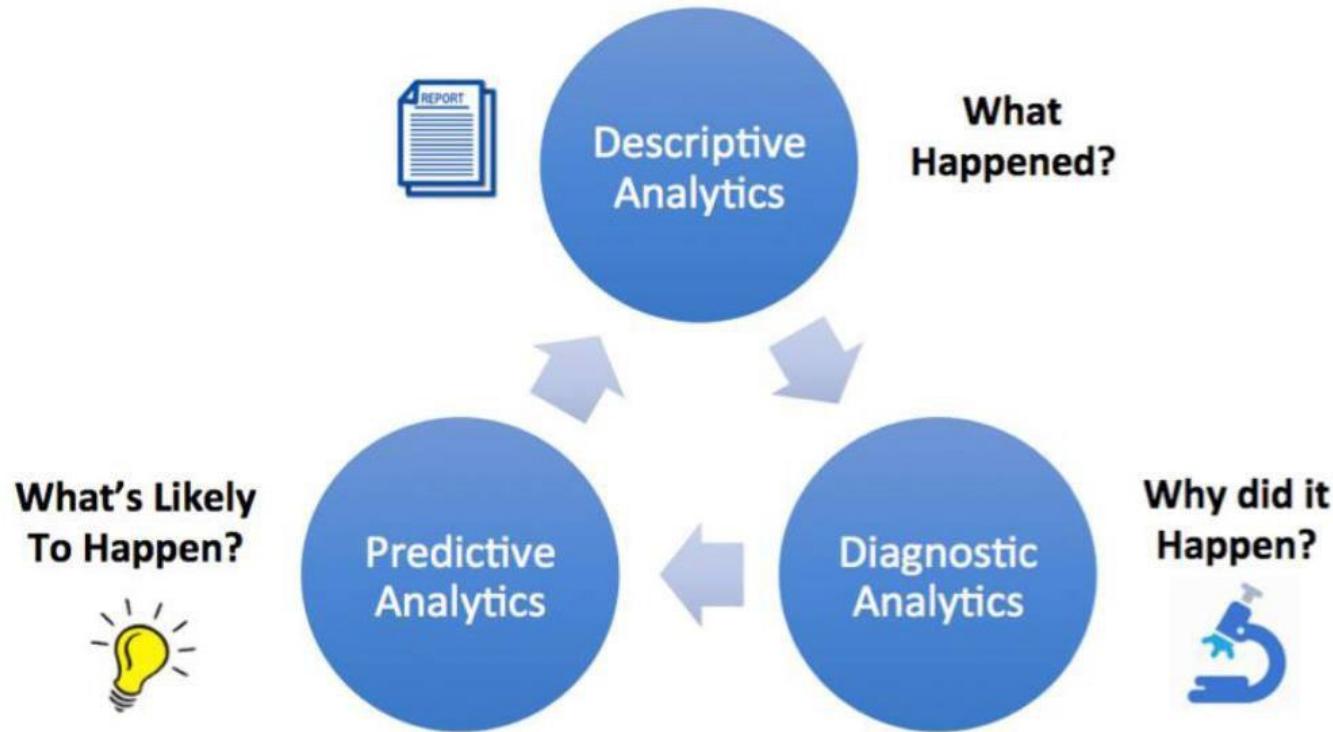
Data science, also known as data-driven science, is an interdisciplinary field about scientific processes and systems to extract knowledge or insights from data in various forms, either structured or unstructured, which is a continuation of some of the data analysis fields such as statistics, machine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

How Is Business Intelligence Different From Data Science?



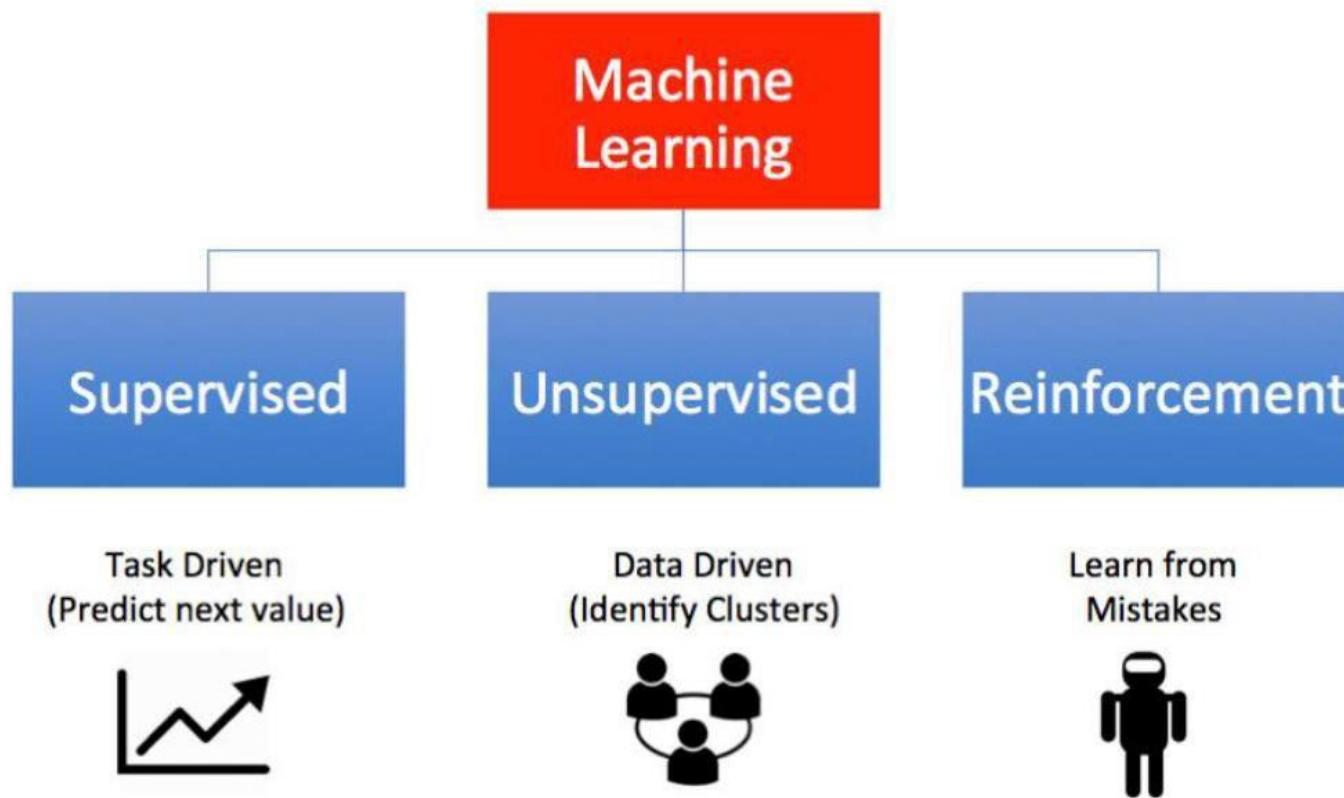
What is Business Analytics & Machine Learning?

Types of Analytics

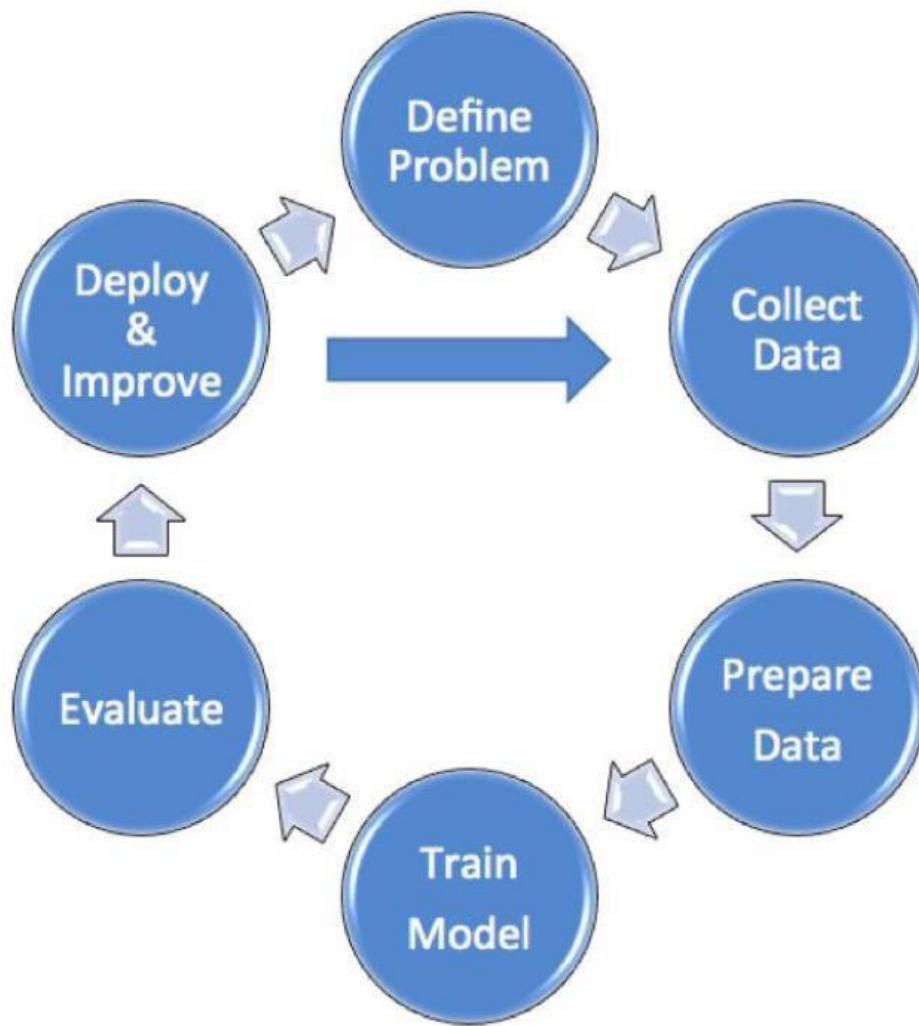


What is Business Analytics & Machine Learning?

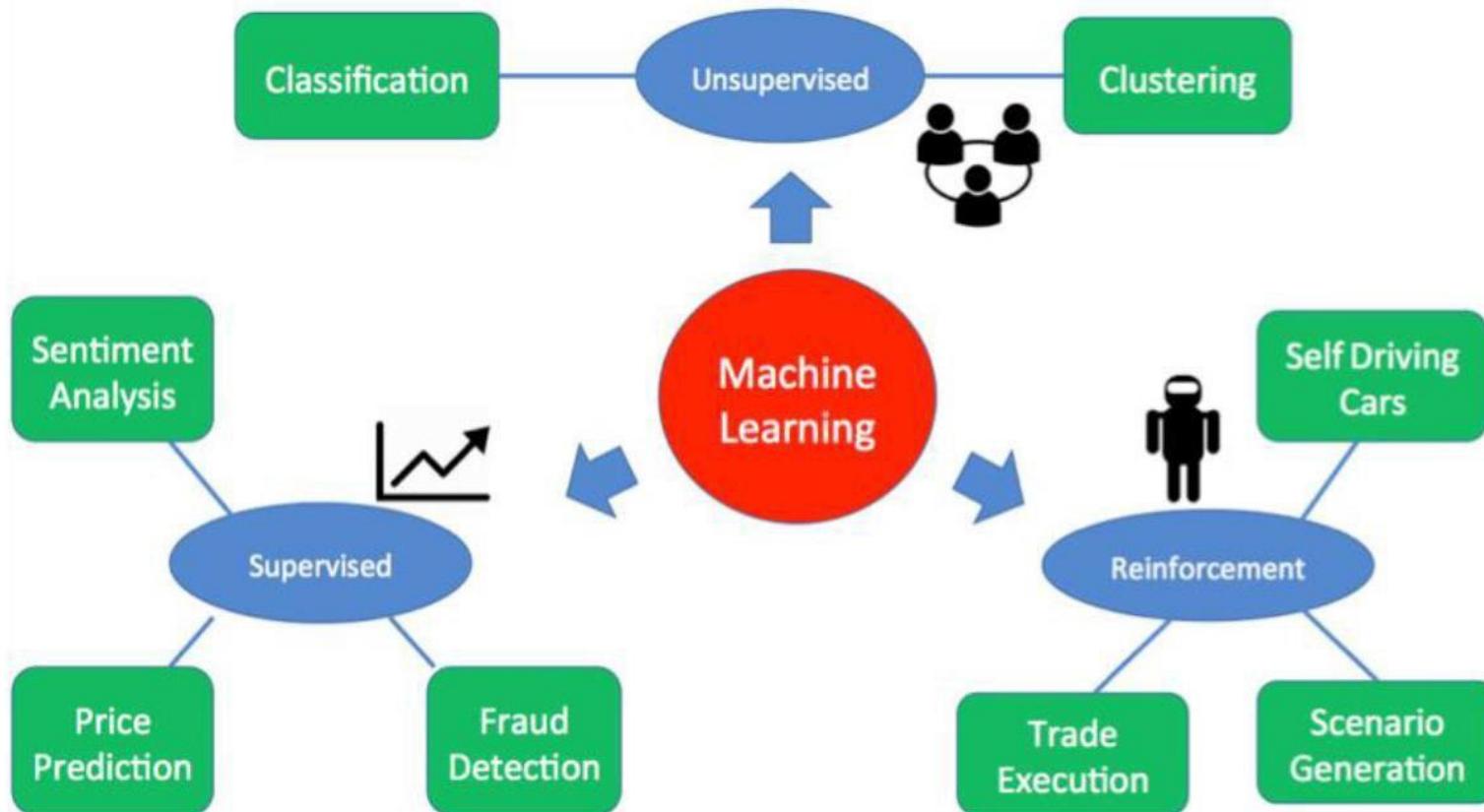
Types of Machine Learning



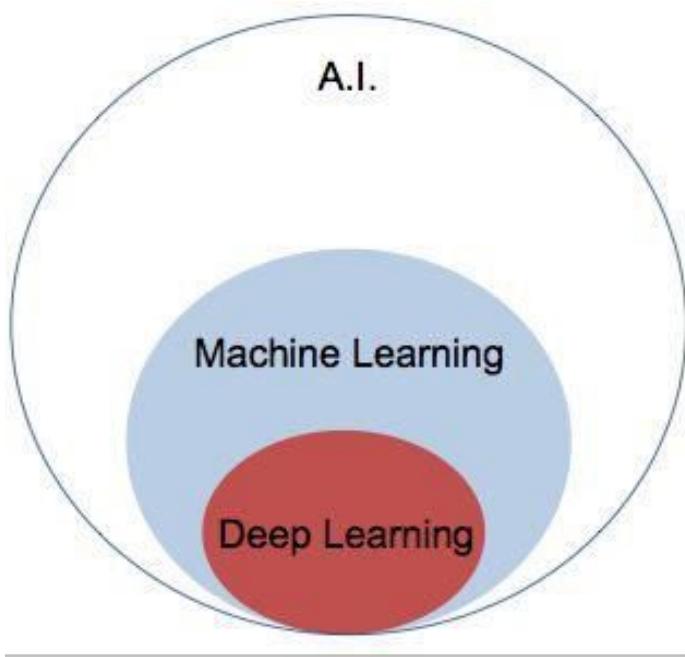
What is Business Analytics & Machine Learning?



What is Business Analytics & Machine Learning?



Clearing The Confusion About Data Science And Big Data



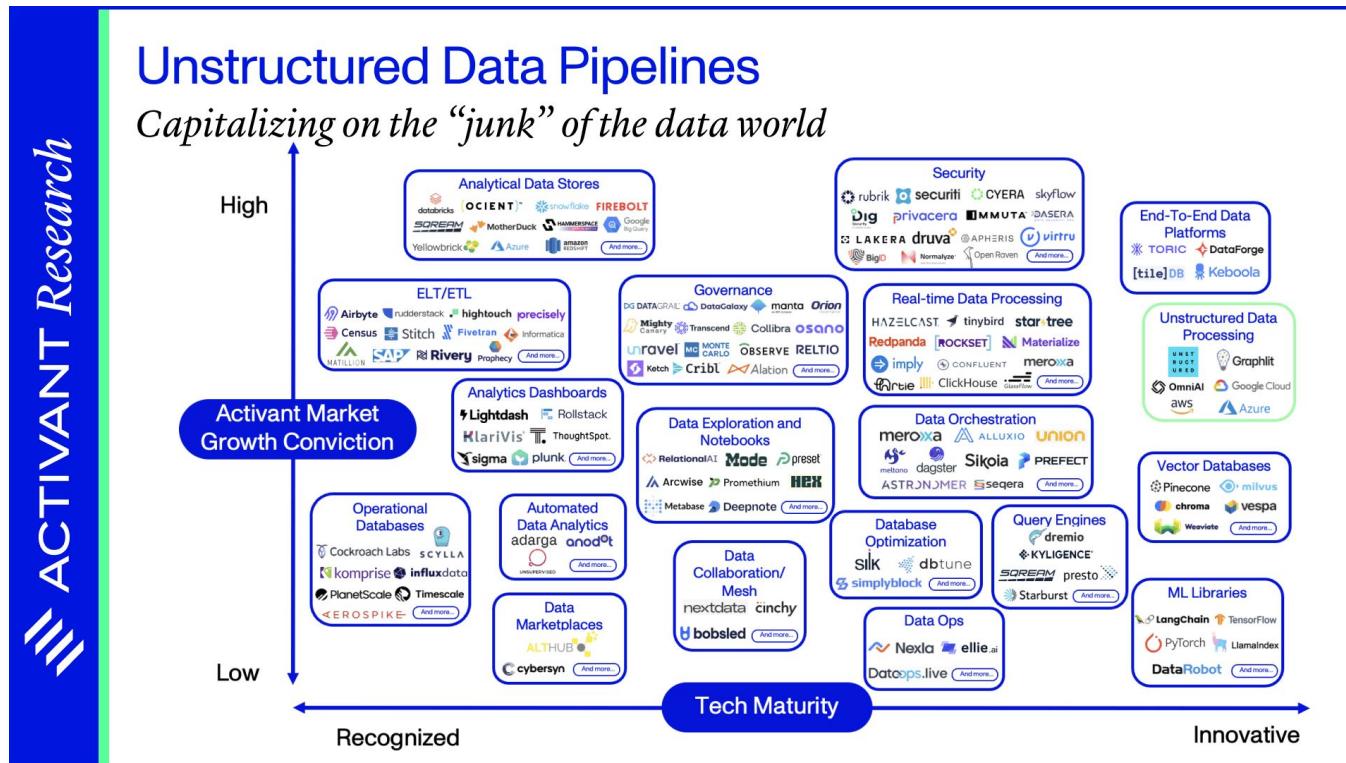
Definition of AI: a branch of computer science trying to get computers to exhibit intelligence.

- Machine Learning is a range of techniques for computers to perform cognitive functions
- Deep Learning (a type of Machine Learning that uses layers of neural networks) has been the most accurate and productive technique in AI research

Frank Chen of a 16z

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.



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
```

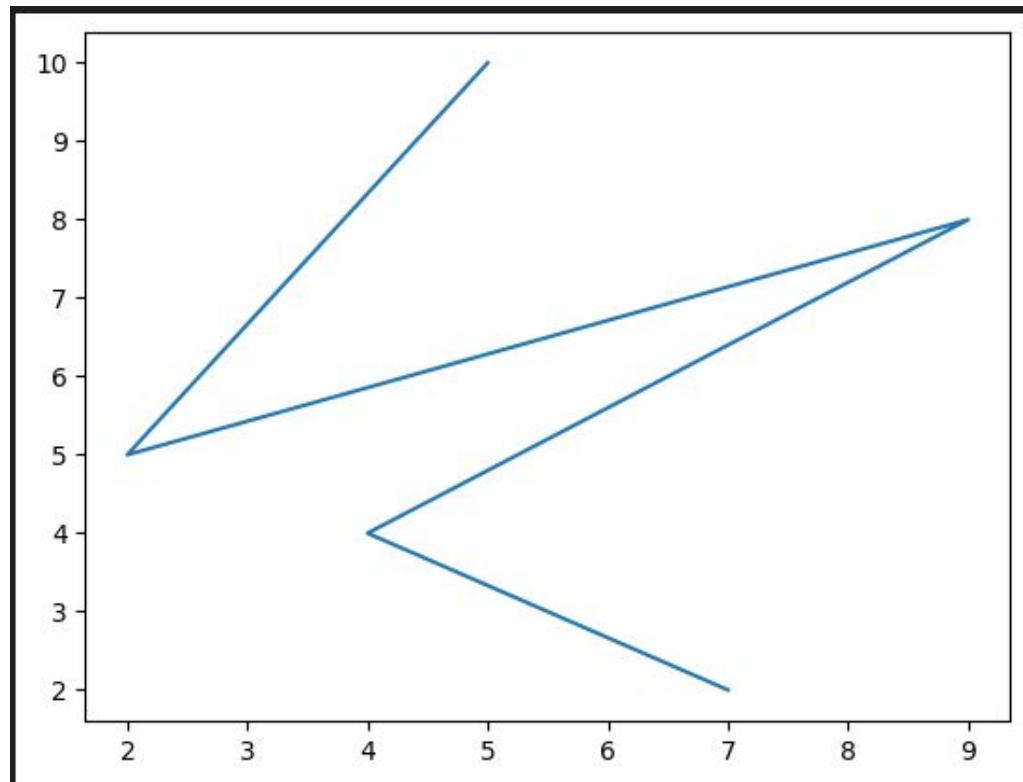
```
y = [10, 5, 8, 4, 2]
```

```
# Function to plot
```

```
plt.plot(x, y)
```

```
# function to show the plot
```

```
plt.show()
```



Exercise 1

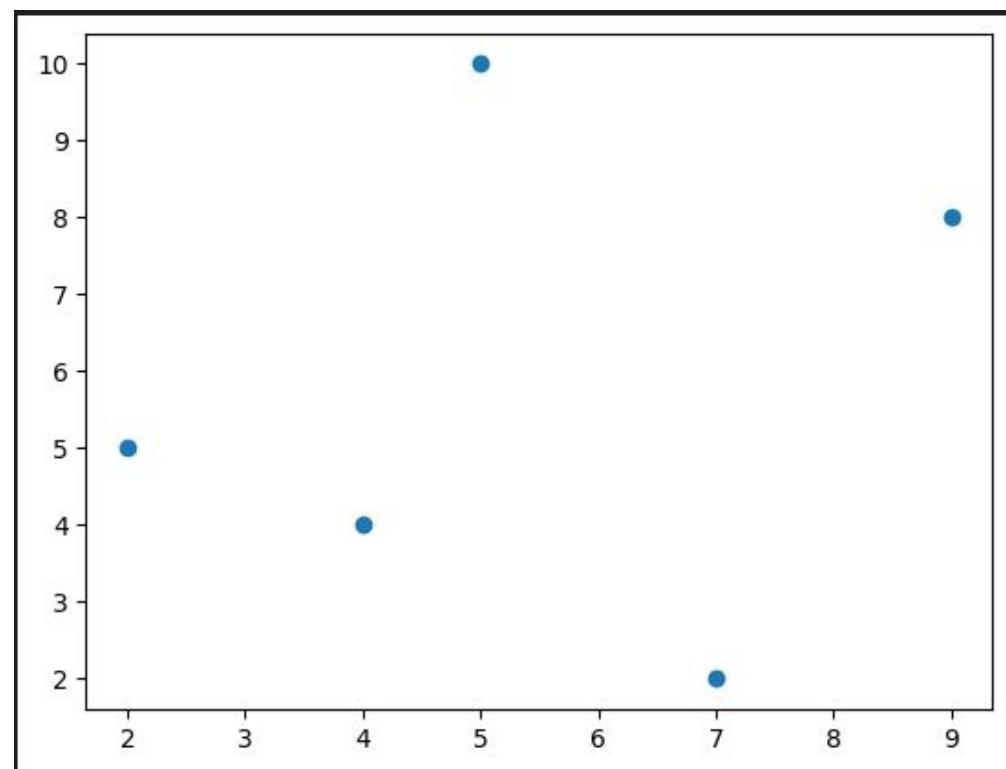
Step 3

```
#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

