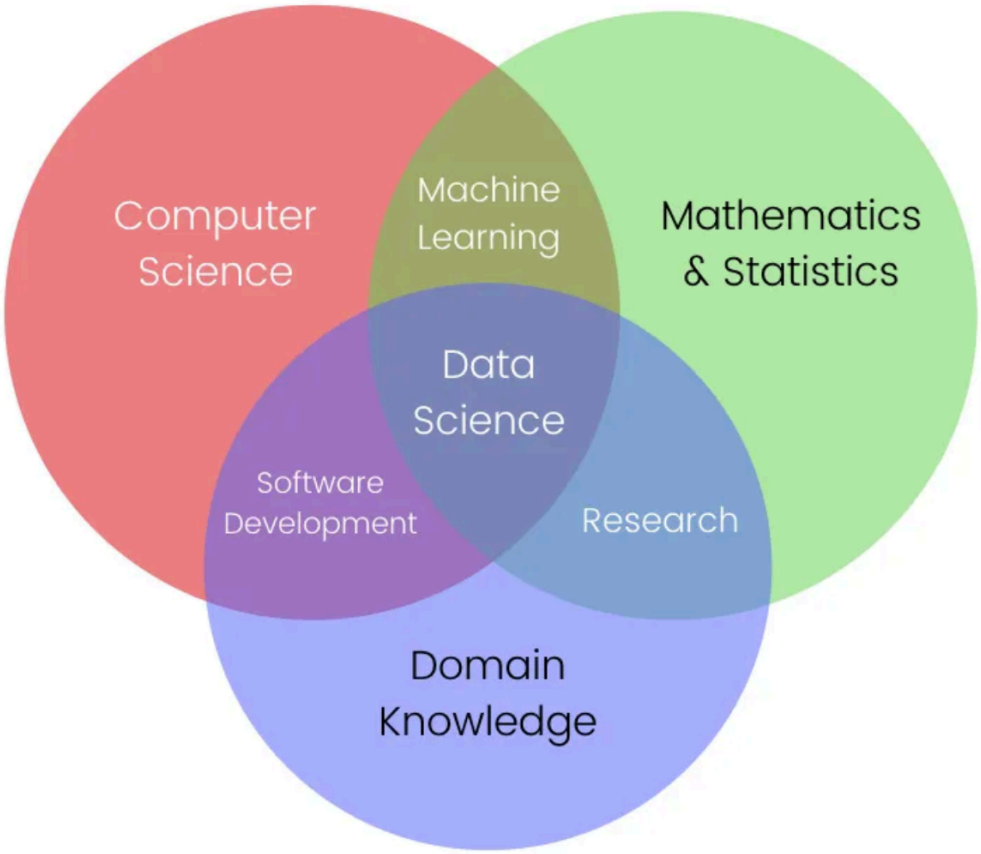
Data Science (วิทยาการข้อมูล) คืออะไร

Data Science หรือในภาษาไทยแปลว่า “วิทยาการข้อมูล” คือ ศาสตร์ที่รวมเอาความรู้ด้านวิทยาการคอมพิวเตอร์(Computer Science) ด้านคณิตศาสตร์และสถิติ(Math & Statistics) ด้านความรู้เฉพาะทาง(Domain Knowledge)มาประยุกต์รวมกันเพื่อจัดเก็บ รวบรวม ตรวจสอบ วิเคราะห์ และนำเสนอข้อมูลที่ออกมาในรูปแบบของข้อมูลเชิงลึก (Insight) เพื่อนำไปใช้ประโยชน์ในด้านต่าง ๆ เช่น เศรษฐศาสตร์ การเงิน โลจิสติกส์ วิศวกรรม การแพทย์ เป็นต้น



ขอบคุณรูปจาก medium.com

Data Science กับ Python

ภาษา Python เป็นหนึ่งในภาษาที่นิยมใช้กันในงานสาย Data Science

ข้อดีของการใช้งาน Python ทำงาน Data Science สามารถสรุปได้ดังนี้

1. เป็นมิตรกับมือใหม่ Python ใช้งานง่ายและมี syntax ที่เรียบง่าย ภาษานี้จึงเป็นเครื่องมือที่เหมาะกับมือใหม่

2. มีชุดเครื่องมือสำหรับคณิตศาสตร์และสถิติ Python มีฟังก์ชั่นในการคำนวณทางคณิตศาสตร์ ทำเรื่องสถิติ และสร้างโมเดลทางสถิติ มันจึงเป็นภาษาที่เหมาะกับการใช้งานทางด้านวิทยาศาสตร์ข้อมูลมาก

3. เหมาะกับการทำ data visualization Python เหมาะกับการทำ data visualization ซึ่งจะช่วยให้เราเข้าใจข้อมูลได้ดี เช่น เข้าใจความสัมพันธ์ที่น่าจะเป็นไปได้ เห็นความสัมพันธ์ที่ไม่ปรากฏเด่นชัด และเห็นเทรนด์ต่างๆ

4. มี open-source library จำนวนมากให้ใช้ Python เป็นภาษาที่มีห้องสมุด open-source จำนวนมากให้ใช้ และเป็นห้องสมุดที่มีมากกว่าส่วนของการคำนวณ สถิติ และ data visualization

5. มีประสิทธิภาพและปรับขนาดได้ Python เหมาะจะนำไปใช้ในงาน Data Science เพราะมันทั้งมีประสิทธิภาพ และใช้กับงานทั้งใหญ่และเล็กได้

6. มีชุมชนที่เข้มแข็ง Python มีชุมชนที่เข้มแข็ง และทำงานต่อเนื่องเพื่อพัฒนา libraries สำหรับงานวิทยาศาสตร์ข้อมูลให้ดีขึ้น

library พื้นฐานสำหรับงาน Data Science ของ Python

ภาษาสารพัดประโยชน์อย่าง Python ถ้าจะต้องจำทุกคำสั่งก็คงจะต้องใช้แรงไม่น้อย เลยมีผู้พัฒนาหลายๆ คน พยายามที่จะนำคำสั่งต่างๆ ของ Python มาสร้างเป็นชุดคำสั่ง หรือเป็น Package เพื่อให้สามารถทำงานตามวัตถุประสงค์แต่ละด้านได้อย่างมีประสิทธิภาพมากขึ้น โดยที่เรียกสิ่งที่ว่านี้ว่า “Python Library”

1. NumPy

มีชื่อเต็มว่า “Numerical Python” โดดเด่นในด้านการคำนวณ และการทำงานกับตัวเลข (NumPy ถือเป็น Scientific Computing Library ที่สำคัญมากของ Python)

นอกจากนี้ NumPy ยังมีความสามารถสำคัญในการสร้าง Array (โครงสร้างข้อมูล) และ Multidimensional Array ได้ ทำให้การคำนวณบน Python มีความรวดเร็วมากขึ้น ซึ่งแม้ Python พื้นฐานเอง จะมี Python list ที่มีความคล้ายคลึงกับ Array แต่ NumPy สามารถจัดการข้อมูลเหล่านี้ได้เร็วกว่าการใช้ Python list ธรรมดาๆ

2. Pandas

สุดยอด Library แห่งการจัดการข้อมูล (Data Wrangling/ Data Cleaning) และการวิเคราะห์ข้อมูล (Data Analysis)

pandas สามารถเชื่อมต่อกับแหล่งข้อมูลได้หลากหลาย หลังจากนั้นก็สามารถจัดเตรียมข้อมูล ทำความสะอาด และจัดรูปแบบให้พร้อมกับการนำไปวิเคราะห์ ตลอดจนการแสดงผล

3. Matplotlib

เป็น Library อันดับหนึ่งในการสร้างกราฟ และทำ Data Visualization (คล้ายกับ MATLAB ซึ่งมาพร้อมกับ Python) โดยที่ Matplotlib สามารถสร้างกราฟได้หลายประเภทเพื่อตอบโจทย์การทำงานของผู้ใช้ให้ได้หลากหลาย เช่น กราฟเส้น แผนภูมิจุดแบบกระจัดกระจาย (Scatter Plot), กราฟแท่งและฮิสโตแกรม, แผนภูมิบ็อกซ์และวิสเกอร์ (Box Plot หรือ Whisker Plot) และอื่นๆ

ทำไม Numpy Array ถึงน่าใช้กว่า List

แน่นอนว่าสิ่งที่เราสงสัย ณ ตอนนี้คือ แล้ว List กับ Numpy Array เนี่ย ตัวไหนมันทำงานได้ดีกว่ากันล่ะ หรือมันทำงานได้รวดเร็วเท่ากันกันแน่ แน่นอนว่ามีวิธีหาคำตอบสำหรับคำถามนี้ครับ ซึ่งก็คือการลงมือเขียนโค้ดนั่นเองเทียบประสิทธิภาพให้ทุกคนเห็นกับตานั่นเองครับ!!! โดยจะยกตัวอย่างให้เห็นแบบสั้นๆ ด้วยการเทียบประสิทธิภาพด้วยการให้ทั้ง 2 ตัวนี้มีสมาชิกทั้งหมด 1,000,000 ตัว (ใช่ครับ คาดว่าประมาณนี้กำลังดีเลยในการยกตัวอย่างง่ายๆ เนื่องจากงาน Data science นั้นต้องจัดการกับข้อมูลที่มีจำนวนมากๆ ดังนั้น 1 ล้านตัวนั้นไม่น้อยเกินไปครับ) โดยเราจะนำสมาชิกทุกตัวใน List และ Numpy Array มาคูณเข้าด้วย 2 ทั้งหมด แล้วมาดูกันครับว่าใครสามารถทำงานได้ดีกว่ากัน

In [1]:
```
#import Numpy library to create Numpy Array and
#import time library for making computing performance metric
import numpy as np
import time as t
```

In [2]:
```
# initialize number of elements and another subject
n = 1000000
ListA = []

# Create a List and Array that has 1M elements

# List
for i in range(n):
    ListA.append(i)

# Numpy Array
numpy_array = np.arange(n)
```

In [3]:
```
# Perform multiply by 2, then observe time efficiency

# By List
start_time_list = t.time()
List = [i*2 for i in ListA]
list_perform_time = t.time() - start_time_list
print('Time used for computation by List is {} seconds'.format(list_perform_

# By Numpy Array
start_time_array = t.time()
numpy_array = numpy_array*2
array_perform_time = t.time() - start_time_array
print('Time used for computation by Numpy is {} seconds'.format(array_perfor
```

```
Time used for computation by List is 0.01481318473815918 seconds
Time used for computation by Numpy is 0.0008368492126464844 seconds
```

In [4]:
```
# Time efficiency of List and Numpy Array on the same task

print(f'Differance between performing by List and Numpy Array are {list_perf
print(f'Ratio between List and Numpy Array computation time is {list_perform
```

```
Differance between performing by List and Numpy Array are 0.0139763355255126
95
Ratio between List and Numpy Array computation time is 17.7011396011396
```

In [5]:
```
# Perform multiply by 2, then observe time efficiency

# By List
start_time_list = t.time()
List = [i**2 for i in ListA]
list_perform_time = t.time() - start_time_list
print('Time used for computation by List is {} seconds'.format(list_perform_

# By Numpy Array
start_time_array = t.time()
numpy_array = numpy_array**2
array_perform_time = t.time() - start_time_array
print('Time used for computation by Numpy is {} seconds'.format(array_perfor
```

```
Time used for computation by List is 0.02100515365600586 seconds
Time used for computation by Numpy is 0.0008189678192138672 seconds
```

In [6]:
```
# Time efficiency of List and Numpy Array on the same task

print(f'Differance between performing by List and Numpy Array are {list_perf
print(f'Ratio between List and Numpy Array computation time is {list_perform
```

```
Differance between performing by List and Numpy Array are 0.0201861858367919
92
Ratio between List and Numpy Array computation time is 25.648326055312953
```

บทสรุป

จะเห็นว่าตัวของ Numpy Array สามารถทำงานได้รวดเร็วกว่าตัวของ List อยู่หลายเท่าเลยทีเดียวครับ

เทียบจากการนำมาคูณด้วย 2แล้ว การทำงานของ Numpy นั้นเร็วกว่า List ถึง 21 เท่าเลยทีเดียว!

เทียบจากการนำมายกกำลัง2 แล้ว การทำงานของ Numpy นั้นเร็วกว่า List ถึง 33 เท่าเลยทีเดียว!

ลองคิดดูนะครับว่าถ้าประมวณผลข้อมูลชุด A ใช้เวลา 1 ชั่วโมงโดยการใช้ Numpy Array แล้วถ้าเราเลือกใช้ List ล่ะ มันจะนานขนาดไหน) ซึ่ง ณ ตอนนี้ทุกคนก็น่าจะพอเห็นภาพแบบคร่าวๆแล้วนะครับว่าเจ้าตัว Numpy Array เนี่ยมันเร็วกว่าเยอะมากๆเมื่อต้องจัดการกับข้อมูลจำนวนขนาดใหญ่และมีความซับซ้อนสูงครับ

List & Array (ndarray)

ชนิดข้อมูล

- Array สมาชิกใน Array ต้องมีชนิดข้อมูลเหมือนกัน

- List สมาชิกมีชนิดข้อมูลต่างกันได้

ขนาด

- Array ขนาดที่แน่นอนเปลี่ยนแปลงขนาดไม่ได้
- List มีขนาดที่ยืดหยุ่นกว่า

Array คือการนำข้อมูลมาอยู่ในกลุ่ม เดียวกัน โดยสมาชิกภายใน Array ต้องมีชนิดข้อมูลเหมือนกัน

In [ ]: