

อรพิน ประวัตินริสุทธ์
ผู้เขียนหนังสือ Bestseller ด้าน Programming



PROVISION

Python สำหรับงาน Data Science Data Visualization และ Machine Learning

วิธีใช้งาน Library ยอดนิยม Pandas, NumPy, Matplotlib และ Scikit-learn
ตัวอย่างการใช้ Machine Learning : ทำนายยอดขาย • วิเคราะห์แนวโน้มมะเร็ง • อนุมัติเงินกู้
Sentimental Analysis (วิเคราะห์ข้อความ) • Image Recognition • Customer Segmentation



DOWNLOAD
[provision.co.th/
Python_Datascience](https://provision.co.th/Python_Datascience)

อรพิน ประวัตินริสุทธิ์
ผู้เขียนหนังสือ Bestseller ด้าน Programming

PROVISION

Python สำหรับงาน Data Science Data Visualization และ Machine Learning



Python สำหรับงาน Data Science Data Visualization และ Machine Learning

ผู้เขียน : อรพิน ประวัติวิสุทธิ
ออกแบบปก : ธนกฤต เกียรติศักดิ์กุล
จัดรูปเล่ม : อภิขญา สุทธิประภา

ISBN : 978-616-204-788-6
978-616-204-802-9 (E-book)

ราคา : 355 บาท

ข้อมูลบรรณานุกรมของหอสมุดแห่งชาติ

อรพิน ประวัติวิสุทธิ

Python สำหรับงาน Data Science
Data Visualization และ Machine Learning.
--กรุงเทพฯ : โปรวิชั่น, 2564. 400 หน้า.
1. ไพธอน (ภาษาคอมพิวเตอร์). I. ชื่อเรื่อง.
005.133

ชื่อและเครื่องหมายการค้าอื่นๆ ที่อ้างอิงในหนังสือฉบับนี้ และในเนื้อหาหรือสื่อบันทึกข้อมูลอื่นๆ ที่เกี่ยวข้อง (ถ้ามี) เป็นสิทธิโดยชอบด้วยกฎหมายของเจ้าของแต่ละราย บริษัท โปรวิชั่น จำกัด มิได้อ้างอิงว่าเป็นเจ้าของ ตัวแทน หรือมีส่วนเกี่ยวข้องแต่อย่างใด นอกจากนั้นทางบริษัทได้ใช้ความพยายามอย่างเต็มที่ในการตรวจสอบความถูกต้องของเนื้อหาเท่าที่จะทำได้ แต่มิได้ยอมรับหรือรับประกันความถูกต้องหรือทันสมัยของข้อมูลต่างๆ ในทุกกรณี ผู้อ่านพึงใช้วิจารณญาณของตนเอง หากพบข้อผิดพลาดหรือสงสัยใดๆ โปรดแจ้งที่อีเมล editor@provision.co.th เพื่อที่ทางบริษัทจะได้ตรวจสอบและแก้ไขตามความเหมาะสมในการพิมพ์ครั้งต่อไป รวมทั้งประกาศให้ทราบผ่านทางเว็บไซต์ www.provision.co.th/update ด้วย

สงวนสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดย บริษัท โปรวิชั่น จำกัด ห้ามนำส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ไปทำซ้ำ ตัดแปลงหรือเผยแพร่ต่อสาธารณชนไม่ว่ารูปแบบใดๆ นอกจากนี้จะได้รับอนุญาตเป็นลายลักษณ์อักษรล่วงหน้าจากทางบริษัทเท่านั้น ชื่อผลิตภัณฑ์และเครื่องหมายการค้าต่างๆที่อ้างถึงเป็นสิทธิโดยชอบด้วยกฎหมายของบริษัทนั้นๆ

PROVISION

จัดพิมพ์โดย : บริษัท โปรวิชั่น จำกัด
11/8 ซอยแจ้งวัฒนะ 14
แขวงทุ่งสองห้อง เขตหลักสี่ กทม. 10210
โทร: 0-2077-4058

ผู้ก่อตั้ง/ประธานกรรมการ : วดีน เพิ่มทรัพย์
กรรมการบริหาร : วงศ์ประชา จันทร์สมวงศ์
อนิรุทธิ์ รัชตะวราห์
ประสานงานกอง บก. : นนทพร ตันติเยี่ยมสกุล

ติดต่อสั่งซื้อหนังสือ

✉ sales@provision.co.th
โทร. 0-2077-4058 หรือ 08-1928-5299
🌐 www.dplussshop.com f dplussshop

ติดต่อโฆษณาและการตลาด

✉ marketing@provision.co.th
สอบถามปัญหา/ติดต่อกอง บก.
✉ editor@provision.co.th

ติดตามข่าวสาร

🌐 www.provision.co.th, www.dplusguide.com
f provision1991, dplusguide
🐦 @Provision1991, @dplusguide
📺 provision1991, dplusguide
📺 @dplusguide

จัดจำหน่ายโดย :
บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)

1858/87-90 ถ.เพชรรัตน
แขวงบางนาใต้ เขตบางนา กรุงเทพฯ 10260
โทร. 0-2826-8000 FAX: 0-2826-8999

ที่ปรึกษากฎหมาย : คุณไพบุลย์ อมรภิญโญเกียรติ
สำนักงานกฎหมาย P&P LAW FIRM โทร. 0-2651-2121

พิมพ์ที่ บริษัท พิมพ์ดี จำกัด
นายเสริม พูนพนิช ผู้พิมพ์ผู้โฆษณา พ.ศ. 2564



DOWNLOAD

provision.co.th/Python_Datascience

INTRODUCTION

ปัจจุบันมีข้อมูลปริมาณมากมายมหาศาล ที่เรียกว่า Big Data เกิดขึ้นตลอดเวลาทุกวัน คงเป็นเรื่องน่าเสียดาย หากเราจะปล่อยให้ข้อมูลเหล่านั้นทิ้งไปโดยไม่นำมาใช้ให้เกิดประโยชน์ใดๆ ดังนั้น จึงมีศาสตร์อย่างหนึ่ง ที่เรียกว่า Data Science เกิดขึ้นมา

Data Science คือ วิทยาศาสตร์ข้อมูล เป็นศาสตร์ที่รวบรวมเอาความรู้ด้านการเขียนโปรแกรม (Programming) ด้านคณิตศาสตร์ (Mathematics) และด้านสถิติ (Statistics) มาประยุกต์รวมกัน เพื่อให้ข้อมูลที่มีอยู่เกิดเป็นความรู้ (Knowledge) ใหม่ สามารถนำเอาข้อมูลเหล่านั้นมาสร้างมูลค่าทางธุรกิจได้

หนังสือเล่มนี้จะสอนงาน Data Science ต่างๆ โดยใช้ภาษา Python เนื่องจากไพธอนเป็นภาษาโปรแกรมมิ่งที่เข้าใจง่าย อีกทั้งยังมีไลบรารีต่างๆ สำหรับงาน Data Science ให้เรียกใช้งานด้วย จึงทำให้การเขียนโปรแกรมเกี่ยวกับงาน Data Science เป็นเรื่องที่ยั่งยืน แม้เราจะไม่มีความรู้ด้านคณิตศาสตร์หรือสถิติระดับสูง ก็สามารถเขียนโปรแกรมจัดการงาน Data Science ด้วย Python ได้โดยง่าย

นอกจากนี้ยังอธิบายถึงการนำ Data Visualization ซึ่งเป็นรูปแบบหนึ่งของการนำเสนอข้อมูลในรูปแบบของกราฟ หรือแผนภูมิ เพื่อสื่อสารให้ผู้อ่านข้อมูลเข้าใจข้อมูลได้ง่ายขึ้นด้วย

ตอนท้ายของหนังสือจะได้กล่าวถึง Machine Learning ให้ได้ทราบกัน โดยจะสอนการออกแบบโปรแกรม Machine Learning ด้วยภาษาไพธอน เพื่อสอนให้คอมพิวเตอร์เกิดการเรียนรู้และการพัฒนาจากประสบการณ์ จนกระทั่งสามารถสร้างโมเดลทำนายผลลัพธ์หรือตัดสินใจการทำงานได้ด้วยตนเองอย่างอัตโนมัติ ซึ่งการทำ Machine Learning ก็จะต้องอาศัยความรู้ด้าน Data Science เข้ามาช่วยงานด้วย

ท้ายสุด หากหนังสือเล่มนี้ขาดตกบกพร่องประการใด ผู้เขียนขออภัยมา ณ ที่นี้ด้วยนะค่ะ และถ้าหนังสือเล่มนี้จะทำให้ผู้อ่านชอบใจอยู่บ้าง ผู้เขียนก็ขอขอบคุณงามความดีให้กับคุณประสิทธิ์ ประวัตติบริสุทธิ และคุณวิภา ประวัตติบริสุทธิ คุณพ่อคุณแม่ของผู้เขียน ผู้ซึ่งมอบกำลังใจดีๆ ให้กับผู้เขียนเสมอมา กำลังใจนี้เป็นส่วนสำคัญยิ่งในการผลักดันให้ผู้เขียนสร้างสรรค์ผลงานเขียนต่างๆ ออกมาสู่สายตาของผู้อ่าน

อรพิน ประวัตติบริสุทธิ

อรพิน ประวัตติบริสุทธิ

Sun Certified Java Programmer 1.4 (SCJP 1.4)

Sun Certified Web Component Developer 1.4 (SCWCD 1.4)

● ประวัติการศึกษา

จบการศึกษาระดับปริญญาตรี สาขาวิชาวิทยาการคอมพิวเตอร์ประยุกต์ จากคณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ และจบการศึกษาปริญญาโท สาขาวิชาเทคโนโลยีสารสนเทศ จากคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

● ประวัติการทำงาน

มีประสบการณ์ด้านการเป็นนักวิเคราะห์และออกแบบระบบให้กับบริษัทเอกชนที่มีชื่อเสียง และมีผลงานเขียนติดอันดับหนังสือขายดี เช่น คู่มือเขียนโปรแกรมด้วย C ฉบับสมบูรณ์, One-stop Python เรียนรู้ภาษาไพธอนในเล่มเดียว, เริ่มต้นเขียนโปรแกรมด้วยภาษา Scratch3, คู่มือเริ่มต้นเขียนโปรแกรมด้วยภาษา JAVA, คู่มือสร้างเว็บไซต์ด้วย HTML5 CSS3 & JavaScript ฉบับสมบูรณ์, พัฒนาเว็บแอปบน Smartphone/Tablet ด้วย jQuery Mobile, คู่มือ Excel 2010/2013

AUTHOR

Contents

CHAPTER

01 ทำความรู้จักกับ Data Science

Big Data และ Data Science คืออะไร	12
ขั้นตอนการทำ Data Science (Data Science Pipeline)	13
Obtain	14
Scrub	14
Explore	15
Model	15
Intepret	16
Data Science กับ Python	17
ไลบรารีพื้นฐานสำหรับการทำ Data Science ของ Python	18
Pandas	18
NumPy	18
SciPy	19
Matplotlib	19
Seaborn	19

CHAPTER

02 การใช้งาน Jupyter Notebook และ Google Colab

ทำความรู้จักกับ Jupyter Notebook	21
Jupyter Notebook กับ Google Colab ต่างกันอย่างไร	21
การติดตั้งโปรแกรม Anaconda	22
การเรียกใช้งาน Jupyter Notebook	25
การสร้าง folder เพื่อจัดเก็บโค้ดโปรแกรม	27
การเขียนโปรแกรมลงบน Jupyter Notebook	28
การลบและเพิ่มเซลล์	29

การคัดลอก/ตัดเซลล์เดิมไปไว้ยังเซลล์ใหม่	31
การสลับที่เซลล์	33
การรวมเซลล์และแตกเซลล์	34
การบันทึกโค้ดโปรแกรม ใน Jupyter Notebook	36
การเปิดไฟล์โค้ดโปรแกรม ใน Jupyter Notebook	38
การเรียกใช้งาน Google Colab	39

CHAPTER

03 พื้นฐานไพธอน

การเขียนคอมเมนต์ในไพธอน	43
กฎการตั้งชื่อของภาษาไพธอน	43
Keywords ในภาษาไพธอน	44
ทำความรู้จักกับตัวแปร	44
การรับและแสดงผลข้อมูล	44
ชนิดข้อมูลของไพธอน	45
Numeric	45
Integers	45
Floating point numbers	45
Complex numbers	45
Boolean	46
String	47
การหาความยาวสตริง	47
การค้นหาตัวอักษรและกลุ่มตัวอักษรในสตริง	47
List	49
การหาความยาวลิสต์	49
การค้นหาตัวอักษรและกลุ่มตัวอักษรในลิสต์	49
การแก้ไขข้อมูลในลิสต์	51



การเพิ่มข้อมูลในลิสต์.....	51	คำสั่งทำซ้ำ.....	81
การลบข้อมูลในลิสต์.....	52	คำสั่งทำซ้ำ while	81
Tuple.....	52	การทำงานของคำสั่ง while	
Set.....	53	ร่วมกับคำสั่ง break และ continue.....	82
การหาความยาวเซต.....	53	การทำงานของคำสั่ง while	
การค้นหาข้อมูลในเซต.....	53	ร่วมกับคำสั่ง else.....	83
การเพิ่มข้อมูลในเซต.....	54	คำสั่งทำซ้ำ for	83
การลบข้อมูลในเซต	54	การทำงานของคำสั่ง for	
Dictionary.....	55	ร่วมกับฟังก์ชัน range().....	84
การหาความยาวดิกชันนารี.....	56	การทำงานของคำสั่ง for	
การค้นหาข้อมูลในดิกชันนารี.....	56	ร่วมกับคำสั่ง break และ continue.....	85
การแก้ไขข้อมูลในดิกชันนารี.....	57	การทำงานของคำสั่งทำซ้ำ for	
การเพิ่มข้อมูลในดิกชันนารี.....	57	ร่วมกับคำสั่ง else.....	86
การลบข้อมูลในดิกชันนารี.....	57	การสร้างและเรียกใช้ฟังก์ชัน.....	86
ตัวดำเนินการ (Operator) ในไพธอน.....	58	วิธีเรียกใช้งาน Built-in modules.....	88
ตัวดำเนินการทางคณิตศาสตร์		เรียกใช้โมดูลด้วยคำสั่ง import.....	88
(Arithmetic operators).....	58	เรียกใช้โมดูลด้วยคำสั่ง from ... import....	88
ตัวดำเนินการทางตรรกศาสตร์ (Logical operators).....	59		
ตัวดำเนินการระดับบิต (Bitwise operators).....	60		
ตัวดำเนินการกำหนดค่า (Assignment operators).....	63		
ตัวดำเนินการเปรียบเทียบ (Comparison operators).....	64		
ตัวดำเนินการเอกลักษณ์ (Identity operators).....	65		
ตัวดำเนินการสมาชิก (Membership operators).....	66		
ลำดับความสำคัญของตัวดำเนินการ.....	67		
การแปลงชนิดข้อมูล.....	69		
Implicit type conversion	69		
Explicit type conversion	69		
คำสั่งเงื่อนไข.....	74		
คำสั่งเงื่อนไข if	74		
คำสั่งเงื่อนไข if-else	75		
คำสั่งเงื่อนไข if-elif-else	77		
คำสั่งเงื่อนไข nested-if	79		
		มาทำความรู้จักกับไลบรารี NumPy กัน.....	90
		การติดตั้ง NumPy.....	90
		การสร้าง ndarray.....	90
		การสร้าง ndarray	
		จากข้อมูลชนิดอื่นของ Python.....	91
		การใช้ฟังก์ชันต่างๆ ของ NumPy	
		สร้าง ndarray.....	92
		zeros() และ ones().....	92
		full().....	93
		arange().....	94
		linspace().....	96

CHAPTER
04 รู้จักกับไลบรารี
NumPy

Contents

eye()	98
empty()	99
การสร้าง ndarray จากเลขสุ่ม	
ด้วยโมดูล Random	100
การเรียกใช้โมดูล Random	100
rand()	101
randint()	103
choice()	105
การกำหนดการกระจายของเลขสุ่มด้วย	
choice() (Random Distribution)	106
การสลับตำแหน่งสมาชิกแบบสุ่มใน	
ndarray (Shuffling & Premutation)	107
โอเปอเรชันกับ ndarray	
(Vectorization & Broadcasting)	108
Vectorization	108
Broadcasting	110
กฎการทำโอเปอเรชันของ ndarray	111
ndarray 2 ตัว มีจำนวนมิติเท่ากัน	
แต่จำนวนสมาชิกไม่เท่ากันในบางมิติ	112
ndarray 2 ตัว มีจำนวนมิติไม่เท่ากัน	
แต่จำนวนสมาชิกเท่ากันในมิติหนึ่ง	
หรือหลายมิติ	114
ndarray 2 ตัว มีจำนวนมิติเท่ากัน	
และจำนวนสมาชิกในต่างมิติเท่ากัน	117
ndarray 2 ตัว ซึ่งตัวใดตัวหนึ่ง	
มีจำนวนสมาชิกเป็น 1	118
ndarray 2 ตัวที่ไม่สามารถนำมา	
ทำโอเปอเรชันกันได้	118
การเข้าถึงสมาชิกใน ndarray (Array Indexing) ...	119
การเข้าถึงสมาชิกใน ndarray	
มากกว่า 1 มิติ	119
Index Slicing	120

การทำ Index Slicing ด้วย operator	121
การทำ Index Slicing ด้วย	
Integer Index Arrays	124
การทำ Index Slicing ด้วย	
Boolean Index Arrays	127
ฟังก์ชันการทำงานอื่นๆ ของ NumPy	128
round()	128
abs()	129
power(), log() และ exp()	129
sum()	130
sqrt()	131
logical_and(), logical_or(),	
logical_not() และ logical_xor()	131
all() และ any()	132
min(), max(), mean() และ median()	132
sort()	133

CHAPTER

05

โครงสร้างข้อมูล ของ Pandas

Pandas คืออะไร	136
รู้จักกับโครงสร้างข้อมูลของ Pandas	136
Series	136
DataFrame	137
Panel	138
เรียนรู้การสร้างข้อมูลแบบ Series	138
การสร้างข้อมูลแบบ Series จาก List	139
การสร้างข้อมูลแบบ Series	
โดยกำหนด index	140
การสร้างข้อมูลแบบ Series	
โดยกำหนด dtype	141
การสร้างข้อมูลแบบ Series จาก Tuple	141
การสร้างข้อมูลแบบ Series จาก Dictionary	142

การสร้างข้อมูลแบบ Series จาก ndarray ของ NumPy.....	142
เรียนรู้การสร้างข้อมูลแบบ DataFrame.....	143
การสร้างข้อมูลแบบ DataFrame จาก Series	144
<i>การสร้าง DataFrame จากข้อมูล Series ชุดเดียว.....</i>	<i>144</i>
<i>การสร้าง DataFrame จากข้อมูล Series หลายชุด.....</i>	<i>144</i>
การสร้างข้อมูลแบบ DataFrame จาก List... ..	146
การสร้างข้อมูลแบบ DataFrame จาก List ของ Dictionary	146
การสร้างข้อมูลแบบ DataFrame จาก ndarray ของ NumPy.....	147
การสร้างข้อมูลแบบ DataFrame จากไฟล์ Excel.....	148
<i>การอ่านข้อมูลจากไฟล์ Excel.....</i>	<i>149</i>
การสร้างข้อมูลแบบ DataFrame จากไฟล์ CSV	156
<i>การอ่านและเขียนข้อมูลไฟล์ CSV.....</i>	<i>156</i>

CHAPTER

06

การจัดการกับข้อมูล Series และ DataFrame ด้วย Pandas

การดึงข้อมูลใน Series	159
การดึงกลุ่มข้อมูลใน Series	162
การดึงกลุ่มข้อมูลโดยระบุตำแหน่งของข้อมูล.....	164
การดึงกลุ่มข้อมูลด้วย loc().....	164
การเรียงลำดับข้อมูลใน Series.....	165
การกำหนดค่า ascending.....	166

การกำหนดค่า na_position.....	166
การกำหนดค่า inplace	166
การเพิ่มข้อมูลใน Series	167
การลบข้อมูลใน Series.....	169
การอัปเดตข้อมูลใน Series	170
การดึงข้อมูลใน DataFrame.....	170
การดึงข้อมูลด้วย head() และ tail().....	170
การดึงข้อมูลด้วย at() และ iat().....	172
การดึงข้อมูลด้วย loc() และ iloc()	172
การดึงข้อมูล 1 คอลัมน์ของทุกแถว.....	173
การดึงข้อมูลหลายคอลัมน์ของทุกแถว	173
การดึงข้อมูล 1 แถวของทุกคอลัมน์.....	174
การดึงข้อมูลหลายแถวของทุกคอลัมน์	175
การดึงข้อมูล 1 ค่าจากแถวและคอลัมน์	176
การดึงข้อมูลหลายแถวหลายคอลัมน์.....	177
การดึงแถวข้อมูลโดยกำหนดเงื่อนไข	178
การดึงแถวข้อมูลด้วยฟังก์ชัน iteritems()... ..	179
การดึงแถวข้อมูลด้วยฟังก์ชัน iterrows()	180
การดึงแถวข้อมูลด้วยฟังก์ชัน itertuples()....	181
การเรียงลำดับข้อมูลใน DataFrame.....	182
การเรียงลำดับค่า index	182
การเรียงลำดับค่าข้อมูลโดยพิจารณาจากคอลัมน์เดียว.....	183
การเรียงลำดับค่าข้อมูลโดยพิจารณาจากหลายๆ คอลัมน์.....	184
การเปลี่ยนชื่อคอลัมน์ใน DataFrame.....	185
การเพิ่มคอลัมน์ใน DataFrame	186
เพิ่มคอลัมน์ด้วยวิธีสร้างลิสต์ของคอลัมน์	186
เพิ่มคอลัมน์ด้วยฟังก์ชัน insert().....	187
เพิ่มคอลัมน์ด้วยฟังก์ชัน assign()	188

Contents

การเพิ่มแถวใน DataFrame.....	188
การลบคอลัมน์และแถวใน DataFrame.....	190
การอัปเดตข้อมูลใน DataFrame	192
การจัดกลุ่มข้อมูลด้วย groupby().....	193
การจัดกลุ่มข้อมูลใน Series	195
การจัดกลุ่มข้อมูลใน DataFrame ด้วยคอลัมน์เดียว.....	196
การจัดกลุ่มข้อมูลใน DataFrame ด้วยคอลัมน์หลายคอลัมน์.....	199

CHAPTER

07 ฟังก์ชันของ Series และ DataFrame ใน Pandas

ฟังก์ชันการทำงานที่น่าสนใจของ Series.....	202
ฟังก์ชันทางคณิตศาสตร์และสถิติ.....	202
ฟังก์ชันตรวจสอบข้อมูลและ ทำงานกับข้อมูลใน Series	204
ฟังก์ชันเกี่ยวกับการทำงาน กับ missing values	207
missing values คืออะไร	207
ฟังก์ชันจัดการกับ index.....	209
ฟังก์ชันการทำงานที่น่าสนใจของ DataFrame.....	212
ฟังก์ชันทางคณิตศาสตร์และสถิติ.....	212
ฟังก์ชันตรวจสอบข้อมูลและ ทำงานกับข้อมูลใน DataFrame	214
ฟังก์ชันเกี่ยวกับการทำงาน กับ missing value	220
ฟังก์ชันจัดการกับ index.....	221

CHAPTER

08 การสร้าง Pivot Table ด้วย Pandas

รู้จักกับ Pivot Table ใน Excel	223
การสร้างรายงานด้วย PivotTable	224
การเลือกข้อมูลเพื่อแสดงใน PivotTable.....	225
Pivot Table ใน Pandas.....	227
สร้าง Pivot Table ด้วย index ตัวเดียว.....	228
สร้าง Pivot Table ด้วย index หลายตัว.....	229
การเพิ่มคอลัมน์ลงใน PivotTable	230
การจัดการกับ missing value ใน PivotTable... ..	231
การแสดงค่าผลรวมของกลุ่มใน PivotTable.....	232
การกรองข้อมูล (Filter) ใน PivotTable.....	233
สร้าง Pivot Table โดยไม่กำหนดค่า values.....	234
สร้าง Pivot Table โดยกำหนด aggfunc หลายตัว.....	235

CHAPTER

09 การทำ Data Visualization ด้วย Pandas

Data Visualization คืออะไร.....	238
Library สำคัญในการทำ Data Visualization.....	238
Matplotlib.....	238
Seaborn.....	239
การเลือกรูปแบบ Visualization ให้เหมาะสมกับข้อมูล.....	239
การพล็อตกราฟด้วยฟังก์ชัน plot()	239
kind : กำหนดประเภทกราฟ	240
x : กำหนดค่าที่จะแสดงในแกน x	
y : กำหนดคอลัมน์ที่ต้องการ พล็อตกราฟพร้อมกับแกน x.....	240



grid : ตีเส้นตารางบนกราฟ.....	241
legend : กำหนดคำอธิบายสัญลักษณ์ ในกราฟ.....	242
subplots : พล็อตกราฟแยกตามคอลัมน์ ...	242
use_index : กำหนดลำดับข้อมูล บนแกน x ด้วย index.....	243
fontsize : กำหนดขนาดฟอนต์.....	244
rot : กำหนดมุมในการหมุนเลเบล ของแกน x.....	244
stacked : พล็อตกราฟแบบวางซ้อนกัน เป็นชั้น.....	245
title : กำหนดหัวเรื่องกราฟ.....	245
layout : กำหนดเลย์เอาท์ในการแสดงกราฟ	246
figsize : กำหนดขนาดกราฟ.....	246
xlabel และ ylabel : กำหนดชื่อเลเบล ของแกน x,y.....	247
position : กำหนดตำแหน่งการจัดวาง ของ bar charts.....	247
xticks และ yticks : กำหนดค่า ลำดับข้อมูลของแกน x,y.....	249
color : กำหนดสีของกราฟ.....	249
ประเภทของกราฟหรือแผนภูมิใน Pandas.....	250
Bar charts.....	250
Horizontal bar charts	252
Line graphs.....	253
Area plots.....	256
Pie charts.....	257
Box plots.....	260
Histogram.....	264
Scatter plots	269
Hexbin plots.....	273
Kernel Density Estimate charts.....	278

CHAPTER

10

การทำ Data Visualization
ด้วย Matplotlib

รู้จักกับไลบรารี Matplotlib.....	282
การบันทึกกราฟออกมาเป็นไฟล์.....	284
เรียนรู้การทำงานของฟังก์ชัน plot() ใน Matplotlib.....	286
linestyle : กำหนดรูปแบบเส้นกราฟ	286
linewidth : กำหนดความหนา ของเส้นกราฟ	287
marker : กำหนดตัวเน้นจุดพิกัด.....	287
markersize : กำหนดขนาดของ ตัวเน้นจุดพิกัด.....	293
markeredgecolor : กำหนดสีขอบ ของตัวเน้นจุดพิกัด	294
markerfacecolor : กำหนดสีด้านใน ของตัวเน้นจุดพิกัด	294
color : การกำหนดสีของกราฟ.....	295
format strings : กำหนดรายละเอียด ให้กับ marker แบบย่อ	295
title : กำหนดหัวเรื่องกราฟ.....	296
xlabel และ ylabel : กำหนดชื่อเลเบล ของแกน x,y.....	296
xticks และ yticks : กำหนดค่าลำดับ ข้อมูลของแกน x,y	297
figure : ปรับแต่ง container สำหรับการพล็อตกราฟ	298
การปรับขนาดรูปภาพกราฟ.....	298
การพล็อตกราฟหลายตัวด้วย subplots.....	299
legend : กำหนดคำอธิบายสัญลักษณ์ ในกราฟ.....	302

Contents

กำหนดสไตล์ของกราฟด้วยฟังก์ชัน style().....	304
กำหนดใช้ฟอนต์ภาษาไทยในกราฟ.....	306
การกำหนดฟอนต์สำหรับออบเจ็กต์ต่างๆ.....	306
การกำหนดฟอนต์สำหรับใช้งาน กับกราฟทุกตัว.....	307
การสร้าง Bar charts.....	308
การกำหนดรูปแบบลายของแท่งกราฟ.....	310
การจัดกลุ่มแท่งกราฟ.....	314
การสร้าง Horizontal bar charts.....	318
การสร้าง line graphs.....	319
การสร้าง Area plots.....	320
การสร้าง Pie charts.....	321
การสร้าง Box plots.....	323
การสร้าง Histogram.....	327
การสร้าง Scatter plots.....	328
การสร้าง Hexbin plots.....	330
การสร้าง Kernel Density Estimate charts.....	332


CHAPTER

11

Machine Learning (ML)

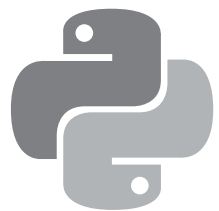
Machine Learning คืออะไร.....	335
ประเภทของ Machine Learning.....	337
Supervised Learning.....	337
Unsupervised Learning.....	340
Reinforcement learning.....	340
ไลบรารี Scikit-learn กับ Machine Learning Algorithm.....	341

Linear Regression.....	341
การทำงานของ Simple Linear Regression.....	341
ตัวอย่าง : ทำนายยอดขายด้วย Simple Linear Regression.....	347
การทำงานของ Multiple Linear Regression.....	348
ตัวอย่าง : ทำนายเงินเดือนด้วย Multiple Linear Regression.....	348
Logistic Regression.....	349
ทำความเข้าใจกับ Training data และ Testing data.....	352
ตัวอย่าง : วิเคราะห์แนวโน้มการเป็น มะเร็งเต้านมด้วย Logistic Regression.....	353
การแสดงผล Confusion Matrix แบบกราฟิก.....	359
การแสดงผล Classification Report.....	360
Support Vector Machine (SVM).....	364
ตัวอย่าง : วิเคราะห์การอนุมัติเงินกู้ด้วย SVM.....	367
Naïve Bayes Classification.....	372
ตัวอย่าง : Sentimental Analysis วิเคราะห์ ความรู้สึกจากข้อความด้วย Naïve Bayes.....	377
K-Nearest Neighbors (K-NN).....	381
ตัวอย่าง : Image Recognition วิเคราะห์และทำนายรูปภาพด้วย K-NN.....	384
การลดมิติข้อมูลด้วย PCA.....	391
K-Means Clustering.....	394
ตัวอย่าง : การทำ Customer Segmentation ด้วย K-Means Clustering.....	397



CHAPTER

01



ทำความรู้จักกับ Data Science

Big Data และ Data Science คืออะไร

ปัจจุบันอินเทอร์เน็ตได้เข้ามามีบทบาทกับการดำรงชีวิตประจำวันของคนในสังคมมากขึ้น สื่อสังคมออนไลน์ (Social Media) ต่าง ๆ ก็มีอยู่เป็นจำนวนมาก เช่น Facebook, Twitter, Instagram, Youtube, Blogs, Flickr, Online Shopping เป็นต้น ทำให้เกิดข้อมูลจำนวนมากมายมหาศาล ที่เรียกว่า Big Data ขึ้น

Big Data คือ ข้อมูลปริมาณมากมายมหาศาล (Volume) ที่มีการเปลี่ยนแปลงอย่างรวดเร็วตลอดเวลา (Velocity) และมีความหลากหลายของข้อมูล (Variety) เช่น เราเปิดใช้งาน Social Media ก็ทำให้มีข้อมูลรูปภาพ เสียง วิดีโอ รวมถึงเนื้อหาข้อมูลต่าง ๆ เกิดขึ้นมากมาย และข้อมูลแทบจะเปลี่ยนแปลงตลอดเวลาทุกวินาทีเลยทีเดียว ซึ่งทำให้ยากที่จะจัดการกับข้อมูลเหล่านี้ด้วยระบบจัดการข้อมูลในรูปแบบเดิม ๆ อย่างเช่น ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)

Big Data เรียกได้ว่าเป็นข้อมูลดิบ เรายังไม่สามารถนำข้อมูลไปใช้ให้เกิดประโยชน์ต่อธุรกิจได้ จึงมี Data Science เกิดขึ้นมา หากจะเปรียบเทียบง่าย ๆ Big Data ก็เหมือนน้ำมันดิบ คือเป็นข้อมูลดิบที่ยังไม่สามารถนำไปใช้งานได้ ส่วน Data Science คือ กระบวนการกลั่นน้ำมันให้พร้อมสำหรับการใช้งาน คือ เป็นกระบวนการกลั่นข้อมูลของ Big Data ให้พร้อมใช้งานนั่นเอง

Data Science คือ วิทยาศาสตร์ข้อมูล เป็นศาสตร์อย่างหนึ่งที่รวมเอาความรู้ด้านการเขียนโปรแกรม (Programming) ด้านคณิตศาสตร์ (Mathematics) และด้านสถิติ (Statistics) มาประยุกต์รวมกัน เพื่อทำให้ข้อมูลที่มีอยู่เกิดเป็นความรู้ (Knowledge) ใหม่ ๆ เกิดเป็นข้อมูลที่มีมูลค่า สามารถนำไปใช้ช่วยสนับสนุนการตัดสินใจวางแผนทางธุรกิจและช่วยสร้างประโยชน์ทางธุรกิจได้

ตัวอย่างเช่น เราได้ข้อมูลลูกค้าปริมาณมาก ๆ มาจากสื่อสังคมออนไลน์ Data Science ก็จะนำข้อมูลเหล่านี้มาผ่านกระบวนการต่าง ๆ เพื่อทำให้ได้ข้อมูลออกมาว่าลูกค้ามีความชอบในเรื่องไหนหรือสนใจเรื่องใดเป็นพิเศษ เพื่อนำข้อมูลไปวางแผนกลยุทธ์ให้เข้าถึงกลุ่มลูกค้าได้มากขึ้น ซึ่งช่วยให้ธุรกิจของเราพัฒนาต่อไปได้ เป็นต้น

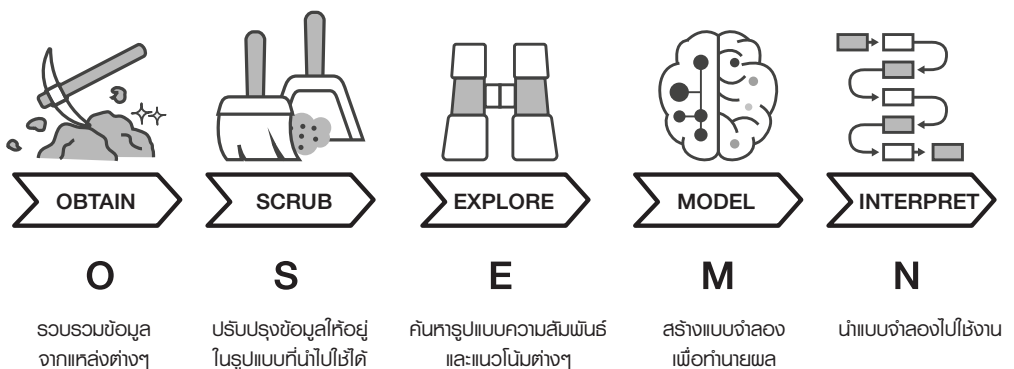
ขั้นตอนการทำ Data Science (Data Science Pipeline)

การทำ Data Science ประกอบด้วยกระบวนการที่ต้องใช้ทักษะหลายอย่างในการทำ ทั้งการวิเคราะห์รูปแบบ (pattern) ของข้อมูล, การคิดคำถามเพื่อสอบถามข้อมูล หรือการเลือกใช้อัลกอริทึมเพื่อนำมาวิเคราะห์ข้อมูล การทำงานเหล่านี้อยู่ในกระบวนการ (process) ที่เป็นขั้นตอนตามลำดับ เรียกว่า Data Science Pipeline ซึ่งต้องใช้ Data Scientist หรือนักวิทยาศาสตร์ข้อมูลมาทำงานกับข้อมูลในกระบวนการนี้ เพื่อได้ผลลัพธ์เป็นข้อมูลเชิงลึกของชุดข้อมูลนั้น

ขั้นตอน 5 ขั้นของ Data Science Pipeline มีตัวย่อคือ “OSEMN” (ฟังเสียงกับคำว่า “Awesome”) โดยประกอบไปด้วย

- Obtain
- Scrub
- Explore
- Model
- iNterpret

Data Science Pipeline



Obtain

ขั้นตอนแรกของการทำ Data Science คือการรวบรวมข้อมูล ข้อมูลนั้นอาจจะมาจากหลาย ๆ แหล่ง ซึ่งวิธีการดึงข้อมูลอาจจะแตกต่างกัน เช่น เป็นข้อมูลในรูปแบบไฟล์ชนิดต่าง ๆ ที่ดาวน์โหลดมาจากเว็บไซต์ เช่น HTML หรือ Excel รวมไปถึงไฟล์ภาพหรือเสียง จากนั้นจึงทำการ extract ข้อมูลจากเนื้อหาของไฟล์นั้น หรือคิวรี (query) ข้อมูลมาจากรฐานข้อมูลหรือดึงข้อมูลผ่าน API (ซึ่งปัจจุบันการดึงข้อมูลจาก Twitter หรือ Facebook มักทำในรูปแบบนี้) รวมไปถึงการรวบรวมข้อมูลจากการทำสำรวจ (survey)

ข้อมูลที่ได้ในขั้นตอนนี้อาจอยู่ในรูปแบบ plain text, csv, JSON, HTML หรือ XML เป็นต้น

Scrub

เมื่อเราได้ข้อมูลจากขั้นตอน Obtain มาแล้วนั้น ข้อมูลอาจอยู่ในรูปแบบที่ต้องมีการปรับรูปแบบหรือมีค่าผิด เช่น มีค่าที่หายไปบางคอลัมน์ใน Excel, ข้อมูลไม่มีความสัมพันธ์กันหรือผิดจากค่าอื่น ๆ ในคอลัมน์เดียวกัน, ข้อมูลอาจเว้นวรรคหรือมีช่องว่างไม่เท่ากันหรือไม่เป็นรูปแบบเดียวกัน, รวมไปถึงอาจมีข้อมูลส่วนที่เราไม่สนใจจะนำมาวิเคราะห์เกินมา ซึ่งขั้นตอนนี้จึงต้องมีการทำงานกับข้อมูลคือการ Scrub หรือ Clean ข้อมูลนั่นเอง

ในขั้นตอน Scrub นี้ เราอาจมีการทำอย่างใดอย่างหนึ่งกับข้อมูลดังนี้

- แยกข้อมูลในคอลัมน์ให้กลายเป็นหลายคอลัมน์
- เปลี่ยนค่าข้อมูลให้อยู่ในรูปแบบที่ต้องการ
- แยกคำ เช่น ชื่อ นามสกุล ให้แยกเว้นวรรค
- เติมค่าให้คอลัมน์ เช่น บางแถวข้อมูลอาจมีคอลัมน์เป็นค่าว่าง
- เปลี่ยนรูปแบบ (format) ข้อมูล

ในกระบวนการนี้อาจใช้ Tool หรือ Programming Language ต่าง ๆ มาช่วยในการทำ scrub และ clean เช่น Python (Pandas), R หรือ SAS

ทั้งนี้ในกระบวนการ OSEMN นั้น กระบวนการที่นักวิทยาศาสตร์ข้อมูล (Data Scientist) ต้องใช้เวลาในการทำงานส่วนใหญ่คือขั้นตอนของการ Obtain และ Scrub ซึ่งในบางครั้งใช้เวลากว่า 80% ของโปรเจกต์ในการ Scrub เพื่อให้ข้อมูลอยู่ในรูปแบบที่พร้อมในการนำไปวิเคราะห์และสร้างโมเดลข้อมูลในขั้นตอนต่อไป

Explore

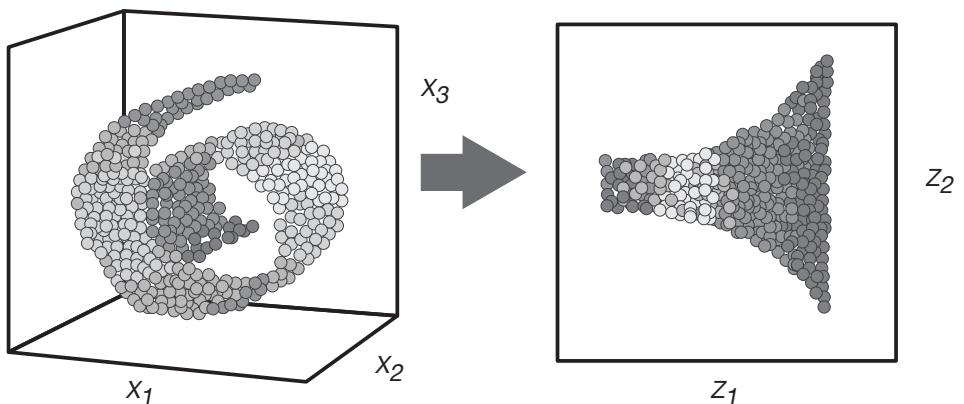
หลังจากทำการ Scrub หรือ Clean ข้อมูลแล้วนั้น ขั้นตอนถัดไปคือการค้นหาข้อมูลหรือส่วนของข้อมูลตามเรื่องที่เราต้องการวิเคราะห์ ซึ่งจะเป็นการเข้าไปค้นหาและทำความเข้าใจรูปแบบข้อมูล (pattern) ในกระบวนการนี้เราจะนำข้อมูลมาแสดงผล (Visualization) ในรูปแบบกราฟ หรือ chart ต่าง ๆ และนำกระบวนการทดสอบทางสถิติมาใช้กับชุดข้อมูล

กระบวนการนี้ต้องอาศัยการวิเคราะห์และสังเกตจากผลลัพธ์ของการคำนวณทางสถิติหรือกราฟต่าง ๆ เพื่อค้นหาหรือ trend ของข้อมูลในเรื่องที่สนใจ ซึ่งสิ่งสำคัญของกระบวนการนี้คือการทดสอบทางสถิติหรือสร้าง Visualization ให้ตรงกับความต้องการทางธุรกิจ (Business User)

ในกระบวนการนี้สามารถนำ Programming Language มาใช้งานได้ เช่น ไลบรารี NumPy, Matplotlib ของ Python หรือ GGplot2 ของภาษา R ในการทำ Visualization

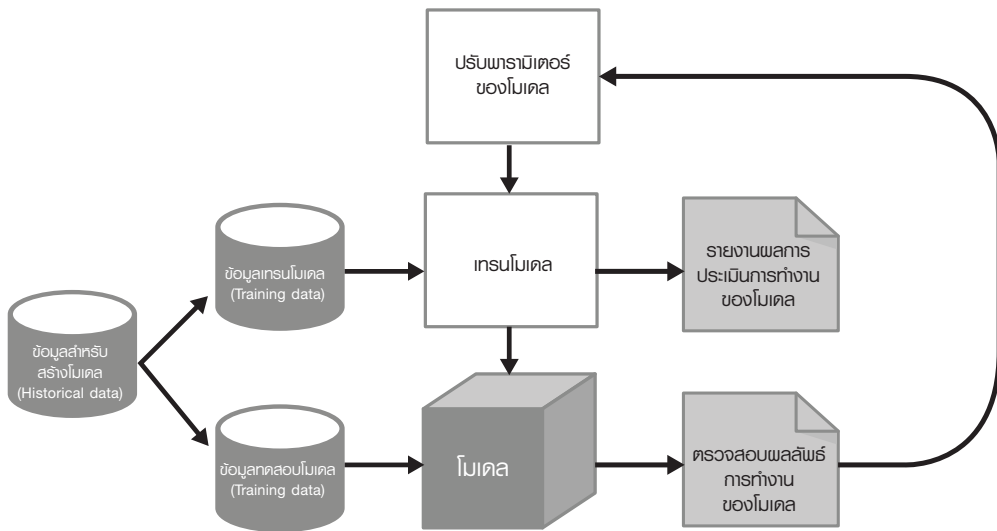
Model

กระบวนการสร้าง Model ของข้อมูลนั้นเสมือนเป็นการนำข้อมูลมา “ทดลอง” ด้วยการใช้ทฤษฎีทางคณิตศาสตร์อย่าง Regression (สมการถดถอย แสดงความสัมพันธ์ระหว่างตัวแปร) และการตั้งสมมติฐานทางสถิติเพื่อหาผลลัพธ์ของข้อมูลมาตอบใจหรือทำนาย (predict) สิ่งที่เกิดขึ้นทางธุรกิจจากข้อมูล โดยนำมาสร้างเป็น Predictive Model ซึ่งในกระบวนการสร้าง Model อาจใช้เทคนิคต่าง ๆ อย่าง Cluster Analysis (การวิเคราะห์กลุ่ม โดยการแบ่งกลุ่มหน่วยข้อมูลที่มีลักษณะที่สนใจเหมือนกันหรือคล้ายกันอยู่กลุ่มเดียวกัน) หรือ Dimensionality Reduction (การลดจำนวนมิติเพื่อบีบอัดข้อมูลเพื่อลดแกมข้อมูลแต่ยังสามารถนำไปทำนายผลลัพธ์ได้) ดังรูป เป็นต้น ซึ่งการวัดความแม่นยำของ Model นั้นขึ้นอยู่กับผลลัพธ์ว่าสามารถตอบใจได้ในมุมมองที่มองไม่เห็น (unseen) จากข้อมูลนั้นได้แค่ไหน



ในกระบวนการนี้คือการทำให้ Machine Learning ซึ่งเราจะนำ Model ที่สร้างขึ้นจากอัลกอริทึม เช่น K-Means มาเทรน (train) จากชุดข้อมูลที่เป็น History เพื่อให้ตัว Model เกิดการเรียนรู้และสามารถจำแนกความแตกต่างภายในกลุ่มข้อมูลได้ ยกตัวอย่างเช่น พฤติกรรมการซื้อสินค้าของกลุ่มผู้ซื้อสินค้าแบบออนไลน์ผ่านแอปพลิเคชันทางมือถือที่มีการค้นหาสินค้าแตกต่างกัน ซึ่งอาจต้องใช้เทคนิคการทำ Cluster Analysis ในการแยกกลุ่ม เป็นต้น เมื่อผ่านการเทรนและได้ Predictive Model ออกมาแล้วนั้น จะมีการตรวจสอบจากข้อมูลอีกชุดว่า Model นั้นให้ผลลัพธ์ที่ถูกต้องและเชื่อถือได้หรือไม่ ซึ่งจะนำผลลัพธ์กลับไปตั้งค่าเป็นพารามิเตอร์เพื่อทำการเทรนครั้งต่อไป ซึ่งจะทำให้ระบบเกิดการเรียนรู้และได้ Predictive Model ที่สามารถทำนายหรือให้ผลลัพธ์ที่ถูกต้องแม่นยำได้

สำหรับในภาษา Python จะมีชุดไลบรารี Scikit-learn สำหรับการทำให้ Machine Learning โดยเฉพาะ หรือหากใช้ภาษา R จะใช้ไลบรารี CARET



Intepret

ขั้นตอนสุดท้ายของกระบวนการ OSEMN ซึ่งถือเป็นขั้นตอนสำคัญที่สุดคือ Intepret ซึ่งขั้นตอนการทำงานคือการสรุปผลลัพธ์หรือ insight ที่ได้จากกราฟหรือการสร้างโมเดลข้อมูล ซึ่งต้องอาศัยการแปลผลเพื่อสรุปให้ผู้บริหารหรือผู้รับข้อมูลเข้าใจในเชิงธุรกิจที่จะนำข้อมูลไปใช้ด้วย

ทั้งนี้ ในการทำทั้ง 5 ขั้นตอนที่ผ่านมา อาจไม่ได้เป็นการเรียงลำดับจาก 0 ถึง N เพียงครั้งเดียว แต่อาจจะมีการย้อนกลับไปทำขั้นตอนก่อนหน้า และอาจมีการวน (iteration) ของการทำ 0 ถึง N หรือขั้นตอนใดขั้นตอนหนึ่งหลายๆ รอบก็ได้

Data Science กับ Python

จากที่กล่าวมา ในการนำข้อมูลมาผ่านกระบวนการวิเคราะห์ข้อมูลเพื่อนำไปประยุกต์ใช้กับธุรกิจนั้นมีความสำคัญเป็นอย่างมาก โดยเฉพาะในการตัดสินใจสำหรับผู้บริหารในเรื่องต่างๆ ในการวางกลยุทธ์ให้สำหรับองค์กร ดังนั้นในกระบวนการทำ Data Science จึงต้องอาศัยการเขียนโปรแกรมเพื่อประมวลผลข้อมูลสำหรับความต้องการทางธุรกิจที่แตกต่างกัน ซึ่งภาษาโปรแกรม (Programming Language) ที่นำมาใช้จะต้องมีความยืดหยุ่นและมีฟังก์ชันการทำงานเพื่อรองรับการประมวลผลทางคณิตศาสตร์ได้ดี ซึ่งภาษาที่นิยมใช้ในปัจจุบันได้แก่ ภาษา Python, R, MATLAB โดยในหนังสือเล่มนี้จะกล่าวถึงการใช้ Python ในกระบวนการ Data Science

ข้อดีของการใช้ Python ทำงาน Data Science สามารถสรุปได้ดังนี้

- เป็นภาษาที่ง่ายต่อการเรียนรู้ สามารถประมวลผลได้โดยเขียนโปรแกรมเพียงแค่นี้ก็บรรทัดเมื่อเทียบกับภาษาอื่นเช่น ภาษา R ซึ่งทำให้เข้าใจได้ง่าย
- เป็นภาษา cross platform ซึ่งสามารถทำงานได้ในทุกระบบปฏิบัติการด้วยโค้ดโปรแกรมชุดเดิม
- ทำงานได้เร็วกว่าภาษาอื่นอย่าง R และ MATLAB
- Python เป็นภาษาที่มีการจัดการทรัพยากรได้ดี ทำให้ใช้หน่วยความจำ (memory) น้อยในการประมวลผล โดยเฉพาะอย่างยิ่งในการต้องทำงานกับข้อมูลจำนวนมากเพื่อแปลงข้อมูล (Data Transformation), การแยกข้อมูลออกเป็นสไลด์และพลิกแกนหรือมิติข้อมูล (Slice and Dice) รวมไปถึงการแสดงผลข้อมูลออกมาเป็นกราฟหรือแผนภูมิ (Visualization)
- Python เป็นภาษาซึ่งมีไลบรารี (Libraries) ต่างๆ มากมายในด้าน Data Science และมีการพัฒนาเวอร์ชันใหม่ๆ เพื่อเพิ่มเติมความสามารถในการประมวลผลออกมาเรื่อยๆ เช่น NumPy, Panda, SciPy เป็นต้น
- Python มีแพ็คเกจซึ่งสามารถเรียกใช้งานโค้ดโปรแกรมของภาษาอื่นเช่น ภาษา C หรือ Java ซึ่งช่วยให้สามารถทำงานร่วมกับการประมวลผลด้วยโปรแกรมอื่นได้

ไลบรารีพื้นฐานสำหรับทำ Data Science ของ Python

Pandas

Pandas เป็นไลบรารีแบบ open source ประสิทธิภาพสูงสำหรับการทำงานกับข้อมูลด้วยการจัดข้อมูลให้อยู่ในโครงสร้าง (Data Structure) ของ Pandas เองคือ Series และ Data Frame

Pandas ถูกนำไปใช้ในงานที่หลากหลาย ทั้งในเชิงการวิเคราะห์ด้านการเงิน เศรษฐศาสตร์ สถิติ รวมไปถึงวิเคราะห์การโฆษณาและประชาสัมพันธ์

การทำงานกับไลบรารี Pandas จะมีขั้นตอนการทำงานที่สำคัญอยู่ 5 ขั้นตอนคือ load, organize, manipulate, model, และ Analyze โดย Pandas เป็นที่นิยมใช้เนื่องจากมีไลบรารีที่รองรับการทำงานในขั้นตอนต่างๆ ได้ดี เช่น ดึงข้อมูล (load) จากไฟล์หลากหลายชนิดมาจัดเตรียม, จัดเรียงข้อมูลหรือจัดการข้อมูลที่หายไป, ปรับรูปแบบข้อมูลหรือทำ Pivot ให้กับชุดข้อมูล, รวมไปถึงการจัดกลุ่ม (Re-grouping) และรวมข้อมูล (Merge and Join) ซึ่งในหนังสือเล่มนี้จะได้กล่าวถึงขั้นตอนต่างๆ ในการทำงานกับ Pandas ต่อไป

NumPy

ย่อมาจาก Numeric Python ถูกสร้างขึ้นในปี ค.ศ. 2005 โดยนักวิทยาศาสตร์ชาวอเมริกันชื่อ Travis Oliphant

NumPy เป็นไลบรารีแบบ open source ที่สามารถใช้งานได้ฟรี การใช้งานจะต้องติดตั้งไลบรารีเพิ่มเติมเนื่องจากไม่ใช่ไลบรารีพื้นฐานของไพธอน แต่ถ้าเราติดตั้งโปรแกรม Anaconda ก็จะมีไลบรารีนี้ติดตั้งมาให้โดยอัตโนมัติ สามารถเรียกใช้งานได้ทันที

ไลบรารีนี้ใช้สำหรับจัดการเกี่ยวกับคณิตศาสตร์และการคำนวณต่างๆ ทั้งข้อมูลเกี่ยวกับวิทยาศาสตร์ วิศวกรรมศาสตร์ สถิติ ธุรกิจ กราฟฟิก ฯลฯ โดยมีความสามารถในการจัดการกับอาร์เรย์หลายมิติ และข้อมูลแบบเมทริกซ์ เช่น เวกเตอร์ (1 มิติ) เมตริกซ์ (2 มิติ) เทนเซอร์ (3 มิติขึ้นไป) เป็นต้น โดยเราจะเรียกอาร์เรย์ใน NumPy ว่า ndarray และเนื่องจาก NumPy จัดเก็บข้อมูลอย่างต่อเนื่องกันบนหน่วยความจำ การเข้าถึงและใช้งานอาร์เรย์ของ NumPy จึงทำได้รวดเร็วกว่าลิสต์มาก อย่างไรก็ตามเนื่องจากอิงอยู่บนพื้นฐานด้านอาร์เรย์และ operation ของมัน ดังนั้นการใช้งาน Numpy ต้องมีความรู้พื้นฐาน Linear algebra, Matrix และ Vector ด้วย

NumPy มักใช้คู่กับ SciPy (Scientific Python) และ Matplotlib (plotting library) โดยนำมาทำงานร่วมกันเพื่อใช้แทน MATLAB

SciPy

SciPy เป็นไลบรารีสำหรับทำงานกับ ndarray ของ NumPy โดยประกอบไปด้วยแพ็คเกจย่อยต่าง ๆ ซึ่งใช้ในการคำนวณตามทฤษฎีทางคณิตศาสตร์และฟิสิกส์ เช่น แคลคูลัส (Calculus), พีชคณิตเชิงเส้น (Linear Algebra), อนุกรม Fourier (Fourier Series), หรือค่าคงที่ทางฟิสิกส์และคณิตศาสตร์ เป็นต้น โครงสร้างข้อมูลของ SciPy จะใช้พื้นฐาน ndarray ของ NumPy แต่จะเป็น array แบบหลายมิติ

Matplotlib

Matplotlib เป็นไลบรารีที่ใช้สำหรับการทำ Data Visualization โดยมันจะมีฟังก์ชันเกี่ยวกับการวาดกราฟต่าง ๆ ให้เรียกใช้งาน เช่น กราฟแท่ง กราฟวงกลม เป็นต้น สามารถวาดกราฟแบบ 2 มิติ หรือ 3 มิติได้ โมดูลที่สำคัญของไลบรารีนี้คือ “pyplot” ซึ่งเป็นโมดูลหลักของการกำหนดรูปแบบของเส้นกราฟ ฟอนต์ รูปแบบแกนของกราฟ และลูกเล่นอื่นๆ อีกมากมาย

ไลบรารีนี้ใช้ร่วมกับ NumPy ในการทำงานแทนการใช้โปรแกรม MATLAB นอกจากนี้ยังสามารถใช้ร่วมกับไลบรารีอื่นที่เกี่ยวข้องกับการวาดกราฟฟีกอย่าง PyQt และ wxPython ได้อีกด้วย

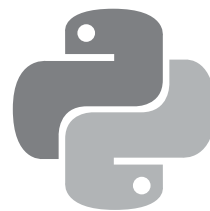
Seaborn

Seaborn เป็นไลบรารีสำหรับการพล็อตกราฟในขั้นตอนการทำ Data Visualization ซึ่งเรียกใช้ไลบรารี Matplotlib ในการทำงานอีกทีหนึ่ง ไลบรารีนี้มีประโยชน์ในการแสดงกราฟค่าการกระจายตัวของข้อมูล (Random Distributions) เนื่องจากมีฟังก์ชันโดยเฉพาะสำหรับการรับข้อมูลแบบอาร์เรย์ไปแสดงการกระจายตัวของชุดข้อมูลในเชิงสถิติ ทั้งการแสดงกราฟแบบระฆังคว่ำ, ปัวซอง (Poisson), ไคสแควร์ (Chi Square) และอื่นๆ รวมไปถึงยังสามารถแสดงได้ทั้งแบบกราฟเส้นหรือกราฟแท่ง



CHAPTER

02



การใช้งาน
Jupyter Notebook
และ Google Colab

ทำความรู้จักกับ Jupyter Notebook

การเขียนโปรแกรมในปัจจุบันส่วนใหญ่เรามักจะเขียนโปรแกรมผ่าน IDE (Integrated Development Environment) ต่าง ๆ มากกว่าการเขียนโปรแกรมด้วย text editor ธรรมดา เนื่องจาก IDE เป็นเครื่องมือพัฒนาโปรแกรมที่ช่วยให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้สะดวกและง่ายขึ้น

นอกจาก IDE จะประกอบด้วย source code editor สำหรับเขียนโปรแกรมแล้ว มันยังรวมตัวแปลภาษา และ debugger ที่ช่วยตรวจสอบข้อผิดพลาดของโปรแกรมไว้ในตัวมันด้วย อีกทั้ง IDE บางตัวยังประกอบด้วยไลบรารีที่จำเป็นสำหรับการเขียนโปรแกรมภาษาต่าง ๆ ด้วย การเขียนโปรแกรมในแต่ละภาษาจึงจำเป็นต้องเลือก IDE ที่เหมาะสมกับการเขียนโปรแกรมภาษานั้น ๆ

Jupyter Notebook เป็น IDE ตัวหนึ่งที่นิยมนำมาใช้ในการเขียนโปรแกรมของ Data Science ซึ่ง Jupyter notebook เป็น open-source web application ที่ทำงานบนเว็บเบราว์เซอร์ในลักษณะโครงสร้างแบบ client-server application

การใช้งาน Jupyter Notebook ในหนังสือเล่มนี้จะสอนติดตั้งโปรแกรม Anaconda เนื่องจาก Anaconda ติดตั้งง่าย ใช้งานไม่ยาก มีไลบรารีการทำงานต่าง ๆ ค่อนข้างครบสำหรับการเขียนโปรแกรม Data Science เช่น ไลบรารี NumPy, Pandas, Matplotlib, Seaborn เป็นต้น โดยที่เราไม่ต้องติดตั้งไลบรารีเพิ่มเติมด้วยตัวเองให้ยุ่งยาก และที่สำคัญ Anaconda มี Jupyter Notebook ผูกติดมากับมันด้วย เรียกได้ว่าลง Anaconda ตัวเดียวก็พร้อมสำหรับการเขียนโปรแกรม Data Science เลยค่ะ

โดยปกติเมื่อติดตั้ง Anaconda แล้ว Jupyter Notebook server จะถูกสร้างขึ้นมาบนเครื่องของเราโดยอัตโนมัติ ดังนั้น การเรียกใช้งาน Jupyter Notebook จึงไม่จำเป็นต้องเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ต เมื่อเราสั่งเรียก Jupyter notebook ให้ทำงาน นอกจาก server จะถูกสตาร์ทให้ทำงานแล้ว มันจะเปิด Jupyter Notebook client บนเว็บเบราว์เซอร์ให้อัตโนมัติด้วย เราสามารถเขียนโปรแกรมต่าง ๆ ของงาน Data Science ลงบน Jupyter Notebook client และสั่งรันเพื่อดูผลลัพธ์ได้

Jupyter Notebook กับ Google Colab ต่างกันอย่างไร

Jupyter Notebook เป็น IDE ที่ใช้สำหรับการเขียนโปรแกรมด้าน Data Science ซึ่งเราสามารถให้ Jupyter Notebook เขียนโปรแกรมทำ Data cleansing, data visualization, machine learning และการทำงานอื่น ๆ ได้ การใช้งาน Jupyter Notebook จำเป็นต้องติดตั้งโปรแกรม Anaconda ลงที่เครื่องคอมพิวเตอร์ของเราก่อน และหลังจากนั้นการเรียกใช้งานก็ไม่จำเป็นต้องเชื่อมต่อใด ๆ กับระบบเครือข่ายอินเทอร์เน็ตอีก

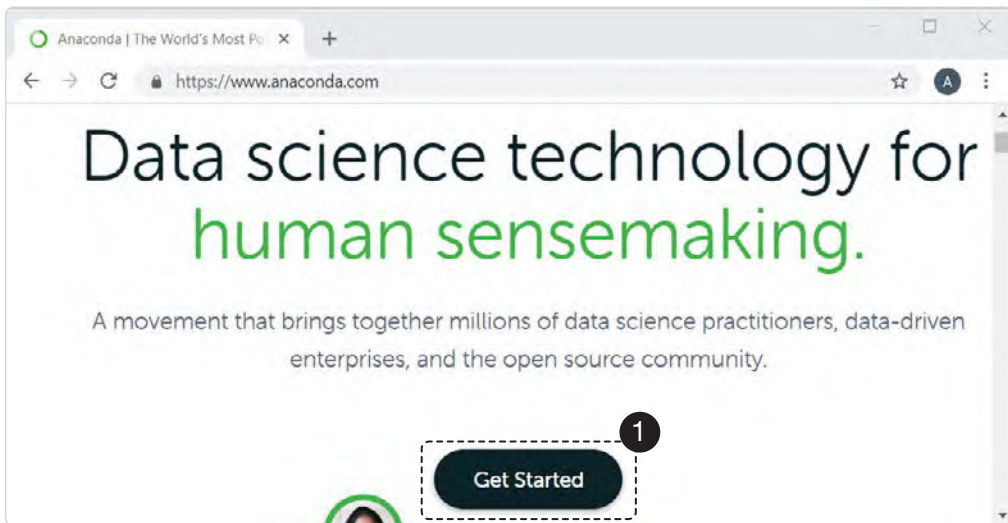
Google Colab (ย่อมาจาก Google Colaboratory) เป็นโปรเจกต์ที่เกิดจากความร่วมมือของทีมงาน Google และทีมงานผู้สร้าง Jupyter Notebook ที่พัฒนา Jupyter Notebook ให้ทำงานอยู่บน Cloud ดังนั้น การเรียกใช้งาน Google Colab จึงไม่จำเป็นต้องติดตั้งโปรแกรมใดๆ ลงบนเครื่องคอมพิวเตอร์ก่อนใช้งาน แต่การใช้งานจะต้องใช้งานผ่านเว็บเบราว์เซอร์ และต้องเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ตทุกครั้งที่ใช้งาน

การใช้งาน Google Colab ทุกคนสามารถใช้งานได้ฟรี แต่เพียงมีบัญชี Google Drive ก็สามารภใช้งานได้แล้ว เมื่อเขียนโปรแกรมเสร็จก็สามารถจัดเก็บโค้ดโปรแกรมไว้บน google drive ของตัวเองได้ และสามารถแชร์โค้ดต่างๆ ให้กับผู้อื่นได้อีกด้วย

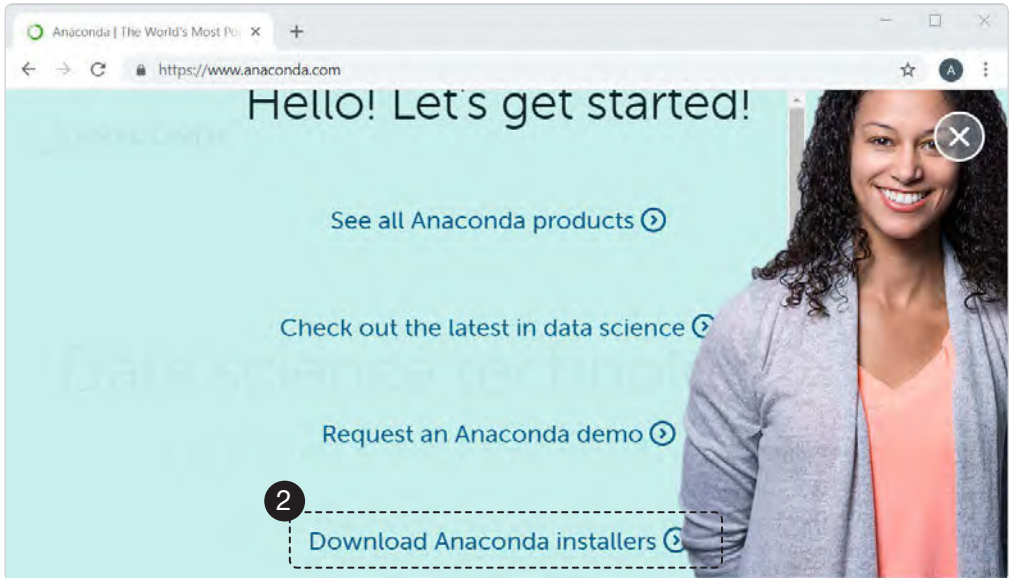
การติดตั้งโปรแกรม Anaconda

การติดตั้งโปรแกรม Anaconda มีขั้นตอนดังนี้

1. เปิดเว็บเบราว์เซอร์ไปที่ <http://www.anaconda.com> แล้วคลิกเลือก **Get Started**



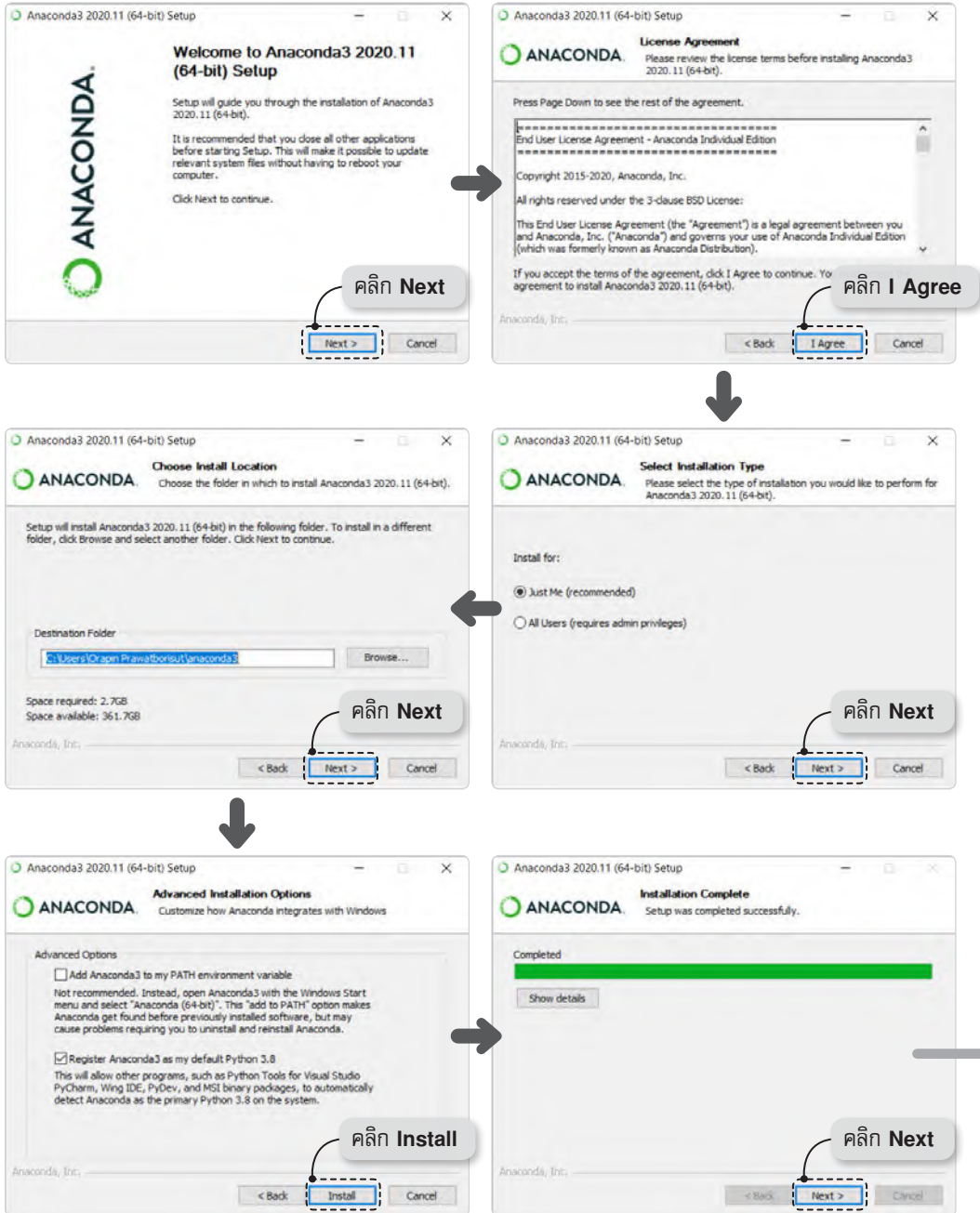
2. คลิกที่ Download Anaconda Installers

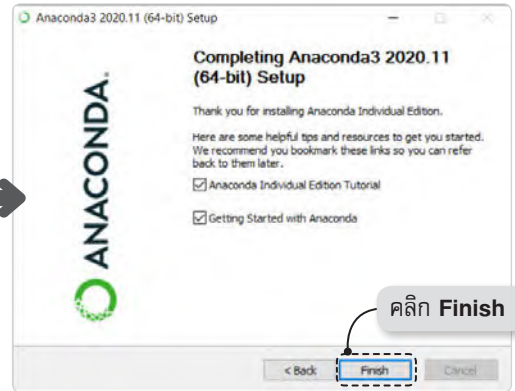
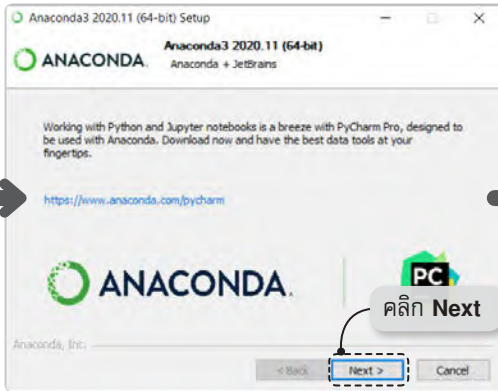


3. เลื่อนไปที่ระบบปฏิบัติการ **Windows** แล้วคลิกเลือก **64-Bit Graphical Installer (457 MB)** แต่ถ้าระบบปฏิบัติการที่เครื่องคอมพิวเตอร์ของผู้อ่านเป็นแบบ 32 บิต ก็ให้คลิกที่ 32 Bit Graphical Installer (403 MB) จากนั้นรออนจนกระทั่งดาวน์โหลดไฟล์สำเร็จ



4. ดับเบิลคลิกที่ไฟล์ที่ดาวน์โหลดมาเพื่อเริ่มการติดตั้ง โดยการติดตั้งให้ทำตามขั้นตอนดังรูป

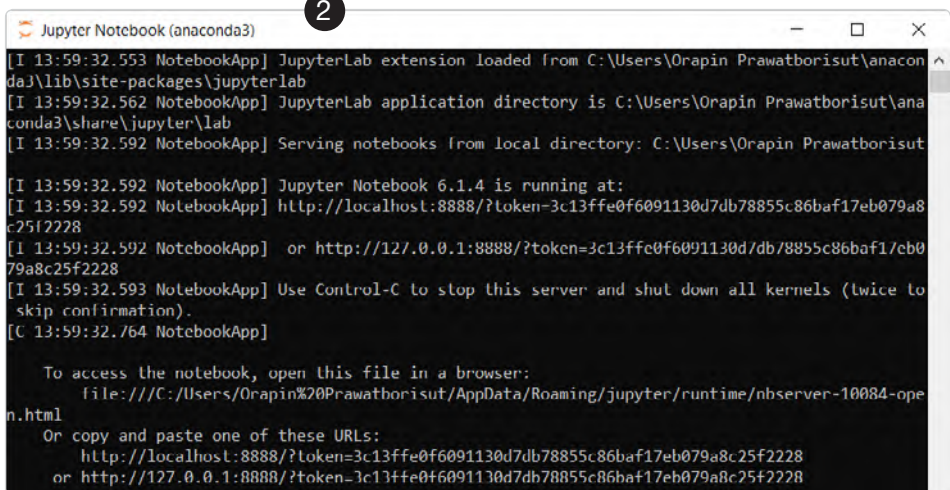
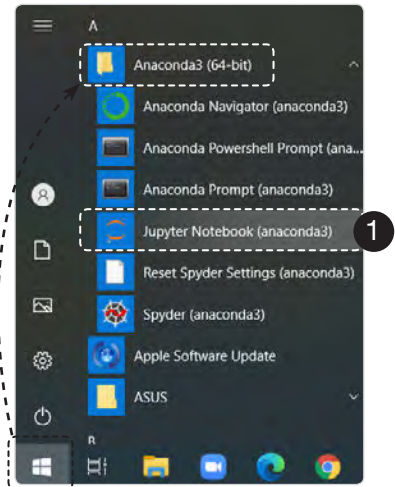




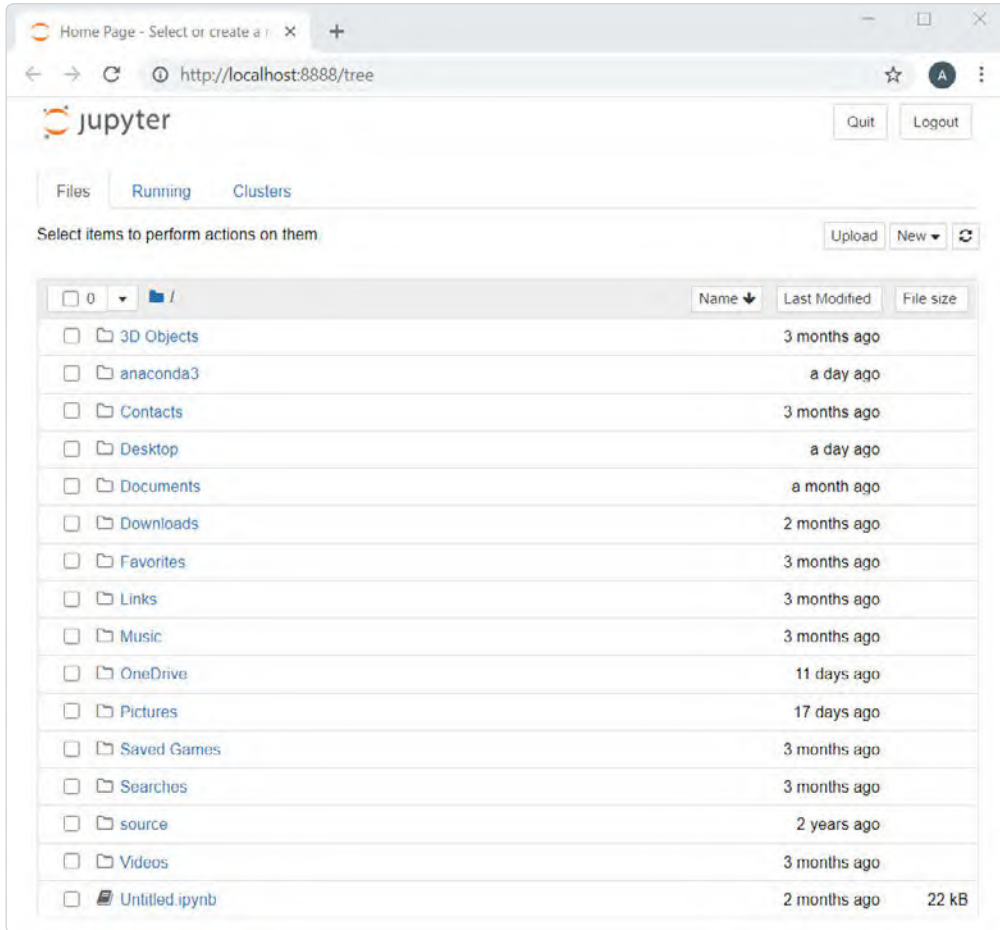
การเรียกใช้งาน Jupyter Notebook

การเรียกใช้งาน Jupyter Notebook มีขั้นตอนดังนี้

1. คลิกปุ่ม **Start** แล้วเลือกที่ **Anaconda3 (64-bit)** จากนั้นคลิกเลือก **Jupyter Notebook (anaconda3)**
2. Jupyter Notebook Server จะถูกสตาร์ทขึ้นมา ดังรูป



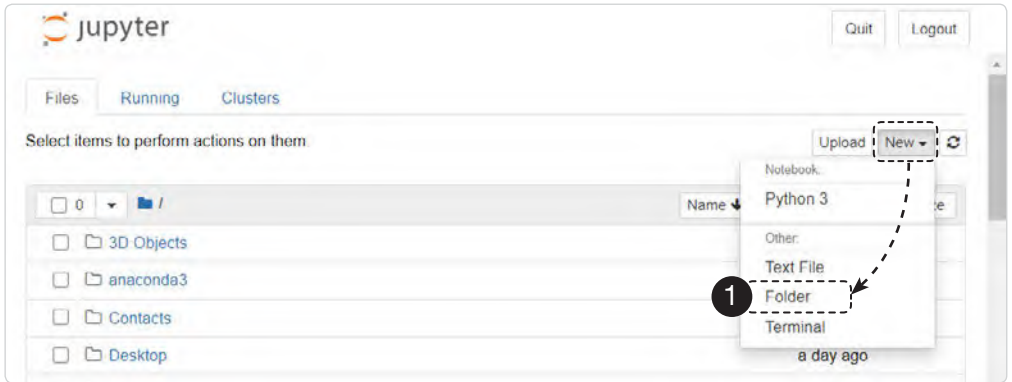
3. เมื่อ Jupyter Notebook Server สตาร์ทเสร็จเรียบร้อยแล้ว Jupyter Notebook Client ก็จะเปิดขึ้นมาบนเว็บเบราว์เซอร์โดยอัตโนมัติ ดังรูป



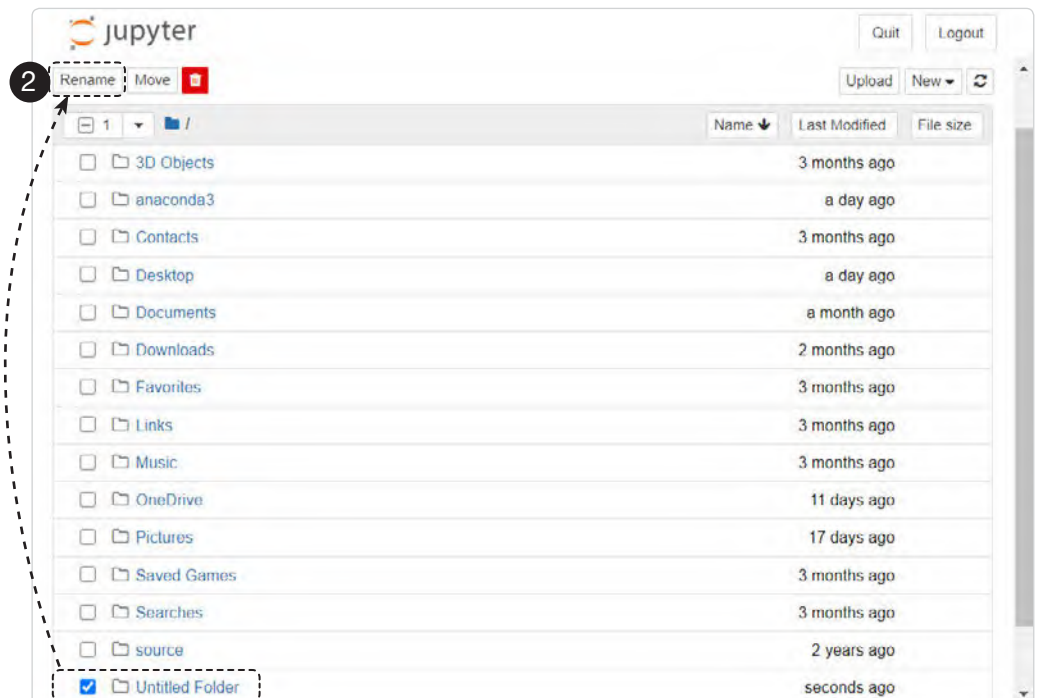
การสร้าง folder เพื่อจัดเก็บโค้ดโปรแกรม

เราสามารถสร้าง folder ใหม่ขึ้นมาใน Jupyter Notebook เพื่อจัดเก็บโค้ดโปรแกรมของเราได้ดังขั้นตอนต่อไปนี้

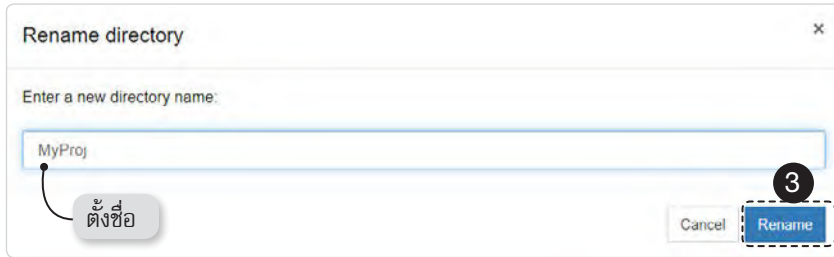
1. คลิกที่ **New** แล้วเลือก **Folder**



2. โฟลเดอร์ใหม่จะถูกสร้างขึ้นมาด้วยชื่อ **Untitled Folder** ให้คลิกเลือก ที่โฟลเดอร์นั้น แล้วคลิกปุ่ม **Rename** เพื่อเปลี่ยนชื่อโฟลเดอร์

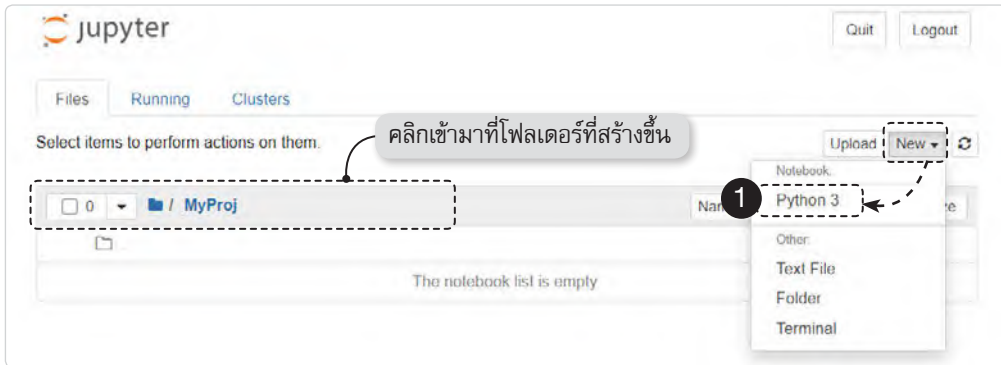


3. กำหนดชื่อโฟลเดอร์ที่ต้องการ และคลิกปุ่ม **Rename**

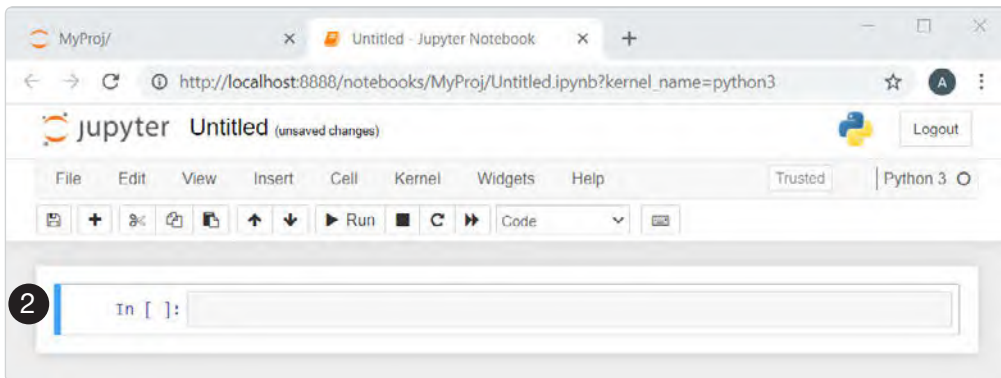


การเขียนโปรแกรมลงบน Jupyter Notebook

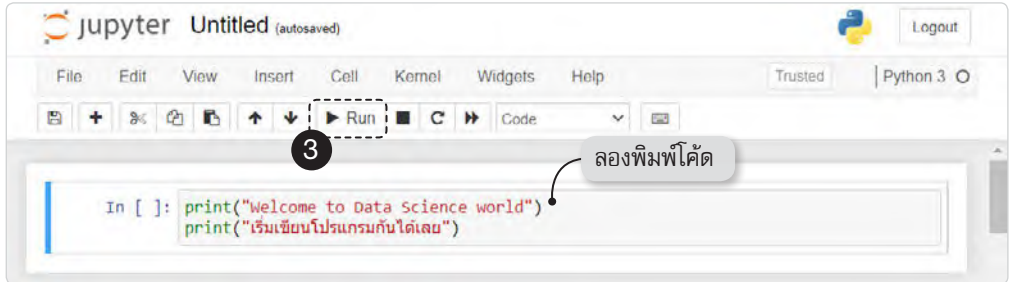
1. หากเราจะเก็บโค้ดโปรแกรมไว้ที่โฟลเดอร์ใด ให้คลิกเข้าไปที่โฟลเดอร์นั้นก่อน (ในที่นี้คลิกเข้าไปที่โฟลเดอร์ MyProj ที่สร้างขึ้นมา) จากนั้นคลิกที่ **New** แล้วเลือก **Python3**



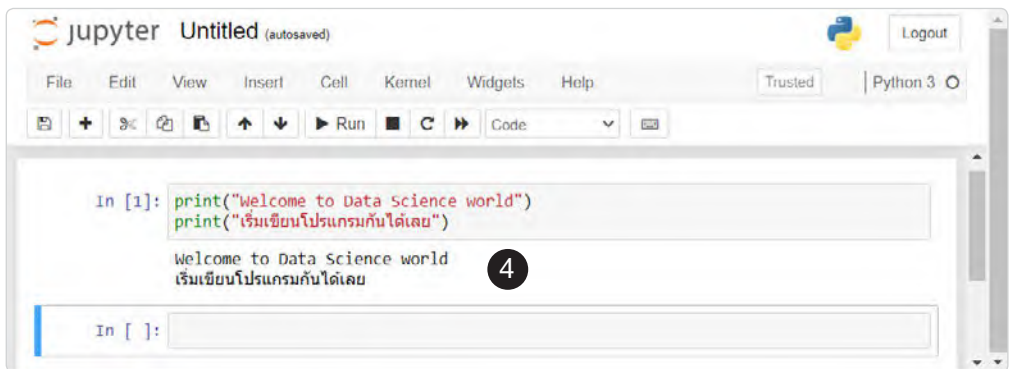
2. จะมีหน้าต่างแสดงขึ้นมาดังรูป เราสามารถเขียนโค้ดโปรแกรมลงไปที่ช่องเซลล์ที่ชื่อว่า `In []:` ได้



3. ให้ลองพิมพ์โค้ดโปรแกรมดังรูป จากนั้นรันเซลล์ปัจจุบันที่เลือกโดยคลิกปุ่ม **Run** (หรือกดคีย์ **Ctrl + Enter** ก็ได้) เพื่อรันดูผลลัพธ์



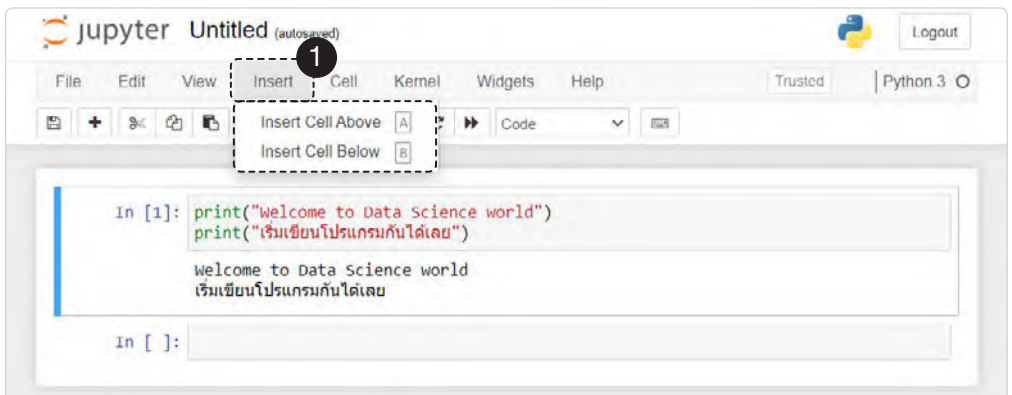
4. เมื่อสั่งรันโปรแกรมแล้ว จะมีผลลัพธ์แสดงออกมา ดังรูป



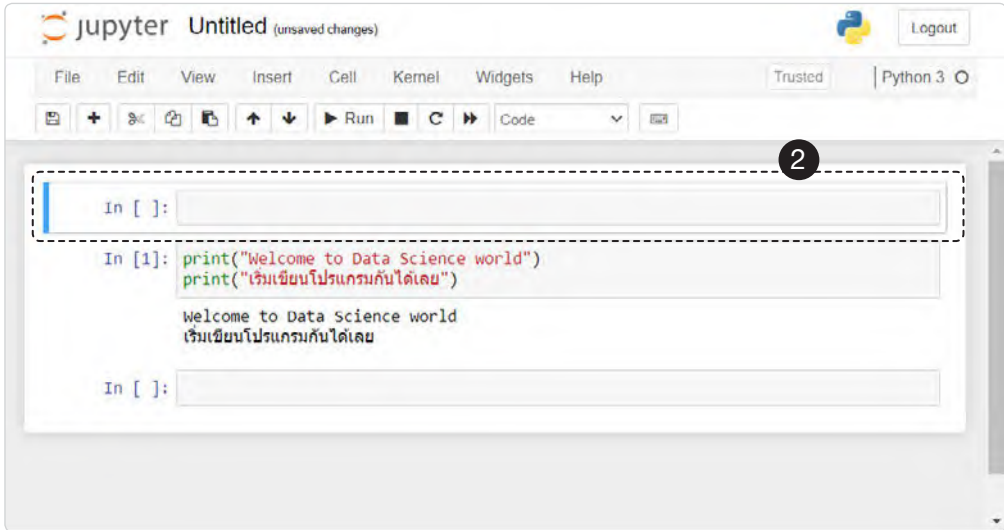
จะพบว่าเมื่อเรารันโค้ดโปรแกรมหนึ่งๆ แล้ว จะพบว่ามีเซลล์ In []: ใหม่แสดงขึ้นมา เราสามารถเขียนโค้ดโปรแกรมอื่นๆ ลงในแต่ละเซลล์เพื่อดูผลลัพธ์ได้ เซลล์แต่ละเซลล์จะแยกการทำงานออกจากกัน

การลบและเพิ่มเซลล์

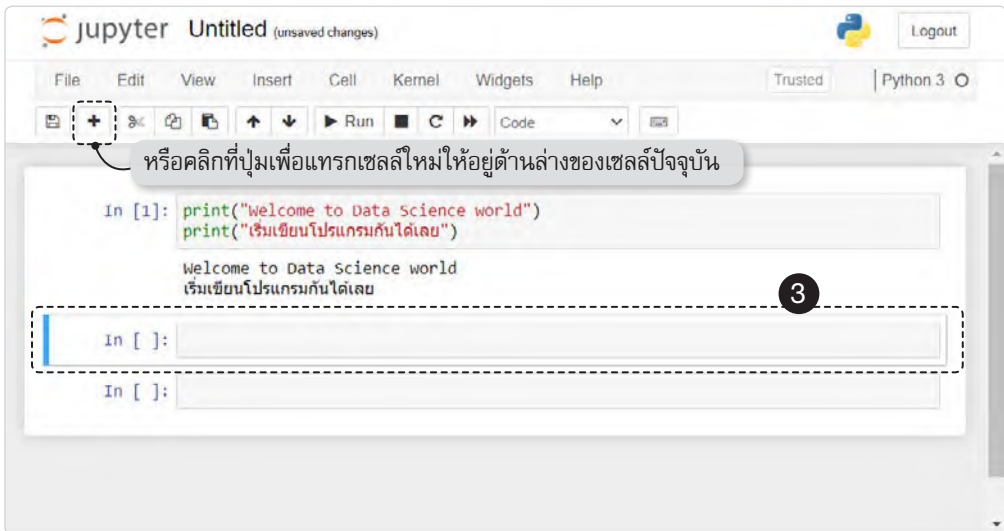
1. ในกรณีที่ต้องการเพิ่มเซลล์ ให้คลิกที่เมนู **Insert**



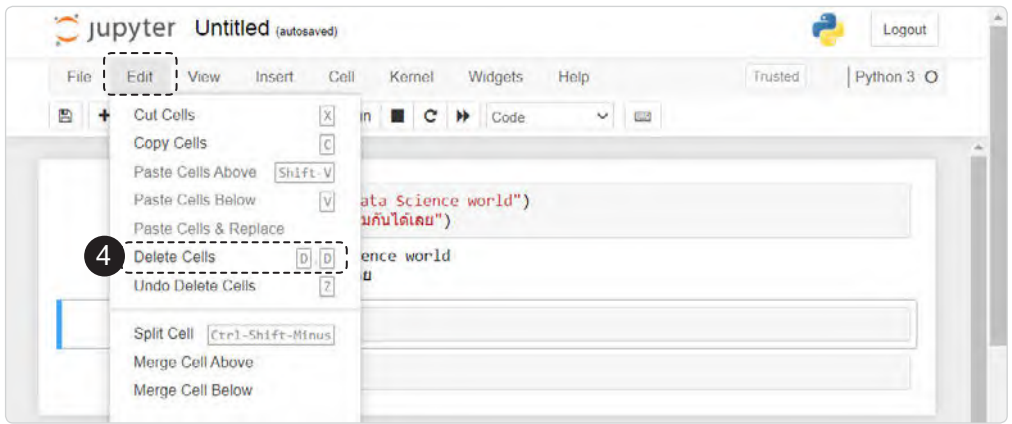
2. หากต้องการเพิ่มเซลล์ใหม่ให้อยู่ด้านบนของเซลล์ปัจจุบันที่ทำงานอยู่ก็เลือกที่ **Insert Cell Above** จะได้ผลลัพธ์ดังรูป




3. แต่ถ้าต้องการให้เซลล์ใหม่อยู่ด้านล่างของเซลล์ปัจจุบันที่ทำงานอยู่ก็ให้เลือกที่ **Insert Cell Below** หรือจะคลิกปุ่ม **+** ก็ได้ จะได้ผลลัพธ์ดังรูป

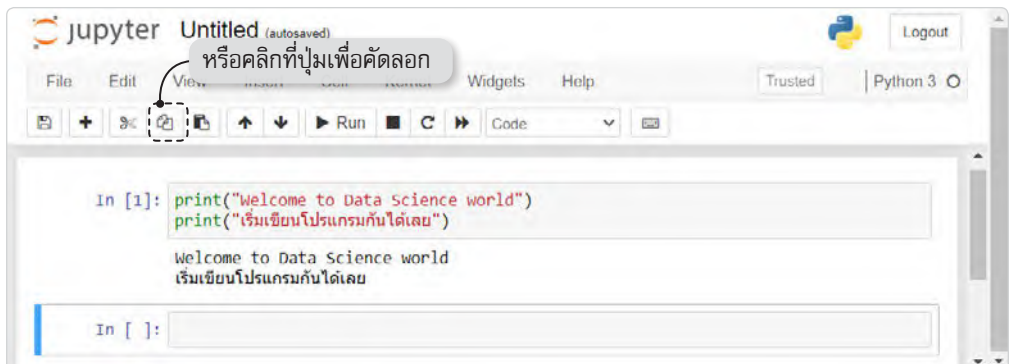
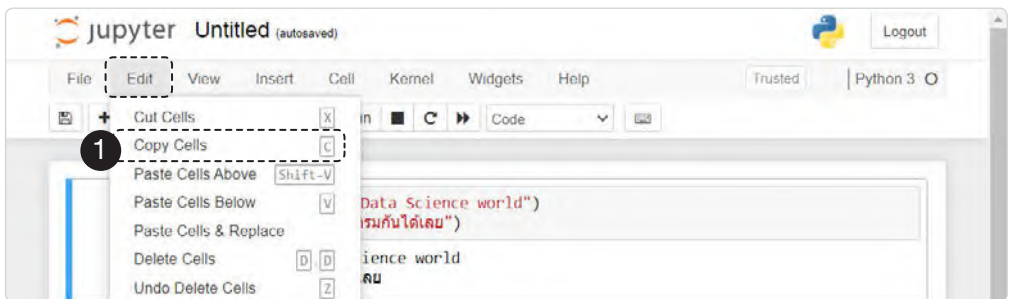


4. กรณีที่ต้องการลบเซลล์ ให้คลิกเลือกที่เซลล์ที่ต้องการลบ จากนั้นไปที่เมนู **Edit** และเลือก **Delete Cells** ดังรูป

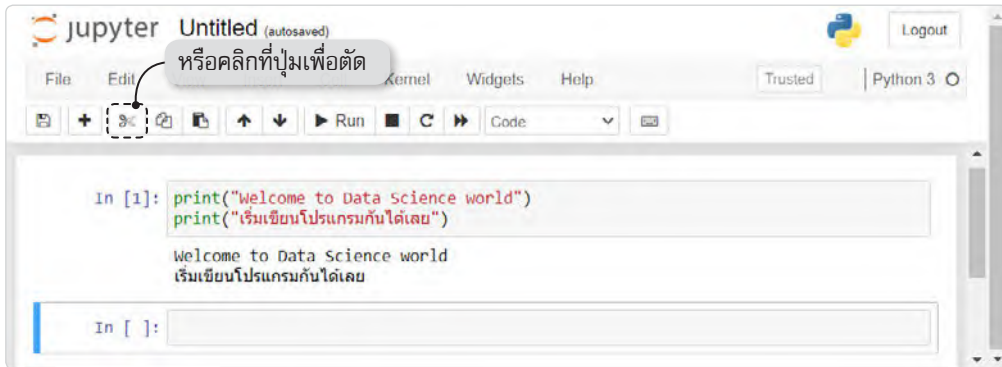
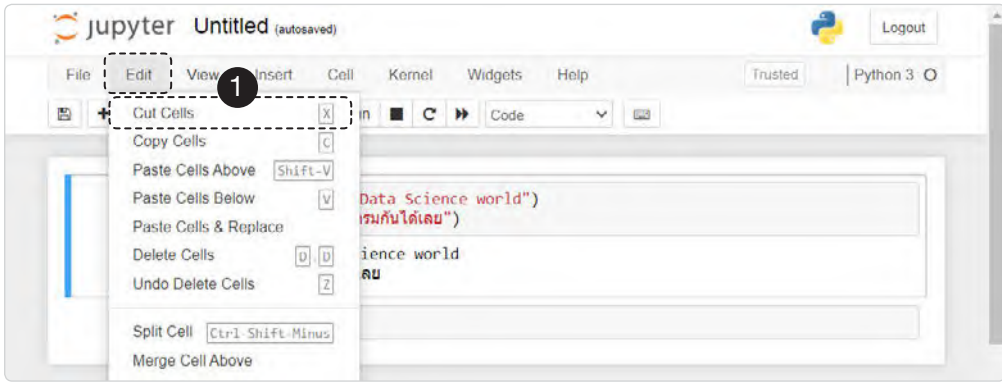


การตัดลอก/ตัดเซลล์เดิมไปไว้ยังเซลล์ใหม่

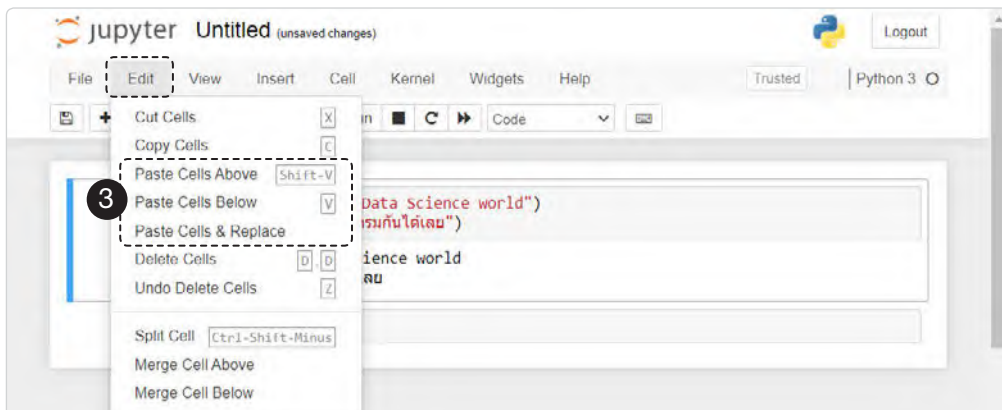
1. การตัดลอกเซลล์ ให้คลิกเลือกเซลล์ต้นทางที่ต้องการตัดลอก จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Copy Cells** หรือจะคลิกปุ่ม  ก็ได้



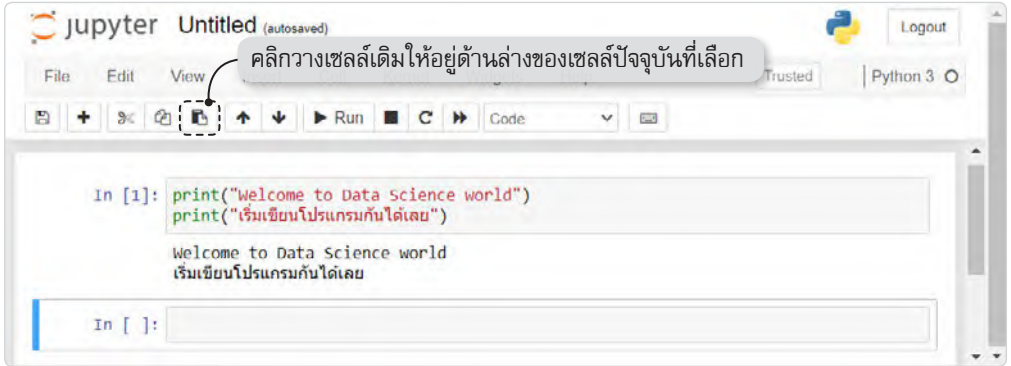
2. การตัดเซลล์ ให้คลิกเลือกเซลล์ที่ต้องการตัด จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Cut Cells** หรือจะคลิกปุ่ม  ก็ได้



3. เมื่อคัดลอกหรือตัดเซลล์เดิมแล้ว หากต้องการนำเซลล์ไปวางไว้ยังเซลล์ใหม่ ให้ไปที่เมนู **Edit** แล้วคลิกที่ **Paste** ซึ่งแบ่งออกได้เป็น 3 ประเภท คือ





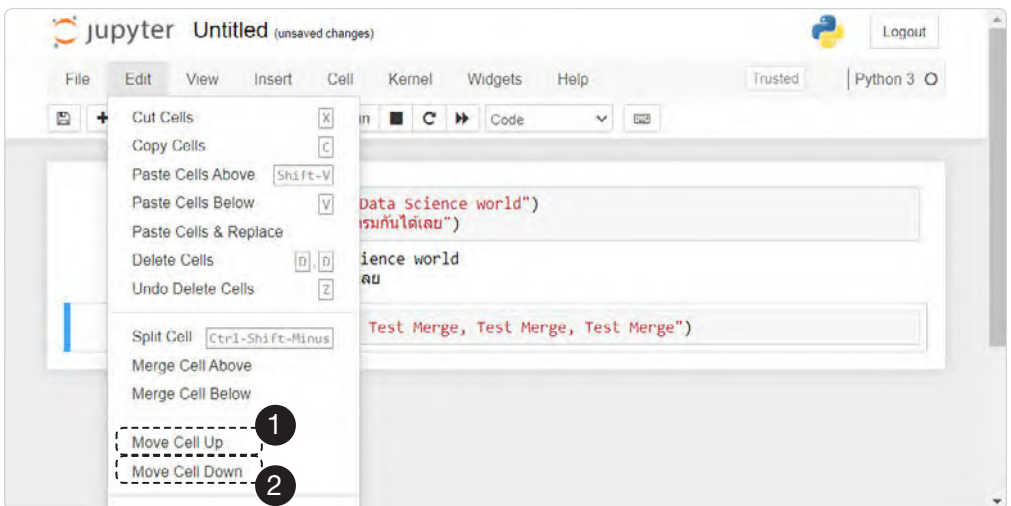
- **Paste Cells Above** วางเซลล์เดิมที่คัดลอกหรือตัดมา ให้อยู่ด้านบนของเซลล์ปัจจุบันที่เลือก
- **Paste Cells Below** วางเซลล์เดิมที่คัดลอกหรือตัดมา ให้อยู่ด้านล่างของเซลล์ปัจจุบันที่เลือก (กรณีนี้สามารถคลิกปุ่ม  เพื่อทำการ Paste Cells Below ได้เช่นกัน ดังรูป)

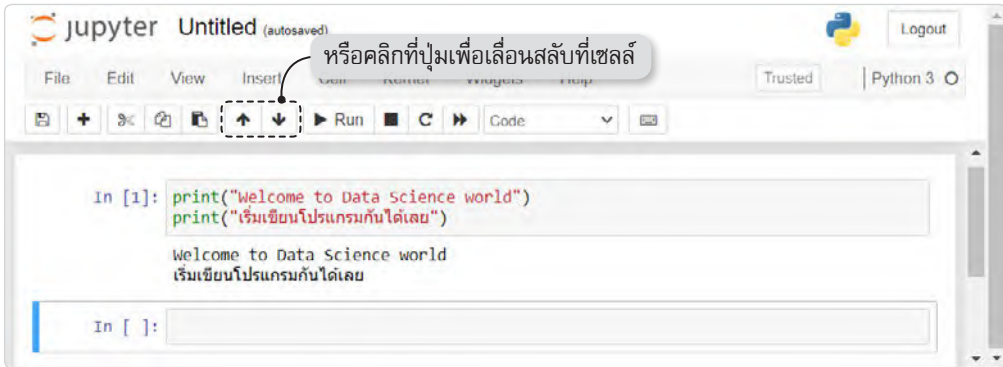


- **Paste Cells & Replace** วางเซลล์เดิมที่คัดลอกหรือตัดมา แทนที่เซลล์ปัจจุบันที่เลือก

การสลับที่เซลล์

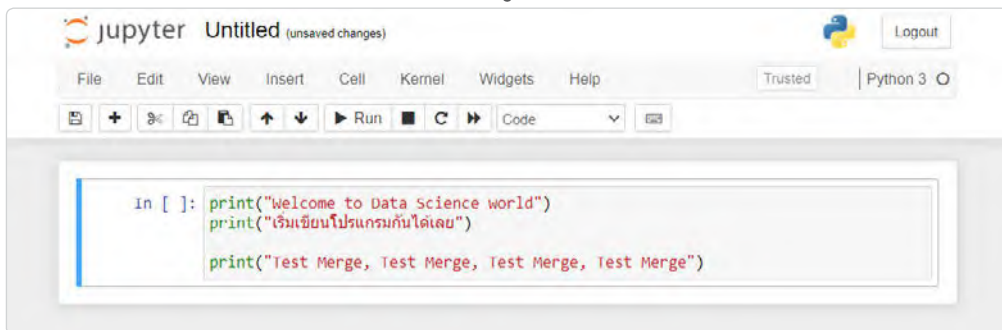
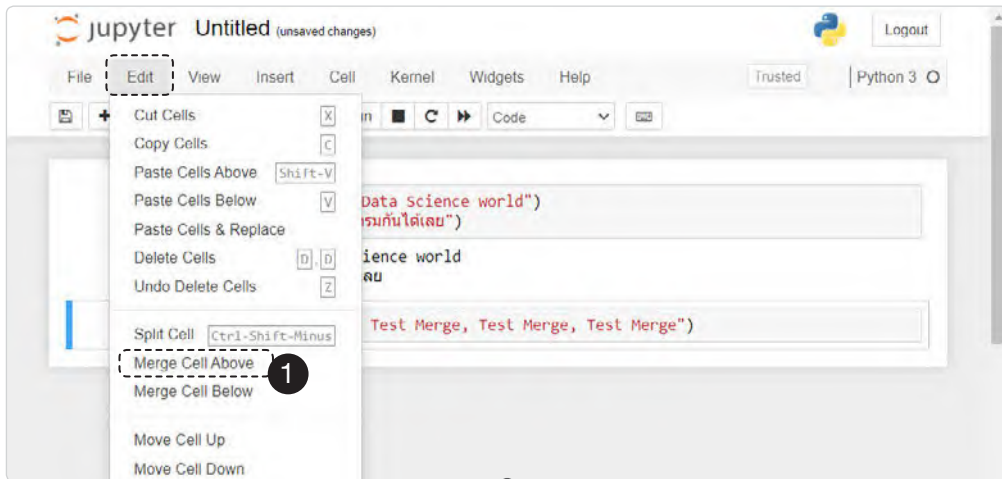
1. การเลื่อนสลับที่เซลล์ปัจจุบันที่เลือกให้ขึ้นไปอยู่ด้านบน ทำได้โดยไปที่เมนู **Edit** แล้วคลิกที่ **Move Cell Up** หรือจะคลิกปุ่ม  ก็ได้
2. การเลื่อนสลับที่เซลล์ปัจจุบันที่เลือกให้ลงไปอยู่ด้านล่าง ทำได้โดยไปที่เมนู **Edit** แล้วคลิกที่ **Move Cell Down** หรือจะคลิกปุ่ม  ก็ได้



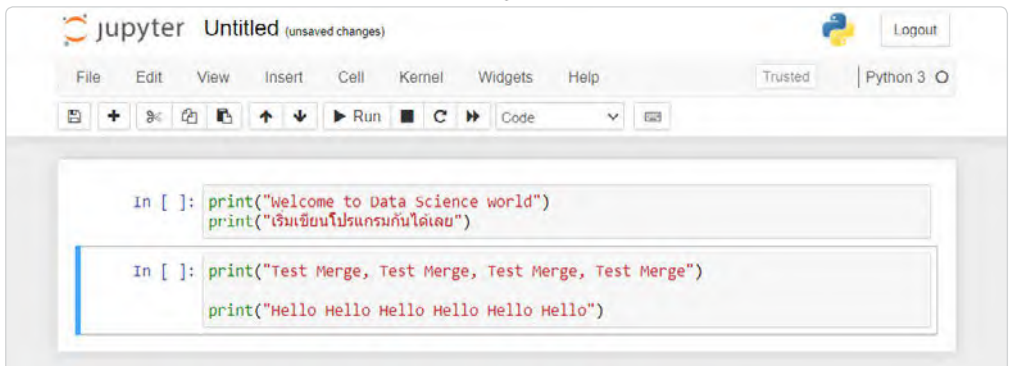
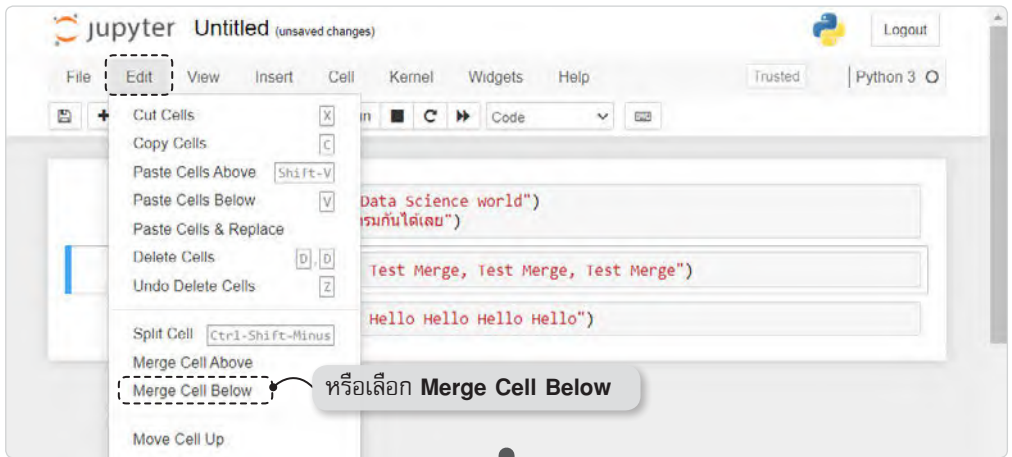


การรวมเซลล์และแตกเซลล์

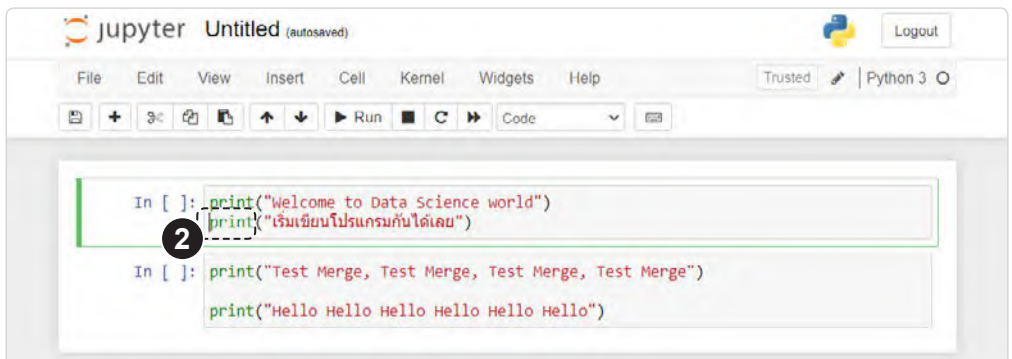
1. การรวมเซลล์ ทำได้โดยคลิกเลือกเซลล์ที่ต้องการ จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Merge Cell Above** แล้วเซลล์ปัจจุบันที่เลือกจะถูกรวมกับเซลล์ที่อยู่ด้านบนเป็นเซลล์เดียวกัน ดังรูป



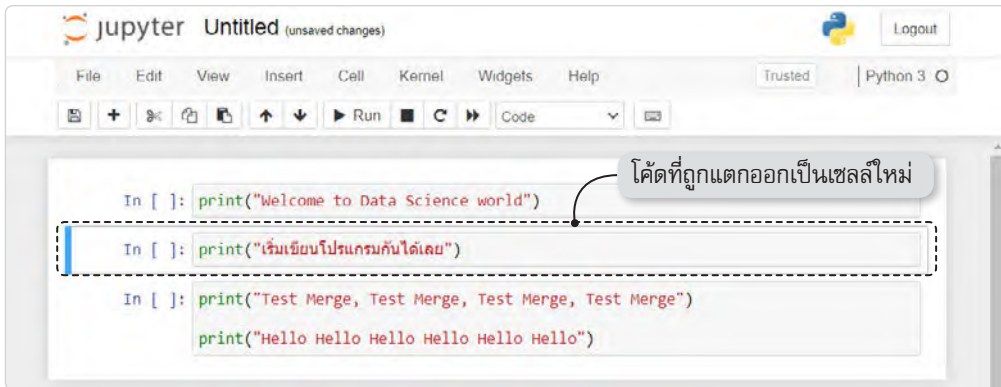
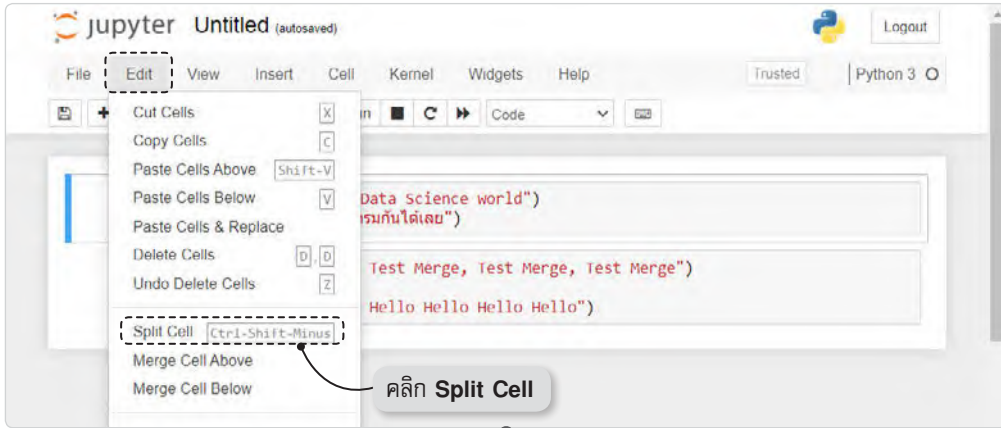
- หรือจะคลิกที่ **Merge Cell Below** ก็ได้ จะมีผลทำให้เซลล์ปัจจุบันที่เลือกจะถูกรวมกับเซลล์ที่อยู่ด้านล่างเป็นเซลล์เดียวกัน ดังรูป



2. การแตกเซลล์ ทำได้โดยคลิกเลือกเซลล์ที่ต้องการ และนำเคอร์เซอร์ไปวางที่จุดแรกของโค้ดที่ต้องการแตกเซลล์ออกไป ดังรูป

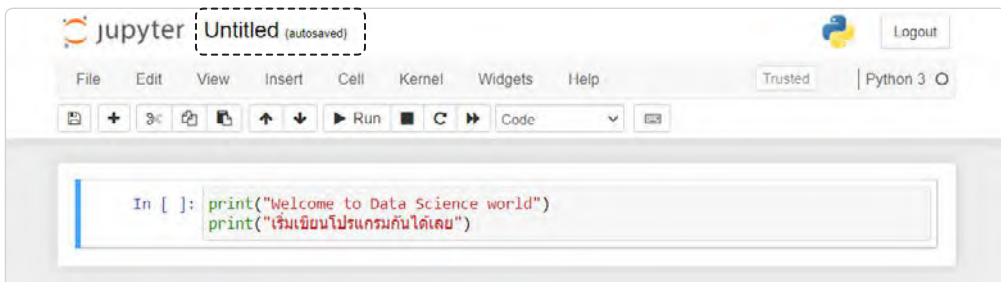


จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Split Cell** แล้วโค้ดของเซลล์ปัจจุบันที่เลือกจะถูกแยกออกไปเป็นเซลล์ใหม่ ดังรูป



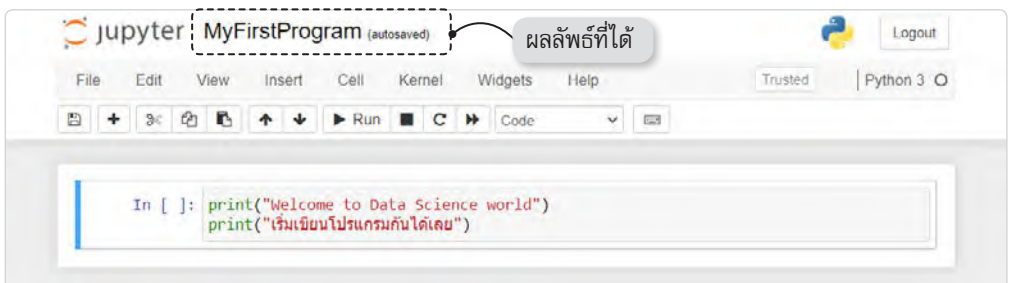
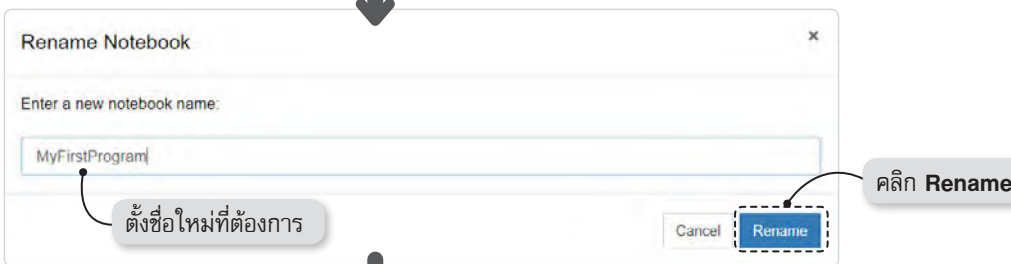
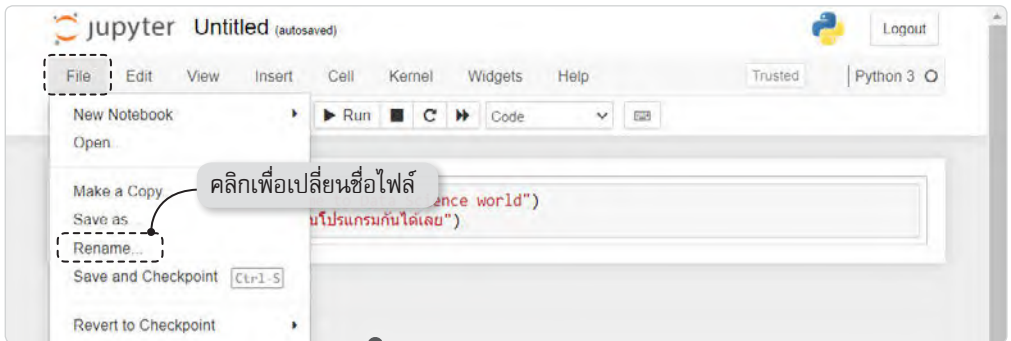
การบันทึกโค้ดโปรแกรมใน Jupyter Notebook

โดยปกติโค้ดโปรแกรมที่เราเขียนจะถูกบันทึกโดยอัตโนมัติ (autosaved) ซึ่งในกรณีนี้โค้ดโปรแกรมจะถูกบันทึกด้วยชื่อ Untitled.ipynb ดังรูป

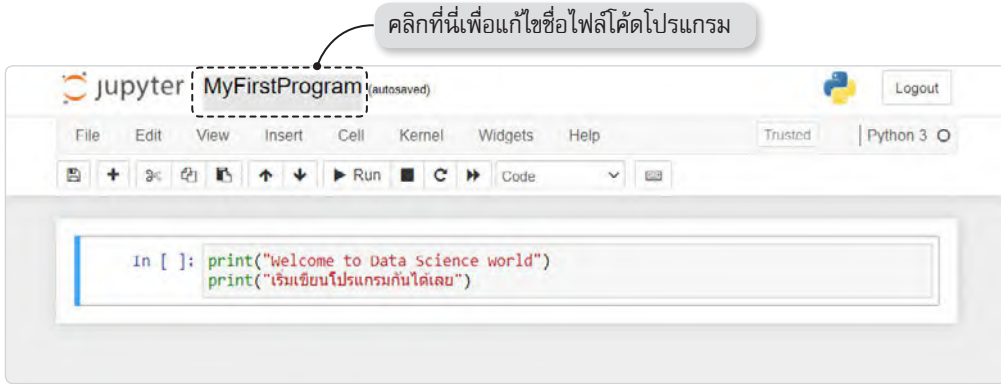




แต่ถ้าเราต้องการบันทึกไฟล์ด้วยชื่ออื่นก็สามารถทำได้ โดยไปที่เมนู **File** แล้วคลิกที่ **Rename...**
 ดังรูป

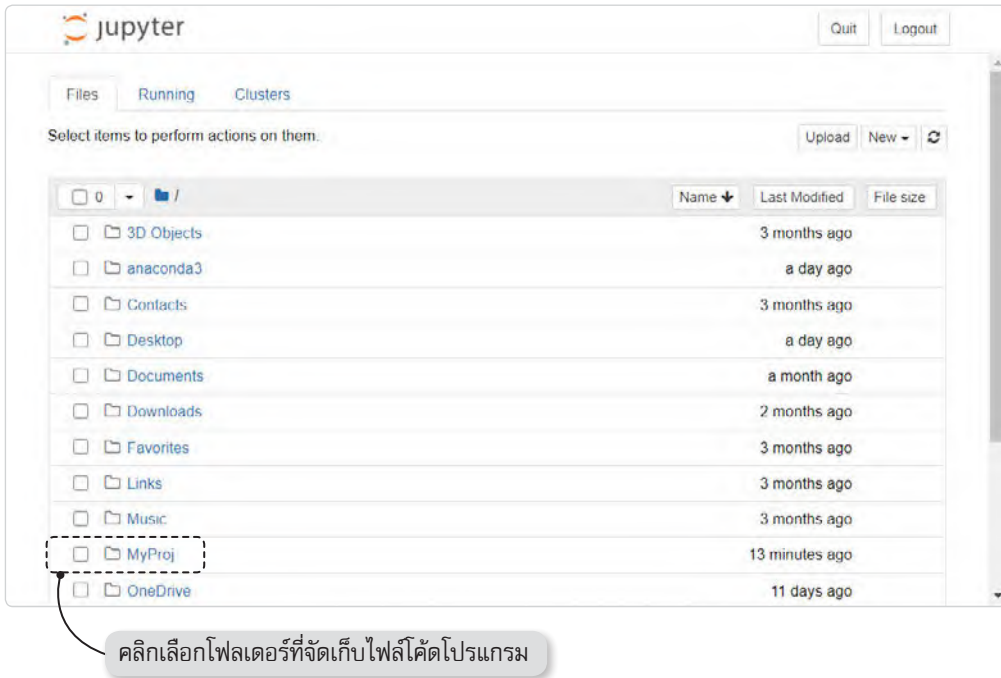


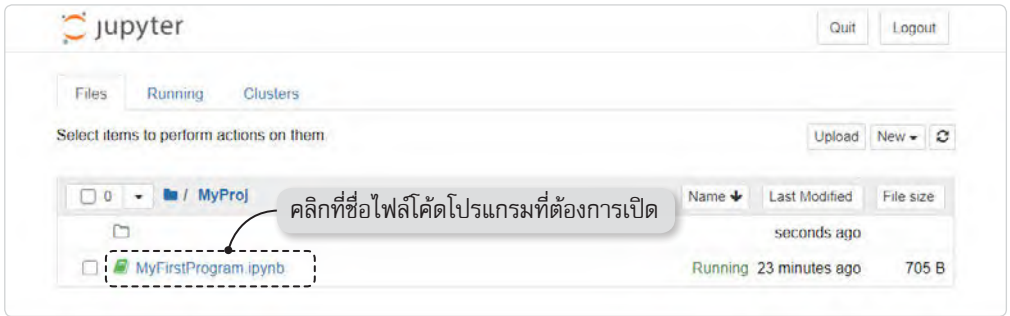
การ Rename ชื่อไฟล์สามารถทำได้อีกวิธีหนึ่ง คือ นำเมาส์ไปคลิกที่ชื่อไฟล์โค้ดโปรแกรมเพื่อแก้ไขชื่อไฟล์ ดังรูป



การเปิดไฟล์โค้ดโปรแกรมใน Jupyter Notebook

การเปิดไฟล์โค้ดโปรแกรมทำได้โดยไปที่หน้าจอของ Jupyter Notebook Client แล้วเลือกไปยังโฟลเดอร์ที่จัดเก็บไฟล์โค้ดโปรแกรมที่ต้องการ จากนั้นคลิกไปที่ไฟล์ที่ต้องการเปิด ดังรูป





การเรียกใช้งาน Google Colab

จากที่เคยกล่าวแล้วว่าการทำงาน Google Colab จะต้องมีบัญชี Google Drive ก่อน ดังนั้น หากผู้อ่านยังไม่มีบัญชีผู้ใช้งานของ google ก็จะต้องทำการสมัครก่อน โดยเปิดเว็บเบราว์เซอร์เข้าไปที่ <https://accounts.google.com/> และ create account ใหม่ขึ้นมา

ขั้นตอนการเรียกใช้งาน Google Colab มีดังนี้

1. เปิดเว็บเบราว์เซอร์เข้าไปที่ <https://drive.google.com> และทำการล็อกอินเข้า google drive ก่อน
2. เปิดเว็บเบราว์เซอร์เข้าไปที่ <https://colab.research.google.com/> จากนั้นเลือกไปที่แท็บ **Google Drive** และคลิกที่ **new notebook** ดังรูป

