อรพิน ประวัติบริสุทธิ์
ผู้เขียนหนังสือ Bestseller ด้าน Programming

eBook

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

PROVISION

อรพิน ประวัติบริสุทธิ์
ผู้เขียนหนังสือ Bestseller ด้าน Programming

# Python สำหรับงาน Data Science

## Data Visualization และ Machine Learning

# Python สำหรับงาน Data Science
## Data Visualization และ Machine Learning


**ผู้เขียน** : **อรพิน ประวัติบริสุทธิ์**
**ออกแบบปก** : ธนกฤต เกียรติศักดากุล
**จัดรูปเล่ม** : อภิชญา สุทธิประภา



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

ราคา : 355 บาท

**ข้อมูลบรรณานุกรมของหอสมุดแห่งชาติ**

อรพิน ประวัติบริสุทธิ์

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





PROVISION

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

**ผู้ก่อตั้ง/ประธานกรรมการ** วศิน เพิ่มทรัพย์
**กรรมการบริหาร** วงศ์ประชา จันทร์สมวงศ์
อนิรุทธ์ รัชตะวราห์
**ประสานงานกอง บก.** นนทพร ตันติเอี่ยมสกุล

**ติดต่อสั่งซื้อหนังสือ**
sales@provision.co.th
โทร. 0-2077-4058 หรือ 08-1928-5299
www.dplusshop.com dplusshop

**ติดต่อโฆษณาและการตลาด**
marketing@provision.co.th

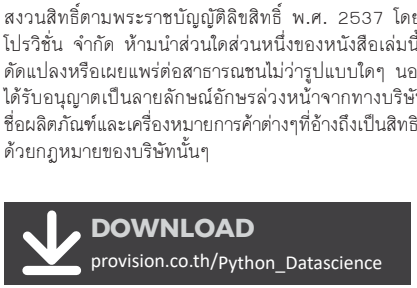
**สอบถามปัญหา/ติดต่อกอง บก.**
editor@provision.co.th

**ติดตามข่าวสาร**
www.provision.co.th, www.dplusguide.com
provision1991, dplusguide
@Provision1991, @dplusguide
provision1991, dplusguide
@dplusguide

**จัดจำหน่ายโดย :**
**บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)**
1858/87-90 ถ.เทพรัตน
แขวงบางนาใต้ เขตบางนา กรุงเทพฯ 10260
โทร. 0-2826-8000 FAX: 0-2826-8999

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

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



DIS_CHU

# INTRODUCTION

ปัจจุบันมีข้อมูลปริมาณมากมายมหาศาล ที่เรียกว่า Big Data เกิดขึ้นตลอดเวลาทุกวัน คงเป็นเรื่องน่าเสียดาย หากเราจะปล่อยผ่านข้อมูลเหล่านั้นทิ้งไปโดยไม่นำมาใช้ให้เกิดประโยชน์ใดๆ ดังนั้น จึงมีศาสตร์อย่างหนึ่งที่เรียกว่า Data Science เกิดขึ้นมา

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

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

นอกจากนี้ยังอธิบายถึงการทำ Data Visualization ซึ่งเป็นรูปแบบหนึ่งของการนำเสนอข้อมูลในรูปแบบของกราฟหรือแผนภูมิ เพื่อสื่อสารให้ผู้อ่านข้อมูลเข้าใจข้อมูลได้ง่ายขึ้นด้วย

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

ท้ายสุด หากหนังสือเล่มนี้ขาดตกบกพร่องประการใด ผู้เขียนขออภัยมา ณ ที่นี้ด้วยนะคะ และถ้าหนังสือเล่มนี้จะทำให้ผู้อ่านชอบใจอยู่บ้าง ผู้เขียนก็ขอมอบคุณงามความดีให้กับคุณประสิทธิ์ ประวัติบริสุทธิ์ และคุณวิภา ประวัติบริสุทธิ์ คุณพ่อคุณแม่ของผู้เขียน ผู้ซึ่งมอบกำลังใจดีๆ ให้กับผู้เขียนเสมอมา กำลังใจนี้เป็นส่วนสำคัญยิ่งในการผลักดันให้ผู้เขียนสร้างสรรค์ผลงานเขียนต่างๆ ออกมาสู่สายตาของผู้อ่าน

อรพิน ประวัติบริสุทธิ์

# AUTHOR

**อรพิน ประวัติบริสุทธิ์**
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, พัฒนาเว็บแอพบน Smartphone/Tablet ด้วย jQuery Mobile, คู่มือ Excel 2010/2013

# Contents

# Contents



CHAPTER

## 06 การจัดการกับข้อมูล Series และ DataFrame ด้วย Pandas

# Contents



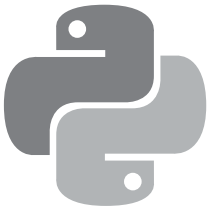
CHAPTER 10

## การทำ Data Visualization ด้วย Matplotlib

# Contents

CHAPTER

# 01



# ทำความรู้จักกับ Data Science

# Big Data และ Data Science คืออะไร

ปัจจุบันอินเทอร์เน็ตได้เข้ามามีบทบาทกับการดำรงชีวิตประจำวันของคนในสังคมมากขึ้น สื่อสังคมออนไลน์ (Social Media) ต่าง ๆ ก็มีอยู่เป็นจำนวนมาก เช่น Facebook, Twitter, Instragram, 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 วีดีโอ 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

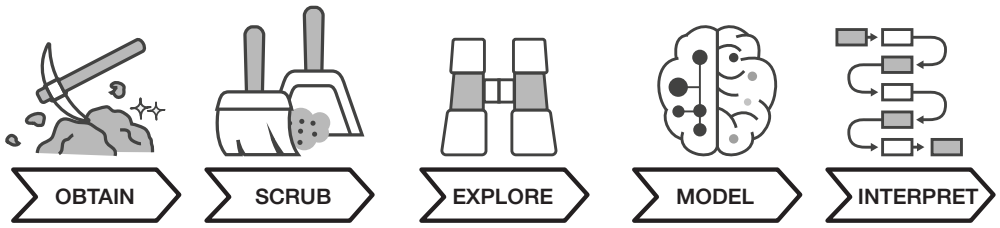
# ขั้นตอนการทำ Data Science (Data Science Pipeline)

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

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

- **O**btain
- **S**crub
- **E**xplore
- **M**odel
- i**N**terpret

## Data Science Pipeline




| O | S | E | M | N |
|---|---|---|---|---|
| รวบรวมข้อมูลจากแหล่งต่างๆ | ปรับปรุงข้อมูลให้อยู่ในรูปแบบที่นำไปใช้ได้ | ค้นหารูปแบบความสัมพันธ์และแนวโน้มต่างๆ | สร้างแบบจำลองเพื่อทำนายผล | นำแบบจำลองไปใช้งาน |

## Obtain

ขั้นตอนแรกของการทำ Data Science คือการรวบรวมข้อมูล ข้อมูลนั้นอาจจะมาจากหลาย ๆ แหล่ง ซึ่งวิธีการดึงข้อมูลอาจจะแตกต่างกัน เช่น เป็นข้อมูลในรูปแบบไฟล์ชนิดต่าง ๆ ที่ดาวน์โหลดมาจากเว็บไซต์ เช่น HTML หรือ Excel รวมไปถึงไฟล์ภาพหรือเสียง จากนั้นจึงทำการ extract ข้อมูลจากเนื้อหาของไฟล์นั้น หรือคิวรี่ (query) ข้อมูลมาจากฐานข้อมูลหรือดึงข้อมูลผ่าน 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

- แตกข้อมูลในคอลัมน์ให้กลายเป็นหลายคอลัมน์
- 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% ของโปรเจ็คในการ Scrub เพื่อให้ข้อมูลอยู่ในรูปแบบที่พร้อมในการนำไปวิเคราะห์และสร้างโมเดลข้อมูลในขั้นตอนต่อไป

# Explore

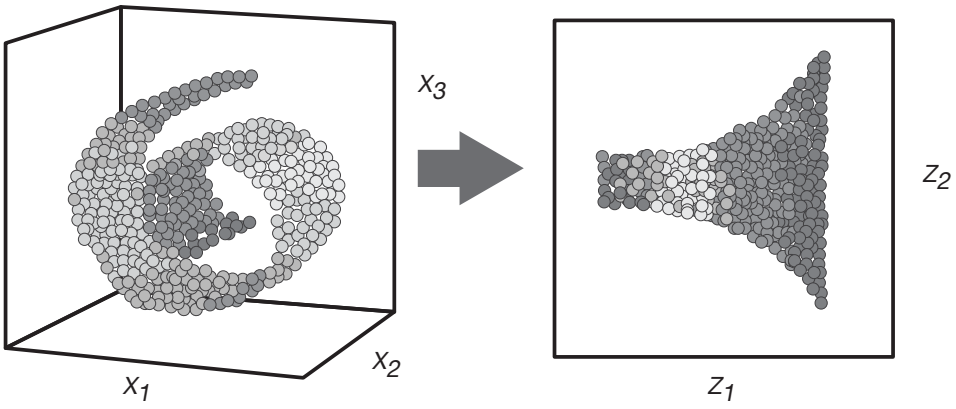
หลังจากทำการ Scrub หรือ Clean ข้อมูลแล้วนั้น ขั้นตอนถัดไปคือการค้นหาข้อมูลหรือส่วนของข้อมูลตามเรื่องที่เราต้องการวิเคราะห์ ซึ่งจะเป็นการเข้าไปค้นหาและทำความเข้าใจรูปแบบข้อมูล (pattern) ในกระบวนการนี้เราจะนำข้อมูลมาแสดงผล (Visualization) ในรูปแบบกราฟ หรือ chart ต่าง ๆ และนำกระบวนการทดสอบทางสถิติมาใช้กับชุดข้อมูล

กระบวนการนี้ต้องอาศัยการวิเคราะห์และสังเกตจากผลลัพธ์ของการคำนวณทางสิถิติหรือกราฟต่าง ๆ เพื่อค้นหารูปแบบหรือ trend ของข้อมูลในเรื่องที่สนใจ ซึ่งสิ่งสำคัญของกระบวนการนี้คือการทดสอบทางสถิติหรือสร้าง Visualization ให้ตรงกับความต้องการทางธุรกิจ (Business User)

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

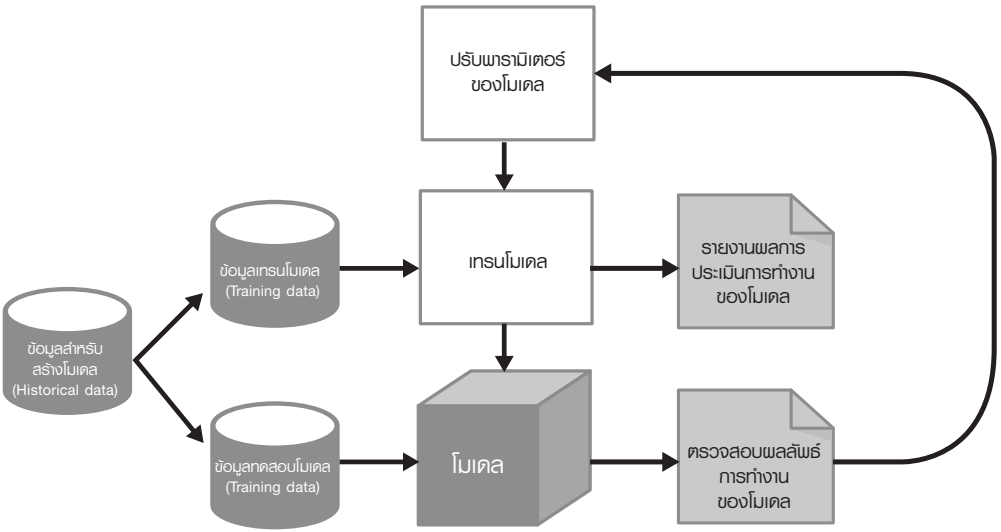
# Model

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



ในกระบวนการนี้คือการทำ Machine Learning ซึ่งเราจะนำ Model ที่สร้างขึ้นจากอัลกอริทึม เช่น K-Means มาเทรน (train) จากชุดข้อมูลที่เป็น History เพื่อให้ตัว Model เกิดการเรียนรู้และสามารถจำแนกความแตกต่างภายในกลุ่มข้อมูลได้ ยกตัวอย่างเช่น พฤติกรรมการซื้อสินค้าของกลุ่มผู้ซื้อสินค้าแบบออนไลน์ผ่านแอพพลิเคชั่นทางมือถือที่มีการค้นหาสินค้าแตกต่างกัน ซึ่งอาจต้องใช้เทคนิคการทำ Cluster Analysis ในการแยกกลุ่ม เป็นต้น เมื่อผ่านการเทรนและได้ Predictive Model ออกมาแล้วนั้น จะมีการตรวจสอบจากข้อมูลอีกชุดว่า Model นั้นให้ผลลัพธ์ที่ถูกต้องและเชื่อถือได้หรือไม่ ซึ่งจะนำผลลัพธ์กลับไปตั้งค่าเป็นพารามิเตอร์เพื่อทำการเทรนครั้งต่อ ๆ ไป ซึ่งจะทำให้ระบบเกิดการเรียนรู้และได้ Predictive Model ที่สามารถทำนายหรือให้ผลลัพธ์ที่ถูกต้องแม่นยำได้

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




## Intepret

ขั้นตอนสุดท้ายของกระบวนการ OSEMN ซึ่งถือเป็นขั้นตอนสำคัญที่สุดคือ Intepret ซึ่งขั้นตอนการทำงานคือการสรุปผลลัพธ์หรือ insight ที่ได้จากกราฟหรือการสร้างโมเดลข้อมูล ซึ่งต้องอาศัยการแปลผลเพื่อสรุปให้ผู้บริหารหรือผู้รับข้อมูลเข้าใจในเชิงธุรกิจที่จะนำข้อมูลไปใช้ด้วย

ทั้งนี้ ในการทำทั้ง 5 ขั้นตอนที่ผ่านมา อาจไม่ได้เป็นการเรียงลำดับจาก O ถึง N เพียงครั้งเดียว แต่อาจจะมีการย้อนกลับไปทำขั้นตอนก่อนหน้า และอาจมีการวน (iteration) ของการทำ O ถึง N หรือขั้นตอนใดขั้นตอนหนึ่งหลาย ๆ รอบก็ได้

# Data Science กับ Python

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

ข้อดีของการใช้ Python ทำงาน Data Science สามารถสรุปได้ดังนี้

- เป็นภาษาที่ง่ายต่อการเรียนรู้ สามารถประมวลผลได้โดยเขียนโปรแกรมเพียงแค่ไม่กี่บรรทัด เมื่อเทียบกับภาษาอื่นเช่น ภาษา R ซึ่งทำให้เข้าใจได้ง่าย
- เป็นภาษา cross platform ซึ่งสามารถทำงานได้ในทุกระบบปฏิบัติการด้วยโค้ดโปรแกรมชุดเดิม
- ทำงานได้เร็วกว่าภาษาอื่นอย่าง R และ MATLAB
- Python เป็นภาษาที่มีการจัดการทรัพยากรได้ดี ทำให้ใช้หน่วยความจำ (memory) น้อยในการประมวลผล โดยเฉพาะอย่างยิ่งในการต้องทำงานกับข้อมูลจำนวนมากเพื่อแปลงข้อมูล (Data Transformation), การแยกข้อมูลออกเป็นส่วนและพลิกแกนหรือมิติข้อมูล (Slice and Dice) รวมไปถึงการแสดงข้อมูลออกมาเป็นกราฟหรือแผนภูมิ (Visualization)
- Python เป็นภาษาซึ่งมีไลบรารี (Libraries) ต่าง ๆ มากมายในด้าน Data Science และมีการพัฒนาเวอร์ชั่นใหม่ ๆ เพื่อเพิ่มเติมความสามารถในการประมวลผลออกมาเรื่อย ๆ เช่น NumPy, Panda, SciPy เป็นต้น
- Python มีแพ็คเกจซึ่งสามารถเรียกใช้งานโค้ดโปรแกรมของภาษาอื่นเช่น ภาษา C หรือ Java ซึ่งช่วยให้สามารถทำงานร่วมกับการประมวลผลด้วยโปรแกรมอื่นได้

# ไลบรารีพื้นฐานสำหรับการทำ 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 เป็นที่นิยมใช้เนื่องจากมีไลบรารีที่รองรับการทำงานในขั้นตอนต่าง ๆ ได้ดี เช่น ดึงข้อมูล (load) จากไฟล์หลากหลายชนิดมาจัดเตรียม, จัดเรียงข้อมูลหรือจัดการข้อมูลที่หายไป, ปรับรูปแบบข้อมูลหรือทำ Pivot ให้กับชุดข้อมูล, รวมไปถึงการจัดกลุ่ม (Re-grouping) และรวมข้อมูล (Merge and Join) ซึ่งในหนังสือเล่มนี้จะได้กล่าวถึงขั้นตอนต่าง ๆ ในการทำงานกับ Pandas ต่อไป

## NumPy

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

NumPy เป็นไลบรารีแบบ open source ที่สามารถใช้งานได้ฟรี การใช้งานจะต้องติดตั้งไลบรารีเพิ่มเติมเนื่องจากไม่ใช่ไลบรารีพื้นฐานของไพธอน แต่ถ้าเราติดตั้งโปรแกรม Anaconda ก็จะมีไลบรารีนี้ติดตั้งมาให้โดยอัตโนมัติ สามารถเรียกใช้งานได้ทันที

ไลบรารีนี้ใช้สำหรับจัดการเกี่ยวกับคณิตศาสตร์และการคำนวนต่าง ๆ 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

Matpolotlib เป็นไลบรารีที่ใช้สำหรับการทำ Data Visualization โดยมันจะมีฟังก์ชั่นเกี่ยวกับการวาดกราฟต่าง ๆ ให้เรียกใช้งาน เช่น กราฟแท่ง กราฟวงกลม เป็นต้น สามารถวาดกราฟแบบ 2 มิติ หรือ 3 มิติได้ โมดูลที่สำคัญของไลบรารีนี้คือ “pyplot” ซึ่งเป็นโมดูลหลักของการกำหนดรูปแบบของเส้นกราฟ ฟอนต์ รูปแบบแกนของกราฟ และลูกเล่นอื่น ๆ อีกมากมาย

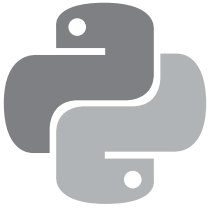
ไลบรารีนี้ใช้ร่วมกับ NumPy ในการทำงานแทนการใช้โปรแกรม MATLAB นอกจากนี้ยังสามารถใช้ร่วมกับไลบรารีอื่นที่เกี่ยวข้องกับการวาดกราฟฟิกอย่าง PyQt และ wxPython ได้อีกด้วย

## Seaborn

Seaborn เป็นไลบรารีสำหรับการพล็อตกราฟในขั้นตอนการทำ Data Visualization ซึ่งเรียกใช้ไลบรารี Matpolotlib ในการทำงานอีกทีหนึ่ง ไลบรารีนี้มีประโยชน์ในการแสดงกราฟค่าการกระจายตัวของข้อมูล (Random Distributions) เนื่องจากมีฟังก์ชั่นโดยเฉพาะสำหรับการรับข้อมูลแบบอาร์เรย์ไปแสดงการกระจายตัวของชุดข้อมูลในเชิงสถิติ ทั้งการแสดงกราฟแบบระฆังคว่ำ, ปัวซอง (Poisson), ไคสแดวร์ (Chi Square) และอื่น ๆ รวมไปถึงสามารถแสดงได้ทั้งแบบกราฟเส้นหรือกราฟแท่ง

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 ที่ทำงานบนเว็บบราวเซอร์ในลักษณะโครงสร้างแบบ 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 จะถูกสตาร์ทให้ทำงานแล้วมันจะเปิด Jupyter Notebook client บนเว็บบราวเซอร์ให้อัตโนมัติด้วย เราสามารถเขียนโปรแกรมต่าง ๆ ของงาน Data Science ลงบน Jupyter Notebook client และสั่งรันเพื่อดูผลลัพธ์ได้

# Jupyter Notebook กับ Google Colab ต่างกันอย่างไร

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

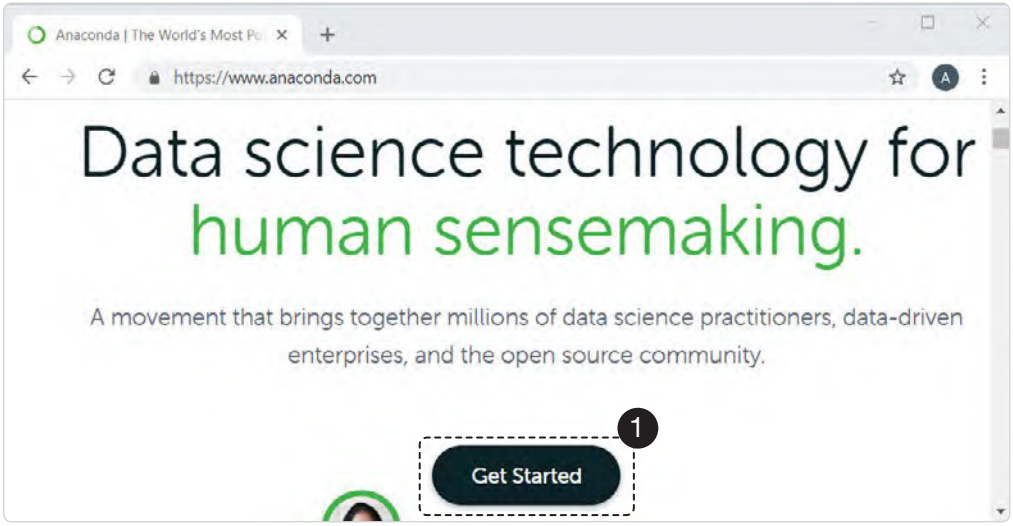
Google Colab (ย่อมาจาก Google Colaboratory) เป็นโปรเจ็คที่เกิดจากความร่วมมือของทีมงาน Google และทีมงานผู้สร้าง Jupyter Notebook ที่พัฒนา Jupyter Notebook ให้ทำงานอยู่บน Cloud ดังนั้น การเรียกใช้งาน Google Colab จึงไม่จำเป็นต้องติดตั้งโปรแกรมใด ๆ ลงบนเครื่องคอมพิวเตอร์ก่อนใช้งาน แต่การใช้งานจะต้องใช้งานผ่านเว็บบราวเซอร์ และต้องเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ตทุกครั้งที่ใช้งาน

การใช้งาน Google Colab ทุกคนสามารถใช้งานได้ฟรี แค่เพียงมีบัญชี Google Drive ก็สามารถใช้งานได้แล้ว เมื่อเขียนโปรแกรมเสร็จก็สามารถจัดเก็บโค้ดโปรแกรมไว้บน google drive ของตัวเองได้ และสามารถแชร์โค้ดต่าง ๆ ให้กับผู้อื่นได้อีกด้วย

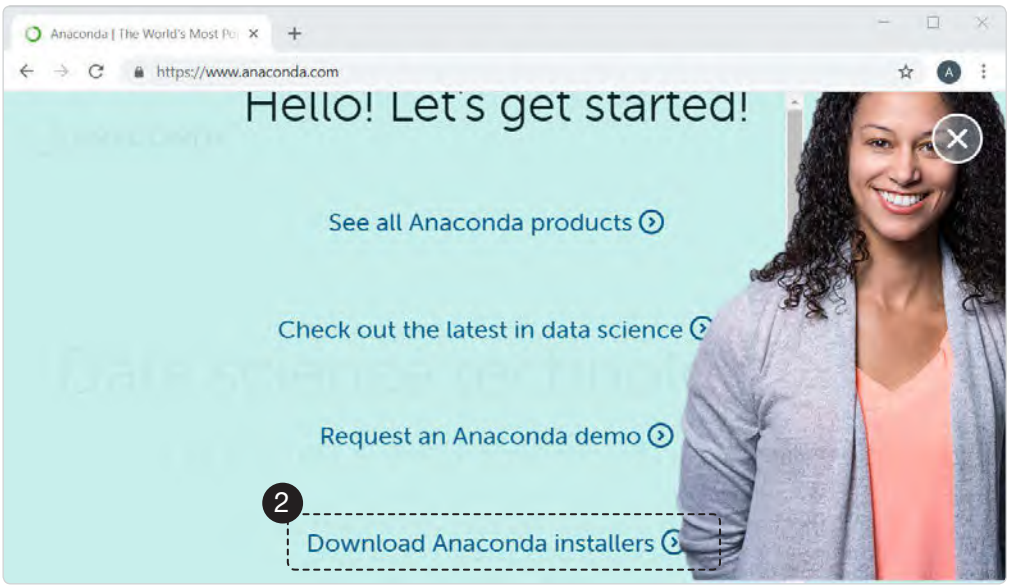
# การติดตั้งโปรแกรม Anaconda

การติดตั้งโปรแกรม Anaconda มีขั้นตอนดังนี้

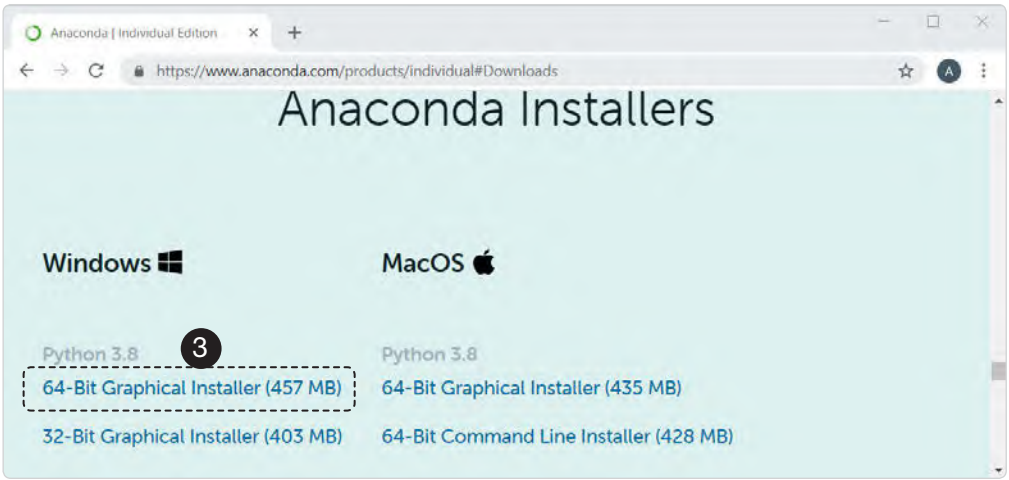
**1.** เปิดเว็บบราวเซอร์ไปที่ http://www.anaconda.com แล้วคลิกเลือก **Get Started**



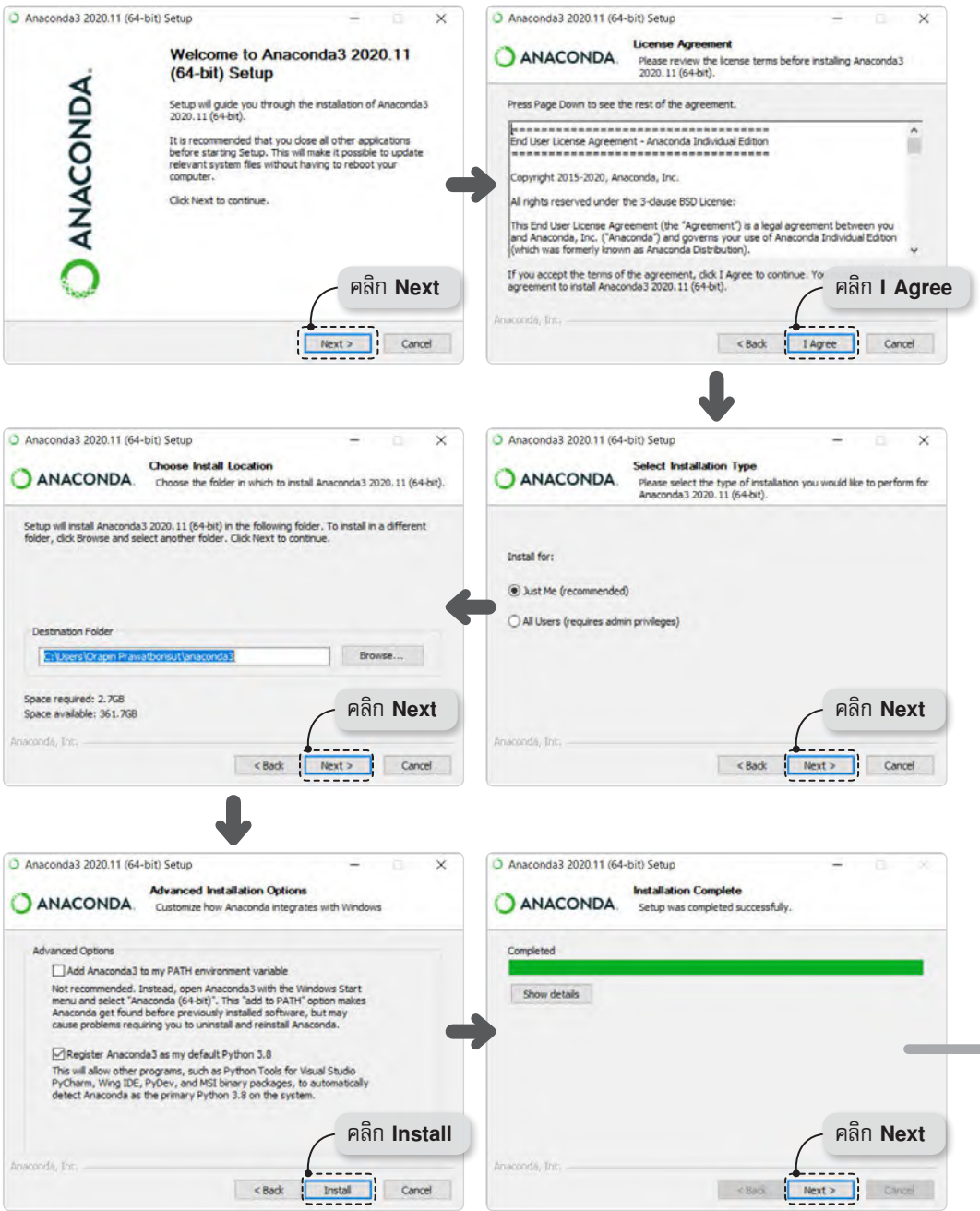
**2.** คลิกที่ **Download Anaconda Installers**

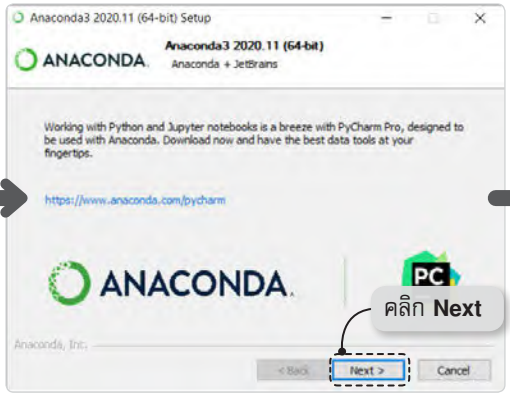



**3.** เลื่อนไปที่ระบบปฏิบัติการ **Windows** แล้วคลิกเลือก **64-Bit Graphical Installer (457 MB)** แต่ถ้าระบบปฏิบัติการที่เครื่องคอมพิวเตอร์ของผู้อ่านเป็นแบบ 32 บิต ก็ให้คลิกที่ 32 Bit Graphical Installer (403 MB) จากนั้นรอจนกระทั่งดาวน์โหลดไฟล์สำเร็จ



**4.** ดับเบิลคลิกที่ไฟล์ที่ดาวน์โหลดมาเพื่อเริ่มการติดตั้ง โดยการติดตั้งให้ทำตามขั้นตอนดังรูป



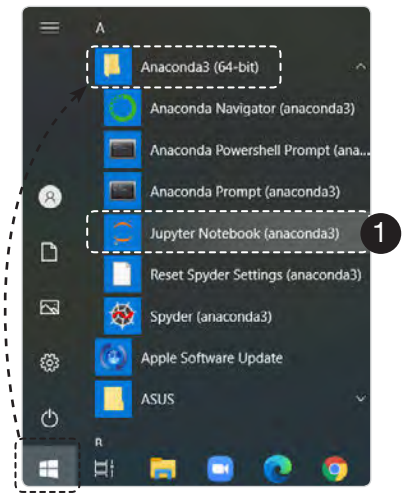


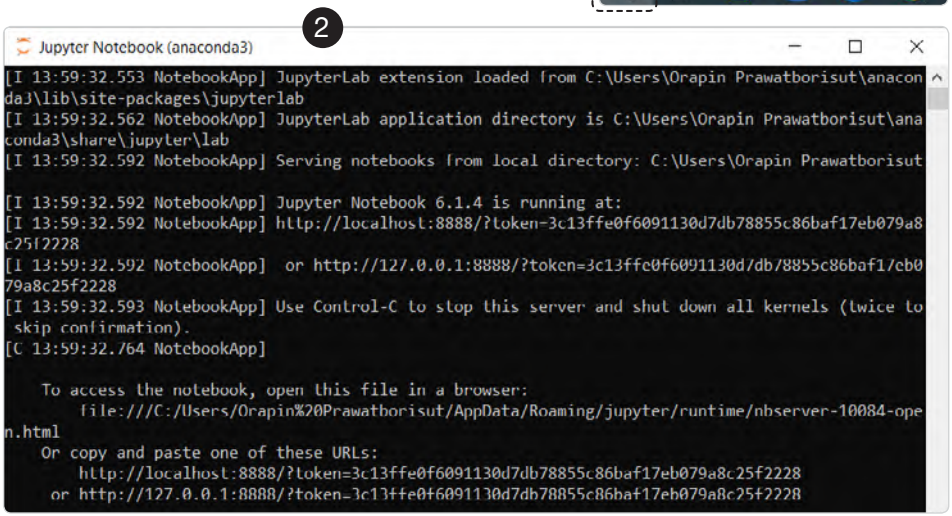


# การเรียกใช้งาน Jupyter Notebook

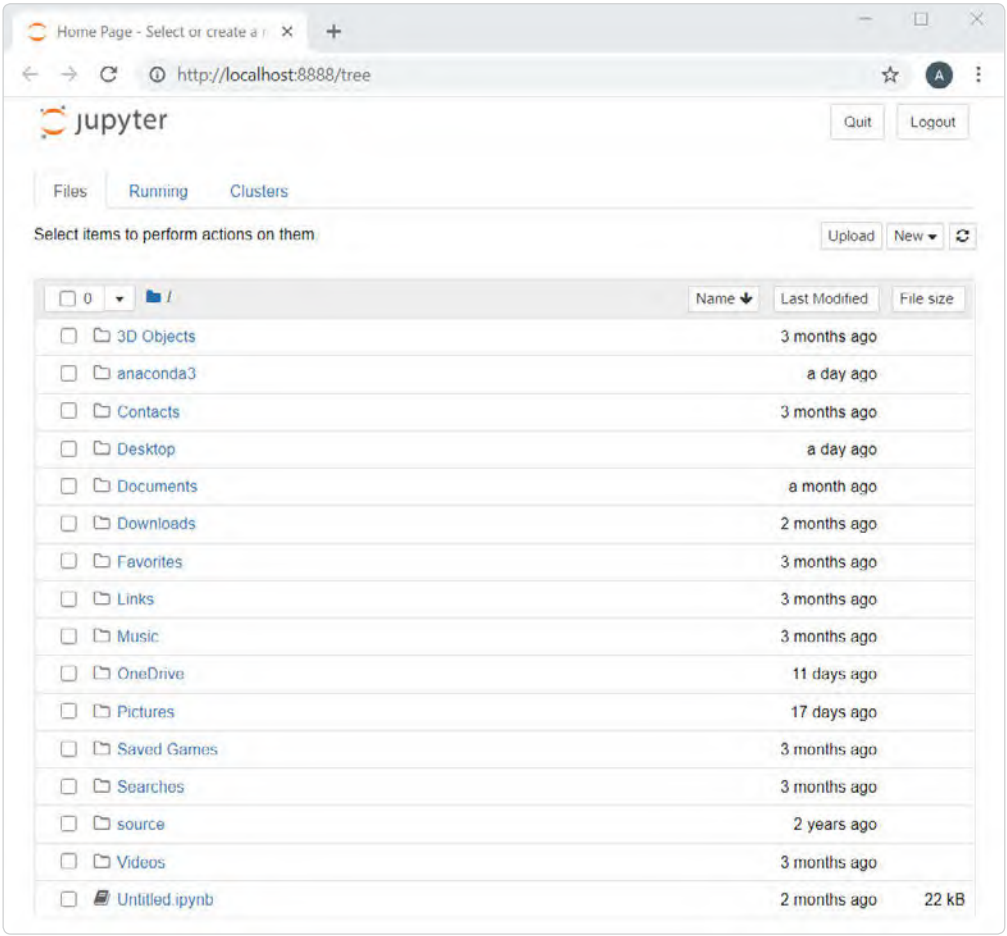
การเรียกใช้งาน Jupyter Notebook มีขั้นตอนดังนี้

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 ก็จะเปิดขึ้นมาบนเว็บบราวเซอร์โดยอัตโนมัติ ดังรูป

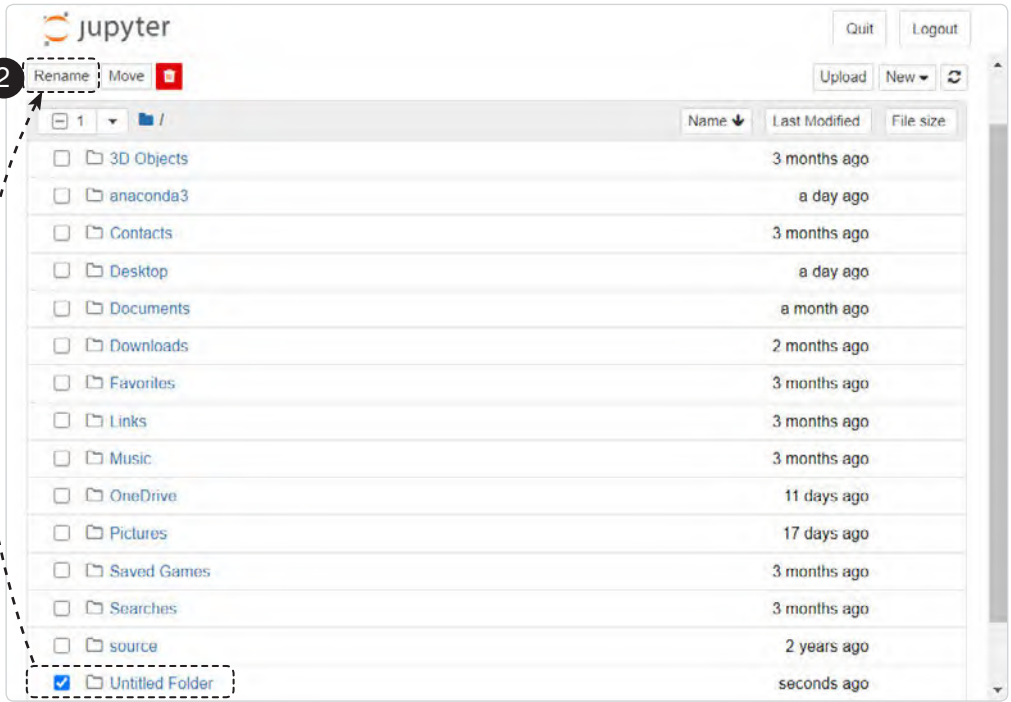


# การสร้าง folder เพื่อจัดเก็บโค้ดโปรแกรม

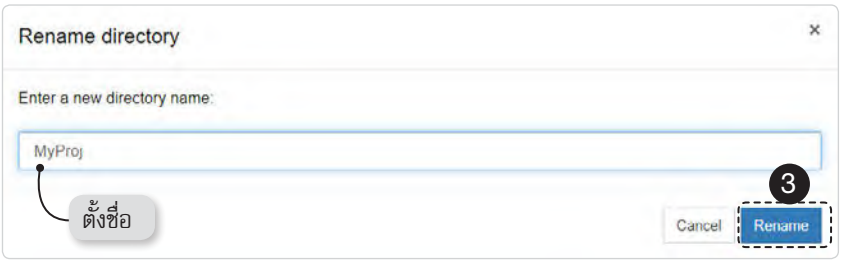
เราสามารถสร้าง folder ใหม่ขึ้นมาใน Juyter Notebook เพื่อจัดเก็บโค้ดโปรแกรมของเราได้ดังขั้นตอนต่อไปนี้

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


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



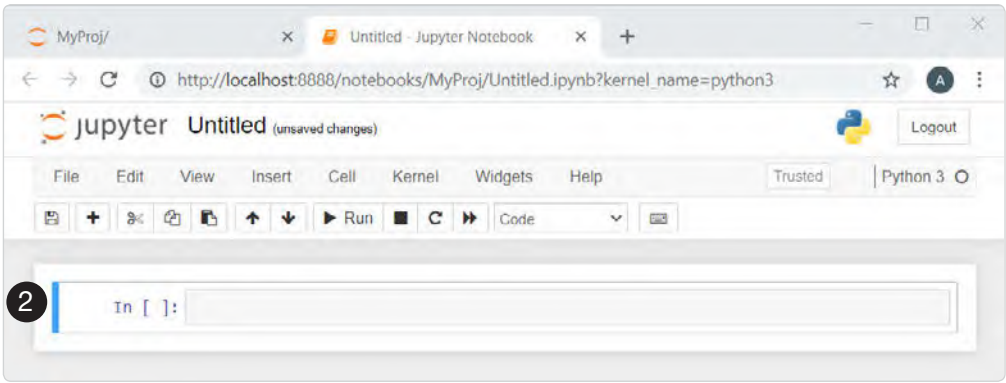
3. กำหนดชื่อโฟลเดอร์ที่ต้องการ และคลิกปุ่ม **Rename**



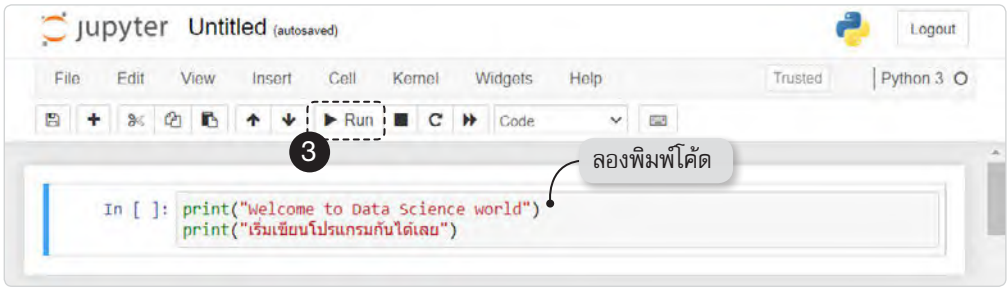

## การเขียนโปรแกรมลงบน Jupyter Notebook

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

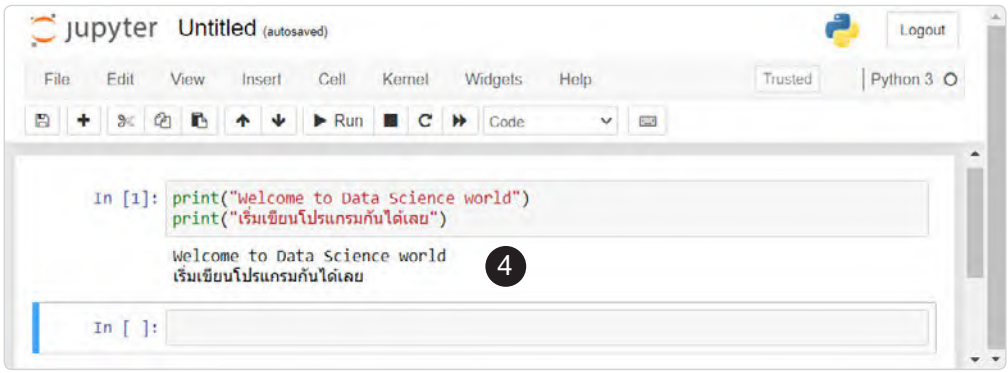

2. จะมีหน้าต่างแสดงขึ้นมาดังรูป เราสามารถเขียนโค้ดโปรแกรมลงไปที่ช่องเซลล์ที่ชื่อว่า `In [ ]:` ได้



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



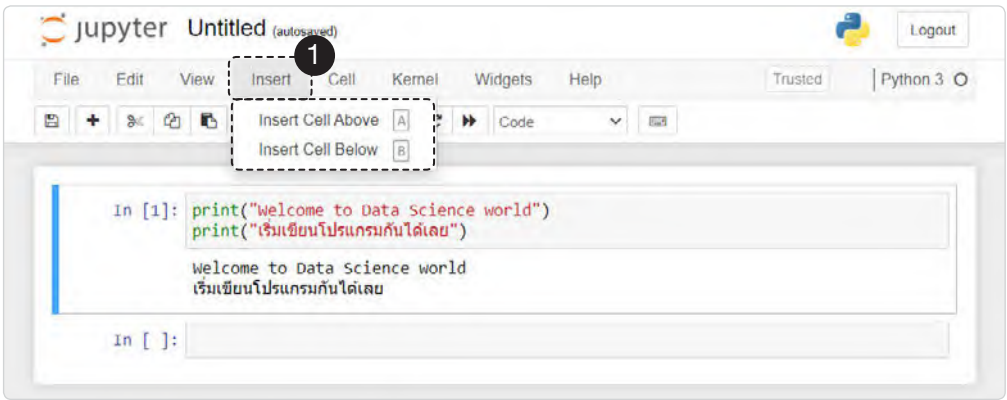

**4.** เมื่อสั่งรันโปรแกรมแล้ว จะมีผลลัพธ์แสดงออกมา ดังรูป



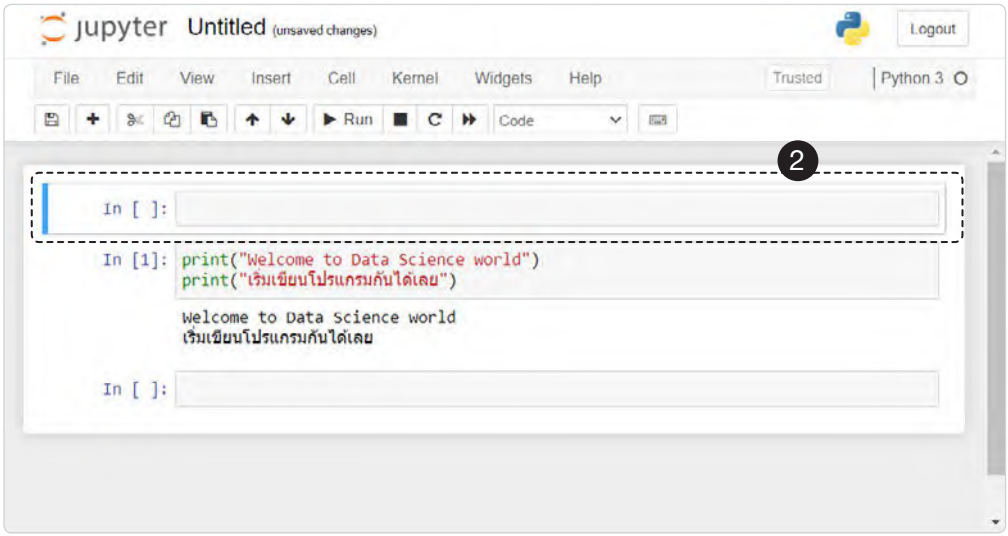

จะพบว่าเมื่อเรารันโค้ดโปรแกรมหนึ่ง ๆ แล้ว จะพบว่ามีเซลล์ `In [ ]:` ใหม่แสดงขึ้นมา เราสามารถเขียนโค้ดโปรแกรมอื่น ๆ ลงในแต่ละเซลล์เพื่อดูผลลัพธ์ได้ เซลล์แต่ละเซลล์จะแยกการทำงานออกจากกัน

## การลบและเพิ่มเซลล์

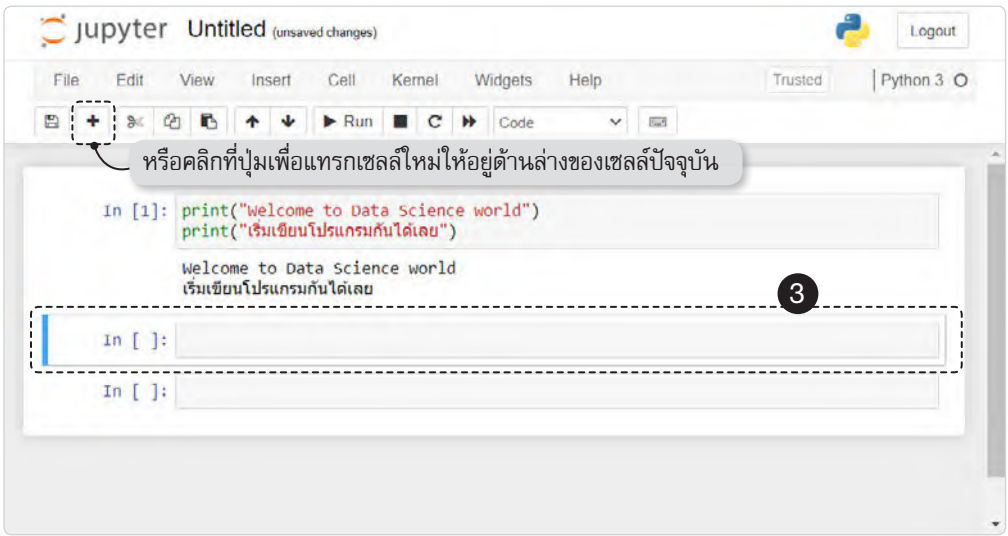
**1.** ในกรณีที่ต้องการเพิ่มเซลล์ ให้คลิกที่เมนู **Insert**



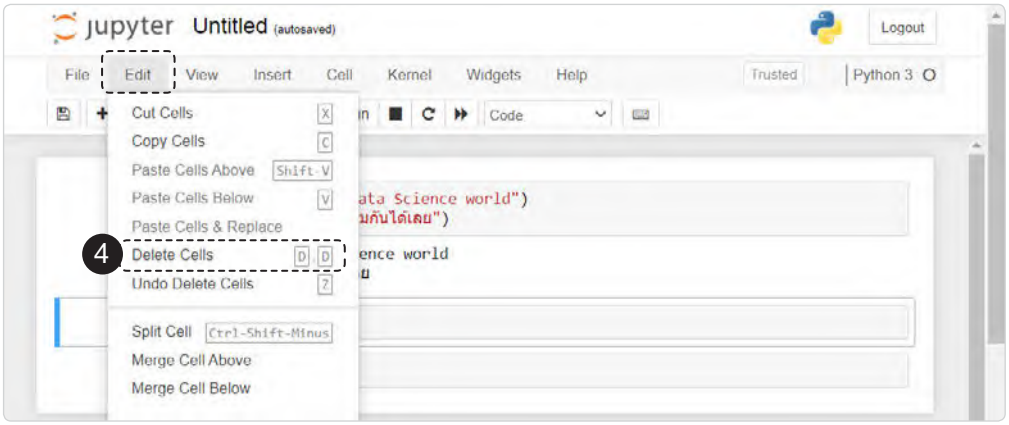
**2.** หากต้องการเพิ่มเซลล์ใหม่ให้อยู่ด้านบนของเซลล์ปัจจุบันที่ทำงานอยู่ก็เลือกที่ **Insert Cell Above** จะได้ผลลัพธ์ดังรูป




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

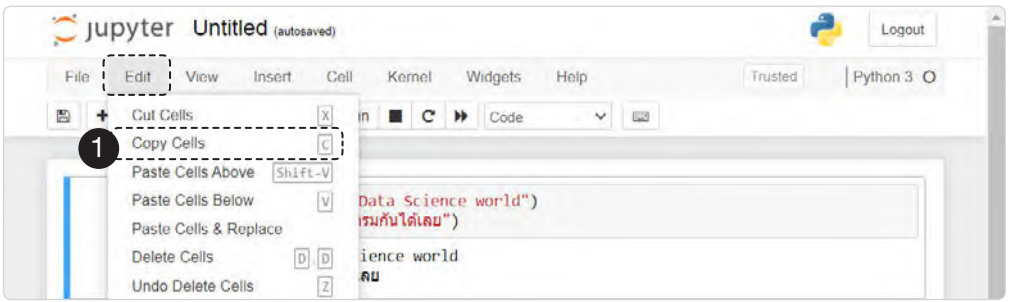


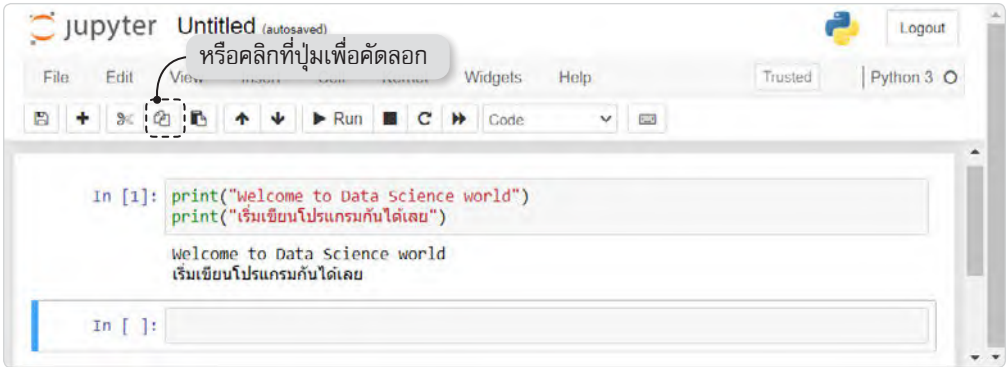
4. กรณีที่ต้องการลบเซลล์ ให้คลิกเลือกที่เซลล์ที่ต้องการลบ จากนั้นไปที่เมนู **Edit** และเลือก **Delete Cells** ดังรูป



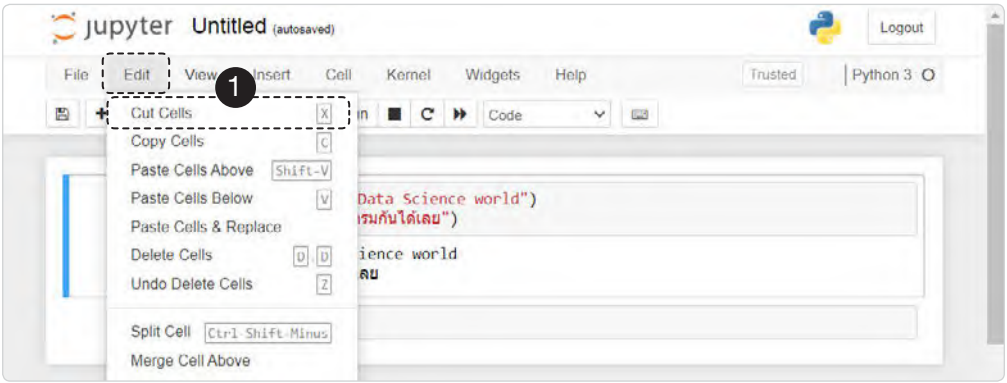

## การคัดลอก/ตัดเซลล์เดิมไปไว้ยังเซลล์ใหม่

1. **การคัดลอกเซลล์** ให้คลิกเลือกเซลล์ต้นทางที่ต้องการคัดลอก จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Copy Cells** หรือจะคลิกปุ่ม ก็ได้



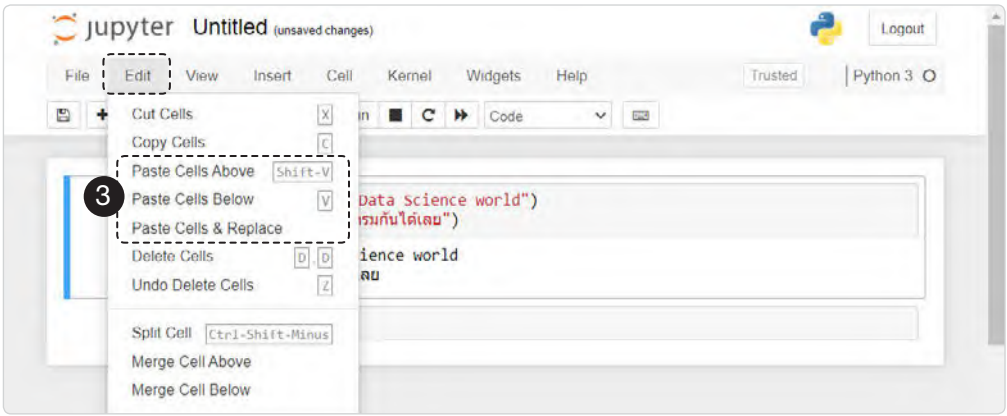



**2. การตัดเซลล์** ให้คลิกเลือกเซลล์ที่ต้องการตัด จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Cut Cells** หรือจะคลิกปุ่ม ✂ ก็ได้

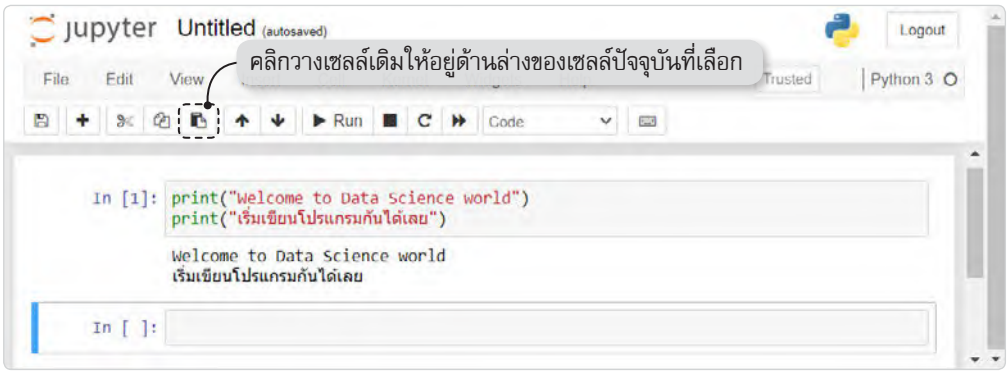




**3.** เมื่อคัดลอกหรือตัดเซลล์เดิมแล้ว หากต้องการ**นำเซลล์ไปวาง**ไว้ยังเซลล์ใหม่ ให้ไปที่เมนู **Edit** แล้วคลิกที่ **Paste** ซึ่งแบ่งออกได้เป็น 3 ประเภท คือ



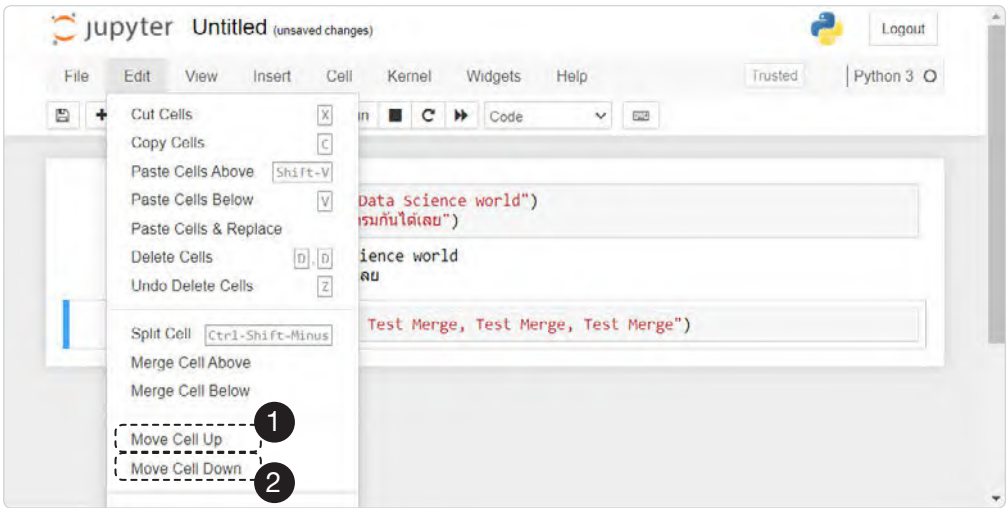
- **Paste Cells Above** วางเซลล์เดิมที่คัดลอกหรือตัดมา ให้อยู่ด้านบนของเซลล์ปัจจุบันที่เลือก
- **Paste Cells Below** วางเซลล์เดิมที่คัดลอกหรือตัดมา ให้อยู่ด้านล่างของเซลล์ปัจจุบันที่เลือก (กรณีนี้สามารถคลิกปุ่ม เพื่อทำการ Paste Cells Below ได้เช่นกัน ดังรูป)

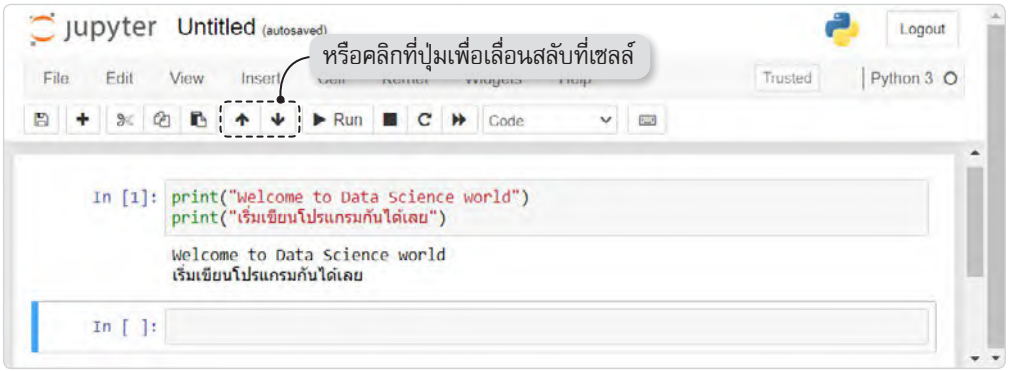



- **Paste Cells & Replace** วางเซลล์เดิมที่คัดลอกหรือตัดมา แทนที่เซลล์ปัจจุบันที่เลือก

## การสลับที่เซลล์

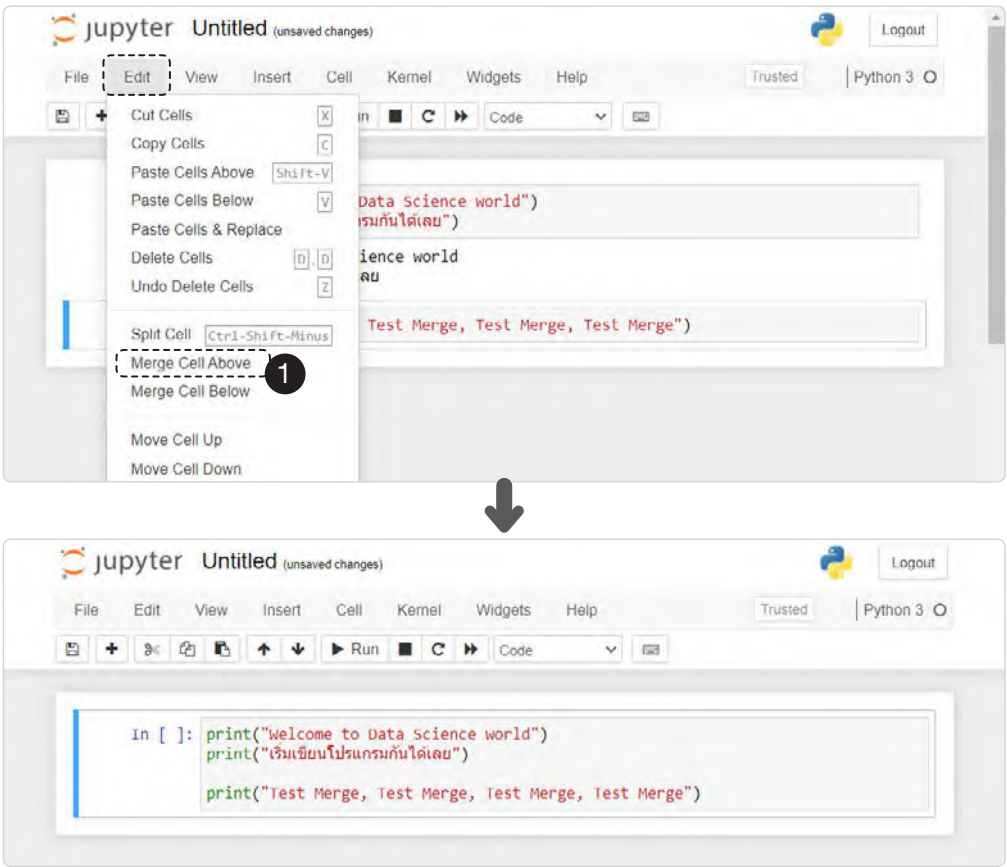
1. **การเลื่อนสลับที่เซลล์ปัจจุบันที่เลือกให้ขึ้นไปอยู่ด้านบน** ทำได้โดยไปที่เมนู **Edit** แล้วคลิกที่ **Move Cell Up** หรือจะคลิกปุ่ม ↑ ก็ได้
2. **การเลื่อนสลับที่เซลล์ปัจจุบันที่เลือกให้ลงไปอยู่ด้านล่าง** ทำได้โดยไปที่เมนู **Edit** แล้วคลิกที่ **Move Cell Down** หรือจะคลิกปุ่ม ↓ ก็ได้



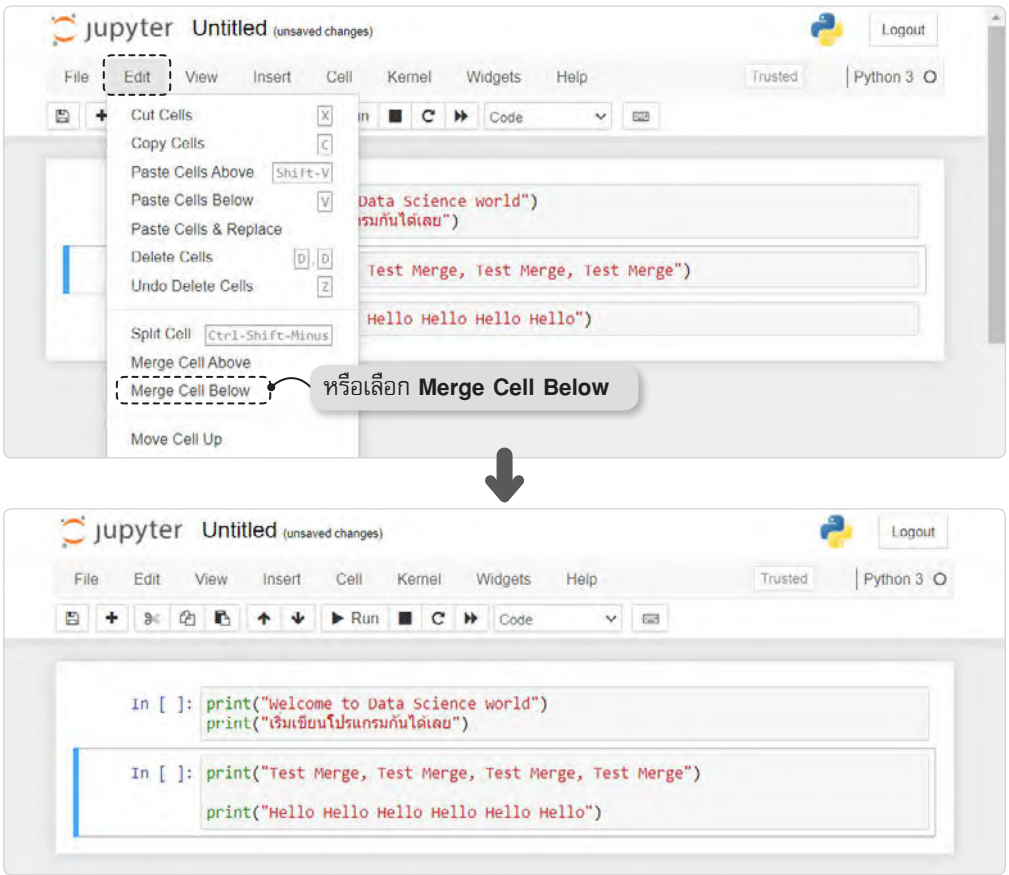



## การรวมเซลล์และแตกเซลล์

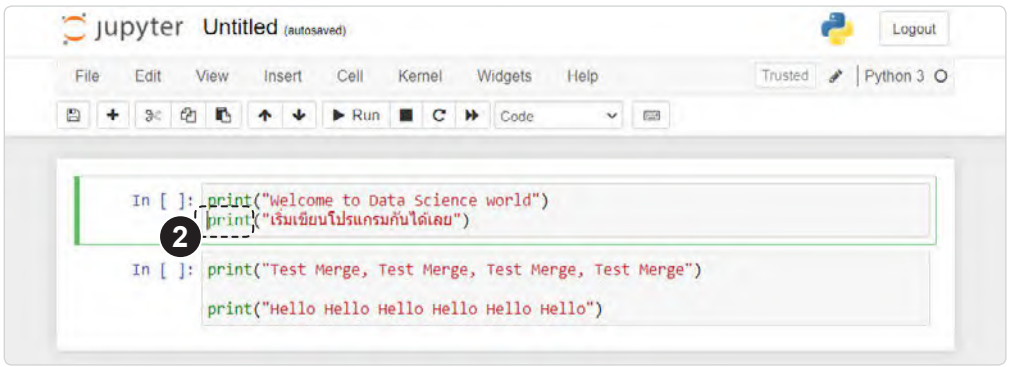
1. **การรวมเซลล์** ทำได้โดยคลิกเลือกเซลล์ที่ต้องการ จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Merge Cell Above** แล้วเซลล์ปัจจุบันที่เลือกจะถูกรวมกับเซลล์ที่อยู่ด้านบนเป็นเซลล์เดียวกัน ดังรูป



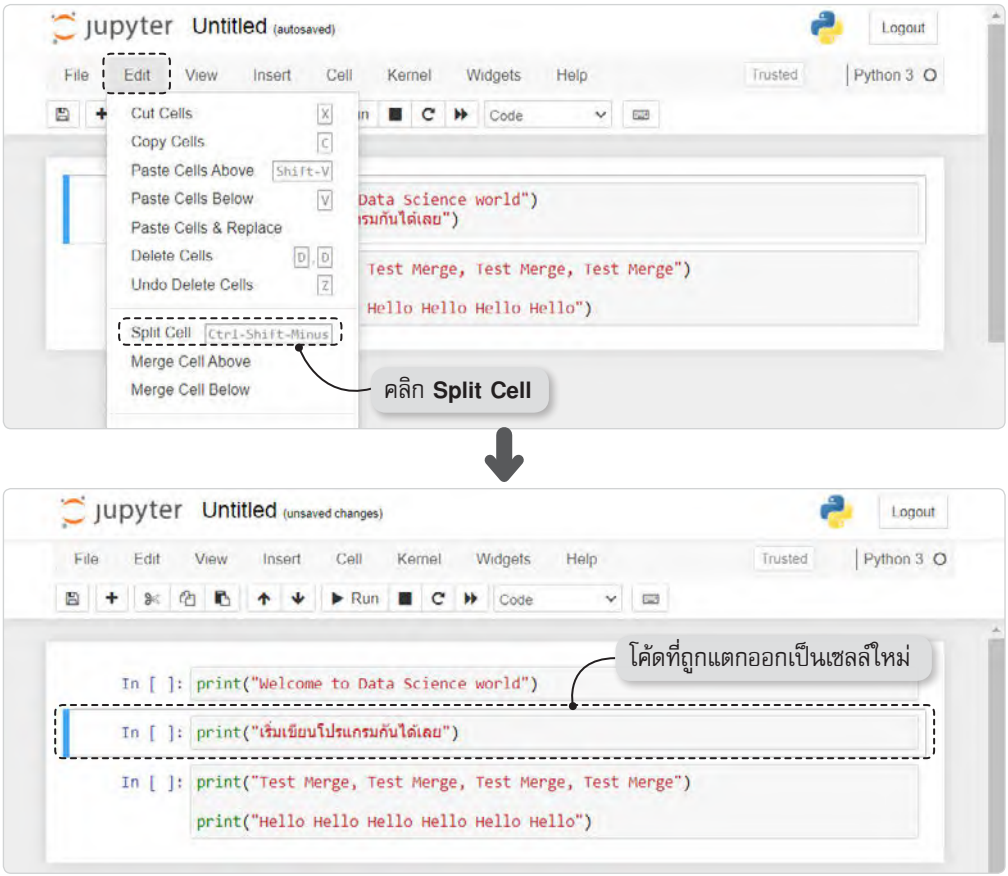
- หรือจะคลิกที่ **Merge Cell Below** ก็ได้ จะมีผลทำให้เซลล์ปัจจุบันที่เลือกจะถูกรวมกับเซลล์ที่อยู่ด้านล่างเป็นเซลล์เดียวกัน ดังรูป




**2. การแตกเซลล์** ทำได้โดยคลิกเลือกเซลล์ที่ต้องการ และนำเคอร์เซอร์ไปวางที่จุดแรกของโค้ดที่ต้องการแตกเซลล์ออกไป ดังรูป

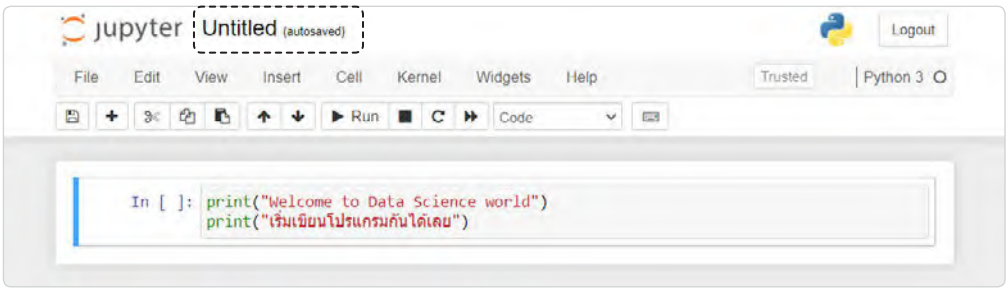


จากนั้นไปที่เมนู **Edit** แล้วคลิกที่ **Split Cell** แล้วโค้ดของเซลล์ปัจจุบันที่เลือกจะถูกแตกออกไปเป็นเซลล์ใหม่ ดังรูป

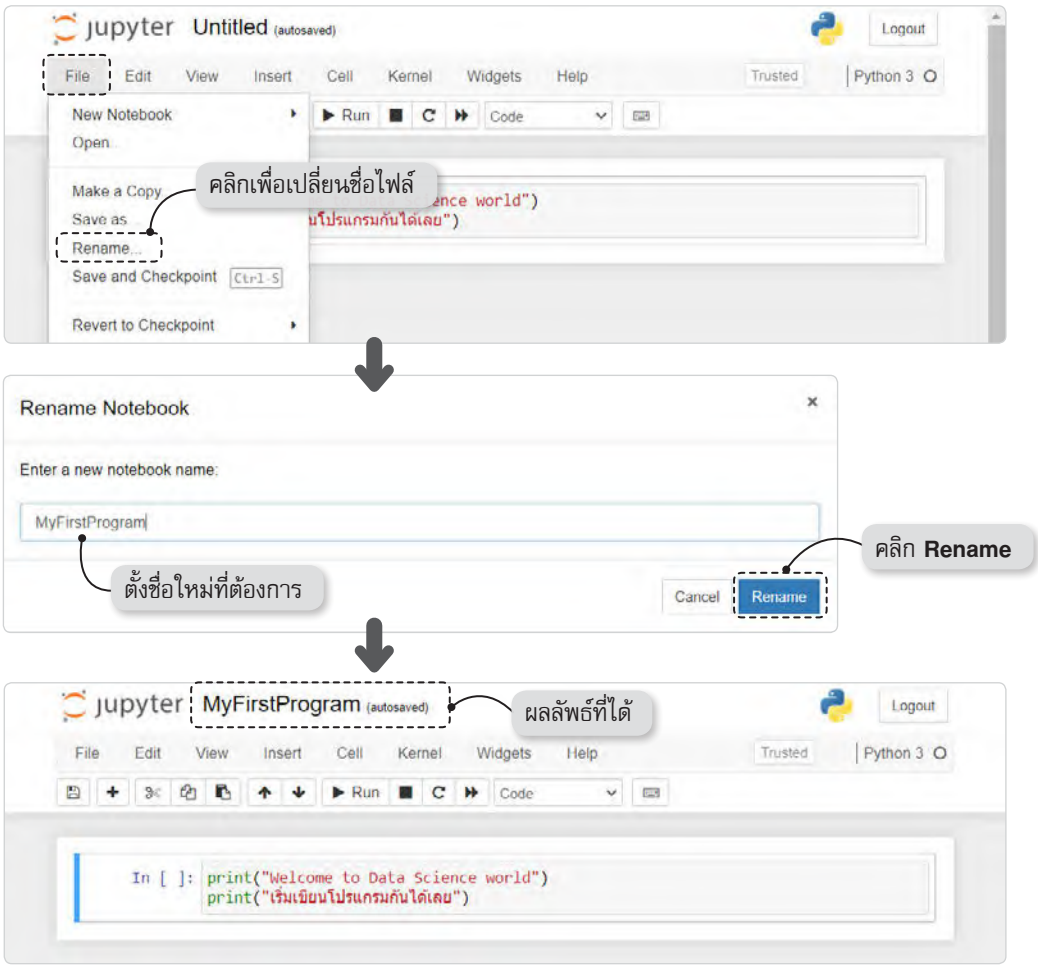



## การบันทึกโค้ดโปรแกรมใน Jupyter Notebook

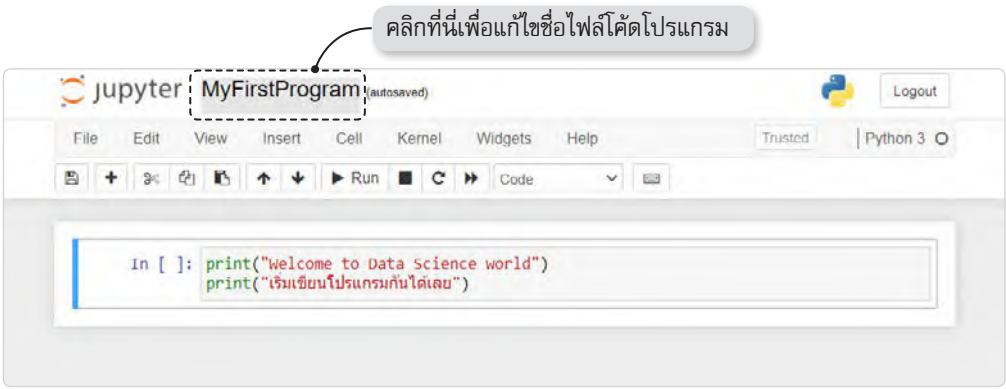
โดยปกติโค้ดโปรแกรมที่เราเขียนจะถูกบันทึกโดยอัตโนมัติ (autosaved) ซึ่งในกรณีนี้โค้ดโปรแกรมจะถูกบันทึกด้วยชื่อ Untitled.ipynb ดังรูป




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

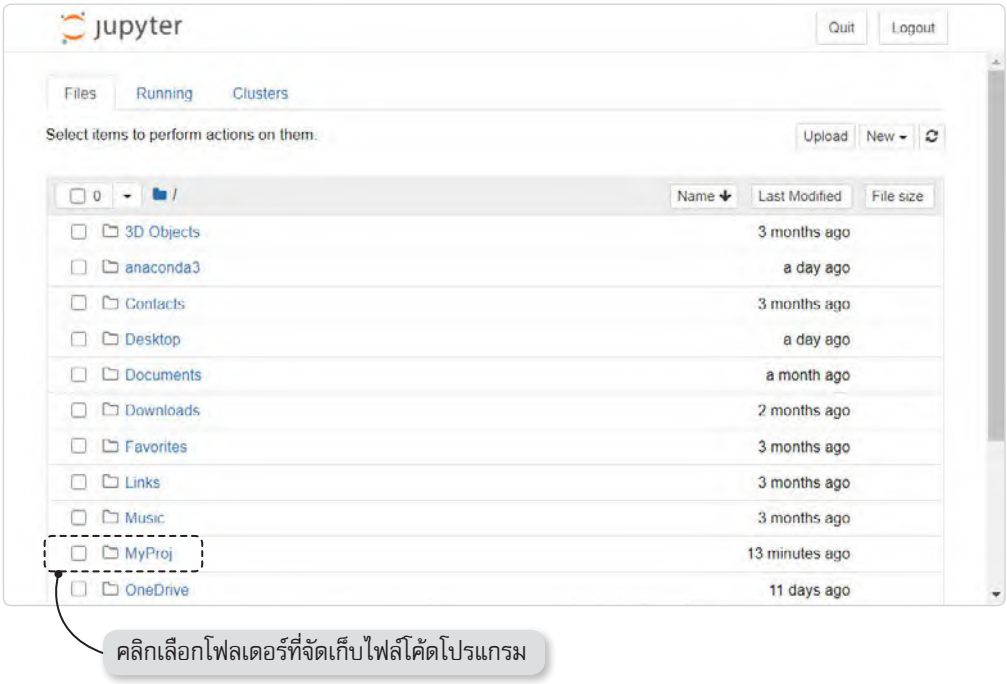


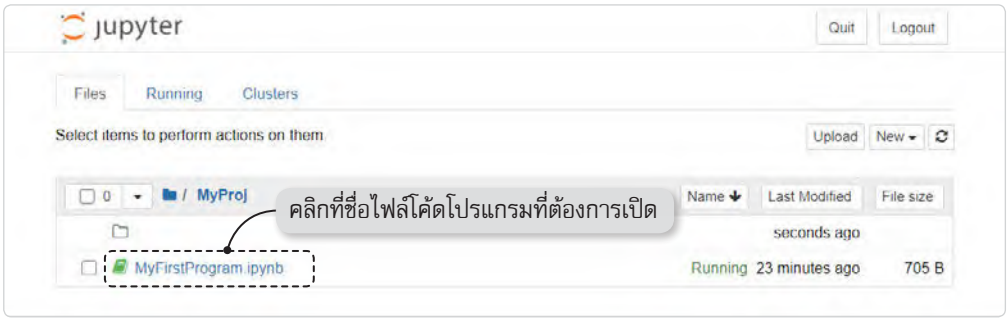
การ Rename ชื่อไฟล์สามารถทำได้อีกวิธีหนึ่ง คือ นำเมาส์ไปคลิกที่ชื่อไฟล์โค้ดโปรแกรมเพื่อแก้ไขชื่อไฟล์ ดังรูป




## การเปิดไฟล์โค้ดโปรแกรมใน Jupyter Notebook

การเปิดไฟล์โค้ดโปรแกรมทำได้โดยไปที่หน้าจอของ Jupyter Notebook Client แล้วเลือกไปยังโฟลเดอร์ที่จัดเก็บไฟล์โค้ดโปรแกรมที่ต้องการ จากนั้นคลิกไปที่ไฟล์ที่ต้องการเปิด ดังรูป






# การเรียกใช้งาน Google Colab

จากที่เคยกล่าวแล้วว่าการใช้งาน Google Colab จะต้องมีบัญชี Google Drive ก่อน ดังนั้น หากผู้อ่านยังไม่มีบัญชีผู้ใช้งานของ google ก็จะต้องทำการสมัครก่อน โดยเปิดเว็บบราวเซอร์เข้าไปที่ https://accounts.google.com/ และ create account ใหม่ขึ้นมา

ขั้นตอนการเรียกใช้งาน Google Colab มีดังนี้

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