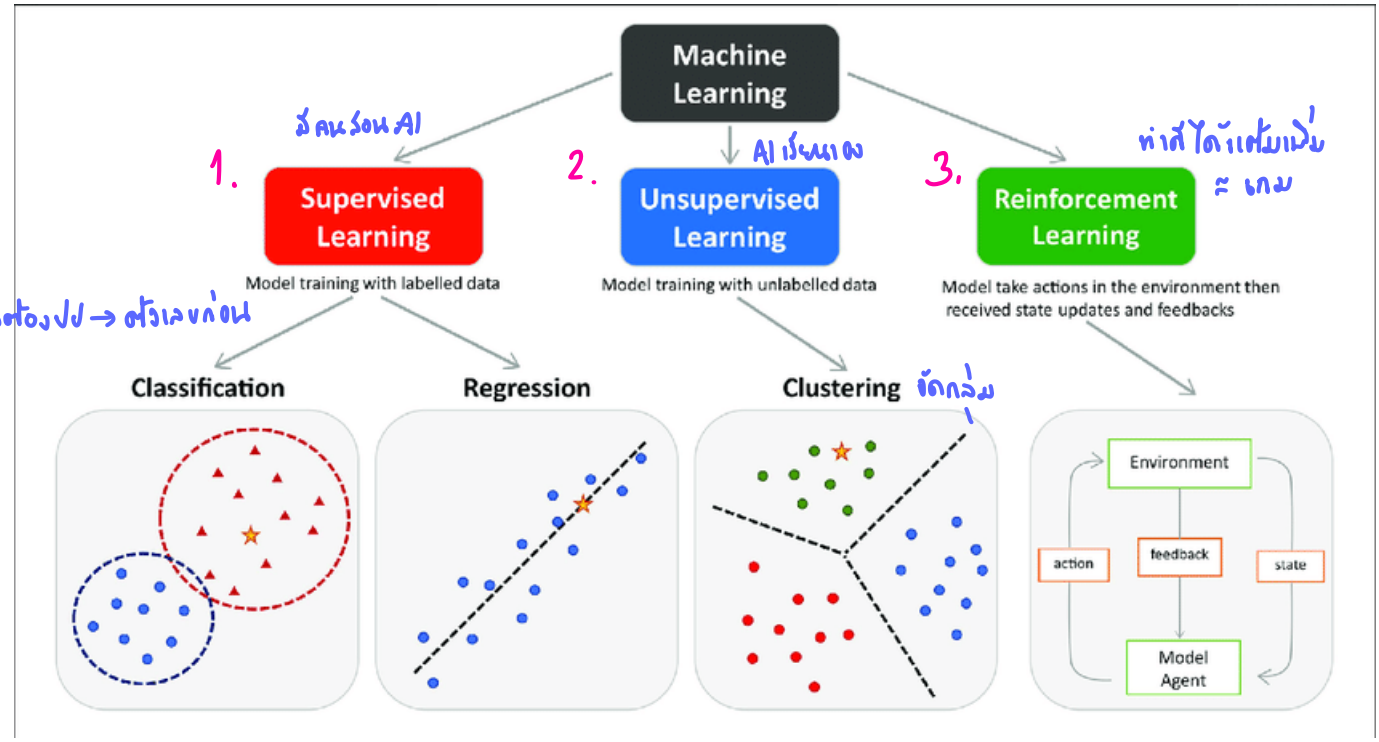
Module 1: Introduction to Data Science Algorithms and Python

Overview of Data Science Algorithms

อัลกอริทึมในการวิเคราะห์ข้อมูลเป็นเครื่องมือที่ใช้ในการวิเคราะห์และสกัดข้อมูลเพื่อหาแนวโน้ม ทำการพยากรณ์ และช่วยในการตัดสินใจ ตัวอย่างของอัลกอริทึมที่ใช้บ่อยที่ประกอบด้วย:

1. **Regression algorithms** สำหรับการพยากรณ์ค่าตัวเลข
2. **Classification algorithms** สำหรับการจำแนกข้อมูลเป็นกลุ่ม
3. **Clustering algorithms** สำหรับการระบุกลุ่มที่คล้ายกันภายในข้อมูล
4. **Basket Analysis** เพื่อค้นหาความสัมพันธ์ระหว่างรายการในธุรกรรม
5. **Time Series Analysis** เพื่อเข้าใจข้อมูลที่ถูกจัดลำดับตามเวลา (ไม่ได้เรียน)



(image from: https://www.sabrepc.com/blog/Deep-Learning-and-AI/Top-10-Popular-Data-Science-Algorithms-and-Examples-Part-1)

Example:

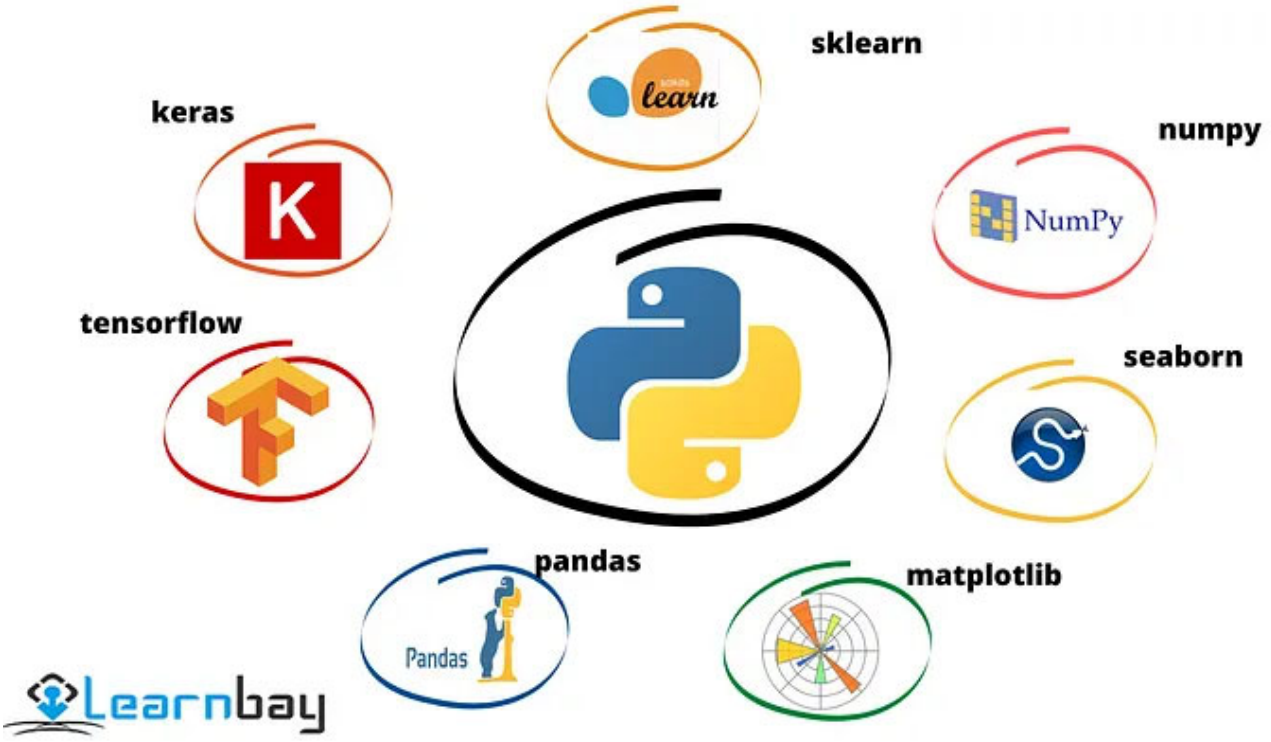
สมมติว่าคุณกำลังทำงานให้กับบริษัทอีคอมเมิร์ซ และคุณต้องการทำนายว่าลูกค้าจะซื้อสินค้าไหนโดยพิจารณาจากประวัติการเรียกดูและการซื้อสินค้าของพวกเขา

→ เป็น classifi (ซื้อ / ไม่ซื้อ)

Python Libraries for Data Science

ภาษา Python มีไลบรารีที่หลากหลายสำหรับงานวิเคราะห์ข้อมูล บางไลบรารีที่สำคัญประกอบด้วย:

- **Pandas:** สำหรับการจัดการและวิเคราะห์ข้อมูล มีรูปแบบข้อมูลที่เรียกว่า DataFrame ซึ่งเป็นโครงสร้างข้อมูลที่มีประสิทธิภาพ
- **NumPy:** สำหรับการคำนวณตัวเลข เป็นการดำเนินการแบบอาร์เรย์และฟังก์ชันทางคณิตศาสตร์
- **Scikit-learn:** สำหรับงานเรียนรู้เครื่องจักร เช่น การจัดกลุ่ม การพยากรณ์ การจำแนก และอื่น ๆ
- **Matplotlib:** สำหรับการแสดงข้อมูลและสร้างกราฟและแผนภูมิ



(image from: https://medium.com/@learnbay/python-libraries-for-data-analysis-and-modeling-in-data-science-c5c994208385)

Pandas

Pandas ให้โครงสร้างข้อมูลเช่น DataFrame และเครื่องมือสำหรับการจัดการและวิเคราะห์ข้อมูล

Example: การโหลดและสำรวจชุดข้อมูล CSV ด้วย Pandas

```
import pandas as pd
data = pd.read_csv('data.csv')
print(data.head())
```

NumPy

NumPy ใช้สำหรับการคำนวณตัวเลขและมีการสนับสนุนสำหรับอาร์เรย์และเมทริกซ

Example: การสร้างอาร์เรย์ NumPy และการดำเนินการพื้นฐาน

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr * 2)
```

Scikit-learn

Scikit-learn เป็นไลบรารีเรียนรู้เชิงเครื่องที่ให้เครื่องมือสำหรับการจำแนก การทำนาย การจัดกลุ่ม และอื่น ๆ

Example: การสร้างโมเดลการทำนายเชิงเส้นแบบง่าย

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
X = [[1], [2], [3]]
y = [2, 4, 6]
model.fit(X, y)
```

Matplotlib

Matplotlib ใช้สำหรับสร้างการแสดงผลและแผนภูมิ

Example: การสร้างแผนภูมิกระจายด้วย Matplotlib

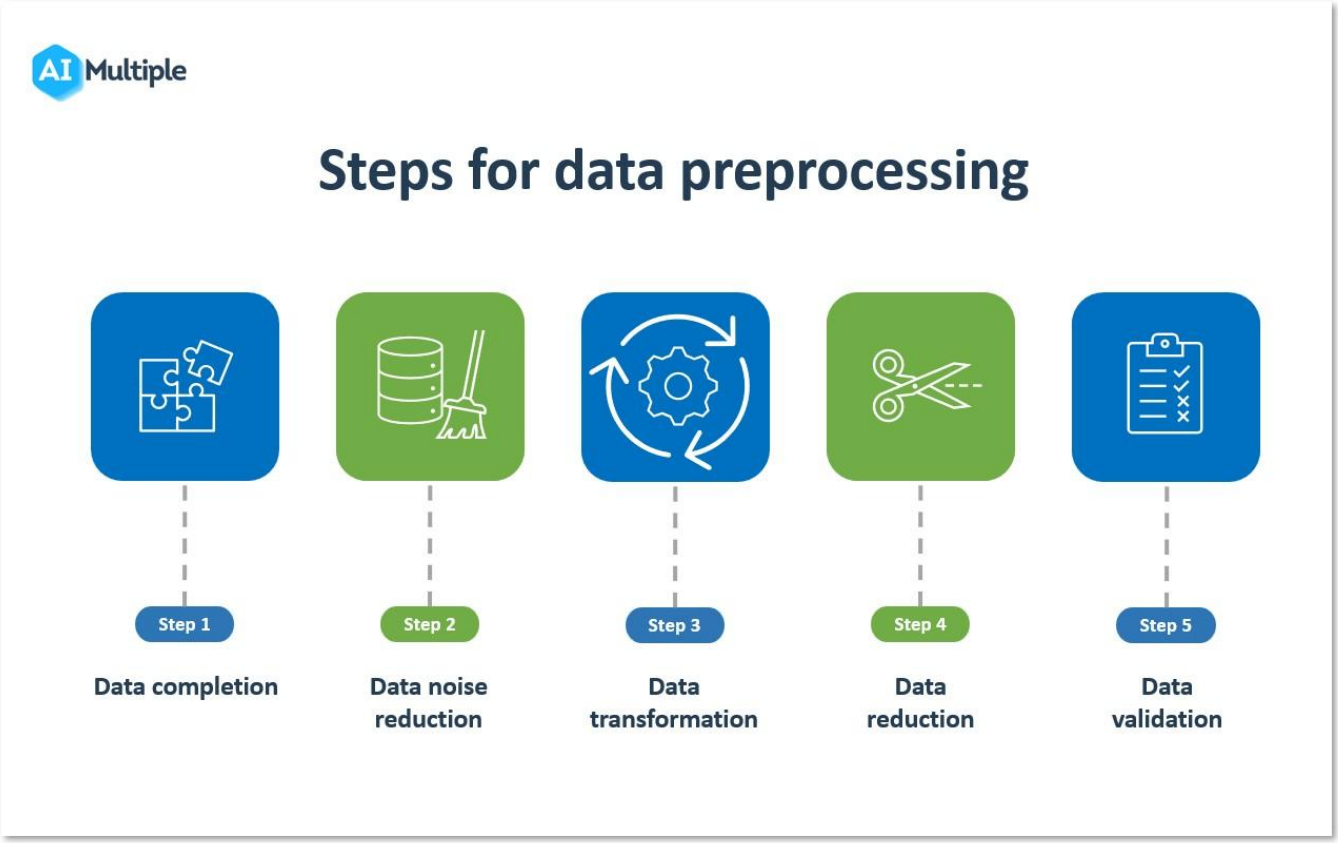
```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 3, 5]
plt.scatter(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

Data Preprocessing and Cleaning

ข้อมูลไม่ดี → model ไม่ดี

การประมวลผลข้อมูลเป็นกระบวนการเตรียมข้อมูลสำหรับการวิเคราะห์ ซึ่งรวมถึง:

- Handling missing values: การกรอกข้อมูลที่ขาดหายไปหรือลบแถวที่มีข้อมูลไม่ครบถ้วน
- Encoding categorical data: การแปลงตัวแปรหมวดหมู่เป็นรูปแบบตัวเลข categorical → num.
- Scaling and normalization: การให้ค่าคุณลักษณะอยู่ในช่วงเดียวกัน
- Removing outliers: การจัดการข้อมูลที่แตกต่างจากข้อมูลอื่นๆ อย่างมาก



(image from: https://research.aimultiple.com/data-preprocessing/)

Example: ใช้ Pandas เพื่อจัดการค่าที่ขาดหายไป

```
import pandas as pd

data = pd.read_csv('data.csv')
data.dropna(inplace=True)  # Remove rows with missing values
```

More Resources:

- Pandas Documentation
- NumPy Documentation
- Scikit-learn Documentation
- Matplotlib Documentation
- Data Preprocessing Techniques