

# Supervised Learning, Python Tutorial

อ. ปรัชญ์ ปิยะวงศ์วิศาล

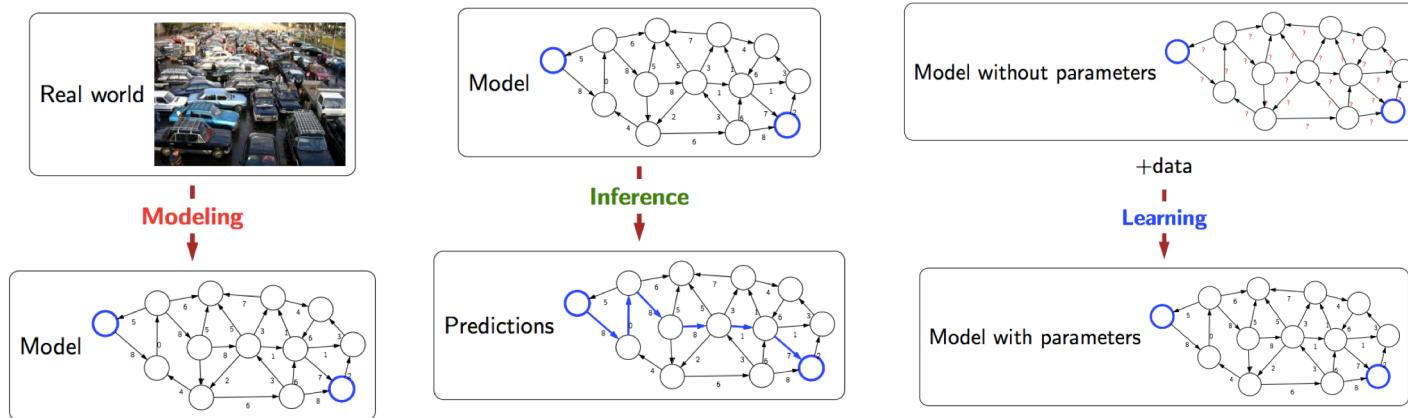
Pratch Piyawongwisal

# Today

- Recap
- Machine Learning
  - Supervised vs Unsupervised Learning
  - Classification vs Regression
- Python Tutorial

# Recap: Artificial Intelligence (AI)

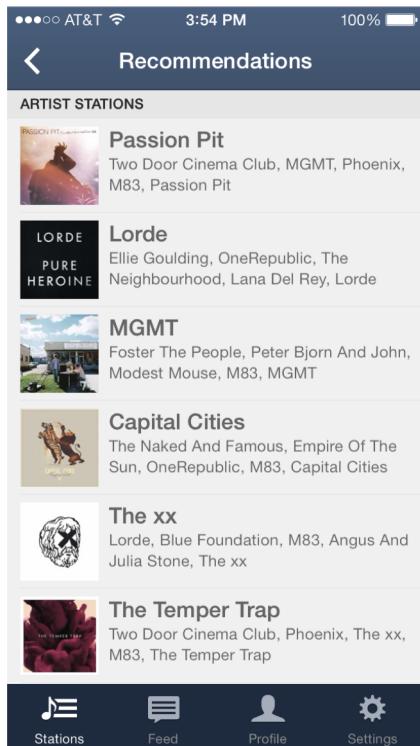
- Study of how to make “Intelligent Agents”
- Strong AI vs Weak AI
- Two sources of complexity that makes AI hard:
  - Computational complexity
  - Information complexity
- Model-Inference-Learning



# Machine Learning (ML) – Informal Definition

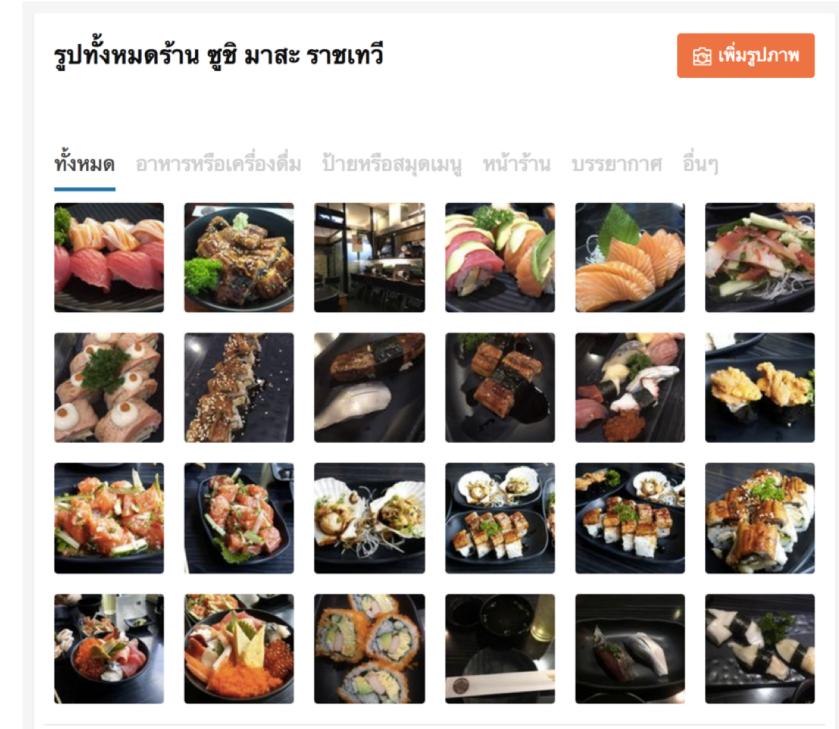
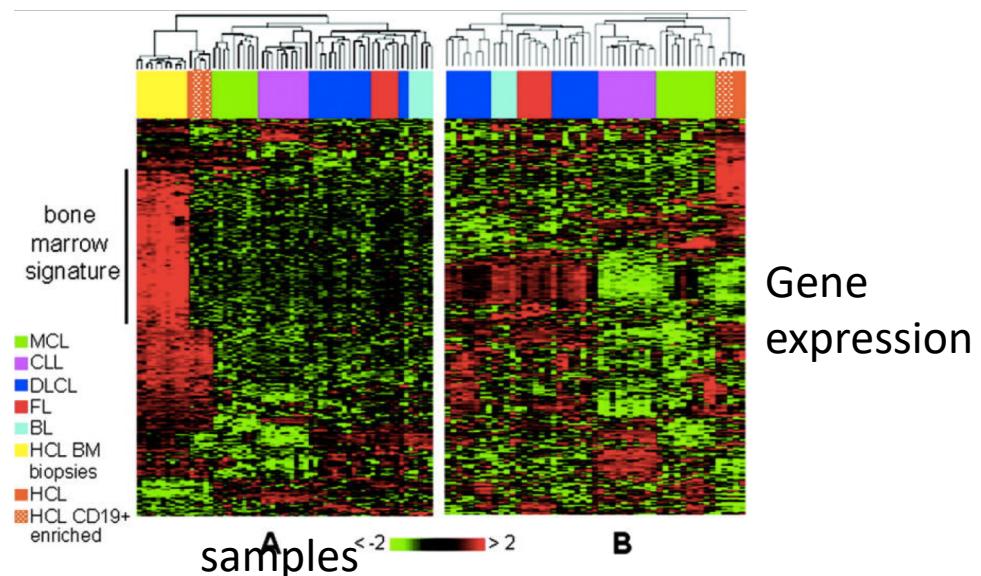
- The science of getting machines to “**learn**” from data and make predictions **without being explicitly programmed**
- การเรียนรู้ของเครื่อง คือการสอนให้เครื่องแก้ปัญหา โดยให้เครื่องเรียนรู้จากข้อมูลเอง แทนการเขียนโปรแกรมแก้ปัญหาตรงๆ
- เครื่องจะหา **pattern** ในข้อมูลเพื่อนำไปใช้ทำงานคำตอบที่ถูกต้องแม่นยำ

# Machine Learning เป็นสิ่งใกล้ตัว



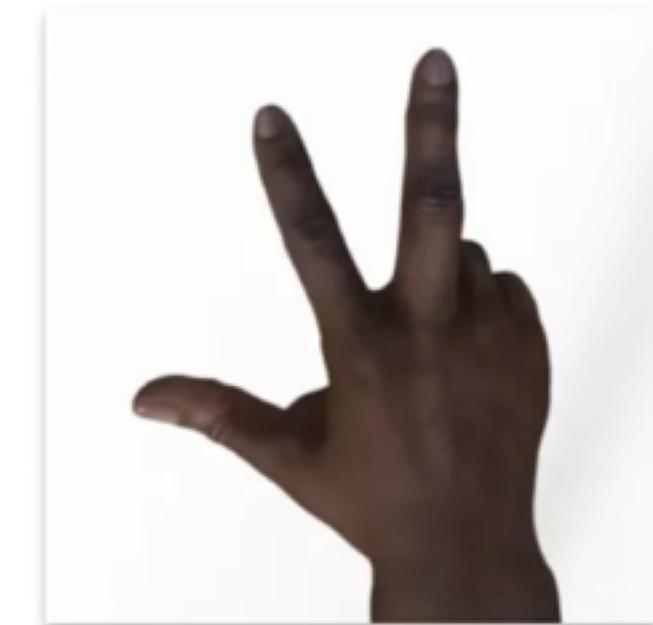
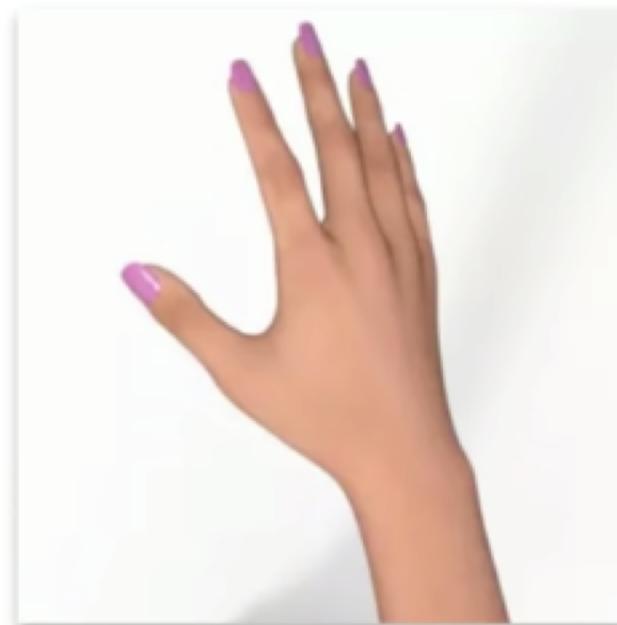
A screenshot of the Amazon.com website. The main heading is "amazon.com" with the Amazon smile logo. To the right, the text "Recommended for You" is displayed. Below this, a message says: "Amazon.com has new recommendations for you based on items you purchased or told us you own." Three book covers are shown with "LOOK INSIDE!" arrows:

- Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop
- Google Apps Administrator Guide: A Private-Label Web Workspace
- Googlepedia: The Ultimate Google Resource (3rd Edition)



# Example 1

- สมมติว่าเราจะเขียนเกม **rock-paper-scissors** โดยใช้ภาพถ่ายจากมือถือ



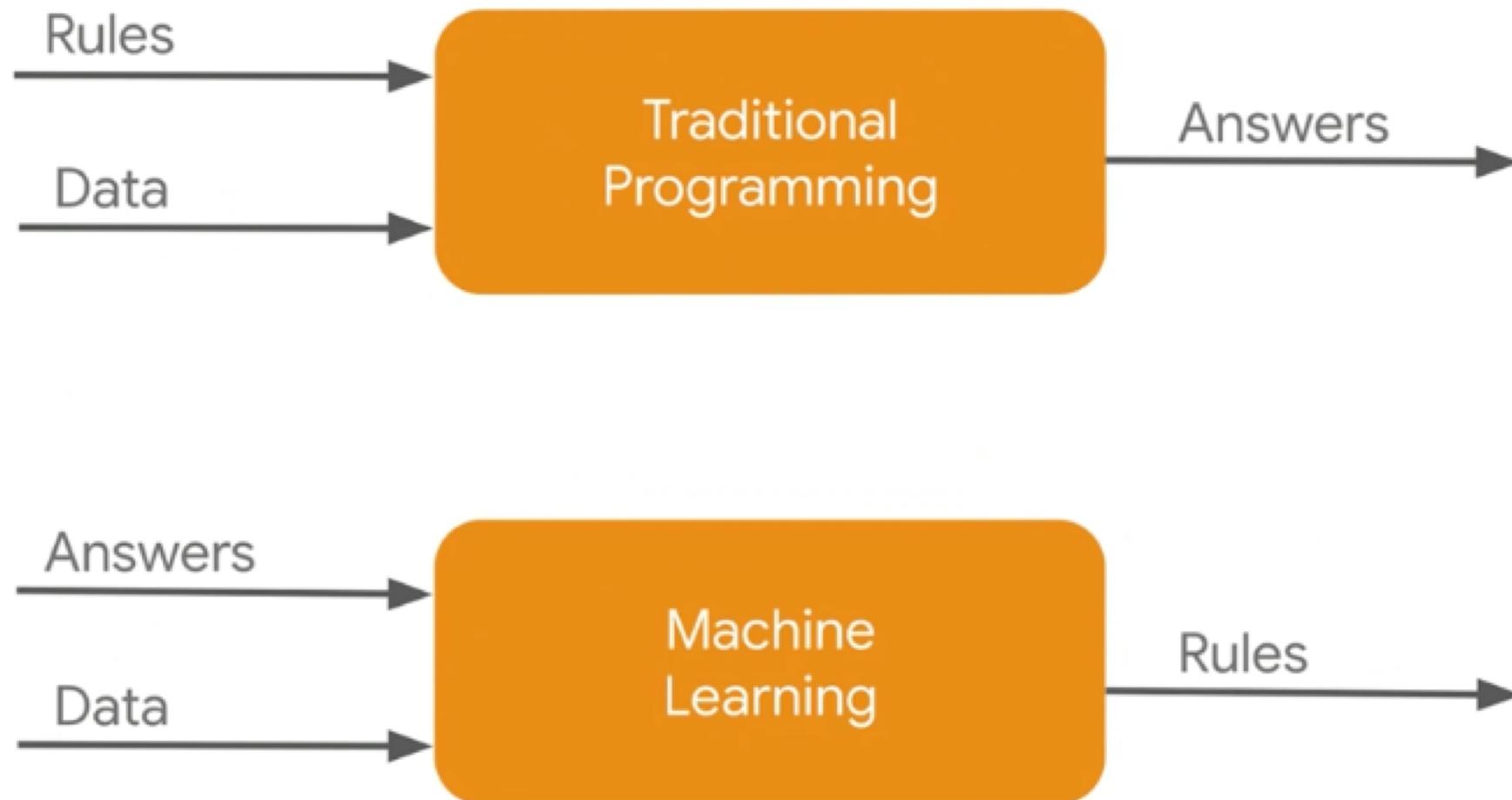
# Example 2

- เขียนโปรแกรมเพื่อคัดกรองอีเมล spam ออกจากอีเมลปกติ (ham)



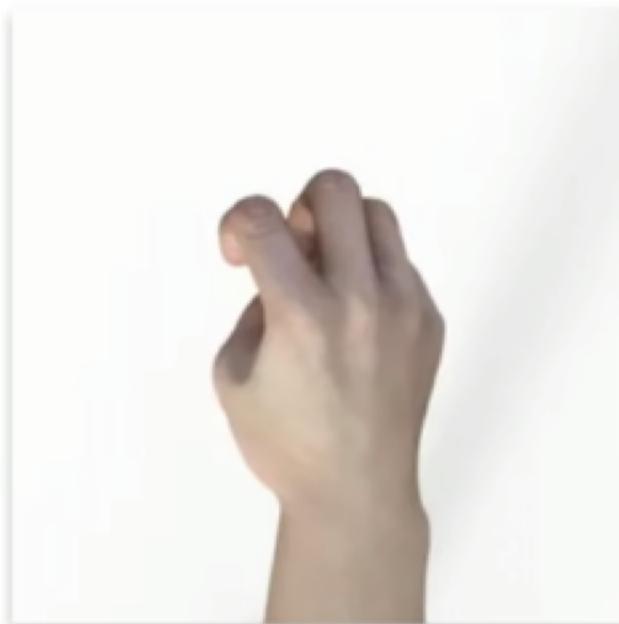
A screenshot of a Gmail inbox search results page. The search bar at the top contains "in:spam". The left sidebar shows navigation links: Compose Mail, Inbox, Sent Mail, Drafts, Spam (595) (which is highlighted), [Imap]/Deleted Items, [Imap]/Drafts, 15 more..., Contacts, and Tasks. The main content area displays a list of 15 spam messages. Each message has a checkbox, a star icon, and a preview text. The first message is "Free Viagra Sample" with the preview "Get the real pills for free - Erectile Dysfun...". The second message is "Try Viagra4Free" with the preview "Age is no longer a barrier for me in bed...". The third message is "VIAGRA (c) Official Vend." with the preview "User steel.tree Brand 84% off Sale - Havi...". The fourth message is "WorldWinner Player Servi." with the preview "Play Bejeweled 2 online - Compete again...". The fifth message is "FTD Exclusive Offer" with the preview "Valentine's Day Roses from \$19.99 - Vale...". The sixth message is "Viagra Sample" with the preview "Viagra for \$0 - Free Cialis http://theirwinter...". Above the message list, there are buttons for Delete forever, Not spam, Move to, Labels, More actions, and Refresh. Below the message list, there is a link to "Delete all spam messages now (messages that have been in Spam)".

Source: <https://towardsdatascience.com/applied-text-classification-on-email-spam-filtering-part-1-1861e1a83246>

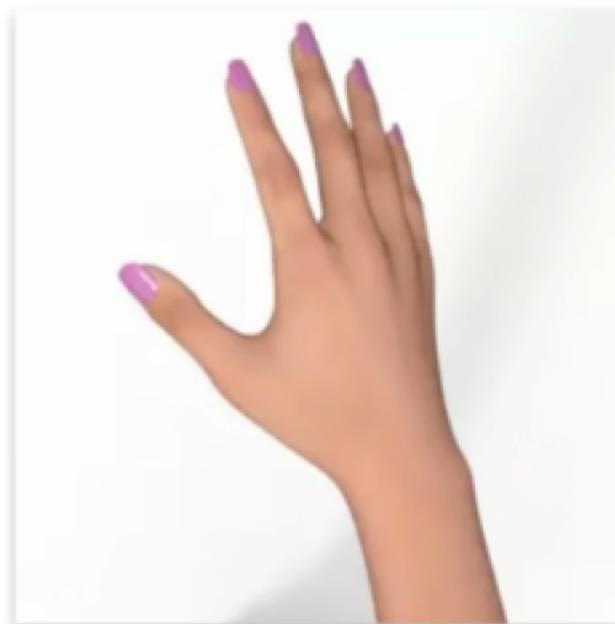


# Data and Labels

- ในการสอนเครื่อง เราจะให้มนุษย์กำหนด **label** เฉลย สำหรับข้อมูลแต่ละชิ้น



Label = ROCK



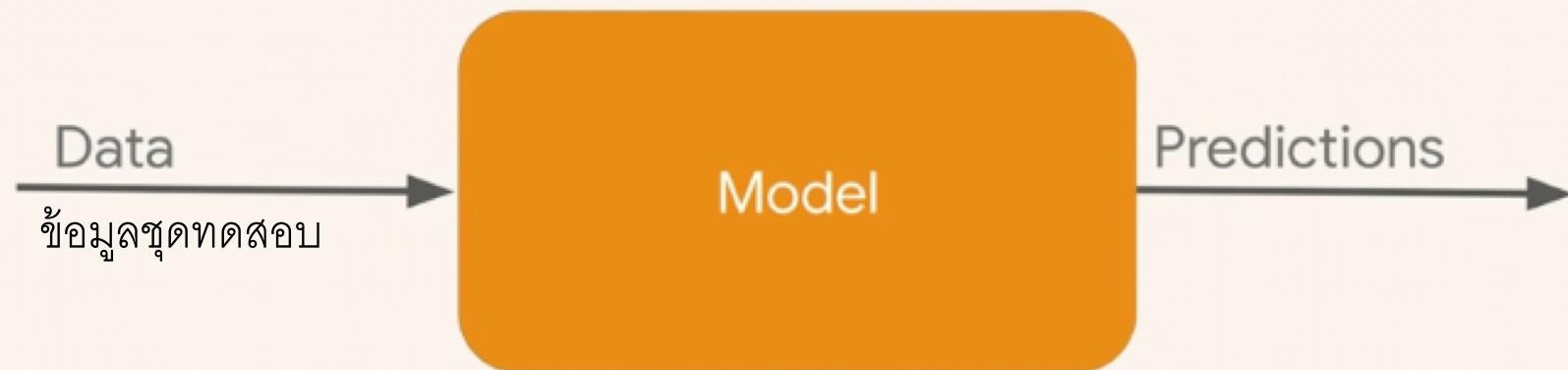
Label = PAPER



Label = SCISSORS



Training Phase



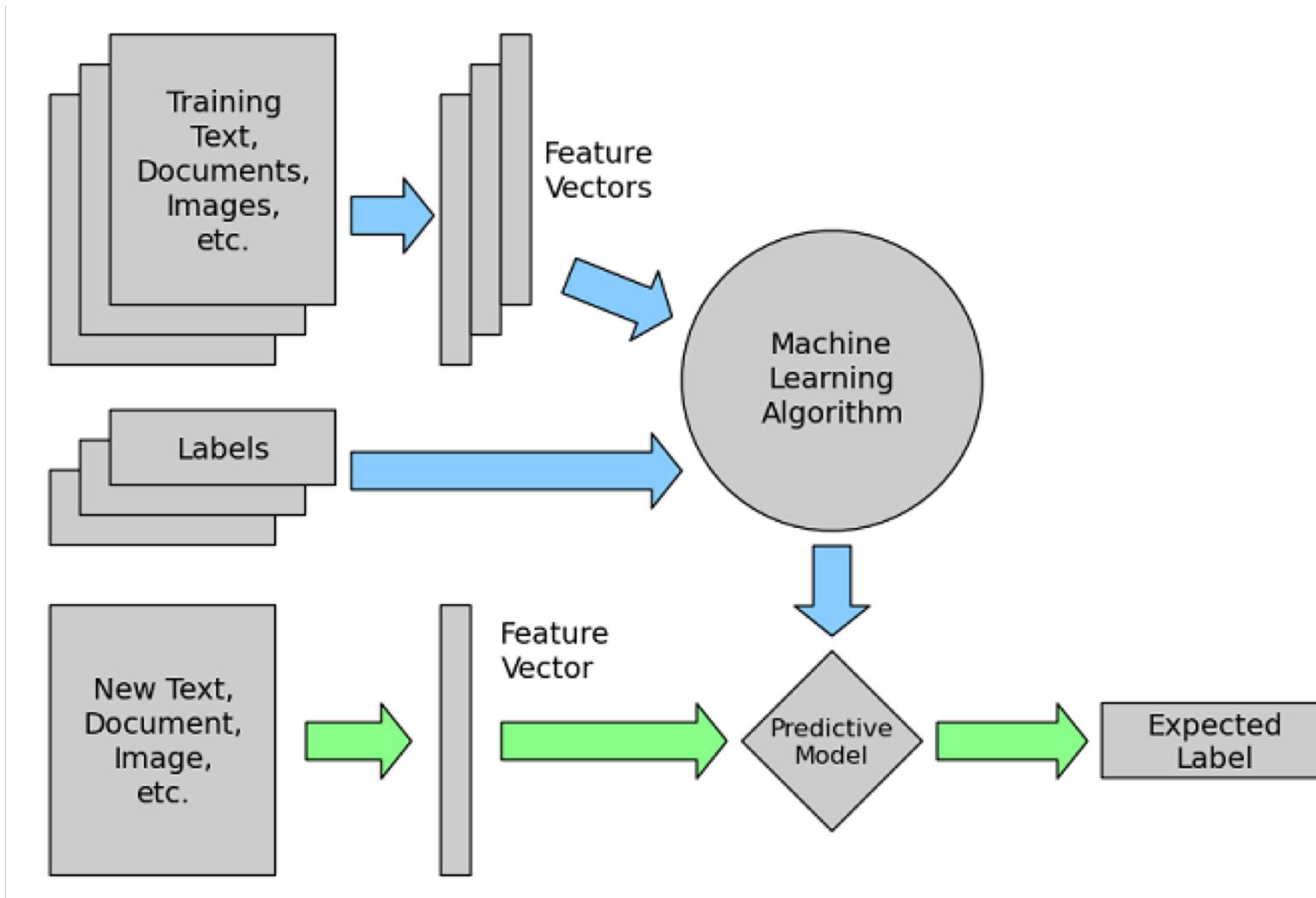
Inference Phase

เช่น Linear Regression,  
SVM, Neural Networks  
(\*ยกเว้น k-NN\*)

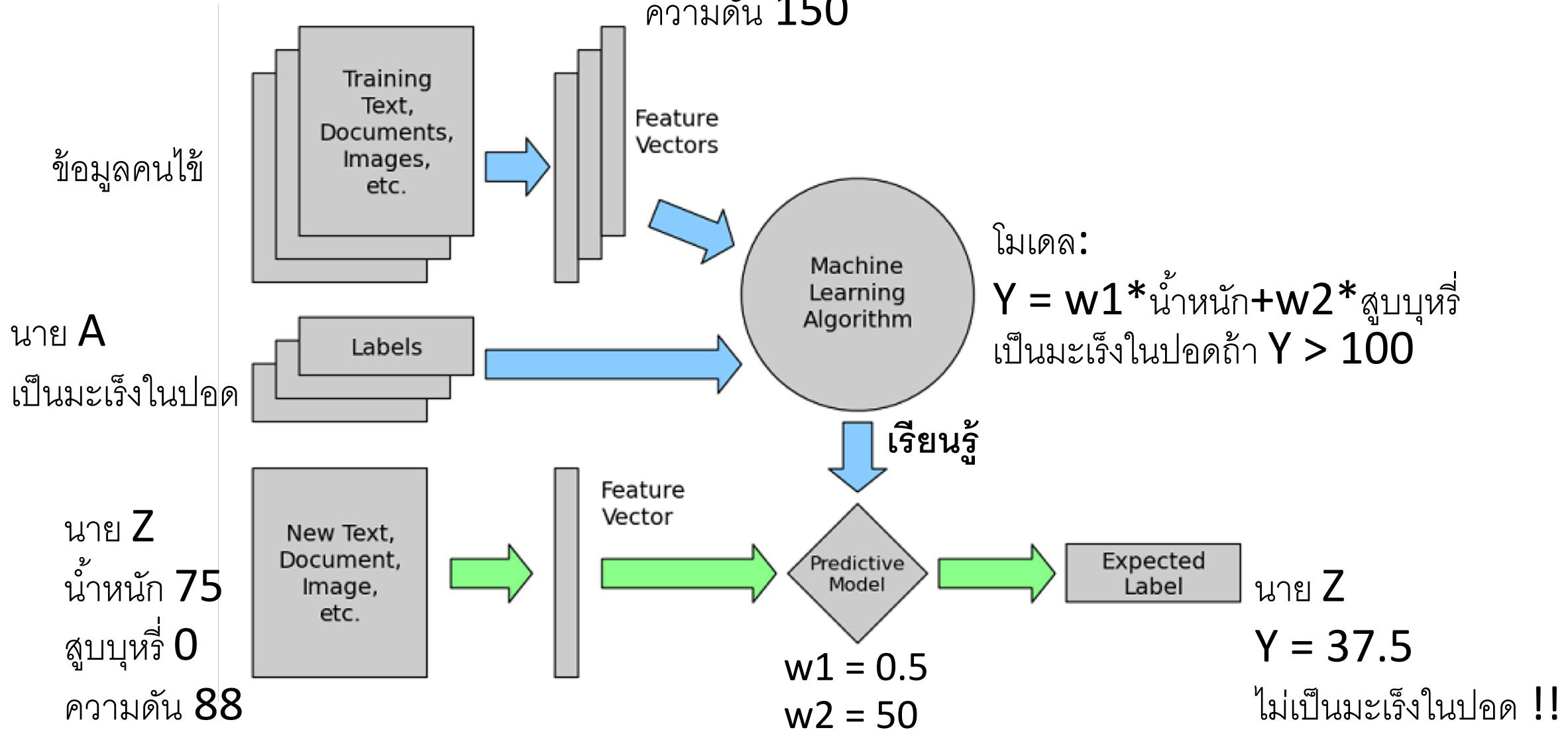
\* inference = การอนุมาน

Source: Machine Learning Zero to Hero (Google I/O'19)

# How ML works?

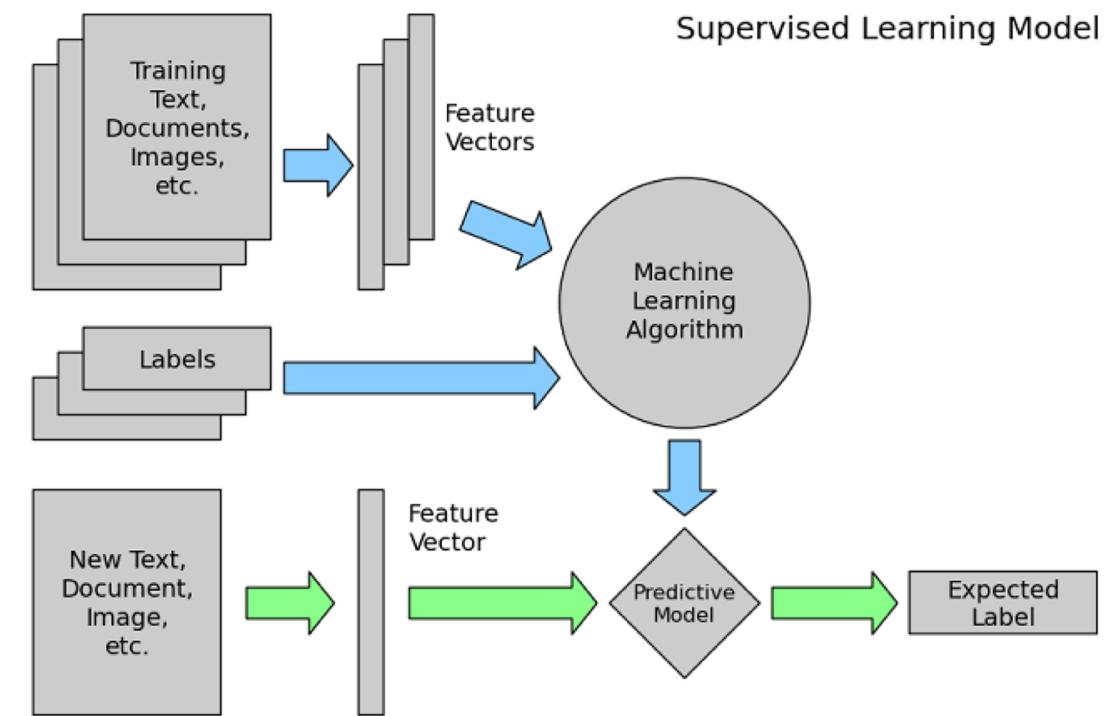


# Example



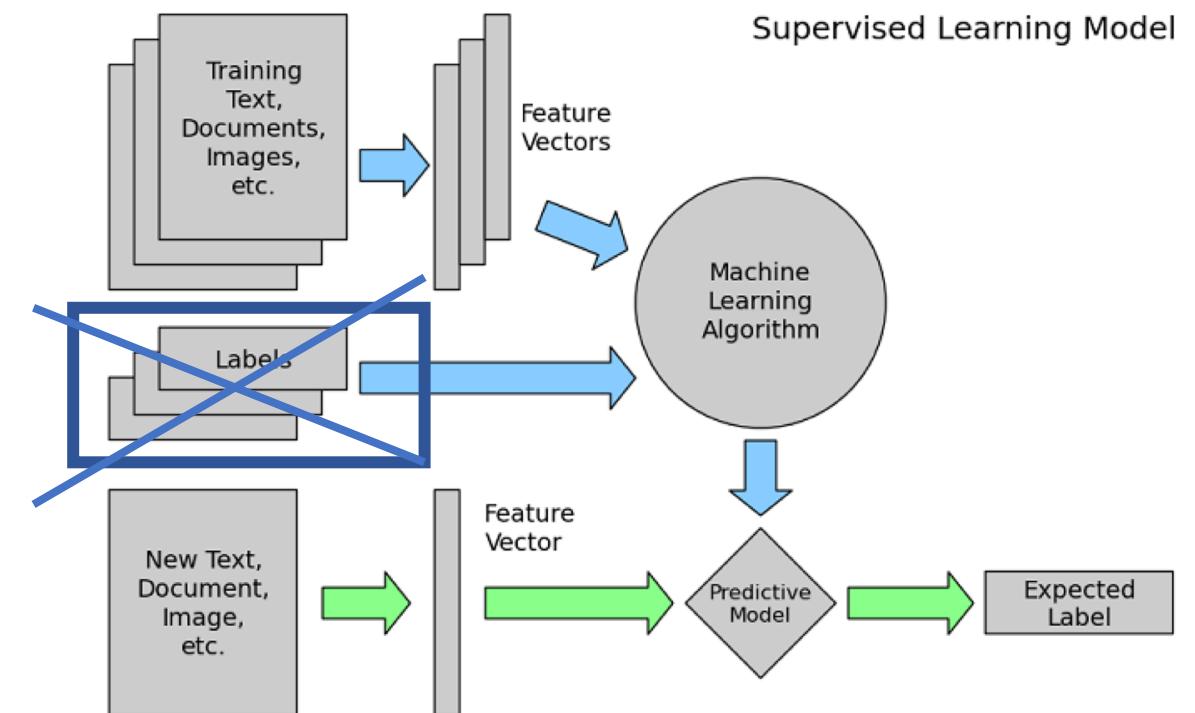
# Must-know Terminology

- Training set ข้อมูลชุดสอน
- Test set ข้อมูลชุดทดสอบ
- Feature คุณลักษณะเด่น
- Class Label ชนิดที่จำแนก
- Model แบบจำลองคณิตศาสตร์
- Predictor ตัวทำนาย
- Classifier ตัวจำแนกชนิด
- Training Error ค่าคาดเคลื่อนในการฝึก
- Testing Error ค่าคาดเคลื่อนในการทดสอบ



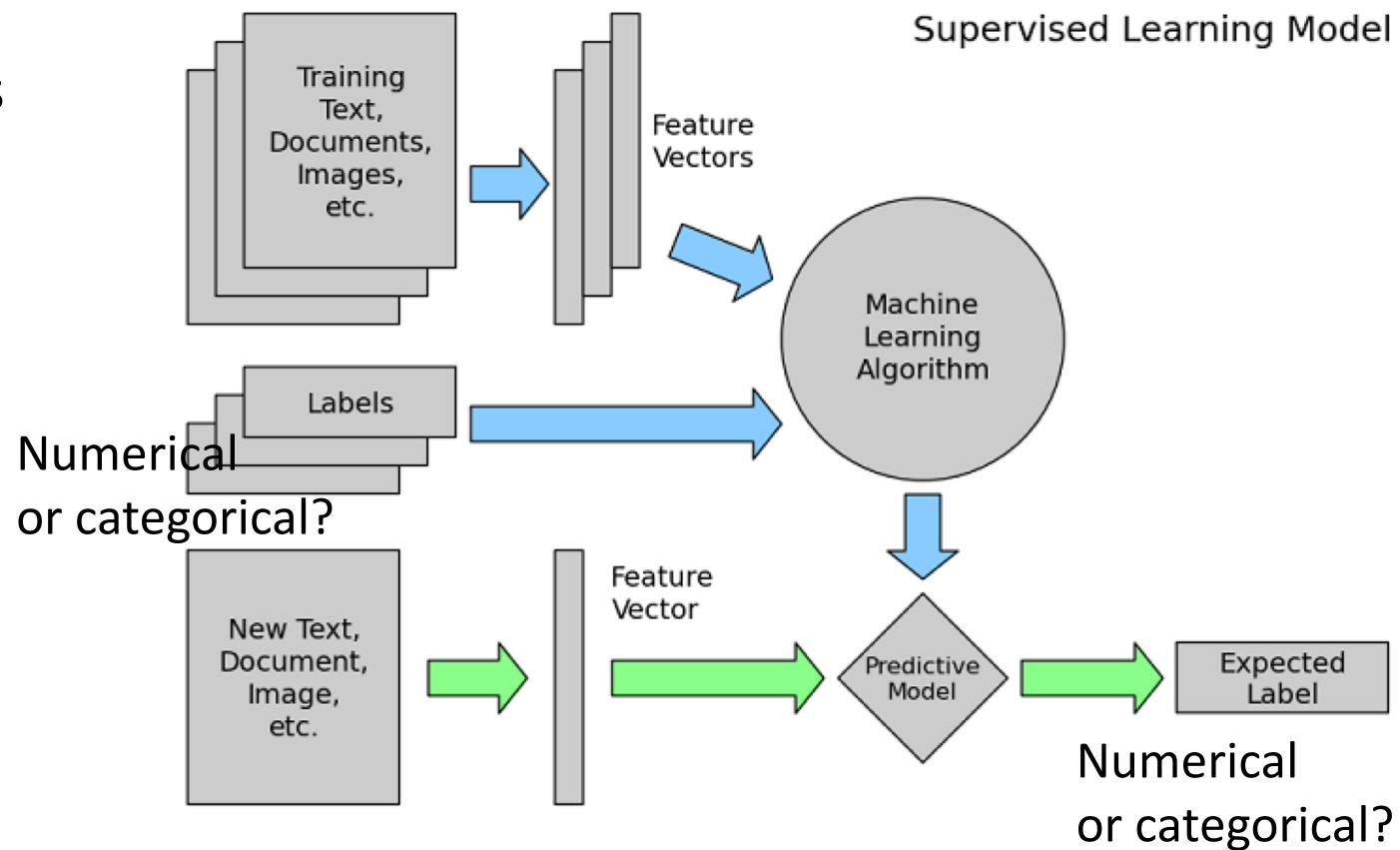
# 2 Types of ML Systems

- Supervised Learning
  - Trained **with** human supervision
  - Training set has **labels**
- Unsupervised Learning
  - Trained **without** human supervision
  - No class labels

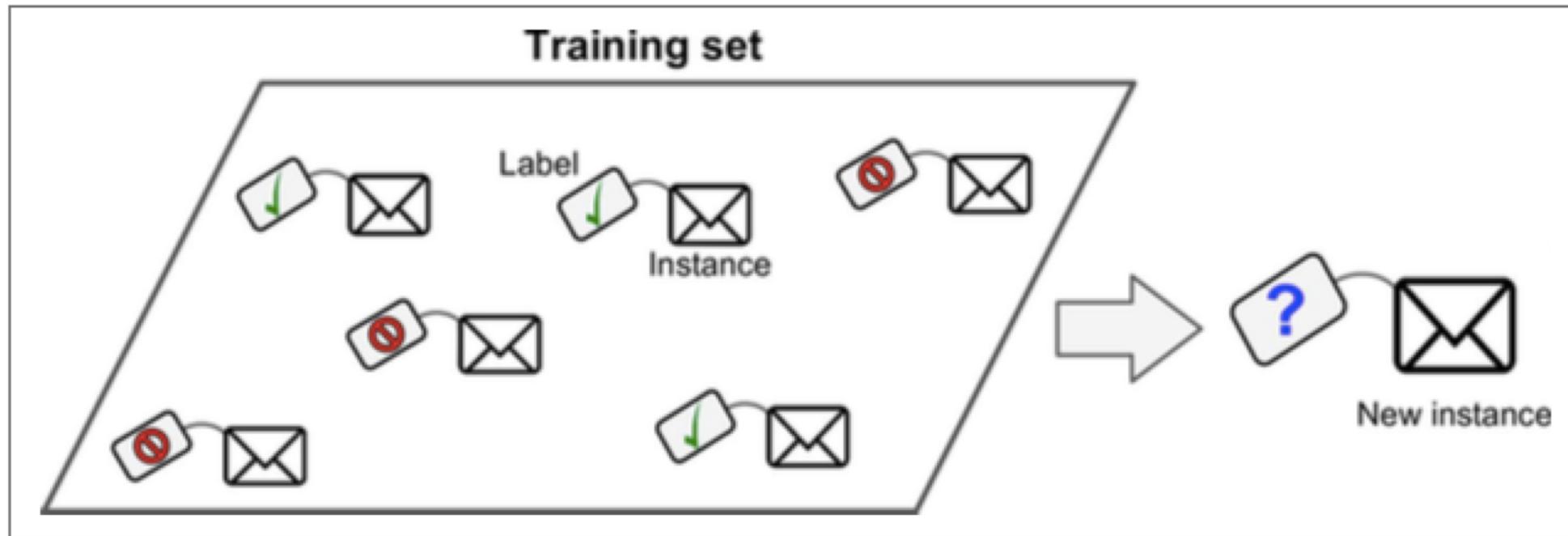


# Common Supervised Learning Tasks

- Classification
  - Predicts class labels/categories
  - Example:
    - cancer/no cancer
    - husky/malamute/chiba/akita
    - 1/2/3/4/5/6/7/8/9/0
- Regression
  - Predicts continuous values
  - Example:
    - House pricing
    - Temperature



# Example – Spam Filtering



Is this Classification or Regression?

# Supervised Learning Algorithms

- k-Nearest Neighbors (k-NN)
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVM)
- Decision Trees
- Neural Network

# ML เหนาะกับงานประเกทได

- ปัญหาที่ต้องแก้ด้วยโปรแกรมที่มีกฎและเงื่อนไขจำนวนมาก
  - e.g. rock paper scissors, handwritten digit recognition
- ปัญหาที่มีความซับซ้อนสูงและไม่สามารถหาคำตอบได้ด้วยวิธีการเดิมๆ
  - e.g. stock market prices, medical diagnosis
- ปัญหาที่ขึ้นกับสภาพแวดล้อมที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา
  - e.g. spam filtering
- ปัญหาการสั่งเคราะห์สารสนเทศที่เป็นประโยชน์จากการข้อมูลจำนวนมากมหาศาล
  - e.g. gene expression data

# Python Tutorial

- Download Python (3.7.x)
  - <https://www.python.org/getit/>
- Anaconda Package
  - <https://www.anaconda.com/download/>
  - Spyder, Jupyter notebook, Numpy, Pandas

# Python

- Python is an interpreted language
- Focuses on simplicity => make programmers' lives easier
- Lots of useful libraries

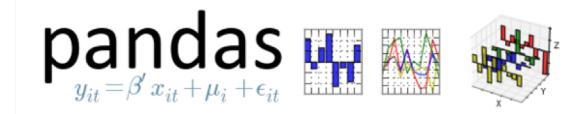


# Anaconda Package for Python

- Bundle of packages for scientific computing
- Numpy, Scikit-learn, Jupyter Notebook, Spyder
- Spyder is an IDE

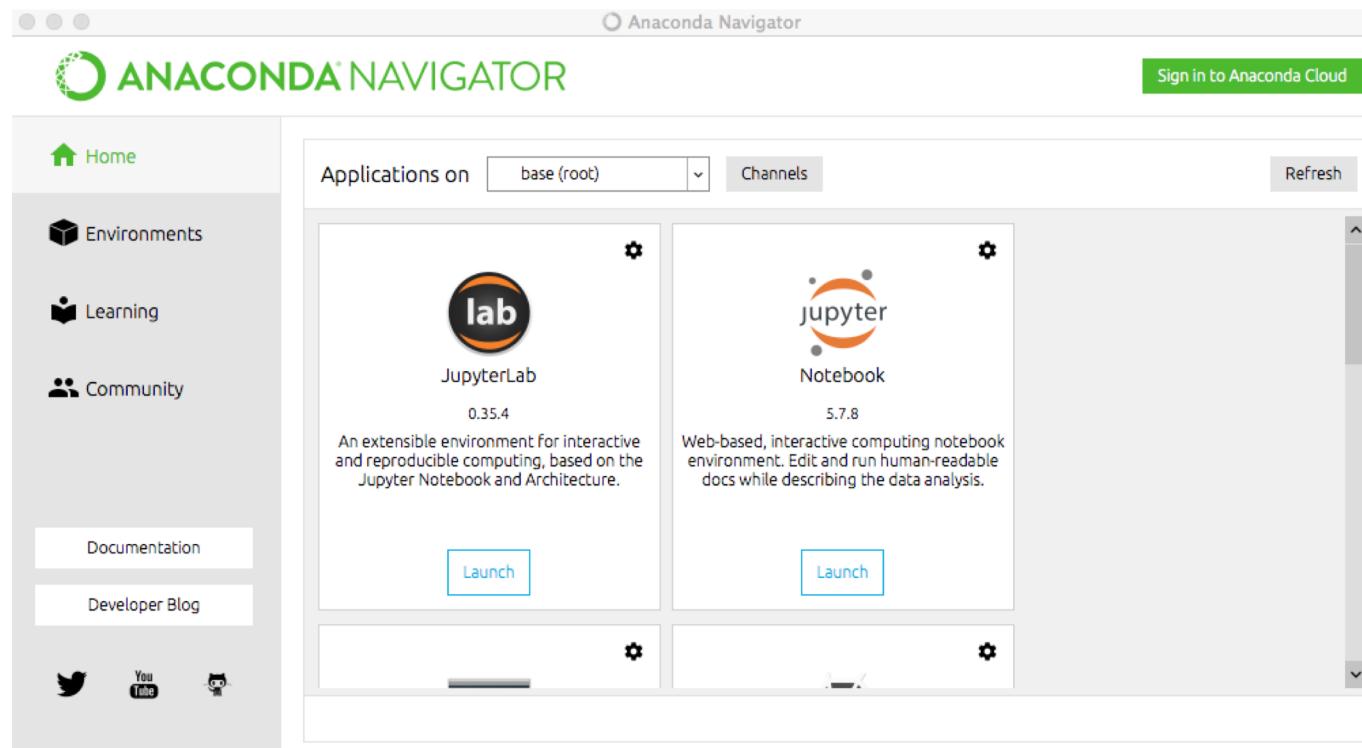


IP[y]: IPython  
Interactive Computing



# Basic Python

- Open up **Jupyter Notebook** from Anaconda Navigator
- Create New Python3 Notebook



# Jupyter Notebook Environment

jupyter lab2-knn Last Checkpoint: 07/13/2018 (autosaved) Logout Trusted Python 3

In [19]: `n_samples = len(digits.images)`  
`data = digits.images.reshape(n_samples, 64)`  
`data.shape`

Out[19]: (1797, 64)

In [40]: `from sklearn.neighbors import KNeighborsClassifier`  
  
`training_data = data[:1000]`  
`training_label = digits.target[:1000]`  
`testing_data = data[1000:]`  
`testing_label = digits.target[1000:]`  
  
`classifier = KNeighborsClassifier(n_neighbors = 5)`  
`classifier.fit(training_data, training_label)`  
  
`predictions = classifier.predict(testing_data)`  
`n_correct = sum(predictions == testing_label)`  
`n_correct/len(testing_label)`

Out[40]: 0.9548306148055207

In [43]: `# confusion matrix`  
`cm = metrics.confusion_matrix(testing_label, predictions)`  
`plt.imshow(cm)`

Out[43]: <matplotlib.image.AxesImage at 0x1a9a9a9400>



## Most used shortcuts

- H show shortcuts
- Shift + Enter run cell
- A insert cell above
- B insert cell below
- D,D delete cell
- Tab code autocomplete
- Shift + Tab function docstring

# Basic commands

- Try executing these commands in Jupyter NB cell:

- 3+4
- 1+2\*3+5/2
- 3/4
- 3./4
- 2\*\*3

# Variables

- Try executing these commands in Jupyter NB cell:

- `a = 3`
- `a`
- `print(a)`
- `print("a=", a)`
- `float(a)/7`
- `nums = [1, 2, 3]`
- `print(nums[0])`

# Math Functions

- Try executing these commands in Jupyter NB cell:
  - `exp(2.0)`
  - `sin(2.0)`
  - Error... why ?

# Importing Math Library

- import math
- math.exp(2.0)
- math.pi
- math.sin(math.pi)

# Different ways of importing

- import math
- import math as m
- from math import pi, exp
- from math import \*
  
- Useful:
  - dir(math)

# Data Structure

- List x = ["apple", "banana"]
- Set x = {"apple", "banana", "cherry"}
- Tuple x = ("apple", "banana", "cherry")
- Dictionary
  - X = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
- Find length:
  - len(x)
- Repeating list:
  - X = [1, 2] \* 3

# Control flow - if

```
from math import *
a = [0, pi/2, pi]
x = a[2]
if sin(x) == 0:
    print("zero")
elif sin(x) == 1:
    print("one")
else:
    print("huh")
```

# Control flow - for

```
for i in range(10):  
    print(i+1)
```

```
list = ["apple", "banana"]  
for i in list:  
    print(i)
```

# Function

```
def multiply(x,y):  
    z = x*y  
    return z
```

```
multiply(2,3)
```

# String

x = 'Hello World'

x[0]

x[0:2]

x[0:-1]

x[2:]

# (Advanced) map, filter, lambda expression

```
items = [1, 2, 3, 4, 5]
squared = list(map(lambda x: x**2, items))
```

```
number_list = [-5, -2, 0, 1, 4, 9]
less_than_zero = list(filter(lambda x: x < 0, number_list))
print(less_than_zero)
```

# (Advanced) zip, unpack

```
a = [1,2,3,4]  
b = [5,6,7,8]  
c = zip(a,b)  
print(list(c))
```

```
x = [[1,2,3,4], [5,6,7,8]]  
c = zip(*x) # unpacking x  
print(list(c))
```

# (Advanced) List comprehension

```
even = [i for i in range(100) if i%2==0]  
print(even)
```

```
squares = [x**2 for x in range(10)]  
print(square)
```

```
sentence = ["this", "is", "a", "list", "of", "words"]  
items = [ word[0] for word in sentence ]
```

# Numpy Basics – Matrix/array

```
import numpy as np
```

```
a = np.array([[0,-1],[1,0]])
```

```
b = np.array([3,4])
```

```
print(a)
```

```
print(b)
```

```
print(a.dot(b))
```

# Plotting with matplotlib

```
import numpy as np  
import matplotlib.pyplot as plt  
  
n = 1000  
x = np.arange(n)  
y = np.random.rand(n)  
plt.scatter(x,y)
```

# Next week

- Supervised Learning Algorithm
  - K-NN