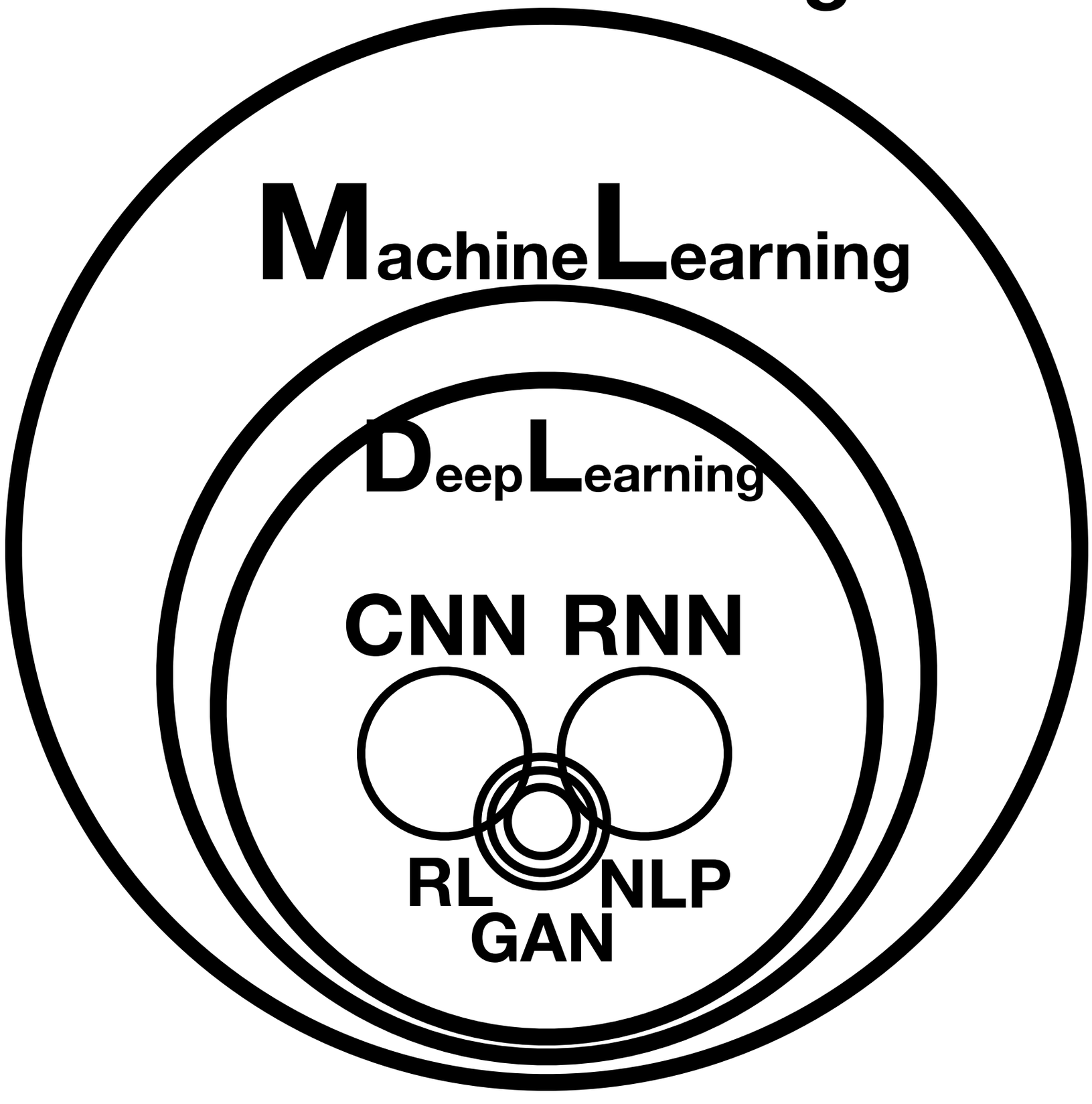
AI 入門

2020.01 IAN Fan

- AI簡介
- Keras實作
- 原理說明



Non-ML:

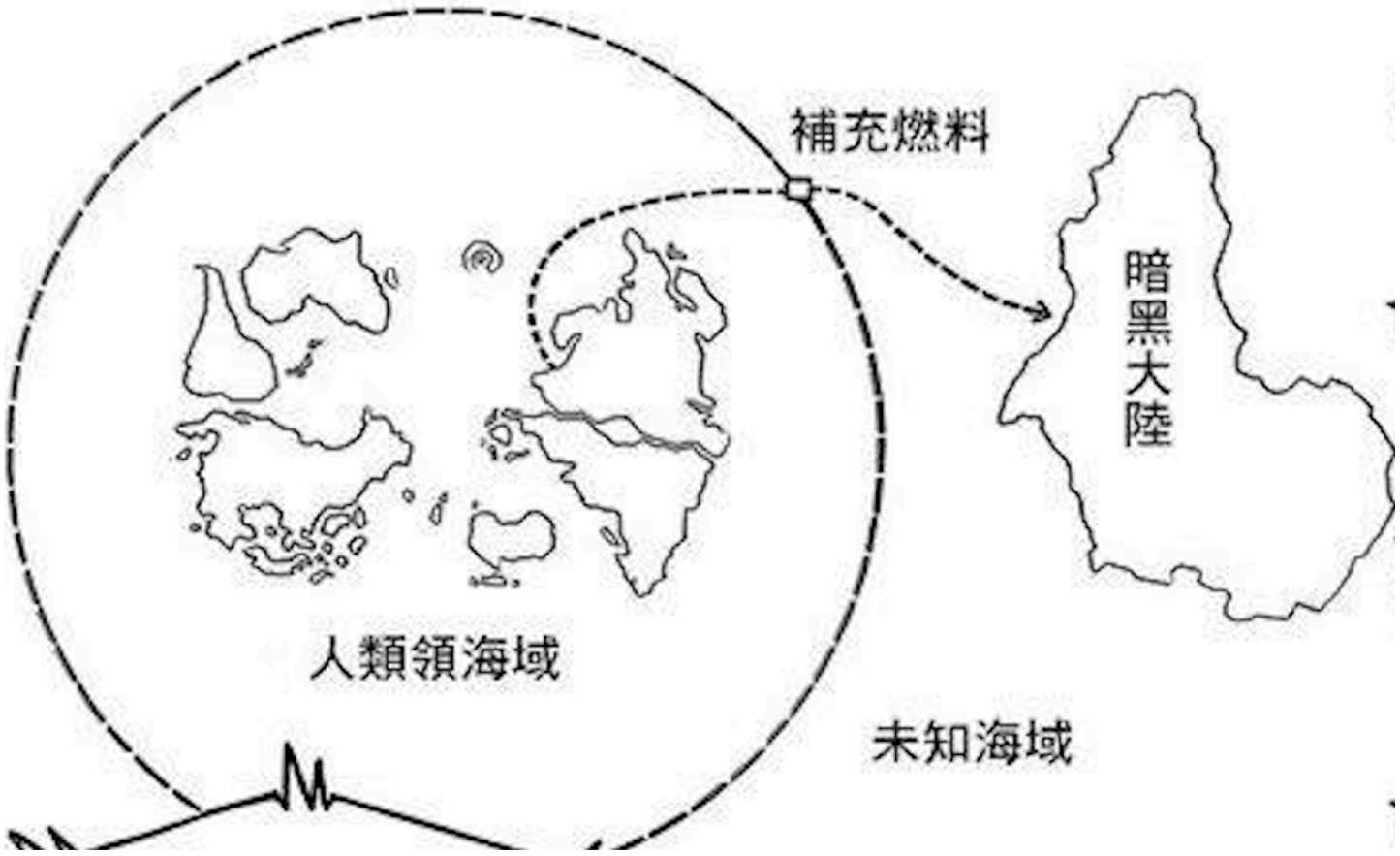
1. 自上而下
2. 規則
3.

```
if {
} else {
}
```

ML:

1. 自下而上
2. 海量數據驅動
3. 從數據中學習、修正，提高準確率

機器學習可以解決的問題：



Regression 迴歸　　**分群 Clustering**

Classification 分類　　**強化學習 Reinforcement Learning**

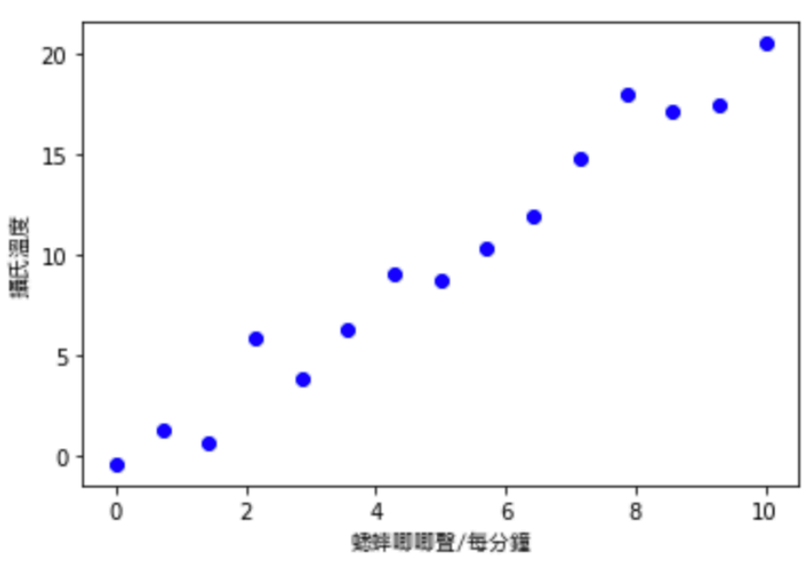
結構學習 Structure Learning

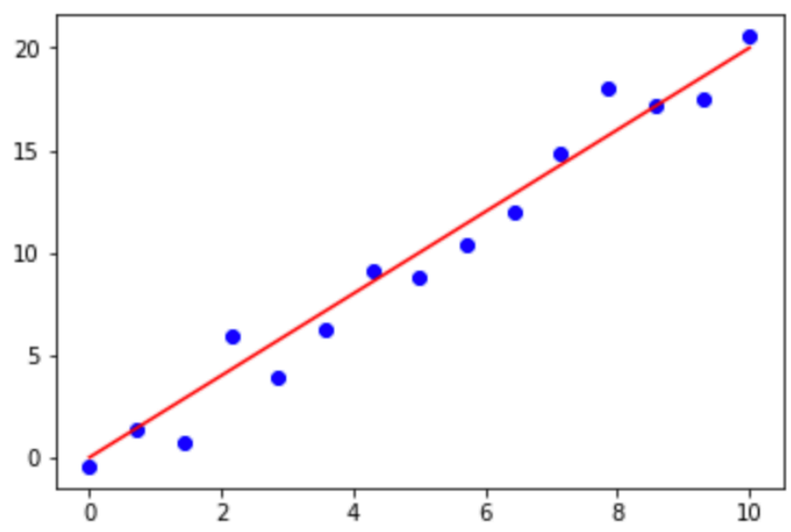
圖片參考自《獵人》

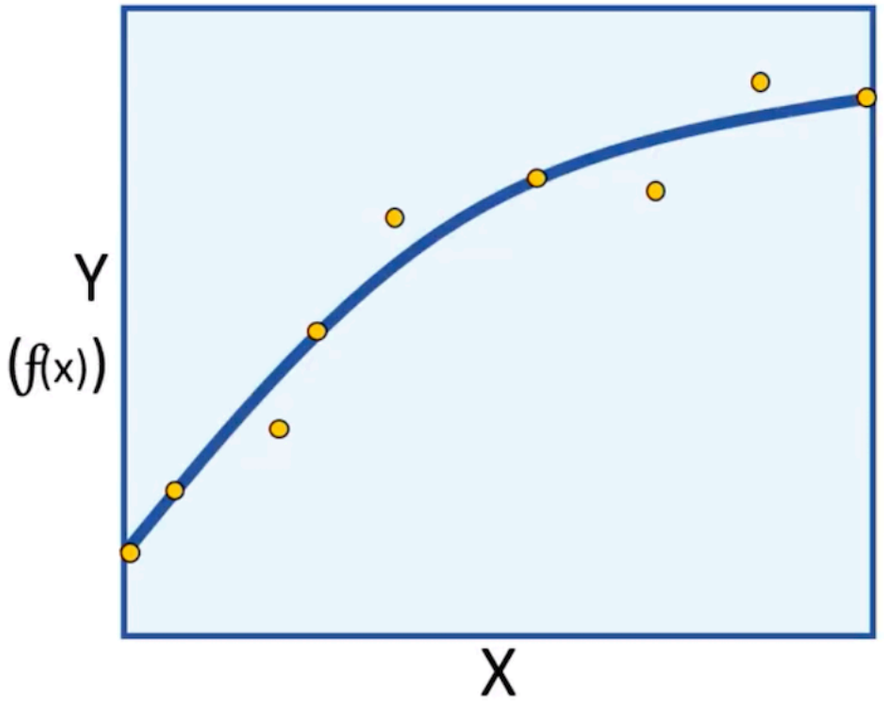
Regression 迴歸：

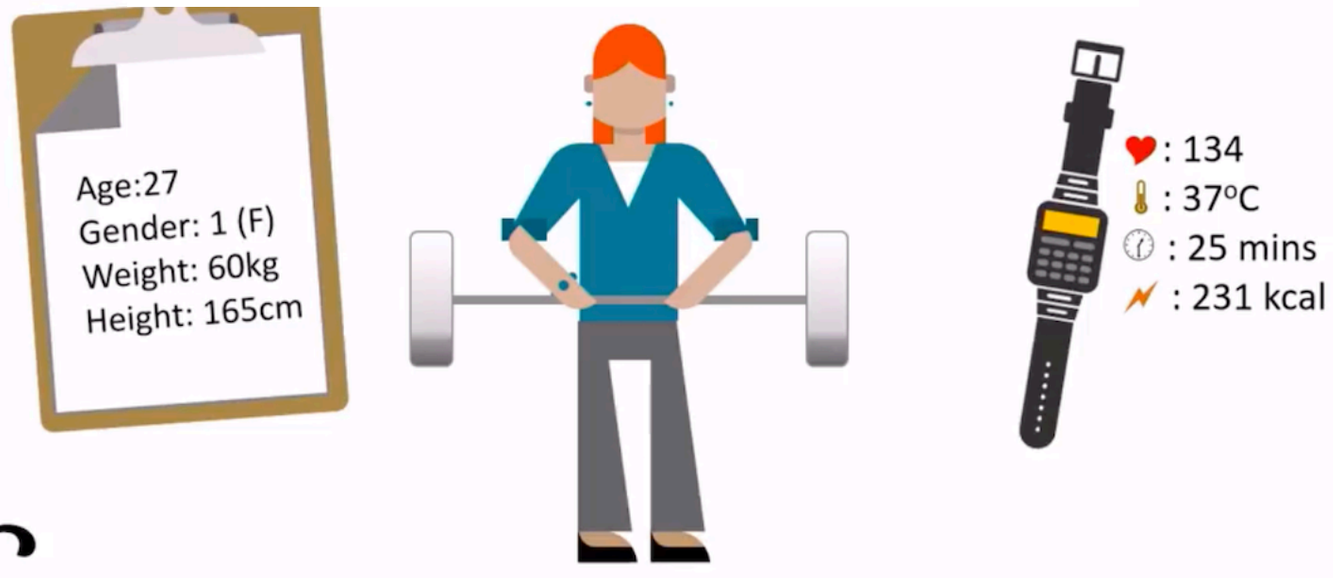
說明：預測連續數值（float）

蟋蟀每分鐘唧唧聲，預測溫度多少？
明天的氣溫為何？
我下週會獲得多少新追蹤者？









$$f([27, 1, 60, 165, 134, 37, 25]) = 231$$

年齡, 性別, 重量, 身高, 心跳, 體溫, 運動時間　　消耗卡路里

參考自Microsoft線上課程

Classification 分類：

說明：這是甲，還是乙？（二元分類）

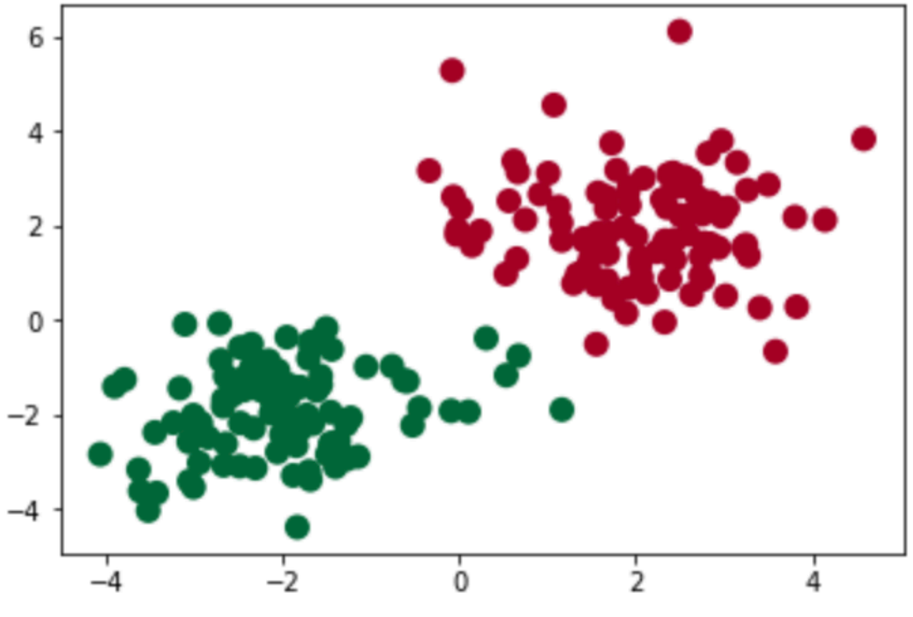
這是一張貓還是狗的圖片？

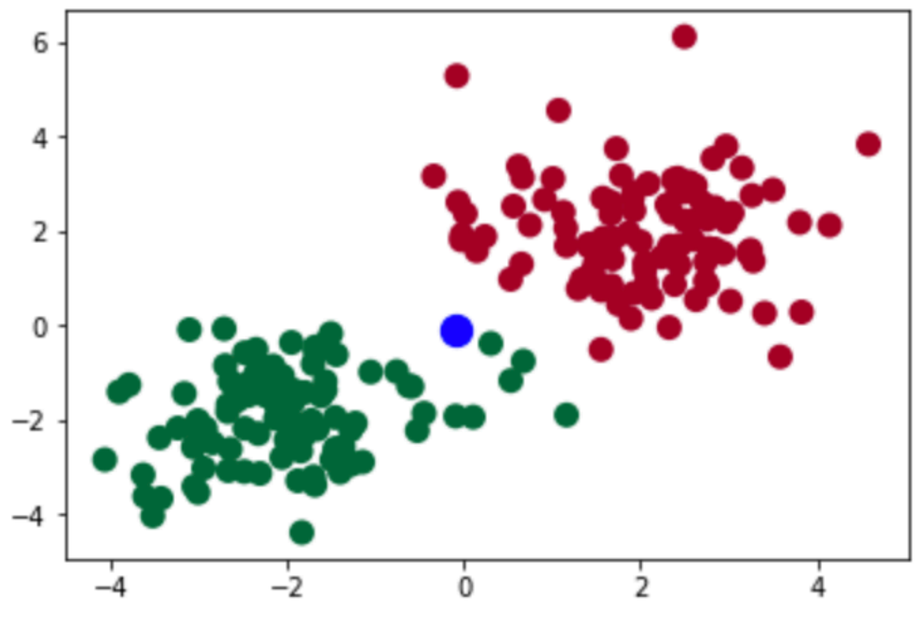
這位顧客會點還是不會點最上面的連結？

說明：這是甲、乙、丙還是丁？（多元分類）

這是哪種動物的圖片？

這則錄音裡的講者是誰？

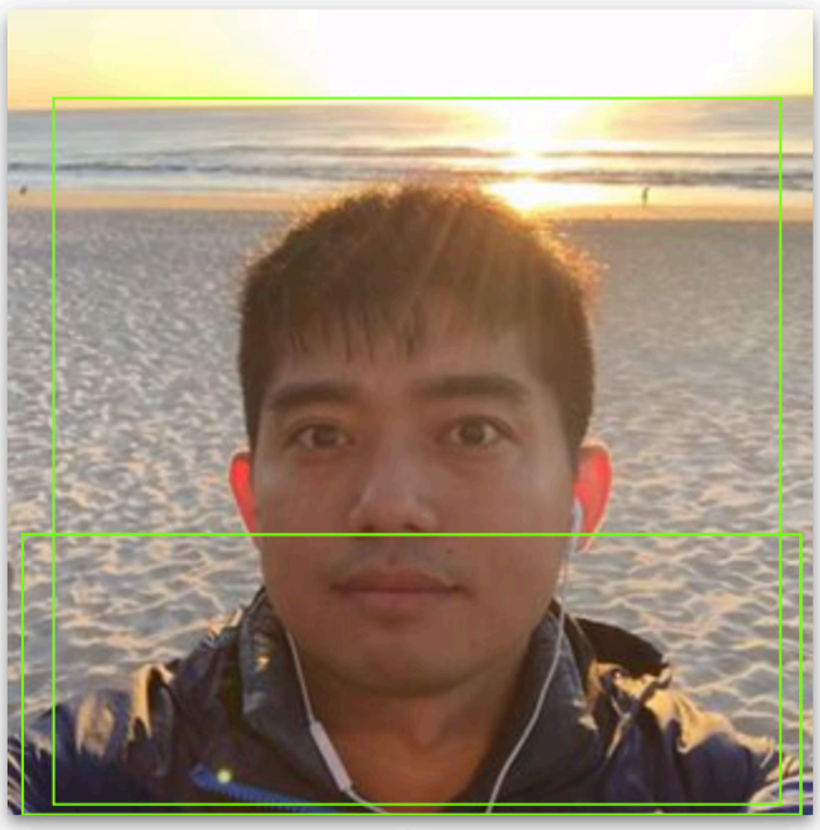




Google 圖片標籤辨識（多標籤分類）
https://cloud.google.com/vision/

Try the API

Faces | **Objects** | Labels | Web | Properties | Safe Search



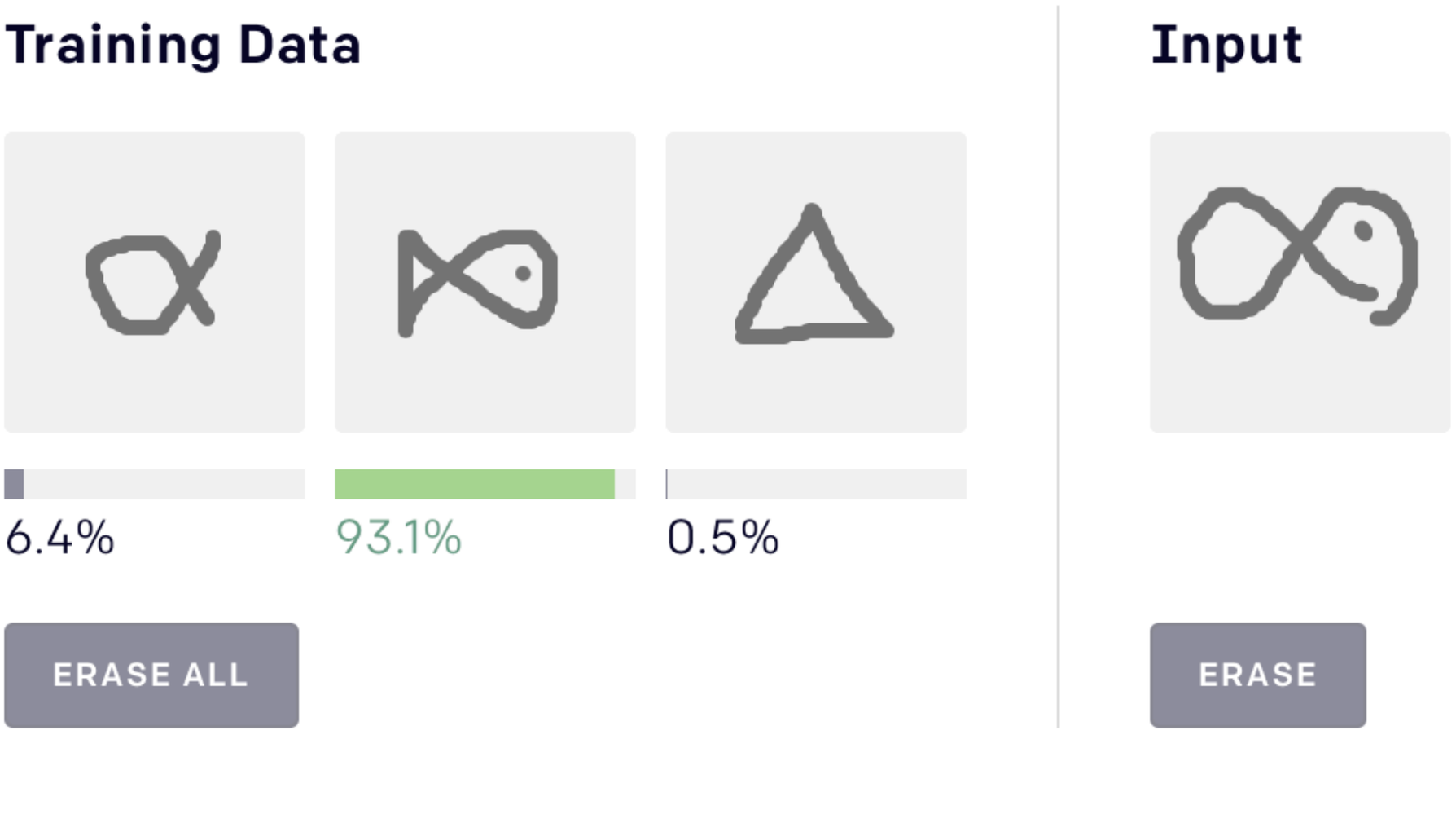
self.jpg

Man 91%

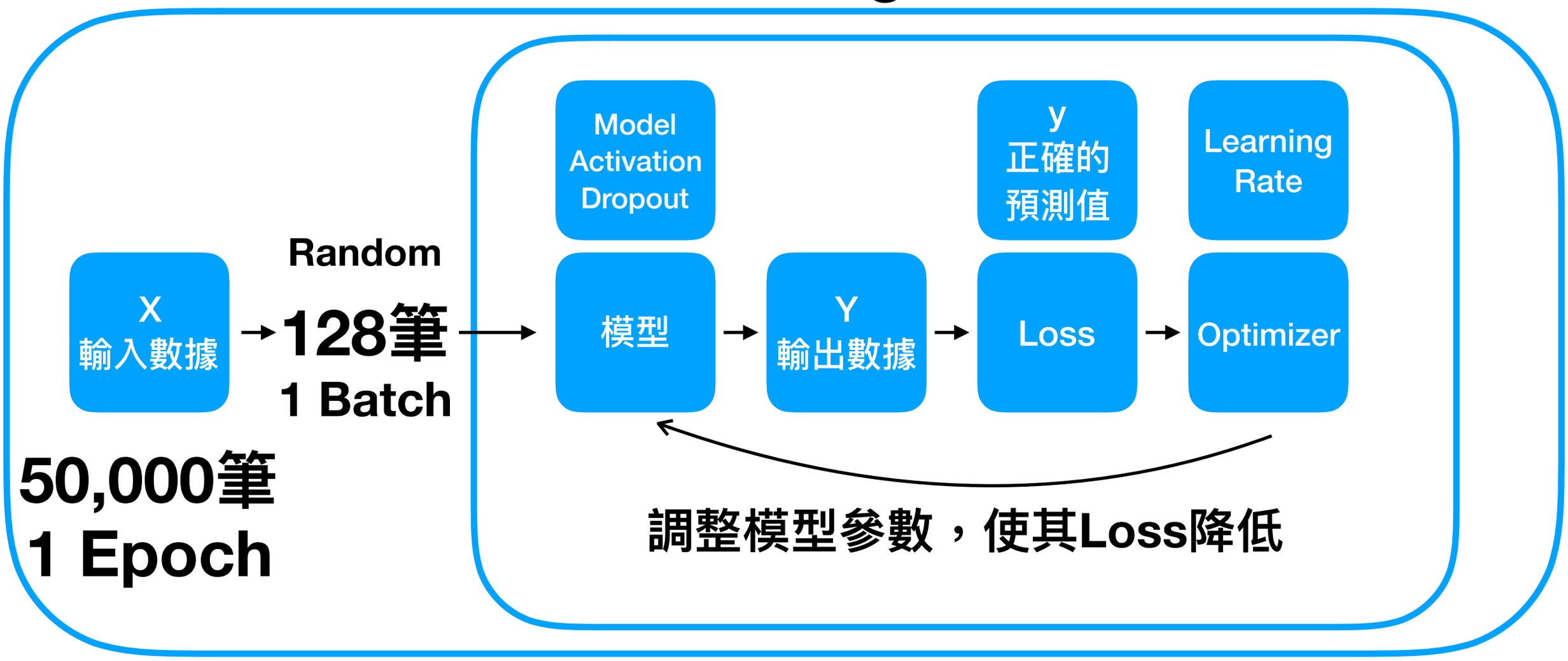
Jacket 64%

Clothing 50%

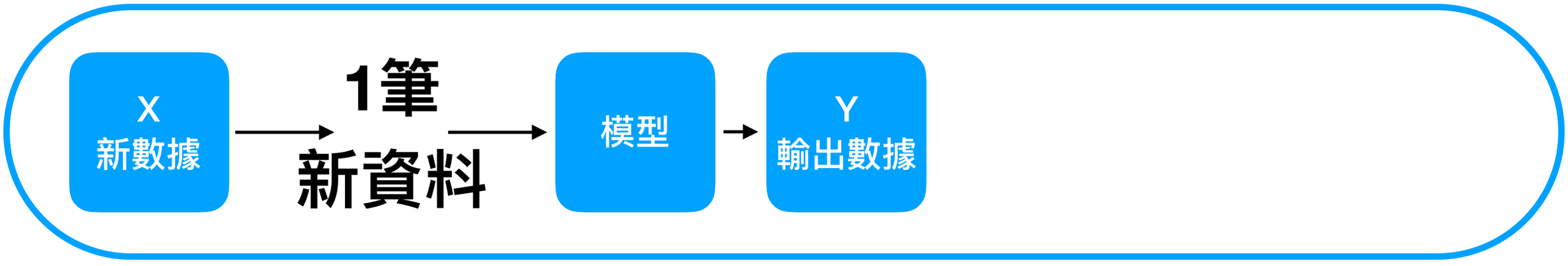
小様本畫畫分類（多類別分類）
https://blog.openai.com/reptile



Training



Prediction



直觀互動

https://playground.tensorflow.org/

11 7 8 9 10 1

Epoch 000,309

Learning rate 0.03 | Activation ReLU | Regularization None | Regularization rate 0 | Problem type Regression

DATA

Which dataset do you want to use?

2

Ratio of training to test data: 50%

3

Noise: 10

Batch size: 30

REGENERATE

FEATURES

Which pro… do you wan…

4

5 + − 2 HIDDEN LAYERS

+ − 6 neurons

+ − 6 neurons

X_1 X_2 X_1^2 X_2^2 X_1X_2 $\sin(X_1)$ $\sin(X_2)$

*This is the output from one **neuron**. Hover to see it larger.*

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

OUTPUT

12

Test loss 0.001

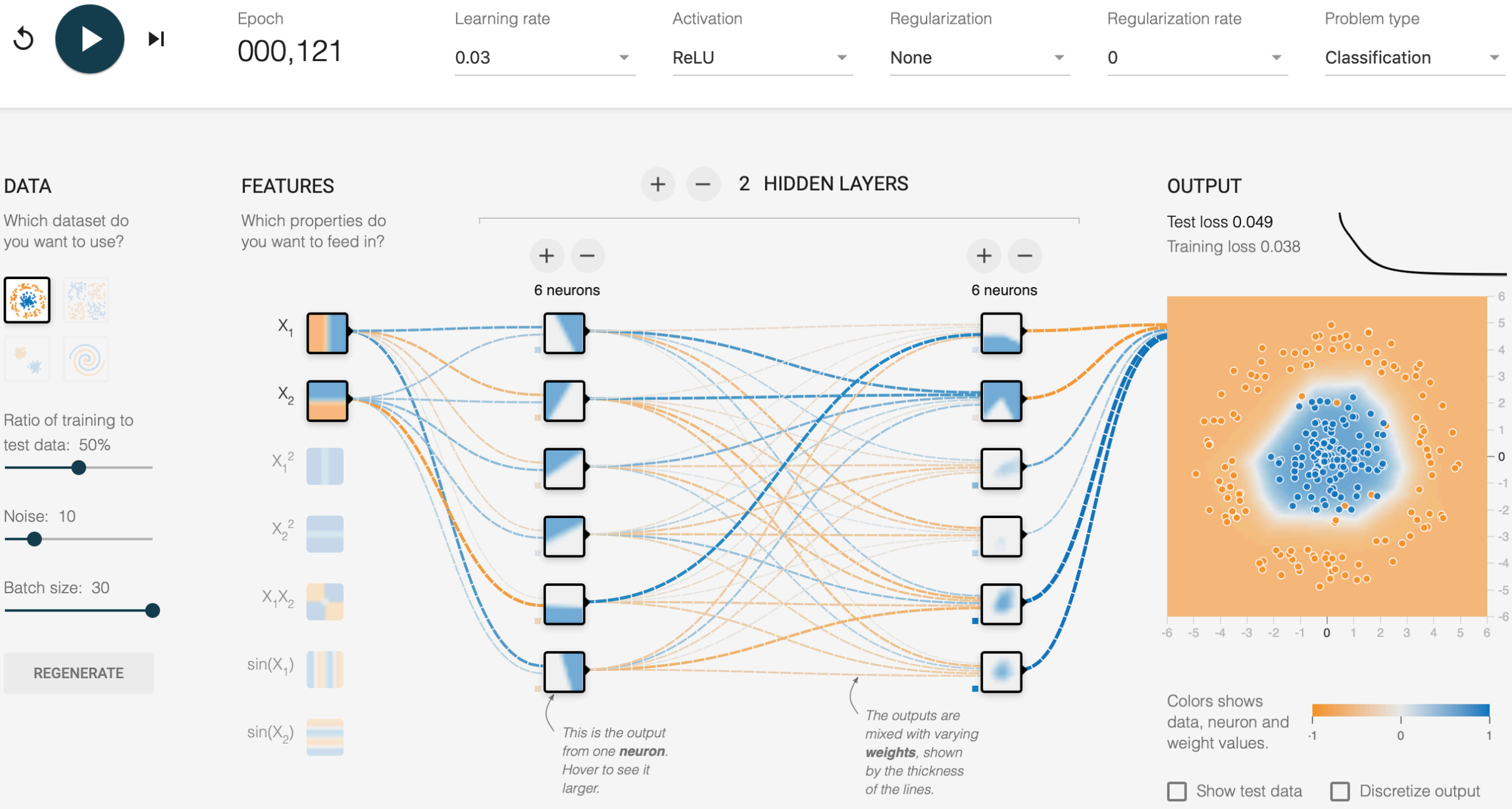
Training loss 0.001

Colors shows data, neuron and weight values. -1 0 1

☐ Show test data ☑ Discretize output

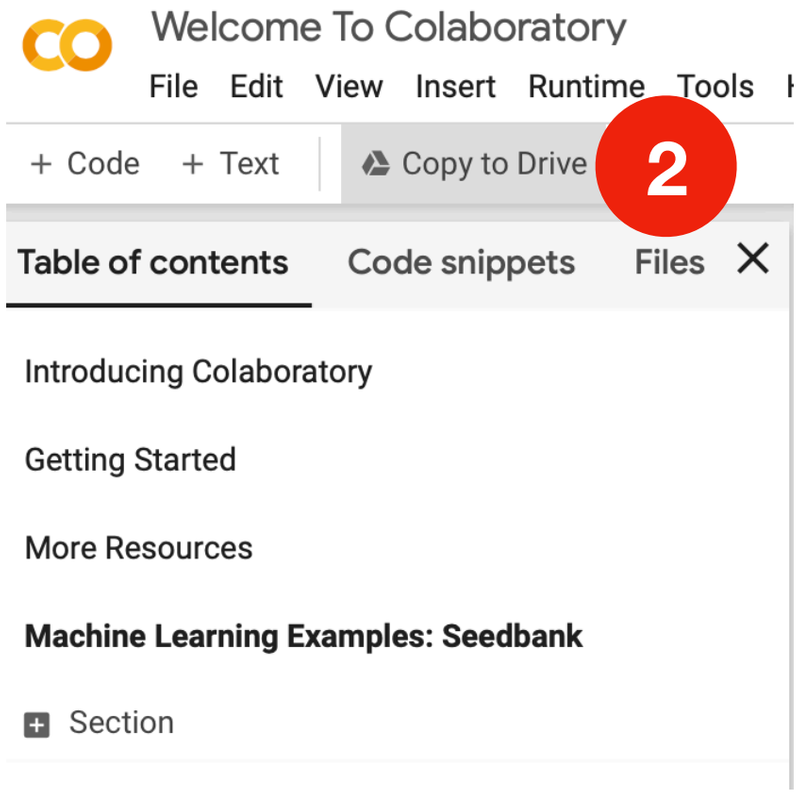
直觀互動

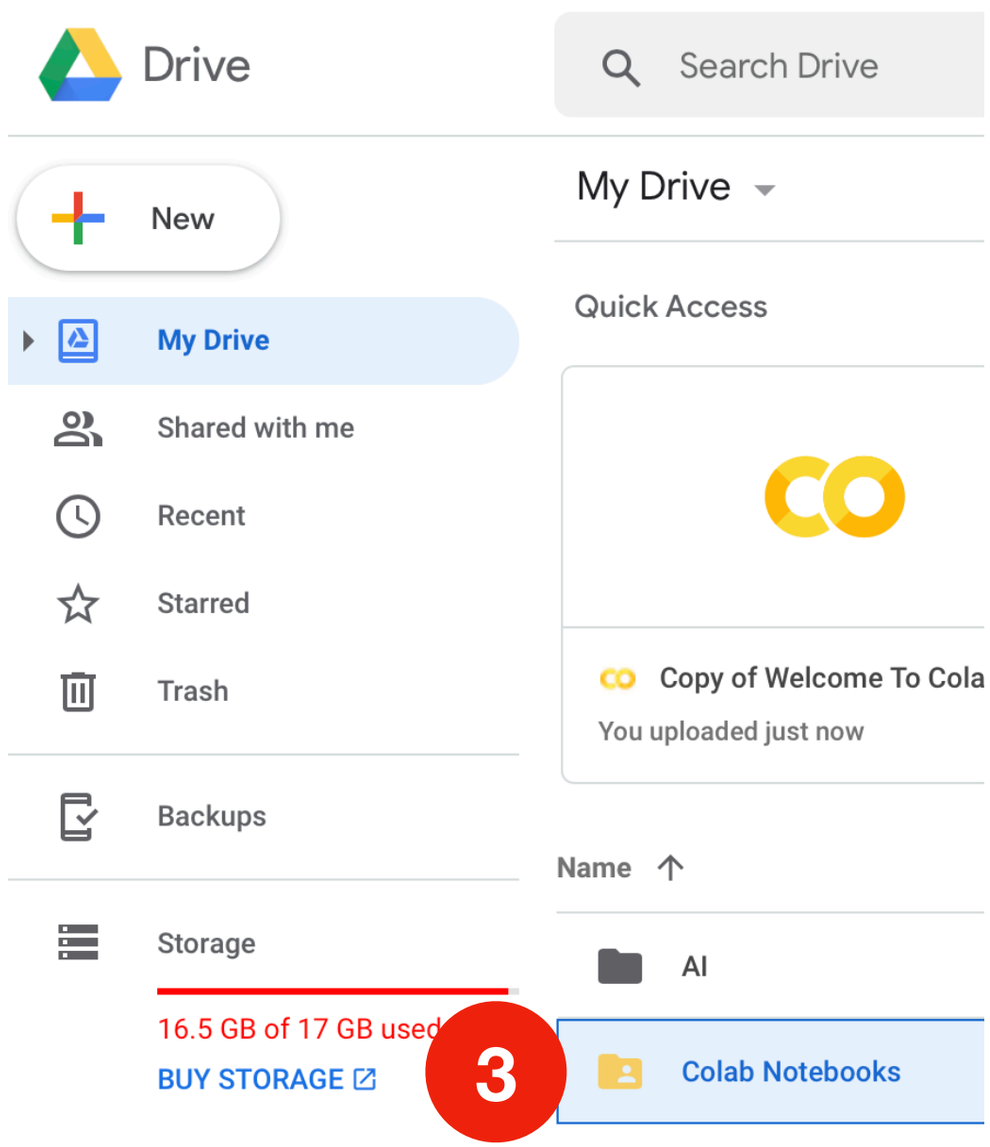
https://playground.tensorflow.org/



Colab

1 https://reurl.cc/W42WaO

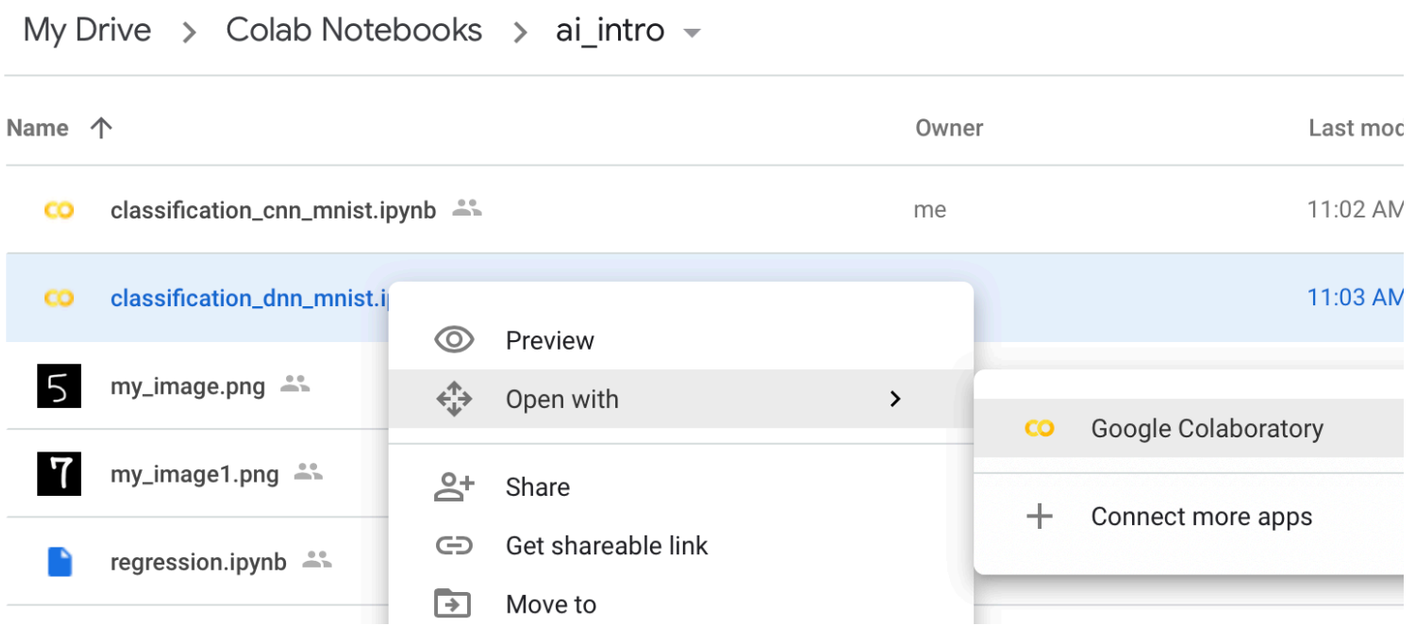




Sample Code

1. 到 https://reurl.cc/yy3az2
   把ai_intro檔案夾複製到你的Google Drive/Colab Notebooks檔案夾裡

2. 右鍵用 Google Colaboratory 打開 classification_dnn_mnist.ipynb



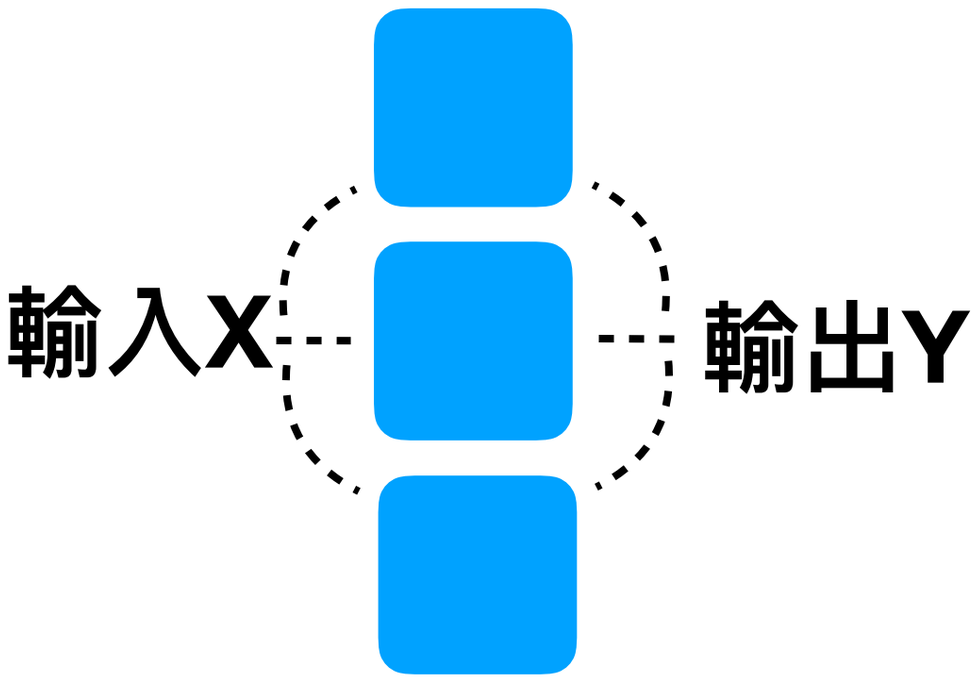
訓練集、測試集

Percentage of Body Fat	Age (years)	Height (inches)	Weight (lbs)
12.3	23	67.75	154.25
6.1	22	72.25	173.25
25.3	22	66.25	154.00
10.4	26	72.25	184.75
28.7	24	71.25	184.25

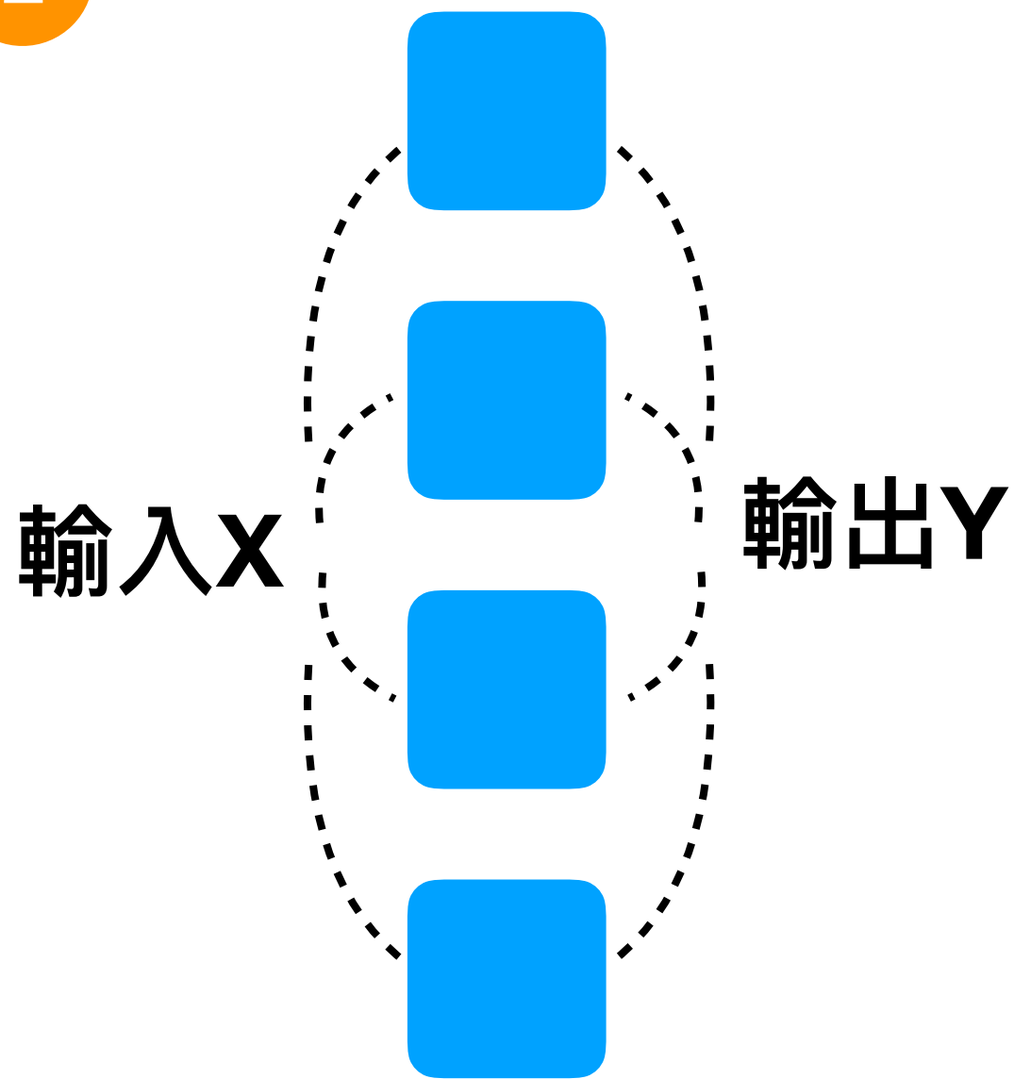
Percentage of Body Fat	Age (years)	Height (inches)	Weight (lbs)
	24	74.75	210.25
	26	69.75	181.00
	25	72.50	176.00
	25	74.00	191.00
	23	73.5	198.25

為什麼要升到高維空間

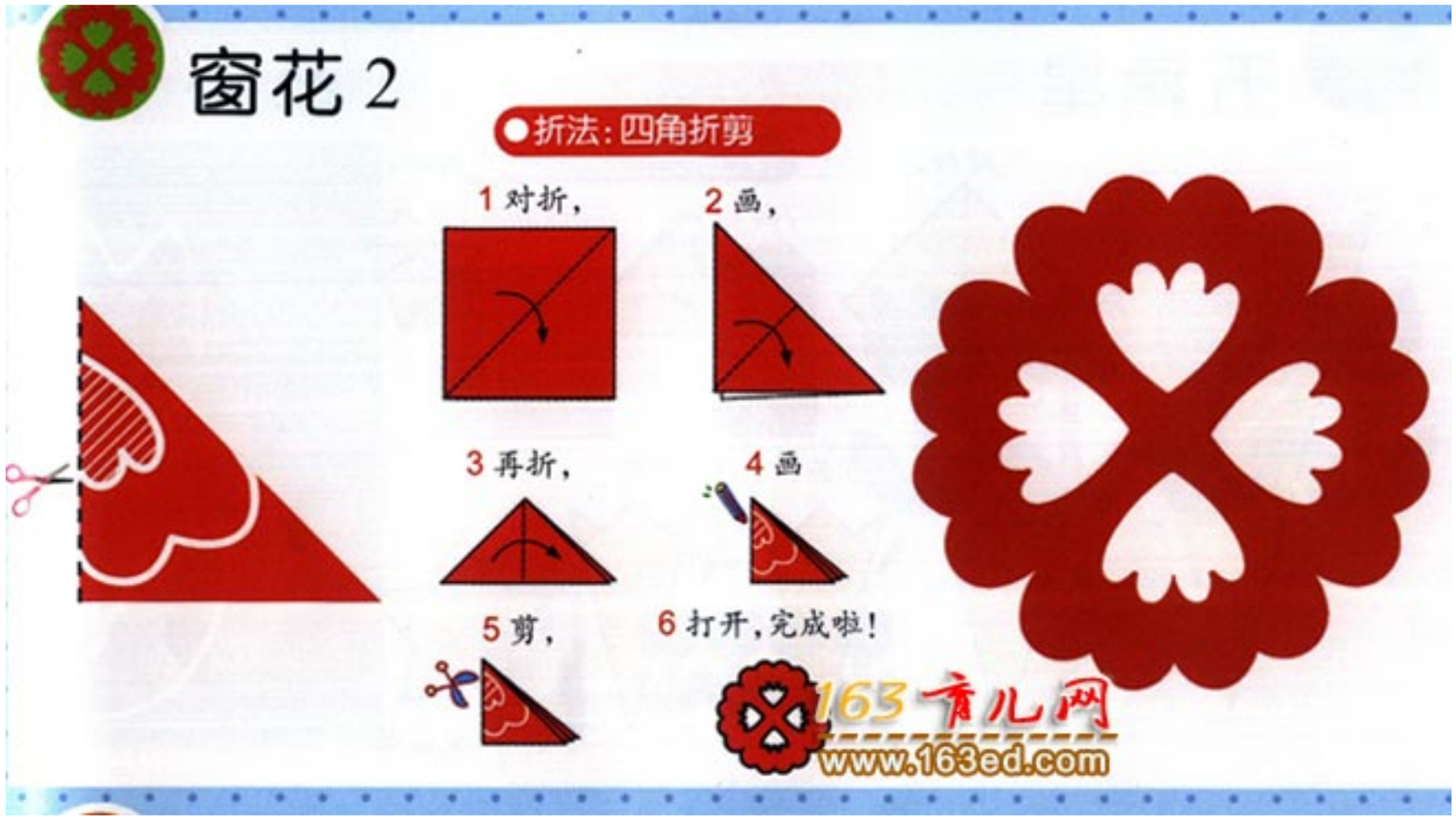
1

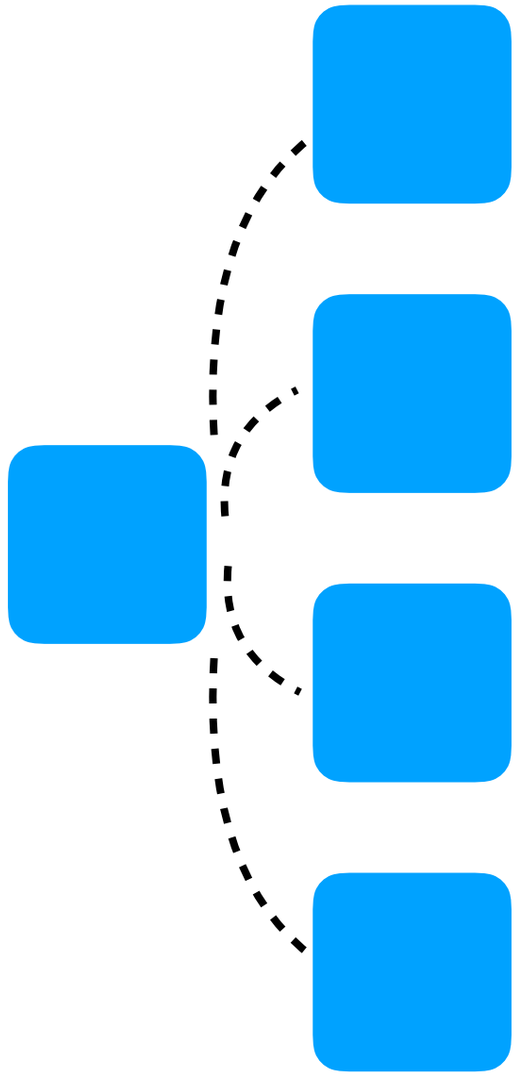


2



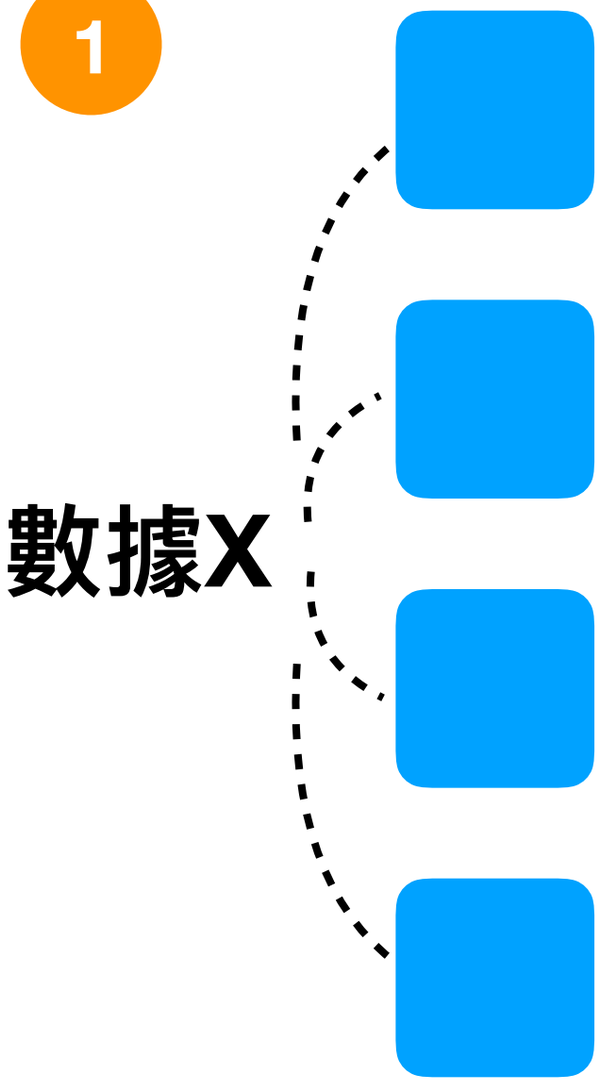
低維到高維空間



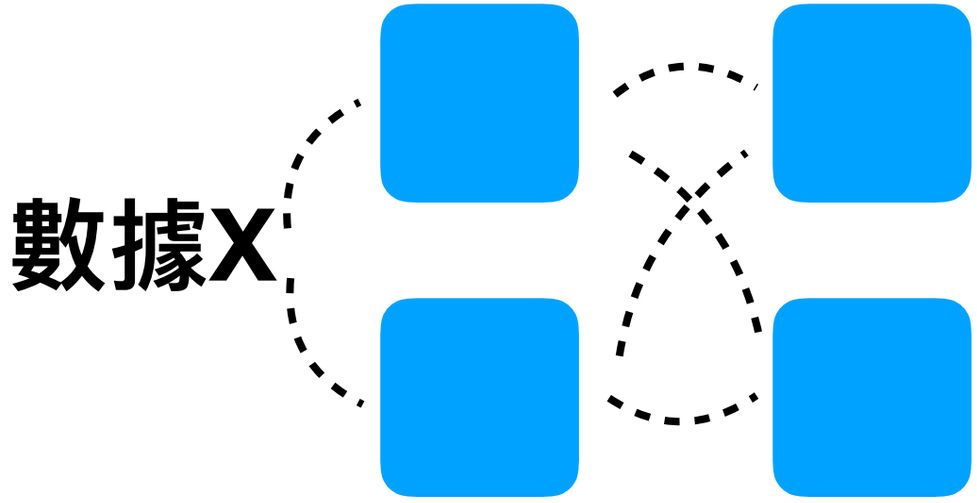


為什麼要「深度」學習

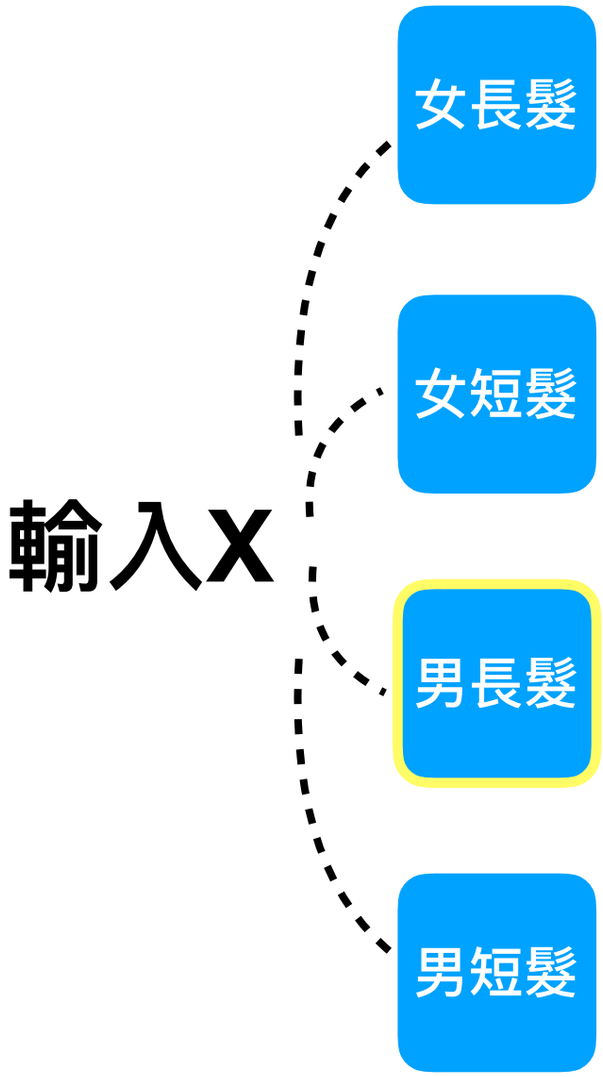
1

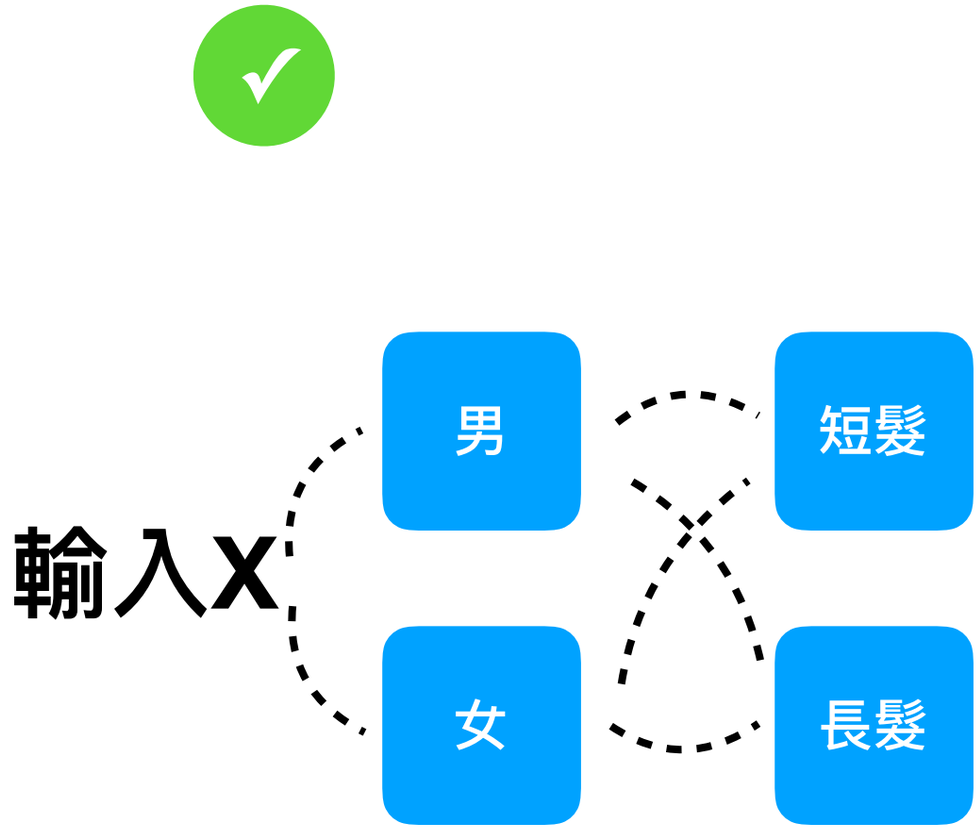


2

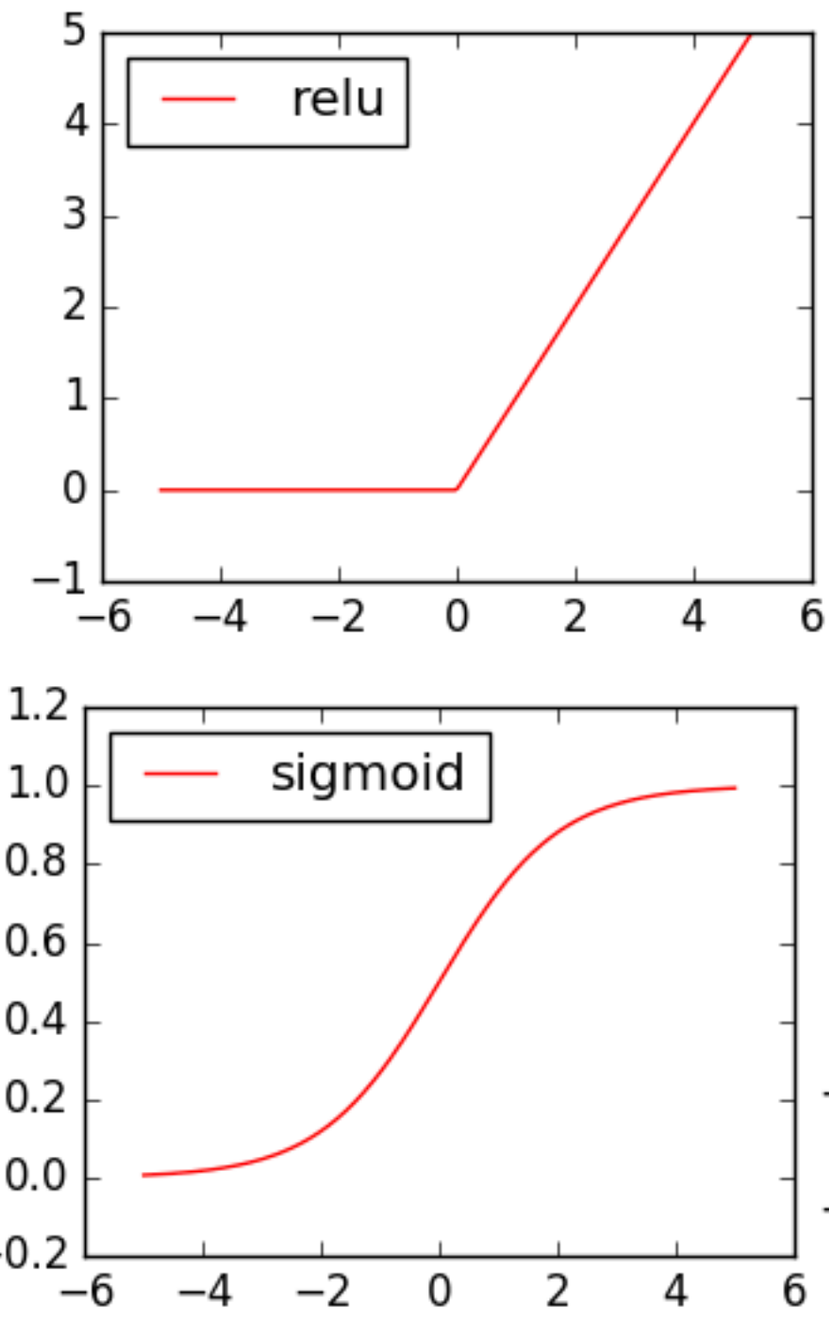


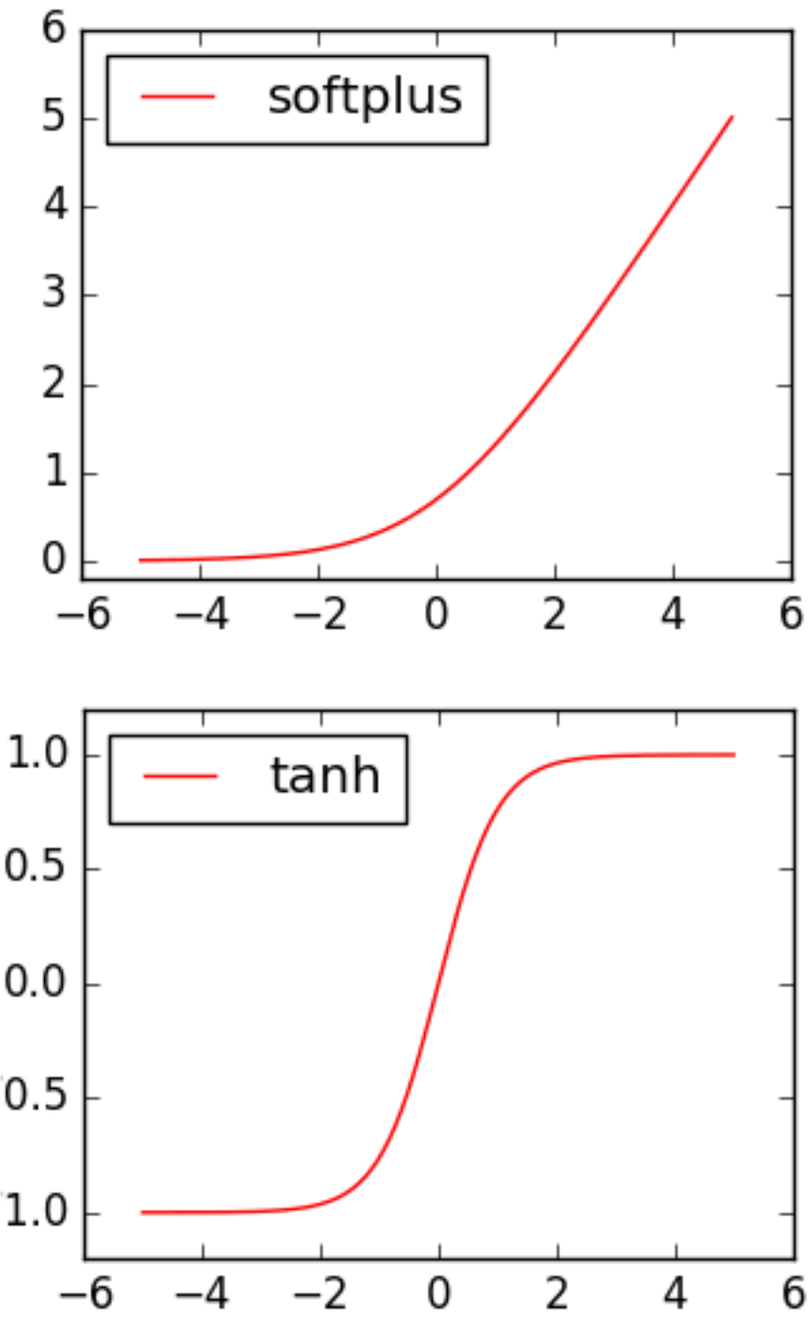
為什麼要「深度」學習



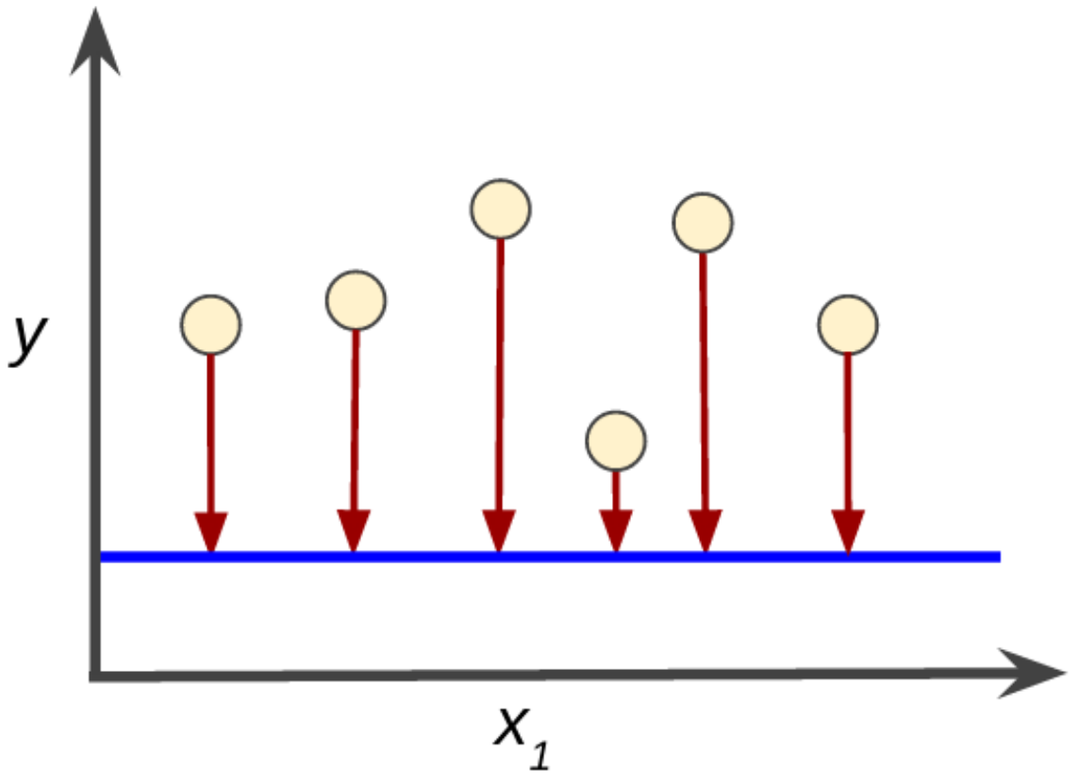


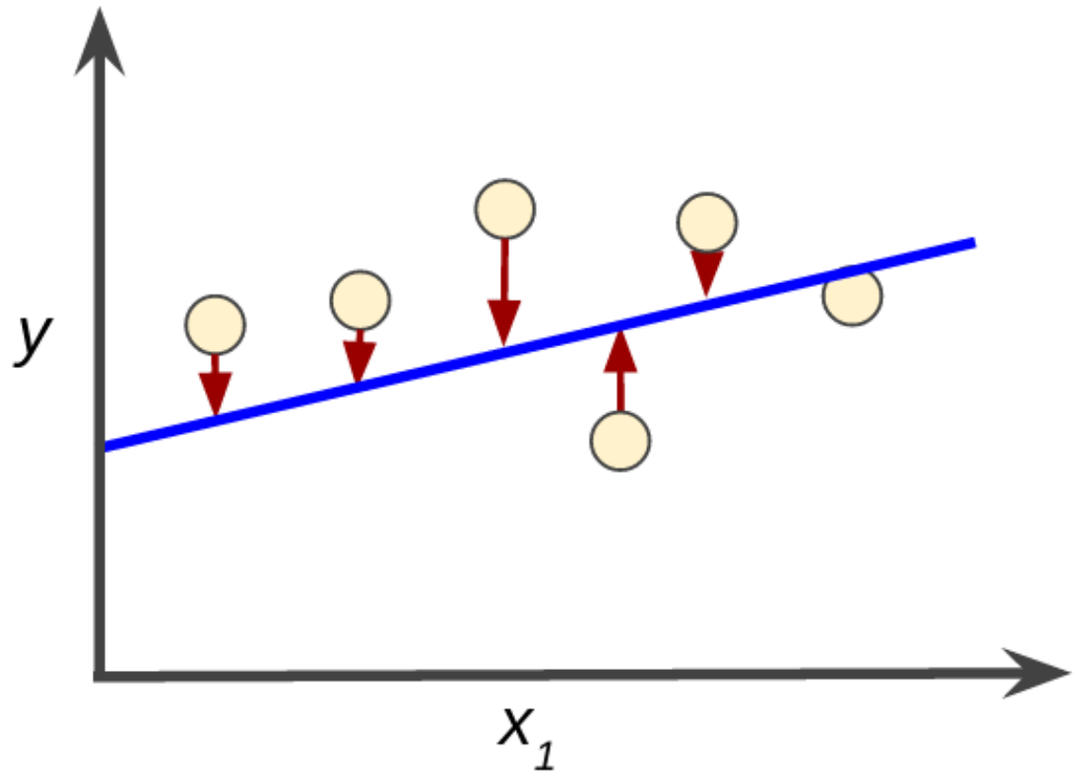
Activation 激勵函數：





Loss Function 損失函數



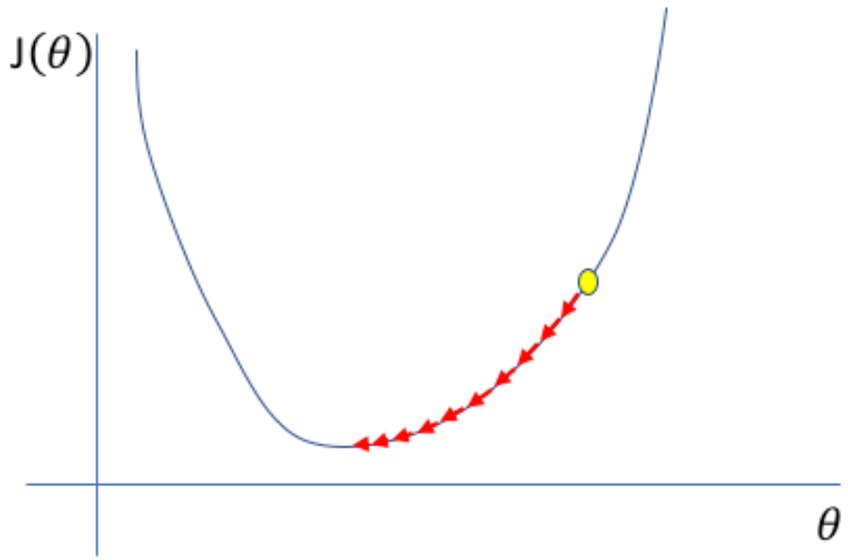


$$MSE = \frac{1}{n}\Sigma\left(y - \hat{y}\right)^2$$

圖片參考 Google Machine Learning Course

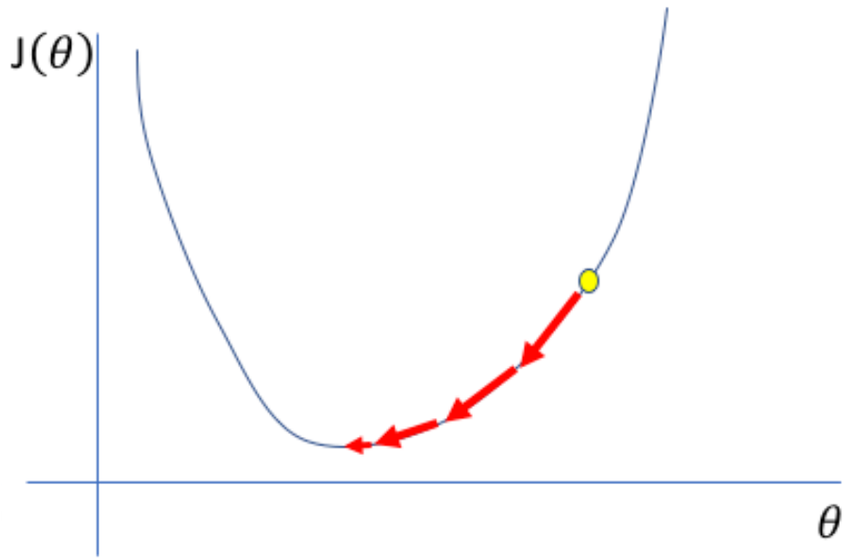
Learning Rate 學習率

Too low



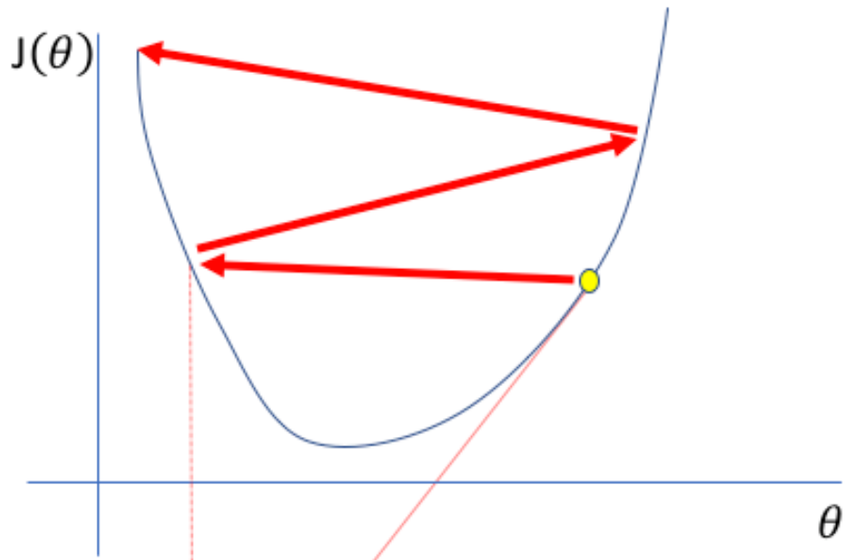
A small learning rate requires many updates before reaching the minimum point

Just right



The optimal learning rate swiftly reaches the minimum point

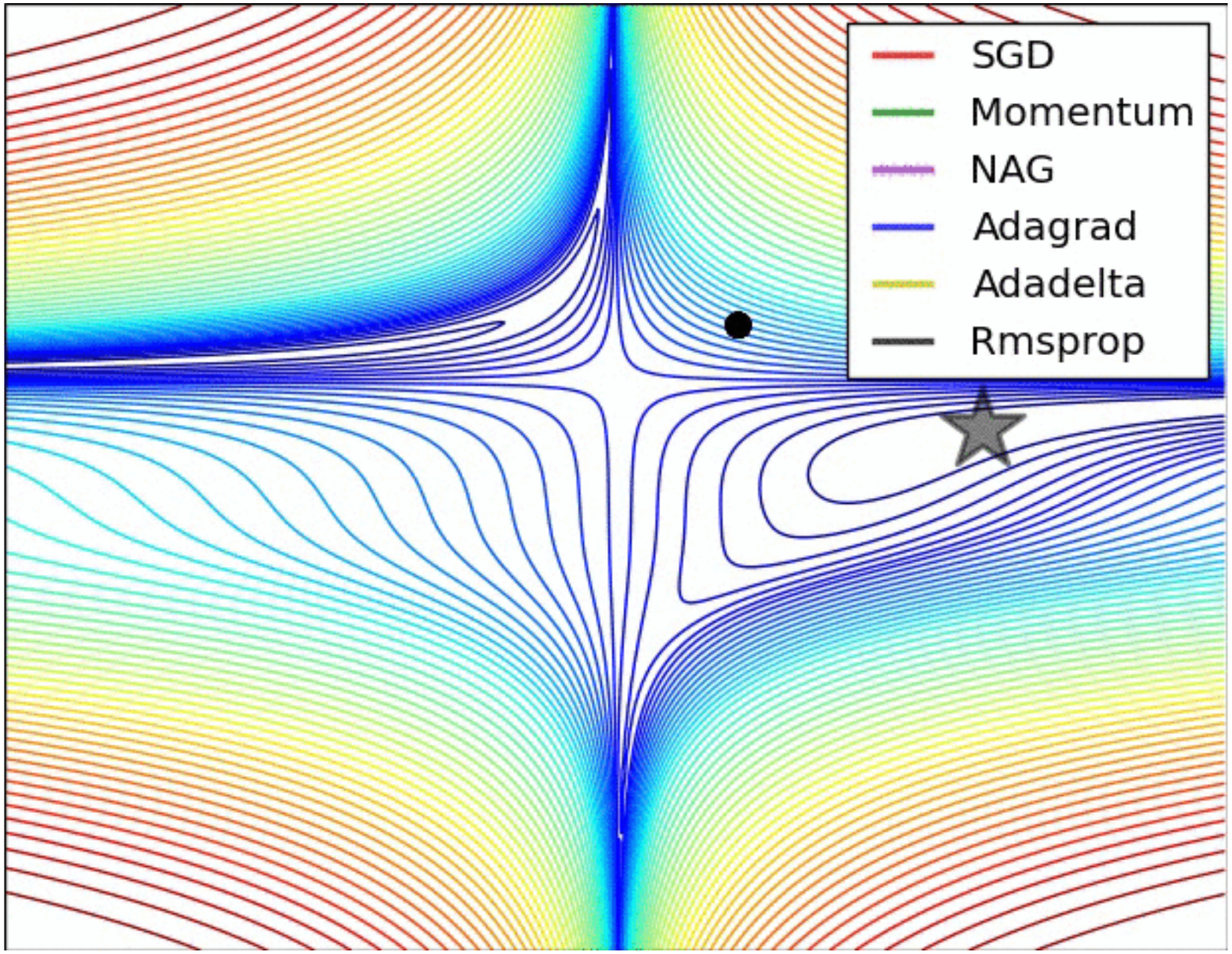
Too high



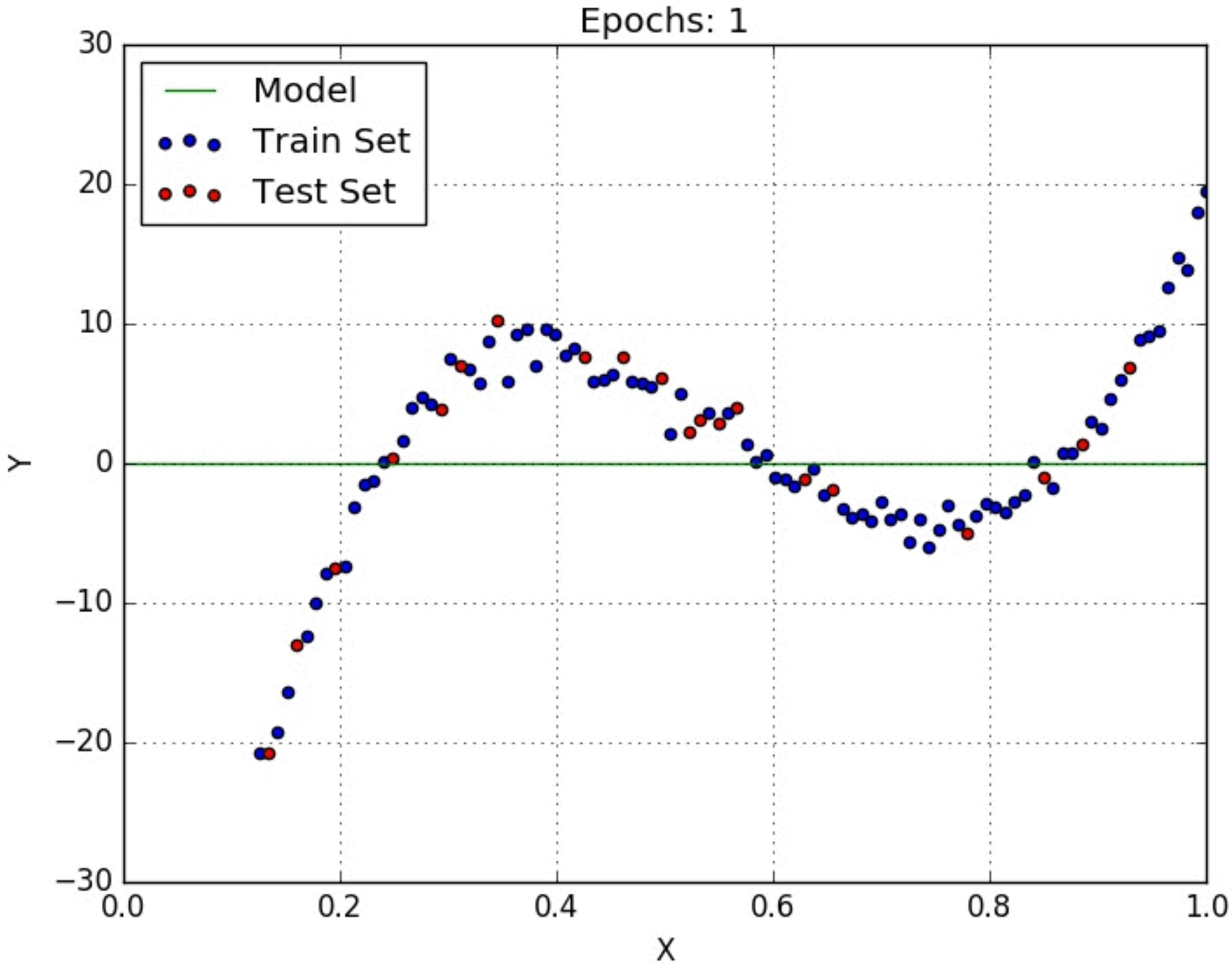
Too large of a learning rate causes drastic updates which lead to divergent behaviors

圖片參考自 https://www.jeremyjordan.me/nn-learning-rate/

Optimizer 優化器



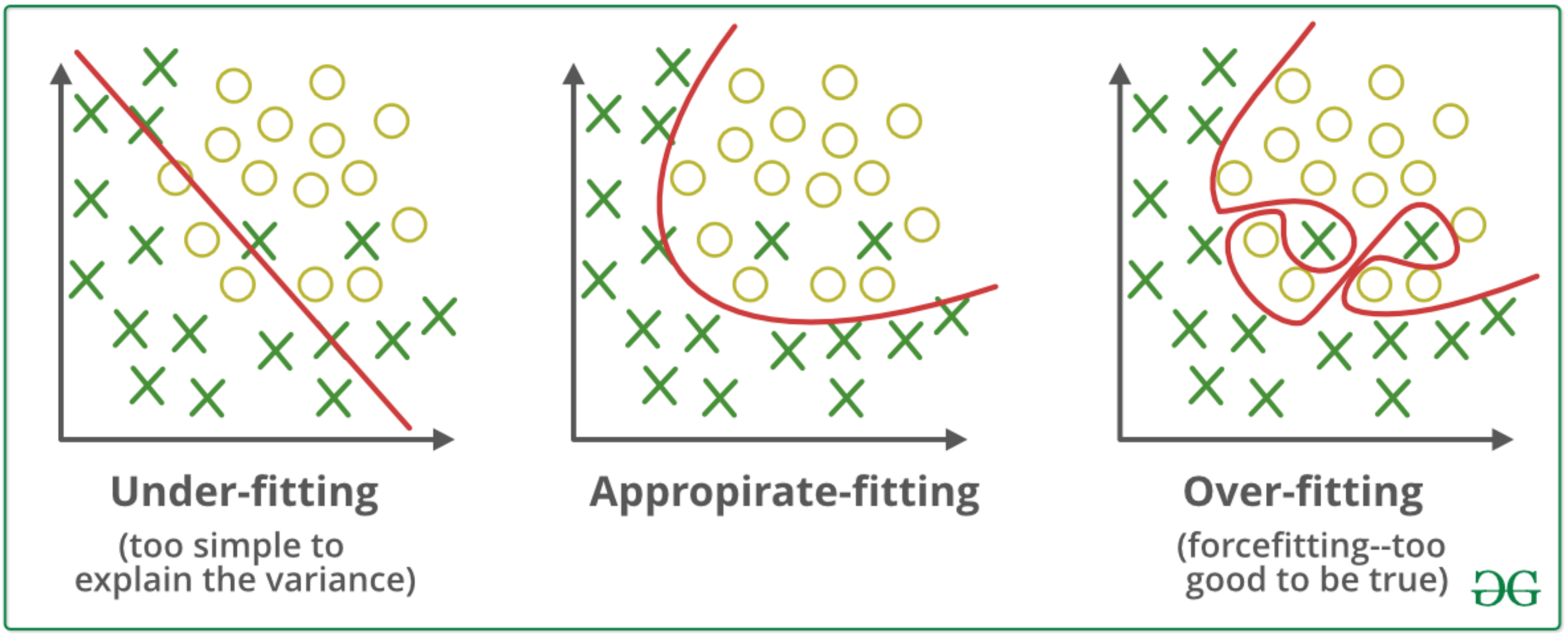
擬合



参考自www.cs.toronto.edu/

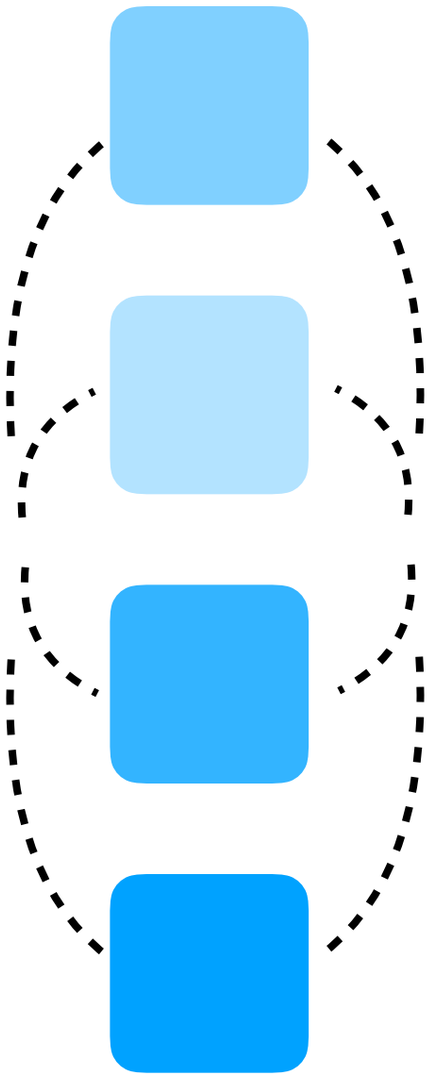
作者Davi Frossard

擬合



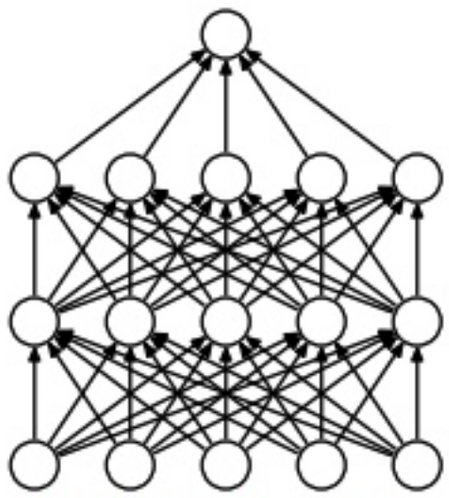
圖片參考自 towardsdatascience.com

用Dropout避免過擬合

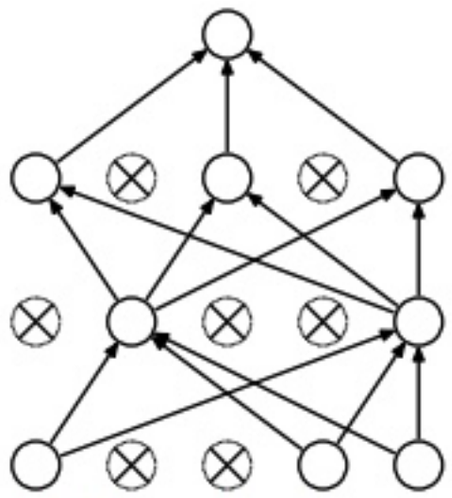


輸入X

輸出Y



(a) Standard Neural Net



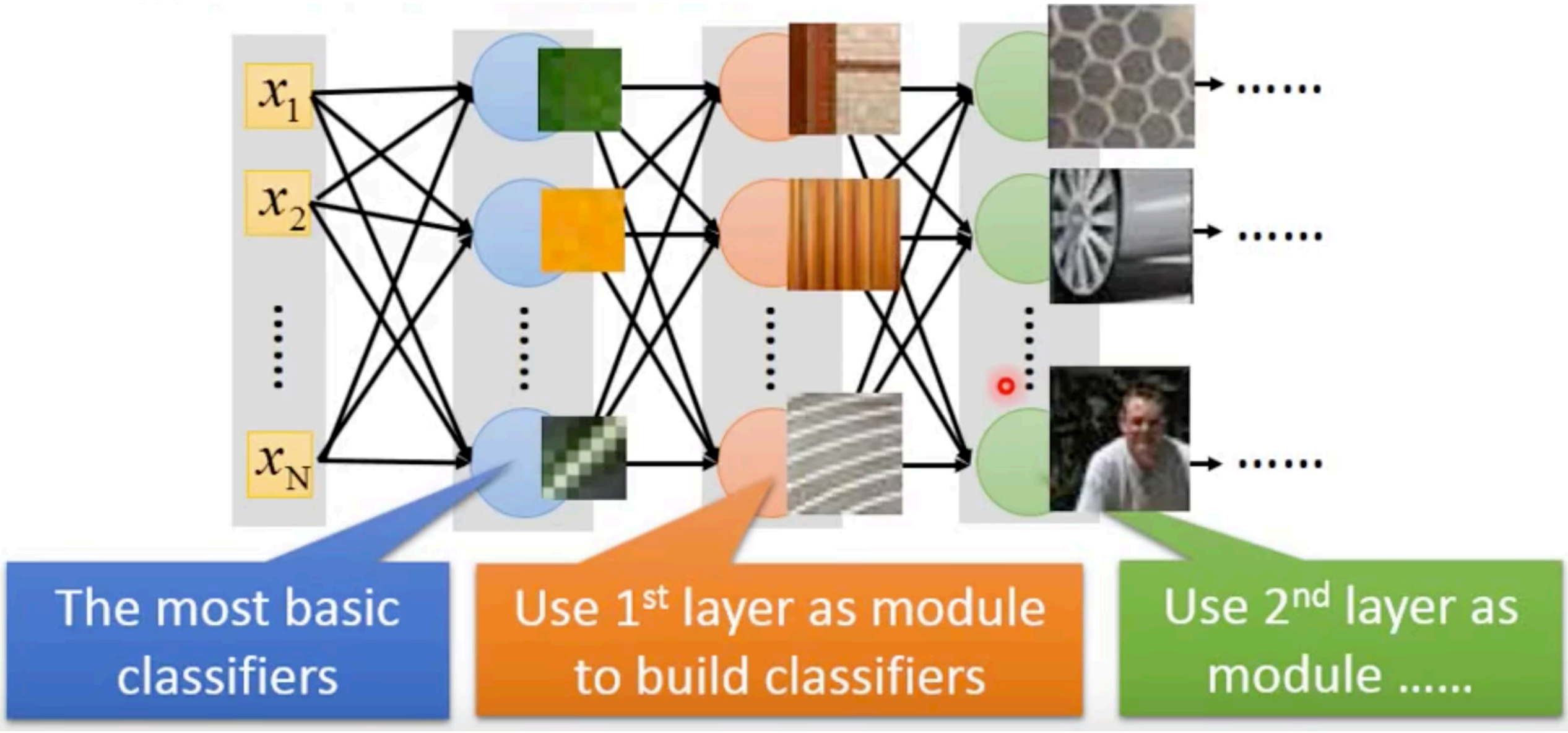
(b) After applying dropout.

Dropout rate 30%:

1. 每次訓練時，每層隨機(例如30%)的Neurons不作用
2. 剩下的Neurons要擔負起預測正確的責任
3. 防止參數過度依賴訓練數據。增加參數的泛化能力

圖片參考自 Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014

Deep CNN-> Modularization

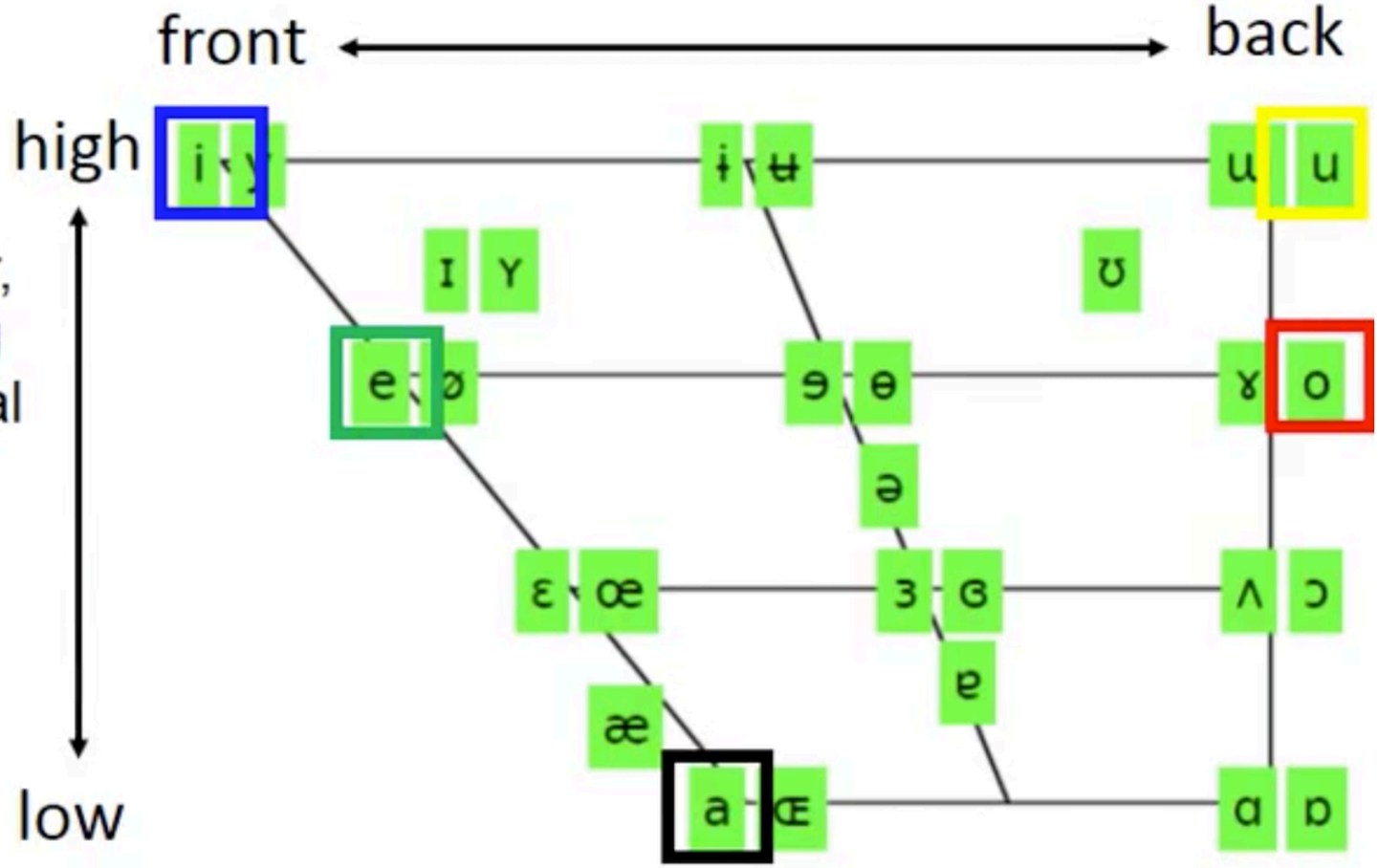


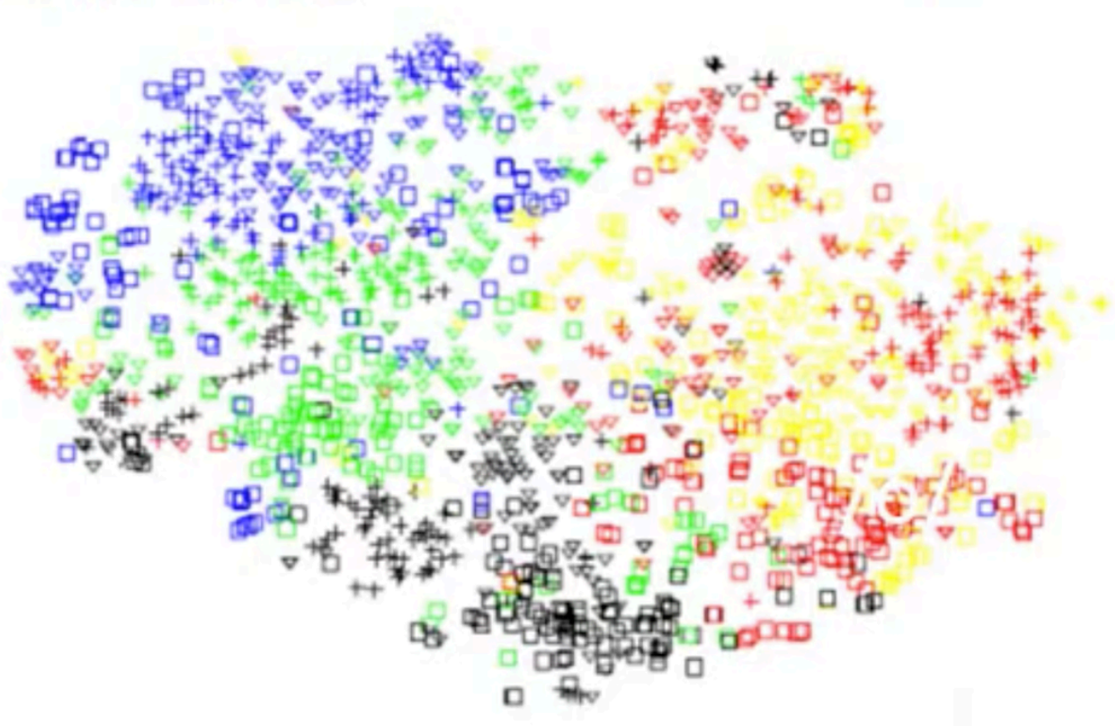
圖片參考自 Zeiler, M.D., & Fergus, R. (2014) Visualizing and understanding convolutional networks. In Computer Vision-ECCV 2014

Modularization

Vu, Ngoc Thang, Jochen Weiner, and Tanja Schultz. "Investigating the Learning Effect of Multilingual Bottle-Neck Features for ASR." *Interspeech*. 2014.

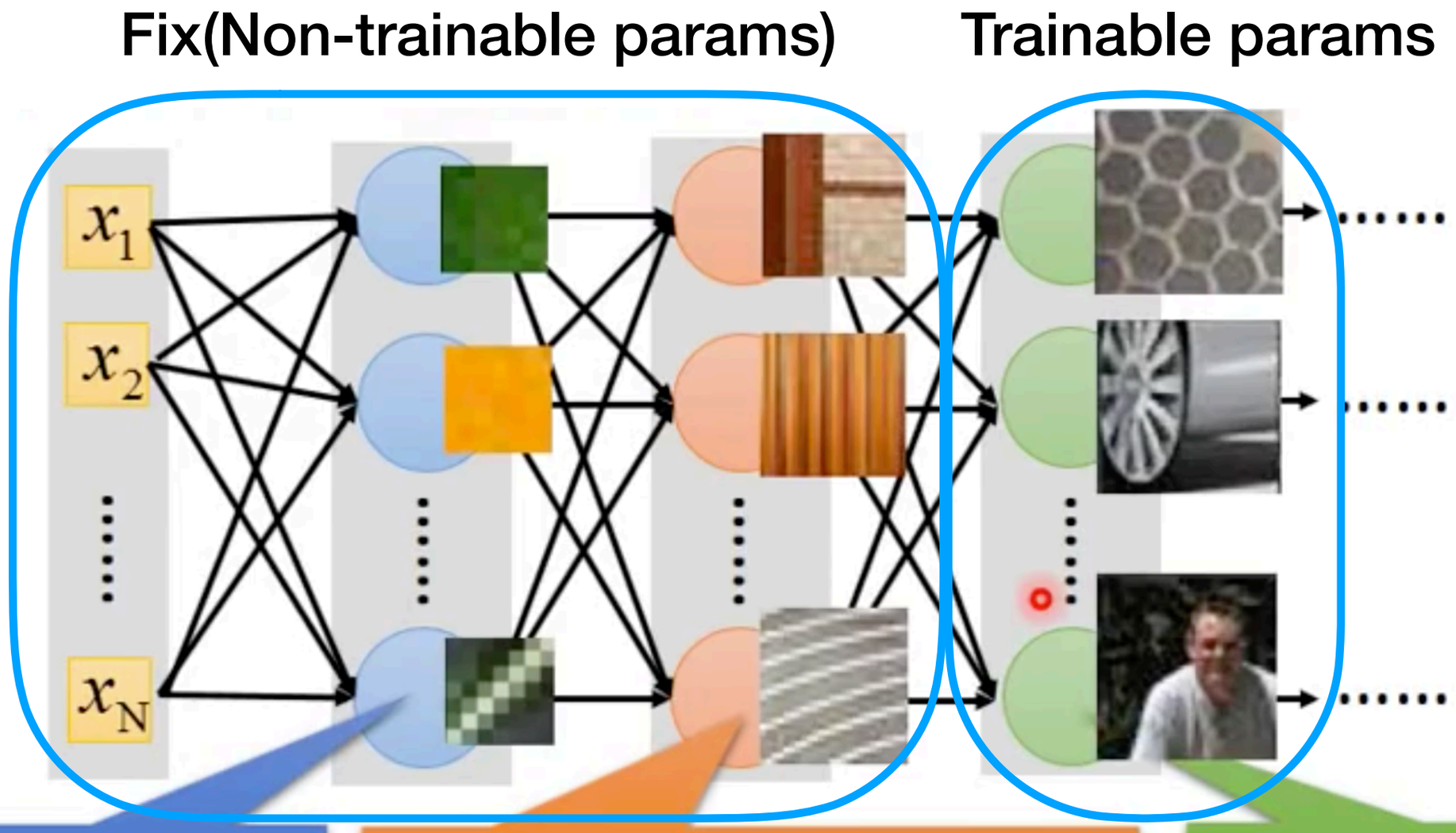
Output of hidden layer reduce to two dimensions





- The lower layers detect the manner of articulation
- All the phonemes share the results from the same set of detectors.
- Use parameters effectively

遷移學習



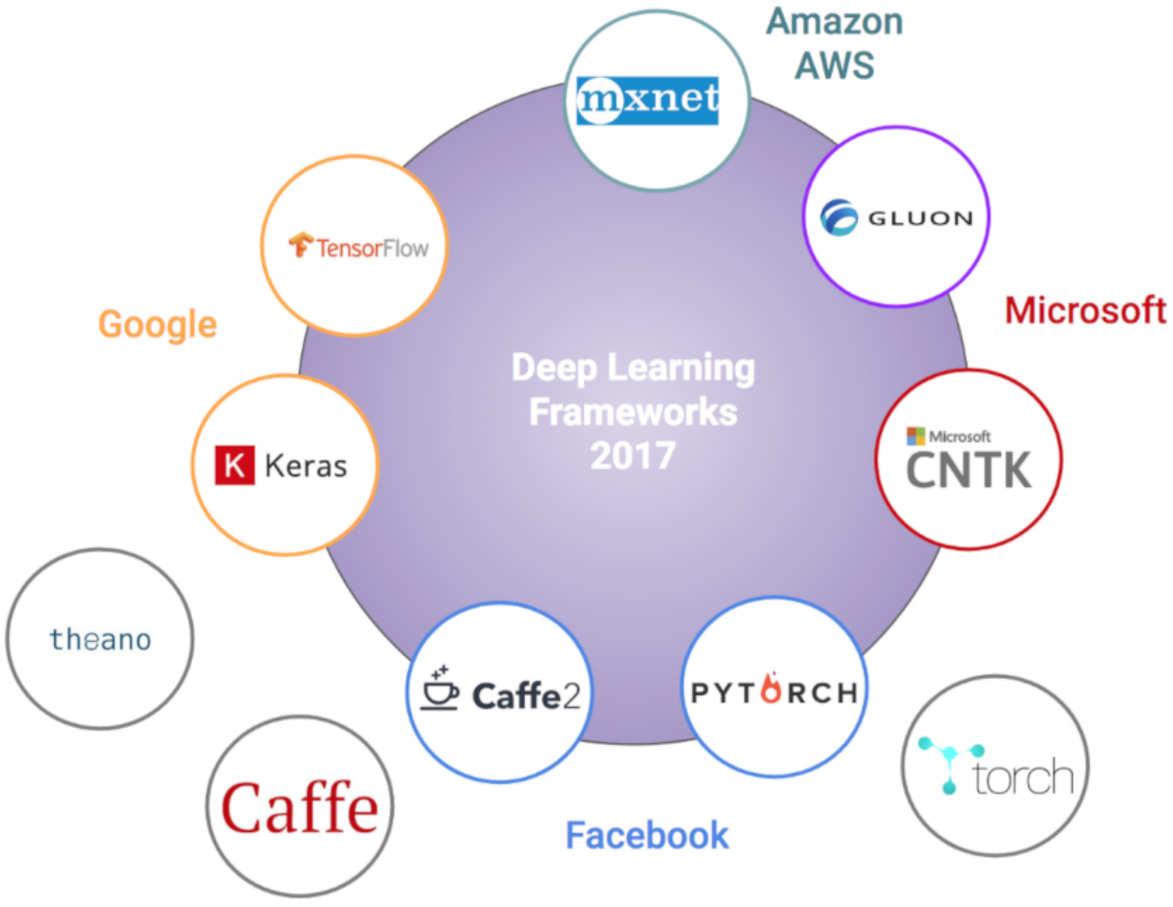
The most basic classifiers

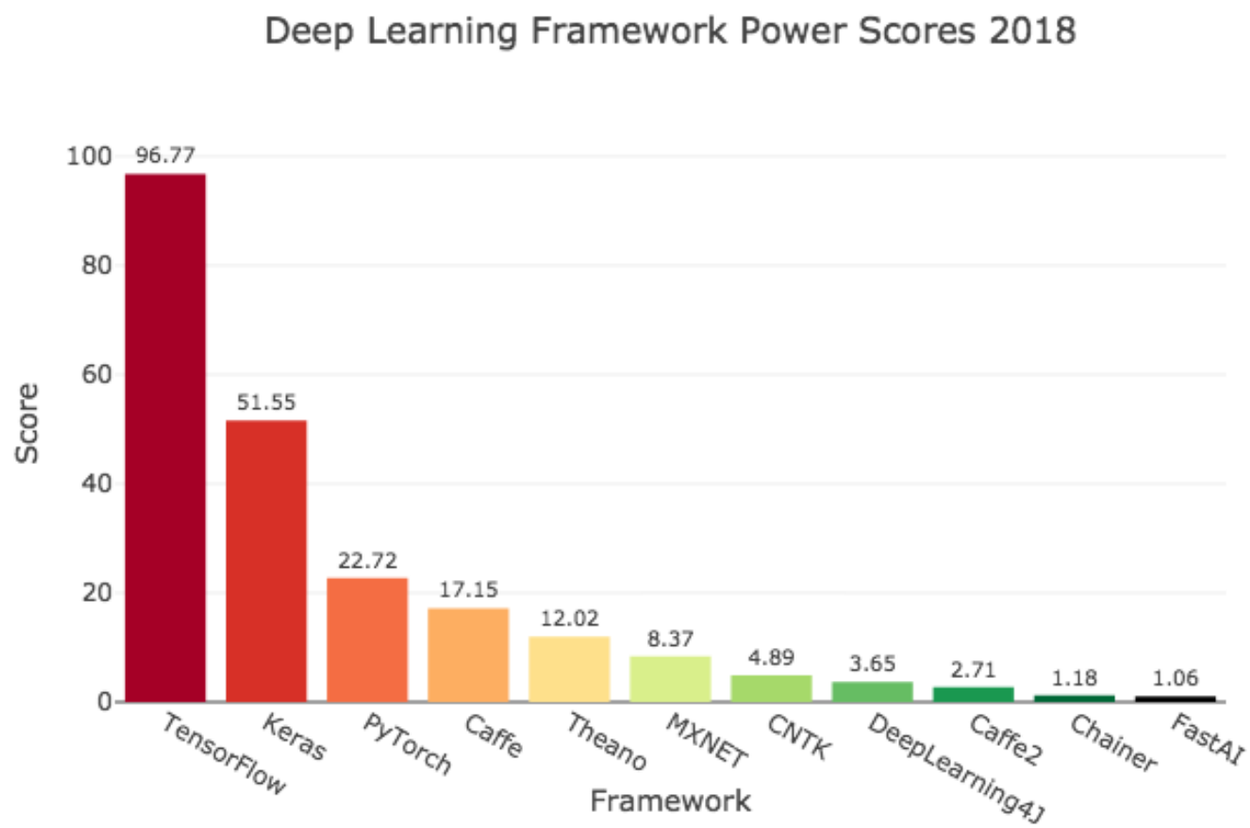
Use 1st layer as module to build classifiers

Use 2nd layer as module .....

Thank You

深度學習的工具：





參考自towardsdatascience.com

作者Jeff Hale

為什麼使用Keras？

Keras仍可用TensorFlow
簡單使用
仍具有一定的彈性

Sklearn:

```
model = GradientBoostingRegressor()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

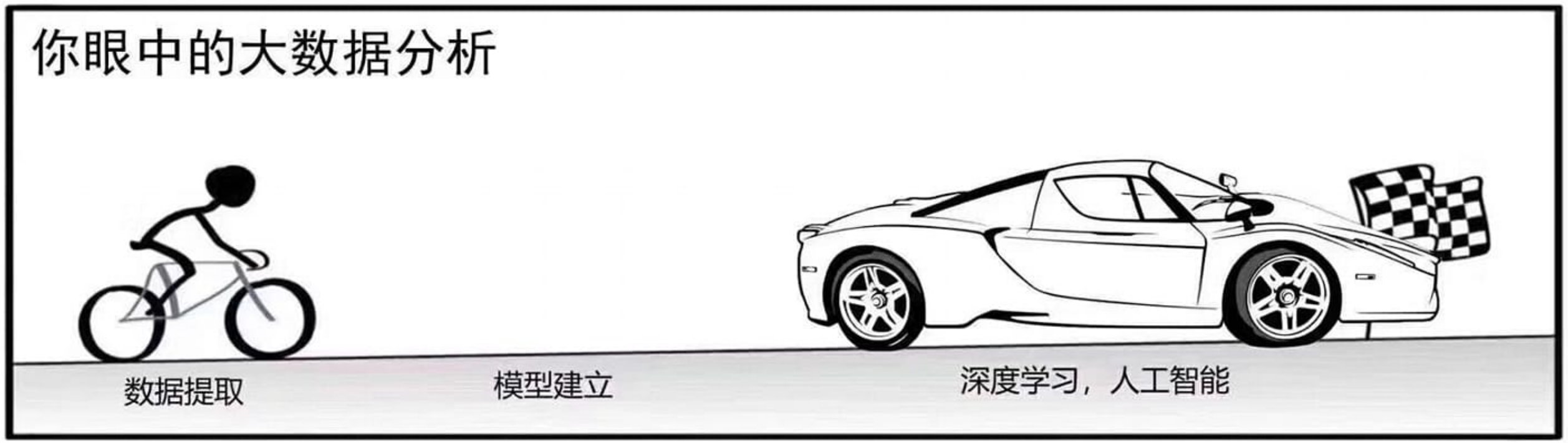
Keras:

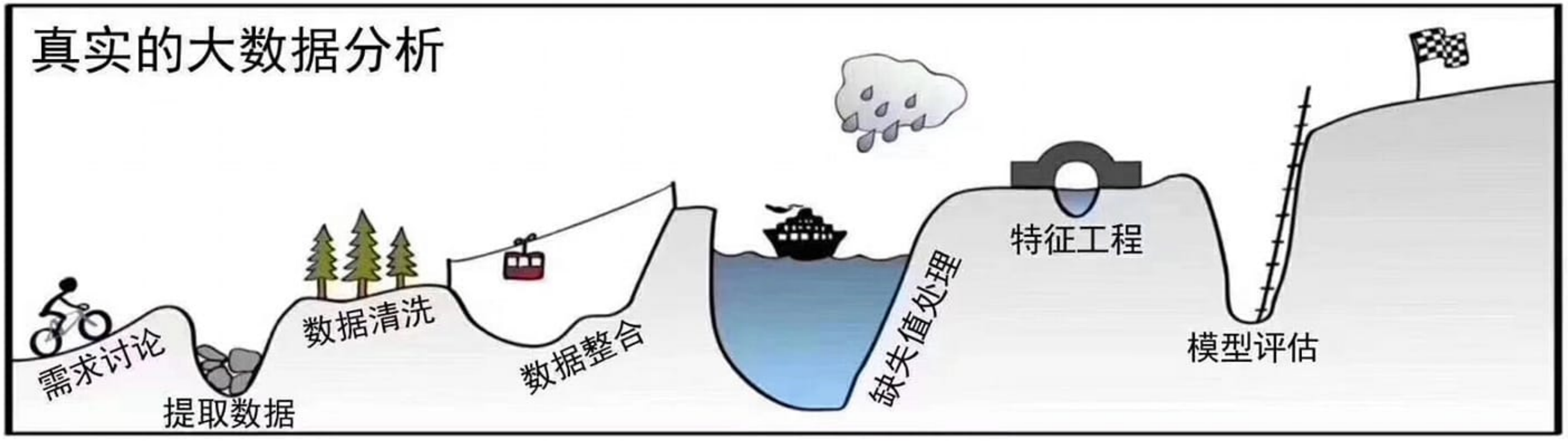
```
model = Sequential()
model.add(Dense(input_dim=1, units=10, activation='relu'))
model.add(Dense(units=20, activation='relu'))
model.add(Dense(units=10, activation='relu'))
model.add(Dropout(rate=0.0))
model.add(Dense(units=1))
model.compile(loss='mse', optimizer=Adam(lr=0.01))
model.fit(x_train, y_train, epochs=30, batch_size=64)
y_pred = model.predict(x_test)
```

TensorFlow:

```
x = tf.placeholder(tf.float32, [None, 1])
y = tf.placeholder(tf.float32, [None, 1])
layer1 = tf.layers.dense(inputs=x, units=10, activation=tf.nn.relu)
layer2 = tf.layers.dense(layer1, 20, tf.nn.relu)
layer3 = tf.layers.dense(layer2, 40, tf.nn.relu)
predict = tf.layers.dense(layer3, 1)
loss = tf.losses.mean_squared_error(labels=y, predictions=predict)
train = tf.train.GradientDescentOptimizer(learning_rate=0.003).minimize(loss)
sess = tf.Session()
sess.run(tf.global_variables_initializer())
for step in range(30000):
    c_ , _ = sess.run([loss, train], feed_dict={x:x_train, y:y_train})
y_pred = sess.run(predict, feed_dict={x:x_test})
```

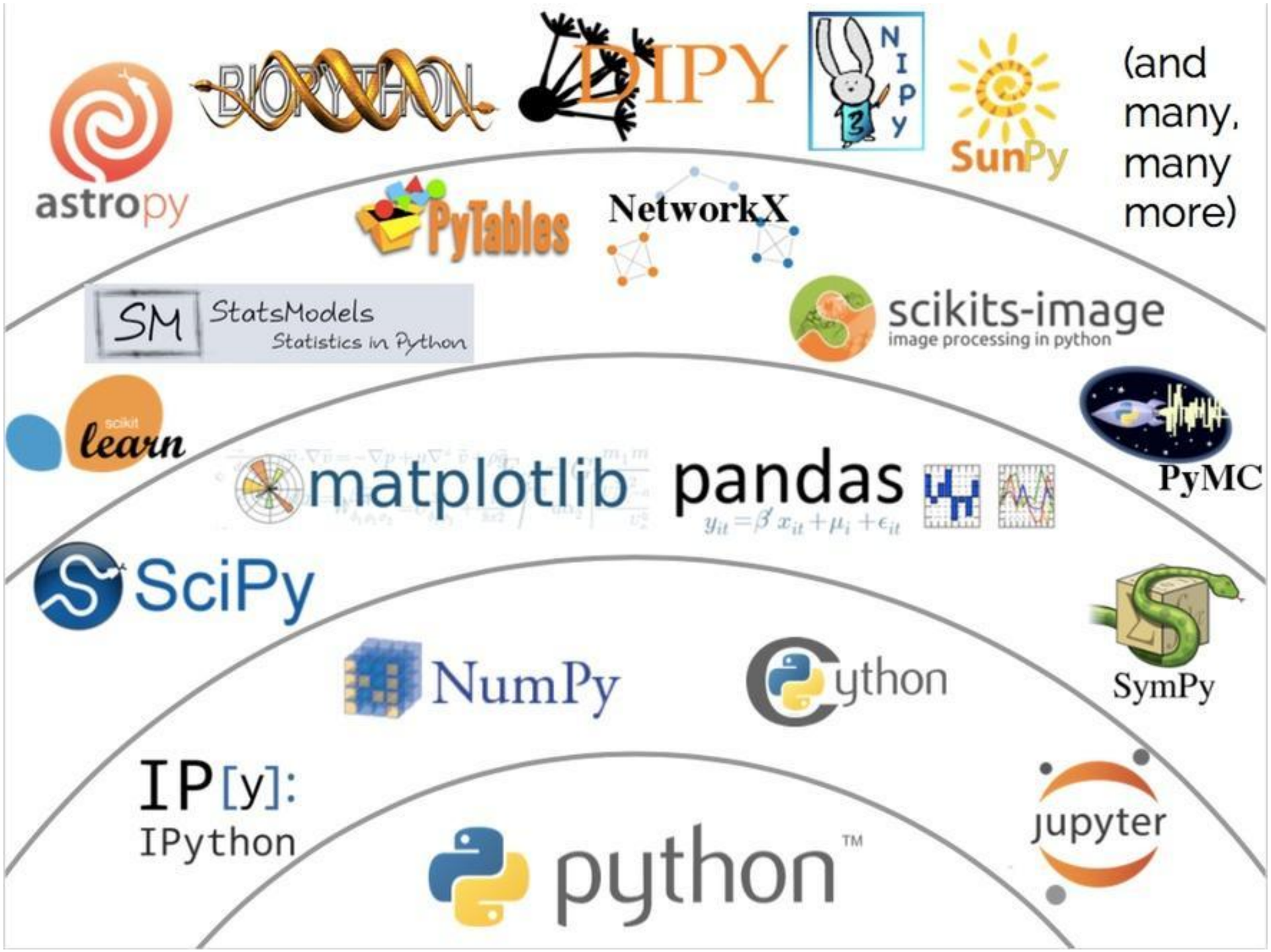
大數據流程





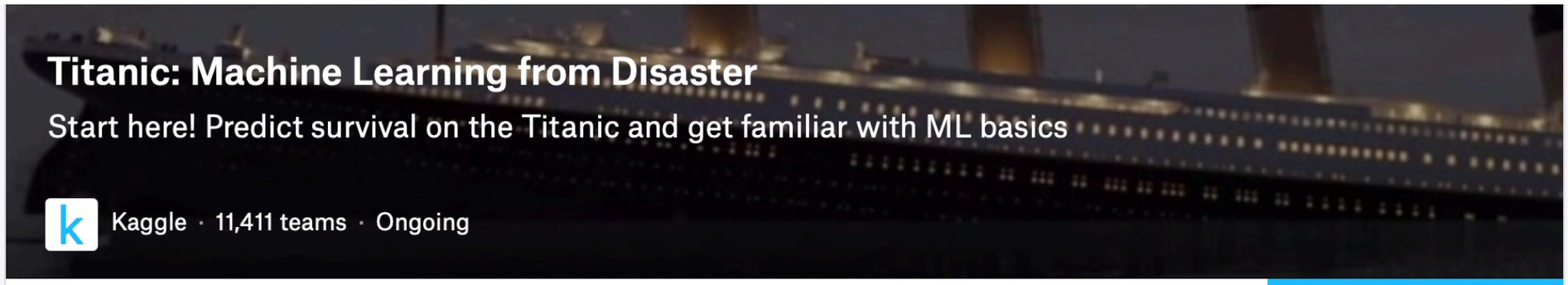
參考自「掘金用户程序员趣事」

數據科學的工具：


astropy
BIOPYTHON
DIPY
NIPY
SunPy
(and many, many more)
PyTables
NetworkX
SM
StatsModels
Statistics in Python
scikits-image
image processing in python
scikit learn
matplotlib
pandas
PyMC
SciPy
NumPy
Cython
SymPy
IP[y]:
IPython
python
jupyter

鐵達尼號倖存者預測

https://www.kaggle.com/c/titanic



Overview Data Kernels Discussion Leaderboard Rules Team My Submissions **Submit Predictions**

Overview

Description

Evaluation

Tutorials

Frequently Asked Questions

Start here if...

You're new to data science and machine learning, or looking for a simple intro to the Kaggle prediction competitions.

Competition Description

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this challenge, we ask you to complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy.

Practice Skills

- Binary classification
- Python and R basics

鐵達尼號資料集/測試集

- PassengerId: 乘客編號
- Survived: 乘客是否存活（1代表存活，0代表死亡）
- Pclass: 艙位是頭等艙、二等艙還是三等艙
- Name: 乘客姓名
- Sex: 乘客的性別
- Age: 乘客的年齡
- SibSp: 在鐵達尼號上兄弟姐妹或者配偶的人數
- Parch: 在鐵達尼號上父母或者子女的人數
- Ticket: 乘客的船票號碼
- Fare: 買的船票價格
- Cabin: 在船上住的房間編號
- Embarked: 在英國哪個港口上的船

```
print(df_train.shape)
df_train.head()
```

```
(891, 12)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
print(df_test.shape)
df_test.head()
```

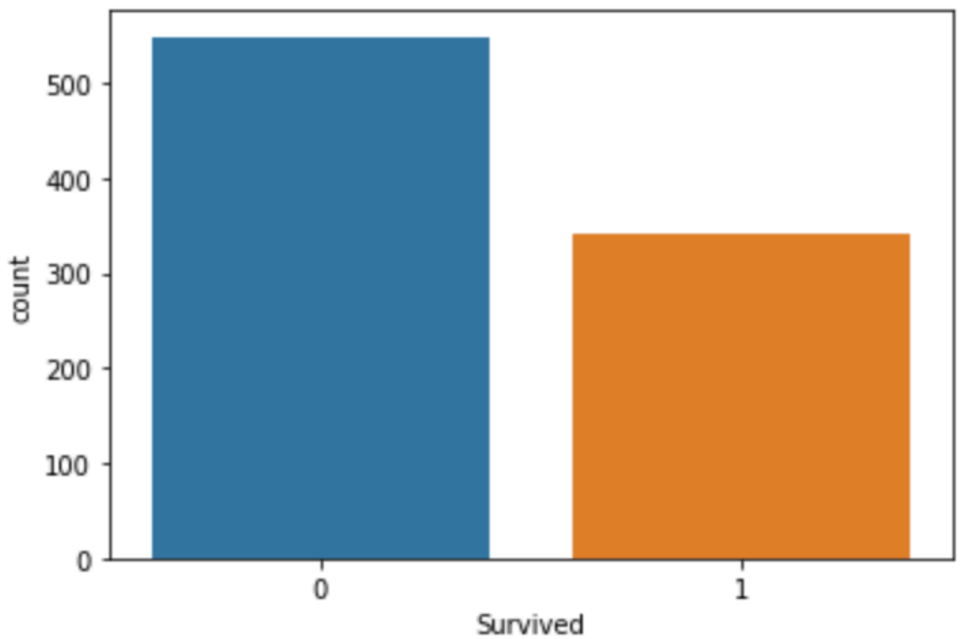
```
(418, 11)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

生存的比例大概是4成、死亡的比例是6成

```
sns.countplot(df_train['Survived'])
```

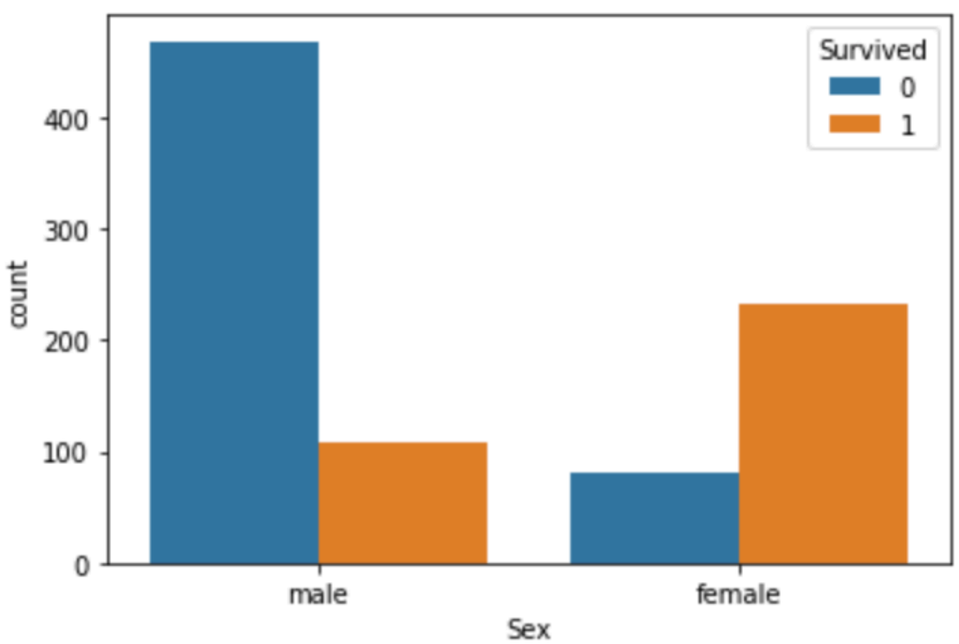
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a14ae7f60>
```



女人生存率是男人的好幾倍

```
sns.countplot(df_train['Sex'], hue=df_train['Survived'])
```

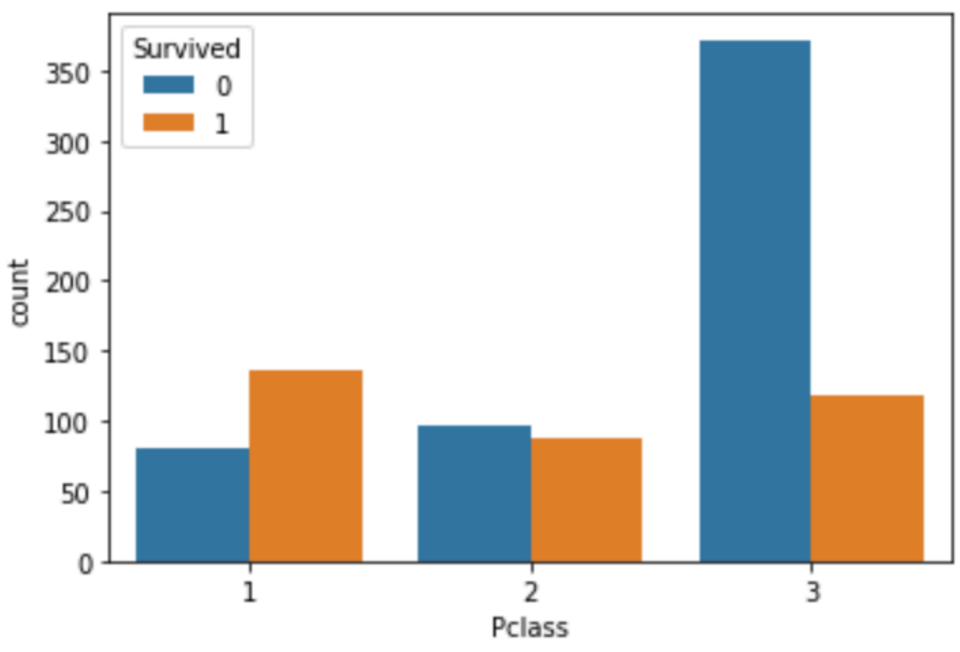
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a14b78550>
```



1等艙的生存率最高、再來是2等艙、最後是3等艙

```
sns.countplot(df_train['Pclass'], hue=df_train['Survived'])
```

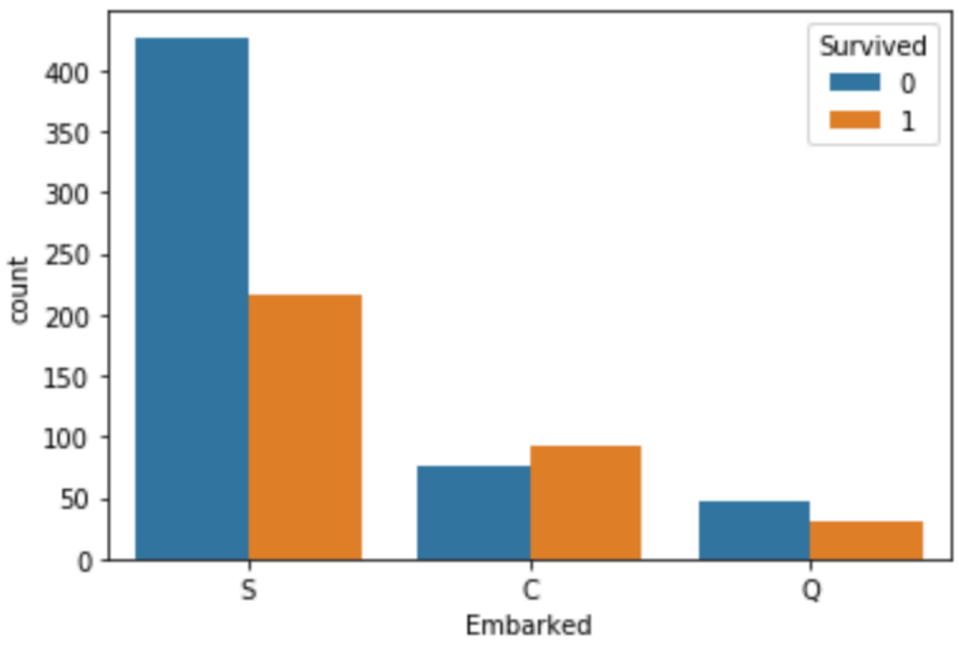
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a14c46dd8>
```



S港出發的都比較容易死亡

```
sns.countplot(df_train['Embarked'], hue=df_train['Surviv
```

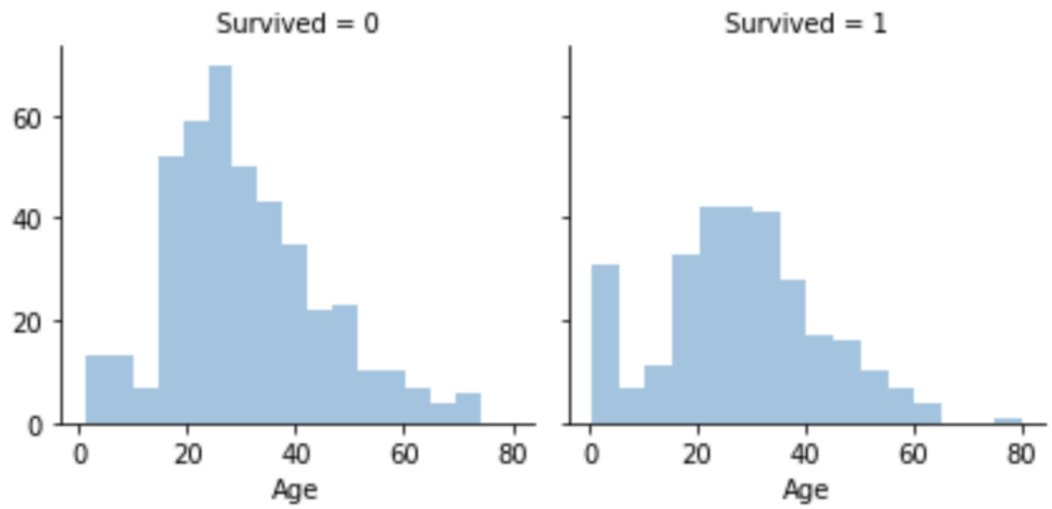
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a14c940f0>
```



年齡小的存活比例高 ¶

```
g = sns.FacetGrid(df_train, col='Survived')
g.map(sns.distplot, 'Age', kde=False)
```

```
<seaborn.axisgrid.FacetGrid at 0x1a14cb7240>
```

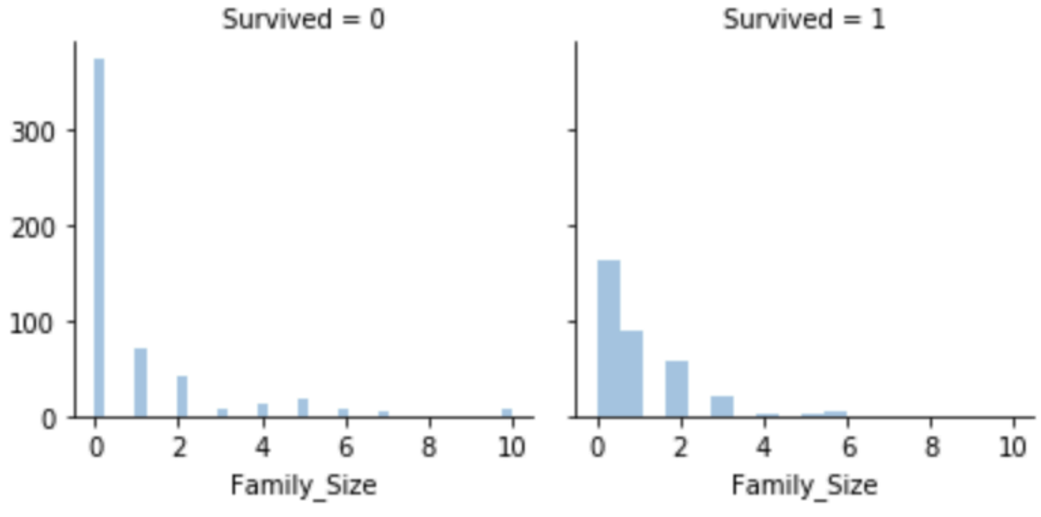


沒有跟家人一起上船的，生存率偏低

```
# 把「父母+小孩」加上「兄弟姊妹+丈夫妻子」的數量變成一個新的欄位叫做家庭大小
df_train['Family_Size'] = df_train['Parch'] + df_train['SibSp']

g = sns.FacetGrid(df_train, col='Survived')
g.map(sns.distplot, 'Family_Size', kde=False)
```

```
<seaborn.axisgrid.FacetGrid at 0x1a14d15668>
```

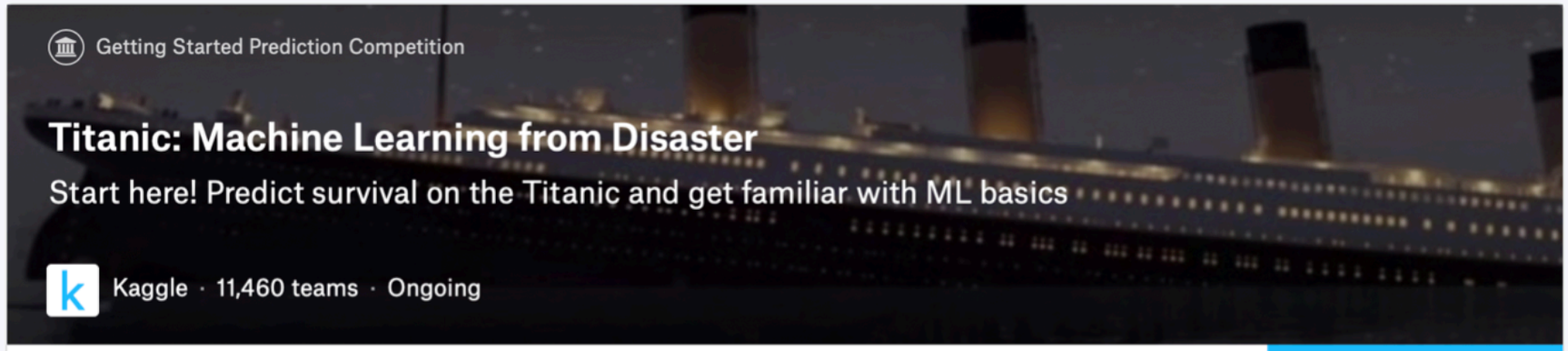


最簡版流程

https://ithelp.ithome.com.tw/articles/10227139

- 讀取csv
- 取出目標欄位
- 取出非特徵欄位
- 合併df_train、df_test
- 填補缺失值
- 編碼
- 歸一化
- 取得X_train、y_train
- 訓練模型
- 預測X_test
- 儲存預測數據

最簡版預測正確率



Overview Data Kernels Discussion Leaderboard Rules Team My Submissions **Submit Predictions**

Public Leaderboard Private Leaderboard

This leaderboard is calculated with approximately 50% of the test data.
The final results will be based on the other 50%, so the final standings may be different.

Raw Data Refresh

#	Team Name	Kernel	Team Members	Score	Entries	Last
8783	**kevinzhang98**	</> Titanic Classifier		0.75598	2	2mo
8784	**physics653**			0.75598	1	2mo
8785	**yoyoyo12**			0.75598	2	2mo
8786	**Ian Fan**			0.75598	3	24d

Your Best Entry ↑

Your submission scored 0.64593, which is not an improvement of your best score. Keep trying!

8787	**orcarex**			0.75598	1	2mo
8788	**erichsiao6246**			0.75598	1	2mo
8789	**DarrenLin**			0.75598	1	2mo

「人工智慧」到底是怎麼回事？

第一個洞見是，每個模型都會帶來歧視。

鐵達尼號模型中對生死影響最重要的變量是：性別、船票價格、年齡。
也就是說除了先天因素，船票越貴、你存活的可能性就越高。
那假設你是一個賣保險的，為了多賺點錢，你是否會多收窮人的保險費，少收富人的保險費。

第二個洞見是，人工「不」智慧。

還有很多別的因素對存活很重要，但是我們根本沒考慮。比如說一個男人能不能存活，跟他當時距離哪個救生艇近很有關係！但是我們根本就沒有這項數據。
模型對鐵達尼號上發生了什麼一無所知，只不過是增加它猜對的概率而已。

第三個洞見是，訓練好的模型適用範圍很侷限。

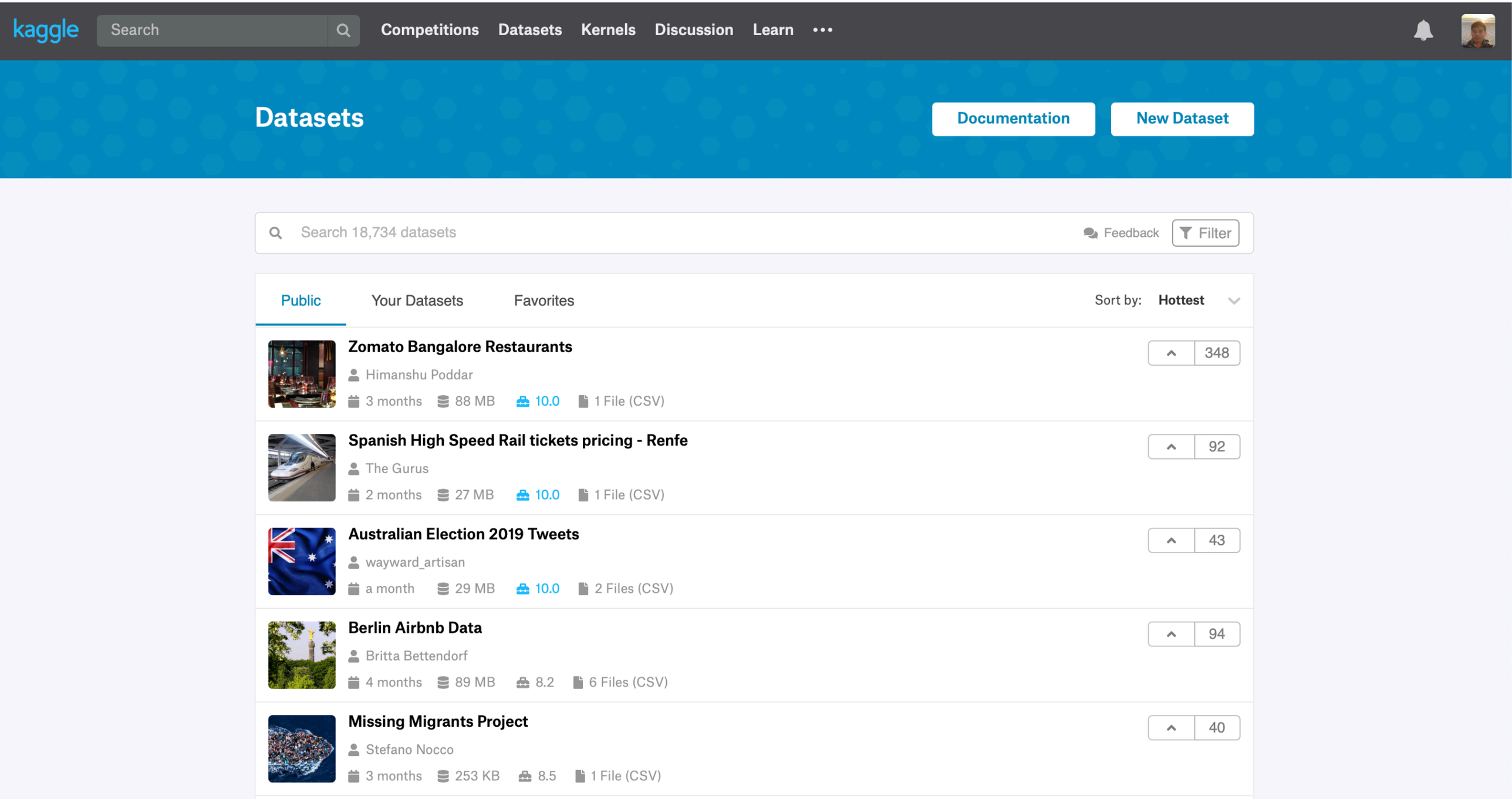
用鐵達尼號訓練出來的模型，只能準確預測這一艘船。
換另一艘船、航線、環境不同，模型就需要新的資料重新訓練。
就像在美國學會開車，但回來台灣開車還是需要適應一樣。

機器學習僅僅是一個統計模型而已。

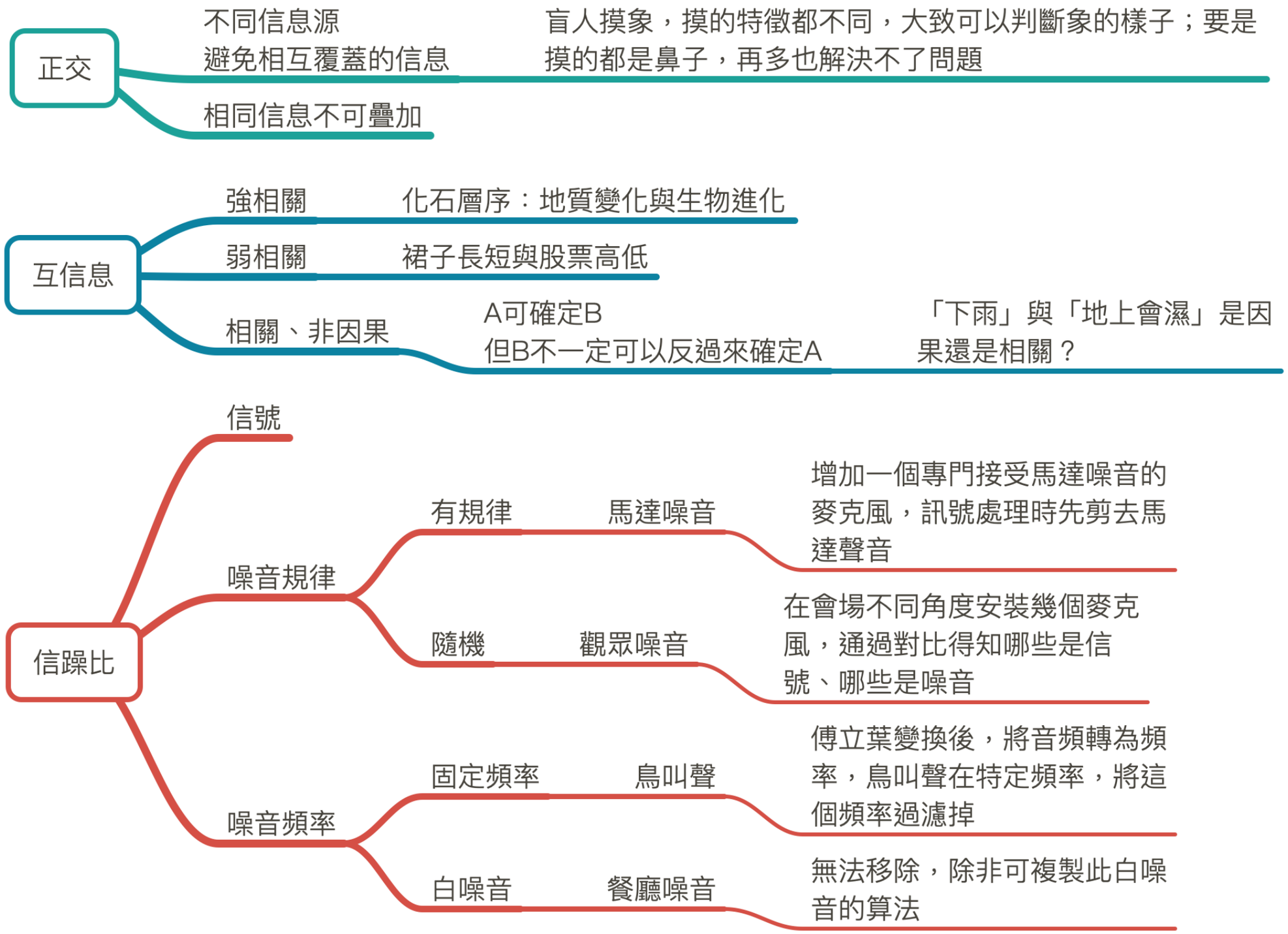
參考自《教你写一个人工智能程序》
作者：万为钢

收集數據

https://www.kaggle.com



收集有效的數據



參考自《信息论40讲》

作者：吴军

收集可量化的數據

每坪單價(萬元/坪)
24~26萬元/坪
17~19萬元/坪
28~30萬元/坪

每坪單價(低)	每坪單價(高)
24.0	26.0
17.0	19.0
28.0	30.0

```
      顏色
0      Red
1     Blue
2    Green
3      Red
4   Yellow
```

Discount_rate
0.9
30 : 5

?

- [0,1] 代表折扣率
- x:y 代表滿 x 減 y 元

收集可量化的數據

每坪單價(萬元/坪)
24~26萬元/坪
17~19萬元/坪
28~30萬元/坪

每坪單價(低)	每坪單價(高)
24.0	26.0
17.0	19.0
28.0	30.0

	顏色
0	Red
1	Blue
2	Green
3	Red
4	Yellow

	顏色	顏色_label
0	Red	2
1	Blue	0
2	Green	1
3	Red	2
4	Yellow	3

	顏色_Blue	顏色_Green	顏色_Red	顏色_Yellow
0	0	0	1	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Discount_rate
0.9
30 : 5

discount_type	discount_rate	discount_top_up	discount_off
-1	1.000000	0.0	0.0
0	0.900000	0.0	0.0
1	0.833333	30.0	5.0

- [0,1] 代表折扣率
- x:y 代表滿 x 減 y 元

收集關鍵的數據

- 例子：怎麼從太空預測股價
- 美國Berkeley大學，購買RS Metrics機構收集的衛星圖像數據（包括美國44個大型零售品牌、66,000家門店衛星圖像），分析Starbucks、Walmart、Cosco等大品牌，根據門店停車場車子數量變動，預測商家的客流量、收入、市場份額，收益率比基本回報率高出5%
- 競爭壁壘，購買一年衛星數據費用高達幾萬美金

參考自《邵恒头条》

避免被數據誤導 - 倖存者偏差

- 例子：第二次世界大戰期間，飛機要如何加強裝甲，才能降低被炮火擊落的機率
- 從安全返航的轟炸機來看，防護中彈多的地方 VS 防護中彈少的地方
- 指揮官認為「應該加強機翼的防護，因為這是最容易被擊中的位置」

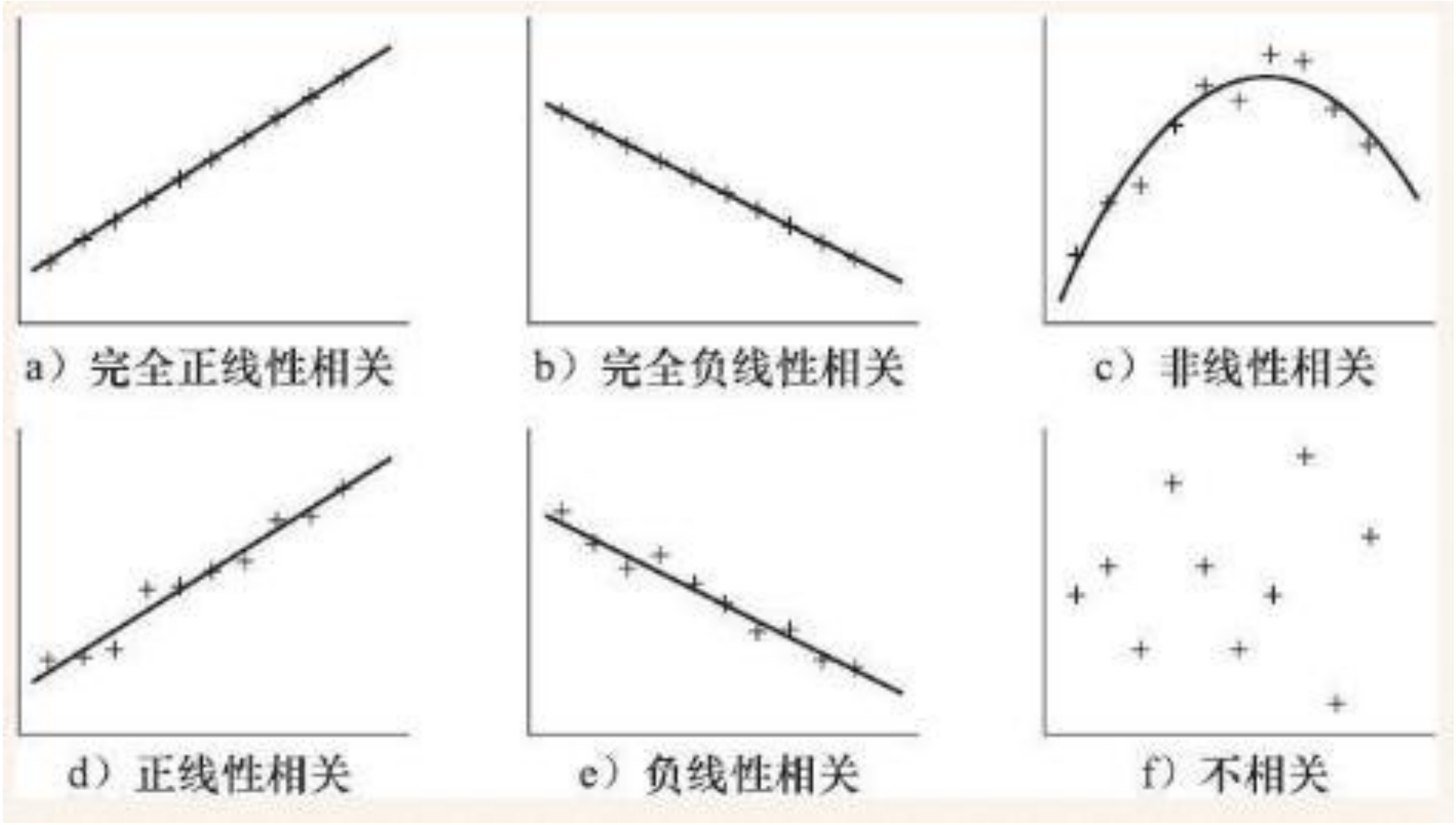
參考自《維基百科》

避免被數據誤導 - 倖存者偏差

- 例子：第二次世界大戰期間，飛機要如何加強裝甲，才能降低被炮火擊落的機率
- 從安全返航的轟炸機來看，防護中彈多的地方 VS 防護中彈少的地方
- 指揮官認為「應該加強機翼的防護，因為這是最容易被擊中的位置」
- 美國哥倫比亞大學統計學沃德教授，認為「應該保護機尾引擎，因為統計的樣本，僅包含沒有因敵火射擊而墜毀並安全返航的轟炸機」
- 機翼被擊中很多次的轟炸機，大多數仍然能夠安全返航
- 機尾彈孔較少的原因並非真的不容易中彈，而是一旦引擎中彈，其安全返航並生還的可能性就微乎其微。
- 軍方最終採取了教授提出的建議，後來證實該決策是正確的

參考自《維基百科》

觀察特徵 - 相關性



參考自「數據分析和挖掘」

```
import seaborn as sns # 另一個繪圖套件
```

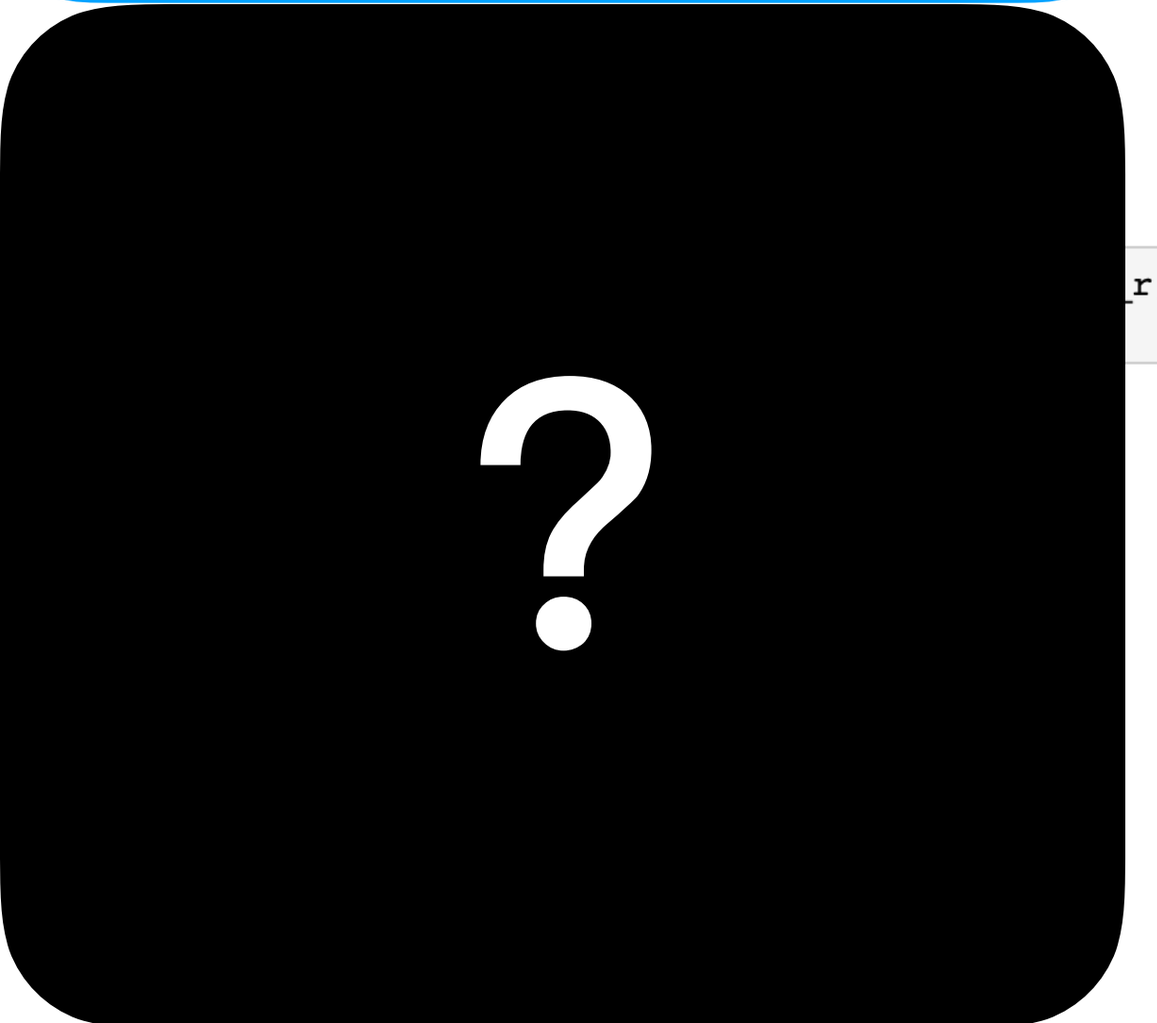
觀察特徵 - 相關性

```
data_array = [[10, 18, 60, 10],
              [40, 50, 27, 9],
              [20, 27, 40, 50],
              [50, 70, 20, 15],
              [30, 39, 31, 36]]
df = DataFrame(data_array, columns=["strenth", "hp", "agility", "lucky"])
print(df)
print()

corr = df.corr()
print(corr)
```

```
   strenth  hp  agility  lucky
0       10  18       60     10
1       40  50       27      9
2       20  27       40     50
3       50  70       20     15
4       30  39       31     36
```

```
r, vmin = -0.25, annot = True, vmax = 0.6)
```



```
import seaborn as sns # 另一個繪圖套件
```

觀察特徵 - 相關性

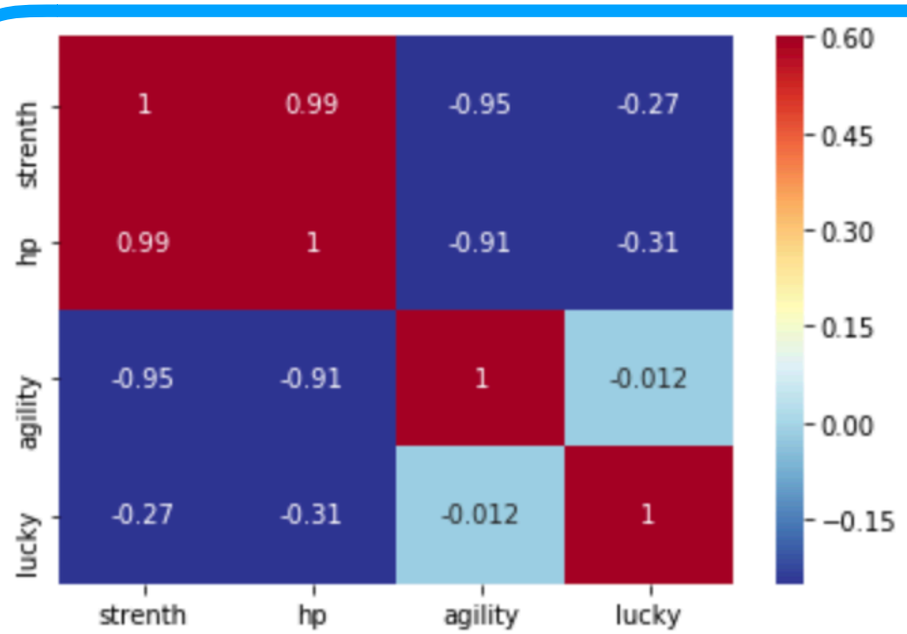
```
data_array = [[10, 18, 60, 10],
              [40, 50, 27, 9],
              [20, 27, 40, 50],
              [50, 70, 20, 15],
              [30, 39, 31, 36]]
df = DataFrame(data_array, columns=["strenth", "hp", "agility", "lucky"])
print(df)
print()

corr = df.corr()
print(corr)
```

```
   strenth  hp  agility  lucky
0       10  18       60     10
1       40  50       27      9
2       20  27       40     50
3       50  70       20     15
4       30  39       31     36

          strenth        hp   agility     lucky
strenth  1.000000  0.988454 -0.952557 -0.269616
hp       0.988454  1.000000 -0.911505 -0.312737
agility -0.952557 -0.911505  1.000000 -0.011581
lucky   -0.269616 -0.312737 -0.011581  1.000000
```

```
heatmap = sns.heatmap(corr, cmap = plt.cm.RdYlBu_r, vmin = -0.25, annot = True, vmax = 0.6)
plt.show()
```



觀察特徵 - 社群的機制

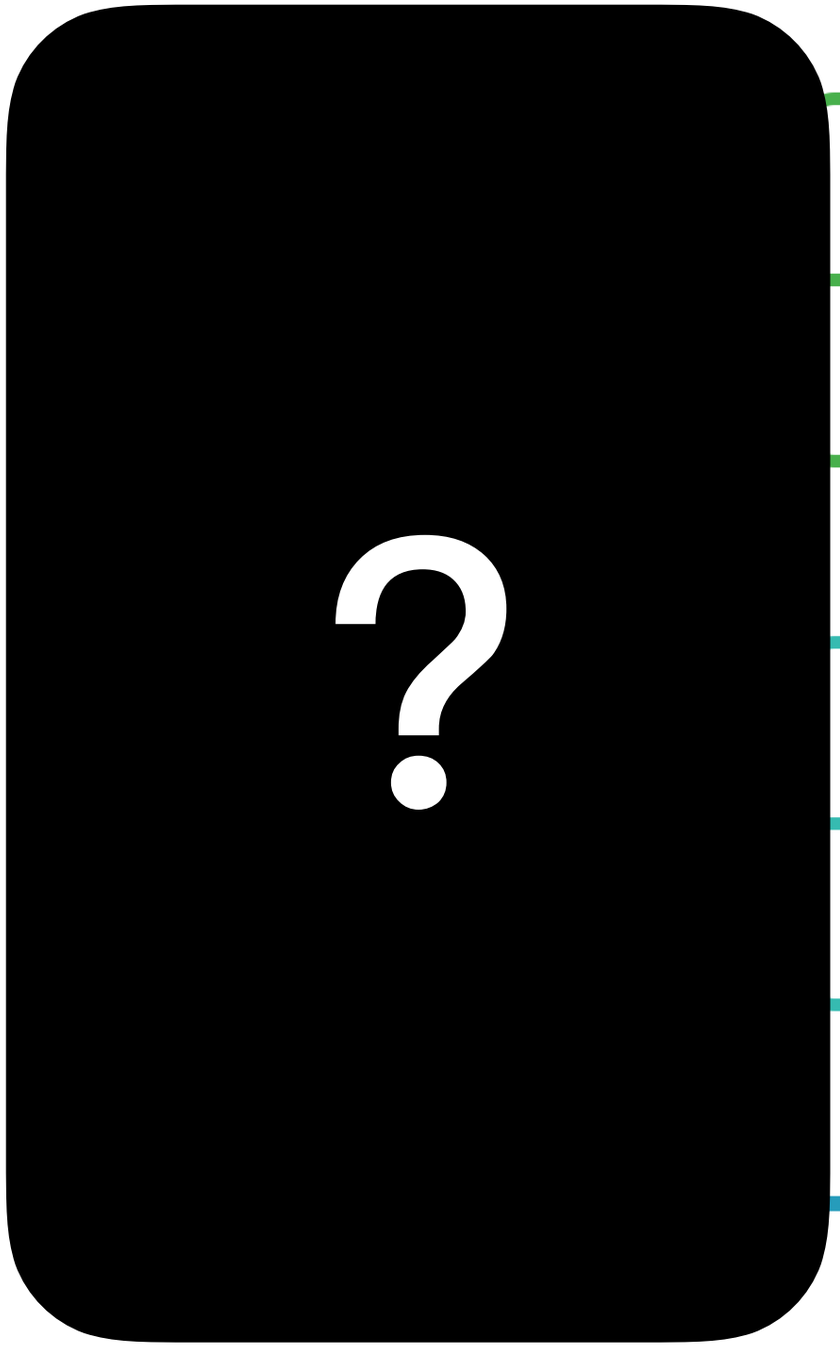
- 《新約聖經．馬太福音》第13章第12節
  「凡有的，還要加給他，叫他有餘；凡沒有的，連他所有的也要奪去」

- 老子《道德經》第77章
  「天之道，損有餘而補不足。**人之道則不然，損不足以奉有餘**。孰能有餘以奉天下，唯有道者」

- 經濟學家 Herbert Simon：**「富者越富」**

- 偏好依附法則：**「名人效應」**

觀察特徵 - 互相影響的機制

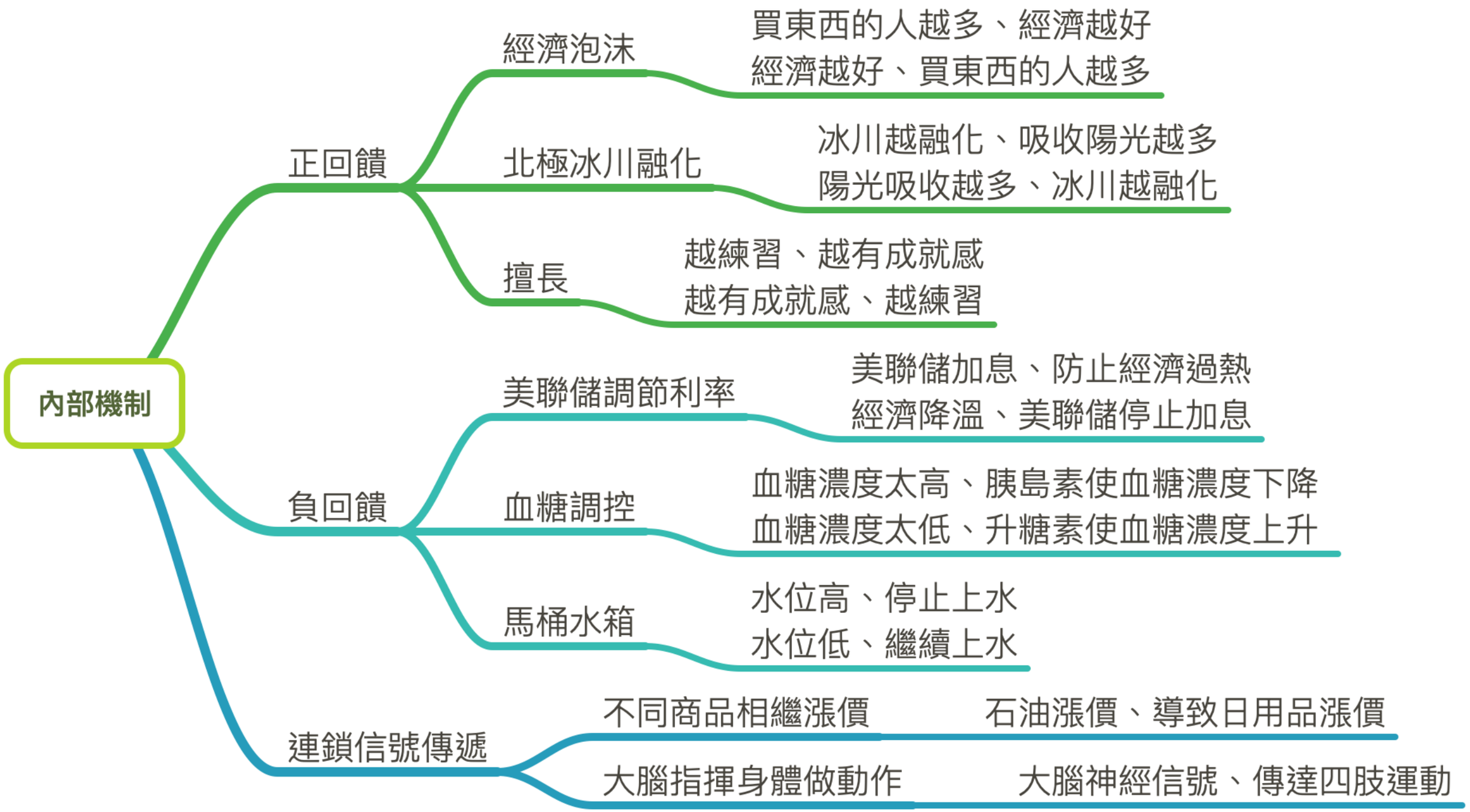


- 經濟泡沫
  - 買東西的人越多、經濟越好
  - 經濟越好、買東西的人越多
- 北極冰川融化
  - 冰川越融化、吸收陽光越多
  - 陽光吸收越多、冰川越融化
- 擅長
  - 越練習、越有成就感
  - 越有成就感、越練習
- 美聯儲調節利率
  - 美聯儲加息、防止經濟過熱
  - 經濟降溫、美聯儲停止加息
- 血糖調控
  - 血糖濃度太高、胰島素使血糖濃度下降
  - 血糖濃度太低、升糖素使血糖濃度上升
- 馬桶水箱
  - 水位高、停止上水
  - 水位低、繼續上水
- 不同商品相繼漲價
  - 石油漲價、導致日用品漲價
- 大腦指揮身體做動作
  - 大腦神經信號、傳達四肢運動

參考自《信息论40讲》

作者：吴军

觀察特徵 - 互相影響的機制



參考自《信息论40讲》

作者：吴军

特徵工程 - 用原始數據組合出新特徵

- 原始數據：身高、體重、腰圍
- 衡量身材肥胖指標

?

參考自《Nature》

特徵工程 - 用原始數據組合出新特徵

- 原始數據：身高、體重、腰圍
- 衡量身材肥胖指標
- BMI = 體重 (kg) ÷ 身高^2 (m)
- BMI 代表衡量身材肥胖的指標（正常BMI在18.5和25之間）
- 男：RFM = 64 - (20×身高（m）÷ 腰圍（m）)
  女：RFM = 76 - (20×身高（m）÷ 腰圍（m）)
- RFM 代表衡量身體中脂肪所占百分比的指標，與儀器身體掃描結果非常接近（RFM=25，代表脂肪重量占體重25%。正常體脂率，男性25%以下，女性35%以下)

參考自《Nature》

實例 - O2O優惠卷使用預測：

1. 核心問題為何？

用戶在收到優惠卷後15天內，是否使用優惠卷。

2. 資料從何而來？

競賽平台

3. 蒐集而來的數據特徵為何？

- **User_id：用戶 ID**
- **Merchant_id：商家 ID**
- **Coupon_id：優惠券** ID (null 代表無優惠券消費)
- **Discount_rate：優惠券折價**：[0,1] 代表折扣率；x:y 代表滿 x 減 y 元
- **Distance：用戶經常活動地點離商家最近距離** (x * 500 公尺), 0 表示低於 500 公尺, 10 表示大於 5 公里。
- **Date_received：優惠券取得時間**。
- **Date：購買商品時間** (如果 Date is null & Coupon_id is not null, 則該紀錄為有優惠券但未使用; 若為 Date is not null & Coupon_id is null, 則為普通消費日期; 若 Date is not null & Coupon_id is not null, 則表示優惠券消費日期)

4. 如何評估成效？

以該用戶取得該優惠券後15日內核銷預測 AUC（ROC 曲線下面積）作為評價標準

特徵工程 - 從7個原始數據，到組合出更多特徵

- User_id：用戶 ID
- Merchant_id：商家 ID
- Coupon_id：優惠券 ID
- Discount_rate：優惠券折價
- Distance：用戶經常活動地點離商家最近距離
- Date_received：優惠券取得時間。
- Date：購買商品時間

?

特徵工程 - 從7個原始數據，到組合出超過70個特徵

- User_id：用戶 ID
- Merchant_id：商家 ID
- Coupon_id：優惠券 ID
- Discount_rate：優惠券折價
- Distance：用戶經常活動地點離商家最近距離
- Date_received：優惠券取得時間。
- Date：購買商品時間

總用戶 u0
總商家 m0
總優惠券種量 d0

用戶特徵：

用戶一共消費多少次 u1
線下領取優惠卷的次數 u2
線下領取優惠券但沒有使用的次數 u3
線下領取優惠卷並核銷的次數 u4
線下普通消費次數 u5
線下領取優惠券到核銷的平均間隔時間 u6
線下正常消費的平均間隔時間 u7
最近一次優惠券消費到當前領券的時間間隔 u8
最近一次普通消費到當前領券的時間間隔 u9
用戶核銷優惠券的平均消費折率 u10
用戶核銷優惠券的最低消費折率 u11
用戶核銷優惠券的最高消費折率 u12
用戶核銷優惠券中的平均與商家距離 u13
用戶核銷優惠券中的最小與商家距離 u14
用戶核銷優惠券中的最大與商家距離 u15
用戶核銷過的不同優惠券種量 u16
用戶核銷幾個商家 u17
用戶平均核銷每個商家多少張優惠券 u18
15/u6 用戶15天內平均使用優惠卷消費幾次 u19
15/u7 用戶15天內平均會普通消費幾次 u20
u16/d1 用戶核銷過的不同優惠券數量佔所有不同優惠券的比重 u21

商家特徵：

商家一共消費次數 m1
商家優惠券被領取後不核銷次數 m2
商家優惠券被領取後核銷次數 m3
m2+m3 商家優惠券被領取次數 m4
商家普通消費次數 m5
商家優惠券核銷的平均消費折率 m6
商家優惠券核銷的最小消費折率 m7
商家優惠券核銷的最大消費折率 m8
核銷商家優惠券的不同用戶數量 m9
商家被核銷過的不同優惠券數量 m10
商家不同優惠券數量 m11
商家平均每種優惠券核銷多少張 m12
商家被核銷優惠券的平均時間 m13
商家被核銷優惠券中的平均距離 m14
商家被核銷優惠券中的最小距離 m15
商家被核銷優惠券中的最大距離 m16
商家發行的優惠券數目 m17
商家有多少人在此店領券 m18
m3/m4 商家優惠券被領取後核銷率 m19
m3/m19 商家優惠券平均每個用戶核銷多少張 m20
m10/m11 商家被核銷過的不同優惠券數量佔所有領取過的不同優惠券數量的比重 m21

優惠卷特徵：

一共發行多少張 d1
沒有使用的數目 d2
核銷多少張 d3
優惠券類型(直接優惠為0, 滿減為1) d4
滿減類優惠券滿減金額 d5
滿減類優惠券減的金額 d6
歷史上用戶領取該優惠券次數 d7
歷史上用戶核銷該優惠券次數 d8
領取優惠券是一周的第幾天 d9
領取優惠券是一月的第幾天 d10
d3/d1 歷史上用戶對該優惠券的核銷率 d11
d6/d5 優惠券折率 d12
當天所領取優惠券裡面優惠券折率排名 d13

用戶-商家組合特徵：

用戶在商家總共消費過幾次 um1
用戶領取商家的優惠券次數 um2
用戶領取商家的優惠券後不核銷次數 um3
用戶領取商家的優惠券後核銷次數 um4
用戶在商家普通消費次數 um5
um4/um2 用戶領取商家的優惠券後核銷率 um6
um1/(u4+u5) 用戶常去商家 um7
um4/u4 用戶常在商家使用優惠卷 um8

其它特徵，這部分特徵利用了賽題leakage，
都是在預測區間提取的：

預測集，商家發行的優惠券數目 m22
預測集，商家有多少人在此店領券 m23
預測集，用戶此次之前領取的所有優惠券數目 u22
預測集，用戶此次之後領取的所有優惠券數目 u23
預測集，用戶領取的所有優惠券數目 u24
預測集，用戶領取的特定優惠券數目 u25
預測集，用戶上一次領取的時間間隔 u26
預測集，用戶上下一次領取的時間間隔 u27

數據源及建議特徵工程方向

数据源大类	原始数据字段	建议特征工程方向
支付流水	支付编号	日/周/月支付频率
	支付账户	来往账户数量、账户间关联图谱
	支付时间	最早最近支付时间、支付时段分布
	支付金额	支付金额总和/平均值/最大值
	支付地点	地点类型分布、较频繁地点
	支付目的	较频繁目的
	支付状态	支付成功/失败次数
财富管理	申购编号	申购频率
	申购时间	最早最近申购时间
	申购金额	申购金额总和/平均值/最大值
	产品类型	产品类型分布、产品偏好
	产品收益	收益总和、日均收益
	持仓金额	当前持仓、历史最大持仓、日均持仓
贷款信息	申请编号	申请频率
	申请时间	最早最近申请时间
	授信金额	授信金额总和/平均值/最大值
	提现金额	授信金额总和/平均值/最大值、提现比例
	资方类型	资方个数
	申请状态	申请通过/拒绝次数
	还款时间	提前结清/正常/逾期总天数
	逾期金额	逾期金额总和/最大值
	还款状态	已结清/正常/逾期笔数
app登录	登录编号	日/周/月登录频率
	登陆时间	最早最近登录时间、时段分布
	操作类型	操作类型分布、业务线偏好
电商流水	订单编号	当月/近3个月/近6个月/近12个月订单总数
	sku编号	当月/近3个月/近6个月/近12个月商品总数
	订单时间	最早最近订单时间、近12个月有消费月份数
	订单金额	当月/近3个月/近6个月/近12个月订单总金额/订单最大金额/平均单笔订单金额
	订单状态	当月/近3个月/近6个月/近12个月实付金额占比
	分期标识	当月/近3个月/近6个月/近12个月分期订单数占比
收货地址	订单编号	当月/近3个月/近6个月/近12个月使用收货地址个数
	订单时间	收货地址使用时长
	收货地址	城市等级、小区档次、地址稳定性、是否涉黑
	地址类型	最频繁收货地址类型、工作与住宅占比
运营商信息	通话数据	当月/近3个月/近6个月/近12个月通话量/通话次数、主叫/被叫/漫游通话量/通话次数占比、通话时段分布
	流量数据	当月/近3个月/近6个月/近12个月流量、流量时段分布
	账单信息	当月/近3个月/近6个月/近12个月账单金额平均值/最大值、当月储值金额、当前欠费金额
	客户信息	在网时长、在网状态、名下手机号码数量/终端设备数量/终端品牌
	互联网访问	各类别app访问总次数/总时长/活跃天数、app类别分布、是否非法网站

参考自《京东数科》

作者：JovialCai