Cyber Security / Information Security：Fake News Detection (Day1)

NTU ANTS lab 陳廷易Leo

臺灣好厲駭講師

Sli.do: #0913

Slide@GitHub: tychen5/NLP_FakeNewsDetection/ISIP_Speaker

Agenda (Text Mining & Information Retrieval 101)

- ★ Introduction to TM & IR
  - ✓ Definition & Application
- ★ Linguistic Preprocessing
  - ✓ Tokenization
  - ✓ Normalization
  - ✓ Stemming
  - ✓ Stopwords
- ★ TF-IDF & Vector Space Model
  - ✓ Document representation
  - ✓ Cosine Similarity

Agenda (Text Mining & Information Retrieval 101)

★ IR Evaluation Metrics

  ✓ Precision / Recall / F-measure

★ Text Classification

  ✓ Multinomial Naïve Bayes

  ✓ Bernoulli Naïve Bayes

★ Feature Selection

  ✓ Chi-Square

  ✓ Log Likelihood Ratio

  ✓ Mutual Information

Text Mining & Information Retrieval (1/2)

★ 文字探勘（Text Mining）被視為是資料探勘（Data Mining）的一環

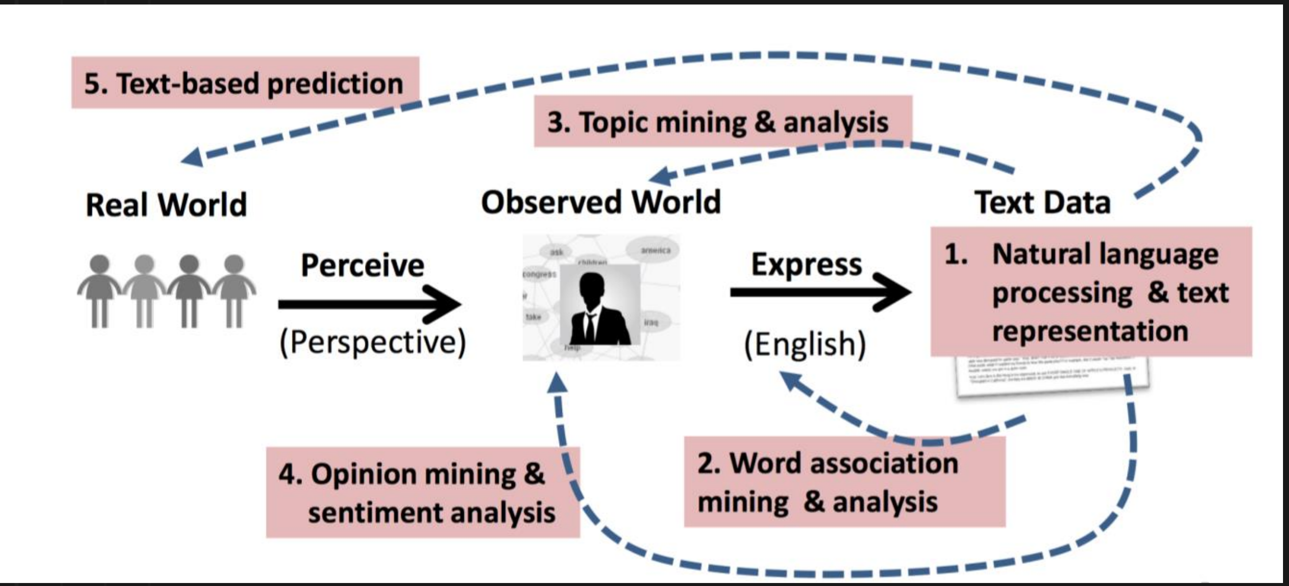
- ✓ DM: structured data. E.g., database tuples
- ✓ TM: unstructured data. E.g., documents

★ 資訊檢索（Information Retrieval）是指因應使用者對資訊的需求提供查尋的方法與查尋過程，希望能對文章進行ranking

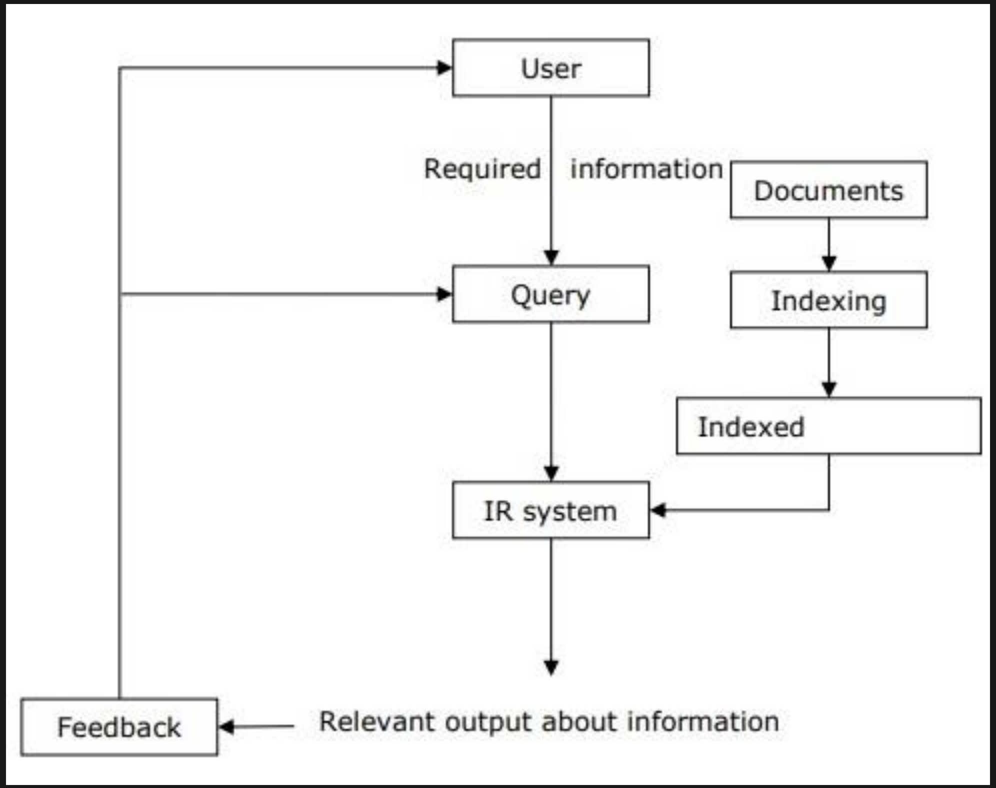
- ✓ Assists users in finding the information they require but it does not explicitly return the answers of the questions
- ✓ Informs the existence and location of documents that might consist of the required information

Text Mining & Information Retrieval (2/2)

★文字探勘（Text Mining）



★資訊檢索（Information Retrieval）



Linguistic Preprocessing

Tokenization

Normalization

Stemming / Lemmatization

Stopwords

Tokenization (1/2)

★ Chopping a character sequence up into pieces, called tokens

- ✓ 將句子或文章切割為具有意義的最小單位

★ Split on all non-alphanumeric characters

- ✓ Cat’s → Cat、s
- ✓ github.com/tychen5/NLP_FakeNewsDetection → github、 com、tychen5、NLP、FakeNewsDetection

★ Hyphens

- ✓ Allowing short hyphenated prefixes on words. E.g., co-worker
- ✓ But not longer hyphenated forms. E.g., the hold-him-back-and-drag-him-away manner
- ✓ Generalize the query to cover all three of the one word. E.g., over-eager → over-eager、over eager、overeager

Tokenization (2/2)

★ Splitting on white space

✓ 缺點：專有名詞(e.g., Los Angles)、日期(e.g., Jul 26, 2018)

✓ hyphens and non-separating whitespace can interact (e.g.,機票San Francisco-Los Angles)

★ Different language presents different issues

✓ German writes compound nouns without space (e.g., Computerlinguistik → computational linguistics)

✓ East Asian Language (e.g., Chinese, Japanese, Korean, and Tai) are written without any spaces between words (e.g.我要去總統府 → 總統、府?? or 總統府??)

■ 3-gram: 我要去、要去總、去總統、總統府

★ 小結

✓ Tokenize沒有一定且正確的通用方法，完全會需要因應不同情境應用決定該如何斷詞

Normalization (1/2)

★ Implicit (透過專家所撰寫的規則進行轉換)

- ✓看起來不一樣但其實是一樣的字，希望可以轉換為一樣的格式
  - ■ E.g., 外來文、日文、同義多詞(‘on-line’、‘online’；‘USA’、‘U.S.A’)
- ✓ 可能潛在問題E.g., ‘C.A.T.’ (美國公司)→ ‘cat’(可愛動物)

★ Explicit (同義詞對應字典)

- ✓ Query expansion: 將query所對應的相關字詞都當成query去尋找documents (浪費時間)
- ✓ Expand during index construction: 在documents建立的時候就將相關同義詞都置入 (浪費空間)
- ✓ E.g., ‘car’ and ‘automobile’ belong to the same class

Normalization (2/2)

- ★ 其他Normalize方法 (其實都只是為了迎合人的惰性..)
  - ✓重音符號、音標符號加與不加
    - ■ E.g., ‘naïve’ →‘naive’
  - ✓大小寫傻傻分不清
    - ■ E.g., ‘Taiwan’ →‘taiwan’
    - ■ 可能潛在問題E.g., ‘Bush’(人名)、‘bush’(灌木)
- ★ 不同語言具有不同的問題
  - ✓ E.g., 法文的the因為不同的gender/number而有不同的表示法le, la, l’, les
  - ✓ E.g., 日文由漢字、片假名、平假命系統所構成，相同的字可能因為不同的撰寫系統而不同
  - ✓ 一個document包含多國語言、跨語言翻譯問題E.g., Beijing、Peking

Stemming & Lemmatization (1/2)

★ For grammatical reasons, documents are going to use different forms of a word (stemming可以算是比較進階的normalization，還原字根如：第三人稱、單複數)

- ✓ E.g., ‘organize’, ‘organizes’, ‘organizing’
- ✓ E.g., car, cars, car’s, cars’ → car

★ Stemming (暴力直接砍字尾，還原回來的不一定是正確的字)

- ✓ E.g., ‘automate(s)’, ‘automatic’, ‘automation’ →‘automat’

★ Lemmatization (進行時態分析、套用字典，還原到最正確的字)

- ✓ E.g., ‘am’, ‘are’, ‘is’ → ‘be’
- ✓ E.g., 如果saw在句子中是名詞就還原回自己，如果判斷是動詞就還原為see

Stemming & Lemmatization (2/2)

★ 小結

- ✓ 在NLP實務上的應用中發現Stemming的效果在搜尋文章時較Lemmatization還要好
- ✓ 然而在某些情境中如operate與operating都會被stem成oper，在英文中operate system跟operating system可能代表不同的含意，但在此情況下若使用者query為operating system可能就會找出其他較無關緊要的句子如operate and system導致系統精確率下降

Stop Words

- ★ 旨在將文章或句子中較沒意義或出現太頻繁的字詞濾除，以避免過多的雜訊並提升搜索效能與效率
  - ✓ E.g., a, an, and, are, as, …, was, were, with, …
- ★ 常見的做法是將所有文章的terms進行詞頻的排序，將出現較頻繁的字詞濾除
- ★ 潛在風險：在某些特定應用情境如歌詞、詩、童謠等等可能都是由那些常見的words所組成，便不太建議濾除
  - ✓ E.g., "*let it be*", "*to be or not to be*", "*As we may think*" …
- ★ 目前較實務的作法尚會透過權重計算的方式(如: TF-IDF)來挑選Stop Words。(欲知後事如何，且聽下回分曉..)

實務練習Part 1

Tokenization

Normalization

Stemming

Stopwords

Colab使用教學步驟

1. 利用Chrome登入Google Drive
2. 進入連結https://drive.google.com/drive/u/1/folders/1fhg-M8ijY0knxkvJyGxnTxGprVQ_9DZA 點選「ISIP_tychen5」資料夾下拉式選單，再點擊「新增至我的雲端硬碟」
3. 在「ISIP_tychen5」資料夾中創建一個自己名字或暱稱的資料夾
   - 更改自己資料夾的共用權限為可以檢視
4. 在自己的資料夾中點選新增=>更多=>Google Colaboratory (若無此選項則需先點及來源網站進行連結後再重新整理)

Colab掛載GD存取資料

1. 利用下方程式碼並執行

```
import os
from google.colab import drive
drive.mount('/content/drive')

data_dir = "/content/drive/My Drive/ISIP_tychen5/data/"
my_dir = "/content/drive/My Drive/ISIP_tychen5/Leo/"
```

2. 點擊URL並登入google帳號、允許權限後複製授權碼，貼上至輸入方塊中併按下enter完成掛載

實務練習Part 1

★ 目標：

- ✓ 利用上述所教的技巧，設計出屬於自己的文本前處理邏輯，撰寫為程式以抽取出data資料夾裡任一篇文章中重要的terms

★ Implement Tips & Guideline：

- ✓ Tokenization.
- ✓ Lowercasing everything.
- ✓ Stemming using Porter’s algorithm.
- ✓ Stopword removal.
- ✓ Save the result as a txt file.

實務練習Part 1 (以28.txt為範例)

★ 輸入：

And Yugoslav authorities are planning the arrest of eleven coal miners and two opposition politicians on suspicion of sabotage, that's in connection with strike action against President Slobodan Milosevic. You are listening to BBC news for The World.

★ 範例輸出(因人而異，無一定標準答案)(Leo/result/output.txt)：

yugoslav author plan arrest eleven coal miner two opposit politician suspicion sabotag connect strike action presid slobodan milosev listen bbc news world

※ 完整參考範例做法請參閱GD中Leo資料夾裡的SampleCode_Leo.ipynb ※

TF-IDF & Vector Space Model

Term Frequency and Weighting

Inverse Document Frequency

TF-IDF weighting

Cosine Similarity & Computing Vector Scores

Term Frequency and Weighting

★ A logical consideration：

- ✓ 如果有一個query term(s)頻繁地出現在某文章中，表示該文章較為重要有關聯
- ✓ 因此希望可以透過scoring mechanism計算出query terms與文章間的match score，以做為權重來排序關聯性文章

★ Term Frequency (TF)：

- ✓ 計算某個term在文章中的出現次數

★ Bag-of-Words model：

- ✓ 一種文章表示法。並沒有考慮文章中term出現的順序，只考慮TF或是有無出現
- ✓ E.g., “term i and term j are synonyms” → { <term,2>, <and,1>, …}
- ✓ E.g., “Leo love cat” == “cat love Leo”

Inverse Document Frequency (1/2)

★ Problem of term frequency weighting：

✓ 在一篇文章中每個term都一樣重要？

✓ 有些字顯然在文章中較具有代表性，但重要性會是幾倍呢？

★ Discriminating power

✓ For instance, a collection of documents on the auto industry is likely to have the term ‘auto’ in almost every document

✓ Need a mechanism for reducing the effect of terms that occur too often in the collection

Inverse Document Frequency (2/2)

★ Document Frequency (DF)：

- ✓ The number of documents in the collection that contain a term

★ Inverse document frequency (IDF)：

- ✓ $idf_t = \log \frac{N}{df_t}$
- ✓ 最小值是0，最大值是logN
- ✓ 在整個corpus中，出現在比較少文章的term比較具有discriminating power(IDF值大)，哪些term太常出現IDF值就小

★ Example of IDF values of terms in the Reuters collection of 806,791 documents：

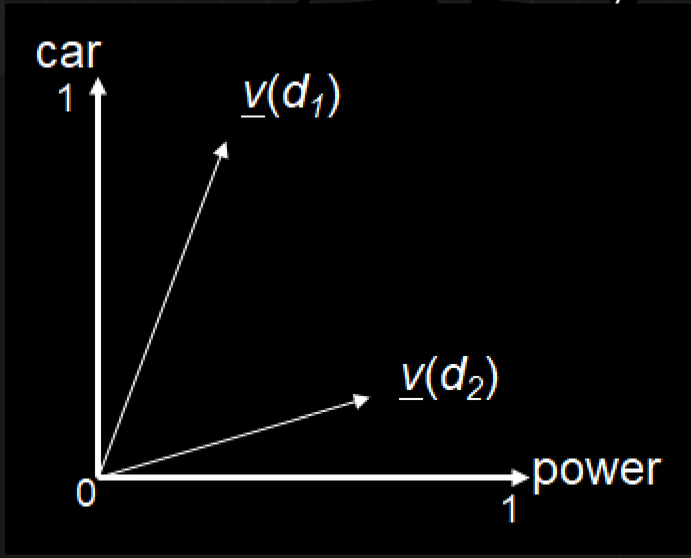
term	DF	IDF
'car'	18,165	1.65
'cat'	6,723	2.08
'insurance'	19,241	1.62
'best'	25,235	1.5

TF-IDF weighting

★ TF-IDF combines the concept of **term frequency** and **inverse document frequency** to assign the weight of term *t* in document *d* as follows：

- ✓ $tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t$
- ✓ **High**, when t occurs many times in d and appears within a small number of documents
- ✓ **Low**, when t is a rare term in d and occurs in virtually all documents in the collection.

Vector Space Model (1/2)



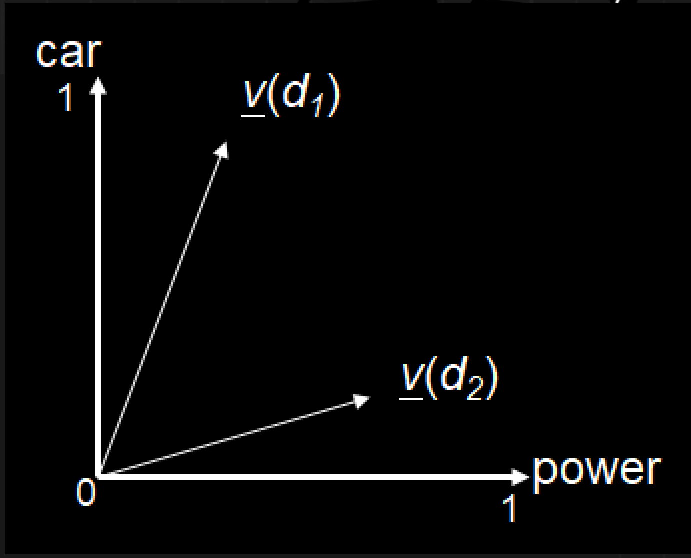
★ We may view each document (or query) as a vector：

- ✓ 向量中每一個維度就代表一個字詞
- ✓ 一個字詞可以用該篇文章所求得的TF-IDF值來代表

★ Cosine Similarity

- ✓ 考量1：如何測量兩個向量的相似性？
  - ■ 測量兩個向量的夾角的餘弦值
- ✓ 考量2：不同文章的長度不同、出現term的數量不一？
  - ■ normalized vectors to unit vector

Vector Space Model (2/2)



★ Cosine Similarity

- ✓ The inner product of the unit vectors
- ✓ $sim(d_1, d_2) = \frac{\underline{V}(d_1)}{|\underline{V}(d_1)|} \cdot \frac{\underline{V}(d_2)}{|\underline{V}(d_2)|}$
- ✓ The range of cosine similarity is on [0,1]
- ✓ Given term-document matrix to find similar documents

E.g., *sim(SaS, PaP) = 0.999* 、*sim(SaS, WH) = 0.888*

term\book	SaS	PaP	WH
'affection'	0.996	0.993	0.847
'jealous'	0.087	0.120	0.466
'gossip'	0.017	0	0.254

實務練習Part 2

Dictionary

DF

TF-IDF unit vector

Cosine Similarity

實務練習Part 2

★ 目標：

- ✓ 利用上述所教的技巧及自己Part1所設計的文本前處理邏輯，撰寫為程式以抽取出各篇文章的tf-idf vector並實作cosine similarity

★ Implement Tips & Guideline：

- ✓ Construct a dictionary based on the terms extracted from the given documents.
- ✓ Record the document frequency of each term.
- ✓ Transfer each document into a tf-idf unit vector.

$$idf_t = \log_{10} \frac{N}{df_t}$$

- ✓ Write a function *cosine(Doc_x, Doc_y)* which loads the tf-idf vectors of documents x and y and returns their cosine similarity.

實務練習Part 2

★ 範例輸出(因人而異，無一定標準答案)(Leo/result/tfidf/ID.txt)：

t_index	tf-idf
2	0.731
11	0.218
22	0.014
…	…

★ Pseudocode： *※完整參考範例做法請參閱GD中Leo資料夾裡的SampleCode_Leo.ipynb※*

```
CosineScore(q)
  float Score[N] = 0
  calculate normalized tf-idf weight for each query term
  for each query term t
    fetch postings list for t
    for each pair (d, w_{t,d}) in postings list
      add w_{t,d} x w_{t,q} to Scores[d]

  return Top K documents of Scores[]
```

IR Evaluation Metrics

Precision

Recall

Accuracy

F1

Evaluation of Unranked Retrieval Sets (1/3)

★ The two most frequent and basic measures for information retrieval effectiveness are precision and recall：

✓ $\text{Precision} = \dfrac{\#(\text{relevant item retrieved})}{\#(\text{retrieved items})}$ = tp / (tp + fp)

✓ $\text{Recall} = \dfrac{\#(\text{relevant item retrieved})}{\#(\text{relevant items})}$ = tp / (tp + fn)

★ Contingency table

contingency table	Relevant	Not relevant
Retrieved	# of true positives (tp)	# of false positives (fp)
Not retrieved	# of false negatives (fn)	# of true negatives (tn)

Evaluation of Unranked Retrieval Sets (2/3)

★ Accuracy is often not an appropriate measure for information retrieval problems：

- ✓ $$\text{Accuracy} = \frac{(tp + tn)}{(tp + fp + fn + tn)}$$
- ✓ 對的真的對、錯的真的錯之比例

★ Accuracy is often not an appropriate measure for information retrieval problems

- ✓ In almost all circumstances, the data is extremely skewed – normally over 99.9% of the documents are in the not relevant category
- ✓ A system can have a great accuracy even if it labels all the documents as non-relevant

Evaluation of Unranked Retrieval Sets (3/3)

★ How to combine (or average) precision and recall into a single value?

- ✓ arithmetic mean
- ✓ max
- ✓ min
- ✓ weighted harmonic mean (**F measure**)

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

where α in [0, 1], representing the weights of *P* and *R*

- ■ When α=1/2, we have balanced precision and recall, and the F measure is called **F1**

$$F_1 = \frac{2PR}{P+R}$$

Evaluation of Text Classification

★ Macro-averaging

- ✓ Compute a simple average over classes

★ Micro-averaging

- ✓ First pool per-document decisions across classes
- ✓ Then compute a measure on the pooled contingency table

	class 1	
	truth yes	truth no
call yes	10	10
call no	10	970

	class 2	
	truth yes	truth no
call yes	90	10
call no	10	890

	pooled table	
	truth yes	truth no
call yes	100	20
call no	20	1860

Macro-averaging precision: (0.5+0.9)/2=0.7
Macro-averaging recall: (0.5+0.9)/2=0.7

Micro-averaging precision: 100/(100+20)=0.833
Micro-averaging recall: 100/(100+20)=0.833

Text Classification

Multinomial Naïve Bayes Model

Bernoulli Naïve Bayes Model

Naïve Bayes Text Classification (1/3)

★ Naïve Bayes classification is a probabilistic learning method. To find the class with the maximum a posteriori (MAP) probability class for the document.

- ✓ $c_{map} = \underset{c \in C}{\operatorname{argmax}} P(c|d) \quad = \underset{c \in C}{\operatorname{argmax}} P(c)P(d|c)$

★ There are two different ways to set up an NB text classifier：

- ✓ **Multinomial model**：represents each document as an M-dimensional vector of frequencies (E.g., $P(d|c) = P(\langle tf_{1,d}, \ldots, tf_{M,d} \rangle \mid c) = \mathrm{P}(\langle 9, \ldots, 13 \rangle \mid \mathrm{cat})$)
- ✓ **Bernoulli model** ：similar to the binary independence model (E.g., $P(d|c) = P(\langle e_1, \ldots, e_M \rangle \mid c) = \mathrm{P}(\langle 1, \ldots, 1 \mid \mathrm{cat} \rangle)$)

Naïve Bayes Text Classification (2/3)

★ Naïve Bayes conditional independence assumption

- ✓ $P(d|c) = P(t_1|c)P(t_2|t_1,c)P(t_3|t_1,t_2,c)\ldots P(t_{n_d}|t_1,t_2,\ldots,t_{n_d-1},c)$
- ✓ 第一個位子可以是term1、term2、...termM；第二個位子也是…；假設長度為max，則probability parameters大約會需要M^{max}個
- ✓ 為了減少參數量NB假設term之間彼此獨立，出現什麼字跟前面出現過什麼字無關，所以參數量可以減少為M*max個

★ Positional independence assumption：

- ✓ 因為max長度不固定、data sparseness，相同term在不同位子的機率會不一樣E.g., $P(X_1\text{=city} \mid c) \neq P(X_3\text{=city} \mid c)$，為了model feasibility進一步減少參數，將假設機率可以共用
- ✓ 不管term的位子在哪機率都共用(參數剩下M (dictionary size))，如同bag of words，不考慮順序位置關係。E.g., P(“TSMC merged with HTC” | c) = P(“HTC merged with TSMC” | c) = $P(X\text{=TSMC}|c)\, P(X\text{=merged}|c)\, P(X\text{=with}|c)\, P(X\text{=HTC}|c)$

Naïve Bayes Text Classification (3/3)

- ★ Naïve Bayes is so called because the independence assumptions we have just made are indeed very naïve
  - ✓ 可簡化成只要看第一個位子term的機率*第二個位子term的機率*...再乘上prior即可看哪個class的機率較大
- ★ NB雖然predict出來的分數排名比例跟原本的樣子有落差，但只要是最高分的就是他所屬的class即可
- ★ NB不論在訓練階段或是推論階段的效率皆與資料集僅呈線性關係
- ★ 因為NB的efficiency跟effectiveness因此成為文字分類的baseline

Multinomial Naïve Bayes (1/2)

★ $c_{map} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{1 \le k \le n_d} P(X = t_k | c)$

✓ $P(c) = \frac{N_c}{N}$; $P(X = t_k | c) = \frac{T_{ct_k}}{\sum_{t' \in V} T_{ct'}}$

★ Many conditional probabilities are multiplied → result in a **floating point underflow**：

✓ $c_{map} = \underset{c \in C}{\operatorname{argmax}} [\log P(c) + \sum_{1 \le k \le n_d} \log P(X = t_k | c)]$

Multinomial Naïve Bayes (2/2)

★ **Zero probability** for a term-class combination that did not occur in the training data

- ✓ E.g., If occurrences of the term WTO in the training data only occurred in China documents. Then the P(X=WTO | c) for the other classes will be zero.

★ Add-one smoothing (Laplace smoothing)：

- ✓ $P(X=t_k|c) = \frac{T_{ct_k}+1}{\sum_{t'\in V}(T_{ct'}+1)} = \frac{T_{ct_k}+1}{\sum_{t'\in V}(T_{ct'})+|V|}$

The Bernoulli Model

★ $c_{map} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{1 \le i \le M} P(U_i = e_i | c)$

✓ $P(c) = \frac{N_c}{N}$; $P(U_i = 1|c) = \frac{N_{ct_i}}{N_c}$; $P(U_i = 0|c) = 1 - P(U_i = 1|c)$

★ To avoid zero probabilities, conditional probabilities：

✓ $P(U_i = 1|c) = \frac{N_{ct_i}+1}{N_c+2}$; $P(U_i = 0|c) = 1 - P(U_i = 1|c)$

Feature Selection

Chi-Square

LLR

PMI

EMI

Why Feature Selection？

- ★ 為了減少noise features，以讓classifier能夠更具efficiency，得以知道哪些term才是對分類真正有幫助的
  - ✓ 每個term對分類準確度的貢獻度未定，並不是多加一個term效果就會增加多少
  - ✓ 有用的feature通常比較少，但若是有用的則挑越多分類效果越好
  - ✓ 若是將$feature$全部採納可能反而會影響分類效果
- ★ 正指標或是反指標都有可能，feature selection只是代表某個term與該類別有否關係

χ^2 (Chi-Square)

★ Test the independence of two random variables

★ In feature selection, the two random variables are：

- ✓ Occurrence of the term: e_t=0/1, absent or present
- ✓ Occurrence of the class e_c=0/1, not about c or about c

★ $\chi^2(D,t,c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$

- ✓ $E_{e_t=1,e_c=1} = N * P(t:present \wedge c:on\ topic) = N * P(t:present) * P(c:on\ topic)$

		term *t*		
		present	*absent*	
class *c*	*on topic*	49	141	190
	off topic	27,652	774,106	801758
		27701	774247	801948

Likelihood Ratios (LLR)

★ Have the advantage that the statistic we are computing is more interpretable than the χ^2 statistic

★ Using MLE and treat the corpus as binomial experiments：

$$-2\log \frac{\left(\frac{n_{11}+n_{01}}{N}\right)^{n_{11}}\left(1-\frac{n_{11}+n_{01}}{N}\right)^{n_{10}} * \left(\frac{n_{11}+n_{01}}{N}\right)^{n_{01}}\left(1-\frac{n_{11}+n_{01}}{N}\right)^{n_{00}}}{\left(\frac{n_{11}}{n_{11}+n_{10}}\right)^{n_{11}}\left(1-\frac{n_{11}}{n_{11}+n_{10}}\right)^{n_{10}} * \left(\frac{n_{01}}{n_{01}+n_{00}}\right)^{n_{01}}\left(1-\frac{n_{01}}{n_{01}+n_{00}}\right)^{n_{00}}}$$

		term *t*	
		present	*absent*
class *c*	*on topic*	n_{11}	n_{10}
	off topic	n_{01}	n_{00}

$N=n_{11}+n_{10}+n_{01}+n_{00}$

Pointwise Mutual Information (PMI)

★ A statistical approach used in modeling word associations

★ To measure the association between term *t* and class *c*：

- ✓ $I(t,c) = \log_2 \frac{P(t \wedge c)}{P(t)P(c)} = \log_2 \frac{P(c|t)}{P(c)} = -\log_2 P\,(c) - (-\log_2 P\,(c|t))$
- ✓ In information theory, -logP(x) can be regarded as the degree of uncertainty of a certain event x
- ✓ Values close to 0 indicate independence

		term *t*	
		present	*absent*
class *c*	*on topic*	3	26
	off topic	17	4

$$I(t,c) = \log_2 \frac{3/50}{20/50 * 29/50} = -1.95$$

Expected Mutual Information (MI)

★ The expected mutual information (MI) of term t and class c is：

- ✓ $I(T,C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(e_t, e_c) \log \frac{P(e_t,e_c)}{P(e_t)P(e_c)}$

		term t	
		present	*absent*
class c	*on topic*	**3**	26
	off topic	17	4

$$\log_2 \frac{3/50}{20/50 * 29/50} + \log_2 \frac{26/50}{30/50 * 29/50} + \ldots$$

實務練習Part 3

Feature Selection

Multinomial NB Classifier

Smoothing

Micro-averaged F1-score

實務練習Part 3

★ 目標：

- ✓ 利用上述所教的feature selection技巧，從所有的term dictionary當中僅挑選出500個以內的字詞做為文字分類器所使用的字典，實作MNB利用訓練資料集將測試資料集進行分類

★ Implement Tips & Guideline：

- ✓ 訓練資料集位在/content/drive/My Drive/ISIP_tychen5/kaggle/training.txt中，第一個column代表類別標籤(共13類)，後面為文章檔名代號
- ✓ Generate an output file that records your classification results. (Exclude all training documents)
- ✓ Employ a feature selection method and use less than 500 terms in your classification. (When classify a testing document, terms not in the selected vocabulary are ignored)
- ✓ To avoid zero probabilities, calculate P(X=t|c) by using add-one smoothing.

實務練習Part 3

★ MNB algorithm (training phase)：

```
TrainMultinomialNB(C, D)
V ← ExtractVocabulary(D)
N ← CountDocs(D)
for each c in C
do
    N_c ← CountDocsInClass(D, c)
    prior[c] ← N_c / N
    text_c ← ConcatenateTextOfAllDocsInClass(D, c)
    for each t in V
    do
        T_ct ← CountTokensOfTerm(text_c, t)
    for each t in V
    do
        condprob[t][c] ← (T_ct+1) / ∑(T_ct'+1)
return V, prior, condprob
```

$P(c)$

$P(X=t|c)$

$$P(X = t_k|c) = \frac{T_{ct_k} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct_k} + 1}{\sum_{t' \in V}(T_{ct'}) + |V|}$$

實務練習Part 3

★ MNB algorithm (testing phase)：

```
ApplyMultinomialNB(C, V, prior, condprob, d)
W ← ExtractTokensFromDoc(V, d)
for each c in C
do
    score[c] ← log prior[c]
    for each t in W
    do
        score[c] += log condprob[t][c]
return argmax_c score[c]
```

$$c_{map} = \underset{c \in C}{\operatorname{argmax}}[\log P\,(c) + \sum_{1 \le k \le n_d} \log P\,(X = t_k|c)]$$

※ 完整參考範例做法請參閱GD中Leo資料夾裡的SampleCode_Leo.ipynb ※

實務練習Part 3

★ 分類結果以CSV檔輸出(須包含header、id與Value值皆為文字string型態)：

id	Value
17	2
18	2
20	2
21	2

★ Evaluation：

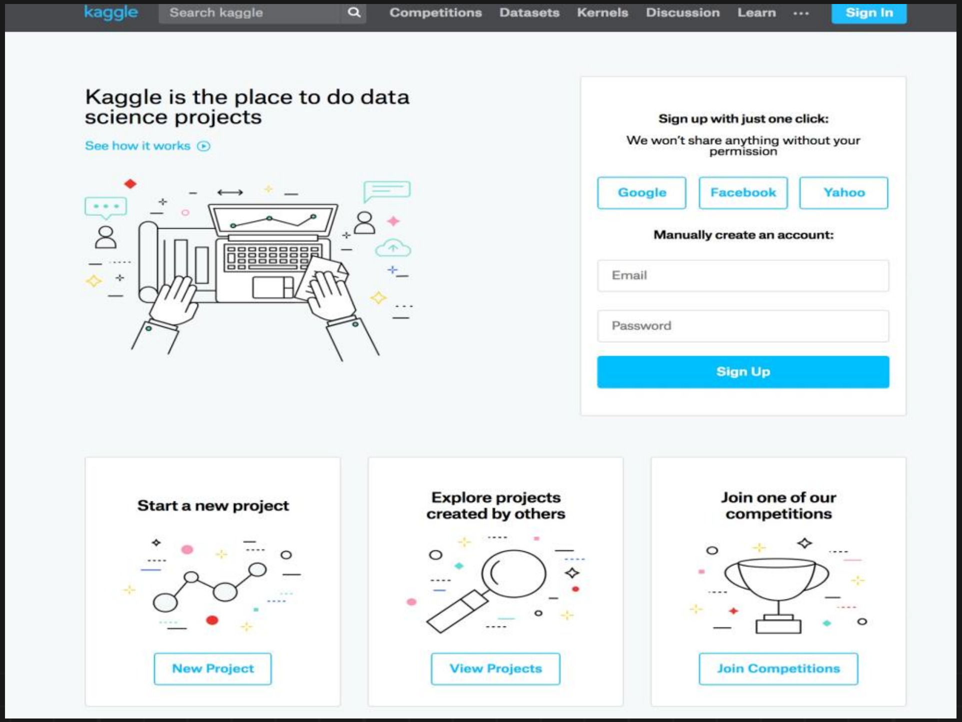
- ✓ Micro-averaged F1-score
- ✓ 測試結果：https://www.kaggle.com/t/9fcbe5ded5464242a7055a9039d8a838

(如果尚未登入，點選連結會導向登入頁面，但登入後有可能直接跳回首頁，建議先登入再點選連結，就可順利利進入kaggle頁面)

※完整參考範例做法請參閱GD中Leo資料夾裡的SampleCode_Leo.ipynb※

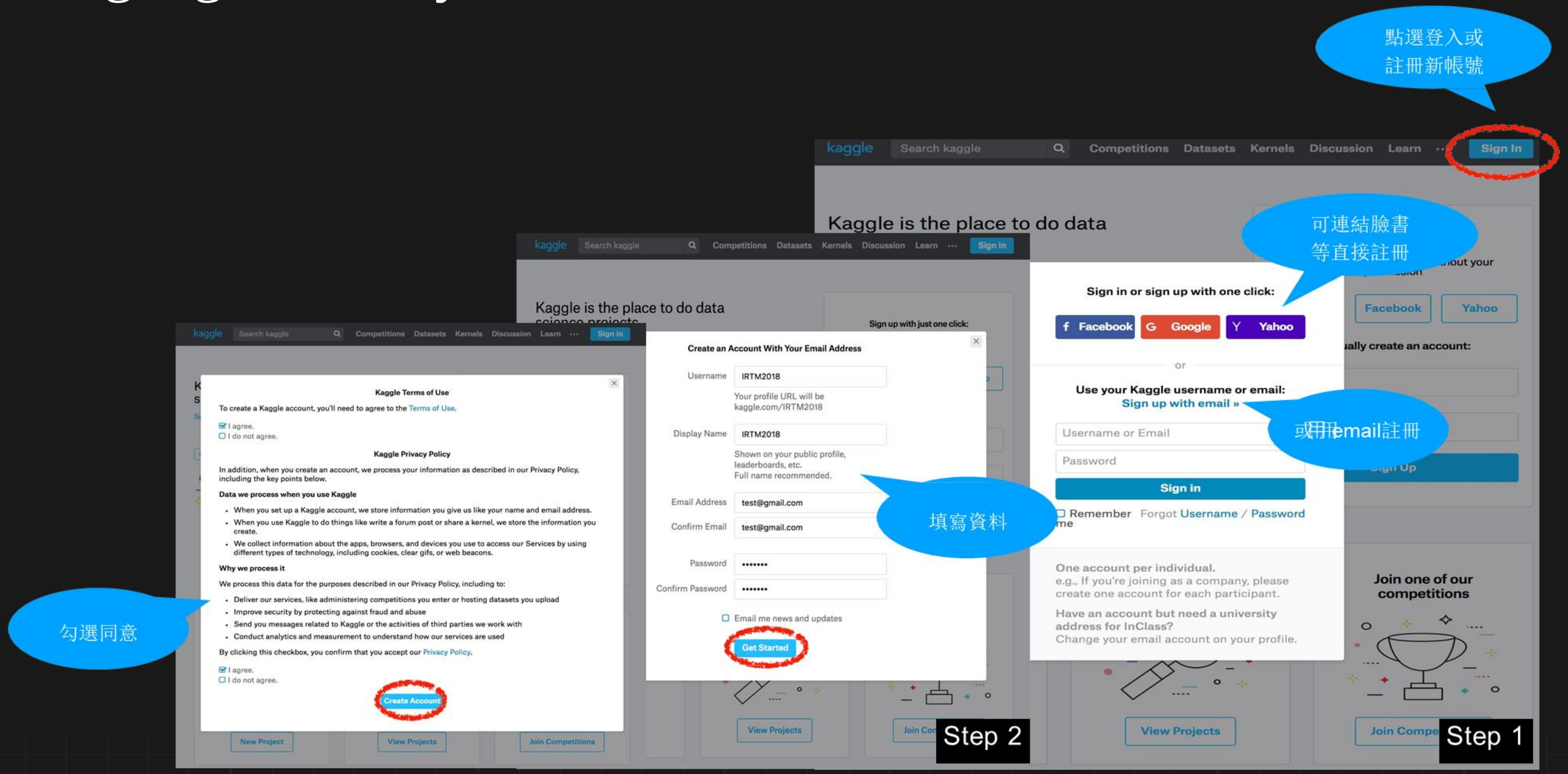
Introduction to Kaggle

★ Kaggle是一個資料科學競賽平台，每個人都可在其上發布資料集舉辦比賽，也可以參與其他人舉辦的比賽，期望透過競賽排名，集合眾人力量找出最好的模型。



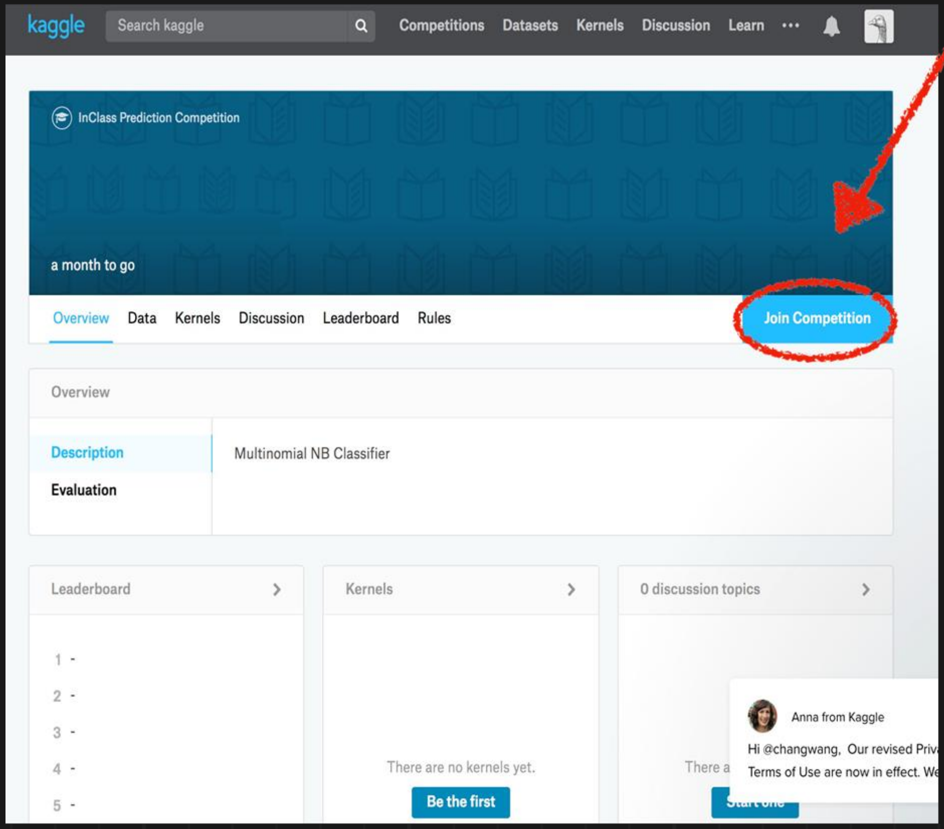
Kaggle註冊步驟

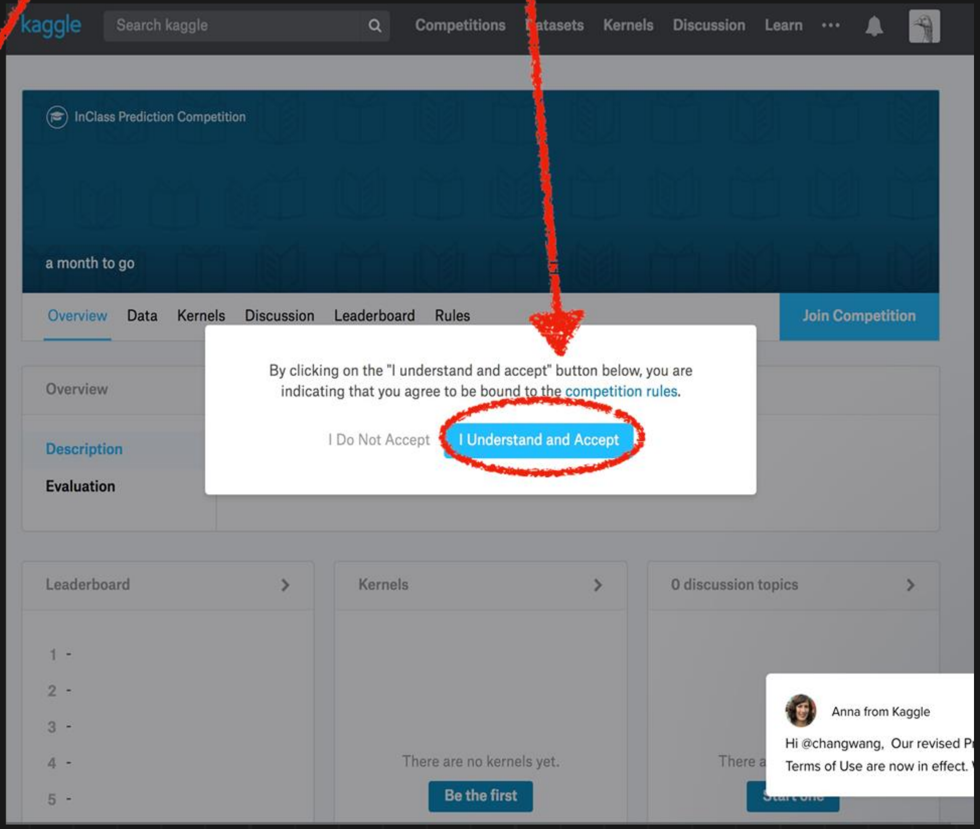
1. 點選首頁右上角角的sign in，選擇登入或註冊新帳號
2. 連結google、fb、yahoo帳號 or 使用email註冊並填寫基本資料



參加Kaggle競賽

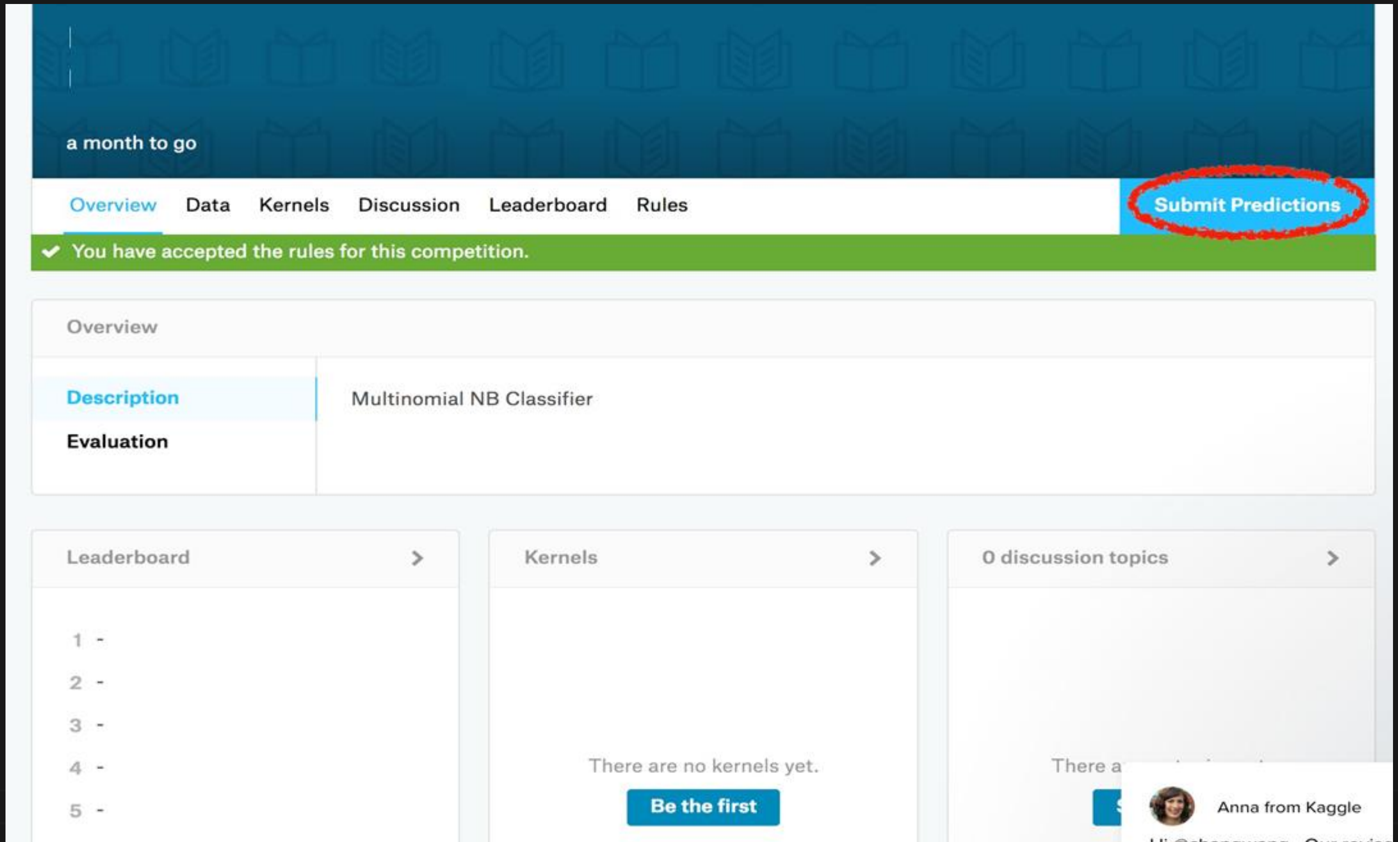
1. 成功進入競賽畫面後，點選右上角的Join Competition，再同意競賽規則即可





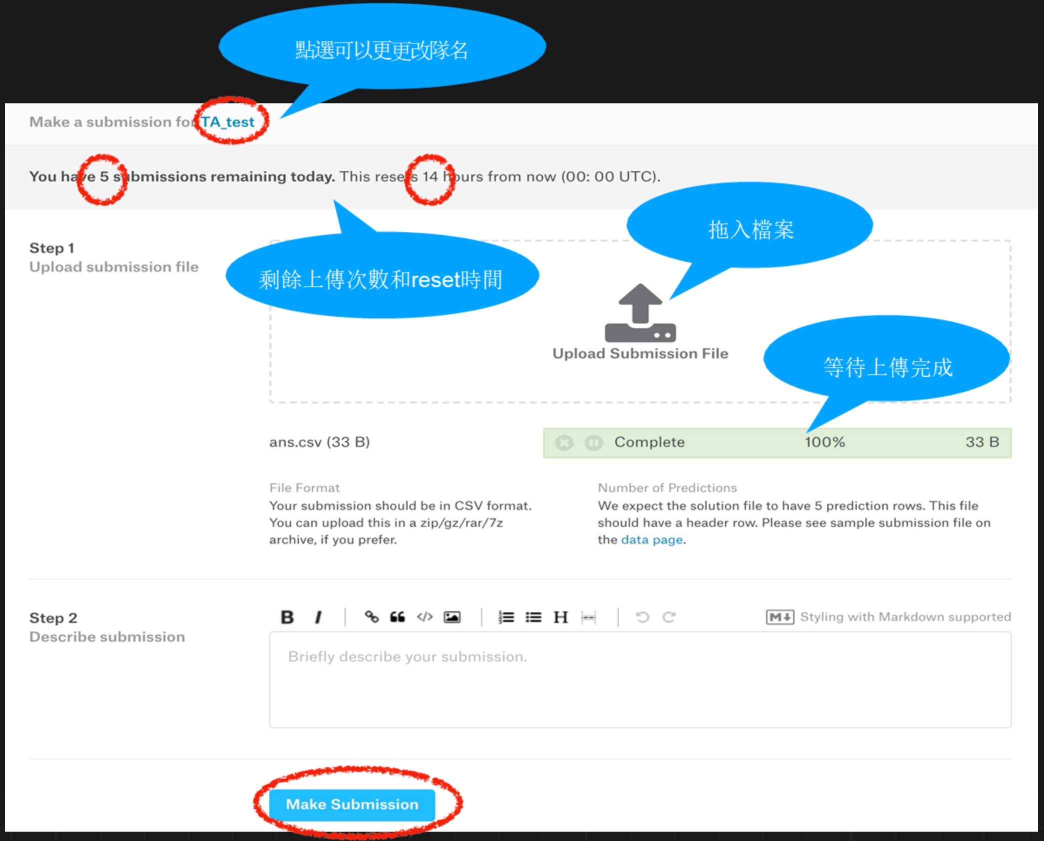
上傳分類結果進行評估 (1/3)

1. 成功參加比賽後，頁頁面面右上角角即會變成Submit Predictions
2. 上傳請注意格式(可參考Data中的sample_upload.csv)，一個人有20次機會



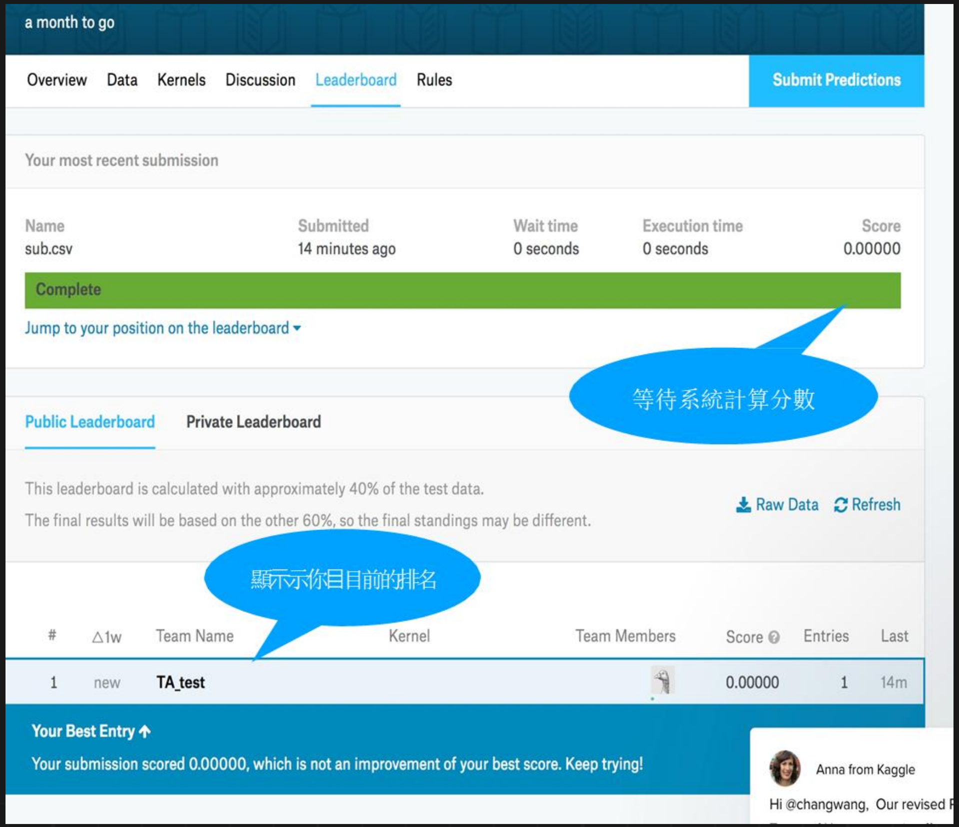
上傳分類結果進行評估 (2/3)

3. 點選右上角角Submit Predictions， 就進入上傳頁面
4. 若上傳等很久，可以重開頁面再上傳，未上傳成功不會計入上傳次數



上傳分類結果進行評估 (3/3)

5. 有時候系統負荷過大大，可能評分很久， 可以嘗試重開頁面，看看是否有評分成功，即有顯示出新的分數。若失敗的話不會有紀錄，也就不會用到上傳次數，重新再上傳即可。



FAKE NEWS

Thank You！
& To be continue...

NTU ANTSlab 陳廷易Leo

FAKE NEWS

Reference

★ A novel text mining approach based on TF-IDF and Support Vector Machine for news classification https://ieeexplore.ieee.org/abstract/document/7569223

★ TEXT CLASSIFICATION USING NAÏVE BAYES, VSM AND POS TAGGER https://pdfs.semanticscholar.org/43d0/0d394ff76c0a5426c37fe072038ac7ec7627.pdf

★ Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008 http://www.im.ntu.edu.tw/~paton/courses.htm

★ Text categorization with Support Vector Machines: Learning with many relevant features https://link.springer.com/content/pdf/10.1007%2FBFb0026683.pdf

★ Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles: http://snap.stanford.edu/mis2/files/MIS2_paper_2.pdf

Q & A

GitHub: tychen5

LinkedIn: tychen5

slido: 0913