Data Mining Final Project Report - Predicting Usefulness of Yelp Reviews
組別：G07

一、Members :

1. 學號：R07946015　系級：資料科學學位學程碩一　姓名: 許睿修
2. 學號：R07946012　系級：資料科學學位學程碩一　姓名: 林啟祐
3. 學號：R06946021　系級：資料科學學位學程碩一　姓名: 譚雋飛

二、Introduction & Motivation :

店家評論對於品質的改善抑或是使用者對於店家的第一印象相當重要，但過多的評論實在讓人無法在有限的時間內一一看過，也不容易去抓取真正關鍵有用的資訊，比如說一間餐廳想要知道它缺點在哪並作改善，或者是哪些優點值得它繼續發展，它可以藉由一些 ”Useful” 的評論給予回饋並改善，因此我們想知道某則評論他是或不是 “Useful”，藉此讓使用者能夠節省時間只要去關注相對有用的資訊就好。我們認為，一個有用的評論會累積相當多的投票數並且新鮮度也是相當重要的參考指標，所謂投票數有點類似臉書的按讚，當使用者認為此篇評論 “Useful” 便有機會給他按下一票，我們的目標在於找出 “Useful” 的評論在它經過時間被廣大的社群投票為好的評論之前，藉由預測他最終可能擁有的投票數。

三、Data Preprocessing / Feature Engineering :

1. Data Description

資料來源是 Kaggle 的 Yelp Recruiting Competition [1]，包含使用者(User)、店家(Business)、使用者對店家的評論(Review)等。其中訓練集的評論紀錄日期是到 2013-01-19，測試集的評論紀錄日期則到 2013-03-12。兩個資料集的詳細資料請參考以下的表格，下個段落將介紹我們從中挑選出的 features。

	User	Anonymous User	Business	Review
numbers	48978	2318	12742	252863

註：由於有些使用者並不想公開個人資訊，Yelp 為了保障個人隱私，這些匿名使用者(Anonymous User)的資料會有部分缺失，資料前處理的過程我們將缺失值填為所有使用者資料的中位數。

	User	Business	Review
data	1. 使用者ID(user_id) 2. 姓名(name)	1. 店家ID(business_id) 2. 店名(name) 3. 地址(full_address)	1. 評論ID(review_id) 2. 使用者ID(user_id) 3. 店家ID(business_id)

	3. 總共評論次數(review_count) 4. 評論的平均評等(average_stars) 5. 其他使用者對自己所有評論的總投票數(votes:funny, useful, cool)	4. 城市(city) 5. 州(state) 6. 緯度(latitude) 7. 經度(longitude) 8. 平均評等(stars) 9. 總評論數(review_count) 10. 類別(categories) 11. 是否有營業(open)	4. 評等(stars) 5. 內文(text) 6. 評論日期(date) 7. 其他使用者對這則評論的投票(votes:funny, useful, cool)

2. User Feature

總共選出五個 features，分別是 average_stars、review_count、votes_funny、votes_useful 和 votes_cool。

3. Business Feature

總共選出三個 features，分別是 stars、review_count 和 open。

4. Review Feature

總共選出兩個 features，第一個叫做 days_from_written_to_recorded，是計算該則評論從發表一直到被紀錄在訓練集，中間經過了多少天；第二個叫做 text_length，是計算內文總共的英文字母數量。另外，雖然訓練集的評論有其他使用者對該評論的投票資訊，例如 votes_funny 或 votes_cool，但是在測試集並沒有提供相關資訊，因此沒有列入訓練用的 features。

Index	votes_useful
review_days_from_written_to_recorded	0.23235
review_text_length	0.305991
business_stars	0.0037168
business_review_count	0.0348817
business_open	-0.0378939
user_average_stars	-0.008521…
user_review_count	0.312815
user_votes_funny	0.487287
user_votes_useful	0.476471
user_votes_cool	0.479016

圖一、十個 features 與 評論 votes_useful (ground truth) 的相關係數

5. Review Text Mining

關於評論的內文(text)，我們進一步用了以下幾種 Text Mining 方法，試著找出對預測評論實用度有幫助的 features。為了衡量這些找到的 feature 是否重要，我們選用上面段落提到的十個 features (這邊稱作 original features)，分別加上新的 feature，以線性回歸模型做預測，觀察 Kaggle Score 是否有變更好 (Evaluation 採用 RMSLE 所以愈低愈好)。

a. Sentiment Analysis

情感分析是用來判斷評論的內文屬於正面還是負面，使用 nltk.sentiment.vader.SentimentIntensityAnalyzer [2] 套件中的 polarity_score，對每篇評論的內文會用一個分數大小表示情感強度，如果分數為正的代表正評價，反之分數為負的代表負評價。

原本資料集中提供評論的資料，有包含使用者給店家的評等(stars)，與 Sentiment Analysis 的目的其實非常相似，因此放入同一張表格做比較。觀察下表可以發現，線性回歸模型加入 polarity_score 或 stars 後，預測的表現反而都變得更差了。

變差的原因可能是正負評和評論是否實用的關係並不大，例如說有些評論雖然是負評，但帶給其他使用者正確資訊以判斷是否要到這間店消費；而有些正評能提供這間店必買物品等資訊，在這兩種情況下的評論都可能得到很多實用票。

Kaggle Score	original features	polarity_score	stars
Private	**0.53554**	0.53647	0.53844
Public	**0.53783**	0.54003	0.54102

b. Readability

根據同樣分析 Yelp 資料集的文獻 [3]，他們發現評論內文的可讀性(Readability)愈高，會讓使用者感覺到評論比較實用。於是我們參考文獻中有用到的四種可讀性指標，使用 textstat 套件計算評論可讀性，並以我們的資料集去做測試。

Formula
$\mathrm{ARI} = 4.71 \times (\frac{\text{Characters}}{\text{words}}) + 0.5 \times (\frac{\text{words}}{\text{Sentences}}) - 21.43$
$\mathrm{CLI} = 5.89 \times (\frac{\text{Characters}}{\text{words}}) - 0.3 \times (\frac{\text{Sentences}}{\text{words}}) - 15.8$
$\mathrm{FRE} = 206.835 - 1.015 \times (\frac{\text{total words}}{\text{total sentences}}) - 84.6 \times (\frac{\text{total syllables}}{\text{total words}})$
$\mathrm{FOG} = 0.4 \times ((\frac{\text{Words}}{\text{Sentences}}) + 100 \times (\frac{N(\text{complex words})}{N(\text{words})}))$

圖二、四種 Readability index：ARI、CLI、FRE、FOG

測試的結果見下表，我們發現線性回歸模型加入 Readability 後，預測結果在 Public 有些改善，但在 Private 的表現變差了，整體來說表現的改善並不顯著。

Kaggle Score	original features	Readability
Private	**0.53554**	0.53561
Public	0.53783	**0.53764**

c. TF-IDF (term frequency–inverse document frequency)

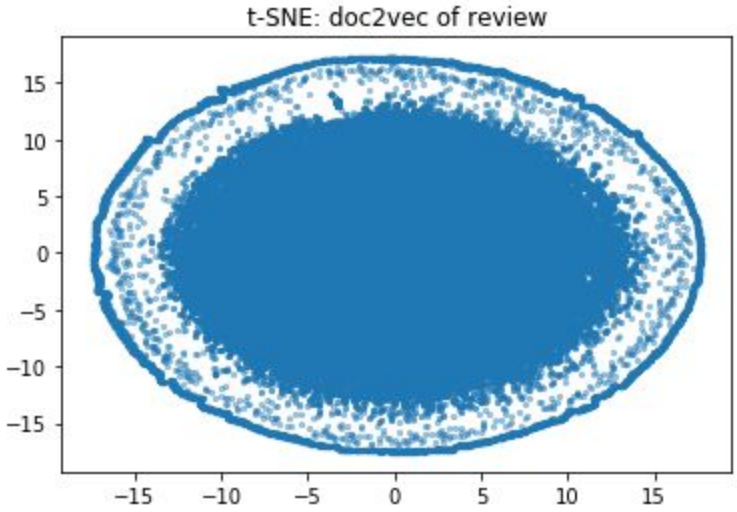
TF-IDF 演算法包含了兩個部分：詞頻(term frequency，TF)跟逆向文件頻率(inverse document frequency，IDF)。詞頻指的是某一個給定的詞語在該文件中出現的頻率。而逆向文件頻率則是用來處理常用字的問題，詞彙是否在多篇文章中多次出現。我們使用 sklearn.feature_extraction.text 套件，轉換成 12613 維的向量表示每個詞在該內文是否出現和其重要性。

我們將線性迴歸模型加入 TF-IDF 的向量後，預測結果仍舊沒比較好。原因可能是維度太高造成的影響，又或者是使用哪些字並不會對於評論造成太大的影響，反而文意可能才是主要的因素。

Kaggle Score	original features	TF-IDF
Private	**0.53554**	0.55926
Public	**0.53783**	0.56330

d. Doc2Vec [4]

doc2vec是考慮評論內文的語意後，將文件轉變成向量表示的方法，整體語意相近的文件會有較接近的向量。先把評論的內文 tokenize 後去掉常用詞，接著我們使用 Gensim.models.doc2vec 套件，經過 100 epoch 的訓練，將資料集內約 25 萬個評論以 128 維的向量表示。透過 sklearn.manifold.TSNE 套件降低至 2 維，我們可以將這些文件的向量視覺化地呈現如下，可以觀察到大部分的向量都集中在圖片中心，而外圍有兩個明顯的圓圈。



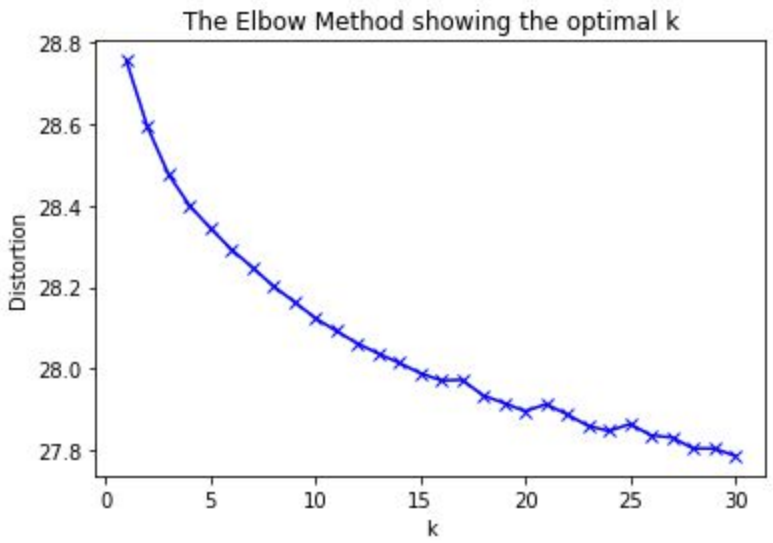
我們將線性回歸模型加入 128 維的 doc2vec 向量後，預測的結果仍舊沒比較好。原因可能是線性回歸模型過於簡單，無法從 doc2vec 中獲得有幫助的資訊，又或者 doc2vec 沒有辦法正確描述評論內文的語意。

Kaggle Score	original features	Doc2Vec
Private	**0.53554**	0.54175
Public	**0.53783**	0.54442

e. K means

假設 doc2vec 有正確地表示評論內文的語意，只是因為線性回歸模型沒辦法利用這項資訊，我們可以進一步對 doc2vec 做 clustering 的前處理。理論上如果有辦法透過分群，找出某一群的評論內文對於預測實用性有幫助，我們可以回去檢查那一群評論的語意特色，進一步分析語意與實用性上的關聯。

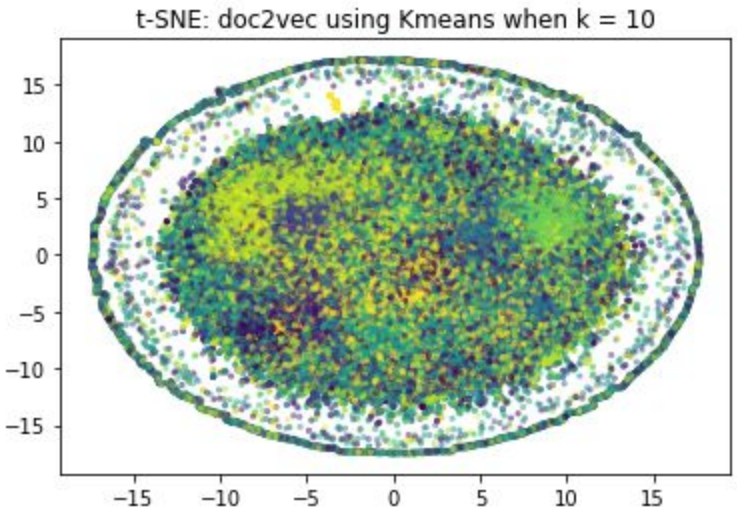
K means 是 clustering 的一個經典方法，我們使用 sklearn.cluster.Kmeans 套件，首先透過 elbow method 找出適合分為幾群，以下圖來說，曲線斜率的轉折點在 k = 15 處，也就是適合分做 15 群。



除了使用 elbow method 找到的最佳 k 值，我們也分別測試了 k = 5, 10, 20 作為對照組。實作上每個評論會屬於某個群，利用 one-hot-encoding 的方式作為 features加入線性回歸模型，結果請參考下表。我們發現在 k = 15 時並沒有最好的表現，反而是在 k = 10 時預測表現在 Private 和 Public 都有所改善。

Kaggle Score	original features	Kmeans_5	Kmeans_10	Kmeans_15	Kmeans_20
Private	0.53554	0.53633	**0.53536**	0.53586	0.53600
Public	0.53783	0.53865	**0.53753**	0.53755	0.53798

下圖為 k = 10 時使用 tSNE 視覺化 doc2vec 的效果，然而各群之間的區分並不明顯。雖然 k = 10 的表現有所提升，不過提升的幅度僅在小數點下第四位，以結果來說並不顯著，因此我們沒辦法有力地支持說評論的語意和實用性之間存在著關聯性。



四、Methodology and Evaluation:

Method \ Kaggle Score	Private	Public	Ranking	Note
Linear Regression (LR)	0.50826	0.50889	94 / 350	使用 selected features 。
Deep Neural Network (DNN)	0.48545	0.48405	62 / 350	使用 selected features 。 共三層，unit = 64, 32, 1, 每層之間有 batch normalization，最後一層有用 ReLU
Support Vector Regression (SVR)	0.50575	0.50356	87 / 350	使用 original features。 Gamma = 0.0001, Epsilon = 0, C = 10
Random Forest Regressor (RFR)	0.52554	0.52579	120 / 350	使用 selected features 。 n_estimators=5, max_depth=5, min_samples_split=3
Ensemble	0.48340	0.48241	59 / 350	DNN + SVR + RFR

註：selected features
定義為 ['review_days_from_written_to_recorded', 'review_text_length', 'business_review_count', 'user_review_count', 'user_votes_funny', 'user_votes_useful', 'user_votes_cool']， 這些 features 的 -1 次方和 2 次 polynomial 項。

五、Conclusion

本次專題的目的是預測一篇評論的實用程度，並以其他使用者對該則評論的投 useful 票數為 ground truth。我們的主要貢獻在於分析資料，從資料中找出對於預測有幫助的重要 features，並以線性回歸模型做實驗和比較結果。

實驗結果顯示使用者(User)過去評論的次數(review_count)，以及得到其他使用者的投票數(votes: funny, useful, cool)，對於預測相當有幫助。原因不難想像，評論數較多的使用者通常較為資深，他們懂得如何發表有實用價值的評論，而得到愈多投票數的使用者，往往也代表他們的意見容易受人認同。

另一方面，店家(Business)的資訊則顯得不那麼重要，大部分店家的 features 與 votes_useful 的相關係數都很低，從模型中拿掉這些 features 也會有較好的表現，代表評論的實用程度與店家的資訊可能並不那麼相關。

關於評論(Review)本身，最有用的兩個 features 是評論的總字數和評論發表後經過的時間。總字數愈長往往代表評論資訊量愈大，有較高的機會包含實用的資訊；評論發表後經過的時間愈長，則有更多機會被其他使用者注意到，進而決定要不要投票給這則評論。至於評論的內文(Text)，我們嘗試了幾種 Text mining 的方法，例如情感分析、可讀性、字的重要性(tf-idf)和文意(doc2vec)等，但是都無法幫助模型在預測結果上有顯著地改善。

總結來說，雖然 Yelp 當初舉辦這個競賽，其中一個目的是想知道評論內文的新穎度是否會是產生實用性的原因，然而我們的研究成果卻顯示無法從內文得到有幫助預測實用性的資訊，僅能知道與哪位使用者較為相關，這應該也算是個有趣的發現吧。

六、Source Code

請至 Github 下載本次報告使用的程式碼：
https://github.com/JuiHsiu/Yelp-Recruiting-Competition

七、Reference

[1] Yelp Recruiting Competition (https://www.kaggle.com/c/yelp-recruiting)

[2] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[3] Zhiwei Liu , Sangwon Park (2015) "What makes a useful online review? Implication for travel product websites"

[4] Quoc Le, Tomas Mikolov (2014) "Distributed Representations of Sentences and Documents"