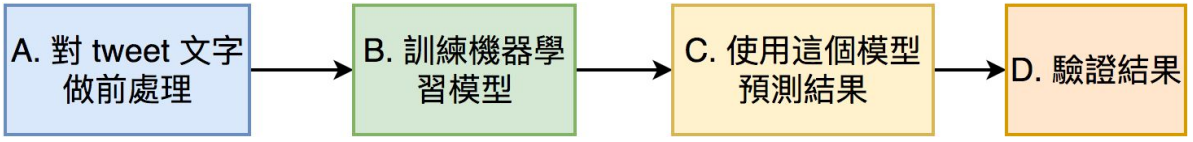
NLP Project 1

1. Name and ID & 2. Division of Work

Name	ID	Division of Work
鄭淵仁	b04902053	**Report:** Methods, Evaluation **Code:** preprocess on tweets, XGBoost, Gradient Boosting Regressor
戴嘉男	a06922301	**Report:** word2vec, GloVe, NTUSD-Fin discussion
王曄庭	b04902054	**Report:** Discussion：machine learning、ensemble
周景軒	b03902087	**Code:** AdaBoost Regressor, Bagging Regressor, Random Forest, Support Vector Regression F1 score

3. Methods

我們主要使用以下流程來實作這一題：



在這些流程中，我們特別在「**對 tweet 文字做前處理**」的部分做了很多不同的方法。接下來我們將分別敘述這些流程。

A. 對 tweet 文字做前處理

首先，我們在對文字做前處理的部分想了以下三種方法：

I. 使用 NTUSD-Fin [2] 提供的 market_sentiment 資料來轉換文字

因為我們認為 snippet 中的文字的 market_sentiment 能大概表現這則 tweet 的 sentiment，所以我們使用這項資料來轉換文字。詳細轉換文字的方法如下：

對每一則 tweet 中的 snippet 都做這個操作：對於 snippet 中的每一個字，我們都用 NTUSD-Fin 轉成 market_sentiment，再取這些 market_sentiment 的平均值、最大值、最小值，最後把這三個數字做為代表這個 snippet 的 vector。

II. 使用 word embedding 資料來轉換文字

我們認為雖然 market_sentiment 能大概表現這個詞的 sentiment，但是只用一個分數可能無法完全表現一個詞在不同面向的 sentiment，那麼直接取平均、最大值、最小值之後，就有可能

會把不同面向的 sentiment 混淆在一起。所以我們改使用 word embedding 來轉換文字。詳細轉換文字的方法如下：

對每一則 tweet 中的 snippet 都做這個操作：對 snippet 中的每一個字，我們都用 pretrain word embedding 轉成 vector，再取這些 vector 中的各個維度的平均值、最大值、最小值，最後把這三組 vector concatenate 起來，做為代表這個 snippet 的 vector。

這個方法是我們參考 ECNU at SemEval-2017 Task 5: An Ensemble of Regression Algorithms [5] 時想到的。

在這項實驗中，我們使用了 **NTUSD-Fin**、**GLOVE** [3]、**Google word2vec** [4] 共三種 word embedding 來轉換文字。

III. 使用 word embedding 資料及 target 的 one-hot encoding 來轉換文字

因為我們認為除了 tweet 內文對 sentiment 有影響力以外，不同的 target 也可能對 sentiment 有影響，例如 "$APPL" 的 sentiment 可能就會偏高。所以我們使用了以下方法來把 target 資料也放入 vector 中：

這個方法是 II 的延伸，我們除了對每一則 tweet 中的 snippet 都轉成三組 vector 以外，我們還把每一則 tweet 中的 target （如 "$APPL"）做 one-hot encoding，再與方法 II 中的三組 vector concatenate 起來。

另外，由於 target 數量龐大，所以我們也設定一個 threshold ，讓出現小於 threshold 次數的 target 都分類到新的 target："$OTHER"。在後面我們會做實驗驗證 threshold 數量的多寡的好處。

B. 訓練機器學習模型

把每一則 tweet 都轉換為 vector 之後，我們把這些 vector 及每一則 tweet 的 sentiment 做為 training data，使用不同的機器學習模型去 fit。

我們一共使用了 **XGBoost Regressor** (XGB)、**Gradient Boosting Regressor** (GBR)、**AdaBoost Regressor** (ABR)、**Bagging Regressor** (BR)、**Random Forest** (RF)、**Support Vector Regression** (SVR) 來 fit training data。其中，在使用這些演算法時，我們都先用 cross validation 求得最好的參數後，再實際使用這些參數去 fit training data。

另外，我們也使用 ensemble 技術來結合上述這些模型的結果。

C. 使用這個模型預測結果

使用不同的機器學習模型去 fit 完資料後，我們就使用這些機器學習模型去 predict 結果。另外，在做分類 Bullish/Bearish/Neutral 的問題時，我們使用 cross validation 來確立 2 個 threshold，用以把分數分類成上述三類。

D. 驗證結果

我們使用助教提供的 Mean Squared Error 及 Micro-average F1、Macro-average F1 來驗證最後的結果。

另外，因為題目提供的 training set 的 label 只有 sentiment 分數，而沒有 Bullish/Bearish/Neutral 的分類，所以我們要手動設定 threshold 來把原題目的 label 分類成上述三類。在實際觀察 training set 的 sentiment 的分布後，我們認為 0.3 是合理 threshold。**接下來的分類問題，我們都會把 sentiment score 使用 0.3、-0.3 來分成這三類，作為正確的分類答案**。

4. Evaluation

由於我們在文字處理及使用機器學習模型時，都提出各種不同的方法，因此我們在這一大節會針對這兩項主題分別比較結果。

A. Baseline Method

前處理的方法	MSE	Micro-average F1	Macro-average F1
在 [-1, 1] 之間亂數產生數字	0.5209	**0.3422**	**0.3181**
都輸出 0	**0.1559**	0.2997	0.1537

B. 不同的文字前處理

在此表格中，我們統一使用 XGBoost Regressor，並且加入 target 資料來比較。

前處理的方法	MSE	Micro-average F1	Macro-average F1
使用文字的 market_sentiment	0.1060	0.5631	0.4877
word embedding (NTUSD-Fin)	0.0918	0.5868	0.5110
word embedding (GLOVE)	0.0957	0.5631	0.5199
word embedding (Google W2V)	**0.0653**	**0.6388**	**0.6305**

C. 是否使用 target 的 one-hot encoding 以及 threshold

在此表格中，我們統一使用 XGBoost Regressor 及 Google word2vec 來比較。

threshold	MSE	Micro-average F1	Macro-average F1
沒有加 target 資料	0.0681	0.6467	0.6250
threshold=0	**0.0653**	0.6389	**0.6305**
threshold=1	0.0676	0.6388	0.6183
threshold=2	0.0666	**0.6577**	0.6239

D. 不同的機器學習模型

在此表格中，我們統一使用 word embedding (Google W2v) , tag (one-hot encoding) 來比較。

Method	機器學習模型	MSE	Micro-average F1	Macro-average F1
single	XGB	**0.0653**	0.6356	0.6152
	GBR	0.0705	**0.6498**	**0.6293**
	ABR	0.0854	0.6277	0.6014
	BR	0.0943	0.5836	0.5299
	RF	0.0950	0.5883	0.5317
	SVR	0.0785	0.6182	0.5766
Ensemble	XGB + GBR	**0.0646**	**0.6530**	**0.6330**

5. Discussion

A. GLOVE、google word2vec、NTUSD-fin的介绍、区别与讨论

NTUSD-Fin中，为了将自然语言中的字词转换为计算机可以理解的数学符号，利用了多种记分方法：requency, CFIDF, chi-squared value, market sentiment score and word vector. 在这份词典中，被标记的字词起码要出现10次，并且，经过chi-squared测试后预期和被观察到的频率有明显不同的字词才会存在于字典中。Market sentiment的得分是由bearish PMI与bullish PMI相减得来。本次作业中将每个tweeter中的snippet提取出来并附上market_sentiment的数值，最后计算出整合句子所有重点词出的最大值，最小值，以及平均值作为这个snippet的向量。同时NTUSD-Fin中也为每个句子中的snippet生成了维度为300的word2vec向量，也被用来做本次作业的sentiment analysis。

Word2vec的作用就是将自然语言中的字词转为稠密向量。word2vec的理论受到了One-Hot Encoder影响的，One-Hot Encoder就是把字词转为多维向量，向量中只有一个值为1，其余都为0。不过，为了避免描述相同事物的字词过多导致的维度灾难，Word2Vec就被用来将One-Hot Encoder所产生的高维度向量转化为低维度的连续值，也就是稠密向量。word2vec主要分为CBOW（Continuous Bag of Words）和Skip-Gram两种模式。CBOW是从原始语句推测目标字词；而Skip-Gram是从目标字词推测出原始语句。Word2vec的训练数据使算法可以为每一个字词找到最合适的向量，这两个方法都用了分类算法来推测字词的情绪。word2vec可以很好地将不相关的字词分开，将相似的字词归类。在这份作业中，主要用了Skip-Gram的方法，将snippet转为维度为300的向量，进而用计算出snippet中所有字词的最大值，最小值和平均值，生成维度为900的最终向量，最后用机械学习的方法将向量转化为分值，从而判断整个句子的情绪。

GLOVE方法是进行词的向量化表示，使得向量之间尽可能多地蕴含语义和语法的信息。Glove融合了Globall Matrix Factorization和Local Context Window。一个是基于奇异值分解（singular value decomposition）的LSA（Latent semantic analysis）算法，该方法对term-document矩阵进行奇异值分解，从而得到term的向量表示和document的向量表示。GLOVE的目的是在语义和语句上都获得更好的表达效果。这种方法相对于word2vec的优势就是利用了词共现的信息，也就是不仅仅只关注word2vec窗口大小的上下文，而是用到了全局信息。GloVe综合了LSA、CBOW的优点，训练更快、对于大规模语料算法的扩展性也很好、在小语料或者小向量上性能表现也很好。在这个作业中，GLOVE只转换了snippet，与这次作业中word2vec的方法相似，先将snippet转换为维度是300的向量，进而用计算出整个句子中的最大值，最小值和平均值，生成一个整个句子的向量，维度为900，最后用机械学习的方法将向量转化为分值，来判断整个句子的情绪。

理论上来讲，其实word2vec，NTUSD-Fin与GLOVE的效果应该相差不多，并没有孰好孰坏之分。甚至GLOVE的性能更好，可以通过更短的训练时间来提高准确率。但是在本次作业中，利用了控制变量的方法发现，word2vec的效果要比GLOVE和NTUSD-Fin要好，分析与推测出的原因有以下几点：第一点，因为直接下载了Google提供的训练数据，这个训练数据的量十分庞大，是导致word2vec准确率高的一个原因。第二点，GLOVE的优点是它利用到了全局信息，而在这次作业中，snippet已经被标识，所以并没有发挥出GLOVE的优势。第三点，word2vec是Predictive的模型，GloVe是Count-based，2014年，Baroni 等人表明 predictive model 几乎在所有的任务中都优于 count-based model[1]。第四点，本次作业中求出整个snippet情绪分数的最大值，最小值和平均值计，增加了word2vec的准确度，很好地弥补了word2vec多义词处理乏力的缺点。第五点，NTUSD-Fin中，由于直接使用字典中的market_sentiment和word2vec，同时，字典的数据量太小，没有很好地处理一词多义，上下文含义和网络用语与正常语言区别的情况。

B. 加入 target 之後作為 training data 後的影響

我們從 Evaluation D 中發現：加入 target 資料後，無論 threshold 設為多少，MSE 都比不加入 target 資料來的小。這證明了我們原先的猜測：「不同的 target 對 sentiment 有影響」。

另外，我們也發現 threshold 設為 0 的 MSE 是最小的。我們認為這是因為大部分 target 只出現過 1、2 次，所以如果設了 threshold 就會把大量的 target 分到 "$OTHER"，反而使得 "$OTHER" 的 label 同時包含了 sentiment 高與低的資料，結果讓機器學習模型無法很好的 fit 到原先的資料。

C. 在建立model的過程中，我們採用不同的機器學習法，會對結果產生怎樣的影響？

在我們测試過的機器學習法中，大致可以分為兩類：bagging以及boosting。就我們測試出來的結果而言，boosting的效果比bagging來得好，因此我們在這裡只著重討論Gradient Boost、XGboost兩種boosting技術。

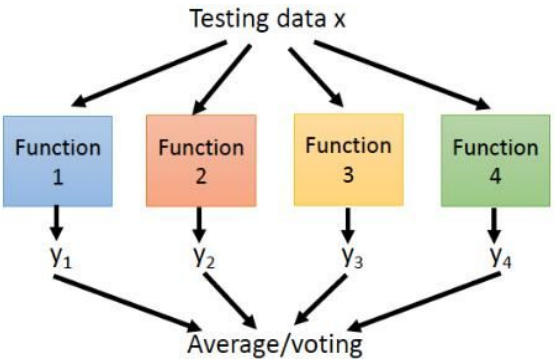
Gradient Boost 是 Boosting 其中一種學習演算法，而XGboost，則是Gradient Boost的一種優化版本，並新增/改良了以下幾種特性：

1. Regularization - 利用Regularization控制model的複雜度，減少overfitting的發生。
2. lossing function - lossing function 增加了二階導數的計算，提升計算的精密度。
3. 對於"尋找最佳分割點的衡量"進行改良，提升分割的速度、效果。
4. XGboost還有一些對於計算效能的改良，如稀疏感知算法、並行化算法等，在此不贅述。

就上述各點，我們可以得知，若是採用XGboost，在各方面來說都會得到較好的結果。

D. 在Evaluation中，為何使用了ensemble的技巧後，performance會比原本還要來的好？

我們所採用的方法，類似於bagging ensemble，也就是簡單的model，其複雜度較低，error較大；而複雜的model，其複雜度較高，error較小。因此若是將兩者(或更多)的model輸出結果作加權平均，得到的結果不但能保證其準確性及穩定性，也能降低結果的error，以及overfitting的發生。



6. Conclusion

在本次作业中，运用了 Gradient Boost、XGBoost 等等机器学习方法，同时用 word2vec、GLOVE 以及 NTUSD-Fin 语料库将 twitter 的情绪进行了计算，最後得到最小 **0.0646 的 MSE**。

我們得出以下结论，第一点，選用合適的機器學習法是非常重要的，採用不同的機器學習法，所產生出來的model，會對結果產生非常大的影響(如XGboost的表現會比gradient boost的表現來的優異許多)。第二点，在本次作业中，运用word2vec的方法比GLOVE以及NTUSD-Fin的准确率更高。尤其训练数量的多少对结果的影响很大。第三点，藉由ensemble不同的model，我們可以得到比單一model更加優秀的結果。第四點，把 target 做 one-hot encoding 可以小量提升精準度。

7. References

[1] http://www.aclweb.org/anthology/P14-1023

[2] Chung-Chi Chen, Hen-Hsen Huang and Hsin-Hsi Chen. 2018. NTUSD-Fin: A Market Sentiment Dictionary for Financial Social Media Data Applications. In Proceedings of the 1st Financial Narrative Processing Workshop, 7 May 2018, Miyazaki, Japan.

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. [pdf] [bib]

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

[5] http://nlp.arizona.edu/SemEval-2017/pdf/SemEval152.pdf