NLP HW3 Twitter Sentiment Analysis
Sofia Guo

Intro

I performed twitter sentiment classification prediction using three classification machine learning models, namely Naïve Bayes, Decision Tree and Random Forest, using a handy tweets dataset I found in Kaggle. I found that the Random Forest Classifier performed best among the three in predicting twitter sentiments. Furthermore, I utilized two methods, namely VaderSentiment Analyzer and TextBlob to re-tag the sentiment scores and polarities of the tweets, the results show that VaderSentiment performed better than TextBlob when compared to the original tagging

Data

The twitter dataset is obtained from the Kaggle website, with the name 'Tweers.csv', which can be found here:

https://www.kaggle.com/code/anuf13/twitter-sentiment-analysis/input

It has 27,481 unique tweets labelled according to their sentiment.

The columns are: 'textID'—a unique text Id; 'text' – the text of the tweet; 'seletced_text' – the selected text from tweet' 'sentiment'—the sentiment of the tweet (already pre-labeled for later machine learning models to train and test).

There isn't a specific topic regarding this tweeter dataset, they are mostly common daily tweets about people's lives and emotions, or we can maybe also call it the 'quotidian tweets dataset'.

(A note here is that, I did spend quite some time trying to figure out the 'tweepy' API, I checked the links of datasets in the Brightspace but the .txt files only contain tweet IDs instead of real tweet texts. We had to perform a further step known as the 'hydrating' of a tweet using its ID, meaning to retrieve a tweet's complete info. I tried to register for a twitter's developer account via this link: https://developer.twitter.com/

However, it seemed that the free version could not scrape much data from twitter and my access is constantly 403 forbidden, and to update my account to v2, I had to pay $100 which is too much…So in the end, I decided to get direct available twitter sentiment dataset from Kaggle and perform some analytics around it)

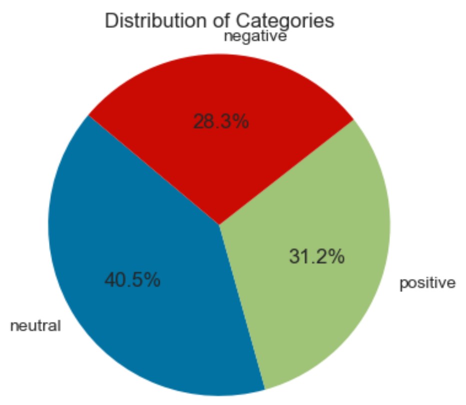
Data Preprocessing

For text preprocessing, I first removed the hyperlinks, hashtags, special characters, numbers, punctuations, omitted character whose length is less than 3 etc. Then, I performed stemming, which included tokenization, conversion of each character to lower case, removing stops words, and finally joining them together.

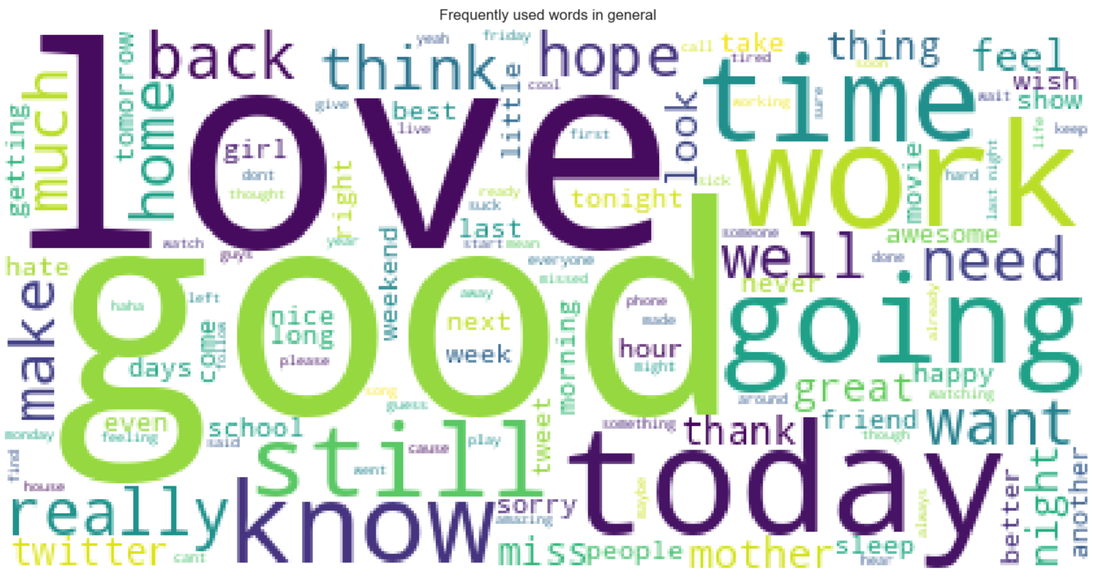
Afterwards, I transformed the cleaned texts into numbers via the 'Bag of Words' technique, and labeled each text as either 'negative' (0), 'neutral' (1), or 'positive' (2).

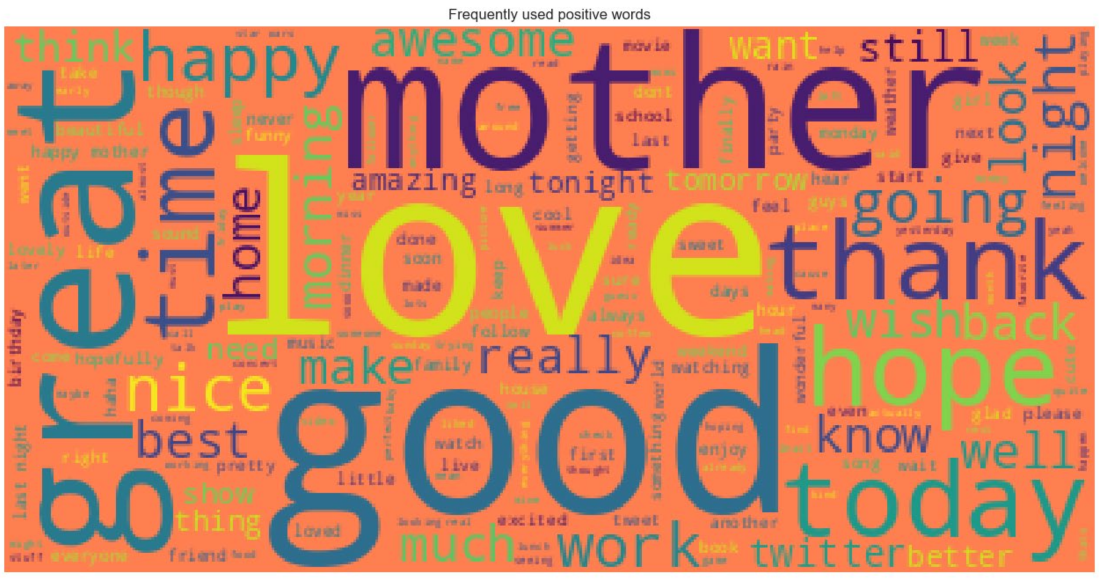
Some Interesting EDAs

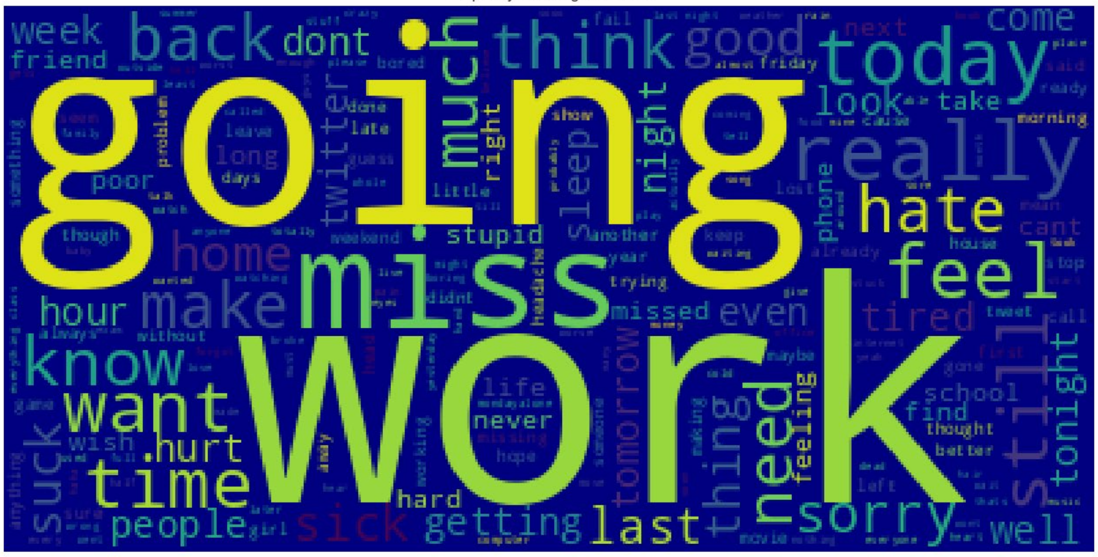
The tweets have already been classified as either 'positive', 'negative' or 'neutral' beforehand. I can plot their distributions in a pie chart, the neutral sentiment tweets occupy the largest portion with 40.5%, the positive sentiment tweets also occupy a fair amount of 31.2%, with the negative tweets occupying 28.3%:

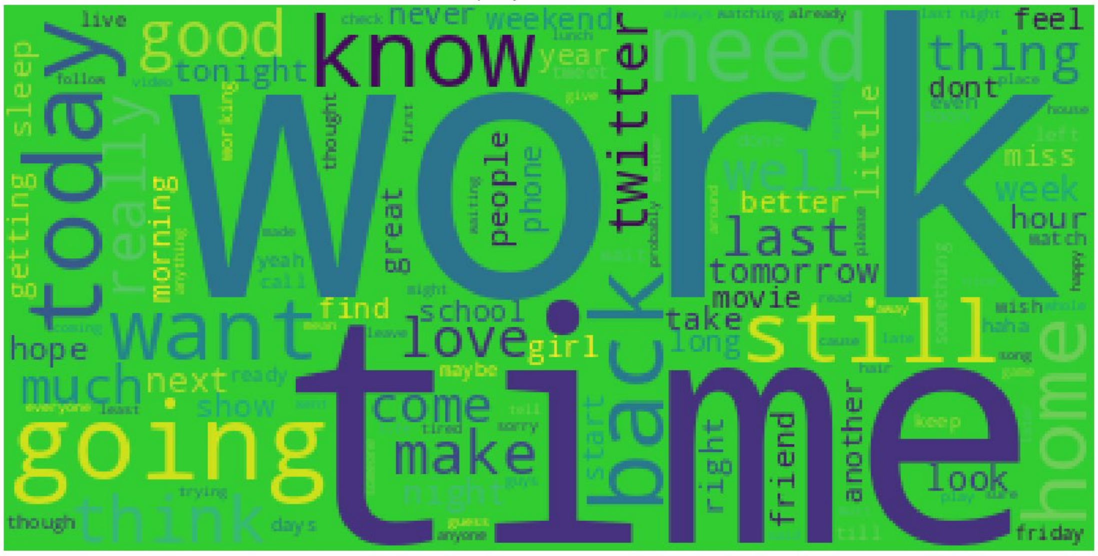


Also, we can plot word Clouds for the general dataset as a whole, for only the positive, negative, and neutral sentiment subsets respectively. We can see that in general, the overall sentiment seems to be positive, with words like ‘good’, ‘love’ occurring a lot of times. The most frequent negative words mainly include ‘going’, work’, ‘miss’, ‘hate’, etc.:






Frequently used negative words

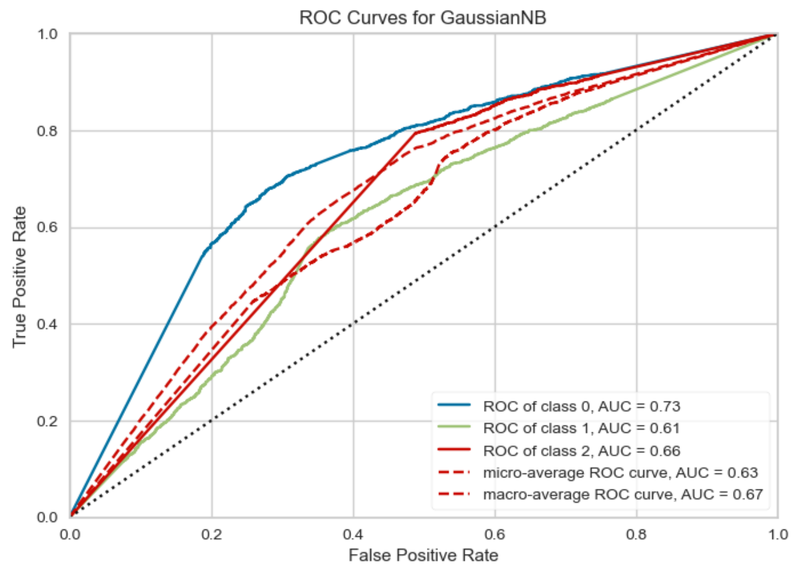

Frequently used neutral words

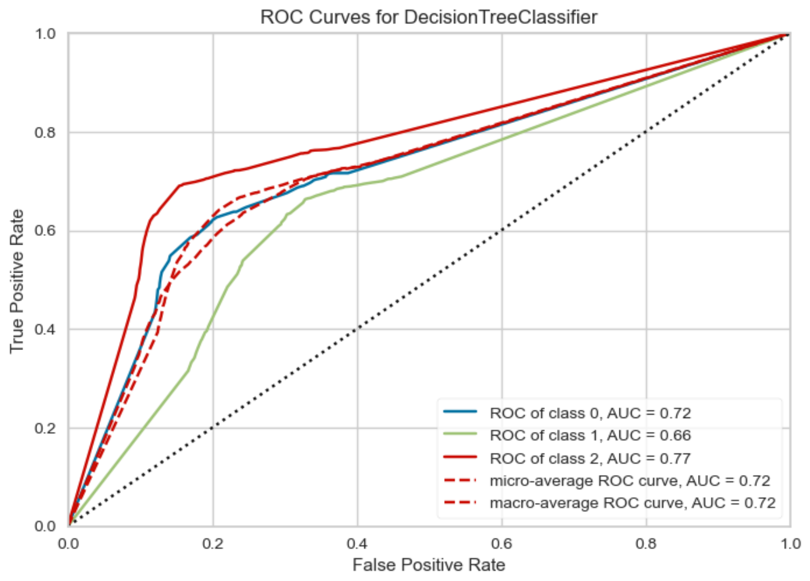
Predictive Modelling

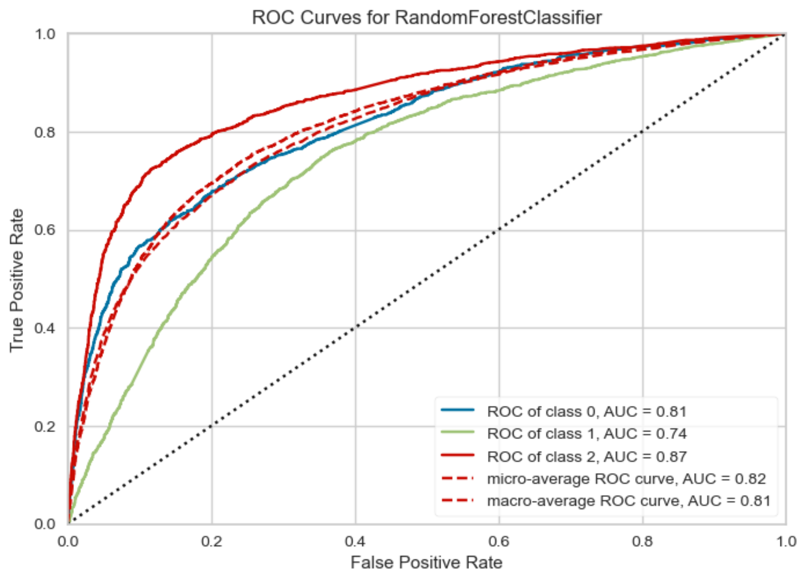
In this section, I built three classifiers – Naïve Bayes, Decision Tree and Random Forest, to test their abilities in successful sentiment classification. The train and validation sets are divided into an 8:2 ratio. The comparison results of these three models can be summarized as the following:

	Naïve Bayes	Decision Tree	Random Forest
Accuracy Score	0.457	0.612	0.663
Cross Validation Score	0.457	0.611	0.656

The ROC curves are shown down below:


ROC Curves for GaussianNB
True Positive Rate
False Positive Rate
ROC of class 0, AUC = 0.73
ROC of class 1, AUC = 0.61
ROC of class 2, AUC = 0.66
micro-average ROC curve, AUC = 0.63
macro-average ROC curve, AUC = 0.67

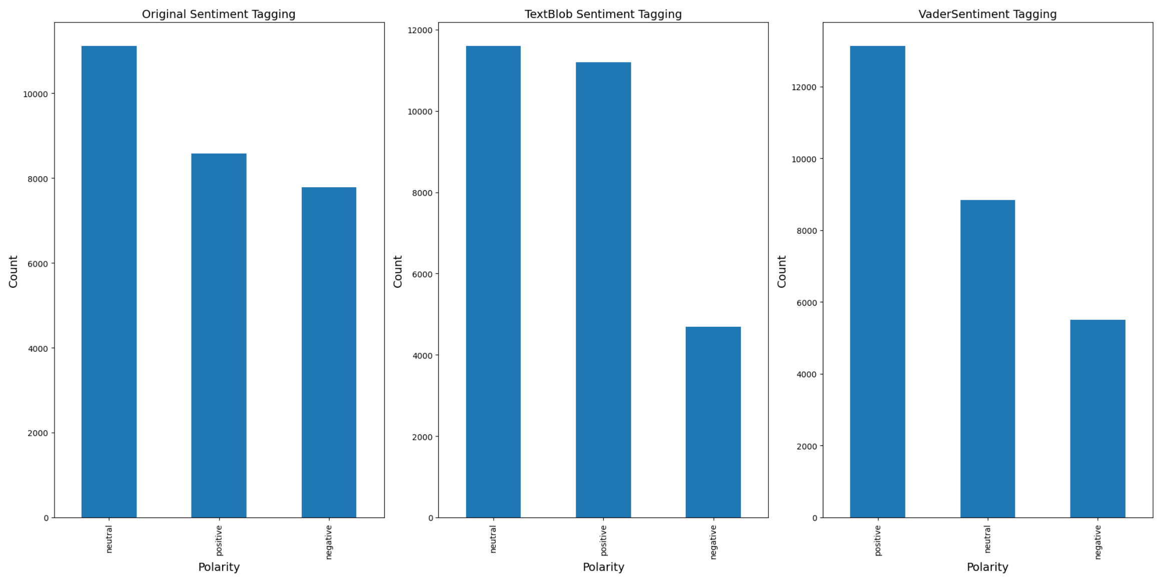

ROC Curves for DecisionTreeClassifier
True Positive Rate
False Positive Rate
ROC of class 0, AUC = 0.72
ROC of class 1, AUC = 0.66
ROC of class 2, AUC = 0.77
micro-average ROC curve, AUC = 0.72
macro-average ROC curve, AUC = 0.72


ROC Curves for RandomForestClassifier
True Positive Rate
False Positive Rate
ROC of class 0, AUC = 0.81
ROC of class 1, AUC = 0.74
ROC of class 2, AUC = 0.87
micro-average ROC curve, AUC = 0.82
macro-average ROC curve, AUC = 0.81

We can see from the comparison table that, among the three classifiers, the Random Forest Classifier performs the best, with the highest accuracy score of 0.663. It also has the largest area under the ROC curves (the AUC score). Therefore, the Random Forest Model performs best in this case, in terms of twitter sentiment classification.

Further Analysis: Comparison of sentiment tagging

In the original dataset, the sentiment of each tweet is already tagged. Assume that the tagging is correct, in this section, I wanted to further employ Textblob and VaderSentiment and re-tag the tweets again, to see that how they perform in determining the sentiment for the tweets in comparison to our original benchmark.



The results indicate that VaderSentiment performed better than TextBlob in determining the correct sentiment tags and polarity scores for our tweet dataset. The mis-tagged ratio for VaderSentiment is around 37.5%, which is lower than that of TextBlob of 41.2%.

Furthermore, it shows that both VaderSentiment and TextBlob tend to classify more tweets under the category tag of 'neutral', which lead to more source of error in comparison to the original tagging. Also, TextBlob tend to classify more tweets as 'positive' as opposed to 'negative', while VaderSentiment could achieve a better balance in distinguishing between good and bad. Therefore, the conclusion is that VaderSentiment serves as a better tagging method in determining sentiment and polarities for social media texts.