Exploring and Deciphering “The Perfect Tweet”

Abstract

For our project we analyzed the tweets from the top 200 accounts to explore and decipher what makes a good tweet that can generate a high engagement rate. We first research the top 200 accounts to extract the tweets from, following which several different levels of features are extracted, which is vital to explain the tweets, finally modelling the data with the respective engagement rates using different machine learning techniques to obtain the best accurate model. Using the most accurate model, the feature importance is derived to determine which feature is more important in predicting the engagement rate, thereby deciphering which feature is key when developing the content for a good tweet. Finally, we test the resulting model on tweets generated from an online tweet generator to predict the estimated engagement rate and see how the online tweet generator performs. This online tweet generator is intended to generate tweets based on historical data of past tweets of a specific account. Therefore, we will select a specific account and generate tweets based on learning from past tweets created by the said account owner. Our comparison will then show the effectiveness of the online tweet generator, as well as our model’s efficiencies in identifying the areas the tweets can improve on to have a better engagement result.

1. Introduction

1.1 Background

As consumers increasingly spend their attention on mobile devices, social media is a powerful tool for businesses, especially those with a global reach.

1.2 Motivation

In recent weeks, Twitter has gained a lot of public attention since Elon Musk became the largest shareholder in the company with a 9% stake and is on a quest to acquire the entire company with a bid of US$43 billion as of the time of writing. Given Elon Musk’s entrepreneurial track record, Twitter may become an even more powerful platform if his bid is successful. As such, businesses who understand how to engage Twitter users will be well positioned to benefit from the growth.

Twitter started off in 2006 as a social networking platform where users can post 140-character posts called "Tweets" that can be seen by other users. The character limits forced users to be more creative in their Tweets which arguably made Twitter more interesting. In 2017, Twitter increased the character limit to 280. The key engagement metrics for Tweets are “likes”, “retweets”, and “replies”.

1.3 Objectives

In this paper, we train our machine learning models on Tweets posted by the top Twitter accounts to find out what are the key factors that affect the engagement rates of a Tweet, and how do we generate the best possible Tweet.

2. Data Extraction

We first extract the top 200 most followed Twitter accounts, as listed by Viral Pitch (https://viralpitch.co /topinfluencers/twitter/top-200-twitter-influencers/), an influencer marketing research platform.

For each of these accounts, we extract all Tweets posted between 1 December 2021 to 18 March 2022, along with the number of “likes”, “retweets" and “replies” of each Tweet. With this, we have a total of 467,380 Tweets.

3. Data Feature Extraction

The feature extraction from the contents of the tweets can be split into 4 sections. Each section involves extracting a specific type of feature, that will be use later and tested during the modelling phase. The 4 sections are:

- **Account Level:** The features represent the account in which the tweets are generated from, which mainly represent the properties of the account owner.
- **Raw Features:** these are the raw features that are typically extracted from messages and content, explaining certain properties of a text.
- **Tweet Type Category:** Since each tweet is from a specific account that handles a specific type of genre, these genre categories are processed to create features.
- **Topics Category:** Even with the tweet genres, the tweets themselves may contain certain common topics amongst all the tweets.
- **TF-ID:** An additional feature extractor to vectorize the frequency of words across the text.

3.1 Account Level

The account level features are aimed to account for impact on the engagement rate of tweets due to the account itself, not the content tweet itself. The accuracy of the estimated feature importance of the other explainable features of the tweet content might be more accurate with the inclusion of these features. In addition, the modelling can confirm if the account level features itself has a high impact on the various engagement rate. The account level features are number of followers and average number of tweets created per day.

3.2 Raw Features

The raw features extracted from the tweets are typical properties of text-based data that help illustrate the text.

Table 1. The raw features and their descriptions

Raw Features		Description
Number of	words	Count of Words in Tweet
	Verbs	Count using Spacy Package
	Nouns	
	Punctuations	Count using nltk Package
	Numbers	Count of Exclusive Numbers
	Misspelled Words	Count validated by django
Sentiment Score		Scoring using nltk "SentimentIntensityAnalyzer"

Since the objective is to quantify what makes a good tweet, these properties can serve as indicators of what tweeters should look out for when crafting their tweets, depending on the results from the modelling phase and the feature importance of these properties.

3.3 Tweet Type Category

Tweets posted by top users are usually content from a certain category, as defined by the account owner. These categories can be defining factors, in which certain categories of content are more favored than others. Therefore, these categories are converted into variables using one hot-encoding.

Table 2. List of categories from generated tweets

News & Politics	Family	Finance & Education	
Entertainment	Food	Sports & Fitness	Health
Technology	Lifestyle	Fashion	Gaming

3.4 Topics Category

Aside from the categories in which the account owner usually tweets about, there could be other topics currently during the period of posting that could be trending, such that tweeting content about the trending topics might help to boost the tweets engagement. Thus, identifying these topics can quantify the "trending" topic variables.

To identify the topics, "Latent Dirichlet Allocation" topic modelling was used. This topic modelling is used to classify text in a post/article to fit a certain topic, where it builds a topic per document such that each word's "presence" in the document is attributable to a small number of topics in the document. Therefore, using the "genism" package to model an estimate the top three topics, the topic modelling was done through 2 iterations:

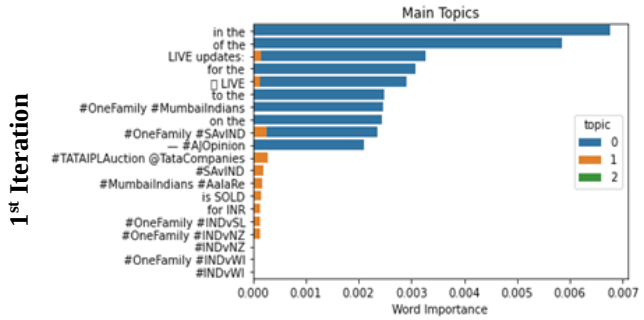


Figure 1. Word Importance from the top 3 topics from the 1st iteration run of LDA modelling.

From the first iteration, excluding the topics that contain main stop words, it is observed that these topic words carry the most weights:

Table 3. The top topics from the 1st iteration

OneFamily	#IND	LIVE	SAvIND

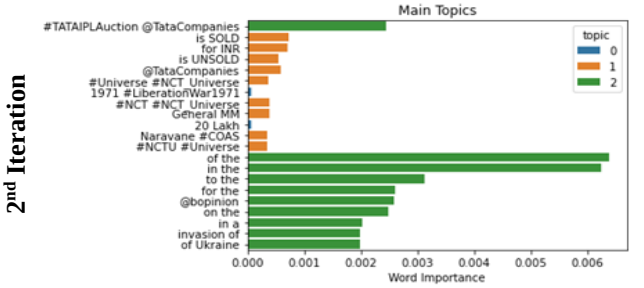


Figure 2. Word Importance from the top 3 topics from the 2nd iteration run of LDA modelling.

For the second iteration, the tweets that contain the topic words from the 1st iteration is removed before extracting the topics:

Table 4. The top topics from the 2nd iteration

SOLD	invasion	TATIPLAuction	UNSOLD	#NCT

Therefore, from the two iterations of topic modelling, the final topics that has the highest weights are:

Table 5. Overall top topics generated from LDA modelling

#OneFamily	LIVE	#SAvIND	SOLD
invasion	TATIPLAuction	UNSOLD	#NCT

One hot encoding is done to specify if a tweet contains a topic or not, creating 9 feature variables describing the topics found in the data.

3.5 TF-ID

Term frequency-inverse document frequency is a very common text vectorizer technique that transforms the text data into vector that is usable in modelling. Using this technique, additional features in the form of vectors of each tweet according to the term frequency and document frequency are created and will be later using in the modelling process as additional features to see if it improves the accuracy of the model to obtain a better estimate of the feature improvement coefficient.

4. Data Exploration

4.1 Multi-collinearity check

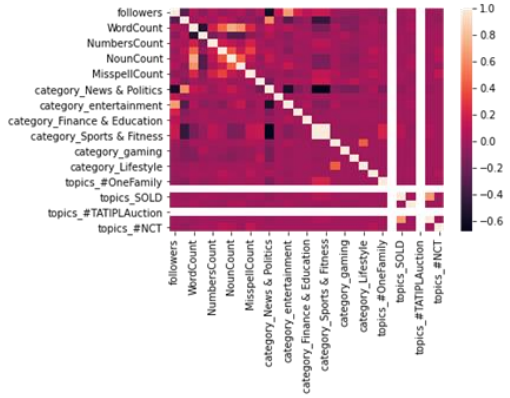
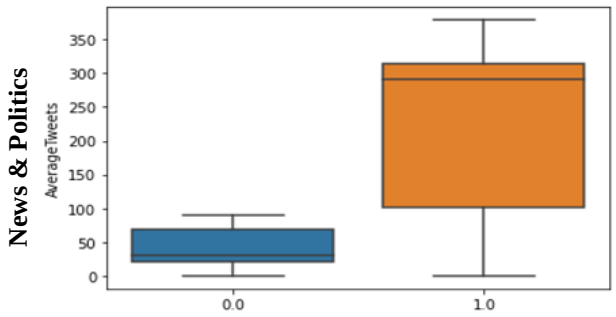


Figure 3. Correlation plot of all 29 features

With 29 features, aside from the TF-ID features, it is crucial to check for multi-collinearity:

- There are some effects between word count and different types of count, which is expected and ignored, since each explains specifics of a tweet.
- There is a high collinearity between AverageTweets, and 3 category tweets: News & Politics, Health, and Sports & Fitness. This could suggest that users from such groups usually have high average tweets per day.

4.2 Average Tweets vs Category (News & Politics, Health, and Sports & Fitness)



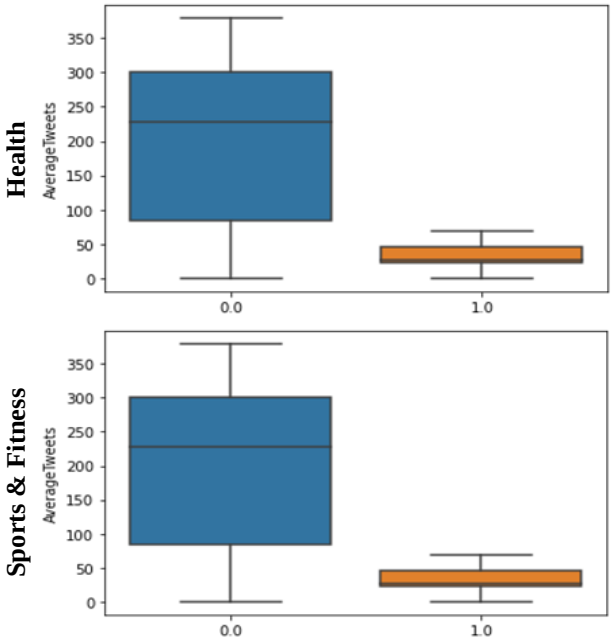


Figure 4. Boxplot of AverageTweets with 3 categories, as specified on the left

From the box plot of average tweets against various categories, it is observed and confirmed that most of the tweets are from News & Politics, and are mainly not about Health, and Sports & Fitness. This supports the earlier collinearity effect, as it stems from the number of tweets and the category the tweets are from. However, this variable may still have certain effect in determining a tweets engagement rate. Hence, these variables are kept and will be used in the modelling phase.

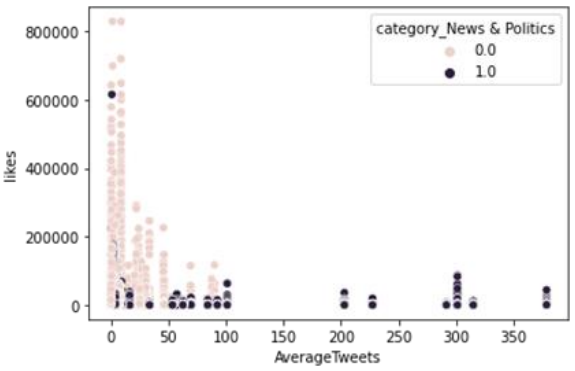


Figure 5. Boxplot of AverageTweets with 3 categories

From the scatter plot of likes against AverageTweets, colored by News & Politics tweets, it is observed that most of the tweets about News & Politics have mostly lower likes, and the average number of tweets per day by these accounts are quite spread out. This shows that keeping the category features is appropriate, since there is not clear trend when compared with the number likes. The modelling phase may uncover other information.

4.3 Category & Topics Exploration

Category

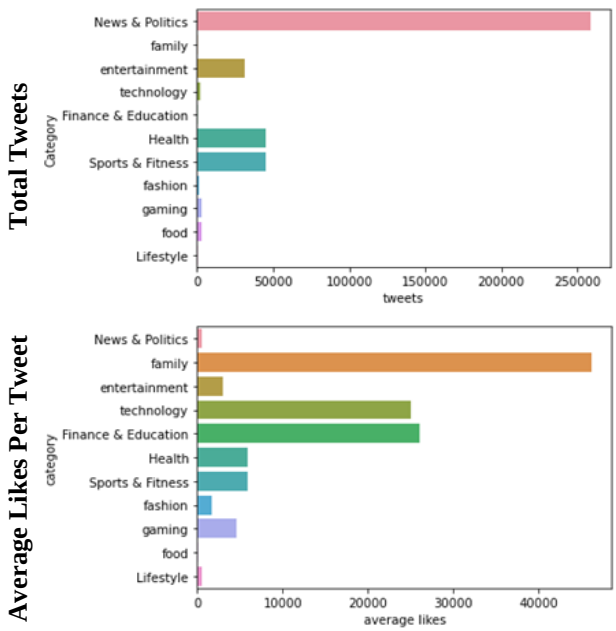


Figure 6. Total number of tweets and average likes per tweet from each category

From the graphs above, it is observed that althrough most of the tweets are mostly from a few categories, this is not proporational to the average engagement rate of these tweets, as shown by the 2nd graph, depicting that tweets about family typically have higher average likes.

One reason could due that due to the higher number of tweets for certain categories, the spread of engagement rate is large, hence resulting in an overall lower average as compared to other categories with lower number of tweet but with a higher proportion of those tweets having higher number of likes.

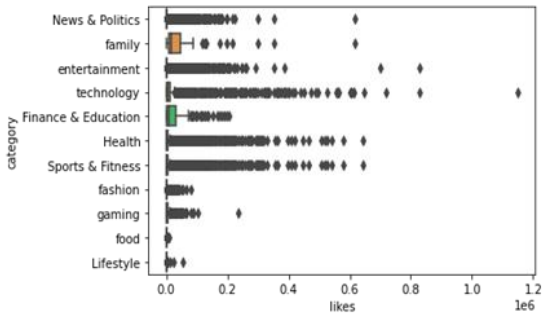


Figure 7. Boxplot of likes for all tweets in each category

As evident from the boxplot of likes for each category, it supports the earlier reasoning that due to the higher proportion of low likes for tweets in categories like News & Politics, the overall average likes is lower as compared to other categories with much lesser number of tweets.

Topics

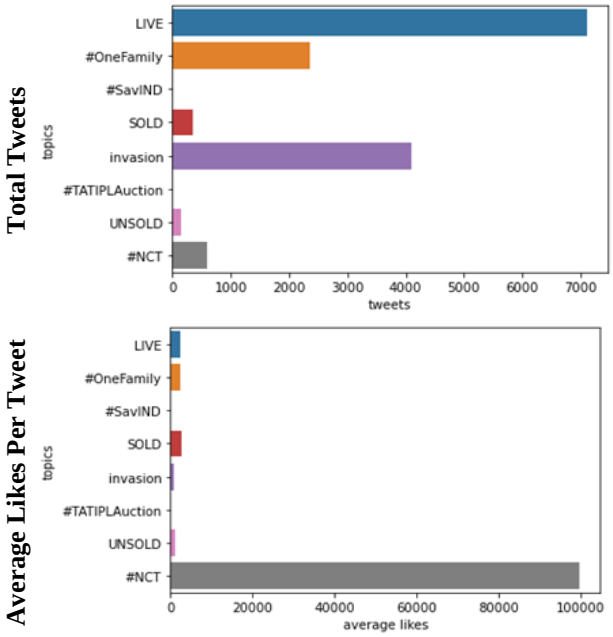


Figure 8. Boxplot of likes for all tweets in each category

Even through most of the tweets are about "LIVE" topics, which essentially could about News, the topic "#NCT" had the overall highest average likes amongst all the topics. This is largely attributed to the fact that the topic "#NCT" is related to a KPOP boy band, which has a very high fanbase, hence contributing to the higher average likes per tweet as compared to other topics. Removing "#NCT", the averages likes is comparable across topics, except for "invasion", which due to the nature of the topic could explain the lower average likes.

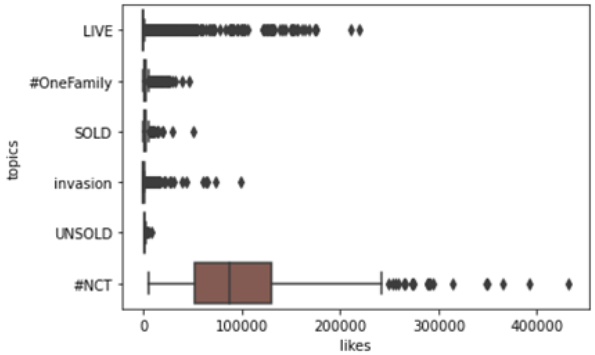


Figure 9. Boxplot of likes for all tweets for each topic

The box plot clearly shows that the topic "#NCT" has an overall higher average likes as compared to other topics such "LIVE" which is clearly observed to have a spread of much lower likes in each tweet.

Hence, posting more tweets per day does not necessarily mean a better engagement rate, rather posting tweets about the trendy topics and category would be more effective, which will be tested in the modelling phase.

5. Modelling

5.1. Key Concepts

Polynomial Regression: Polynomial regression is a form of regression analysis to analyze the relationship which is modelled as an nth degree polynomial.

Random Forest: Random Forest is an ensemble learning method by constructing a multitude of decision trees with different samples for the output variable. It takes the majority vote for classification tasks and average value for regression tasks.

XGBoost: XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

LightGBM: Light gradient boosting machine is a gradient boosting framework that is based on decision tree algorithms.

Mean Absolute Error: Mean absolute error (MAE) is a measure of errors between true observation and predicted observation.

5.2. Hyper-Parameter Tuning & Modelling

All the four models introduced are used for modelling for the prediction of replies, retweets, and likes with hyper-parameter tuning performed.

Mean absolute error (MAE) is the model performance indicator being used for the hyper-parameter tuning and model selections in this project.

Polynomial Regression: Due to the limited computing power resources, the modelling with polynomial regression is performed up to 3rd degree only. However, the performance of the 1st, 2nd and 3rd degree polynomial regression are not differentiable based on MAE evaluated. Hence, the 1st degree of Polynomial Regression, linear regression, is considered as the modelling baseline in this project given the same performance level with higher degree regression but less resources needed.

Table 6. Training MAE for Polynomial Regression Models

Deg.	Replies MAE	Retweets MAE	Likes MAE	Select?
1ST	61.186101	199.219833	1579.115932	✓
2ND	61.186108	199.219840	1579.115934	x
3RD	61.186113	199.219842	1579.115934	x

Random Forest: Grid search is implemented for hyper-parameter finetuning, which formed 4 sets of hyper-parameters for random forest modelling. In addition, cross validation is used to refine the modelling outcome and there are 18 fits in total for the modelling.

- N estimator: 100, 150, 200
- Max depth of decision tree: 8, 10
- Cross validation folds = 3

Table 7. The best model's hyper-parameters for random forest

Target Variable	N Estimator	Max Depth
Replies	100	8
Retweets	150	10
Likes	150	10

XGBoost: Both grid search and cross validation are used for XGBoost modelling to fine out the optimal hyper-parameters for the optimal performance. There are 10 fits in total for the modelling.

- N estimator: 80, 160, 240
- Max depth of decision tree: 6, 9
- Cross validation folds = 5

Table 8. The best model's hyper-parameters for XGBoost

Target Variable	N Estimator	Max Depth
Replies	80	9
Retweets	80	9
Likes	80	9

LightGBM: In total, there are 3600 fits for LightGBM modelling with grid search and cross validation implemented.

- N estimator: range (50, 150,10)
- Max depth of decision tree: range (3,15)
- Column sample by tree: 0.6, 0.8, 1
- Cross validation folds = 5

Table 9. The best performer's hyper-parameters for LightGBM

Target Variable	N Estimator	Max Depth	Column Sample By Tree
Replies	130	7	0.6
Retweets	140	8	0.8
Likes	140	6	0.6

5.3. Modelling Result for Replies, Retweets, Likes

With the hyper-parameter tuning for each model, the best performer for each model is obtained for replies, retweets and likes, respectively. Hence, the models can be compared with each other with its own best MAE performance.

Based on the train MAE and validation MAE, XGBoost is selected to be the final model to be used for tweets' replies given the best performance among all the models.

Table 10. Modelling Result for Replies

Models	Train MAE	Validation MAE	Select?
Poly Regression (baseline)	61.19	-	x
Random Forest	53.44	58.00	x
XGBoost	47.07	55.31	✓
LightGBM	54.46	57.09	x

For tweets' retweets, it is also XGBoost having the best performance compared with the rest models.

Table 11. Modelling Result for Retweets

Models	Train MAE	Validation MAE	Select?
Poly Regression (baseline)	199.22	-	x
Random Forest	146.91	164.69	x
XGBoost	138.24	158.70	✓
LightGBM	158.86	166.50	x

Like tweets' replies and retweets, the comparison for tweets' likes is conducted as well and XGBoost outperforms other models.

Table 12. Modelling Result for Likes

Models	Train MAE	Validation MAE	Select?
Poly Regression (baseline)	1579.12	-	x
Random Forest	1179.06	1309.37	x
XGBoost	1117.73	1277.00	✓
LightGBM	1302.83	1349.55	x

5.4. Modelling Result for Replies, Retweets, Likes

With the final model selected for replies, retweets and likes, the prediction on the test dataset is generated.

Table 13. Test MAE with the best model selected

Target Variable	Test MAE
Replies	52.42
Retweets	158.03
Likes	1274.41

5.5 Explanation

5.5.1 Global Explanation

We use 3 different methods to show the global explanation of our models. The first one is built-in feature importance plot. In XGBoost, the default method to calculate feature importance is using the average gain across all splits where feature was used. The second method is Permutation Feature Importance (PFI). In this method, we will shuffle one of the feature values at a time to get some noise from the original dataset and use trained model to make predictions on this shuffle dataset. The performance deterioration measures the importance of the variable we have just shuffled. The third method is SHAP (Shapley Additive exPlanations) values (Lundberg & Lee, 2017). This method has a solid math theory and can be applied to both global and local interpretation. It can ensure fairness and accuracy in explanation. Under normal conditions, compared with the latter 2 methods, the results of built-in feature importance are less convincing, which can be reflected in the later results.

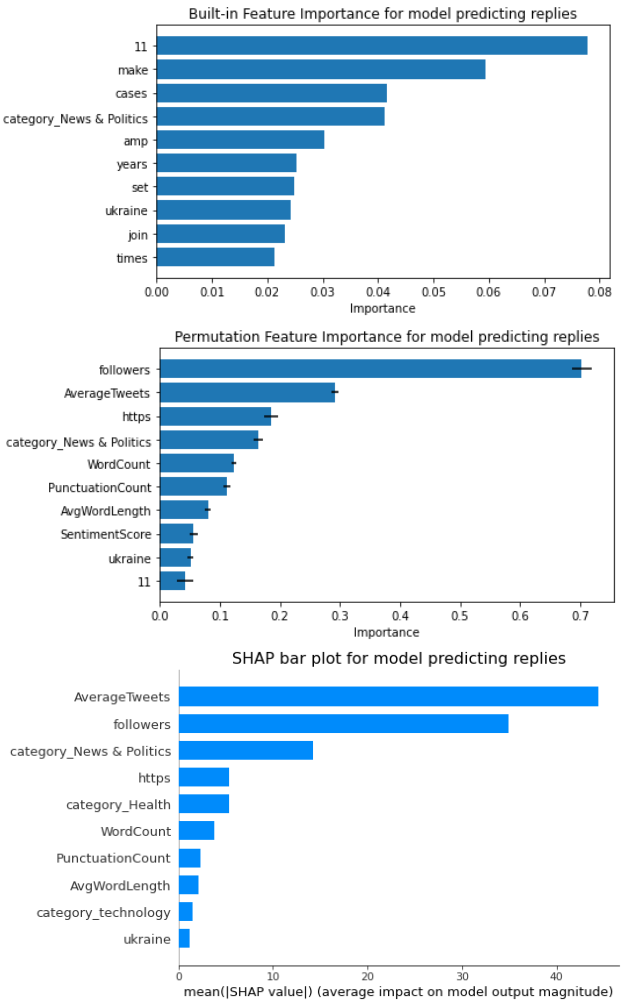


Figure 10. Feature importance plots of Replies

According to the built-in feature importance plot, the most important features are mainly TF-ID features, which seems to be over-fitting. However, in PFI and SHAP results, most important features are some features such as number of followers and daily average number of tweets, which seems to be more convincing. A total of 3 features appears in the 3 results at the same time: https, category_News & Politics and Ukraine. We have successfully found the reasons why these 3 features are very important. On one hand, if we add hyperlinks, hashtags, or pictures when tweeting, we will get a URL that starts with *https*. Therefore, enriching our tweets, rather than just a few sentences, will make our tweets more attractive. On the other hand, because we collected the data up to March 18, 2022, and the war between Russia and Ukraine broke out on February 24, 2022, which directly affected the global political and economic patterns, the category of news and politics and Ukraine became the focus of global Internet users. Consequently, we need to grasp the current hot news if we want to send a popular tweet.

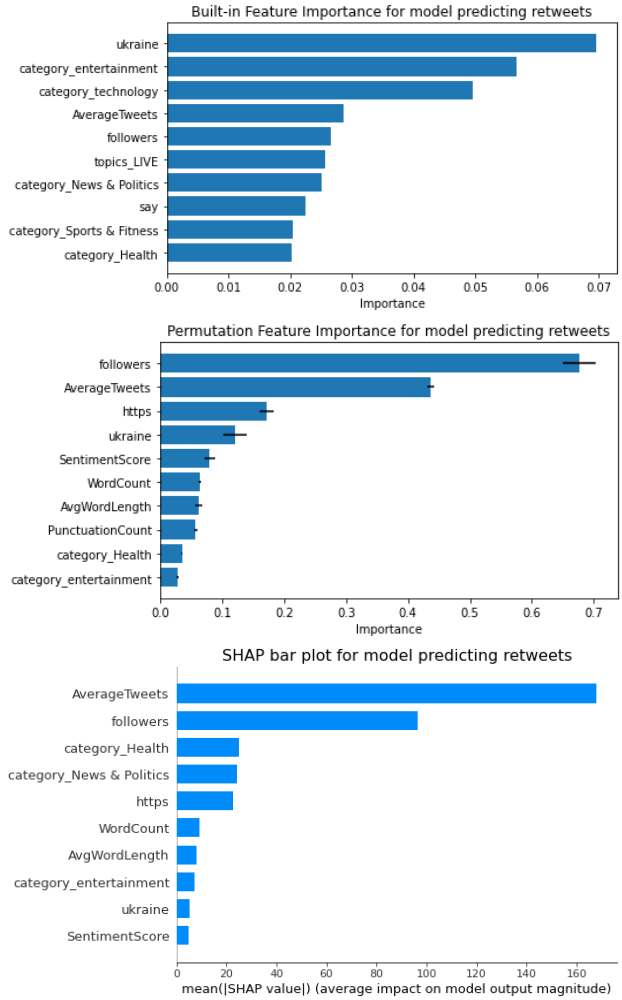


Figure 11. Feature importance plots of Retweets

As for retweets, the important features identified by 3 methods simultaneously are number of followers, the daily average number of tweets, Ukraine, category_Health and category_entertainment. as it's easy to understand that the first 2 features are closely related to the number of retweets, but it's hard to say how and to what extend they will influence the prediction. thus, we will use local interpretability to provide targeted interpretation. concerning the remaining 3 categories or topics, we think that when users want to collect or share some content with their friends, such as some methods to keep healthy or some interesting games, they will retweet. therefore, when one wants to increase the number of retweets, it may be a good idea to share some practical content.

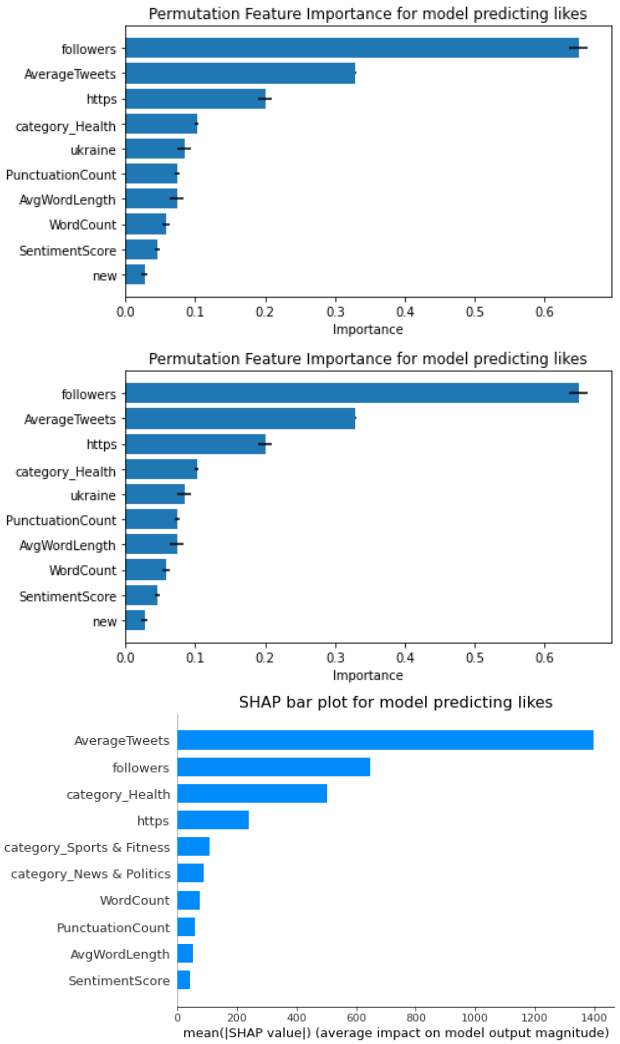


Figure 12. Feature importance plots of Likes

For likes, the common important features are category_Health, daily average number of tweets and number of followers. We think the reason why these variables are important have been introduced in the retweets section.

5.5.2 Local Explanation

We will use 2 different local explanation methods: SHAP and LIME (Local Interpretable Model-agnostic Explanations). In LIME, we will create some new data points around an instance we are interested and use the trained model to make predictions. Then, we fit a simple explainable model to the perturbed data and use the feature weights to explain the local behavior.

Figure 13. Screenshot of a tweet from "Manchester United"

We randomly sampled from ManUtd[1] to show some local interpretability.

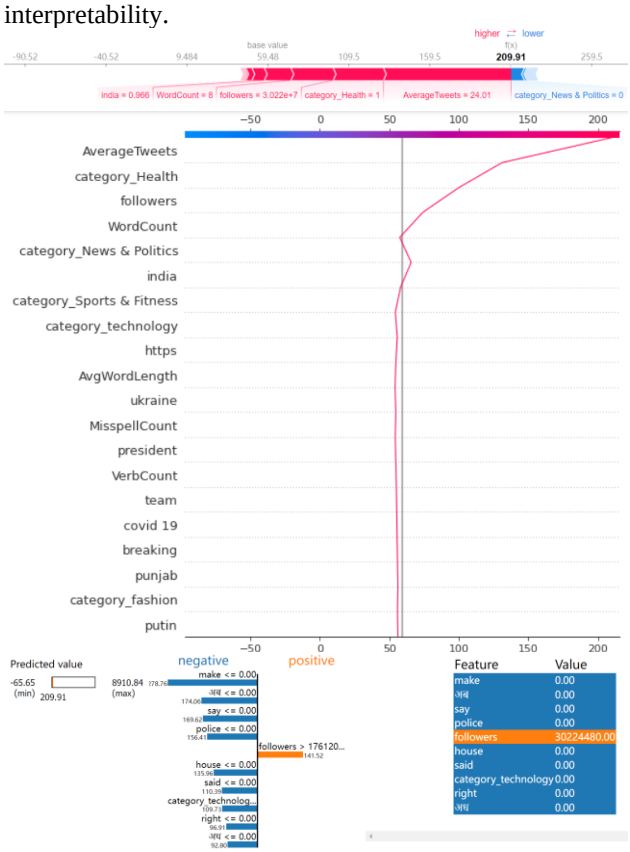


Figure 14. SHAP force plot, SHAP decision plot and LIME plot of Replies

According to SHAP result, features like daily average number of tweets, category_Health, number of followers have a positive impact on the prediction and this tweet is not a news category, which reduces the prediction value.

From the result of LIME, many TF-IDF features have large negative effects on the prediction. Therefore, if user wants to increase the number of replies, he may need to use some words like make, say and police. Another interesting thing shown by LIME is that, there always be some important TF-IDF features like "करे", "करत", "गय". We finally find that they are emoji. However, unfortunately, we don't have enough time to figure out which specific emoji each symbol represents since they are so similar. Anyway, in our sample tweet, LIME suggests that we use one of the emoji.

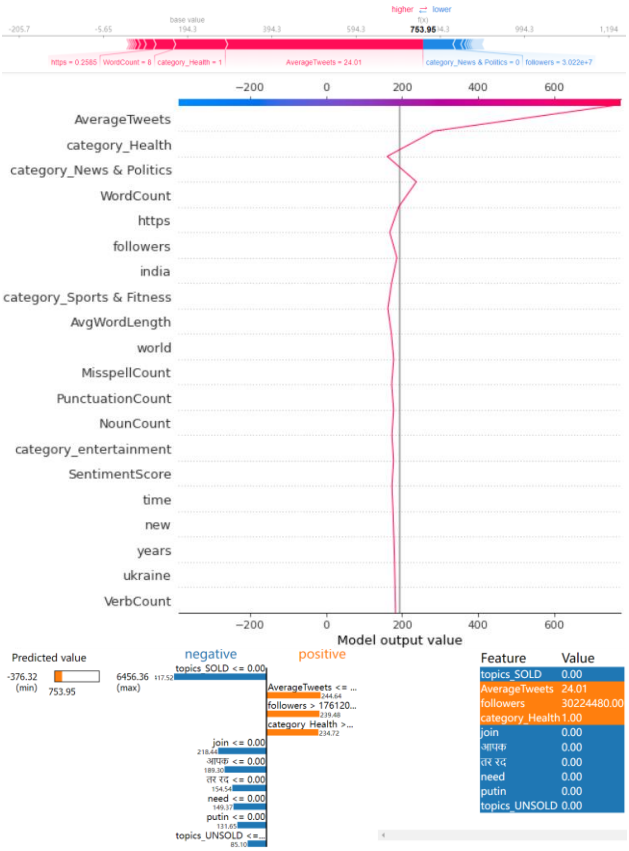


Figure 15. SHAP force plot, SHAP decision plot and LIME plot of Retweets

Combining the 2 results, we believe that users should not tweet too many every day. LIME advises that it's better not to exceed 57 a day. They also both agree that users should tweet some health-related content, if possible, to increase the number of retweets. A major contradiction between 2 methods about the retweets arises from followers. SHAP believes that it reduces the prediction while LIME thinks it increase the final prediction. However, we don't think this is what we can change subjectively in the short term, so we can pay more attention on suggestions on other features.

[1] https://twitter.com/ManUtd/status/1494522060982439937

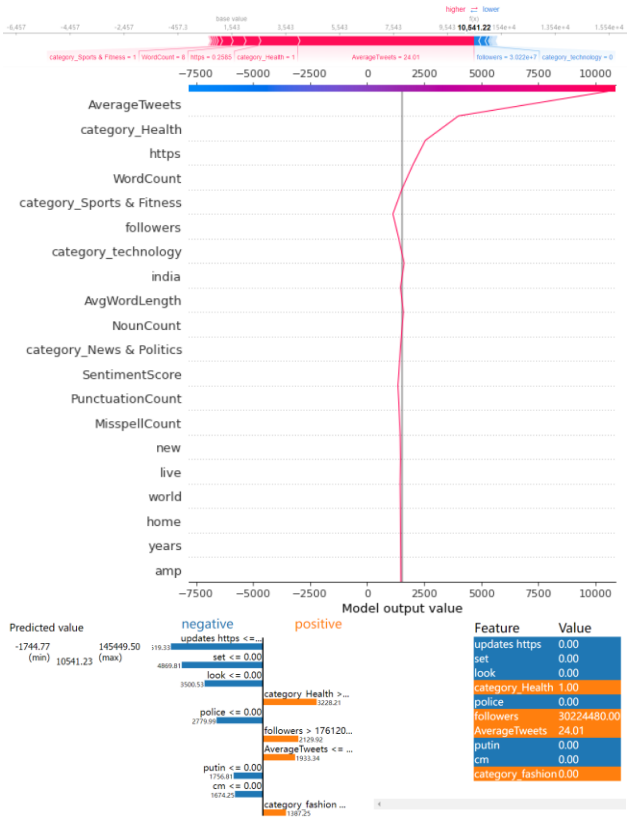


Figure 16. SHAP force & decision plot and LIME plot of Likes

SHAP suggests tweeting something related to technology while LIME recommends some words to use. It's also interesting that LIME believes that content related to fashion will receive less likes.

6. GPT2-Simple Evaluation

We trained a GPT2-Simple model on all of Elon Musk's original Tweets (excluding retweets) since he registered his account in June 2009, with a total of 2,118 Tweets. We then generated 1,000 Tweets based on the model, with the "temperature" parameter set to 1.0. Finally, we run our models on these 1,000 generated Tweets to get the best and worst Tweets based on the predicted engagement levels, which are shown in the tables below.

Table 14. Best tweets with the best predicted results

Tweets	Replies	Retweets	Likes
These guys want us to die so bad they can taste it.	23,026	73,910	354,052
15 mins of static fire	26,746	40,232	312,575
Important news in a few hours …	17,169	42,475	385,821
Good Starship news!	15,405	42,732	378,620
Stop gendering memes … I mean mimes	20,522	28,291	248,752

Table 15. Worst tweets with the worst predicted results

Tweets	Replies	Retweets	Likes
After 2021 Formula 1 cars will have pressure deflection, glare, pucker, bend and bulge lights, traction &ampl traction GPS connected lane changing via touchscreen Tesla LA storybook will be on-par with a Tesla Model S.	785	1,978	20,833
No, I-I-I-don-t-dream-I-would-break-this	1,921	1,027	14,243
Sat/Sun/Moon/Earth/Hyperloop completed!	1,893	1,109	15,063
We took this to heart by Polytopia	731	1,814	24,721
Stop gendering memes !!!	1,112	1,935	20,242

Based on what we see above, there are a few interesting observations. Firstly, the top Tweets seemed easier to read, with very simple words that were well spaced apart. The top Tweets appear to sound confident, positive, and cheeky.

Very interestingly, "Stop gendering memes … I mean mimes" is among the top Tweets, while "Stop gendering memes !!!" is among the worst. These two Tweets have very similar contents but had very different results. The former is cheeky while the latter seems more aggressive.

We noted that this is just a small sample, but it does seem to provide us with some insights on what kind of tone and content would generate more engagement. We can extend this analysis across the other Twitter accounts to gain deeper insights.

7. Limitations

The raw features such as word count, do not exactly help to quantify exactly how much words to stick to, although from previous studies, the guideline would be from 71-100 words.

Another key limitation, from an account perspective, is that the number of followers is a key feature, hence an account with little to no followers will not generate high engagement rate from its tweets.

In addition, due to the limitation of computing power resources, we were only able to test a few parameters for certain modelling technique. Firstly, we did not try "GridsearchCV" with more hyperparameters and wider range of each hyperparameter on models such as Random

Forest and Xgboost which runs relatively slow. Hence, our final model's parameters may not be the best. Secondly, we could try some neural network algorithms if we had better computing power.

In the local explanation section, we find many strange symbols that should be emojis, which can be difficult to identify what exactly each emoji represents.

Users will also need to repeat the whole process of training the models on the latest Tweets periodically to ensure that the model is always up to date with the latest trends in Twitter. Language patterns, hot topics and other factors can change from time to time, and hence, we would recommend repeating the process every 2 to 4 weeks for better results.

8. Future Works

Although the category is considered popular now, it might not be couple of months of years later. Hence, another future work may involve quantifying how long a topic or category usually stays trendy, what are the factors that determine it. In addition, instead of using word count directly, perhaps a deeper study into the length words, together with other features such as the topics category or the number of misspelled words as interaction variables, would help determine the best length of words to use in different cases.

In addition, a different word model can be created to convert emoji to probable phrases and words to better quantify such symbols.

A more advanced NLP model can be built to generate high performing Tweets directly from our model results. For example, we might be able to feed parameters such as topic, word count range, tonality etc, along with the fixed parameters such as the category of the account, or whether the account is a person or a brand etc. Ideally, the NLP model will be able to automatically generate a list of possible good Tweets for the user to select, or perhaps inspire the user to write even better Tweets.

9. Conclusion

In this paper, we explored a systematic approach to help user generate the best possible Tweet. The key steps to our approach can be found in Table 16:

Table 17. Key steps to generate the best possible tweet

Step	Description
1	Extract recent Tweets from the top Twitter accounts.
2	Perform feature engineering on the extracted Tweets.
3	Train machine learning models to predict engagement levels (likes, retweets, and replies).
4	Analyze model results to see what affects engagement levels.
5	Generate Tweet based on model predictions.
6	Choose the best Tweet based on model predictions.
7	Post Tweet.
8	Repeat Periodically.

Users will have to repeat the process periodically to ensure that the model is always up to date with the latest trends in Twitter. Language patterns, hot topics and other factors can change from time to time, and hence, we would recommend repeating the process every 2 to 4 weeks for better results.

Based on the current trends, it is best for Tweets to mention hot topics such as Ukrainian war or to be about health or technology, to contain picture, to contain emojis, to contain hashtags, to be easily readable, to be positive and non-aggressive. Users should also not Tweet more than 24 times a day.

GitHub Code Link

https://github.com/BHLooi/BT5153_Group17

References

Bird, S., & Loper, E. (2004). NLTK. *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions* -. https://doi.org/10.3115/1219044.1219075

Setiabudi, R., Iswari, N. M., & Rusli, A. (2021). Enhancing text classification performance by preprocessing misspelled words in Indonesian language. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *19*(4), 1234. https://doi.org/10.12928/telkomnika.v19i4.20369

Elbagir, S., & Yang, J. (2020). Sentiment analysis on Twitter with Python's Natural Language Toolkit and vader sentiment analyzer. *IAENG Transactions on Engineering Sciences*. https://doi.org/10.1142/9789811215094_0005

Canini, K., Shi, L., & Griffiths, T. (2009, April 15). *Online inference of topics with Latent Dirichlet allocation*. PMLR. Retrieved April 22, 2022, from http://proceedings.mlr.press/v5/canini09a.html

Mansfield, E. R., & Helms, B. P. (1982). Detecting multicollinearity. *The American Statistician*, *36*(3), 158. https://doi.org/10.2307/2683167

Fuchikami, J. (2018, November 17). *K-popping: An introduction to Korean popular music albums*. eVols at University of Hawaii at Manoa: Home. Retrieved April 22, 2022, from https://evols.library.manoa.hawaii.edu/handle/10524/63129

Lundberg, S. M. and Lee, S. I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Wikimedia Foundation. (2022, March 1). *Random Forest*. Wikipedia. Retrieved April 23, 2022, from https://en.wikipedia.org/wiki/Random_forest

Wikimedia Foundation. (2022, April 19). *XGBoost*. Wikipedia. Retrieved April 23, 2022, from https://en.wikipedia.org/wiki/XGBoost

Wikimedia Foundation. (2022, February 8). *Lightgbm*. Wikipedia. Retrieved April 23, 2022, from https://en.wikipedia.org/wiki/LightGBM

Wikimedia Foundation. (2021, November 3). *Mean absolute error*. Wikipedia. Retrieved April 23, 2022, from https://en.wikipedia.org/wiki/Mean_absolute_error

Wikimedia Foundation. (2021, July 6). *Polynomial regression*. Wikipedia. Retrieved April 23, 2022, from https://en.wikipedia.org/wiki/Polynomial_regression

Woolf, M. (2020, January 16). *How to build a Twitter text-generating AI bot with GPT-2*. Max Woolf's Blog. Retrieved April 24, 2022, from https://minimaxir.com/2020/01/twitter-gpt2-bot/