A Sentiment Analysis of COVID-19 Tweets and its effects on Likeability and Retweet-ability.

Andy Lee
500163559
CIND820-D10

Introduction

Social media has expanded tremendously in the past decade. It is common for government officials to use Twitter to relay public messages and announcements (Ontario Government, 2020), and also for individual users to rely on Twitter as their primary source of news (Glenski, et al., 2018). In the past, social media platforms has chosen a laissez-faire approach when dealing with user-generated messages (Caplan, 2017), however with a traffic of 500 millions tweets posted on Twitter per day (Pereira-Kohatsu, et al., 2019) where majority are posted by individuals users (Oren, et al., 2020), Twitter has slowly become a breeding ground for rumours and misinformation (Shao, et al., 2018); an ongoing issue social media platforms struggle to find a balance to address (Culliford, 2020).

As we step into 2020, the COVID-19 pandemic fills our everyday news while causing disturbance to all levels of society across the global (Hinshaw, 2020). Due to the novelty of the virus, governments and health officials often struggle to form straightforward and coherent guidelines and policies due to discovery of new information (Farzan, et al., 2020). This can contribute to public mistrust on the health authorities; indirectly fanning the spread of rumours and misinformation (Kouzy, et al., 2020). In turn, it hinders the effectiveness of public health policies when people feed on misinformation (Bode & Vraga, 2017) as studies have shown misinformation are spread and consumed by like-minded people; ricocheting like an echo chamber (Vicarioa, et al., 2016).

Monitoring the vast amounts of social media messages have become humanly impractical, therefore research into using machine learning and data analytic techniques to tackle the problem has become a popular research topic in recent years, and many papers have described partial or complete solutions with different degrees of success. In our capstone project, we will attempt to implement a small part of a big puzzle; create a new feature using sentiment analysis tools to score each tweet, and then build regression models to predict whether sentiment score, along with other features, contribute to retweets or likes. This will allow us to better understand whether a tweet's sentimentality can influence how it resonates with other people.

Literature Review

Here is the literature review of some academic papers describing challenges when conducting Natural Language Processing on social media message since they come in a various of formats compare to traditional literature.

Twitter rumour detection in the health domain

Sicilia et al. in "Twitter rumour detection in the health domain" (Sicilia, et al., 2018, pp. 34-35) described the construction of a complete detection system that can distinguish rumour or non-rumour tweets on topics related to the health sector. Rumours are defined as information from an unverified source, non-rumours are information that can be referenced to credible sources and official pages, and unknown are information that cannot be verified has either true or false (Sicilia, et al., 2018, p. 35). Data downloaded from Twitter, separated based on user or network level, feature selected based on performance, and finally trained into classification model (Sicilia, et al., 2018, pp. 35-36). The author used various machine learning techniques such as Support Vector Machine, Nearest Neighbour, and Random Forest and compared the results based on their averaged accuracies and compared their p-values (Sicilia, et al., 2018, pp. 38-39). The study was able to achieve an over accuracy of 74%, precision of 73%, and recall of 74% (Sicilia, et al., 2018, p. 39).

Ravi in "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications" conducted a comprehensive review and experimented on different sentiment analysis techniques described in over 300 papers on six tasks, namely: subjectivity classification, sentiment classification, review usefulness measurement, lexicon creation, opinion spam detection, and aspect extraction" (Ravi, 2015, pp. 4-5). The study compared dozens of different sentiment analysis approaches and techniques with various degree of success on certain machine learning techniques when compared to vote count based measures (Ravi, 2015, pp. 55-57). Ravi concluded that there is still room for growth on intelligence-based techniques such as Random Forest (Ravi, 2015, p. 64).

Carlos et al. in "Detecting and Monitoring Hate Speech in Twitter" observed that social media platforms dominate much of the internet, and while hundreds of millions of messages go through these platforms, hate messages begin to spread in wide varieties of subjects (Pereira-Kohatsu, et al., 2019, p. 2). The author then devises a system accepting text and emoji as input, and can identify and classify hate speech in Twitter, and monitoring negative sentiments (Pereira-Kohatsu, et al., 2019, pp. 1-2). After experimenting with 19 strategies of feature and classification models, the authors conclude LSTM and MLP-NN the best, achieving AUC of 0.828 (Pereira-Kohatsu, et al., 2019, p. 31).

Social media platforms currently are mostly unmoderated; allowing rumours or unverified information to spread and circulate easily on their platform (Zubiaga, et al., 2018, p. 32:1). Zubiaga et al. in "Detection and Resolution of Rumours in Social Media: A Survey" describes combining different techniques in rumour detection, tracking, stance classification, and veracity classification to form a complete rumour detection system, which will provide users with early warnings of messages containing uncertain information (Zubiaga, et al., 2018, pp. 32:1-32:2).

Sewalk et al. in "Using Twitter to Examine Web-Based Patient Experience Sentiments in the United States: Longitudinal Study" believes health care needs to shift strategy; tending more towards patient experience, emotion and engagement (Sewalk, et al., 2018, p. 2). As Twitter has become a platform for people to voice their opinions, the authors conducted sentiment analysis on patient tweets in Twitter as a means to evaluate patient health treatment experiences, such as treatment wait time (Sewalk, et al., 2018, pp. 10-11).

Dataset

For this project, we will use a dataset of Tweets IDs, filtered with 90+ COVID-19 related keywords and hashtags (link), publicly available for download from IEEE DataPort (https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset). Since the dataset only contains tweets IDs, it must be converted to Tweets using a Hydrator program. The resulting hydrated CSV have 34 attributes: `coordinates, created_at, hashtags, media, urls, favorite_count, id, in_reply_to_screen_name, in_reply_to_status_id, in_reply_to_user_id, lang, place, possibly_sensitive, retweet_count, retweet_id, retweet_screen_name, source, text, tweet_url, user_created_at, user_screen_name, user_default_profile_image, user_description, user_favourites_count, user_followers_count, user_friends_count, user_listed_count, user_location, user_name, user_screen_name, user_statuses_count, user_time_zone, user_urls, user_verified`.

For our project, we will be using 9 features where the "compound" feature will be generated by sentiment analysis tool:

Table 1 List of attributes used in this project.

Feature	Description
favorite_count	No. of "liked" this tweet has
retweet_count	No. of retweet this tweet has
user_followers_count	No. of follower Tweet's account has
user_statuses_count	No. of tweets and retweets this user issued
user_favourites_count	No. of tweets this user liked
user_followers_count	No. of followers this account has
user_friends_count	No. of users this account follows
user_listed_count	No. of public lists this user is a member of
Compound	New feature: sentiment score calculated by sentimental analyzing tool with values between -1 negative and +1 positive.

Approach

Figure 1 shows a sequential approach of our capstone project. The steps should align with the code in Jupyter notebook.

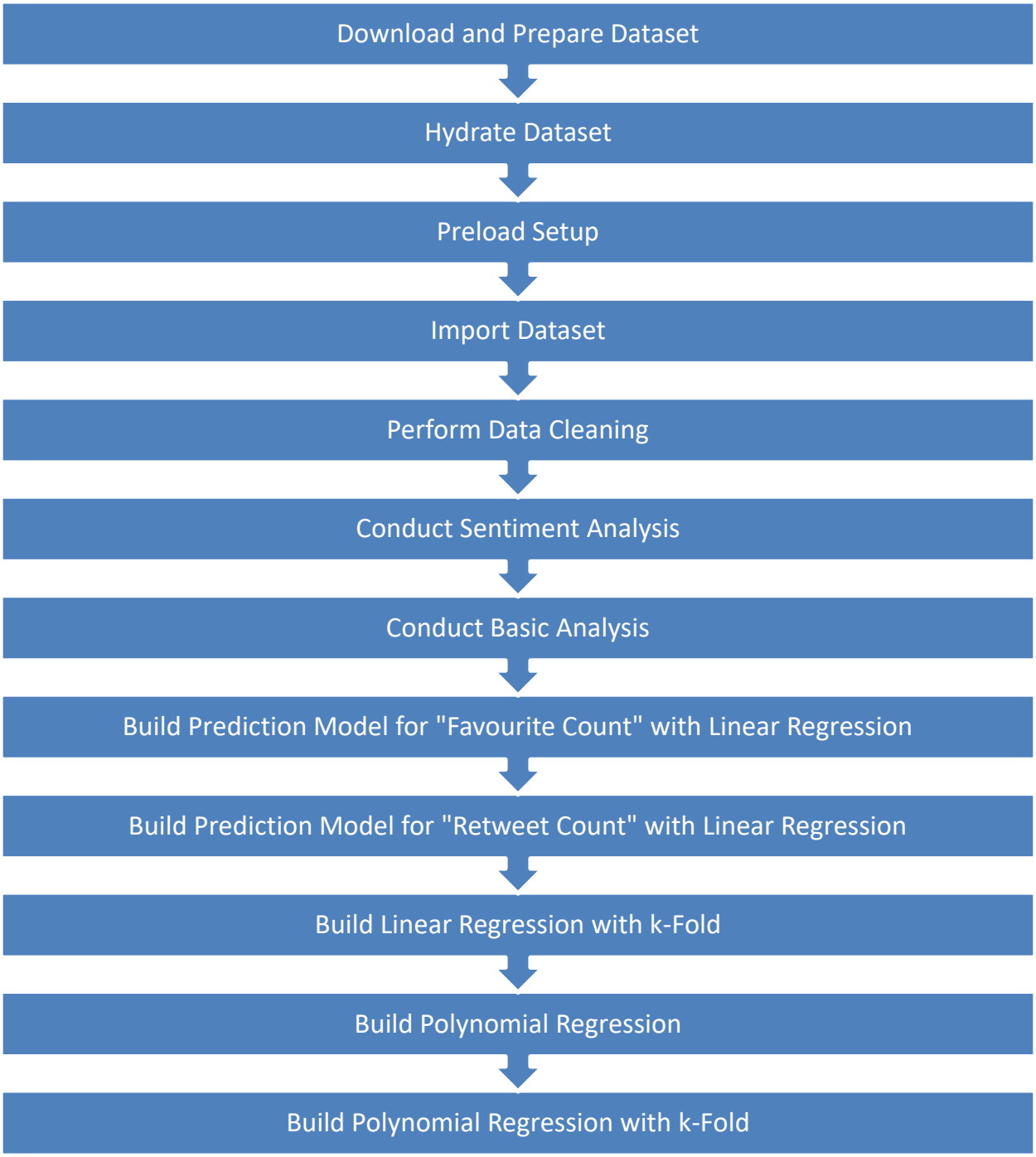

Figure 1 Project approach diagram

Step 1: (Pre-Python) Download and Prepare Dataset

IEEE.org Coronavirus Tweet Dataset (https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset) contains a list of individual datasets broken by their date since March 2020. Choose any date and download its CSV. The CSV file contains only two features: Tweet ID and sentiment score. Since our task is to conduct our own sentiment analysis on the tweets, we will discard the sentiment score and preserving only Tweet ID. Open the CSV file with any CSV editor of your choice, such as Excel, and remove the sentiment score (2nd attribute) from the CSV. There should only be one attribute left, and one tweet ID per row. Save this file as .txt and close the file.

Step 2: (Pre-Python) Hydrate dataset

The tweet IDs will need to be converted into Tweets. This process involves loading the .txt file to the Hydrator program which will sequentially pull Tweets using Twitter's API. The procedures are as follows:

Download Hydrator from https://github.com/DocNow/hydrator, then follow the installation procedures, and open the Hydrator program after installation is complete. In the Hydrator program, click "Setting" tab on the top, and follow the instructions to link the program to your Twitter account. This provides permission for the Hydrator program to read Tweets using the Tweet IDs. Then, switch to the "Add" tab, click "Select Tweet ID file" and select the previously created .txt file with the Tweet IDs. Under "Title" field, give the dataset any name, then click "Add Dataset" button, and the program will switch to the "Dataset" tab. Click the "Start" button for the hydration process to begin. Note that this will take some time as there is a limit to how many tweets a Twitter Account can retrieve within every 15 minute timeframe. Once the hydration process is complete, click "CSV" button and a dialog box will pop up asking for a new CSV filename to save the results into—provide any descriptive filename. Upload this CSV file onto a cloud storage service and proceed to the next step—we used Google Drive for this project.

Step 3: (Python) Preload Setup

On Jupyter (Google Colab) notebook, preload basic parameters such as display tables in full screen widths, increasing number of columns displayed, or other settings to improve productivity while working on the notebook.

Step 4: Import Dataset in Pandas Dataframe

We need to give Colab access to the CSV file previously stored in Google Drive. First, mount Google Drive onto the Colab project by loading the Google Colab library, then run and follow instructions (see Figure 2). Google will ask whether to grant permissions to Colab, and upon acceptance it will return with an authorization code.

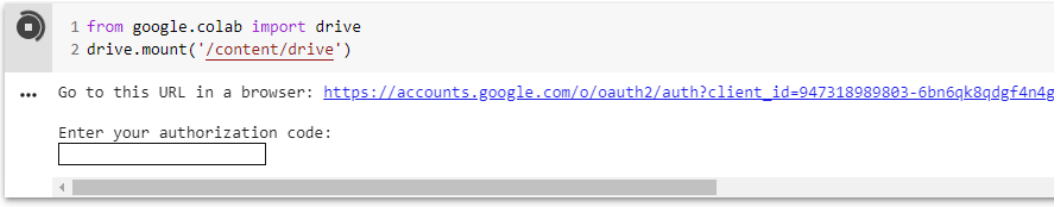

Figure 2 Instructions to follow the link to get authorization code

Copy and paste the authorization code back to Colab notebook into the provided text box. If the code is valid, a message will appear "Mounted at <*some path*>" (see Figure 3) which indicates the authorization is successful and Colab is given read privileges to the CSV file in Google Drive. Find out the file path and use read_csv to load the CSV file, then store the dataset in a Pandas dataframe.

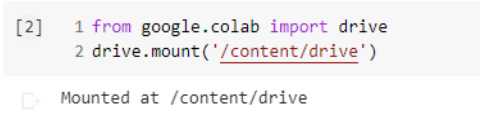

Figure 3 Status should "Mounted" after successfully pasting a valid authorization key.

Step 5: Perform Data Cleaning

Take a moment to observe and understand the data's structure and composition of the dataframe. Then, perform data cleaning such as handle null values, convert attribute data types, and remove noise from strings. For more detailed understanding of Twitter's data structure, visit Data dictionary reference at https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/user-object. Notice that the feature labels in our CSV might be longer and appear to have extra prefixes; this is because the raw data retrieved from Twitter API is in multi-dimensional JSON format, and the structure has to be flatten to two dimension before saved as CSV.

Since retweets are duplicates of the original tweet, we will filter the dataset to only include Original Tweets, excluding retweets, and save them into a new dataframe—call it dfOriginal.

The sentiment analyze algorithm works best on English text, and tweets that are not written in English waste processing resources and can affect our results. Using langdetect library, we parse each Tweet and exclude any data points containing complete or partial non-English tweets. This process will take some time to complete.

Step 6: Sentiment Analysis

From NLTK library, import SentimentIntensityAnalyzer and apply the polarity_scores function onto each Tweet in dfOriginal. The function returns a dictionary object: probability of negative, probability of neutral, probability of positive, and normalized compound score. We can expand the dictionary object using their key as feature and value as datapoint into 4 new features and merge the new features back to the dataframe for easier analysis.

The "compound" score is calculated from raw sentiment intensity and not derived from negative, neutral, and positive probabilities. It scores between -1.00 and +1.00 and provides us with a single dimension sentiment score which is easier to work with.

Step 7: Basic Analysis

Now that we have all features we need, we can conduct some basic analysis such as, correlation, mean, standard deviation, and quartiles on the 9 features.

Using the "compound" score, we can plot a histogram based on frequency and observe its distribution. By categorizing each tweet using the compound score as: negative between -1 to -0.1, neutral between -0.1 to +0.1, and positive between +0.1 to +1, we can plot a pie chart to show the ratio between each category.

Step 8: Build Prediction Model for "Favourite Count" with Linear Regression

Of the 9 features, we use "favorite_count" as dependent variable, and the rest as independent variable. Sample 80% of the dataset into train set, and 20% into test set. Using sklearn library, linear_model class, LinearRegression() method, load the dependent and independent variables from the train set into LinearRegression(), and then using the independent variable from the test set to predict the dependent variable. Then compare the predicted dependent variable with the actual dependent variable from test set to obtain metrics on the fit of the model, such as MSE, RMSE, MAE, R^2, and adjusted R^2.

Next, plot a scatterplot using the predicted dependent variable as y-axis, and actual dependent variable from test set as x-axis will give us a visualization of how well the prediction versus actual values are. If outlying values "stretch out" the graph hindering visualization, we can limit the scatterplot to show prediction within 3-sigma, or 2-sigma range.

Step 9: Build Prediction Model for "Retweet Count" with Linear Regression

Build another linear regression model just like Step 8, but substitute "retweet_count" as the dependent variable.

Step 10: Linear Regression with k-fold

In Step 8 and 9, the linear regression was built using only one sampling of the dataset. In this step, we will modify and replace the sampling using *k*-fold cross-validation.

From sklearn.mode_selection, import KFold. Initialize some variables used to accumulate MSE, RMSE, and R^2 scores. Initialize KFold and specify the number of folds, load the KFold split onto a variable. Then create a for loop to iterate through this variable, and within each iteration run LinearRegression() just like Step 8's "favorite_count" as dependent variable. After iteration is done, take the accumulated MSE, RMSE, and R^2 and divide by the number of KFold to get the average score respectively. Print the average MSE, RMSE, and R^2 scores on screen.

After "favorite_count" as dependent variable is complete, repeat Step 10 again but substitute "retweet_count" as the dependent variable—just like Step 9.

Step 11: Build Polynomial Regression

Here we will use polynomial regression on a single sampling (expand based on Step 8) to see whether we can build a better model than the previous linear regression in Step 8.

First from sklearn.pipeline, import make_pipeline, and from sklearn.preprocessing import PolynomialFeatures. Specify the number of degrees to iterate through. Degrees is the number of terms in a polynomial equation, and more terms adds complexity which requires more computation power and memory—we used 1, 2, 3, and 6 for our model. Just like in Step 8, sample 80% of the dataset to train set and remainder 20% to the test set. Create an iteration by degrees, and for each degree, use make_pipeline to load the dependent and independent train set, along with the number of degrees (PolynomialFeatures) into the LinearRegression() which will perform polynomial regression. Then, we feed the independent variables from the test set into the polynomial regression, which will return predicted values, and store those values in a multi-dimensional numpy.array.

To create a scatterplot using matplotlib, iterate through each row in the array and cycle a different colour for each column. This will result to a scatterplot where each colour represents its own degree prediction (see example Figure 21 Polynomial regression with favorite_count as dependent variable).

Next, get the R^2 score by iterating the array again and use r2_score() to generated the R^2 score for each degree (see Figure 4).

```
58 for i, degree in enumerate(degrees):
59     test_r2 = round(r2_score(Y_test, y_test_pred[:, i]), 2)
60     print("Polynomial degree {0}: test score={1}".format(degree, test_r2))
```

Figure 4 R^2 score for polynomial regression

Step 12: Build Polynomial Regression with k-Fold

For Polynomial Regression with k-Fold, use Step 10's *k*-fold implementation, and iterate each fold using Step 11's implementation. Note that the variables used to accumulate MSE, RMSE, and R^2 score will need to change to multi-dimensional since there are extra dimensions for polynomial regression—degrees.

Complete this step for both dependent variables: favorite_count and retweet_count.

Results

In this section, we will first showcase some interesting observations, then results from the regression models.

Data Cleaning

Tweet text can be filled with noise which might hinder sentiment analysis algorithm. Figure 5 shows some example of Tweet text noises: \n for newline that is concatenated with ordinary English words; user referrals (@RehamKhan1) using a user's handle name which has no sentiment meaning on its own other than referencing another user; and hashtag (#CoronaJihad) with multiple words concatenated together without spaces in between.

	text
714006	RT @Maashish81us: Live #CoronaJihad\n\nIn india , these terrorist skull caps are spreading Corona in india\n\n80% Corona Infected people are #M…
368670	Good.\nPlease always try to tweet about highlighting positive things of life. \n\nNo use going after IK/PTI.\n\nYou can and should contribute alot more in different ways.\n\n@RehamKhan1 https://t.co/IkvY8XibLF

Figure 5 Example of Tweets with noise

Observations from Tweets and Compound Score

Observing the compound score generated by the algorithm, we notice some interesting results. Figure 6 shows two tweets where the one on top has no context, and the bottom tweet has a negative sarcastic tone; both tweets scored 0.00. Figure 7 shows another example of negative sarcastic tone scoring 0.00.

494723	This 👇👇👇	0.0000
444492	... the IOF are filth, Hazmat suits don't contain their scum.	0.0000

Figure 6 Tweets with sarcastic tone or without context

10781	Trump cannot remember that he learned of the corona virus in January!	0.0000

Figure 7 Tweet with sarcastic tone

Figure 8 shows a tweet that reads neutral, but has a score of 0.8176 suggests some words, such as "positive", can incorrectly boost scores.

338328	50 New positive cases in TodayTotal number of positive cases in stands at 124.	0.8176

Figure 8 Tweet with "positive" word

Figure 9 and Figure 10 show tweets written in complete sentences and given a score that appears justified, suggests the sentiment analyze works best with proper and complete sentences.

104640	To all the doctors, nurses, police men, service men, honourable Prime Minister & every Indian fighting Corona, I believe in the power of social media. We stand strong with you'll. We'll stay home to save India. A poem written by me!	0.8122

Figure 9 Tweet written in complete sentence

302884	Honorable prime minister of India good evening.Please give a chance to Dr Biswaroop Roy Chowdhury to cure corona patients 🙏	0.8126

Figure 10 Another tweet written in complete sentence.

Sentiment Score Distribution

We take the compound feature from the dataset and plot a frequency histogram (Figure 11) to observe the distribution of the sentiment score. There are a few observations based on the graph: there is an extremely high concentration of sentiment scored between 0.00 and 0.05; the two extremes at -1.00 and +1.00 have the lowest count; and the histogram appears to be multimodal. The compound score can also be grouped into 3 categories based on Deo et al.'s threshold designations: negative for compound scores between -1.00 and -0.10, neutral between -0.10 and 0.10, and positive between 0.10 to 1.00. The distribution based on this categorization (see Figure 12) shows percentage of negative and positive are fairly close, where neutral is the smallest due to its narrow thresholds.

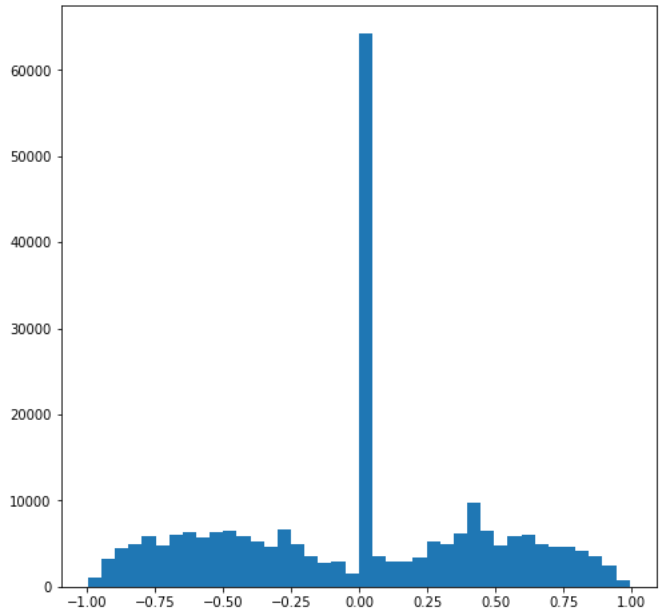

Figure 11 Histogram for Sentiment Score in 40 bins

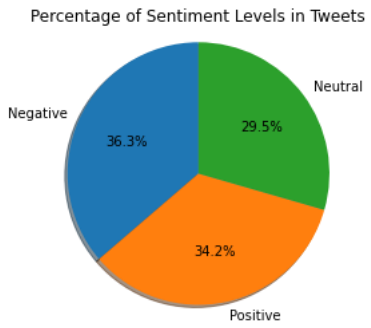

Figure 12 Percentage of Negative, Neutral and Positive Sentiment Levels in Tweets

Basic Analysis

Figure 13 shows a table of correlation between 9 features which will be used as our dependent and independent variables for regression model. Favorite_count (number of likes of a tweet) and retweet_count (number of retweets of a tweet) has the highest correlation of 0.91, and user_followers_count (number of followers this tweet's account has) and user_listed_count (number of public lists this users is a member of) has the second highest correlation of 0.61. The introduced feature "compound" has a very low correlation with all other features.

	favorite_count	retweet_count	user_followers_count	compound	user_statuses_count	user_favourites_count	user_friends_count	user_listed_count
favorite_count	1.000000	0.908576	0.165683	-0.001108	0.015366	0.007852	0.005575	0.051646
retweet_count	0.908576	1.000000	0.090025	-0.003071	0.015991	0.007261	0.007084	0.039251
user_followers_count	0.165683	0.090025	1.000000	0.003693	0.116452	0.000535	0.032721	0.609057
compound	-0.001108	-0.003071	0.003693	1.000000	-0.022604	-0.028941	-0.006560	0.000249
user_statuses_count	0.015366	0.015991	0.116452	-0.022604	1.000000	0.326723	0.105958	0.128835
user_favourites_count	0.007852	0.007261	0.000535	-0.028941	0.326723	1.000000	0.100920	0.016049
user_friends_count	0.005575	0.007084	0.032721	-0.006560	0.105958	0.100920	1.000000	0.048344
user_listed_count	0.051646	0.039251	0.609057	0.000249	0.128835	0.016049	0.048344	1.000000

Figure 13 Correlation between 9 features

Figure 14 shows a table with some basic statistics of each feature. The standard deviation for all features except compound is high relative to its mean, which suggests the distribution is spread out to the right.

	favorite_count	retweet_count	user_followers_count	compound	user_statuses_count	user_favourites_count	user_friends_count	user_listed_count
count	245389.00000	245389.00000	245389.00000	245389.00000	245389.00000	245389.00000	245389.00000	245389.00000
mean	9.52971	2.70475	9251.20519	-0.01655	27468.30799	18028.51423	1284.43071	34.64758
std	384.21091	122.27732	250678.23602	0.47919	75560.36124	39592.12977	10409.72569	680.04372
min	0.00000	0.00000	0.00000	-0.99520	1.00000	0.00000	0.00000	0.00000
25%	0.00000	0.00000	64.00000	-0.39760	1156.00000	510.00000	135.00000	0.00000
50%	0.00000	0.00000	331.00000	0.00000	6344.00000	4200.00000	399.00000	1.00000
75%	1.00000	0.00000	1307.00000	0.36120	24616.00000	17807.00000	1038.00000	6.00000
max	108137.00000	35137.00000	62855265.00000	0.99190	4472178.00000	1254400.00000	4322723.00000	202433.00000

Figure 14 Describing the 9 features

Linear Regression

Favorite_count as dependent variable

Figure 15 shows a scatterplot with favorite_count as dependent variable and the eight other features as independent variable. Observations are dense when the count is under 20, and then gradually spread to the right as the count increases. Prediction does not appear to do exceptionally well, especially when the count is under 20.

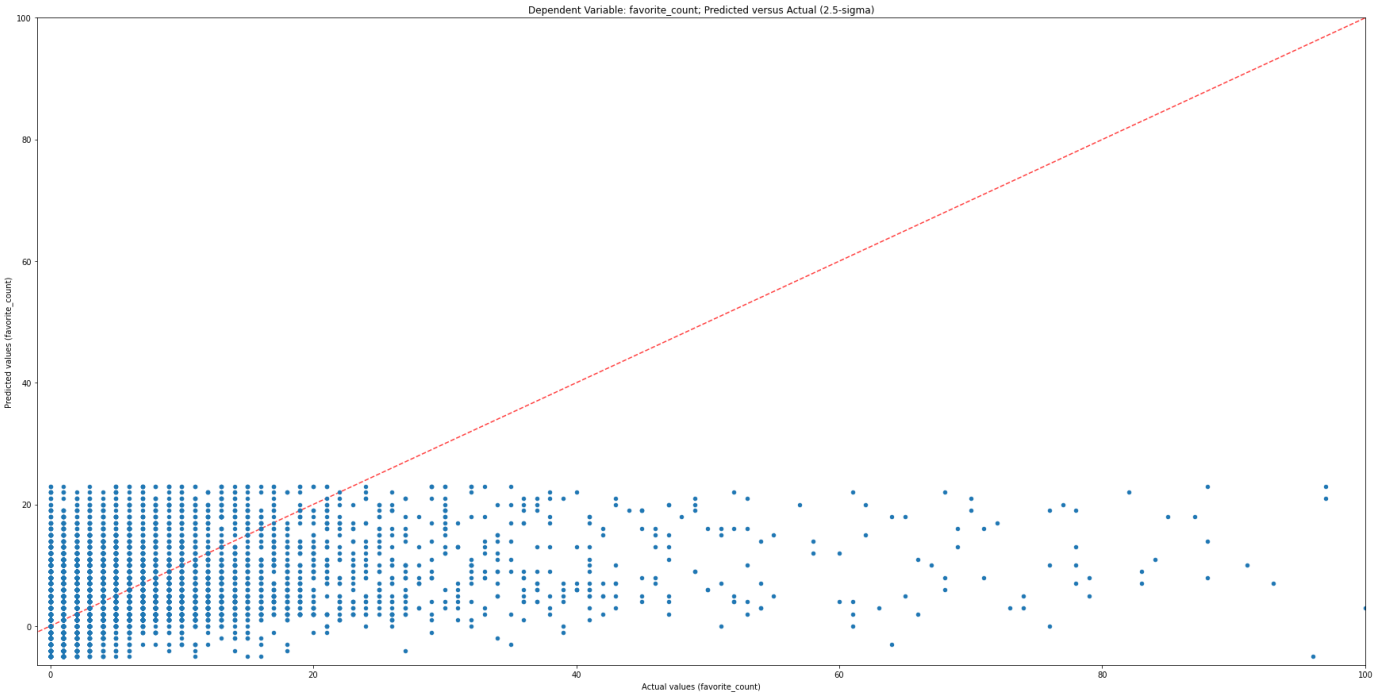

Figure 15 Scatterplot for Predicted vs Actual; favorite_count as dependent variable; Linear Regression

The metrics from Figure 16 shows that retweet_count and compound feature have the highest coefficient in the linear regression model. The RMSE of the model is 182.77 which is high give that retweet_counts are mostly under 20, however it also has an $R^2 = 0.87$, which shows the model fits close to the regression line.

```
Coefficient     Label                  Metrics
=====================================  ======================================
 2.536713       retweet_count          Mean Square Error (MSE): 33404.189046
 0.000211       user_followers_count   Root MSE:                182.768129
 1.016376       compound               Mean Absolute Error:     7.281756
-0.000038       user_statuses_count    R^2:                     0.873217
 0.000057       user_favourites_count  Adjusted R-Square:       0.873199
-0.000056       user_friends_count
-0.034559       user_listed_count

R^2-score (train set): 0.799859
```

Figure 16 Metrics for Figure 15

retweet_count as dependent variable

Next, we build another linear regression model with retweet_count as dependent variable and created the scatterplot as shown in Figure 17. Similar to the previous linear regression shown in Figure 15, there is a high concentration of count under 25, and gradually disperse to the right and slightly upwards as the count increases.

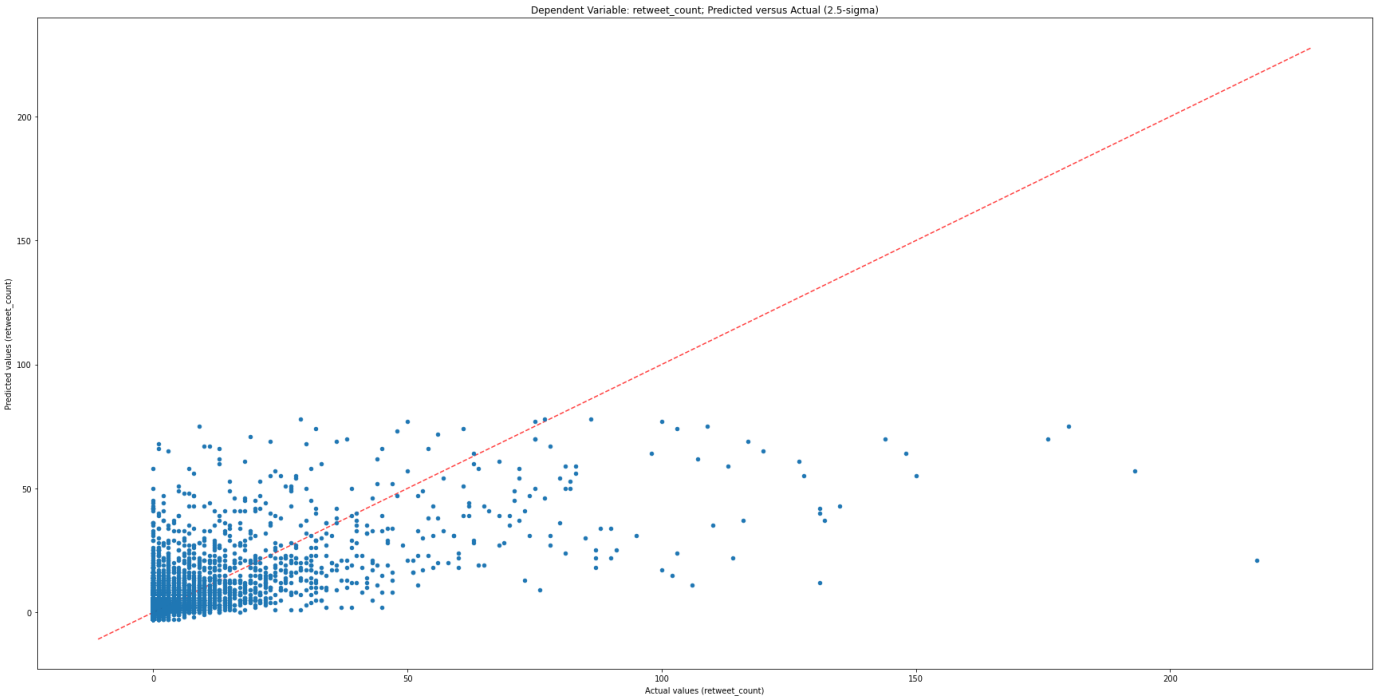

Figure 17 Scatterplot for Predicted vs Actual; retweet_count as dependent variable; Linear Regression

The metrics for Figure 17 is shown in Figure 18 where favorite_count and compound features have the highest coefficient in the model. The RMSE of the model is lower at 35.14 and a high R^2 iof 0.93.

Coefficient	Label	Metrics	
0.311235	favorite_count	Mean Square Error (MSE):	1234.588084
-0.000055	user_followers_count	Root MSE:	35.136706
-0.449341	compound	Mean Absolute Error:	1.967480
0.000013	user_statuses_count	R-Square:	0.934187
-0.000014	user_favourites_count	Adjusted R-Square:	0.934178
0.000025	user_friends_count		
0.009861	user_listed_count		

Figure 18 Metrics for Figure 17

Linear Regression with *k*-fold

Since the previous linear regressions is conducted with only one sampling, we will use *k*-fold = 5 and reconduct the two linear regression models again to get a more accurate RMSE and R^2 of the models.

K-fold, favorite_count as dependent variable

Figure 19 shows the RMSE to be 171.28 compare to previous 182.77 is a slight improvement. R^2 is 0.73 compare to previous 0.87 is a slight decrease.

```
Dependent Variable: favorite_count
Independent Variables: ['retweet_count', 'user_followers_count', 'compound', 'user_statuses_count', 'user_favourites_count', 'user_friends_count', 'user_listed_count']
K-fold: 5
MSE                     RMSE                    R-Square
================================================================================================
31564.218170            177.663216              0.877825
10521.044828            102.572145              0.704872
130028.329157           360.594411              0.378130
10211.723622            101.053073              0.782832
13119.346895            114.539718              0.930201

Average
================================================================================================
39088.932535            171.284512              0.734772
```

Figure 19 K-fold linear regression with favorite_count as dependent variable

K-fold, retweet_count as dependent variable

Figure 20 shows the RMSE to be 43.49 compare to previous 35.14 is a slight improvement. R^2 is 0.79 compare to previous 0.93 shows a slight decrease in average model fit.

```
Dependent Variable: retweet_count
Independent Variables: ['favorite_count', 'user_followers_count', 'compound', 'user_statuses_count', 'user_favourites_count', 'user_friends_count', 'user_listed_count']
K-fold: 5
MSE                     RMSE                    R-Square
================================================================================================
1001.568224             31.647563               0.945563
798.111384              28.250865               0.605363
10974.758804            104.760483              0.685319
728.033332              26.982093               0.750776
666.921016              25.824814               0.959674

Average
================================================================================================
2833.878552             43.493164               0.789339
```

Figure 20 K-fold linear regression with retweet_count as dependent variable

Polynomial Regression

Next, we will attempt to produce a better model with another regression—polynomial regression. First, we create a polynomial regression for each of the two dependent variables—favorite_count and retweet_count using one sampling and degrees 1, 2, 3, and 6 to create a scatterplot for visual observation. Then, we will run the models again with *k*-fold, just like previous linear regression models. Due to the increased in computation resources for polynomial regression in multi-degrees, we must lower the number of independent variables from 9 features to 5 features. Although we have chosen the four features with the lowest coefficients to drop, the coefficient values changes from time to time depending on the train set's sampling, therefore there is a chance it might not be the lowest coefficients for all other linear regression models. Also, this adds complications when conducting comparison between linear regression and polynomial regression performances due to slightly different parameters.

Figure 21 shows a polynomial regression with favorite_count as dependent variable with four degrees. The test score is the R^2 score, which shows degree 1 and 2 having the highest scores, and degrees 3 and 4 performing much worse. Degree 1 has the highest score which suggests the model is just as good as linear regression.

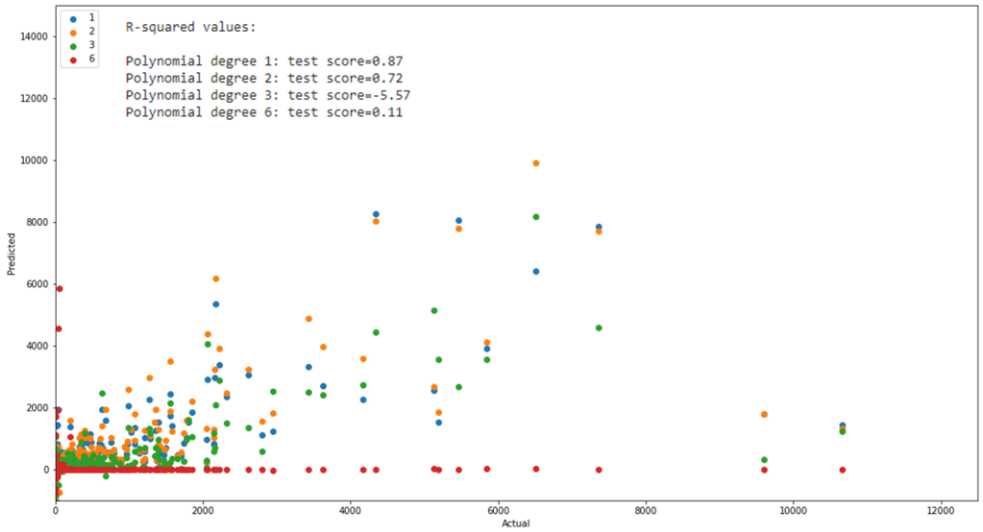

Figure 21 Polynomial regression with favorite_count as dependent variable

Figure 22 shows another polynomial regression with retweet_count as dependent variable. The metric shows degree 1 having the highest at 0.94, and degrees 2 performing much worse. Degree 1 being the highest score also suggests that the model is just as good as linear regression.

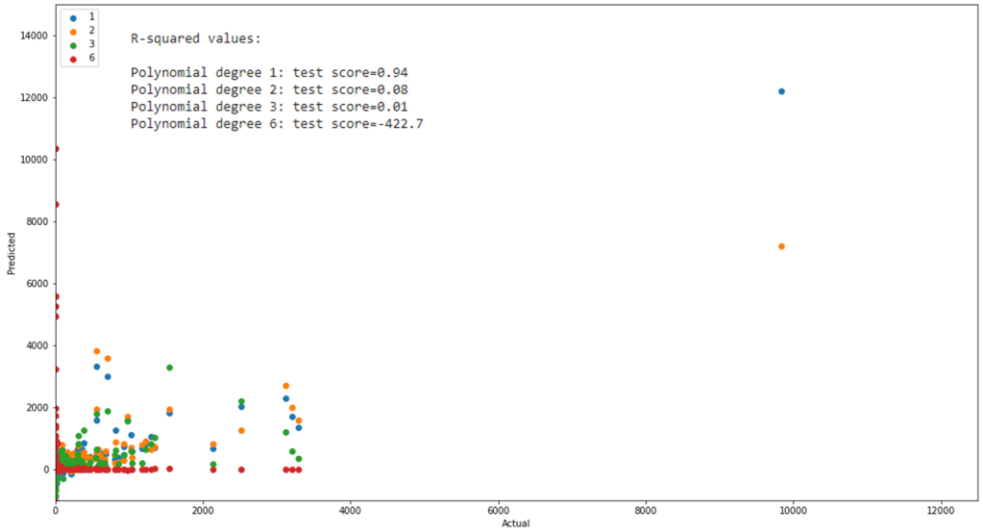

Figure 22 Polynomial regression with retweet_count as dependent variable

Polynomial Regression with *k*-fold

Next, we will run the same polynomial regression with *k*-fold = 5 and four degrees on the same two dependent variables respectively.

Figure 23 shows the polynomial regression with k-fold cross validation on favorite_count as dependent variable, with an average R^2 for 1 degree; the highest of all degrees at $R^2 = 0.73$ compare to 0.87 from single sampling, with an average RMSE of 171.73.

```
Dependent Variable: favorite_count
Independent Variables: ['retweet_count', 'user_followers_count', 'compound', 'user_statuses_count', 'user_verified']
K-fold: 5
Polynomial              Root MSE                        R-Square
=============================================================================================
Degree 1:               176.1003309179579               0.88
Degree 2:               283.8671924780902               0.69
Degree 3:               906.2368606263769               -2.18
Degree 6:               856.3979925552468               -1.84
---------------------------------------------------------------------------------------------
Degree 1:               101.57117876732413              0.71
Degree 2:               113.61553501192776              0.64
Degree 3:               164.8107158447843               0.24
Degree 6:               1067.8173629126154              -30.98
---------------------------------------------------------------------------------------------
Degree 1:               361.7648735168686               0.37
Degree 2:               578.1225176563707               -0.6
Degree 3:               10089.810890024544              -485.89
Degree 6:               38011377.335343964              -6910168447.01
---------------------------------------------------------------------------------------------
Degree 1:               106.07747474685985              0.76
Degree 2:               132.28407898975604              0.63
Degree 3:               429.9725980224988               -2.93
Degree 6:               21654.500707596068              -9971.23
---------------------------------------------------------------------------------------------
Degree 1:               113.14836415856713              0.93
Degree 2:               284.245851156108                0.57
Degree 3:               1548.2663280761865              -11.75
Degree 6:               15713.147572038102              -1312.61
---------------------------------------------------------------------------------------------
Degree  Average RMSE    Average R^2
1        171.732444      0.730000
2        278.427035      0.386000
3        2627.819479     -100.502000
6        7610133.839796 -1382035952.734000
```

Figure 23 Polynomial Regression with k-fold and favorite_count as dependent variable

Figure 24 shows the polynomial regression with *k*-fold cross validation on retweet_count as dependent variable, with an average R^2 for 1 degree—the best of the four degrees at $R^2 = 0.79$ compare to 0.94 from single sampling, with an average RMSE of 43.45.

```
Dependent Variable: retweet_count
Independent Variables: ['favorite_count', 'user_followers_count', 'compound', 'user_statuses_count', 'user_verified']
K-fold: 5
Polynomial              Root MSE                        R-Square
=============================================================================================
Degree 1:               30.619157472439177              0.95
Degree 2:               143.57465611146827              -0.12
Degree 3:               197.92059429558387              -1.13
Degree 6:               140.91449714880363              -0.08
---------------------------------------------------------------------------------------------
Degree 1:               27.778555948274388              0.62
Degree 2:               33.17703268992806               0.46
Degree 3:               44.99600146557491               -0.0
Degree 6:               1595.006459641652               -1256.94
---------------------------------------------------------------------------------------------
Degree 1:               105.01778633851833              0.68
Degree 2:               109.10569803537199              0.66
Degree 3:               1123.4793102354063              -35.19
Degree 6:               185441.21308761396              -986024.03
---------------------------------------------------------------------------------------------
Degree 1:               28.269222892920038              0.73
Degree 2:               47.405288929894574              0.23
Degree 3:               55.005669883911004              -0.04
Degree 6:               10069.950456869094              -34712.13
---------------------------------------------------------------------------------------------
Degree 1:               25.580491200820568              0.96
Degree 2:               73.46282149176497               0.67
Degree 3:               241.73438207411132              -2.53
Degree 6:               5081.564815280062               -1560.37
---------------------------------------------------------------------------------------------
Degree  Average RMSE    Average R^2
1        43.453043       0.788000
2        81.345099       0.380000
3        332.627192      -7.778000
6        40465.729863    -204710.710000
```

Figure 24 Polynomial Regression with k-fold and retweet_count as dependent variable

Comparing Results from Linear Regression with Polynomial Regression

We compare the results between Linear Regression and Polynomial Regression based on the k-fold cross validation.

Linear Regression with *k*-fold cross-validation where favorite_count as dependent variable reports RMSE of 171.28 and $R^2 = 0.73$, whereas Polynomial Regression with 1-degree being the best performer reports RMSE of 171.73 and $R^2 = 0.73$. Since both results are almost identical, therefore linear regression is the preferred model as it is less complex and consumes less computation resources.

Linear Regression with *k*-fold cross-validation where retweet_count as dependent variable reports RMSE of 43.49 and $R^2 = 0.79$, whereas Polynomial Regression with 1-degree being the best performer reports RMSE of 43.45 and $R^2 = 0.79$. Both results are almost identical, therefore linear regression is the preferred model.

On a side note, it should be expected that Polynomial Regression of 1-degree is essentially the same as linear regression. The metrics also suggests Polynomial Regression does not yield a better fit with given parameters.

Future Studies and Conclusion

Future studies can consider other alternative approaches, such as: use classification instead of regression, by categorizing based on the sentiment score to provide a discrete values instead of continuous value; instead modelling regression from -1.00 through 1.00, consider splitting into three sentiment groups—negative, neutral, and positive and and conduct independently study for each category as they might have different patterns or regression fit, for example logarithmic regression might be suitable for negative and positive; group users by screen name to calculate each user's mean sentiment score as a new feature to see whether there are any effects on number of followers and number of favourite tweets, based on the assumption that each user have their own writing styles and their tweets might fall within a certain sentiment score range.

In conclusion, our regression models were able to predict favorite_count and retweet_count to some extent. However, based on observations from sampling the scatterplots, there are reasonable doubts that under count of less than 20 for favorite_count and less than 25 for retweet_count will not yield any meaningful prediction. Favorite_count and retweet_count is highly correlated with each other and has the highest coefficient; significantly contribute to each other's prediction model. Compound feature derived from the sentimentality of Tweets is the second most significant coefficient in both regression models. Based on these reasons, we can confidently say COVID-19 Tweets' Sentimentality does influence Likeability (favorite_count) and Retweet-ability (retweet_count). Although this analysis did not yield us any direct benefits, it increases our understanding on the inadequacies of sentiment analysis tools when used on social media messages. Also, it helps serves as a stepping stone for us to envision other new features that might help us improve our prediction models.

References

Bode, L., & Vraga, E. K. (2017). See Something, Say Something: Correction of Global Health Misinformation on Social Media. *Health Communication, 33*(9), 1131-1140. doi:10.1080/10410236.2017.1331312

Caplan, L. (2017, 10 11). *Should Facebook and Twitter Be Regulated Under the First Amendment?* Retrieved from WIRED: https://www.wired.com/story/should-facebook-and-twitter-be-regulated-under-the-first-amendment/

Carvalho, L. d., Silva, M. V., Costa, T. d., Oliveira, T. E., & Oliveira, G. A. (2020, 9). Public Health and the COVID-19 pandemic: challenges for global health. *Research, Society and Development, 9*(7). doi:10.33448/rsd-v9i7.4188

Culliford, E. (2020, 10 6). *Twitter is testing how its misinformation labels can be more obvious, direct*. Retrieved from Reuters: https://ca.reuters.com/article/us-usa-election-twitter-labels-focus-idCAKBN26R1ML

Deo, G. S., Mishra, A., Jalaluddin, Z. M., & Mahamuni, C. V. (2020). Predictive Analysis of Resource Usage Data in Academic Libraries using the VADER Sentiment Algorithm. *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, 221-228.

Farzan, A. N., Noack, R., Beachum, L., Taylor, A., Iati, M., Bellware, K., . . . Kornfield, M. (2020, 9 21). *CDC reverses statement on airborne transmission of coronavirus, says draft accidentally published*. Retrieved from Washington Post: https://www.washingtonpost.com/nation/2020/09/21/coronavirus-covid-live-updates-us/

Glenski, M., Weninger, T., & Volkova, S. (2018, 11 21). Propagation From Deceptive News Sources Who Shares, How Much, How Evenly, and How Quickly? *IEEE Xplore*, 1071-1082. doi:10.1109/TCSS.2018.2881071

Hinshaw, D. (2020, 10 12). *As Covid Cases Surge, More Public-Health Experts Say Lockdowns Aren't the Answer*. Retrieved from Wall Street Journal: https://www.wsj.com/articles/public-health-experts-rethink-lockdowns-as-covid-cases-surge-11602514769

Kouzy, R., Jaoude, J. A., Kraitem, A., Alam, M. B., Karam, B., Adib, E., . . . Baddour, K. (2020, 3 13). Coronavirus Goes Viral: Quantifying the COVID-19 Misinformation Epidemic on Twitter. *Cureus*. doi:10.7759/cureus.7255

Ontario Government. (2020, 10 1). *Official Government of Ontario Twitter*. Retrieved from Twitter: https://twitter.com/ongov

Oren, E., Martinez, L., Hensley, R. E., Jain, P., Ahmed, T., Purnajo, I., . . . Tsou, M.-H. (2020, 7 29). Twitter Communication During an Outbreak of Hepatitis A in San Diego, 2016–2018. *American Journal of Public Health*. doi:10.2105/AJPH.2020.305900

Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F., & Camacho-Collados, M. (2019, 11 19). Detecting and Monitoring Hate Speech in Twitter. *sensors, 19*(21), 4654. doi:10.3390/s19214654

Ravi, K. (2015, 6). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based System, 89*, 14-46. doi:10.1016/j.knosys.2015.06.015

Sewalk, K. C., Tuli, G., Hswen, Y., Brownstein, J. S., & Hawkins, J. B. (2018, 10). Using Twitter to Examine Web-Based Patient Experience Sentiments in the United States: Longitudinal Study. *Journal of Medical Internet Research, 20*(10). doi:10.2196/10043

Shao, C., Hui, P.-M., Wang, L., Jiang, X., Flammini, A., Menczer, F., & Ciampaglia, G. L. (2018, 4). Anatomy of an online misinformation network. *PLoS One, 13*(4). doi:10.1371/journal.pone.0196087

Sicilia, R., Giudice, S. L., Pei, Y., Pechenizkiy, M., & Soda, P. (2018, 11 15). Twitter rumour detection in the health domain. *Expert Systems With Applications, 110*, 33-40. doi:10.1016/j.eswa.2018.05.019

Vicarioa, M. D., Bessib, A., Zolloa, F., Petronic, F., Scalaa, A., Caldarellia, G., . . . Quattrociocchia, W. (2016, 1 19). The spreading of misinformation online. *Proceedings of the National Academy of Sciences - PNAS, 113*(3), 554-9. doi:10.1073/pnas.1517441113

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, a. R. (2018). Detection and Resolution of Rumours in Social Media: A Survey. *ACM Computing Surveys, 51*(2). doi:10.1145/3161603