CMPT353 Final Report
Sentiment Analysis on COVID-19 Tweet

MinZhi(Chole)Huang
Yuxin(Lacey)Liang

Department of Computing Science, Simon Fraser University, Burnaby, BC, Canada

1 Abstract

This report focuses on sentiment analysis on tweets that have COVID-19 hashtags and obtained between July 25th, 2020, and August 20th, 2020. Tweet data is obtained from Kaggle (Dataset). **Our study aims to analyze how people's emotions changed on Twitter during the pandemic and select the best machine learning model**. Our analyses included Data Overview, Data Cleaning and Processing, Exploratory Data Analysis, and Machine Learning. The analysis results, interpretations, conclusions, and limitations are presented below.

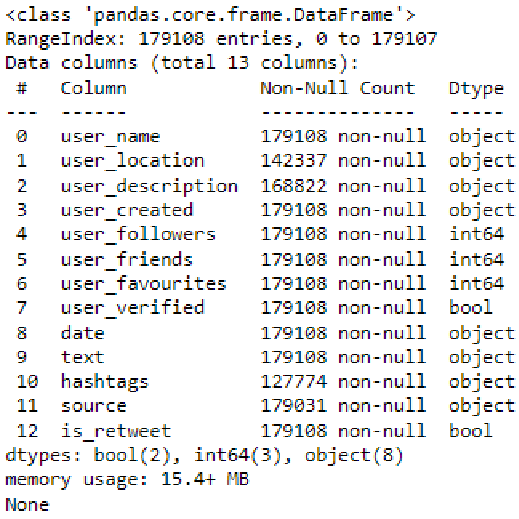
Keywords: COVID-19; Sentiment Analysis; Machine Learning; Tweet

2 Introduction

The COVID-19 pandemic, also known as the coronavirus pandemic, is an ongoing global pandemic of coronavirus disease 2019 (COVID-19). With the increased cases and severity, officials of governments all over the world took imposed the lockdown and strict rules. This has significantly changed the way of people's daily life and made social media become one of the main ways for people to express their feelings and opinions about the virus. News, reports, and papers showed people's levels of stress and anxiety increased during the pandemic [1][2]. To verify if this is true, we applied sentiment analysis to a Kaggle tweets dataset. "Sentiment analysis (also known as opinion mining or emotion AI) is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study effective states and subjective information." [3]. We aimed to see if people's tweet's emotions become more positive or negative during the time period of the data, we also want to analyze the most common hashtags, the most common words mentioned in tweets, and the top locations that people tweet positively/negatively.

3 Data Overview

The original data file consists of one data frame with 179108 rows and 13 columns. Since we do not need all of them to perform analysis, we cleaned 'text' variable and selected 'user name','date','text','hashtags', and 'user location' variables for our visualization, analysis, and machine learning.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 179108 entries, 0 to 179107
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   user_name         179108 non-null  object
 1   user_location     142337 non-null  object
 2   user_description  168822 non-null  object
 3   user_created      179108 non-null  object
 4   user_followers    179108 non-null  int64 
 5   user_friends      179108 non-null  int64 
 6   user_favourites   179108 non-null  int64 
 7   user_verified     179108 non-null  bool  
 8   date              179108 non-null  object
 9   text              179108 non-null  object
 10  hashtags          127774 non-null  object
 11  source            179031 non-null  object
 12  is_retweet        179108 non-null  bool  
dtypes: bool(2), int64(3), object(8)
memory usage: 15.4+ MB
None
```

4 Data Cleaning and Exploring

4.1 Data Cleaning

We cleaned the data to analysis-ready data. The following normalization and cleanings were applied to 'text' in the dataset:

1. Remove website, mentions, hashtags, numbers, stopwords, special characters and punctuation.
2. Convert to lower case
3. Drop NA rows
4. Clean the hashtags, includes remove brackets, convert to lowercase and remove underscore, then filter the text with hashtags with the top 50 most used hashtags
5. Select columns:

Username, Date, Text, Hashtags, User Location, and drop the rest of the columns

4.2 Data Visualization

We conducted the data visualization to understand our dataset better and identify patterns and trends if there is any. The visualizations are as follow :

1. Histogram and word clouds for top hashtags and words use

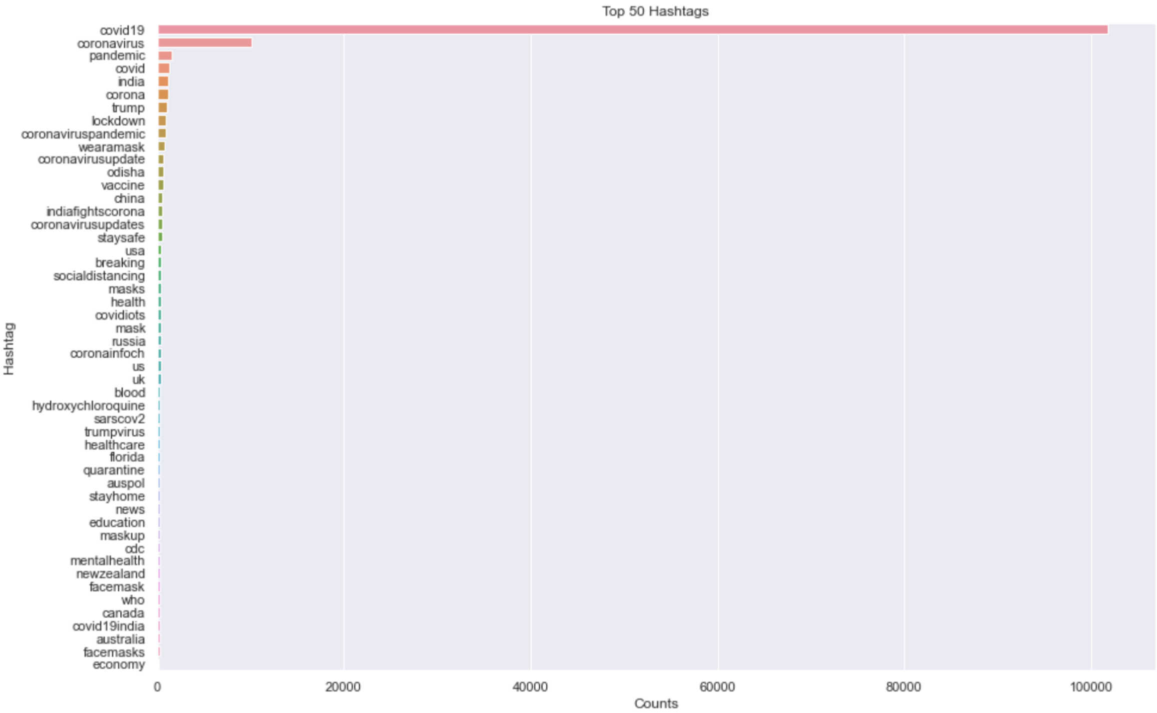
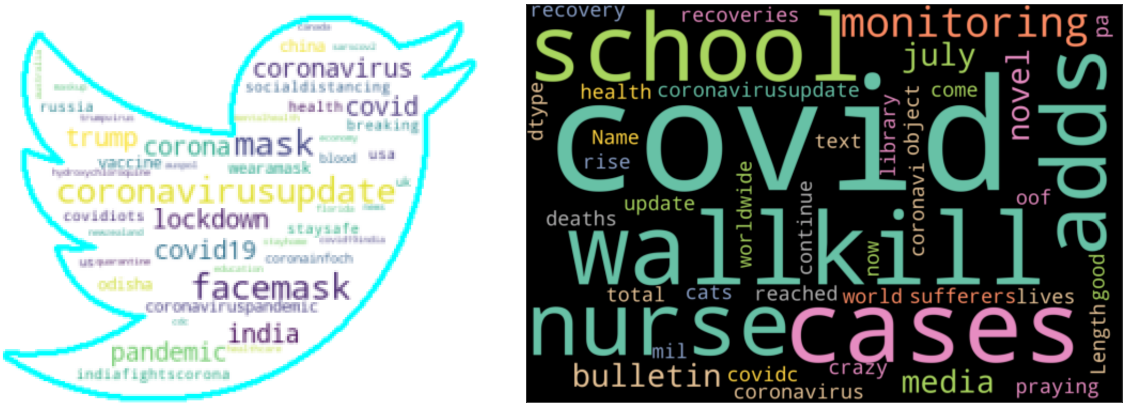


Figure 1: Histogram for top hashtags



(a) Top 100 hashtags

(b) Top 100 words

Figure 2: Word clouds for top hashtags and words used

From the word could visualizations, we see some of the top words for hashtags are covid 19, coronavirus, pandemic, and some of the top words for words used in tweets are covid, nurse, cases, and school. We noticed there are some words in the word could like deaths, sufferers, and praying that might indicate the negative emotions of people.

2. Overall Tweet counts percentage and Daily Tweet counts change

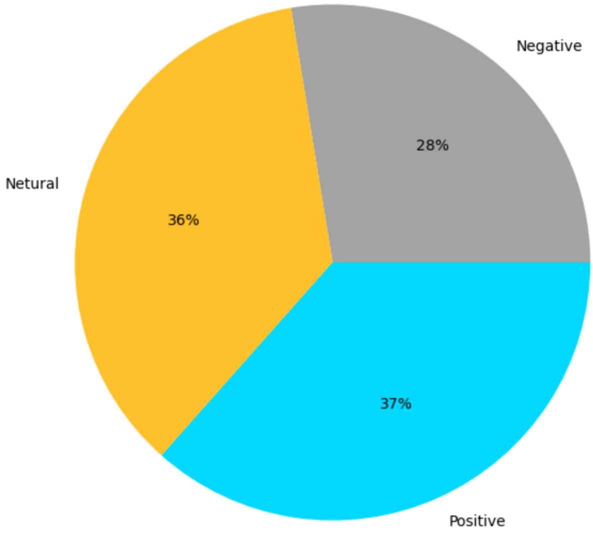


Figure 3: Overall percentage

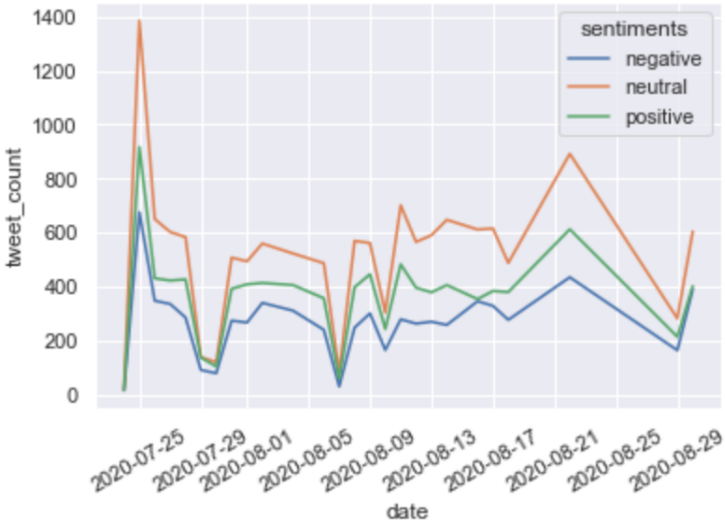


Figure 4: Total counts with time line

From the visualization, we could see the number of negative tweets decrease over time, and the number of positive tweets has a slight increase, which indicates that people are more positive about covid 19 from July 2020 to August 2020. Moreover, we can see that between 2020-08-09 to 2020-8-17, the negative tweets have increased while the positive tweets have decreased. That's might indicate the negative emotions of people.

3. Additional insight: Top 20 countries with high sentiment scores

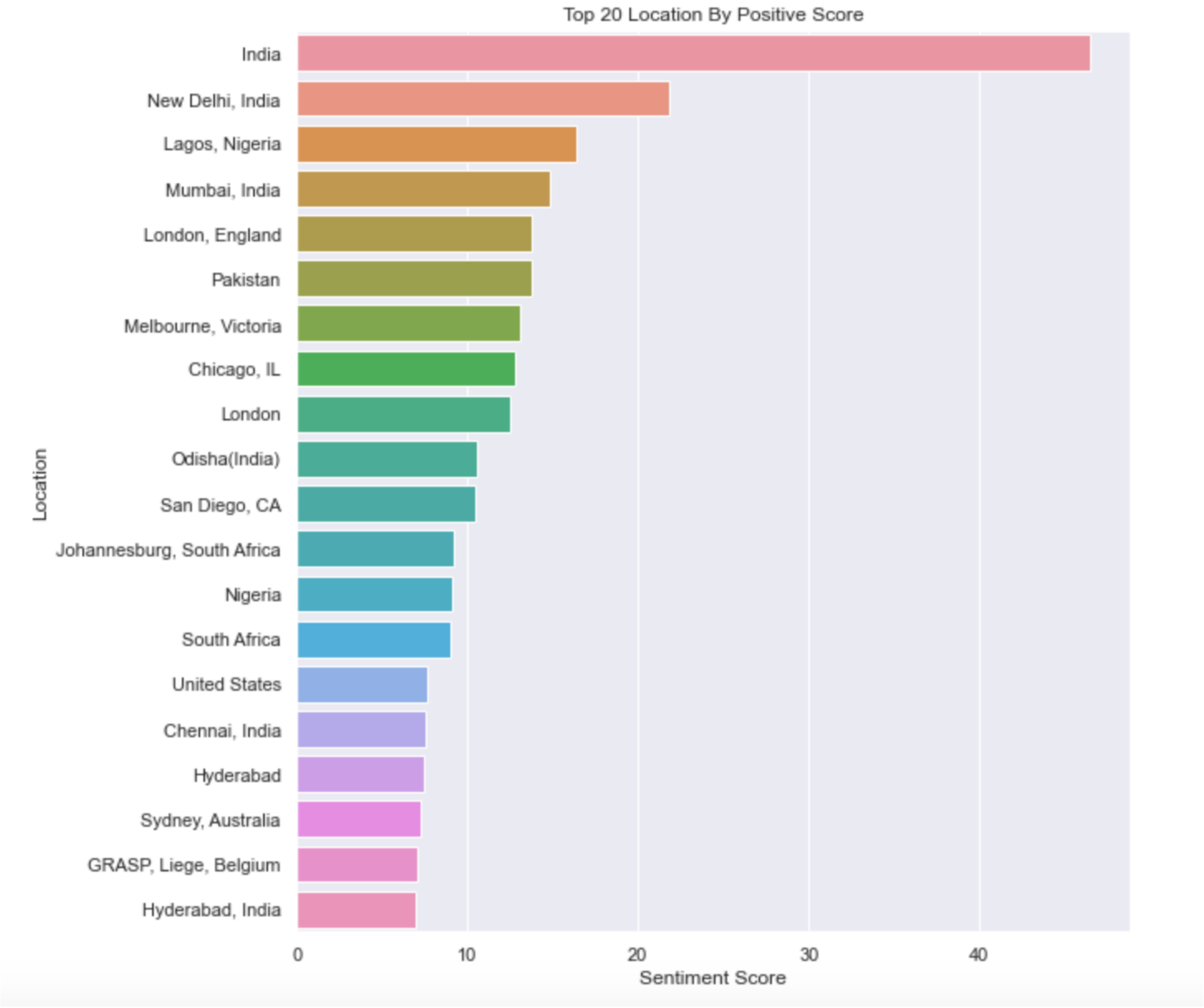


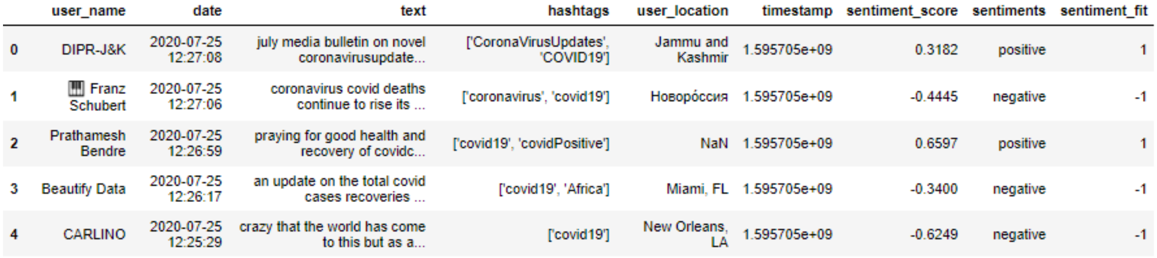
Figure 5: Top 20 Location by positive sentiment score

Besides the analysis scope-focused visualization, we also did an interesting overall data visualization to see the top 20 locations with most positive tweets.

5 Data Analysis

5.1 Sentiment Analysis with NLTK

For sentiment analysis, we install the NLTK, which is a Python open source library that provides a set of diverse natural languages algorithms. We apply the vader sentiment analyzer(SentimentIntensityAnalyzer()) to obtain the polarity scores for each tweet. The SentimentIntensityAnalyzer function will provide a compound score, which is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive).[4] The resulting compound score is stored in our dataset named sentiment_score. Then we also convert the sentiment_score into two categorical variables, one is called 'sentiment', which has string values of positive, negative, or neutral; the other one is called sentiment_fit, which holds numerical values. If the sentiment_score is positive, we assign sentiment value equal to 'positive', sentiment_fit=1 ; if the sentiment_score is negative, we assign sentiment value equal to 'negative', sentiment_fit=-1; if the sentiment_score is zero, we assign sentiment value equal to 'neutral', sentiment_fit=0. we create this numerical variable sentiment_fit for machine learning model fitting which we will show in the next session. Below is dataset we get after cleaning the data and getting the sentiment score.



	user_name	date	text	hashtags	user_location	timestamp	sentiment_score	sentiments	sentiment_fit
0	DIPR-J&K	2020-07-25 12:27:08	july media bulletin on novel coronavirusupdate...	['CoronaVirusUpdates', 'COVID19']	Jammu and Kashmir	1.595705e+09	0.3182	positive	1
1	🎹 Franz Schubert	2020-07-25 12:27:06	coronavirus covid deaths continue to rise its ...	['coronavirus', 'covid19']	Новоро́ссия	1.595705e+09	-0.4445	negative	-1
2	Prathamesh Bendre	2020-07-25 12:26:59	praying for good health and recovery of covidc...	['covid19', 'covidPositive']	NaN	1.595705e+09	0.6597	positive	1
3	Beautify Data	2020-07-25 12:26:17	an update on the total covid cases recoveries ...	['covid19', 'Africa']	Miami, FL	1.595705e+09	-0.3400	negative	-1
4	CARLINO	2020-07-25 12:25:29	crazy that the world has come to this but as a...	['covid19']	New Orleans, LA	1.595705e+09	-0.6249	negative	-1

Figure 6: Dataset after sentiemtn analysis

5.2 Machine learning model training and testing

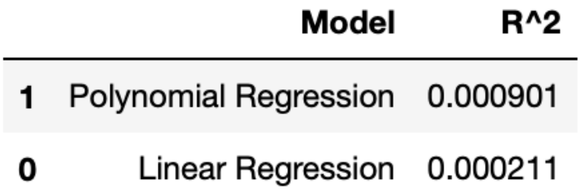
Before performing machine learning model fitting, we also need to convert each tweet to a numerical representation. To achieve this, firstly we break down each tweet into words by using word_tokenize function from nltk library, next, applying lemmaization to reduce words to their base word, then we apply function CountVectorizer() to convert each tweet to a numerical vector, this function will also remove the stopwords, which are considered as noise in the tweet. After these extra cleaning steps, we split the dataset using function train_test_split, and assign the test size to be 40% of the dataset.

For the regression model, we fitted the linear regression model. We used time as our independent variable and sentiment score as a dependent variable.For classification models, we fitted Naive Bayes, K-Nearest Neighbors, Multinomial Naive Bayes, Random Forest, Logistic Regression, and Multi-layer Perceptron. For all the models, we used tweets, which were tokenized, as an independent variable and sentiment category (positive, negative, and neutral) as a dependent variable. For the train test data split, we split 40% of the data into the training set and 60% of the data as a testing set with 42 as random state. We trained the models on the training dataset and evaluated the result in the testing dataset.

5.3 Machine learning model selection

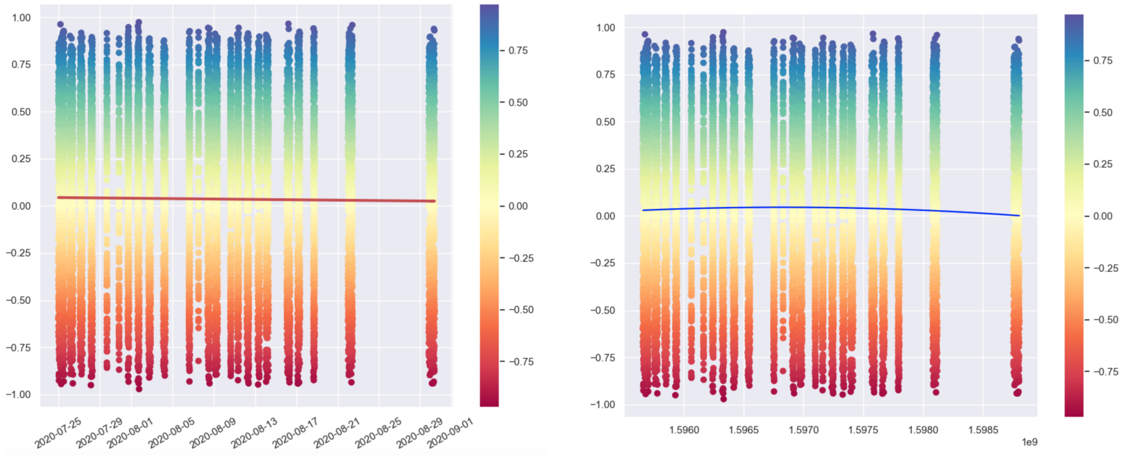
1.Regression Models

We evaluated the performance of regression models with single time r-square.



	Model	R^2
1	Polynomial Regression	0.000901
0	Linear Regression	0.000211

Figure 7: R-square for regression models



(a) Regression model fit

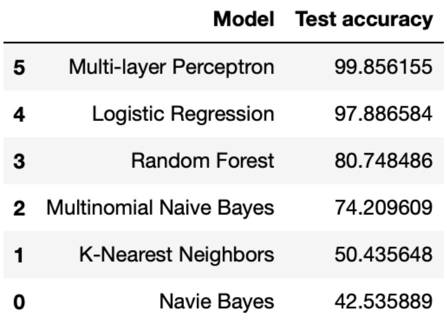
(b) Polynomial model fit

Figure 8: Visualization for model fit

From the score tables, we could see, that for a single time evaluation, the linear regression model had an R-square value of 0.000211, and the polynomial regression had an R-square value of 0.000901. Based on the R-square value, we concluded the polynomial regression model had better performance and would be a better model for the sentiment scores prediction.

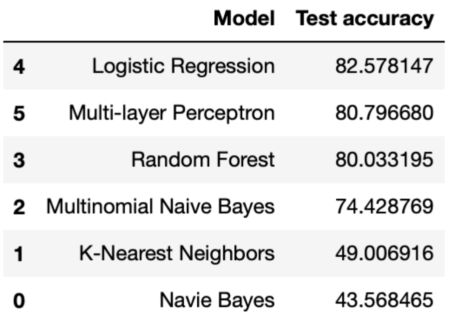
2.Classification Models

We evaluated the performance of classification models with k-folder cross validation (where k=5).



	Model	Test accuracy
5	Multi-layer Perceptron	99.856155
4	Logistic Regression	97.886584
3	Random Forest	80.748486
2	Multinomial Naive Bayes	74.209609
1	K-Nearest Neighbors	50.435648
0	Navie Bayes	42.535889

(a) model scores for single time running



	Model	Test accuracy
4	Logistic Regression	82.578147
5	Multi-layer Perceptron	80.796680
3	Random Forest	80.033195
2	Multinomial Naive Bayes	74.428769
1	K-Nearest Neighbors	49.006916
0	Navie Bayes	43.568465

(b) Mean model scores with 5 folder cross validation

Figure 9: Model mean scores for different classification models

From the score tables, we could see, for a single time evaluation, the Multi-layer Perceptron (neural networks) achieved the best performance (highest model score) and the second one is Logistic Regression. For 5 folders cross-validation, Logistic Regression model achieved the best performance and the second top is Multi-layer Perceptron (neural networks). Based on the evaluation, we decided to choose the logistic regression model as the best prediction model for sentiment categories.

6 Conclusion

The project result is limited by the dataset timeline, a more robust analysis, and accurate conclusion could be obtained by a dataset with a longer timeline (8 months or more). Limitations and possible future works are listed in the section below. However, with our regression model fit visualization, we conclude that people's twitters are more negative about covid 19 from July 2020 to August 2020 (other analyses did not give us much information). Also, based on the regression line, we anticipate that the sentiment scores will be going down. For the models, we concluded polynomial regression is the best-fitted model for sentiment scores (evaluated based on r-square), and the logistic regression model is the best-fitted model for sentiment categories (evaluated based on k-folder cross-validation).

7 Analysis Limitation & Possible Future Work

1. The timeline for this dataset is limited, a wilder range of time dataset could be applied.

2. For the data cleaning part, we could implement spelling checking, and make the word tokenization and lemmatization function more efficient instead of using for loop.

3. The folder number of cross-validation could be increased to obtain a more accurate evaluation, here, due to running time we limited the folder number to five.

4. The interaction between variables could be checked to see if, besides the time/date variable, other variables (location, etc.) should also be included in the model fitting.

5. Parameter tuning could be applied to all the models with parameters, for example, techniques like grid search and cross-validation could be used to find the best parameters.

6. Other model evaluation methods/standards/visualizations, like mean absolute error, root mean squared error and precision-recall curve could be used to obtain a more certain conclusion.

8 Project Experience Summary

Minzhi(Chloe) Huang:
Sentiment Analysis on COVID-19 Tweet

- Created multiple data visualizations which includes word could, overall percentage of pie chart, etc.
- Trained and evaluated the machine learning models by using k-folders cross validation.
- Presented the analysis results and interpretations into report.

Yuxin(Lacey) Liang:
Sentiment Analysis on COVID-19 Tweet

- Perform data cleaning, data tokenization
- Self-learning Sentiment Analysis and use the python library NLTK to implement Sentiment Analysis
- Applied machine learning concepts, using Naive Bayes, K-Nearest Neighbors, Multinomial Naive Bayes,Random Forest, Logistic Regression, and Multi-layer Perception to fit and predict data.

References

[1] J. Alison, Brunier; Carla, Drysdale (2022, March 2).COVID-19 pandemic triggers 25% increase in prevalence of anxiety and depression worldwide.World Health Organization.https://www.who.int/news/item/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide

[2] Salari, N., Hosseinian-Far, A., Jalali, R. et al. Prevalence of stress, anxiety, depression among the general population during the COVID-19 pandemic: a systematic review and meta-analysis. Global Health 16, 57 (2020). https://doi.org/10.1186/s12992-020-00589-w

[3] Sentiment analysis. (2022, April 15). In Wikipedia.https://en.wikipedia.org/wiki/Sentiment_analysis

[4] https://github.com/cjhutto/vaderSentiment/blob/master/README.rstfeatures-and-updates