Analysis of Covid-19 tweet Sentiment

Yuming Chang
Georgia Institute of Technology
Atlanta, USA
ychang394@gatech.edu

1 ABSTRACT

Analyzing the sentiment of people's posts on social media is helpful for understanding the opinion on crucial topics and providing customized services based on those information. It's impossible for people to do sentiment analysis by themselves since the massive amount of data and bias from a different person. In this project, I will introduce a method to convert text data into a numeric matrix using the TF-IDF method. After that, three machine learning and deep learning model are developed: Naive Bayes, XGBoost, Support Vector Machine (SVM), Random Forest, and Convolutional Neural Network (CNN) to compare the performance of each model on the sentiment prediction. And in this project, the Naive Bayes model will serve as the baseline. The webpage for this project is: https://changyming.github.io/8803Project/

2 INTRODUCTION

Social media has played more and more important roles in our daily life. People can freely share their opinions and comments and share real-time information on social media. The government, policymakers, and companies must understand the sentiment of users based on their posts to make better decisions. Because of the free expression on social media, that results in tons of good real-time news and information. Twitter is one of the largest social media, which has a huge amount of active users every day, which means it is important to analyze users' sentiments. Past few years, Covid-19 is a hot-spot topic for the entire society. Covid-19 has also aroused controversy, which caused heated discussions and provided sufficient amounts of data to do sentiment analysis to analyze the opinions on the Covid-19. In recent years, many researchers have put effort into using machine learning models such as Support Vector Machine (SVM), Decision Trees, and CNN to analyze the sentiment of the text, which can also provide useful insights for misinformation detection. In this project, serval machine learning and deep learning models will be implemented for misinformation detection and prediction.

Based on the sentiment analysis. Using the Natural Language Process(NLP) method, sentiment analysis can automatically extract the information from the text and classify the sentiment from those texts[1]. In order to use the machine learning model to analyze and predict sentiments, we need to convert the text into numeric values using Natural Language Process methods. So the problem in this project will be to convert the text information into a numeric value and train and test models using those data.

3 PROBLEM DEFINITION

Using the Natural Language Process(NLP) method, sentiment analysis can automatically extract the information from the text and classify the sentiment from those text[1] . In order to use the machine learning model to do the sentiment analysis, we need to convert the text into numeric values using Natural Language Process methods. So the problem in this project will be to convert the text information into a numeric value and train and test models using those data.

4 CHANGE FROM THE PROPOSAL AND MILESTONE REPORT

Compared with the original project proposal solely focused on general sentiment analysis of Twitter data, the modification is that I will focus on the Covid-19 vaccine misinformation detection. Based on the feedback on the project proposal from Professor, the project needs to be more related to urban computing problems. So in order to achieve this goal, Compared with the Milestone report, I will adopt a new Twitter dataset, which contains tweets on Covid-19 information, to train and test the models I will implement. The detail of the new dataset will be discussed in Section 5. Because the dataset used in the Milestone report required us to use TextBlob for the sentiment analysis myself, which led to the poor performance of each model, So in this final report, I will add a dataset that contains the ground truth label for each covid-19 tweet text. Due to the poor performance of the Graph neural network model, I removed the Graph neural network model and added Random Forest, XGBoost, and Naive Bayes models in the final project.

5 RELATED WORK

In recent years, plenty of researchers have focused on using different methods to analyze the text's sentiment. Dey[2] has developed a Naive Bayes classifier to do the sentiment analysis. Assuming that each word is independent, the Naive Bayes classifier they developed is based on the Bayes theorem by using the frequency of the words and the text category as the feature to build the joint probability. The strength of the naive Bayes classifier is compared with other machine learning models, and the naive Bayes classifier requires fewer data to train. However, the weakness of the naive bayes model they proposed is that in this model, they assume each word appears independently, which in some cases is not valid, and each word may have a different meaning when they appear in the word combination. Astri[3] uses Multinomial Logistic Regression which uses the softmax activation function to do the logistic regression. The strength of this method is that it requires a small amount of data to train the model and does well even though the input features has a relationship of multi-linear. Moreover, the weakness of those methods is that this model assumes the input variable has a linear relationship with each other, which may not be accurate in the text classification. Tripathi[4] develop the random forest model to do the sentiment analysis on the Twitter text. The random forest model produces the output by combining the results of the different

decision trees in the random forest model. Furthermore, in their research, they picked up some features from the dataset as the input feature, then based on the input features selected to determine the best number of nodes in each decision tree and built multiple decision trees. The strength of the random forest model is that this model can often generate good classification results and deal with missing data. However, the weakness of the random forest model is that sometimes they need lots of features to fit the model, but in the text classification task, we need to avoid the overfitting issue during the training period by carefully choosing several features. Also, we need to address that the training time of the random forest model may be longer compared with the primary decision tree model or other machine learning models. Rahman[5] developed the Support Vector Machine (SVM) model to do the sentiment analysis. In their research, they use the Principle Component Analysis method to find the essential features using the matrix build by calculating the weights of different features; then, they use grid research to find the best parameter in the SVM models. The advantage of the SVM model is that when the dataset is highly dimensional, the performance of the SVM model will perform well. However, the weakness of their work is that they use a small number of data to train their model in this research, but the problem is that the SVM model may perform worse when the dataset is too large, so we need to see how the performance of SVM model when dataset become large. Malarvizhi[6] has developed the Convolutional Neural Network (CNN) model to analyze sentiment. They first convert the different lengths of text data into the matrix with the same dimension and pad those matrices together. After that, they put the padded matrix into the embedding layer of the CNN model. After generating the embedding results, the embedding results will go through the convolutional layer to get the classification results. The strength of this method is that the CNN model has multiple layers, which enable the model to capture some vital information from the text. However, the weakness of the CNN model is that train the CNN model needs a large amount of data, and it is time-consuming to train the CNN models. Swathi[7] proposed the Long-short Term Memory (LSTM) networks to investigate the sentiment of the tweet data. After cleaning the tweet text data by removing the special characters, usernames, and hashtags, they put the processed text dataset into the tokenization module and fed those data into the LSTM model they built. The strength of the LSTM model is that the LSTM model can handle the long sequence of text well compared with other models, but the weakness of the LSTM model is that it needs more data to train.There are also some research on Covid-19 information detection. Hayawi[8], et al. collected over 15000 tweets on Covid-19 vaccine and used serval learning methods: XGBoost, LSTM, BERT transformer model to find the best perform model. In their research, they found BERT transformer model have the best performance based on their experiments, which achieve 0.98 F1-score and 0.97 precision. Mulahuwaish[9] et al. also collect 1375592 tweets about the Covid-19 vaccine and build the CNN+Bi-GRU and the performance of the CNN+Bi-GRU is better than Bi-LSTM model based on the data they collected, which achieve over 0.92 accuracy. Reshi[**?**] et al used different Lexicon-Based Methods: TextBlob, Valence Aware Dictionary for Sentiment Reasoning (VADSR), AFINN to generate the sentiment based on tweet dataset about the Covid-19 vaccine and build some learning models to do the sentimal prediction. Based on their results, using TextBlob method to assign the sentiment of tweet data had the best performance.

6 DATA

In this project, COVID-19 Tweets[10] will be used to generate the sentiment, train, validate and test the models built. This dataset was stored in a CSV file—the detail of the dataset provided in Figure 1. From Figure 1, we can know there are 41157 data in this dataset. Moreover, have 6 features in this dataset. However, in this project, I analyze and predict the sentiment of tweets. So we do not need so many features. I will keep OriginalTweet and label Column in this project in this dataset. The distribution of each class of the original class is shown in Figure 2, we can see the original dataset have five labels, extremely negative, negative, neutral, positive, extremely positive. And the class of positive have the most number of data.

```
data
```

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative
...	...	...	...	...	...	...
41152	44951	89903	Wellington City, New Zealand	14-04-2020	Airline pilots offering to stock supermarket s...	Neutral
41153	44952	89904	NaN	14-04-2020	Response to complaint not provided citing COVI...	Extremely Negative
41154	44953	89905	NaN	14-04-2020	You know itÂ s getting tough when @KameronWild...	Positive
41155	44954	89906	NaN	14-04-2020	Is it wrong that the smell of hand sanitizer i...	Neutral
41156	44955	89907	i love you so much || he/him	14-04-2020	@TartiiCat Well new/used Rift S are going for ...	Negative

41157 rows × 6 columns

Figure 1: screenshot of dataset

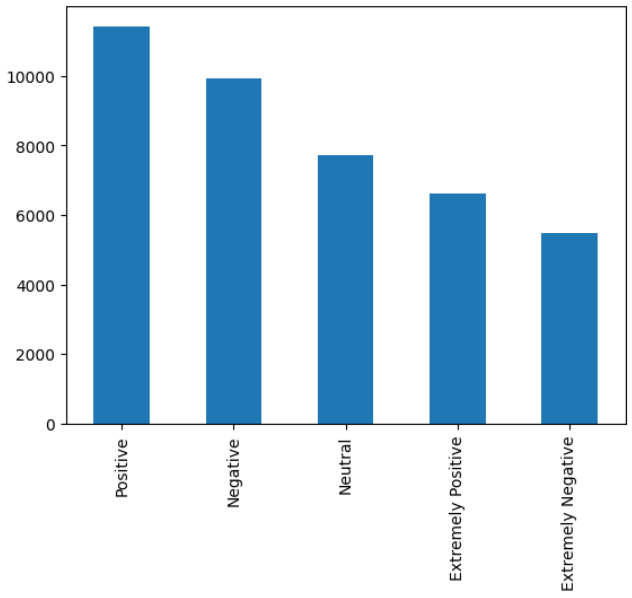


Figure 2: Distribution of each class of original dataset

6.1 Clean the tweets

Figure 1 shows some links, special characters like @ and hashtags, etc. In order to avoid the impacts of those meaningless special characters, we need to remove them from the data process.

6.2 Convert All The Words to Lowercase

Some research has also proved that the case of words may also influence the sentiment analysis results. So to avoid those influences, I am going to convert all the words to lowercase.

6.3 Covert the text label to the numeric value

From the Figure 1, we can see the current label is text, and have many class, so in this project, I will merge those 5 class to 3 classes and convert them into the numeric values: 0: extremely negative, negative, 1: neutral, extremely positive and positive. The reason why I merge the neutral class into the positive class is that I found when I do the classification, the most wrong classified class for class positive in all the model is the class neutral, which means the neutral and positive attitude may very similar. So in order to improve the performance of all the model, I decide to merge neutral and positive class. Also in order to ensure each class have the same amount of data and save the training time, I choose 10000 data from each class to generate the new dataset. The processed dataset and the new distribution of the class is shown in Figure 3 and Figure 4

	OriginalTweet	Sentiment	preprocess_tweet	label
0	#Nigeria to lower petrol pump prices to 130 #n...	Extremely Negative	nigeria to lower petrol pump prices to 130 nai...	0
1	Will a reusable bag ban make grocery store sho...	Negative	will a reusable bag ban make grocery store sho...	0
2	So very day I go to the supermarket is the sam...	Extremely Negative	so very day i go to the supermarket is the sam...	0
3	For those desperately in need of #ToiletPaper,...	Negative	for those desperately in need of toiletpaper t...	0
4	@StopandShop @StopandShop gave associates a 10...	Negative	stopandshop stopandshop gave associates a 10 r...	0
...	...	...	...	...
19995	Smart brands are using cutting-edge methods to...	Positive	smart brands are using cuttingedge methods to ...	1
19996	Went out to our local Edeka supermarket for so...	Positive	went out to our local edeka supermarket for so...	1
19997	Grocery store employee dies after being diagno...	Neutral	grocery store employee dies after being diagno...	1
19998	Spoke to someone in Cameroon and found out tha...	Positive	spoke to someone in cameroon and found out tha...	1
19999	Are you a hospital, school, or other essential...	Extremely Positive	are you a hospital school or other essential b...	1

20000 rows × 4 columns

Figure 3: Screenshot of processed dataset

6.4 Using TF-IDF to convert tweet into vector

In this project, I will build several machine learning and deep learning methods to predict the sentiment of the tweet on the Covid-19 vaccine. There we need to vectorize the tweet text. In the Natural Language Process, TF-IDF is the widely used statistical method. The equation of TF-IDF is shown below. In this project, I will use the TfidfVectorizer function from sklearn to vectorize the tweet text. The code used is shown in Figure 9. From the code, in this TF-IDF function, I currently set the max feature to 1000.

$$TF = \frac{number\_of\_times\_the\_word\_shown\_in\_documents}{number\_of\_words\_in\_documents}$$

$$IDF = log(\frac{number\_of\_documents\_in\_the\_corpus}{number\_of\_documents\_in\_corpus\_contain\_the\_word})$$

$$TF - IDF = TF * IDF$$

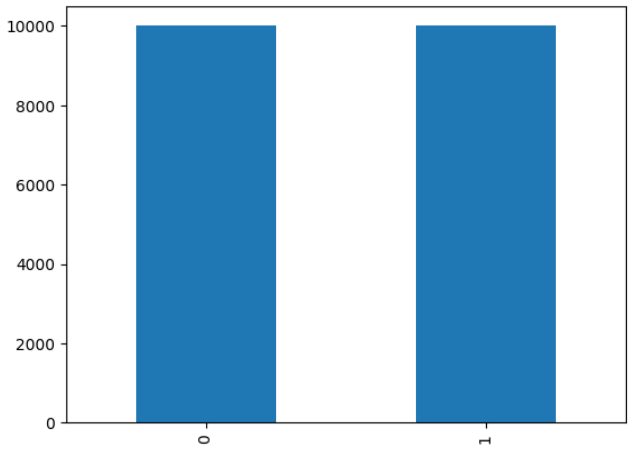


Figure 4: Screenshot of processed dataset

7 MODELS

7.1 SVM model

SVM is one of the supervised learning methods to do classification. By mapping the data to high dimensional space, the algorithm draws the hyperplane to separate different class data. In this project, I will Scikit-learn's SVC model's default hyperparameters and set the random state equal to 42, to train and test the model.

7.2 Random Forest model

Random Forest combine multiple decision tree on sub datasets. In this project I used number of estimator as 100, criterion as Gini impurity and random state=42.

7.3 Naive Bayes model

Based on the Bayes Rule, Naive Bayes model is one of statistical machine learning model to do the classification task. In this project, I am using the default parameters of the fuction of MultinomialNB from Skiti-learn, and set the random state to 42.

7.4 XG Boost model

XG Boost is an emsemble learning method that can handle large dataset and missing values and It is widely used in classification and regression. So in this project, the XG Boost classifier for xgboost library will be used.

7.5 CNN model

Through this milestone report, I have implemented a basic CNN model. CNN model has convolutional layers, which enables it to do text classification. In this project, I used TensorFlow to build the CNN model. The structure of the CNN model is shown in Figure 10. Based on the experiment, I try the different number of convolutional layers in the CNN structure to generate the best results. Based on the experiments, I found when there are 3 convolutional layer and adam optimization function and a 0.0001 learning rate, they produce the best performance. The CNN structure is shown based on Figure 10.

```
CNN_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 1, 3998, 32)       128

 max_pooling2d (MaxPooling2D  (None, 1, 1332, 32)      0
 )

 batch_normalization (BatchN  (None, 1, 1332, 32)      128
 ormalization)

 conv2d_1 (Conv2D)           (None, 1, 1330, 64)       6208

 max_pooling2d_1 (MaxPooling  (None, 1, 443, 64)       0
 2D)

 batch_normalization_1 (Batc  (None, 1, 443, 64)       256
 hNormalization)

 conv2d_2 (Conv2D)           (None, 1, 441, 64)        12352

 flatten (Flatten)           (None, 28224)             0

 dense (Dense)               (None, 2)                 56450

=================================================================
Total params: 75,522
Trainable params: 75,330
Non-trainable params: 192
_________________________________________________________________
```

Figure 5: Best performance CNN model

8 EXPERIMENTS

8.1 Experiments Environment

In this project, all the models are built in the Google Colab with the default environmental of the Google Colab. The library used in this project includes Scikit-Learn, Tensorflow, and xgboost.

8.2 Data Split

In this project, I use Scikit-learn, train_test_split function to split the train, validation, and test data using the ratio 7:1:2, with the random state=42, shuffle=True, and stratify = the label of the dataset to ensure that there are the same amount of data of each class in the training dataset.

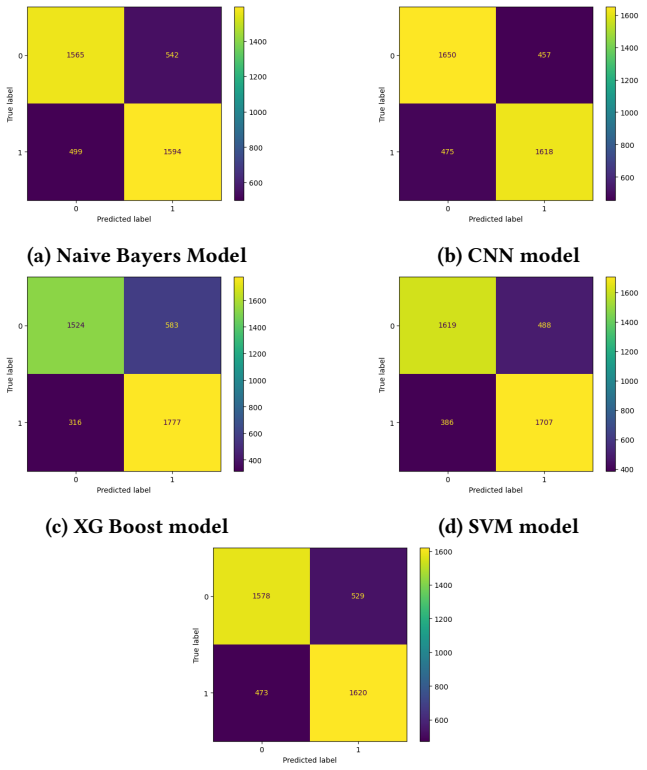
9 EVALUATION METHOD

In this project, the models will be evaluated based on the accuracy and F1 score. And based on the accuracy and F1 score to get the best performance model through the project. The formula of the accuracy and F1 score is shown in Figure 6

Evaluation Method	Equation
Accuracy	$\frac{Total\ Number\ of\ Correct\ Predicton}{All\ Number\ of\ prediction}$
Recall	$\frac{True\ Positive}{True\ Positive + False\ Negitive}$
Precision	$\frac{True\ Positive}{True\ Positive + False\ Postitive}$
F1-score	$2 * \frac{Precision * Recall}{Precision + Recall}$

Figure 6: number of data of each sentiment

Also in the model modification process, the confusion matrix of each model will also be used to evaluate performance of each model



(a) Naive Bayers Model

(b) CNN model

(c) XG Boost model

(d) SVM model

(e) Random Forest Model

Figure 7: Confusion Matrix for each model

10 REUSLTS AND ANALYSIS

The confusion matrix is provided in Figure 7, accuracy, recall, precsion and F1-socre and training time table is provide in Table 8

Performance Comparison

Model	accuracy	precision	recall	F-1 score	Training Time (second)
Naïve Bayes	0.75	0: 0.76 1: 0.75	0: 0.74 1: 0.76	0: 0.75 1: 0.75	0.34
Random Forest	0.76	0: 0.77 1: 0.75	0: 0.76 1:0.77	0: 0.76 1: 0.76	41.47
XG Boost	0.79	0: 0.83 1:0.75	0: 0.72 1:0.85	0:0.77 1: 0.80	118.38
CNN	0.78	0:0.78 1:0.78	0: 0.78 1: 0.77	0: 0.78 1: 0.78	543.86
SVM	0.79	0: 0.81 1: 0.78	0: 0.77 1:0.82	0: 0.79 1: 0.80	1267.70

Figure 8: Table of each performance measure

Figure 7 shows that each model has produced the correct class for each test data. Furthermore, the CNN model has the most significant

number of correct predictions for class 0, which is negative, while the XG Boost model has the smallest number of correct predictions for class 0. However, the XG Boost model has the largest number of correct predictions for class 1, and the Naive Bayes model has the smallest number of correct predictions for class 1. Due to the simplicity of the Naive Bayes model, the training time for Naive Bayes has the lowest training time. In this problem, SVM has the longest training time of all the models. I think this is because the current dimension of the data is 4000, and the number of training data is also large; since the training complexity for SVM is between $O(n^2)$ and $O(n^3)$, the training time required for SVM may very be large. From the accuracy results, we can see that the model's accuracy is better than the baseline model, Naive Bayes, which is 0.75. The recall measures how many positive values are missing in each model; we can see in most models that the recall for class 1 is positive, and neutral is better than class 0, which is negative. This means most models have less missed positive value for class 1. So most of the models perform better in producing the true positive value for class 1. The precision measure ratio between the correct predicted positive and all the predicted positive, in this measurement, we can see the precision for class 0 for almost model is better than class 1, which means, The accuracy of predicting class 0 is better than class 1. For the F-1 score, we can see they are almost aligned with accuracy. Moreover, the SVM and XGBoost have the best performance, 0.79, and the CNN model's accuracy is 0.78.

11 FUTURE WORK

I have transferred the tweet text to the numeric matrix using the TF-IDF method, but there are many other methods to do such a task. I can further explore the different covert methods and see how they will impact the model's performance. Currently, I convert the text data using TF-IDF can feed the original matrix into each model. In the future, I will explore if the normalized and PCA methods can help improve each model's performance. I have built serval basic models using Scikit-learn and Tensorflow. However, more advanced deep learning models, such as CNN-LSTM and Transformer model, need to explore further to see if the performance on this given dataset can be improved.

12 CONCLUSION

In this project, I explore how to convert text data into numeric values using TF-IDF. Furthermore, I tested the performance of serval machine learning models and basic CNN models, such as the Naive Bayes Model, SVM, Random Forest model, XG Boost model, and CNN model, and used Naive Bayes Model as the baseline model. Based on the results, we can see that the accuracy of each model is better than the baseline model. However, the training time for other models is longer than the baseline model. In this project, The XG Boost and SVM have the best prediction accuracy. However, due to the number of training data and the dimension of the dataset, the training time for the SVM is very long. In future research, we need to investigate how different vectorized methods will influence the performance of each model, and if the use normalization method will improve the performance and test more advanced deep learning models such as CNN-LSTM.

REFERENCES

[1] Doaa Mohey Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338, 2018. doi: 10.1016/j.jksues.2016.04.002.

[2] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, and Sweta Tiwari. Sentiment analysis of review datasets using naïve bayes' and k-nn classifier. *International Journal of Information Engineering and Electronic Business*, 8(4):54–62, 2016. doi: 10.5815/ijieeb.2016.04.07.

[3] W.P. Ramadhan, S.T.M.T. Astri Novianty, and S.T.M.T. Casi Setianingsih. Sentiment analysis using multinomial logistic regression. *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, 2017. doi: 10.1109/iccerec.2017.8226700.

[4] Jyotsna Singh and Pradeep Tripathi. Sentiment analysis of twitter data by making use of svm, random forest and decision tree algorithm. *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, 2021. doi: 10.1109/csnt51715.2021.9509679.

[5] Mohammad Rezwanul, Ahmad Ali, and Anika Rahman. Sentiment analysis on twitter data using knn and svm. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017. doi: 10.14569/ijacsa.2017.080603.

[6] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick. Sentiment analysis on twitter data by using convolutional neural network (cnn) and long short term memory (lstm). *Wireless Personal Communications*, 2021. doi: 10.1007/s11277-021-08580-3.

[7] T. Swathi, N. Kasiviswanath, and A. Ananda Rao. An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis. *Applied Intelligence*, 52(12):13675–13688, 2022. doi: 10.1007/s10489-022-03175-2.

[8] K. Hayawi, S. Shahriar, M.A. Serhani, I. Taleb, and S.S. Mathew. Anti-vax: A novel twitter dataset for covid-19 vaccine misinformation detection. *Public Health*, 203: 23–30, 2022. doi: 10.1016/j.puhe.2021.11.022.

[9] K. Gyorick M. Maabreh A. Gupta A. Mulahuwaish, M. Osti and B. Qolomany. "covidmis20: Covid-19 misinformation detection system on twitter tweets using deep learning models. *the 14th International Conference on Intelligent Human Computer Interaction (IHCI-2022)*, 2022.

[10] rajeshmore1. Rajeshmore1/capstone-project-2: Corona virus sentiment analysis.this challenge asks you to build a classification model to predict the sentiment of covid-19 tweets.the tweets have been pulled from twitter and manual tagging has been done then. URL https://github.com/rajeshmore1/Capstone-Project-2.

Milestone Report for Analysis of Covid-19 Vaccine Sentiment

Yuming Chang
Georgia Institute of Technology
Atlanta, USA
ychang394@gatech.edu

1 INTRODUCTION

Social media has played more and more important roles in our daily life. People can freely share their opinions and comments and share real-time information on social media. The government, policymakers, and companies must understand the sentiment of users based on their posts to make better decisions. Because of the free expression on social media, that results in tons of good real-time news and information. Twitter is one of the largest social media, which has a huge amount of active users every day, which means it is important to analyze users' sentiments. Past few years, Covid-19 is a hot-spot topic for the entire society. The Vaccine of Covid-19 has also aroused controversy, which caused heated discussions and provided sufficient amounts of data to do sentiment analysis to analyze the opinions on the Covid-19 Vaccine. In recent years, many researchers have put effort into using machine learning models such as Support Vector Machine (SVM), Decision Trees, and CNN to analyze the sentiment of the text, which can also provide useful insights for misinformation detection. In this project, serval machine learning and deep learning models will be implemented for misinformation detection and prediction.

Based on the sentiment analysis. Using the Natural Language Process(NLP) method, sentiment analysis can automatically extract the information from the text and classify the sentiment from those texts[1]. In order to use the machine learning model to analyze and predict sentiments, we need to convert the text into numeric values using Natural Language Process methods. So the problem in this project will be to convert the text information into a numeric value and train and test models using those data.

2 CHANGE FROM THE PROPOSAL

Compared with the original project proposal solely focused on general sentiment analysis of Twitter data, the modification is that I will focus on the Covid-19 vaccine misinformation detection. Based on the feedback on the project proposal from Professor, the project needs to be more related to urban computing problems. So in order to achieve this goal, I will adopt a new twitter dataset, which contains the tweets on the Covid-19 vaccine information to train and test the models I will implement. The detail of the new dataset will be discussed in Section 5.

3 RELATED WORK

In recent years, plenty of researchers have focused on using different methods to analyze the text's sentiment. Dey[2] has developed a Naive Bayes classifier to do the sentiment analysis. Assuming that each word is independent, the Naive Bayes classifier they developed is based on the Bayes theorem by using the frequency of the words and the text category as the feature to build the joint probability. The strength of the naive Bayes classifier is compared with other machine learning models, and the naive Bayes classifier requires fewer data to train. However, the weakness of the naive bayes model they proposed is that in this model, they assume each word appears independently, which in some cases is not valid, and each word may have a different meaning when they appear in the word combination. Astri[3] uses Multinomial Logistic Regression which uses the softmax activation function to do the logistic regression. The strength of this method is that it requires a small amount of data to train the model and does well even though the input features has a relationship of multi-linear. Moreover, the weakness of those methods is that this model assumes the input variable has a linear relationship with each other, which may not be accurate in the text classification. Tripathi[4] develop the random forest model to do the sentiment analysis on the Twitter text. The random forest model produces the output by combining the results of the different decision trees in the random forest model. Furthermore, in their research, they picked up some features from the dataset as the input feature, then based on the input features selected to determine the best number of nodes in each decision tree and built multiple decision trees. The strength of the random forest model is that this model can often generate good classification results and deal with missing data. However, the weakness of the random forest model is that sometimes they need lots of features to fit the model, but in the text classification task, we need to avoid the overfitting issue during the training period by carefully choosing several features. Also, we need to address that the training time of the random forest model may be longer compared with the primary decision tree model or other machine learning models. Rahman[5] developed the Support Vector Machine (SVM) model to do the sentiment analysis. In their research, they use the Principle Component Analysis method to find the essential features using the matrix build by calculating the weights of different features; then, they use grid research to find the best parameter in the SVM models. The advantage of the SVM model is that when the dataset is highly dimensional, the performance of the SVM model will perform well. However, the weakness of their work is that they use a small number of data to train their model in this research, but the problem is that the SVM model may perform worse when the dataset is too large, so we need to see how the performance of SVM model when dataset become large. Malarvizhi[6] has developed the Convolutional Neural Network (CNN) model to analyze sentiment. They first convert the different lengths of text data into the matrix with the same dimension and pad those matrices together. After that, they put the padded matrix into the embedding layer of the CNN model. After generating the embedding results, the embedding results will go through the convolutional layer to get the classification results. The strength of this method is that the CNN model has multiple layers, which enable the model to capture some vital information from the text. However, the weakness of the CNN model is that train the CNN model

needs a large amount of data, and it is time-consuming to train the CNN models. Swathi[7] proposed the Long-short Term Memory (LSTM) networks to investigate the sentiment of the tweet data. After cleaning the tweet text data by removing the special characters, usernames, and hashtags, they put the processed text dataset into the tokenization module and fed those data into the LSTM model they built. The strength of the LSTM model is that the LSTM model can handle the long sequence of text well compared with other models, but the weakness of the LSTM model is that it needs more data to train.There are also some research on Covid-19 information detection. Hayawi[8], et al. collected over 15000 tweets on Covid-19 vaccine and used serval learning methods: XGBoost, LSTM, BERT transformer model to find the best perform model. In their research, they found BERT transformer model have the best performance based on their experiments, which achieve 0.98 F1-score and 0.97 precision. Mulahuwaish[9] et al. also collect 1375592 tweets about the Covid-19 vaccine and build the CNN+Bi-GRU and the performance of the CNN+Bi-GRU is better than Bi-LSTM model based on the data they collected, which achieve over 0.92 accuracy. Reshi[**?**] et al used different Lexicon-Based Methods: TextBlob, Valence Aware Dictionary for Sentiment Reasoning (VADSR), AFINN to generate the sentiment based on tweet dataset about the Covid-19 vaccine and build some learning models to do the sentimal prediction. Based on their results, using TextBlob method to assign the sentiment of tweet data had the best performance.

4 TEXTBLOB

Based on Reshi[10] et al., the results using the TextBlob method to assign the sentiment have the best performance in their experiments. So in the project, I will also use TextBlob methods to assign the sentiment of the Covid-19 vaccine Twitter.

TextBlob is a lexicon-based method for tasks with raw text data in NLP. Textblob has 2918 lexicons, and they calculate the polarity score based on personal opinions or facts[10]. There is a library in Python, TextBlob can help us to calculate the polarity score based on the TextBlob methods, which return the corresponding scores. In this project, I will use the polarity score to assign the sentiment. The classification criteria are shown in Table 1, and algorithm is shown in Algorithm 1

Polarity Score	Sentiment	Numerical Value
<0	Negative	0
=0	Neutral	1
>0	Positive	2

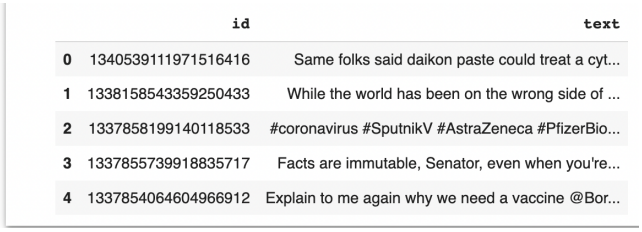
5 DATA

In this project, COVID-19 All Vaccines Tweets[11] will be used to generate the sentiment, train, validate and test the models built. This dataset was stored in a CSV file—the detail of the dataset provided in Figure 1. From Figure 1, we can know there are 228207 data in this dataset. Moreover, have 16 features in this dataset. However, in this project, I analyze and predict the sentiment of tweets. So we do not need so many features. I will keep the id and text columns in this project in this dataset. The processed dataset is shown in Figure 2.

Algorithm 1 Assign Sentiment for tweet data

```
Using TextBlob compute the polarity score of each tweets
for iteration = 1, 2, . . . do
    if polarity score<0 then
        Sentiment = 'negative'
        numeric_value=0
    if polarity score==0 then
        Sentiment = 'neutral'
        numeric_value=1
    if polarity score>0 then
        Sentiment = 'positive'
        numeric_value=2
```

Figure 1: screenshot of dataset



	id	text
0	1340539111971516416	Same folks said daikon paste could treat a cyt...
1	1338158543359250433	While the world has been on the wrong side of ...
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...
3	1337855739918835717	Facts are immutable, Senator, even when you're...
4	1337854064604966912	Explain to me again why we need a vaccine @Bor...

Figure 2: processed datasets

5.1 Clean the tweets

Figure 1 shows some links, special characters like @ and hashtags, etc. In order to avoid the impacts of those meaningless special characters, we need to remove them from the data process. The code in this process is shown in Figure 3, and the results of the processed dataset are shown in Figure 4. From Figure 4, we can see the use of the code described in Figure 3. I have successfully removed all the special characters.

Figure 3: screenshot of code for remove special characters

	id	text	preprocess_tweet
0	1340539111971516416	Same folks said daikon paste could treat a cyt...	Same folks said daikon paste could treat a cyt...
1	1338158543359250433	While the world has been on the wrong side of ...	While the world has been on the wrong side of ...
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	coronavirus SputnikV AstraZeneca PfizerBioNTec...
3	1337855739918835717	Facts are immutable, Senator, even when you're...	Facts are immutable Senator even when youre no...
4	1337854064604966912	Explain to me again why we need a vaccine @Bor...	Explain to me again why we need a vaccine wher...

Figure 4: screenshot of processed dataset

5.2 Convert All The Words to Lowercase

Some research has also proved that the case of words may also influence the sentiment analysis results. So to avoid those influences, I am going to convert all the words to lowercase. The results of the converted dataset are shown in Figure 5. From Figure 5, we can see from the preprocess_tweet column that all the tweet text has been converted into lowercase

	id	text	preprocess_tweet
0	1340539111971516416	Same folks said daikon paste could treat a cyt...	same folks said daikon paste could treat a cyt...
1	1338158543359250433	While the world has been on the wrong side of ...	while the world has been on the wrong side of ...
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	coronavirus sputnikv astrazeneca pfizerbiontec...
3	1337855739918835717	Facts are immutable, Senator, even when you're...	facts are immutable senator even when youre no...
4	1337854064604966912	Explain to me again why we need a vaccine @Bor...	explain to me again why we need a vaccine wher...

Figure 5: screenshot of processed dataset

5.3 Using TextBlob to Generate the Sentiment of Tweet

After obtaining the processed tweet text, we need to use the TextBlob method to assign the sentiment of each Tweet. The code used to generate the sentiment is shown in Figure 6, and the results of processed data are shown in Figure 7. From Figure 7, we can see that the sentiment for each tweet has been successful. After that, I explore the distribution of sentiment in this dataset. The results are shown in Figure 8. The results show that the neutral class has the most number in this dataset, and the positive class has the lowest number.

5.4 Using TF-IDF to convert tweet into vector

In this project, I will build several machine learning and deep learning methods to predict the sentiment of the tweet on the Covid-19 vaccine. There we need to vectorize the tweet text. In the Natural Language Process, TF-IDF is the widely used statistical method. The equation of TF-IDF is shown below. In this project, I will use the TfidfVectorizer function from sklearn to vectorize the tweet text. The code used is shown in Figure 9. From the code, in this TF-IDF function, I currently set the max feature to 1000.

$$TF = \frac{number\_of\_times\_the\_word\_shown\_in\_documents}{number\_of\_words\_in\_documents}$$

$$IDF = log(\frac{number\_of\_documents\_in\_the\_corpus}{number\_of\_documents\_in\_corpus\_contain\_the\_word})$$

$$TF - IDF = TF * IDF$$

```
] from textblob import TextBlob

al_list=[]
nu_list=[]

for i in range(len(data2)):
  sent=TextBlob(data2['preprocess_tweet'].iloc[i])
  polarity = sent.sentiment.polarity

  if polarity < 0:
    al= 'negative'
    nu=0
  if polarity ==0:
    al='neutral'
    nu=1
  if polarity >0:
    al='positive'
    nu=2

  al_list.append(al)
  nu_list.append(nu)

] data2['Sentiment']=al_list

] data2['numeric']=nu_list
```

Figure 6: screenshot of code for assigning sentiment to tweet using TextBlob

```
data2.head(5)
```

	id	text	preprocess_tweet	Sentiment	numeric
0	1340539111971516416	Same folks said daikon paste could treat a cyt...	same folks said daikon paste could treat a cyt...	neutral	1
1	1338158543359250433	While the world has been on the wrong side of ...	while the world has been on the wrong side of ...	negative	0
2	1337858199140118533	#coronavirus #SputnikV #AstraZeneca #PfizerBio...	coronavirus sputnikv astrazeneca pfizerbiontec...	neutral	1
3	1337855739918835717	Facts are immutable, Senator, even when you're...	facts are immutable senator even when youre no...	negative	0
4	1337854064604966912	Explain to me again why we need a vaccine @Bor...	explain to me again why we need a vaccine wher...	neutral	1

Figure 7: screenshot of processed dataset after assigning Sentiment

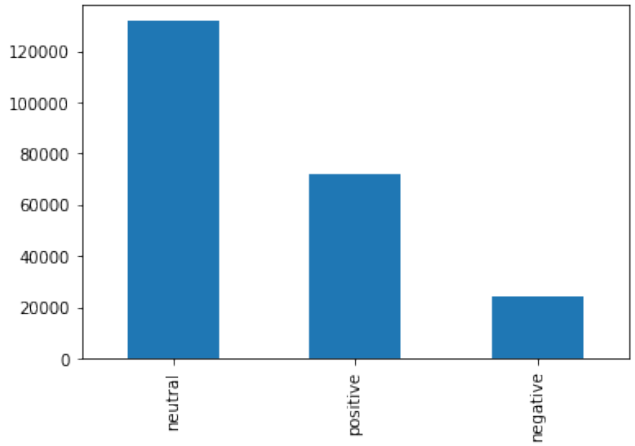


Figure 8: Distribution of each sentiment in this dataset

```
tfidf=TfidfVectorizer(ngram_range=(1, 2),min_df=2,max_features=1000)
tfidf.fit(data2['preprocess_tweet'])
tfidf_df=tfidf.transform(data2['preprocess_tweet']).toarray()
```

Figure 9: TF-IDF Code

6 MODELS

6.1 CNN model

Through this milestone report, I have implemented a basic CNN model. CNN model has convolutional layers, which enables it to do text classification. In this project, I used TensorFlow to build the CNN model. The structure of the CNN model is shown in Figure 10. The CNN model I built currently has 5 convolutional layers based on Figure 10. I also use adam optimization function and a 0.001 learning rate.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 998, 1, 32)        128

max_pooling2d (MaxPooling2D  (None, 998, 1, 32)        0
)

batch_normalization (BatchN  (None, 998, 1, 32)        128
ormalization)

conv2d_1 (Conv2D)            (None, 996, 1, 64)        6208

max_pooling2d_1 (MaxPooling  (None, 996, 1, 64)        0
2D)

batch_normalization_1 (Batc  (None, 996, 1, 64)        256
hNormalization)

conv2d_2 (Conv2D)            (None, 994, 1, 64)        12352

max_pooling2d_2 (MaxPooling  (None, 994, 1, 64)        0
2D)

batch_normalization_2 (Batc  (None, 994, 1, 64)        256
hNormalization)

conv2d_3 (Conv2D)            (None, 992, 1, 64)        12352

max_pooling2d_3 (MaxPooling  (None, 992, 1, 64)        0
2D)

batch_normalization_3 (Batc  (None, 992, 1, 64)        256
hNormalization)

conv2d_4 (Conv2D)            (None, 992, 1, 64)        4160

flatten (Flatten)            (None, 63488)             0

dense (Dense)                (None, 64)                4063296

dense_1 (Dense)              (None, 3)                 195

=================================================================
Total params: 4,099,587
```

Figure 10: CNN model built currently

6.2 Graph Convolutional Network Model

The advantage of the Graph Convolutional Network Model is that it can deal with the structure dataset compared with CNN and RNN. In order to explore how the Graph Convolutional Network performance on this project. I also built a Graph Convolutional Network using Pytorch. The structure is shown in Figure 11. Currently, I used 4 graphs of convolutional layers in the GCN model. The current learning rate is 0.1. The optimization function is Stochastic Gradient Descent (SGD).

7 EVALUATION METHOD

In this project, the models will be evaluated based on the accuracy and F1 score. And based on the accuracy and F1 score to get the best performance model through the project. The formula of the accuracy and F1 score is shown in Figure 5

```
hidden_channels = 16
dropout_probability = 0.5


class GCN(nn.Module):
    def __init__(self, dataset, hidden_channels):
        super(GCN, self).__init__()
        torch.manual_seed(98765)
        self.dataset = dataset

        self.conv1=GCNConv(dataset.num_node_features,64)

        self.conv2=GCNConv(64,32)

        self.conv3=GCNConv(32,hidden_channels)

        self.conv4=GCNConv(hidden_channels,data.num_classes)


    def forward(self, data):
        x, edge_index = data.x, data.edge_index
        x=self.conv1(x,edge_index)
        x=F.relu(x)
        x=F.dropout(x,p=dropout_probability,training=self.training)
        x=self.conv2(x,edge_index)
        x=F.relu(x)
        x=F.dropout(x,p=dropout_probability,training=self.training)
        x=self.conv3(x,edge_index)
        x=F.relu(x)
        x=F.dropout(x,p=dropout_probability,training=self.training)
        x=self.conv4(x,edge_index)
        out=F.log_softmax(x,dim=1)

        return out

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

gcn_model = GCN(data, hidden_channels=hidden_channels)
```

Figure 11: structure of proposed GCN model

Evaluation Method	Equation
Accuracy	$\frac{Total\ Number\ of\ Correct\ Predicton}{All\ Number\ of\ prediction}$
Recall	$\frac{True\ Positive}{True\ Positive + False\ Negitive}$
Precision	$\frac{True\ Positive}{True\ Positive + False\ Postitive}$
F1-score	$2 * \frac{Precision * Recall}{Precision + Recall}$

Figure 12: number of data of each sentiment

Also in the model modification process, the confusion matrix of each model will also be used to evaluate performance of each model

8 FUTURE WORK

Currently, I have built a basic CNN and GCN model. I will build a simple Support Vector Machine(SVM) model using sklearn. Furthermore, train, and test those models using the vector generated using TF-IDF based on tweet texts. Moreover, I will use the Evaluation method discussed above to evaluate the performance of each model. I will also do serval experiments, such as testing different numbers of convolutional layers in the CNN model and different graph convolutional layers in the GCN model. Different learning rates in each model to see the performance of each model and find

the best combination of parameters of each model and compare the performance.

9 DIFFICULTIES IN THIS PROJECT

The first difficulty in this project is that the dataset did not contain the ground truth label. So I need to use the TextBlob method to generate the sentiment of each model. Also, there are other methods in NLP to do sentiment analysis. So It will cost time to find the best methods.

The second difficulty in this project is constructing the vectors based on the tweet text using TF-IDF. How find the best parameters to generate the best vectors to ensure the best performance of each model is also time-consuming.

The third difficulty is that finding the optimal parameters of each model also requires more time.

REFERENCES

[1] Doaa Mohey Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338, 2018. doi: 10.1016/j.jksues.2016.04.002.

[2] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, and Sweta Tiwari. Sentiment analysis of review datasets using naïve bayes‘ and k-nn classifier. *International Journal of Information Engineering and Electronic Business*, 8(4):54–62, 2016. doi: 10.5815/ijieeb.2016.04.07.

[3] W.P. Ramadhan, S.T.M.T. Astri Novianty, and S.T.M.T. Casi Setianingsih. Sentiment analysis using multinomial logistic regression. *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, 2017. doi: 10.1109/iccerec.2017.8226700.

[4] Jyotsna Singh and Pradeep Tripathi. Sentiment analysis of twitter data by making use of svm, random forest and decision tree algorithm. *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, 2021. doi: 10.1109/csnt51715.2021.9509679.

[5] Mohammad Rezwanul, Ahmad Ali, and Anika Rahman. Sentiment analysis on twitter data using knn and svm. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017. doi: 10.14569/ijacsa.2017.080603.

[6] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick. Sentiment analysis on twitter data by using convolutional neural network (cnn) and long short term memory (lstm). *Wireless Personal Communications*, 2021. doi: 10.1007/s11277-021-08580-3.

[7] T. Swathi, N. Kasiviswanath, and A. Ananda Rao. An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis. *Applied Intelligence*, 52(12):13675–13688, 2022. doi: 10.1007/s10489-022-03175-2.

[8] K. Hayawi, S. Shahriar, M.A. Serhani, I. Taleb, and S.S. Mathew. Anti-vax: A novel twitter dataset for covid-19 vaccine misinformation detection. *Public Health*, 203:23–30, 2022. doi: 10.1016/j.puhe.2021.11.022.

[9] K. Gyorick M. Maabreh A. Gupta A. Mulahuwaish, M. Osti and B. Qolomany. “covidmis20: Covid-19 misinformation detection system on twitter tweets using deep learning models. *the 14th International Conference on Intelligent Human Computer Interaction (IHCI-2022)*, 2022.

[10] Aijaz Ahmad Reshi, Furqan Rustam, Wajdi Aljedaani, Shabana Shafi, Abdulaziz Alhossan, Ziyad Alrabiah, Ajaz Ahmad, Hessa Alsuwailem, Thamer A. Almangour, Musaad A. Alshammari, and et al. Covid-19 vaccination-related sentiments analysis: A case study using worldwide twitter dataset. *Healthcare*, 10(3):411, 2022. doi: 10.3390/healthcare10030411.

[11] Amartyanambiar. Covid-19 vaccine sentitmental analysis, Sep 2021. URL https://www.kaggle.com/code/amartyanambiar/covid-19-vaccine-sentitmental-analysis.

Sentiment Analysis of Twitter Post

Yuming Chang
Georgia Institute of Technology
Atlanta, USA
ychang394@gatech.edu

1 INTRODUCTION

In recent years, social media has become increasingly critical to people's lives. People share their life and comments about the movie, services, news, and feelings on social media. It's essential for companies and policymakers to understand the users' sentiments on specific events to enhance the service and provide customized content to serve users and make more profit. So it is essential to do sentiment analysis to achieve those goals. Twitter is one of the largest companies with many active users on their platform, providing sufficient amounts of data to do sentiment analysis. In recent years, many researchers have put effort into using machine learning models such as Support Vector Machine (SVM), Decision Trees, and CNN to analyze the sentiment of the text. In this project, serval machine learning and deep learning models will be implemented for sentiment analysis and prediction.

2 PROBLEM DEFINITION

Using the Natural Language Process(NLP) method, sentiment analysis can automatically extract the information from the text and classify the sentiment from those text[1] . In order to use the machine learning model to do the sentiment analysis, we need to convert the text into numeric values using Natural Language Process methods. So the problem in this project will be to convert the text information into a numeric value and train and test models using those data.

3 RELATED WORK

In recent years, plenty of researchers have focused on using different methods to analyze the text's sentiment. Dey[2] has developed a Naive Bayes classifier to do the sentiment analysis. Assuming that each word is independent, the Naive Bayes classifier they developed is based on the Bayes theorem by using the frequency of the words and the text category as the feature to build the joint probability. The strength of the naive Bayes classifier is compared with other machine learning models, and the naive Bayes classifier requires fewer data to train. However, the weakness of the naive bayes model they proposed is that in this model, they assume each word appears independently, which in some cases is not valid, and each word may have a different meaning when they appear in the word combination. Astri[**?**] uses Multinomial Logistic Regression which uses the softmax activation function to do the logistic regression. The strength of this method is that it requires a small amount of data to train the model and does well even though the input features has a relationship of multi-linear. Moreover, the weakness of those methods is that this model assumes the input variable has a linear relationship with each other, which may not be accurate in the text classification. Tripathi[3] develop the random forest model to do the sentiment analysis on the Twitter text. The random forest model produces the output by combining the results of the different decision trees in the random forest model. Furthermore, in their research, they picked up some features from the dataset as the input feature, then based on the input features selected to determine the best number of nodes in each decision tree and built multiple decision trees. The strength of the random forest model is that this model can often generate good classification results and deal with missing data. However, the weakness of the random forest model is that sometimes they need lots of features to fit the model, but in the text classification task, we need to avoid the overfitting issue during the training period by carefully choosing several features. Also, we need to address that the training time of the random forest model may be longer compared with the primary decision tree model or other machine learning models. Rahman[4] developed the Support Vector Machine (SVM) model to do the sentiment analysis. In their research, they use the Principle Component Analysis method to find the essential features using the matrix build by calculating the weights of different features; then, they use grid research to find the best parameter in the SVM models. The advantage of the SVM model is that when the dataset is highly dimensional, the performance of the SVM model will perform well. However, the weakness of their work is that they use a small number of data to train their model in this research, but the problem is that the SVM model may perform worse when the dataset is too large, so we need to see how the performance of SVM model when dataset become large. Malarvizhi[5] has developed the Convolutional Neural Network (CNN) model to analyze sentiment. They first convert the different lengths of text data into the matrix with the same dimension and pad those matrices together. After that, they put the padded matrix into the embedding layer of the CNN model. After generating the embedding results, the embedding results will go through the convolutional layer to get the classification results. The strength of this method is that the CNN model has multiple layers, which enable the model to capture some vital information from the text. However, the weakness of the CNN model is that train the CNN model needs a large amount of data, and it is time-consuming to train the CNN models. Swathi[6] proposed the Long-short Term Memory (LSTM) networks to investigate the sentiment of the tweet data. After cleaning the tweet text data by removing the special characters, usernames, and hashtags, they put the processed text dataset into the tokenization module and fed those data into the LSTM model they built. The strength of the LSTM model is that the LSTM model can handle the long sequence of text well compared with other models, but the weakness of the LSTM model is that it needs more data to train.

4 MODELS

4.1 SVM

4.1.1 Description. SVM is one of the supervised learning methods to do classification. By mapping the data to high dimensional space, the algorithm draws the hyperplane to separate different class data. In this project, the SVM model will be used as the baseline model.

4.1.2 Implementation Details.

(1) Covert text data into the matrix using the TF-IDF method
(2) Separates the total data into training, validation, and test data with the ratio: 7:1:2
(3) Built SVM model using Scikit-learn
(4) Train the SVM model using training data
(5) Use 5 fold cross-validation to evaluate the model
(6) Using test data to evaluate the performance of the SVM model
(7) Generate the classification report and confusion matrix to evaluate the performance of the model

4.2 CNN model

4.2.1 Description. CNN is the deep learning method model, it has been wildly used in the image classification area, but due to the convolutional layer of the CNN model, we can also use the CNN model to do text classification in the natural language process area. In order to feed data into the CNN model, we need to transform the text data into the matrix. The detailed structure of the CNN model shows in Figure 1. In this project, I am going to use the term frequency-inverse document frequency method to build the vectorized data, then feed the vectorized data into the CNN model. The classification output will be produced after the fully connected layer.

4.2.2 Implementation Details.

(1) Covert text data into the matrix using the TF-IDF method.
(2) Separates the total data into training, validation, and test data with a ratio: 7:1:2.
(3) Built CNN model using TensorFlow.
(4) Trains the CNN model using training data.
(5) Using test data to evaluate the performance of the CNN model.
(6) Changes the number of convolutional layers to investigate the model's performance with different layers and get the training time of each CNN model.
(7) Tuning the hyperparameters of the CNN model, such as the learning rate and optimization function, to find the optimal hyperparameters.
(8) Based on the training time and accuracy to find the best combination of the hyperparameters and the number of convolutional layers.
(9) Generate the classification report and confusion matrix to evaluate the model's performance.
(10) Error case analysis.

4.3 Graph Convolutional Network Model

4.3.1 Description. In recent years, more and more graph or network-based data have been published, such as road network, profile, and

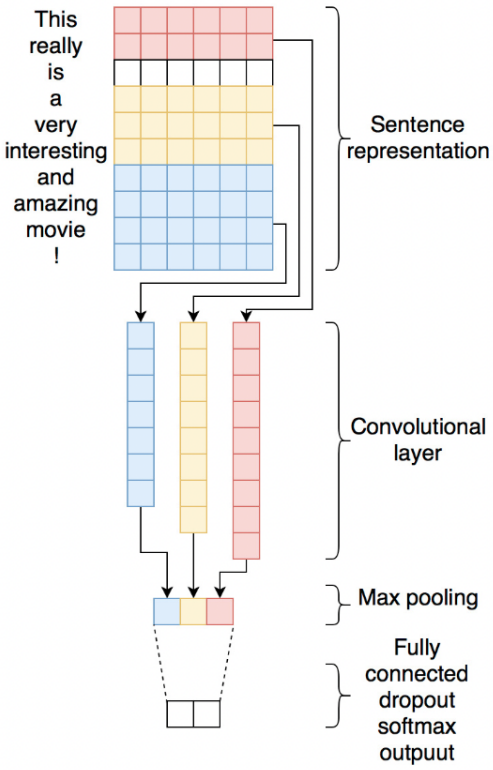


Figure 1: structure of proposed CNN model[7]

recommend systems data. The traditional deep learning method have good test performance on the traditional dataset. However the performance of those traditional methods on graph-based or network-based dataset may not be promising. So Graph Convolutional Network(GCN) can help us to overcome the shortcomings of CNN or RNN. GCN model can help us learn the features of graph and network data through its hidden layers. The GCN model structure in this project shows in Figure 2. We also need to construct the text-graph from the original Twitter post. The detail on how to construct the text-graph is in section 4.3.2

4.3.2 Implementation Details.

(1) Covert text data into the matrix using the TF-IDF method.
(2) Compute the cosine similarity between each pair of the vectorized data
(3) Using the networkx to build a graph based on the cosine similarity. In this graph, each post is a node.
(4) Built GCN model.
(5) trains the GCN model using training data.
(6) using test data to evaluate the performance of the GCN model.
(7) based on the training time and accuracy to find the best combination of the hyperparameters and the number of hidden layers.
(8) Generate the classification report and confusion matrix to evaluate the model's performance.

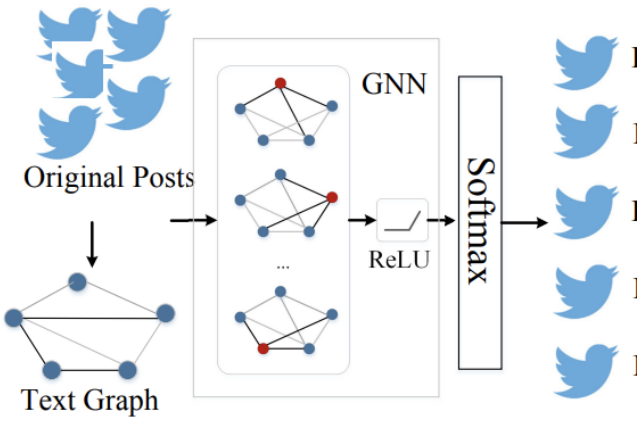


Figure 2: structure of proposed GCN model[8]

(9) Error case analysis.

5 PLATFORM

In this project, each model will be implemented and tested in Google Colab using Python, with Pandas, Numpy, Sciti-learn, and Tensorflow, Pytorch with the Google Colab default environments.

6 DATA

In this project, Sentiment140 dataset[9] will be used to train, validate and test the models built. This dataset was stored in a CSV file—the detail of the dataset provided in Figure 3. From Figure 3, we can know there are 1600000 data in this dataset. Moreover, have 6 features in this dataset, which are sentiment, ids, date, flag, user, and text. In this dataset, the sentiment information is already be stored as numeric values (0-negative, 4-positive). The distribution of data based on the sentiment information is shown in Figure. We can see this dataset contains 80000 negative data and 80000 positive data. From Figure 4, some space characters, website information, and hashtags need to be removed from the text to improve the performance of models. We also need to convert all the Uppercase characters to lowercase characters. In this project, the whole dataset will be split into training, validation, and test dataset using the ratio 7:2:1.

	sentiment	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
...	...	...	...	...	...	...
1599995	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...
1599996	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com - Very cool to hear old Walt interv...
1599997	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you ready for your MoJo Makeover? Ask me f...
1599998	4	2193602064	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	tinydiamondz	Happy 38th Birthday to my boo of alll time!!! ...
1599999	4	2193602129	Tue Jun 16 08:40:50 PDT 2009	NO_QUERY	RyanTrevMorris	happy #charitytuesday @theNSPCC @SparksCharity...

1600000 rows × 6 columns

Figure 3: screenshot of dataset

```
data['sentiment'].value_counts()

0    800000
4    800000
Name: sentiment, dtype: int64
```

Figure 4: number of data of each sentiment

7 EVALUATION METHOD

In this project, the models will be evaluated based on the accuracy and F1 score. And based on the accuracy and F1 score to get the best performance model through the project. The formula of the accuracy and F1 score is shown in Figure 5

Evaluation Method	Equation
Accuracy	$\frac{Total\ Number\ of\ Correct\ Predicton}{All\ Number\ of\ prediction}$
Recall	$\frac{True\ Positive}{True\ Positive + False\ Negitive}$
Precision	$\frac{True\ Positive}{True\ Positive + False\ Postitive}$
F1-score	$2 * \frac{Precision * Recall}{Precision + Recall}$

Figure 5: number of data of each sentiment

Also in the model modification process, the confusion matrix of each model will also be used to evaluate performance of each model

8 CONTRIBUTION

Data Cleaning	Yuming Chang
Data Processing	Yuming Chang
Build Models	Yuming Chang
Evaluate Models	Yuming Chang
Modify Models	Yuming Chang

9 TIMELINE

02/23/2023-02/28/2023	Data Cleaning and processing
03/01/2023-03/15/2023	Build models
03/16/2023-03/23/2023	Model training and evaluation
03/24/2023-04/02/2023	Model modification and final report

10 CONCLUSION

In the project, I am going to implement three machine learning and deep learning models: SVM, CNN, and GCN to do the sentiment analysis and compare their performance and improve the performance of each model by tuning hyperparameters of each model based on the evaluation results to understand each models' weakness and strength better. By the end of this project, I am supposed to find the best model to do the sentiment analysis on the given Twitter text dataset.

REFERENCES

[1] Doaa Mohey Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338, 2018. doi: 10.1016/j.jksues.2016.04.002.

[2] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, and Sweta Tiwari. Sentiment analysis of review datasets using naïve bayes' and k-nn classifier. *International Journal of Information Engineering and Electronic Business*, 8(4):54–62, 2016. doi: 10.5815/ijieeb.2016.04.07.

[3] Jyotsna Singh and Pradeep Tripathi. Sentiment analysis of twitter data by making use of svm, random forest and decision tree algorithm. *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, 2021. doi: 10.1109/csnt51715.2021.9509679.

[4] Mohammad Rezwanul, Ahmad Ali, and Anika Rahman. Sentiment analysis on twitter data using knn and svm. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017. doi: 10.14569/ijacsa.2017.080603.

[5] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick. Sentiment analysis on twitter data by using convolutional neural network (cnn) and long short term memory (lstm). *Wireless Personal Communications*, 2021. doi: 10.1007/s11277-021-08580-3.

[6] T. Swathi, N. Kasiviswanath, and A. Ananda Rao. An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis. *Applied Intelligence*, 52(12):13675–13688, 2022. doi: 10.1007/s10489-022-03175-2.

[7] Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng. Cnn for situations understanding based on sentiment analysis of twitter data. *Procedia Computer Science*, 111:376–381, 2017. doi: 10.1016/j.procs.2017.06.037.

[8] Yuqing Yang. Covid-19 fake news detection via graph neural networks in social media. *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2021. doi: 10.1109/bibm52615.2021.9669662.

[9] KazAnova. Sentiment140 dataset with 1.6 million tweets, Sep 2017. URL https://www.kaggle.com/datasets/kazanova/sentiment140?resource=download.