Assignment 2 Group: 98

Tutors: Tahsin Samia, Nguyen Tung Anh

Group members: Jiajie Wu (490484243), Ruochen Pi (500055496)

Assignment cover sheet

Contribution of each member:

Name	Group work	Code part	Report parts
Jiajie Wu	Sort out the task quantity and division of labor	1) Data preprocessing (lowercase, replace url, replace Emojis, Lemmatization) 2) Word vector (text encoding, FastText, word2vec, etc.) 3) Modeling, parameters fine-tuning and output of a method.	1) Beginning, introduction and above code part description 2) Code operation manual
Ruochen Pi	Develop Gantt charts and define cooperation plans	1) Modeling and result output of the two methods 2) Model evaluation and model performance of all methods	1) Results in the display, discussion, summary writing + description of the above code part 2) References

1. Abstract

This report explores a sentiment analysis on the Sentiment 140[1] dataset with 1.6 million tweets by the logistic regression model, random forest model and Gated recurrent units model. 1,600,000 tweets with 2 classes were used, its labels are positive or negative. Some methods of preprocessing for tweets like removing URL and replacing Emojis to words also were used. The evaluation metrics for this task are accuracy, precision, recall, and confusion matrix. In the end, the result of this experiment was shown with figures by comparison of different methods. The accuracy of the logistic regression model achieves 82.5% in the test set. The accuracy of the random forest method and GRU are 75% and 76%.

2. Introduction

In this task, we aim to study sentiment analysis with Sentiment140 dataset.
People always publish their opinions on the Internet, and that information are numerous and messy, but they are very important because those opinions represent individual feelings and emotions. Analysing opinions is often called sentiment analysis, which in computers is a form of natural language processing. Sentiment analysis has many applications. Such as, when companies or government departments collect a large number of people's feedback on social media, they will interpret and analyze the content, but the problem is that it would take time for staff to read all the available comments at once. Therefore, the task can be solved through natural language processing. The publisher's needs would be identified quickly and efficiently. And those results of sentiment analysis can apply on marketing or after-sales service etc. Therefore, the task of sentiment analysis plays a crucial role.

In this study, we used the Sentiment140 dataset. It includes 1.6 million tweets extracted from Twitter. Each tweet has six fields, including emotion value label (positive or negative), tweet ID, tweet date, tweet logo, sending user, and tweet text. Logistic regression, random forest in machine learning, and GRU model were used to solve the problem of judging the emotional value of tweets. In order to suit the input of the model, we cleaned the data and converted these into the numeric format. The pre-processing contains lowercase, process the url, replace Emojis with phases and lemmatization etc.

For the performance of this task, we used evaluation metrics like accuracy, precision, recall and confusion matrix. The best accuracy of the study is 82.5% by logistic regression algorithm in testing. The second one is GRU model, which achieved 76%. And the Random Forest model is the last one and its accurcy is 75%.

3. Previous work

In this section we provide relevant background on previous work and other methods and their performances on sentiment analysis. Sentiment analysis is a task in the field of natural language processing. More than a decade has passed since the pioneering work of Go et al. [5] They were one of the first researchers to try to classify emotions in twitter messages. Their approach is based on so-called noise labels. They considered using emoticons to train machine learning algorithms, obtained promising results, and opened the way for many other people with similar research interests. Nowadays, there are many processing methods for sentiment analysis, among which the advanced transform and Bert deep learning methods are popular. However, there are still many machine learning methods to solve sentiment analysis problems.

Thejaswini N et al.(2022) worked on twitter sentimental analysis using rule dased and machine learning method. They employed naive Bayes algorithm ,support vector machine algorithm, logistic regression and evaluating each model using accuracy score and f1 score. Their study result is that the best algorithm to fit is logistic regression than SVM, naive Bayes. Their logistic regression designed algorithm achieved 0.83 accuracy. [6]

Ankit et al.(2018) also studied the ensemble classification systems for Twitter sentiment analysis. In their experiment's result, the performance of the Random Forest methods on Twitter Sentiment Analysis Dataset is 70.61%. The accuracy of the Logistic Regression is 73.44%. [7]

For the GRU model, Mircea Moca et al.(2021) solved the sentiment analysis by using GRU. Their solution first converts text into a digital representation. Then, the input is preprocessed. The information is passed to one or more loop layers for processing. The attention mechanism is then optionally applied. The final classification is completed by a feedforward layer, which outputs the score of each considered emotion, positive or negative, and the larger score determines the category. In the end, the accuracy of GRU in their task achieved 79.81%.[8]

After our group completed the coding work, we collected relevant content information from various forum sites, paper sites, and libraries. The three methods used by the group in this study found other users on Kaggle. Our group selected several other researchers' codes for analysis and found that BENOY[2] used logistic regression methods with 80% accuracy in their study, but they didn't use preprocessing, or in other words, handled it very simply. They used the CountVectorizer function, which comes with a function that has word separation and lowercase alphabetization, and then transformed the text into the one-hot format.

In this regard, our group members were puzzled that using the same algorithm, the data without pre-processing was more accurate than the pre-processed data as training data. So, our group checked the code several times and found that it was not converting the word numbers into word embedding.

In addition, our group also studied the method used by RYUJIN_AM in the code, they used random forest and used a multi-layer preprocessing method.[3] We were inspired by their research after studying it. At first, the group used KNeighborsClassifier, SVM, etc., and found that for natural language processing, the running time of KNeighborsClassifier was very long, and finally, KNeighborsClassifier was replaced by the random forest method.

When we browsed the forum and chose to preprocess content and directions, inspired by help.sentiment140. In help.sentiment140, it is mentioned that in sentiment analysis, it is relatively safe to deal with negation and multiple negations.[4] There are also technical difficulties in the detection of sarcastic sentences. Of course, there are also problems such as case conversion and emoji processing. After discussing, the group decided to use lowercase, replace url, replace Emojis, n't convert to not, and Lemmatization for the text processing.

4. Methods

In this task, group 98 choose two machine learning classifiers, which are Logistic Regression, and Random Forest, and one deep learning method, which is the Gated recurrent units model.

4.0 Pre-processing

Twitter text data extracted from social media platforms are large and disorganized, which contain unnecessary data. Therefore, preprocessing text data is an important step. Data preprocessing can impact the performance of the learning models. An effective pre-processing can reduce the size of the featured set extracted from the dataset from 30% to 50%, and leave behind only significant features which are highly correlated with the target value. [9] There are many methods to preprocess text data, such as removing numbers, usernames, punctuations, stop words, lower case transformation, and stemming etc. But not all methods could be effective. Some methods even could reduce the accuracy of the model. Therefore, selecting the preprocessing methods is essential. In this case, we use five preprocessing steps to clean the text data. In this task, we did not remove the stop word because the sentence is short and it will remove some useful information after removing stopword.

Lower Case Transformation

The text tweets are converted to lower case after lower case transformation. Because the model are case sensitive. One word with the uppercase or lowercase will be treated as different words. And the calculation of the frequency of words affects the results of the model. Therefore, lowercase transformation is a necessary step.

Replace url

We consider the url is unless information and it will affect the sentence meaning. In fact, we did a comparative experiment and we found that the results were almost the same with or without the url. Therefore, in order to reduce training time , we processed the url.

Replace Emojis

This part is very important, many NLP tasks would remove punctuation but it also would remove the Emojis. We always chat with Emojis online and those Emojis can express our feelings. Therefore, we keep Emojis in the sentence and transfer them into words so that the model could identify their meanings.

n't convert to not

To some extent, ‘not’ can indicate our negative attitude. So we converted the n’t abbreviation to a full 'not' and kept it in the text.

Lemmatization

Lemmatization involves the conversion of words into their root forms by deleting affixes from the words. This step is similar to the lowercase step, the same word with different tenses. By the process of Lemmatization, the complexity of the feature would be reduced, which increases the learning ability of the classifier.

Term Frequency Inverse Document Frequency (TF-IDF)

TF-IDF is a feature extraction technique and widely used for Text analysis and music information retrieval. In TF-IDF, each term in the document is weighted according to its term frequency (TF) and inverse document frequency (IDF)[10].

$$W_{i,j} = TF_{i,j}(\frac{N}{D_{f,t}})$$

N is the total number of documents in the corpus, Df,t is the number of doc- uments containing the term t, and TFi,j is the number of occurrences of term t in a document d.

4.1 Logistic Regression

4.1.1The theory behind the technology

In this study, the group firstly selected the Logistic regression (LR) model, which is a generalized Linear Model (GLM) and belongs to supervised learning algorithm, requiring labeling data. Specifically, this is a linear model derived from the exponential distribution family. It can be used for regression, dichotomy and multiple classification, and dichotomy is the most common. Logistic regression, Y | X are assumed to be released Bernoulli. Bernoulli publishing is also a 0-1 distribution or two-point publishing, which is a discrete probability publishing. This question mainly focuses on judging the emotion value of Twitter, 0 or 4, which also belongs to binary distribution.

4.1.2 Reasons for choosing this algorithm

The team used logistic regression, although considering that the model was prone to underfitting and could not handle a large number of data with multiple features or variables. However, it is simple to implement and has a very small amount of computation, high speed and low storage resource occupation. Moreover, it can solve the multicollinearity problem well for logistic regression.

4.2 Random Forest

4.2.1The theory behind the technology

Random Forest is an extension of bagging algorithm. A random forest is constructed from decision tree algorithms. It combines many classifiers to provide solutions to both classification and regression problems.[11,12]

The main process of random forest is to conduct training set by taking repeated samples randomly from original data and select features from each training set. Base on those features, create decision trees and save all the predictions after predicting the result by each decision tree. Lastly, select the best decision tree to the final model by voting.

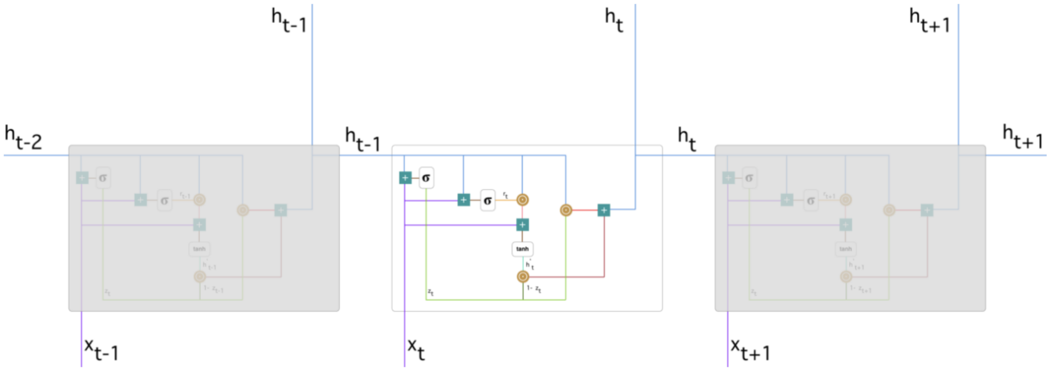
4.2.2Reasons for choosing this algorithm

In this study, we had 160,000 training data. Compared with other machine learning algorithms, random forest has great advantages in processing large amounts of data. In addition, it can handle high-dimensional data well without feature selection. Random forest uses unbiased estimation for generlization error and has strong model generalization ability. Most importantly, the training is fast and easy to implement.

4.3 Gated recurrent units model

4.3.1 The theory behind the technology

Gated Recursive unit (GRU), is an improved and upgraded version of RNN hidden layer method. It can capture remote connections better, and it can effectively solve the vanishing gradient problem. Like LSTM, it is also proposed to solve the gradient problems in long-term Memory and back propagation. [12] Its input and output structure is the same as that of ordinary RNN. In order to solve the problem of vanishing gradient of standard RNN, Gru uses the so-called update and reset gate. Therefore, they can be trained to retain information long ago without changing over time or deleting information unrelated to prediction.



Recurrent neural network with Gated Recurrent Unit

Figure 4.2.2 the structure of GRU[15]

The Reset Gate is responsible for the short-term memory of the network, This is the equation of the reset gate:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

Update gate for long-term memory and the equation:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

Update gate helps the model determine how much information from the past needs to be transferred to the future.

4.3.2 Reasons for choosing this algorithm

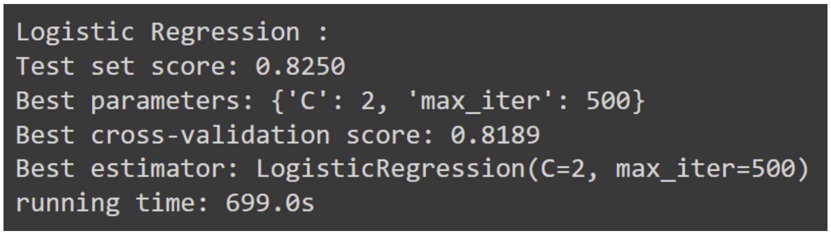
GRU can achieve a similar performance of LSTM, and it is easier to train and can greatly improve the training efficiency. Therefore, considering the computing power and time cost of hardware, GRU is more selected to be used in many cases compared with LSTM .
In this sentiment analysis task, GRU uses fewer training parameters and therefore uses less memory and executes faster than LSTM from the perspective of both LSTM and GRU layers of work.. Considering the need for less memory consumption and the desire for faster results, GRU was chosen for this task.

5. Experiments and Discussions

5.1 Describe experiments, comparison, and evaluation

5.1.1 Tunning parameter

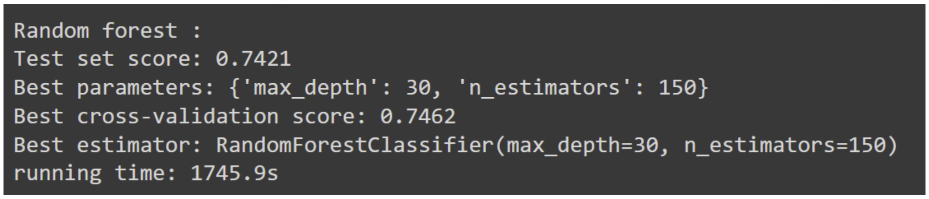
The team used cross-validation to fine-tune each algorithm.
In this task, we choose solver, max_iter, penalty and tol parameters of logistic regression to adjust. The control variables are max_iter and penalty, which are displayed below. As shown in the graph, the team found that the accuracy was higher whe C = 2. However, when the team adjusts the max_iter parameter, there is no significant changes in the accuracy. As for running time, when C = 2, we use less time to run the code, which is almost 0.7s. For changing max_iter, the running time is not affected much. Finally, the team chose a parameter with high accuracy, C = 2, even though its running time was slightly longer. So, the best parameter is C = 2 and max_iter = 500.
The tunning result figure is shown below.



```
Logistic Regression :
Test set score: 0.8250
Best parameters: {'C': 2, 'max_iter': 500}
Best cross-validation score: 0.8189
Best estimator: LogisticRegression(C=2, max_iter=500)
running time: 699.0s
```

Figure 5.1-1 The tunning result of Logistic Regression

In the random forest method, the team continues to call skLearn's classifier package. The team chooses max_depth, min_samples_leaf _iter, n_estimators, min_samples_split and random_state parameters of random forest to adjust. The control variables max_depth and n_estimators were tunning by the grid search function. And its result is displayed below. Control variables found that these two parameters had no significant impact on the accuracy, but only on the running time. Finally, the team choose the parameter n_estimators = 150 and max_depth = 30.



```
Random forest :
Test set score: 0.7421
Best parameters: {'max_depth': 30, 'n_estimators': 150}
Best cross-validation score: 0.7462
Best estimator: RandomForestClassifier(max_depth=30, n_estimators=150)
running time: 1745.9s
```

Figure 5.1-2 The accuracy and running time of Random Forest

5.1.2 Evaluation method

After the tuning of each method, three models with optimal parameters were obtained for further comparison. The team uses accuracy, precision, recall, confusion matrix, and running time to compare the performance of all the methods.

Confusion Matrix is used to observe the performance of the classifier, where the matrix column bar represents the predicted category of the instance, and the row bar represents the real category of the instance.

Table 5.1 Confusion matrix

	Negative(predicted value)	Positive(predicted value)
Negative(actual value)	TN	FP
Positive(actual value)	FN	TP

For accuracy, a given test data set is the ratio of the number of samples correctly classified by the classifier to the total number of samples. That is the accuracy of the test data set when the loss function is 0-1 loss [14].

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{S}$$

Figure 5.1 Definition of Accuracy Evaluation

For precision evaluation, it means what percentage of all positive predictions are actually positive.[1]

$$\mathrm{Prec} = \frac{\mathrm{TP}}{M_T}$$

Figure 5.2 Definition of Precision Evaluation

For recall evaluation, it means the percentage predicted positive of the total positive.[1]

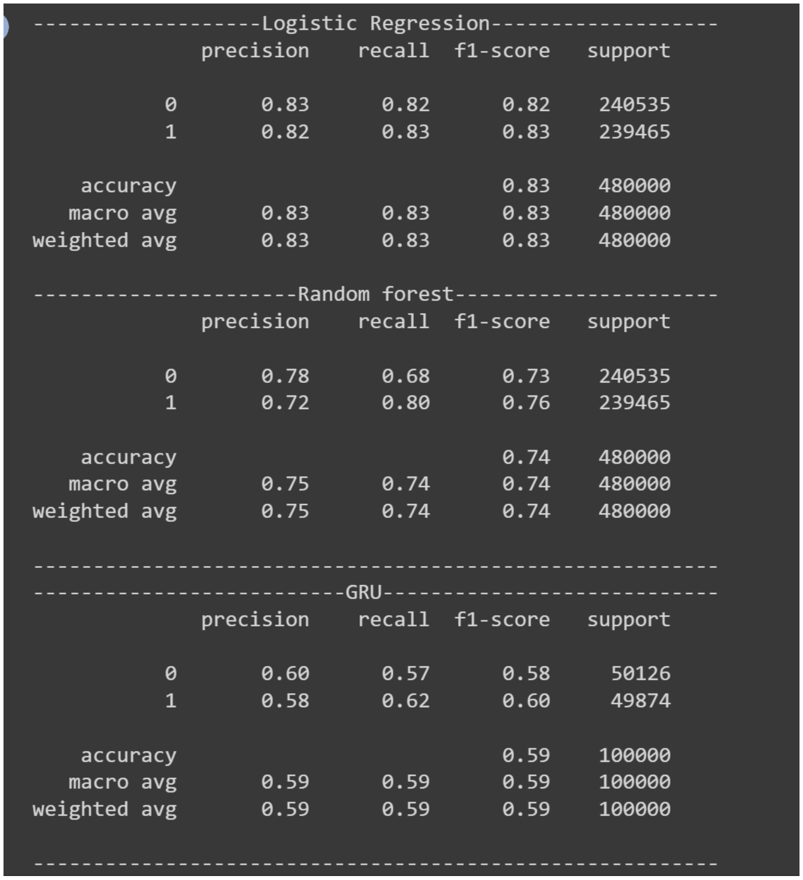
$$\mathrm{Recall} = \frac{\mathrm{TP}}{N_T}$$

Figure 5.3 Definition of Recall Evaluation

In addition, the miss rate and false-positive rate can be obtained through the confusion matrix, which is different in this experiment.

5.1.3 Evaluation result

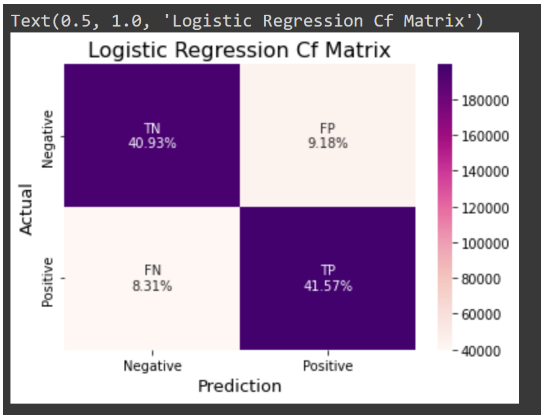
The team evaluated the results of a series of methods by the above formula as shown below.From this result, we can clearly see that the accuracy of logistic regression is up to 83%, random forest 78%, and GRU only 60%.



```
-------------------Logistic Regression-------------------
              precision    recall  f1-score   support

           0       0.83      0.82      0.82    240535
           1       0.82      0.83      0.83    239465

    accuracy                           0.83    480000
   macro avg       0.83      0.83      0.83    480000
weighted avg       0.83      0.83      0.83    480000

----------------------Random forest----------------------
              precision    recall  f1-score   support

           0       0.78      0.68      0.73    240535
           1       0.72      0.80      0.76    239465

    accuracy                           0.74    480000
   macro avg       0.75      0.74      0.74    480000
weighted avg       0.75      0.74      0.74    480000

---------------------------------------------------------
--------------------------GRU----------------------------
              precision    recall  f1-score   support

           0       0.60      0.57      0.58     50126
           1       0.58      0.62      0.60     49874

    accuracy                           0.59    100000
   macro avg       0.59      0.59      0.59    100000
weighted avg       0.59      0.59      0.59    100000

---------------------------------------------------------
```

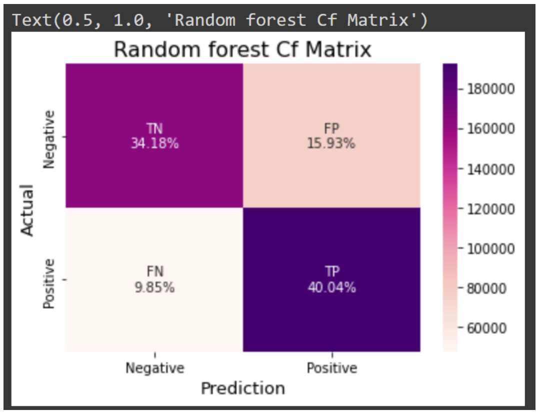
Figure 5.4 evaluation result

The team output three ways of confusion matrix as shown below.



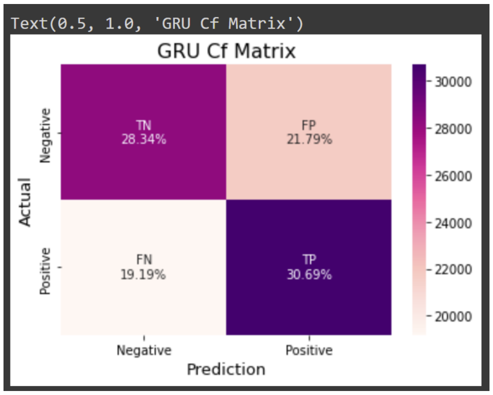
```
Text(0.5, 1.0, 'Logistic Regression Cf Matrix')
```

Figure 5.5logistic regression Cf Matrix



```
Text(0.5, 1.0, 'Random forest Cf Matrix')
```

Figure 5.6 Random forest Cf Matrix



```
Text(0.5, 1.0, 'GRU Cf Matrix')
```

Figure 5.7 GRU Cf Matrix

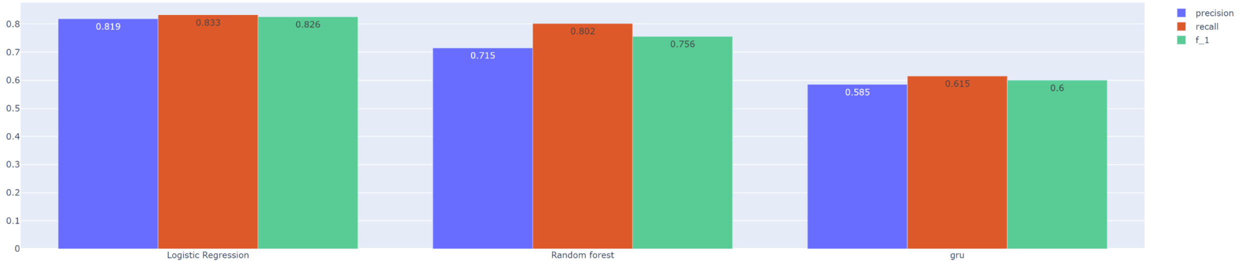


Figure 5.8 Compare methods

5.2 Team reflection

The team believes that the progress of the whole experiment is relatively smooth. We learned that the unstructured data set and a large amount of redundant data would deeply affect the performance of the training model, including accuracy, accuracy and recall rate. For better performance, the team preprocessed the data.

Some of the difficulties in this study are data processing. Tweets have punctuation, emojis and urls, and these are all things to deal with. The experimental environment is based on COLab and runs on Google, which makes teamwork easier

6. Conclusion

To sum up, this project aims to build three algorithm models, conduct sentiment analysis on tweets on Twitter, and select the optimal model. Our group established three models: Logistic regression, random forest and Gated Recurrent Units model. We found the logistic regression classifier to be the most accurate, and it may be the most suitable classifier model for this project.

The team knows that some issues remain unresolved, such as detecting and handling sarcasm. In the future, the team can continue to optimize preprocessing to take into account multiple scenarios. In this experiment, linear regression has higher accuracy and faster running speed, which is the best model method for this experiment.

However, the disadvantage of logistic regression is that it has only high accuracy in dealing with dichotomies, and the data must be linearly separable. In addition, when the feature space is large, the performance is poor. Therefore, in the future, when we use logistic regression algorithm to process sentiment analysis similar to Sentiment 140, we cannot judge more features of negative emotions such as anger, sadness and depression. We can combine the advantages of other algorithms to solve more related problems

References

[1] Jayaswal V. Performance metrics: Confusion matrix, precision, recall, and F1 score; 2020.

[2] BENOY, Kurianbenoy. power of simple logistic regression; 2020.

[3] RYUJIN_AM, Apoorvm. NLP randomforest and gradientboosting; 2019.

[4] For academics - sentiment140 - a Twitter sentiment analysis tool.

[5] Go, A.; Bhayani, R.; Huang, L. CS224N Technical Report: Twitter Sentiment Classification Using Distant Supervision. Stanford: Stanford University; 2009.

[6] Thejaswini N; Mr. Yadhu Naik B H. Twitter Sentimental Analysis Using Rule Based and Machine Learning Method. 2022

[7] Ankita, Nabizath Saleena: An Ensemble Classification System for Twitter Sentiment Analysis.

[8] Building a Twitter Sentiment Analysis System with Recurrent Neural Networks.

[9] Rustam, F.; Ashraf, I.; Mehmood, A.; Ullah, S.; Choi, G.S.: Tweets classification on the base of sentiments for US airline companies; 2019.

[10] Zhang, W.; Yoshida, T.; Tang, X. A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Syst. Appl; 2011.

[11] Freund, Yoav and Robert E. Schapire.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55; 1995: 119-139.

[12] Breiman, L.. Random Forests: Machine Learning; 2004: 5-32.

[13] Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.

[14] Li Hang: Statistical Learning Methods, Beijing: Tsinghua University Press; 2012.

[15]Kostadinov S. Understanding GRU networks; 2019.

Appendix

A: Summary of folder

The folder contains 4 files.

a) report.pdf

The report.pdf file is the report of 5318 assignment2.

b) code.pdf

The code.pdf file is the format of the code of 5318 assignment2.

c) code.ipynb

The code.ipynb file is the code of 5318 assignment2.

- emo_unicode.py

The py file is the import library about processing Emojis

d) trained model.h5

The trained model.h5 is the best model.

B: Hardware and software environments

Hardware:

Colab:

GPU: T4 ; RAM: 20G

Software:

Python 3.8

Requirements of this task:

```
# import library
import pandas as pd
import numpy as np
import re
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
nltk.download('stopwords')
from nltk.corpus import stopwords as sw
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, precision_score
from nltk.tokenize.treebank import TreebankWordDetokenizer
nltk.download('averaged_perceptron_tagger')
from nltk import pos_tag
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
from IPython.display import display, clear_output
from IPython.core.pylabtools import figsize
from emo_unicode import EMOTICONS_EMO # cite by https://github.com/NeelShah18/emot.git
from tqdm import tqdm
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
import copy
import seaborn as sns
import time
import torch
from torch.utils.data import TensorDataset
from torch.utils.data import DataLoader
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from sklearn.experimental import enable_halving_search_cv  # noqa
from sklearn.model_selection import HalvingRandomSearchCV
from scipy.stats import randint
import plotly.graph_objects as go
import time
import warnings
warnings.filterwarnings("ignore")
#clear_output()
%matplotlib inline
#plt.style.use("ggplot")
```

C: Code.ipynb

There are six parts in this code.ipynb:

1) Load data

2) Data pre-processing

3) Build model

4) Parameter Tuning

5) Classifier comparisons

6) Predict test dataset with the best model

Run all code from up to dwon can run normal and remember add the 'emo_unicode.py**' file into the path with main code.ipynb.**

C：System Information

Time of this report: 5/22/2022, 15:29:24
Machine name: DESKTOP-0H97T1N
Machine Id: {2A87F8D7-8B63-4B50-9A30-36FDF38E82AC}
Operating System: Windows 11 family 64-bit (10.0, Build 22000)
(22000.co_release.210604-1628)
Language: Chinese (Simplified) (Regional Setting: Chinese (Simplified))
System Manufacturer: LENOVO
System Model: 81C4
BIOS: 8GCN32WW (type: UEFI)
Processor: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (8 CPUs), ~2.0GHz
Memory: 16384MB RAM
Available OS Memory: 16226MB RAM
Page File: 18168MB used, 4458MB available
Windows Dir: C:\WINDOWS
DirectX Version: DirectX 12
DX Setup Parameters: Not found
User DPI Setting: 288 DPI (300 percent)
System DPI Setting: 288 DPI (300 percent)
DWM DPI Scaling: UnKnown
Miracast: Available, with HDCP
Microsoft Graphics Hybrid: Not Supported
DirectX Database Version: 1.2.2
DxDiag Version: 10.00.22000.0653 64bit Unicode

DxDiag Notes

D：How to run code

Run the code from top to bottom.
There are 24 blocks, which is 5 modules, including Load data and analyse data, Data pre-processing, Build model, Parameter Tuning and Evaluation and Comparison. Run the code from the bottom up in the order it was written.