Sentiment classification in the review of Genshin Impact

732A92 Text Mining

Chenjian Shi (chesh532)

Aug 16,2022

Abstract

Nowadays, video games are a vital part of the entertainment industry, and mobile games play an essential role. Recently, a mobile game named Genshin Impact has become increasingly popular and achieved many international awards. People talk about this game on Twitter and Reddit. Therefore, we want to do the sentiment analysis on the comments on Twitter and Reddit. However, we do not have labeled data of comments on Twitter and Reddit, so in this paper, we would like to use the reviews of Genshin Impact from the Google play store to build a sentiment classifier for classifying the data from Twitter. Finally, we found that the classifier based on the BERT has the best performance among the models in this paper. Using the BERT model to predict Twitter's data, we find that most people stay neutral but tend to give it a positive comment.

Contents

1 Introduction

Nowadays, video games are a vital part of the entertainment industry, and more and more people choose this way to relax. Meanwhile, people would like to play mobile games instead of traditional PC games or console games due to mobile phones' convenience and increasing performance. So they could play the games they like anywhere they like, for example, in the subway, or the Cafe, instead of playing games in a fixed place. Genshin impact, first published in Sep 2020, has become more popular. According to the report from BBC news(Baggs 2021), Genshin Impact stands at the top of mobile games, which earns $2 billion after 'unheard of' success in the first year. Moreover, this game was recently awarded many international awards in the field, like The Game Award for Best Mobile Game, Etc. (Awards n.d.). A newborn game could rarely get these achievements.

As we know, Twitter and Reddit are the most famous discussion places for people. So we could try to learn why Genshin Impact is so popular with people by analyzing the comments on Twitter and Reddit. However, currently, we do not have a data set with a sentiment label to analyze, which means it is hard for us to learn the reason from a comment since we do not know whether it is positive. So we need to find a way to get a model that could do the sentiment classification for the comments on Twitter and Reddit. Fortunately, we have the Google play store. The Google play store is one of the most popular video game stores on mobile phones. Therefore, massive reviews of games about the players' views on the games could be used to study the player's sentiment classification. Besides, we could use its score system as the label.

In this paper, we will try to apply these several machine learning and text mining skills to analyze the reviews from the Google play store to train the Sentiment classifier and then use the best classifier to do the sentiment classification on Twitter's data.

2 Theory.

2.1 Data Balancing

2.1.1 Synthetic Minority Over-sampling Technique (SMOTE)

Usually, the data in the natural lie is unbalanced, which means the number of each class/label is different. We need to balance the data to meet the requirement of the models, which requires the data has an equal number per class of labels. In this case, the Synthetic Minority Over-sampling Technique (SMOTE)(Chawla et al. 2002) would be used to balance the data. It may be the most popular oversampling method to do the data balancing. SMOTE could not only over-sampling the minority class but also the under-sampling of the majority class. The algorithm of SMOTE is processing as below,

For each sample, x in the minority class, use the Euclidean distance as the standard to calculate its distance to all samples in the minority class sample set S and get its k nearest neighbors.

Then a sampling ratio is set according to the sample imbalance ratio to help determine the sampling ratio N. For each minority class sample x, several samples are randomly selected from its k nearest neighbors, assuming that the selected nearest neighbor is xn.

At last, for each randomly selected xn, then construct a new sample with the original sample according to the following formula: $x_{new} = x + rand(0, 1) * |x - xn|$

However, there would create bias in the data sets.

2.2 Machine Learning Models

2.2.1 Non-deep Learning

Linear Support Vector Classification (LSVC) is a kind of supervised learning model and related learning algorithm for analyzing data in classification and regression analysis.(Cortes and Vapnik 1995) The LSVC implements a "one-vs-the-rest" multi-class strategy, thus training n classes.(Pedregosa et al. 2011)

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1} \max(0, 1 - y_i(w^T \phi(x_i) + b)),$$

Figure 1: Linear Support Vector Classification formula

Logistic regression (LR) is a generalized linear regression analysis model, often used in data mining, automatic disease diagnosis, economic forecasting, and other fields.(Pedregosa et al. 2011)

$$w^T x + b = ln \frac{P(Y=1|x)}{1 - P(Y=1|x)}$$

$$P(Y=1|x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

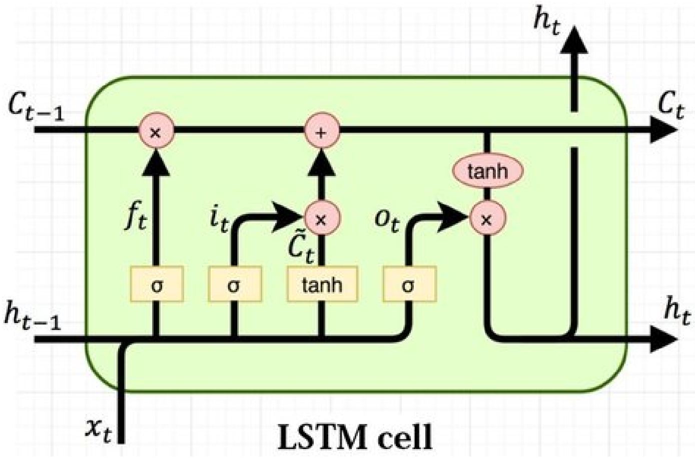
Figure 2: Logistic Regression formula

2.2.2 Deep Learning

Bidirectional Long Short-Term Memory (BiLSTM) :

Long short-term memory (LSTM) is a particular recurrent neural network (RNN), mainly to solve the problem of gradient disappearance and gradient explosion during long sequence training. Simply put, LSTM can perform better in longer sequences than ordinary RNNs.

The LSTM model is composed of the input word x_t at time t, the cell state C_t, the temporary cell state $\hat{C}_t$, the hidden layer state h_t, the forgetting gate f_t, the memory gate i_t, and the output gate o_t. For the calculation process of LSTM, first, forgetting the information in the cell state and memorizing new information, the valuable information for subsequent calculations is transmitted. Then discard useless information, and the hidden layer state h_t is output at each step. Besides, the forgetting, memory, and output are controlled by the forgetting gate f_t, memory gate i_t, and the output gate o_t, which are calculated by the last time hidden layer state h_{t-1} and current input X_t.(Hochreiter and Schmidhuber 1997)



$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

Figure 3: LSTM-theory

(Varsamopoulos, Bertels, and Almudever 2018)

Bidirectional Long Short-Term Memory (BiLSTM) comprises forwarding LSTM and backward LSTM. (Cornegruta et al. 2016)

For example, input “I”, “Love”, “You” in turn to get three vectors $[h_{L0}, h_{L1}, h_{L2}]$. Enter “you”, “love”, and “me” in turn to get three vectors $[h_{R0}, h_{R1}, h_{R2}]$. Finally, the forward and backward hidden vectors are spliced to obtain$[(h_{L0}, h_{R0}), (h_{L1}, h_{R1}), (h_{L2}, h_{R2})]$, namely $[h_0, h_1, h_2]$

For sentiment classification tasks, the representation of the sentences we use is often $[(h_{L2}, h_{R2})]$. Because it contains all the forward and backward information

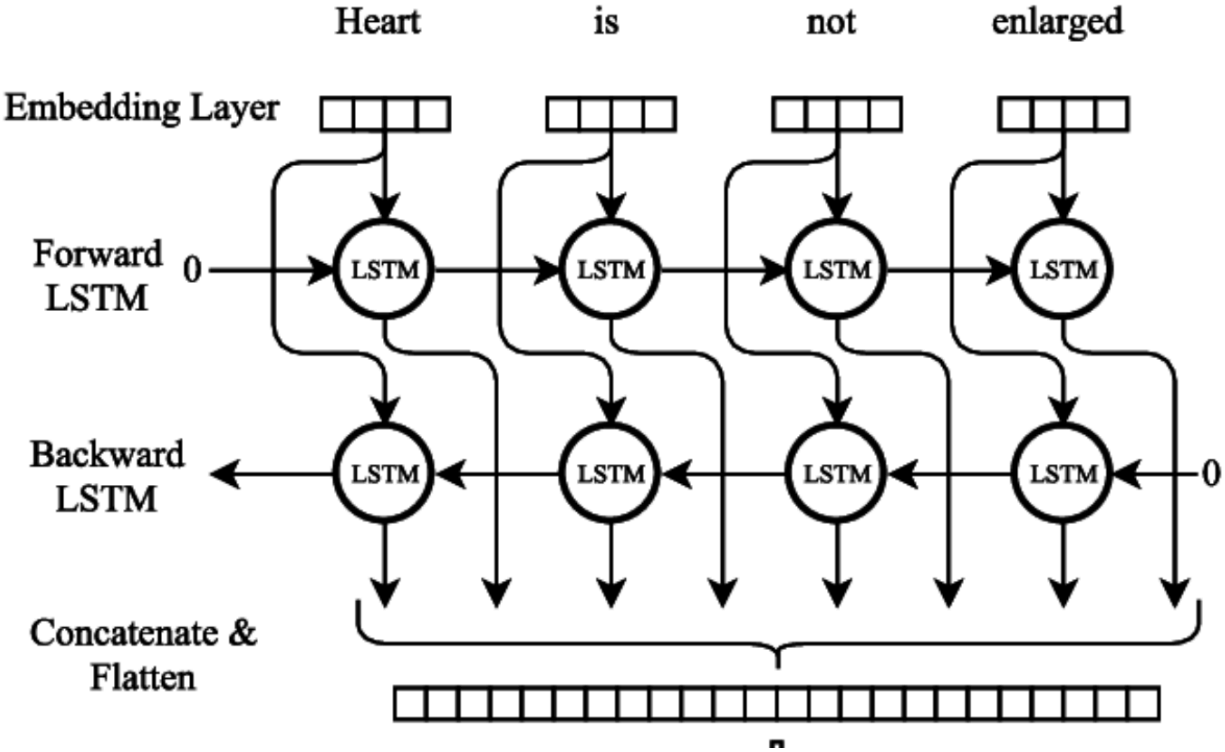


Figure 4: BiLSTM

(Cornegruta et al. 2016)

Bidirectional Encoder Representations from Transformers (BERT) :

BERT, a transformer-based bidirectional encoding representation, is a pre-training model. The two tasks during model training are to predict the words masked in the sentence and determine whether the two input sentences are upper and lower. After the pre-trained BERT model is added to the corresponding network according to the specific task, the downstream tasks of NLP can be completed, such as text classification, machine translation, Etc.(Vaswani et al. 2017)

Although BERT is based on the transformer, it only uses the encoder part of the transformer. Its overall framework is formed by stacking the encoders of multiple layers of transformers. The encoder of each layer is composed of a layer of muti-head-attention and a layer of feed-forward. The large model has 24 layers, each layer has 16 attention, and the miniature model has 12 layers. Each layer has 12 attention. The main role of each attention is to re-encode the target word through the relevance of the target word to all words in the sentence. Therefore, calculating each attention includes three steps: calculating the correlation between words, normalizing the correlation, and obtaining the encoding of the target word through the weighted summation of the correlation and the encoding of all words.

When calculating the correlation between words through attention, first linearly transform the input sequence vector (512*768) through three weight matrices, respectively generate three new sequence vectors of query, key, and value, and use each word. The query vector is multiplied with the key vector of all words in the sequence to obtain the correlation between words, and then this correlation is normalized by softmax. Next, the normalized weight is summed with the value weight. Finally, get a new encoding for each word.

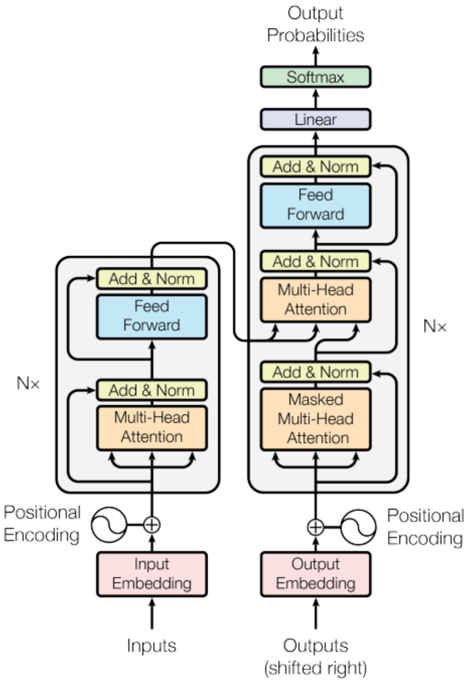


Figure 5: BERT-theory

(Figure from (Vaswani et al. 2017))

In BERT, the input vector is summed by three different embeddings, namely(Paul and Saha 2020):

Wordpiece embedding: The vector representation of the word itself. WordPiece refers to the division of words into a limited set of standard subword units, which can compromise words' effectiveness and the characters' flexibility.

Positional Embedding: Encodes the positional information of words into feature vectors. Because our network structure does not have RNN or LSTM, the position information of the sequence cannot be obtained, so the position embedding needs to be built. There are two ways to build positional embeddings: BERT initializes a positional embedding and then learns it through training, while Transformer builds positional embeddings by formulating rules.

Segment Embedding: A vector representation used to distinguish two sentences. It is used to distinguish asymmetric sentences, such as questions and answers.

The input of the BERT model is wordpiece token embedding + segment embedding + position embedding, as shown in the figure:

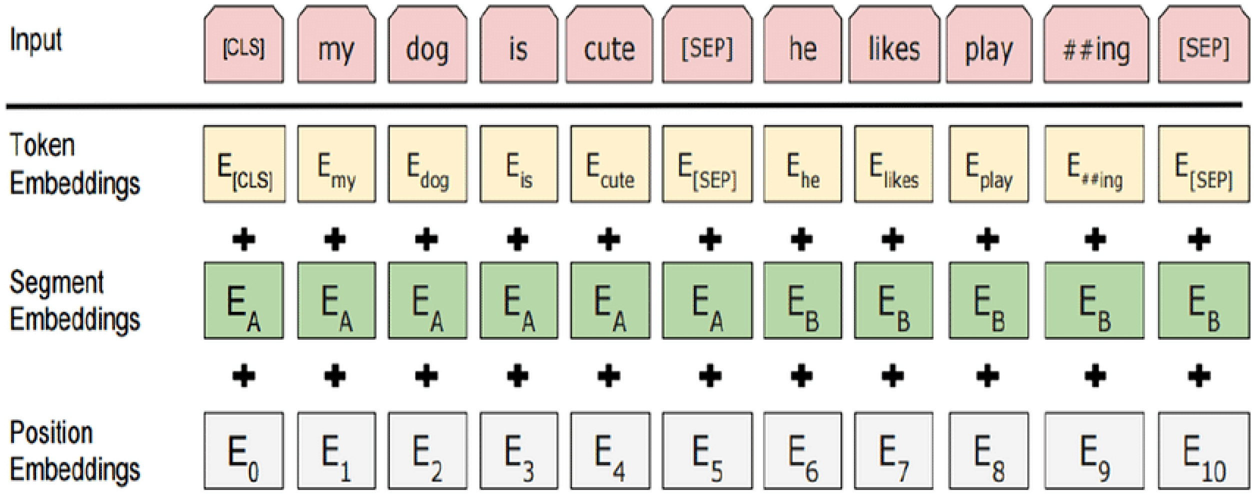


Figure 6: BERT-Input

(Figure from (Paul and Saha 2020))

2.2.3 Metrics

According to the task goal, we have select 3 differernt kinds of metrics from both scikit-learn (Pedregosa et al. 2011) and tensorflow(Abadi et al. 2015)

- Kappa score: The score lies in the range [-1, 1]. A score of -1 represents complete disagreement between two raters, whereas a score of 1 represents the complete agreement between the two raters. A score of 0 means agreement by chance.
- fbeta-score: It is the weighted harmonic mean of precision and recall. Output range is [0, 1]. Works for both multi-class and multi-label classification. In order to compare the result between non-deep learning models and deep learning models, here we set the $\beta = 1$ in the fbeta-score so that the fbeta-score would be theoretically the same as the f1-score.
- Area Under the ROC Curve (AUC): AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.("Classification: Roc Curve and AUC | Machine Learning Crash Course | Google Developers," n.d.)

3 Data.

3.1 Data access

3.1.1 Google Play

The data was scraped from the reviews of Genshin Impact on Google play(Google n.d.) on Jan 09th, 2022, by python script, which is implemented from a package named google-play-scraper(JoMingyu n.d.). According to the aim of this report, only choose the data of "Time," "Score," and "Content," which stand for the date, content, and the score of one review, and save them as Excel files. Besides, the score range in the Google play store is from 1 to 5, which could be regarded as a different attitude to things, like 1 (Extremely Negative), 2 (Negative), 3 (Neutral), 4 (Positive), 5 (Extremely Positive). The data we have is like below:

```
## # A tibble: 38,598 x 3
##    Time                Score Content
##    <dttm>              <dbl> <chr>
##  1 2022-01-05 04:22:08     5 I literally love this game. Playing on a phone the~
##  2 2022-01-07 04:54:02     5 The game itself is wonderful and enchanting, I lov~
##  3 2022-01-08 00:08:37     5 This is an amazing game it has amazing graphics an~
##  4 2022-01-04 23:52:11     5 Full fledged game that you can play for free. Ther~
##  5 2022-01-05 01:08:32     5 This game is so fun! The story is thrilling and in~
##  6 2022-01-05 09:25:18     5 Honestly the only downfall on mobile is only touch~
##  7 2022-01-05 01:11:29     5 This is a very lovely game. The graphics are amazi~
##  8 2022-01-04 23:14:50     5 I played more than 1 year, since I was playing thi~
##  9 2022-01-06 05:40:17     5 Am amazing game s far. Really don't understand why~
## 10 2022-01-05 07:06:38     5 I don't normally play these type on games whether ~
## # ... with 38,588 more rows
```

3.1.2 Twitter

The data from Twitter was scraped by Twint, a package used for scrapping the data with specific contents from Twitter(Project n.d.). According to the goal, we collected tweets with the tag "GenshinImpact" the data date is selected from 2022-01-09 to 2022-01-10. However, due to the limitation of the computational resources, we only collect 1522 rows of data for later usage. The data we have is like below:

```
## # A tibble: 1,522 x 2
##    Time                Tweet
##    <chr>               <chr>
##  1 2022-01-10 00:59:55 @DashSlasher901 @HakimiDoStuff @Tuont0 y'all can fix her~
##  2 2022-01-10 00:59:53 @GamerUnderworld @GenshinImpact But the main event is ba~
##  3 2022-01-10 00:59:48 NBA 2 League Genshin Impact FC  Hearthstone  I think thi~
##  4 2022-01-10 00:59:46 My Server: NA My Adventure Rank: 38 My first 5white_medi~
##  5 2022-01-10 00:59:18 #PS5Share, #GenshinImpact #shenhe # SHENHE IS SO PRETTY
##  6 2022-01-10 00:59:14 No think, just her.. Changed up her design aa bit.. #she~
##  7 2022-01-10 00:59:09 Happy Birthday Now! two_hearts You're the best malewife ~
##  8 2022-01-10 00:59:08 happy birthday note :) #GenshinImpact #ka #ayaka #!
##  9 2022-01-10 00:58:28 <NA>
## 10 2022-01-10 00:58:13 Genshin Impact: un studio del pozioni Dusty Wrack Event ~
## # ... with 1,512 more rows
```

We could find that the language in the Tweets is not so 'English', which contains some non-English word and some abbr. In this case, this data cannot be used in directly, but need a cleaning.

3.2 Data Overview

3.2.1 Google Play

Due to the limited resources or other cloud platforms like Colab, it is hard to use all these data since it takes about 10 hours to clean 3000 records. So, the data had to be sampled according to the original data's distribution with a ratio rate of 10%. The sampled data would be split into training and test sets. This data size for the training model may not be sufficient, leading to a low-performance model.

Wordcloud(Oesper et al. 2011) and matplotlib(Hunter 2007) would be used for data visualization.

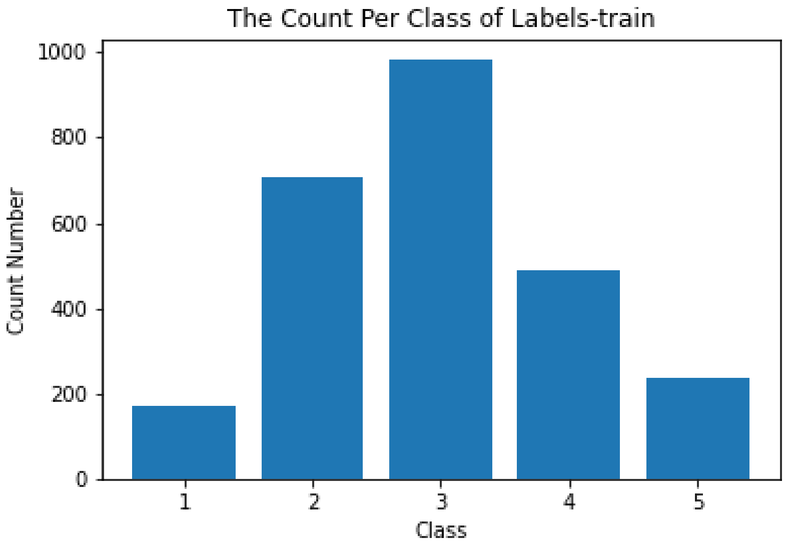


Figure 7: The Count Per Class of Labels for train data

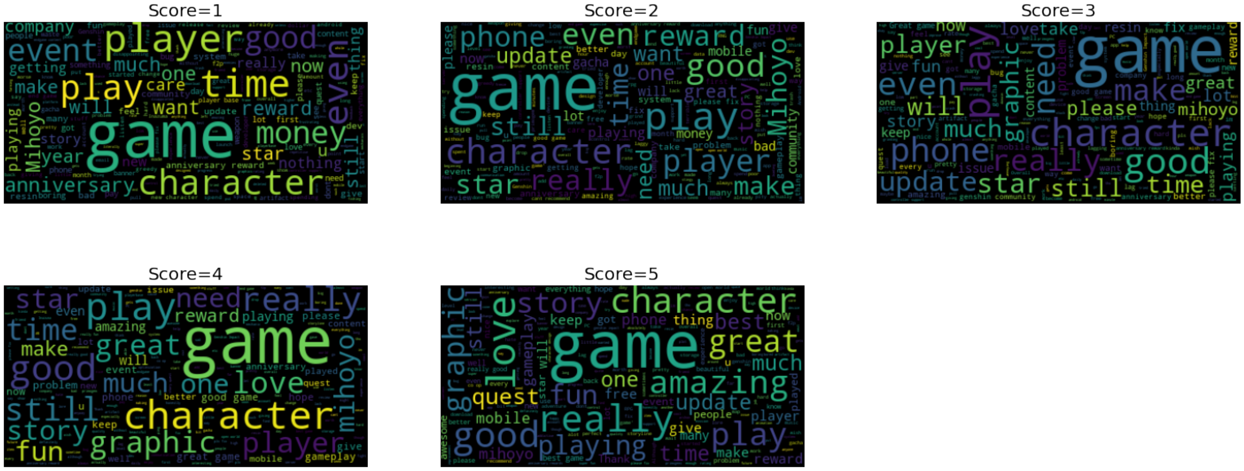


Figure 8: The wordcloud per class of labels for train data

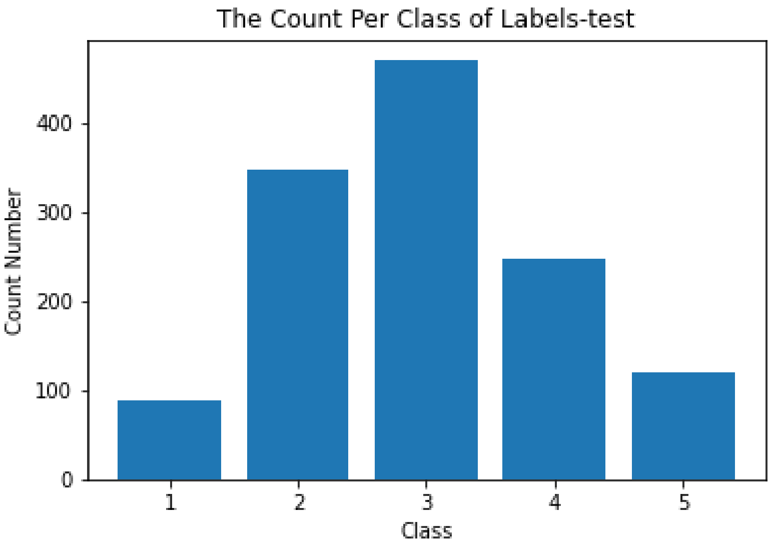
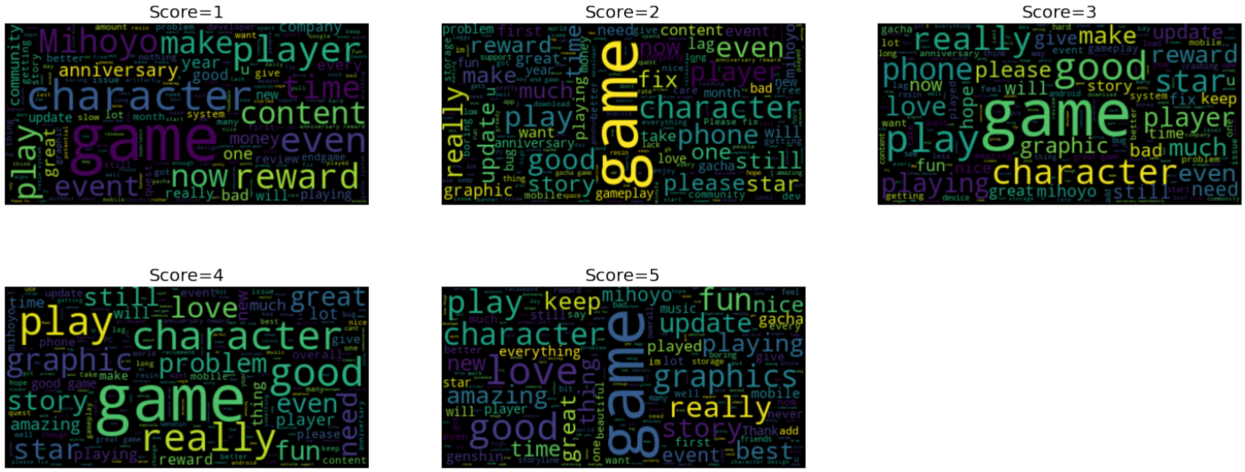


Figure 9: The Count Per Class of Labels for test data


Score=1

Score=2

Score=3

Score=4

Score=5

Figure 10: The wordcloud per class of labels for test data

According to the figures above, we could find that the train data are imbalanced, which is necessary to balance before training. However, this data balancing would make bias to the results.

3.2.2 Twitter

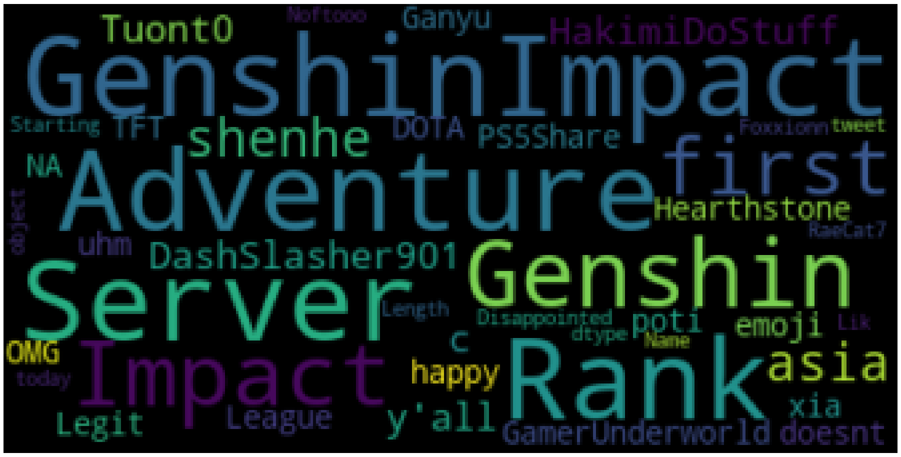


Figure 11: The wordcloud of Twitter's Data

3.3 Data Cleaning

In practice, the most important and troubling part of deploying a machine learning model might be data cleaning. When deploying a machine learning model, the choice of model and the parameter adjustment depends on the objective performance of the model, which usually has standard rules to follow. However, the data we used in the model cannot be changed if the performance is not good. Besides, if the data is of low quality, for example, the data size is too small, or the label/features are not correctly written in the data sets, all these would lead to bad model performance.

As for the data we use here, the content contains much informal spelling or abbreviation. Besides, even if we set the language to 'en,' there may still be non-English words or emojis. So to get a higher quality of data, we would use the way introduced below to do the data cleaning:

a) Contextual Spell Check It is a spell check package (Goel n.d.) from the spacy universe, which uses the BERT model to correct the spelling of the words. So it could be added to the spacy pipeline. Here the doc._.outcome_spellCheck would be used as the correct content after the process. Besides, it could also be used for language detection in a way.

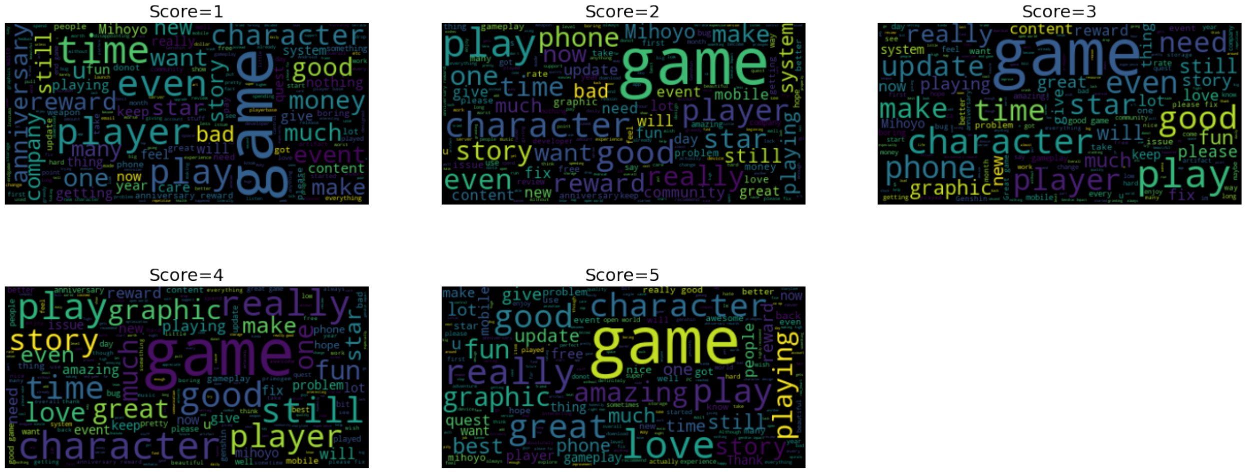
b) emoji process Since many emojis in the text may represent the user's sentiment, we use the package emote (Neel Shah n.d.) to process the emoji in the context. Moreover, this package could help us to transform the emoji into relative English. For example, the smiling face may be transformed into the "happy" word.

The cleaning work is implemented by using the spacy(Honnibal et al. 2020) pipeline.

Since the processing time of this cleaning takes too much time (10 hours for 3000 records). The data would be processed and saved as Excel files and used to deploy later.

As introduced above, some data may not be in English or not spelled correctly. In this case, the return of the doc._.outcome_spellCheck would be null, which means some data contains the NA value in the data set. So we need to drop the NA value.

3.3.1 Google Play



Score=1

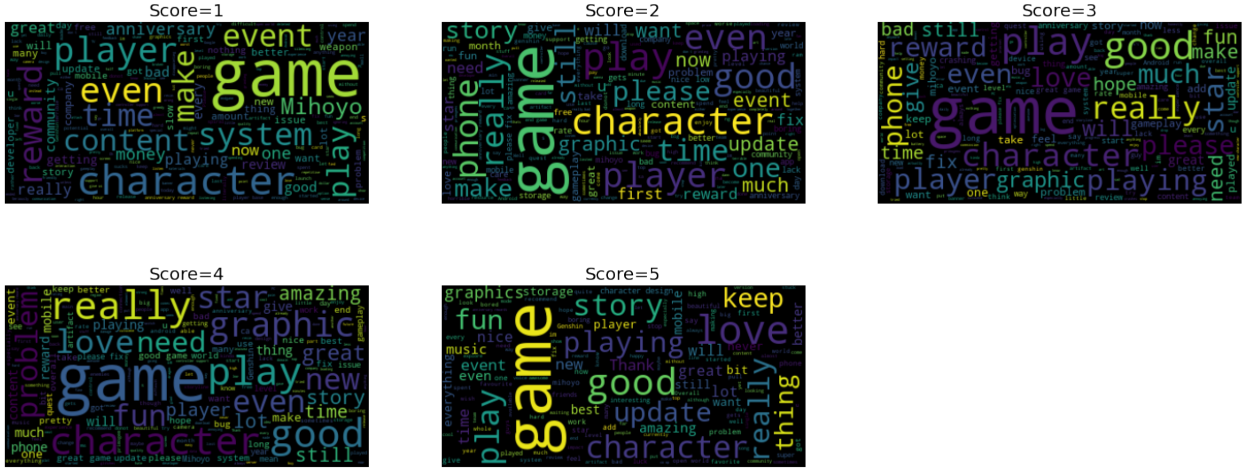
Score=2

Score=3

Score=4

Score=5

Figure 12: The wordclod per class of labels for train data



Score=1

Score=2

Score=3

Score=4

Score=5

Figure 13: The wordclod per class of labels for test data

By comparing the word cloud and shape of data before cleaning, we could find that, although the highest frequency keyword for the different labels would still be ‘game,’ the frequency is not higher than before, and other words like ‘reward’ get higher frequency.

3.3.2 Twitter

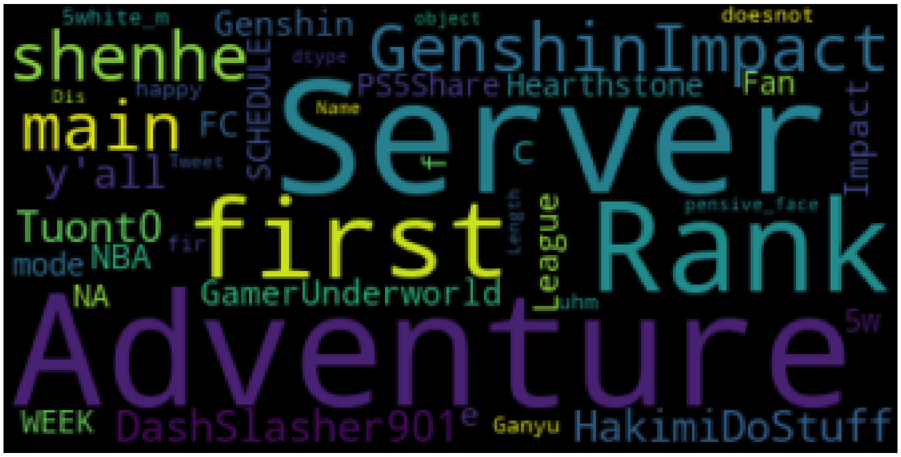


Figure 14: The wordclod for Twitter data

Here is the word cloud for Twitter's data; according to the plot, we could find that some terms, like 'GenshinImpact,' 'Adventure,' and 'Rank,' and so on, have changed.

3.4 Data Balance

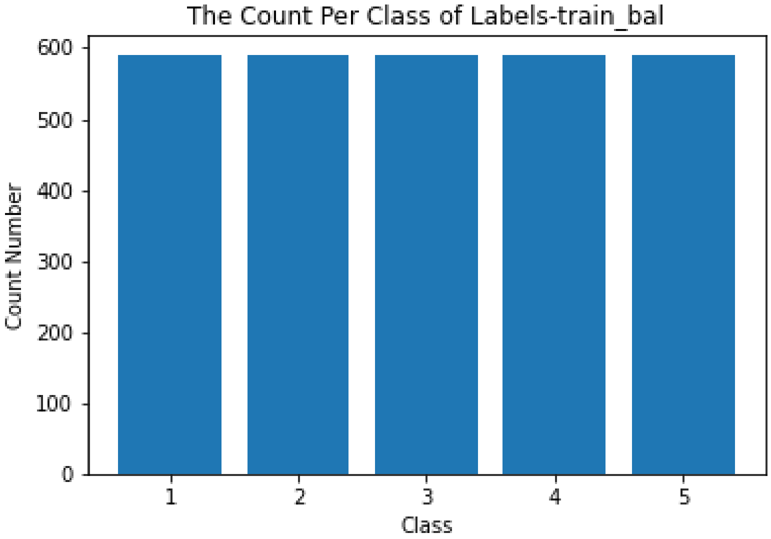


Figure 15: The Count Per Class of Labels for balanced train data

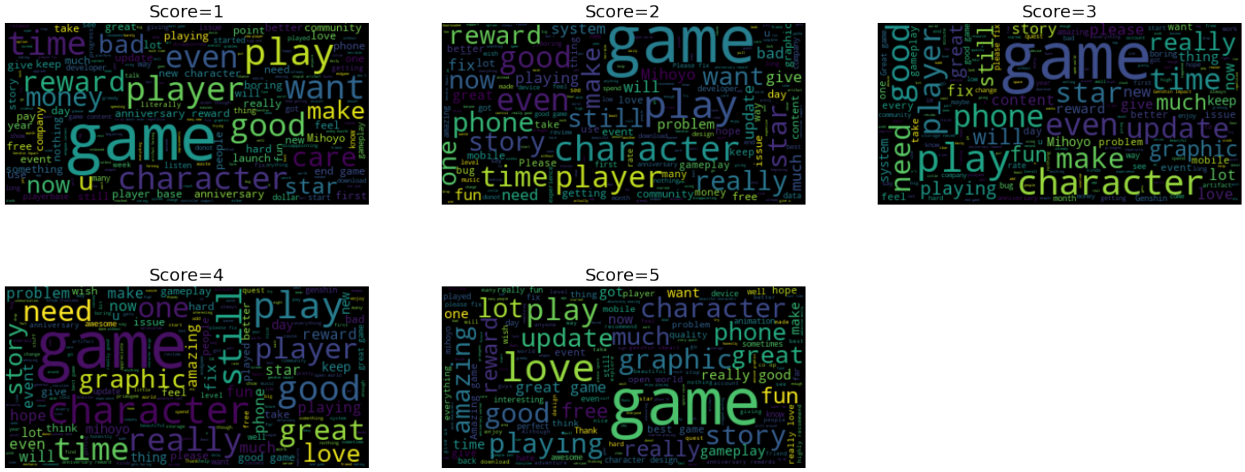


Figure 16: The wordclod per class of labels for balanced train data

There is the visualization of the training data we used in the later model training; we could find that all the label has been balanced, and all have the same count number of the different classes. However, the word cloud seems that not too much change.

4 Method.

First, the data has already been divided into training and test data. Then, the SMOTE mentioned previously would be used on train data to balance the data.

After we have balanced training data, we could do the necessary data preprocessing based on different models' needs. For example, for a non-deep learning model, the preprocessing would be vectorization. Moreover, for the deep learning model, the preprocessing would be tokenization.

Then we could first apply the non-deep learning models, like LSVC and Logistic regression. Then applies the deep learning models. Finally, we would select the best performance model to predict Twitter's data after training all the models.

5 Results.

5.1 Google Play

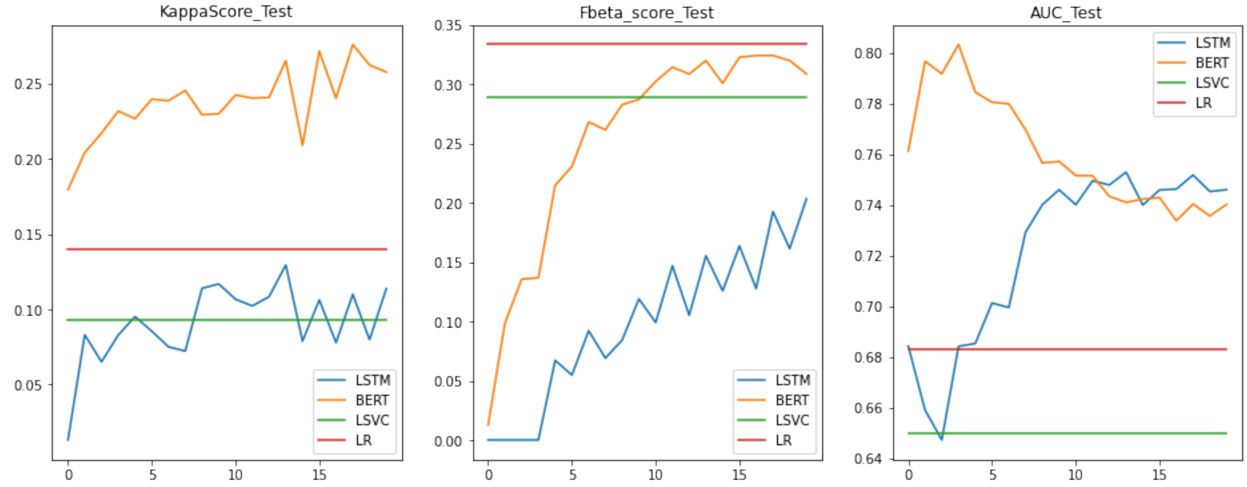


Figure 17: Result-Summary

We first focus on the Kappa score. According to the definition, we could find that the BERT model takes the highest score, which means it has the highest classification precision among those models. Then we come to the Fbeta score, in which the linear regression model has the highest score, but the BERT model has a close score to the linear regression model. In this report, we set the beta value to 1, which means there is a balanced evaluation between the focus of recall and the precision. Finally, we come to the AUC score; we could find that the LSTM and BERT have a similar score, much higher than the two other models.

In this case, according to the result above, we could find that the BERT model has the best performance among these models. Therefore, we would use the BERT model to predict Twitter's data in this case.

5.2 Twitter

```
## # A tibble: 940 x 3
##    Time                Tweet                                              Score
##    <chr>               <chr>                                              <dbl>
##  1 2022-01-10 00:59:55 @DashSlasher901 @HakimiDoStuff @Tuont0 y'all can f~     5
##  2 2022-01-10 00:59:53 @GamerUnderworld @GenshinImpact But the main event~     4
##  3 2022-01-10 00:59:48 NBA 2 League Genshin Impact FC  Hearthstone  I thi~     5
##  4 2022-01-10 00:59:46 My Server: NA My Adventure Rank: 38 My first 5whit~     4
##  5 2022-01-10 00:59:18 #PS5Share, #GenshinImpact #shenhe # SHENHE IS SO P~     5
##  6 2022-01-10 00:59:14 No think, just her.. Changed up her design aa bit.~     3
##  7 2022-01-10 00:59:09 Happy Birthday Now! two_hearts You're the best mal~     4
##  8 2022-01-10 00:59:08 happy birthday note :) #GenshinImpact #ka #ayaka #!     3
##  9 2022-01-10 00:58:13 Genshin Impact: un studio del pozioni Dusty Wrack ~     4
## 10 2022-01-10 00:57:48 @GenshinImpact Heyyy,,mind giving me xiao? |====ba~     3
## # ... with 930 more rows
```

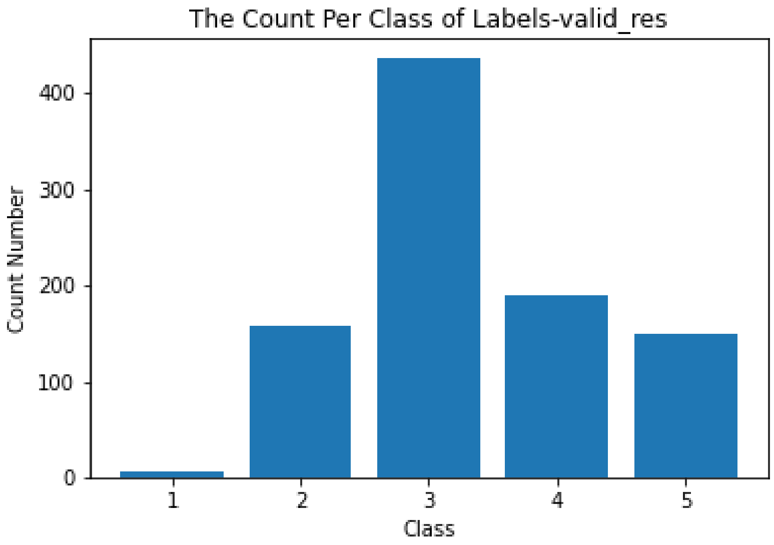
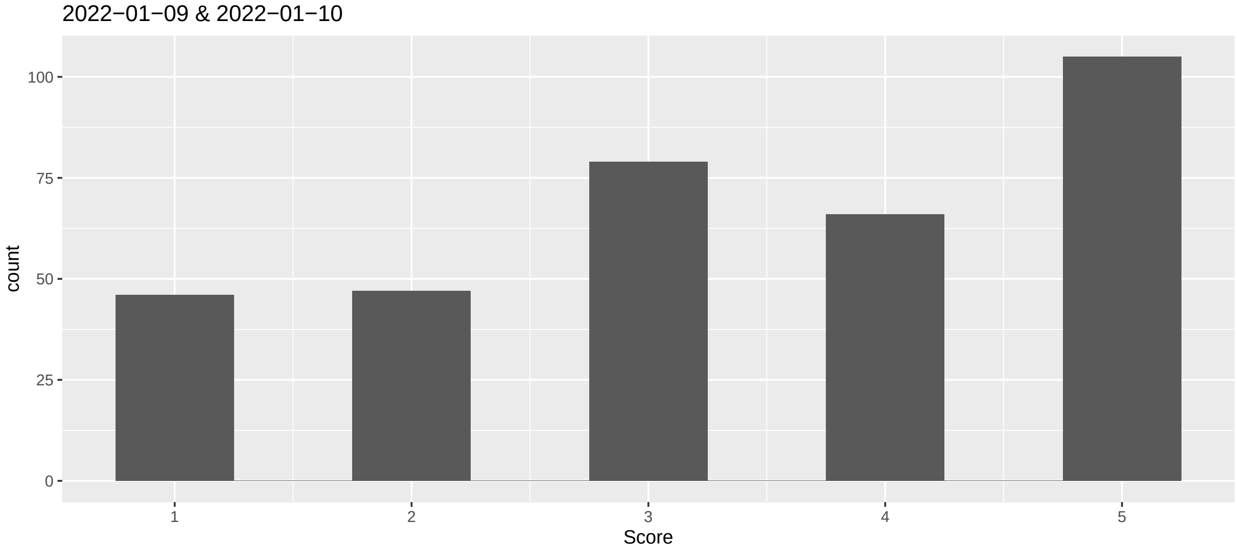


Figure 18: The Count Per Class of Labels for Twitter data

Here is the result of the prediction by using the BERT model. According to the result, we could find that the model predicts this data most belongs to class 3, which is the neutral one.

6 Discussion.

We used the BERT model in the previous section to predict Twitter's data. We could find that the number of scores means more people give positive comments on this game than negatively. However, according to Google play's data, which is shown below. Most people comment highly on this game, and other scores seem to have similar numbers.



In this case, there is maybe some problem with the prediction data (Twitter's data) or the original data (Google Play's data). So, we select some data from different scores in both data sets. Then, we would use the random sampling function built in R with the seed number of 2022.

```
## [1] "Google Play:"
```

Score 1:“At least give us some more primogems. If even lvl 80 trial characters can’t complete the current event with lvl 65 monsters, what are *we* supposed to do.”

Score 2:“Bounty quest in liyue,the enemy was still within the bounty area but it refilled it’s HP”

Score 3:“Best RPG I’ve ever played, great storyline and characters the reason I’ve only put 3 stars is due to the amount of space this game takes up. Within the next couple of updates I’ll have to delete the app for good because it takes most of my storage”

Score 4:“Server: Europe Adventure Rank: 56 almost 57 First 5*: Tartaglia Last 5*: Red Main character: Albedo/Kazuha New weapon: Black Swords element: Anemo/Geo Fav region in Teyvat: Dragonspine Fan food: Matsutake Meat Rolls Blue colour: Curcuma #GenshinImpact #!”

Score 5:“i love the game, but from time to time, I tend to forget about the game since sometimes it gets boring, but overall, wonderful game :)”

```
## [1] "Twitter:"
```

Score 1: “@SleeplesslyS @GenshinImpact This is disappointing. I didn’t notice before but...person_facepalming . female_sign_?I feel huge disappointment towards them for doing this.”

Score 2:“i miss text :(@GenshinImpact RELEASE HIM FROM THE BASEMENT”

Score 3:“My Server:EU My Adventure Rank:45 My first 5white_medium_star:Tartaglia My last 5white_medium_star:Jean My favourite character:Sucrose My favourite weapon:polearm My favourite element:water My favourite region in Teyvat: Inazuma My favourite food: Tontsuku Ramen My Seelie colour: orange #GenshinImpact #!”

Score 4:“This game is awesome! This is the only RPG I play, because most of them aren’t interesting enough. I’ve been playing This for months. 10 out of 10 would recommend.”

Score 5:“Mission success! We are a #shenhe haver!....by drawing her :) I think my art improved #GenshinImpact #.”

According to the data above, we could find that there is no special problem for both data sets, which shows a reasonable reason for being graded correctly. However, it is for humans, but not for machines. So, for example, we could find that Score 3 in Google play’s data contains the word ‘best’ but gets a grade of 3. In this case, it may ‘mislead’ the models like linear regression. But for, a model like the BERT, which considers the whole sentence but not just words, would avoid a problem like this.

7 Conclusion.

By comparing the results above, we could find that all these models we used in this paper do not perform well on these data. However, the BERT is the best model among them.

If we want to get a well-performance machine learning model, it is essential to have a high-quality and large enough data set for training and testing since it is the basis of a model. The model learns the feature(information) from this data set and applies it to other places. However, due to the limitation of the computer resources, we have to use a small data set, which may not contain enough features for models to learn. Besides, the quality of the data is also not good enough. Although it is common in real life that the information is imbalanced, the method we used here, oversampling, is not enough for balancing the data, which also brings a bias to the data. For further work, we may consider combining oversampling and undersampling to create more balanced data.

What is more, although the deep learning model usually performs better than the non-deep learning model when applied to some extensive data analysis, it takes time and needs to find the best parameters for it. Like in this paper, the performance of the four models is quite similar. However, the non-deep learning model results much quicker and easier than the deep learning model.

8 Further work

In order to improve the performance of the model, we could have a try on the method below:

- 1) Try to use the combination of oversampling and undersampling to create more balanced data.
- 2) Try to use other measurements(metrics) for models.
- 3) Try to use other kinds of models, for example, CNN, Etc.
- 4) If possible, try to use a complete data set.

Reference

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." https://www.tensorflow.org/.

Awards, The Game. n.d. "Best Mobile Game | Nominees | the Game Awards." Accessed January 14, 2022. thegameawards.com/nominees/best-mobile-game.

Baggs, Michael. 2021. *BBC News*. BBC. https://www.bbc.com/news/newsbeat-58707297.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. "SMOTE: Synthetic Minority over-Sampling Technique." *Journal of Artificial Intelligence Research* 16 (June): 321–57. https://doi.org/10.1613/jair.953.

"Classification: Roc Curve and AUC | Machine Learning Crash Course | Google Developers." n.d. *Google*. Google. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=zh_cn.

Cornegruta, Savelie, Robert Bakewell, Samuel Withey, and Giovanni Montana. 2016. "Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks." *CoRR* abs/1609.08409. http://arxiv.org/abs/1609.08409.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–97.

Goel, Rajat. n.d. "Contextual Spell Check spaCy Universe." Accessed January 9, 2022. spacy.io/universe/project/contextualS

Google. n.d. "Genshin Impact - Apps on Google Play." Accessed January 9, 2022. play.google.com/store/apps/details?id=com

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. "spaCy: Industrial-strength Natural Language Processing in Python." https://doi.org/10.5281/zenodo.1212303.

Hunter, J. D. 2007. "Matplotlib: A 2d Graphics Environment." *Computing in Science & Engineering* 9 (3): 90–95. https://doi.org/10.1109/MCSE.2007.55.

JoMingyu, etc, kluhan. n.d. "Google-Play-Scraper." Accessed January 9, 2022. github.com/JoMingyu/google-play-scraper.

Neel Shah, etc, Shubham Rohilla. n.d. "Emote." Accessed July 7, 2022. github.com/NeelShah18/emot.

Oesper, Layla, Daniele Merico, Ruth Isserlin, and Gary D Bader. 2011. "WordCloud: A Cytoscape Plugin to Create a Visual Semantic Summary of Networks." *Source Code for Biology and Medicine* 6 (1): 7.

Paul, Sayanta, and Sriparna Saha. 2020. "CyberBERT: BERT for Cyberbullying Identification." *Multimedia Systems*, November, 1–8. https://doi.org/10.1007/s00530-020-00710-4.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Project, TWINT. n.d. "Twint." Accessed July 9, 2022. github.com/twintproject/twint.

Varsamopoulos, Savvas, Koen Bertels, and Carmen Almudever. 2018. "Designing Neural Network Based Decoders for Surface Codes," November.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." https://arxiv.org/abs/1706.03762.