A study on the prediction of user participation and game outcome of Wordle guessing game based on Arima time series and BERT neural network model

Yuanhang Wang *

School of Automation, Northwestern Polytechnical University Xi'an, China

* Corresponding author: wyh2020302526@mail.nwpu.edu.cn

Abstract. Wordle is a popular daily puzzle game in which players try to guess a five-letter word six times or less. This report develops predictive models for the distribution of reported outcomes and the number of reported outcomes reported. The data were first preprocessed, and then an ARIMA(1,1) prediction model was developed by ADF testing and plotting ACF and PACF plots, and calculated that there would be 7,260 reported outcomes on March 1, 2023. In addition, two attributes were extracted, namely the number of repeated letters and the information entropy. Through Pearson correlation analysis, the report investigates the relationship between the attributes and the variance of the scores. It can be seen that entropy is significantly negatively correlated with variance and the number of repeated letters is significantly positively correlated. The report uses the One-hot method to encode words, applies the BERT neural network architecture, and uses the SGD algorithm for optimization to predict the distribution of the reported results. For the word EERIE, the calculated percentages were 1.16%, 5.75%, 20.68%, 32.60%, 24.72%, 12.39%, and 2.70%. The calculated MSE errors indicate the high confidence level of the model. The study provides insight into user behavior and improves game design.

Keywords: ARIMA, BERT, Word Guessing Game.

1. Introduction

Wordle, a word-guessing game with a matrix of yellow and green blocks as the interactive interface, is popular worldwide and currently supports more than 60 languages.

The rules of this game are very simple: it requires the player to guess a word consisting of 5 letters within 6 times. Each time the letters are entered, the game gives the player some clues in color: if the letters are entered in the target word but in the wrong position, the grid will be shown in yellow; if the letters are entered in the target word and in the correct position, the grid will be shown in green; if neither is correct, it will be shown in gray. The details are shown in Figure 1 below. Players need to adjust their guessing strategy according to the feedback after each guess to finally guess the word. The biggest difference in the hard mode is that players need to use all the letters marked in green and yellow in their next guess, and the green letters need to be in the right place.

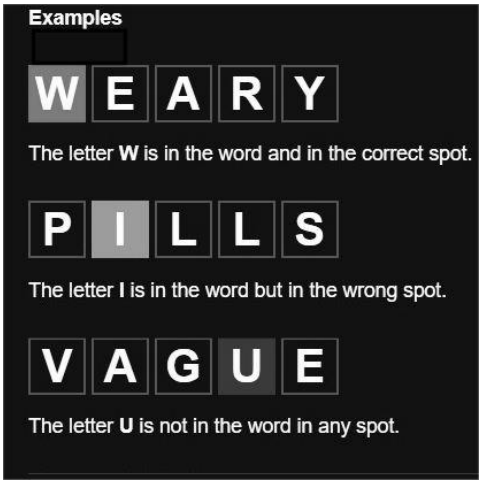


Figure 1. Rules of Wordle

Wordle has only one challenge per day, and it keeps statistics on the player, including the percentage of wins and a breakdown of the number of attempts needed to get the correct answer each day.

To predict the amount of user engagement and user guessing results, Laurent Poirrier, a former research assistant professor at the University of Waterloo, has a series of discussions on his blog about "Is it possible to guarantee the answer to any Wordle within 6 guesses". And Laurent found the best decision tree for Wordle by enumerating all possible decision trees. With careful thought, some clever optimization techniques and over a thousand hours of CPU time, he found a decision tree of depth 5(≤6 guesses) that yielded a strategy to solve the Wordle puzzle. However, a decision tree of depth 4 is well beyond the scope of its computational resources. Therefore, it remains unknown whether all Wordle puzzles can be solved in 5 guesses[1].Luckily, Alex Peattie outlined why all Wordle puzzles cannot be solved in 5 or fewer guesses - thus identifying 6 as the minimum number of guesses needed to guarantee a win[2].Later, Jonathan Olson, Peter Tseng, Alex Selby, Mark Fisher et al. finally determined by finding different decision trees that: In the case of the 2309 words identified by Wordle, the problem can be solved in an average of 3.4201 guesses, with a worst case of 5 guesses, and in difficult mode, it can be solved in an average of 3.5076 guesses, with a worst case of 6 guesses; and for a complete dictionary of 12972 words, the Wordle can be solved in an average of 4.07771 guesses, with a worst case of 6 guesses, and in the hard mode, it can be solved in 4.52629 guesses on average, with 7 guesses in the worst case, that is, Wordle cannot solve 100 % within 6 guesses in the hard mode[3]. And 3blue1brown proposed a strategy using information entropy in one of his videos on wordle and simulated an average number of guesses of 4.124, which also provides us with ideas. In this paper, an ADF test was performed, ACF and PACF plots were drawn, and an ARMA (1,1) model was developed to predict the number of reports, a relatively simple algorithm that does not require other external variables. This article also extracted two attributes, the number of repeated letters and information entropy. The relationship between the attributes and the score variance was investigated by Pearson correlation analysis, and a word attribute analysis model was established. Finally, this paper used the One-hot method to encode words, applied the BERT neural network architecture, and used the SGD algorithm combined with MSE errors for optimization to build a neural network prediction model to analyze seven guess proportions for five letters. The BERT network has a stronger feature extraction capability, better performance and higher accuracy rate.

2. Report quantity prediction model

2.1. Trend analysis and prediction algorithm selection

The relationship between Number of reported results and Data for the 359 data is shown as Figure 2.

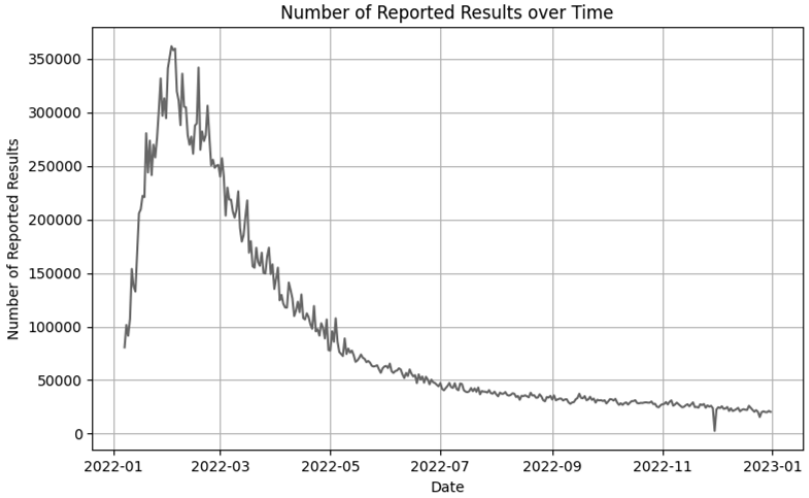


Figure 2. Relationship between Number of reported results and Data

It was observed that the number showed an essentially linear increasing trend until 2022-2-4. And after that, there is a slow decaying trend. So, we used the ARIMA method to predict.

Firstly, the observed system time series data are processed to remove remove words whose length is not equal to five or contain non-English letters. After that, the ACF plot of the data is drawn to observe whether it is a smooth time series. If the autocorrelation coefficient of the ACF plot is greater than 0 for a long time, it means that the series has a strong long-term correlation and the trend is not smooth, so the d-order difference operation is done to transform it into a smooth time series. Finally, the autocorrelation coefficient ACF and the partial autocorrelation coefficient PACF are obtained separately for the smooth time series, and the optimal stratum p and order q are obtained by analyzing the autocorrelation and partial autocorrelation plots to be used for the ARIMA model fitting for subsequent prediction [4, 5]. This process can be represented in Figure 3.

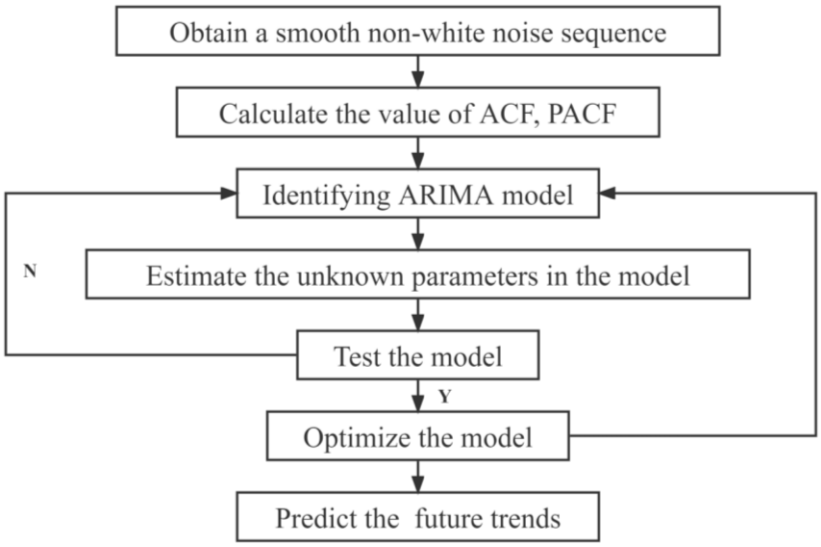


Figure 3. ARIMA Model

2.2. Solution of ARIMA algorithm

An ADF test was performed on the data and the results are shown in Table 1.

Table 1. Quantity-ADF test table

Difference order	t	p	Threshold value 1%	Threshold value 5%	Threshold value 10%
0	-3.867	0.002	-3.450	-2.870	-2.571

The t-statistic for the ADF test of the time series data was -3.867, the p-value was 0.002, the 1%, 5% and 10% thresholds are -3.450, -2.870, and -2.571, respectively. $p=0.002<0.01$, with higher than 99% certainty of rejecting the original hypothesis, at which point the series is smooth. Therefore, the ADF test can be satisfied at the 0th order difference.

The ACF and PACF plots of the data were then plotted, as shown in Figures 4 and 5.

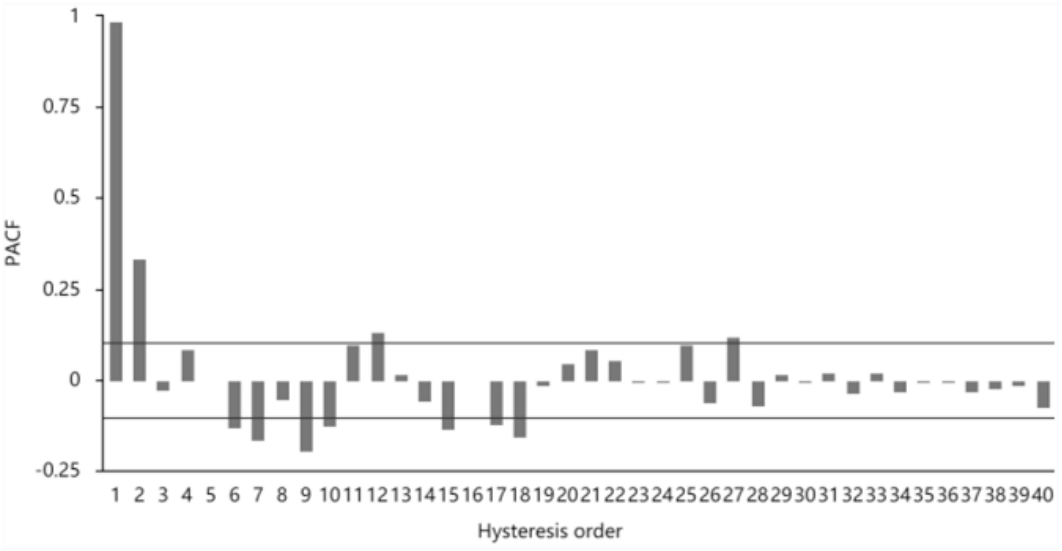


Figure 4. ACF

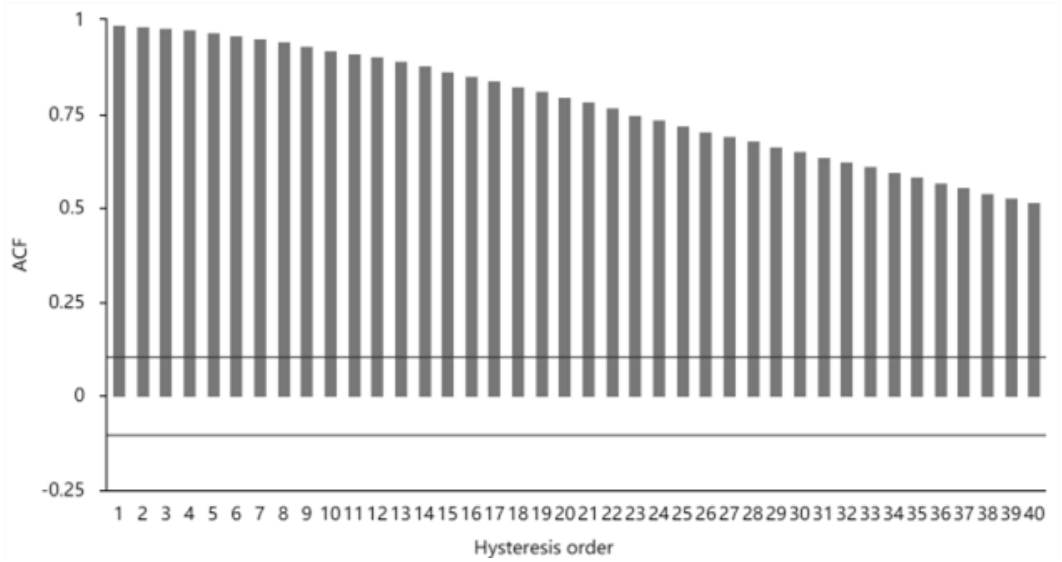


Figure 5. PACF

Combining the above two figures and minimizing the AIC, BIC values as much as possible, we choose the ARMA (1, 1) model for forecasting. The details are shown in Table 2.

Table 2. ARIMA model parameters table

	Symbol	Coefficient	Standard error	z	p	95% CI
Constant term	c	-287.292	660.285	-0.435	0.663	-1581.428 ~1006.844
AR Parameters	α1	-0.300	0.024	-12.687	0.000	-0.346 ~-0.254

The AIC value was 7761.950 and the BIC value was 7773. 591.The model equation is:

$$y(t) = -287.292 - 0.3y(t-1) \quad (1)$$

The predicted results are then given, as shown in Figure 6.

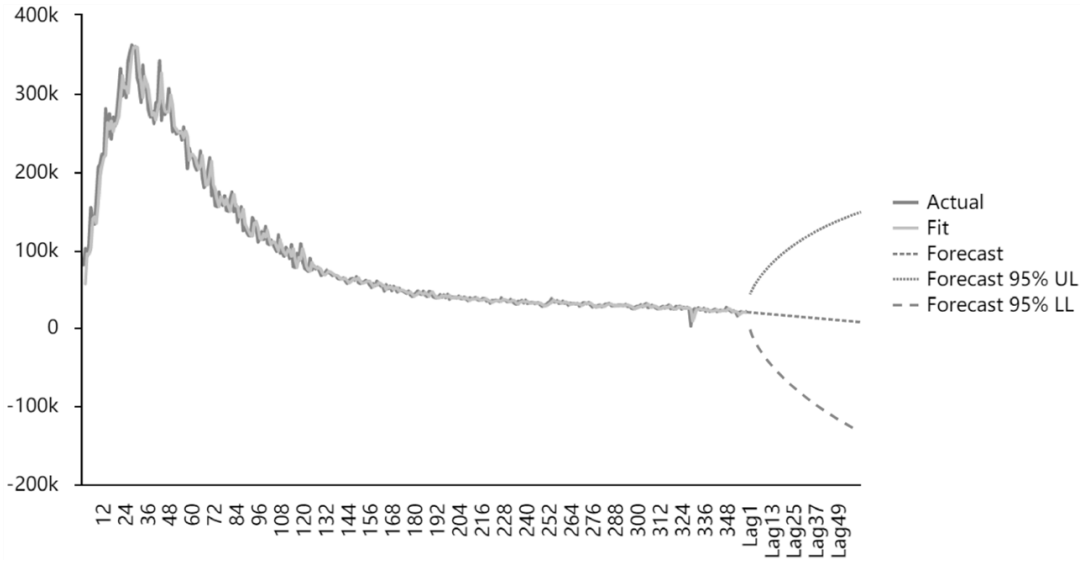


Figure 6. Fitting and prediction for quantitative models

Based on the chart above and the model equation, the report forecast the number of results reported on March 1, 2023 to be 7260.

3. Results

3.1. Information entropy and repeated letters

Information entropy is a description of the uncertainty of a thing and is often used as a quantitative indicator of the information content of a system. The amount of information and the probability are negatively correlated; the smaller the probability, the greater the amount of information. We can calculate the information entropy of each word from the information entropy formula based on the probability of each letter appearing in the message.

$$H = -\sum p_i \log_2(p_i) \tag{2}$$

p_i is the probability of each letter appearing in the message.

For repeated letters, we calculate the difference between the total number of letters and the number of letter types for each word. For example, if the word "cacao" has three letters "c", "a" and "o", but the total number of letters is five, the output is 2. By analogy, we calculate the difference for all words.

3.2. Analysis of experimental results

Next, the report discusses the effect of word attributes on the results [6].

The frequency of occurrence of each letter in the 359 days was calculated by eliminating the interfering items such as spaces in individual words and the letter "ï", as shown in Table 3.

Table 3. Frequency of letters

Letter	Frequency	Letter	Frequency	Letter	Frequency
a	8.8%	j	0.2 %	s	4.8%
b	1.7%	k	1.9 %	t	7.2%
c	4.0%	l	6.2 %	u	3.6%
d	3.0%	m	3.1 %	v	1.4%
e	10.3%	n	4.9 %	w	1.7%
f	1.8%	o	7.4 %	x	0.4%
g	2.8%	p	3.6%	y	3.4%
h	3.9%	q	0.3 %	z	0.3%
i	5.7%	r	7.5 %		

We also calculated the difference of all words. Now, it is needed to test the correlation between the information entropy, the repetition rate of the letters and the percentage of words in the seven guesses. Since these seven percentages can be approximated as normally distributed and always sum to 100, their variance can be used to represent the seven sets of variable information.

Next, pearson correlation analysis was used to investigate the correlation between the variance and entropy, difference terms respectively. The pearson correlation coefficient indicates the strength of the correlation. The calculation formula is as follows [7].

$$r = \frac{\sum (t - \bar{t})(y - \bar{y})}{\sqrt{\sum (t - \bar{t})^2 \sum (y - \bar{y})^2}} \tag{3}$$

The calculation results are shown in Table 4.

Table 4. Results of Pearson correlation analysis

	Variance
entropy	-0.194**
difference	0.114*
*p<0.05 **p<0.01	

The correlation coefficient between variance and entropy is -0.194 and shows a significance at 0.01 level, thus indicating a significant negative correlation between variance and entropy. The correlation coefficient between variance and difference was 0.114 and showed a significance at 0.05 level, thus indicating a significant positive correlation between variance and difference.

Consequently, the attributes of the words affect more significantly the reported percentage of scores played in difficult mode.

4. Neural network prediction model

4.1. One-hot coding and BERT neural network architecture

For the second question to predict the percentage of the ins based on the given words, we have the following idea.

Firstly, read in and pre-process the data, after removing words that are not equal to five in length or contain non-English letters, the training and testing data sets are divided in the ratio of 8:2.

Then, encode the words by using the one hot encoding method. The specific steps are to create 130 all-zero vector, then set the index dimension corresponding to the corresponding letter at each position to 1 and keep the other elements unchanged to obtain the final one-hot vector. After that, we adopted the BERT neural network architecture, which only used the encoder part of the classical Transformer architecture, and completely discarded the decoder part [8, 9]. The specific architecture is shown in Figure 7.

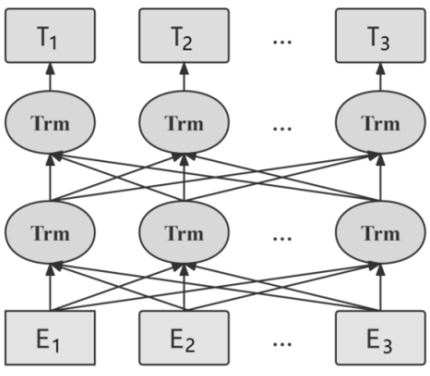


Figure 7. Structure of Bert network

The error function of the training and test sets uses the MSE mean squared error, which is the mean of the sum of squares of the errors of the corresponding points of the predicted and original data, and is calculated as [10]

$$MSE = \frac{SSE}{n} = \frac{1}{n}\sum i = 1^{m} \omega i \ （yi - \hat{y}i）^2 \tag{4}$$

The MSE mean square error is used as the error function between the training set and the test set, and the SGD algorithm is used with the learning rate set to 0.01.

$$W \leftarrow W - \eta \frac{\partial L}{\partial W} \tag{5}$$

The weight parameter to be updated is denoted W, and the gradient of the loss function with respect to W is denoted $\frac{\partial L}{\partial W}$. η is the learning rate.

The GPU is invoked to improve model training speed.

4.2. Training and testing of neural network

After the first training, the error line graph of the model on the test set under different epochs is shown in the Figure 8.

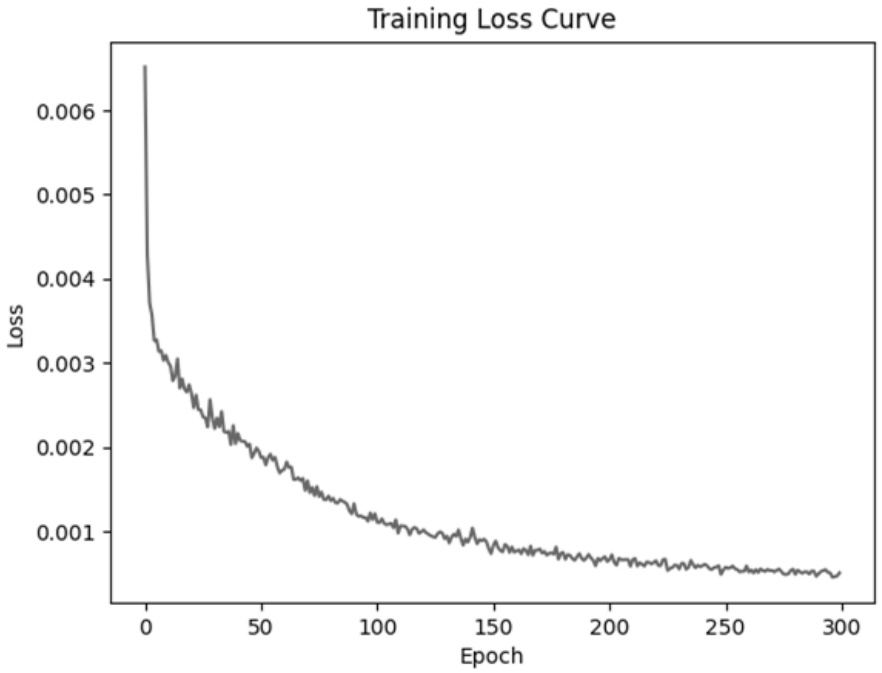


Figure 8. Training Loss Curve

After that, 300 epochs with relatively stable error are selected as the epoch of formal training. The loss function curve of the training set is drawn as shown in the Figure 9.

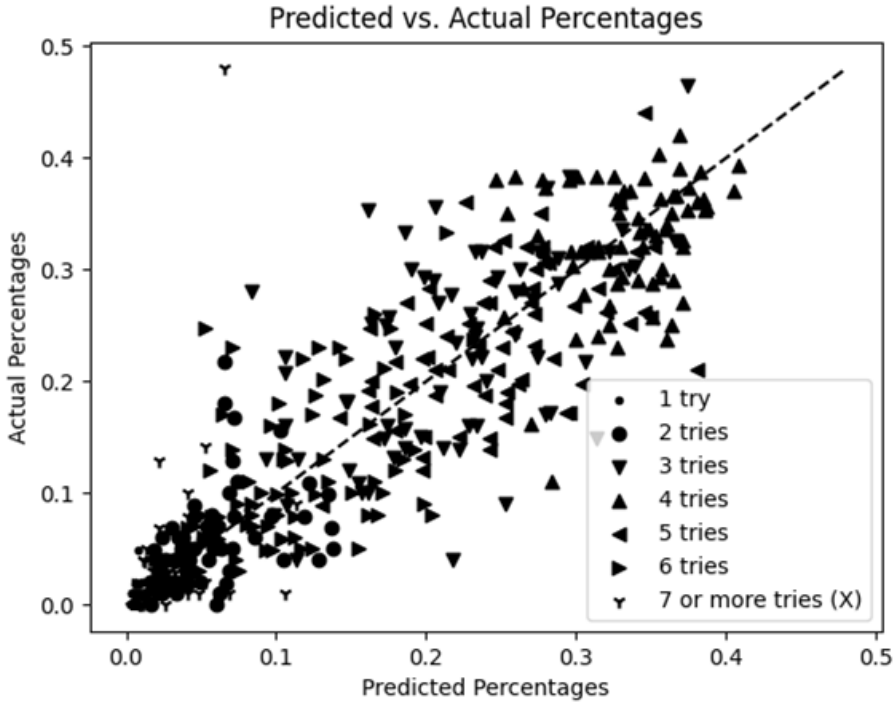


Figure 9. Predicted Percentages VS Actual Percentages

The final obtained training set Loss is 0.000509, and test set Loss is 0.000737. As expected, the credibility of the model is high.

Finally, we plot the performance of this model on the test set data.

The dots in different colors in the graph represent different number of attempts, the horizontal coordinate is the prediction percentage, the vertical coordinate is the true percentage, and the closer the dot is to the 45-degree axis means the model is about accurate. By analyzing this graph, it is clear that: the prediction is more accurate.

For the word "EERIE", the word was converted to lowercase and entered into the neural network model for prediction, and the seven percentages were 1.16%, 5.75%, 20.68%, 32.60%, 24.72%, 12.39%, 2.70%.

5. Conclusion

Wordle is a popular daily puzzle game where players try to guess a five-letter word in six tries or less. The report develops predicting models for the distribution of the reported results and the number of reported results. It can improve the game design and provide insights into user behavior.

The report first preprocessed the data, then built an ARMA (1,1) prediction model by ADF test and plotting ACF and PACF graphs, and calculated that there would be 7260 reported results on March 1, 2023. In addition, we extracted two attributes, the number of repeated letters and the information entropy. By Pearson correlation analysis, the report investigated the relationship between attributes and the variance of scores. It can be noted that there is a significant negative correlation between entropy and variance, and a significant positive correlation for Number of repeated letters. Consequently, the word attribute will significantly affect the percentage of scores reported that were played in Hard Mode.

The report also used the One-hot method to encode the words, applied the BERT neural network architecture and used the SGD algorithm for optimization to predict the distribution of the reported results. For the word EERIE, the percentages were calculated to be 1.16%, 5.75%, 20.68%, 32.60%, 24.72%, 12.39% and 2.70%. By calculating the MSE error, it is calculated that the loss of the training set is 0.000509 and the loss of the testing set is 0.000737, which is a high confidence level of the model.

References

[1] Poirrier, Laurent. (2022) Laurent's Notes, Mathematical optimization over Wordle decision trees -- Laurent's notes.

[2] Peattie, Alex. (2022) Establishing the minimum number of guesses needed to (always) win Wordle, Alex Peattie -- Building software and startups, based in London.

[3] Poirrier, Laurent. (2022) Laurent's Notes, Wordle-solving state of the art: all optimality results so far -- Laurent's notes.

[4] Feng S. M., Yan L., Zhang Y. F. et al. A combined prediction model based on ARIMA-Intervention-LSTM [J]. Journal of Hebei University of Technology, 2023, 52 (02): 28 - 34.

[5] Ma Xia, Zhang Jinzhu, Li Can et al. Teaching case design of ARIMA model with application of R software [J]. Fujian Computer, 2022, 38 (11): 113 - 117.

[6] Zhang Zihao. A comparative study of technical differences among Chinese Super League teams based on Pearson correlation analysis [J]. Science and Technology Innovation and Application, 2020, No. 307 (15): 51 - 54.

[7] Liu Chao, Zhang Ailin, Li Shugang et al. LSTM working face gas concentration prediction model based on Pearson feature selection and application[J/OL]. Coal Science and Technology: 1 - 9 [2023-05-20].

[8] Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." (2018).

[9] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need [J]. arXiv, 2017.

[10] Qian Shenghua. An automatic scoring system for subjective questions based on twin networks and BERT model [J]. Computer Systems Applications, 2022, 31 (03): 143 - 149.