Predicting the Social Craze

Summary

Last year, a word-guessing game called *Wordle* had gone from an intriguing game to a social craze. In this game, players have a daily chance to guess a five-letter within six tries. By utilizing the data submitted by players on *Twitter*, we have developed a time series model by **Prophet algorithm** to predict the trend of the number of reported results. In addition, we have established models for evaluating word difficulty and predicting the distribution of the number of tries in hard mode.

For the first problem, we compared the performance of **triple exponential smoothing** algorithm with the Prophet algorithm. Our results show that the former method can only predict the short-term variations in the number of reported results, while the latter method can make more accurate predictions, including taking into account the impact of holidays. Thus, we generated reasonable forecast interval through the Prophet algorithm.

For the second problem, we adopted **ridge regression** model to predict the distribution of percentage of the reported results. We assume that the distribution of percentage relies on the attributes of the word itself, such as the number of vowels. Therefore, by extracting feature vectors from words and fitting the data using ridge regression, we achieved a high level of accuracy in the model's prediction. And its pattern is similar to the original data. For the word "EERIE", we estimate that most people (over 80%) can win the game by guessing four to six tries.

For the third problem, we defined the difficulty of a word based on the distribution of percentage of reported results. It is divided into five levels from the hardest words rated as five to the easiest words rated as zero. Inspired by **word embedding** and **association rules**, we establish a refined form of feature vector to represent word itself, which is different from that in problem two. Finally, we obtained satisfying rating results by implementing a **BP neural network**. For the word "EERIE", it was rated as 4.31, indicating that the word is relatively difficult. This conclusion is intuitive because the letter "E" appears three times in the word, confusing players and increasing the difficulty.

For the last problem, we found an intriguing feature when analyzing the data file, that is the great impact of holidays and other social events. This confirmed our assumption that no other significant external factors except holidays that will influence the user data in the file. It’s also one of the reasons why we implement the Prophet algorithm.

Keywords: Triple Exponential Smoothing, Prophet Algorithm, Ridge Regression, Neuron Network, Word Embedding, Association Rules

Contents

1 Introduction 3

1.1 Problem Background 3

1.2 Restatement of the Problem 3

1.3 Our Work 4

2 Data Pre-Analysis 5

3 Assumptions and Justifications 5

4 Analysis and Modelling 6

4.1 Prediction of Reported Results 6

4.1.1 Data Analysis 6

4.1.2 Triple Exponential Smoothing 7

4.1.3 Prophet Algorithm 7

4.1.4 The Solution of Triple Exponential Smoothing 8

4.1.5 The Solution of Prophet Model 10

4.2 Predicting the Probability Distribution by Using Ridge Regression 11

4.2.1 Data Analysis 11

4.2.2 Establishment of the Word Feature Vector 12

4.2.3 Ridge Regression 12

4.2.4 The Solution of Ridge Regression 13

4.3 Difficulty Criteria Inspired by Word Embedding and Association Rules 14

4.3.1 The Extraction of Word Feature Vectors 14

4.3.2 The Establishment of the Backpropagation Neuron Network 15

4.4 Other Intriguing Features 16

5 Conclusion 17

5.1 Strengths and Weaknesses 17

5.2 Further Discussion 17

References 18

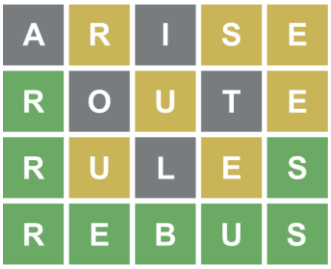
Appendices 19

1 Introduction

1.1 Problem Background

Wordle is an elegant word-guessing game offered by the *New York Times*. It was released in October 2021. Since then, it had taken the Internet by storm. The gameplay is simple: players have six attempts to guess an unknown five-letter word. For each letter in the guessed word, it is marked in the form of the following:

- Green: indicates the letter appears in the same position in the target word
- Yellow: indicates the letter appears in the target word, but in a different position
- Grey: indicates the letter does not appear in the target word



A	R	I	S	E
R	O	U	T	E
R	U	L	E	S
R	E	B	U	S

Figure 1.1.1

Using this feedback, players can adaptively choose a sequence of up to six words. Players win if they correctly guess the target word within six guesses, and they lose otherwise. The game has a “Hard Mode” option, which requires players to include the correct letters marked as green and yellow in subsequent guesses. *Wordle* has a single daily solution the same for everyone.

Obviously, the difficulty of *Wordle* varies depending on the daily word. Plenty of players report their score on Twitter. At the same time, much detailed information about these reports is included in the problem attachments. In order to help predict its popularity and the subsequent maintenance, it is particularly useful to build a mathematical model to analyze the data.

1.2 Restatement of the Problem

Currently, ***MCM*** has collected a dataset of daily results submitted by the players of *Wordle*. This is the starting point for our exploration, in which we need to meet the following requirements:

- A model to account for the change in the number of reported results in both standard and hard mode. We need to predict the range of reported results on March 1, 2023 by using this model.

- Discuss whether any properties of the word affect the distribution of marks reported in Hard Mode. Building a model to predict the distribution of marks and evaluate its level of precision. Predict the distribution of scores of the word EERIE released in March 1, 2023.

- A model to classify solution words by difficulty and analyze its degree of precision. Explain the difficulty of the word EERIE.
- Discover other intriguing features of the data set.

1.3 Our Work

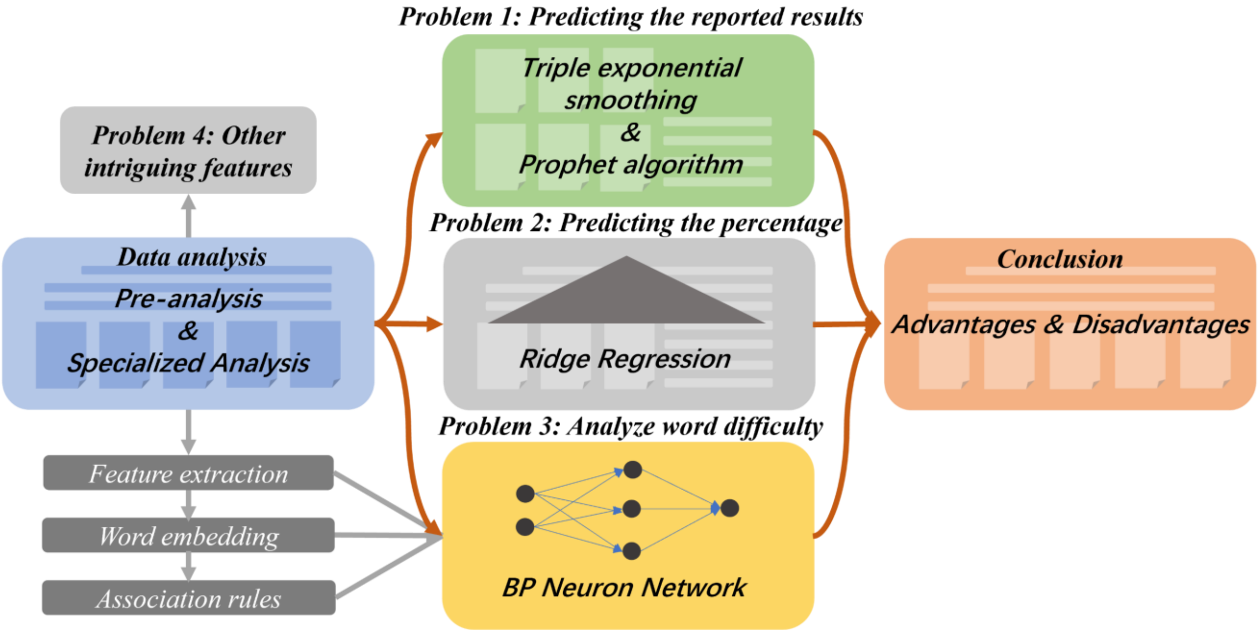

Problem 1: Predicting the reported results
Triple exponential smoothing & Prophet algorithm
Problem 4: Other intriguing features
Problem 2: Predicting the percentage
Data analysis
Pre-analysis & Specialized Analysis
Ridge Regression
Conclusion
Advantages & Disadvantages
Problem 3: Analyze word difficulty
Feature extraction
Word embedding
Association rules
BP Neuron Network

Figure 1.3.1: Overview of Our Work

Regarding the first and fourth question, we closely examined the data and came up with two initial models. One is the exponential smoothing method in classical time series models, which has a high degree of fitting, but it has a large deviation for the long-term forecast required by the question and cannot meet the requirements. Therefore, we used the prophet algorithm-based forecasting model, which has some fluctuation features and a certain periodicity in time change. Other interesting features is generated as a by-product when modeling.

Regarding the percentage distribution prediction in the second question, we believe that word attributes have a higher correlation with the percentage and are not sensitive to time changes. Therefore, we consider extracting relevant features of the words to form a feature vector as input to explore the relationship with the percentage distribution. Considering the randomness and fluctuation of the game itself, we decided to use the ridge regression algorithm that can discard some information to seek a more accurate and effective curve and reduce the model's uncertainty factors.

Regarding the third question, we used the percentage distribution as the basis for determining the difficulty rating. After defining the difficulty standards and preprocessing the data, we searched for word attributes with higher correlation with difficulty to construct a feature vector as input to a multi-input, single-output neural network model to fit the relationship between our constructed feature vector and difficulty.

2 Data Pre-Analysis

Data reliability must be ensured before modeling. After examining the official file (Problem_C_Data_*Wordle*.xlsx), we find the numerical and textual information are all collected within a year, from January 7, 2022 to December 31, 2022. To solve problems found in the data and establish a reliable data set, we carried out the following measures:

Remove irrelevant data. We notice that the file contains a small number of four-letter-words. Considering that *Wordle* is a five-letter-word guessing game, we weed out those incorrectly formatted words.

…	Word	…	Number of reported results	Number in hard mode	…
…	clen	…	26381	2424	…

Table 2.1: Example of four-letter-words

After summing up all the percentages of (1, 2, 3, 4, 5, 6, X), we notice that some of the sums are far greater than 100%. In order to make them reliable, we normalize those data.

…	Word	…	1 try	2 tries	…	7 or more tries	Sum
…	nymph	…	1%	2%	…	9%	126%

Table 2.2: Example of data sums are far greater than 100%

3 Assumptions and Justifications

Assumption: The data provided in this problem is valid and reliable.
Assumption: The trend of reported results will continue in the future. In other words, this trend has no sudden changes or disruptions in the future.

Justification: plot the historical data of reported results over time and check for any long-term trends. If the trend is relatively stable and there are no major disruptions, this assumption is more likely to hold true.

Assumption: There are no significant external factors except holidays that will influence the user data in the file, including the number of the reported results and the percentage of players who guess the word in each number of tries.

Justification: To ensure accurate trend prediction, it is necessary to eliminate unpredictable external factors. Comparing to the data in the file, we found that *Wordle*'s significant updates had little impact on reported scores or percentage distribution[1]. We also considered the impact of holidays and news events and found two specific days that related with reported scores. These days were considered as holiday when building the model (Section 5.1.1). However, we reconsider this as an interesting feature and explain it in Section 5.4.

Assumption: The percentage of players who guess the word in different numbers of tries is a representative and unbiased sample of the broader *Wordle* player population.

Justification: If this data is biased, our model's predictions made may not accurately reflect the real condition.

Assumption: To predict more accurately, we assume that the percentage of players who guess the word in each number of tries will remain relatively stable.

Justification：We plotted the percentage of players in each category (1, 2, 3, 4, 5, 6, X) over time to look for any patterns or trends (Section 5.2.1). The percentages remained stable over time, suggesting that this assumption is valid.

Assumption: When playing *Wordle*, winning the game with only one attempt is pure luck, so there is no predictive or referential significance.

Justification：According to probability theory, **t**here are 6 to the power of 6 (46,656) possible guess combinations. Thus, the probability of winning on the first guess is 1/46,656, which is about 0.002%, making it unlikely to succeed in the first try. However, using strategies and techniques can improve success rate in subsequent guesses, so success does not rely solely on luck[8].

Assumption: The difficulty level of the word in *Wordle* is solely determined by the word itself, and is not affected by other factors such as the time of day, day of the week, or other external factors.

4 Analysis and Modelling

4.1 Prediction of Reported Results

4.1.1 Data Analysis

For problem one, we extracted the number of reported results and the number in hard mode from the data file.

The two figures below show the trends of the number of reported results and the number in hard mode from January 7, 2022 to December 31, 2022. Overall, both curves exhibit a rise-and-fall trend and tend to be stable afterwards. By observing two figures, we notice that there are significant sharp drops in both curves at two certain points. We assume these drops are caused by holiday events when building the model. However, we will account for these events in Section 5.4.

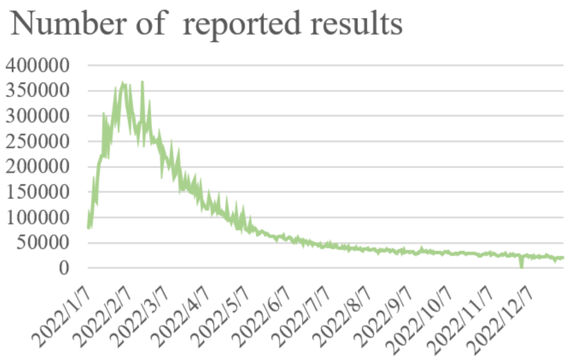


Figure 4.1.1

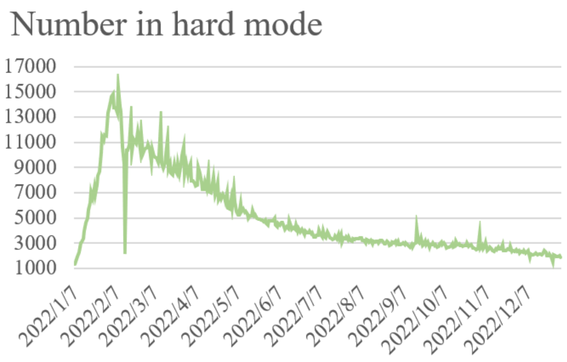


Figure 4.1.2

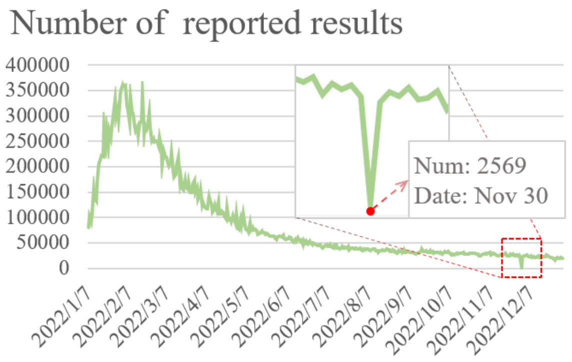


Figure 4.1.3

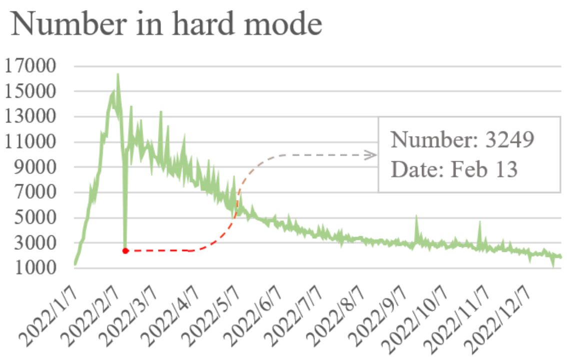


Figure 4.1.4I

4.1.2 Triple Exponential Smoothing

When looking at the number of reported results in the last three quarters, we notice a steady decline. And its trend is a quadratic curve.

Hence, we use triple exponential smoothing to predict reported results[7]. This method gives the past data a declining weight, and the weighted average of observation is carried out in chronological order.

Generally, the influence of historical data on the future decreases with time. Therefore, this method is practical and intuitive, and the prediction model is listed as follows:

$$\begin{cases} S_t^{(1)} = \alpha y_t + (1-\alpha)S_{t-1}^{(1)} \\ S_t^{(2)} = \alpha S_t^{(1)} + (1-\alpha)S_{t-1}^{(2)} \\ S_t^{(3)} = \alpha S_t^{(2)} + (1-\alpha)S_{t-1}^{(3)} \end{cases} \tag{1}$$

Where $S_t^{(1)}$, $S_t^{(2)}$ and $S_t^{(3)}$ is the basic, double and triple weighted average of the current observation, α is the smoothing factor and $0 < \alpha < 1$.

$$y_{t+m} = a_t + b_t m + c_t m^2,\ m = 1,\ 2,\ \cdots, \tag{2}$$

Where

$$\begin{cases} a_t = 3S_t^{(1)} - 3S_t^{(2)} + S_t^{(3)}, \\ b_t = \dfrac{\alpha}{2(1-\alpha)^2}[(6-5\alpha)S_t^{(1)} - 2(5-4\alpha)S_t^{(2)} + (4-3\alpha)S_t^{(3)}], \\ c_t = \dfrac{\alpha^2}{2(1-\alpha)^2}[S_t^{(1)} - 2S_t^{(2)} + S_t^{(3)}]. \end{cases} \tag{3}$$

4.1.3 Prophet Algorithm

By considering factors such as trends over time and holidays comprehensively, we also carried out the Prophet model to predict the number of reported results and the number in hard mode. The formula for this model is as follows:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{4}$$

Where $y(t)$ is the target variable at time t, $g(t)$ represents the trend, $s(t)$ represents the seasonal change, $h(t)$ represents the influence of holidays, and ϵ_t represents the loss. In this model, we only take into account trends over time and holidays, which are:

$$g(t) = \sum_{i=1}^{k} \beta_i f_i(t) \tag{5}$$

Where $f_i(t)$ represents the k base functions for the trend and β_i is the coefficient for each base function. Typically, the trend component can be modeled using a linear or non-linear function form, and the Prophet model uses a logarithmic linear function, which can better handle the exponential growth characteristics of the target variable.

$$h(t) = \sum_{j=1}^{J} k_j I(t \in [t_j - \tau, t_j + \tau]) \tag{6}$$

Where t_j is the j-th holiday, k_j is the impact coefficient of holiday, τ represents the duration of holiday's impact, and $I(\cdot)$ is an indicator function.

Based on the equation listed above, we developed our model, shown as Figure 5.1.3.1 and Figure 5.1.3.2:

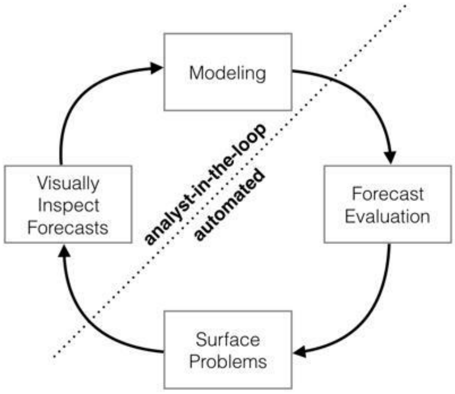


Figure 4.1.3.1: Prophet's Flow Chart

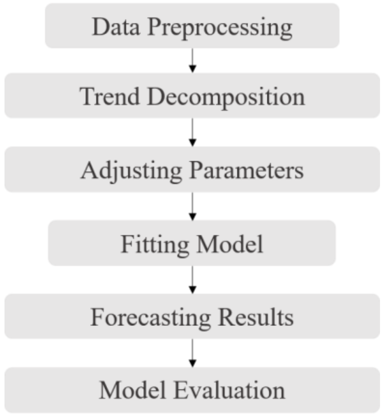


Figure 4.1.3.2: Overall Establishment

4.1.4 The Solution of Triple Exponential Smoothing

We implement the triple exponential smoothing method , we predicted the number of both variables for March 1, 2023. However, we cannot generate the predicted interval. Figure 4.1.4.1 and Figure 4.1.4.2 demonstrate the trend of the number of reported results and the number in hard mode.

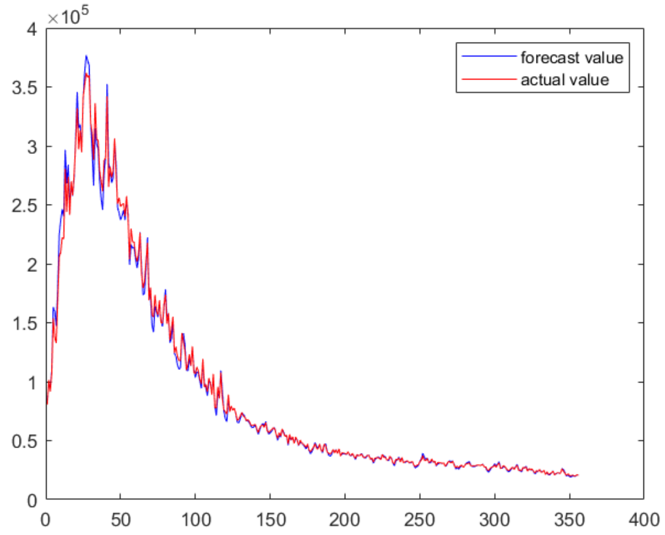


Figure 4.1.4.1: The prediction of reported results

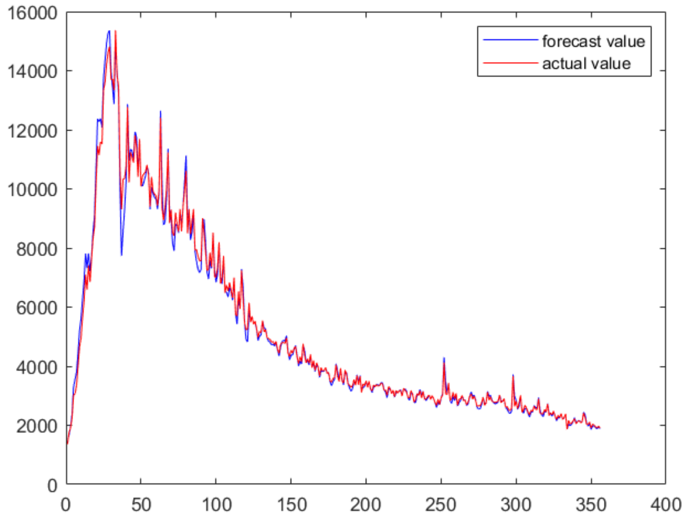


Figure 4.1.4.2: The prediction of the number in hard mode

Date	Number of reported results	Forecast interval
2023-03-01	106403.738	NAN

Table 4.1.4.1

Date	Number of reported results	Forecast interval
2023-03-01	1302.379	NAN

Table 4.1.4.2

We found that the triple exponential smoothing method perfectly fit the trend of the variation of the original data. However, we notice that the decline of the curve is relatively slight at two specific dates, which is February 13 and November 30 mentioned at Section 4.1.1, indicating that it is unsuitable for predicting the effects of holidays. As a result, this method cannot solve problem due to these defects, instead, we adopt the Prophet Model to generate our final predicting results.

4.1.5 The Solution of Prophet Model

By using the Prophet model, we predicted the interval of the number of both variables for March 1, 2023. Figure 5.1.5.1 and Figure 5.1.5.2 demonstrate the trend of the number of reported results and the number in hard mode.

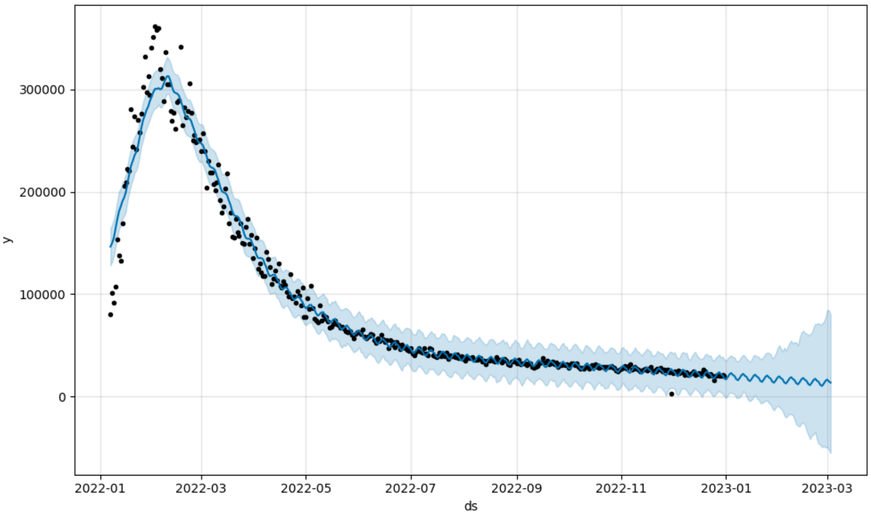


Figure 4.1.5.1: The prediction of reported results

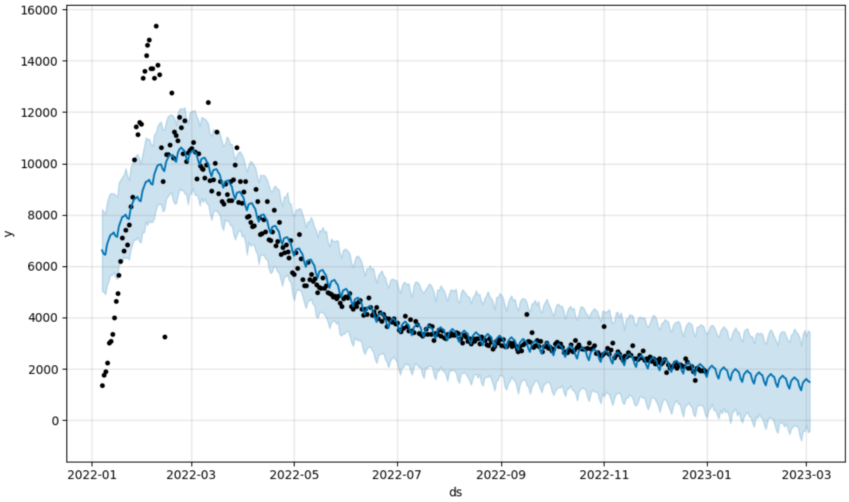


Figure 4.1.5.2: The prediction of the number in hard mode

Figure 5.1.5.1 shows the trend in the number of reported results, which is estimated to be 16,468 on March 1, 2023 and the length of interval is up to 84,385

Date	Number of reported results	Forecast interval
2023-03-01	16468.671	[-49371.860, 84385.713]

Table 4.1.5.1

According to Figure 5.1.5.2, the number of submissions in hard mode on March 1, 2023 is predicted to be 1604, but obviously it has a wide range of fluctuations.

Date	Reported results in hard mode	Forecast interval
2023-03-01	1604.589	[-298.659, 3587.532]

Table 4.1.5.2

Consequently, it can be noticed that our model better fits the trend of the number of reported results and the number in hard mode despite the forecast range fluctuates widely. However, due to the short time of prediction, we believe that the predicted value of this model is reasonable. While in the long-term, it is not difficult to speculate that the model has large deviations in prediction.

4.2 Predicting the Probability Distribution by Using Ridge Regression

4.2.1 Data Analysis

Since there are only 359 data points (lexical and numerical data) in the file, which is relatively limited, it may lead to unstable regression parameters. Therefore, to handle correlated features and achieves better performance, we adopted the Ridge Regression model.

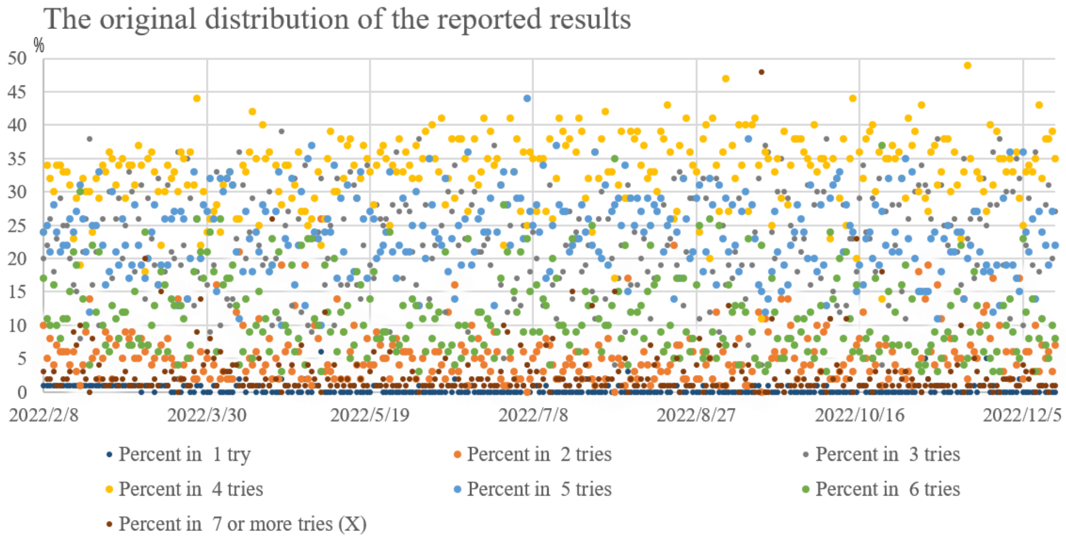


Figure 4.2.1.1: The original distribution of the reported results

By observing the Figure 5.2.1.1, we notice that the variation of time had little impact on reported scores or percentage distribution, indicating that there are no significant external factors that will influence the user data in the file. In other words, a word appears time-independent and corresponding distribution should be almost the same.

In order to eliminate the influence of the scale of features, we standardized the data by using the following method: This is because the impact of the regularization term is influenced by the scale of the features, and standardization can eliminate this influence.

4.2.2 Establishment of the Word Feature Vector

Since it can assume that time has little impact on the distribution of percentages, we believe only the attributes of the word itself can affect this distribution. Thus, we can build a more reasonable prediction model.

The similarity and discriminability of individual words is essential for feature extraction, which refers to the shared features of different words and the unique features of specific words. For each word in the file, we constructed its feature vector $\boldsymbol{x}$ in dataset $\boldsymbol{X}$ as follows[5]:

$$\boldsymbol{x} = [x_0,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6] \tag{7}$$

x_0 is the bias term;

$x_1, \cdots, x_5$ are the alphabetical positions of each letter in the word;

x_6 is the ratio of the number of vowels to the number of consonants in one word.

4.2.3 Ridge Regression

After determining the transformation relationship between words and vectors, we use the Ridge regression algorithm to fit the trend of the data. The goal of Ridge Regression is to find a set of weights $\boldsymbol{w}$ such that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$, here $\boldsymbol{X}$ is the feature matrix of all the words in the file and $\boldsymbol{y}$ is the target variable. In this problem, predicting the distribution of the reported results.

To avoid overfitting, we add a regularization term to the objective function of Ridge Regression, which adds an L2 regularization term[6]:

$$\min_{\boldsymbol{w}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}||^2 + \alpha||\boldsymbol{w}||^2 \tag{8}$$

Where α is the regularization coefficient that controls the complexity of the prediction model. A larger α will make the model tend to be simpler, while a smaller α will make the model tend to be more complex.

We aim that our model have the smallest possible variance without seriously affecting the accuracy of the model deviation, so we set the value of λ to 10.

The solution of Ridge Regression uses the following formula:

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{9}$$

Where I is an 7×7 identity matrix, and 7 is the number of features in each sample.

4.2.4 The Solution of Ridge Regression

Using the Ridge Regression, we have plotted the result of the distribution of percentage. Similar to the analysis of the original data, the corresponding distribution is almost the stable and remain constant over time.

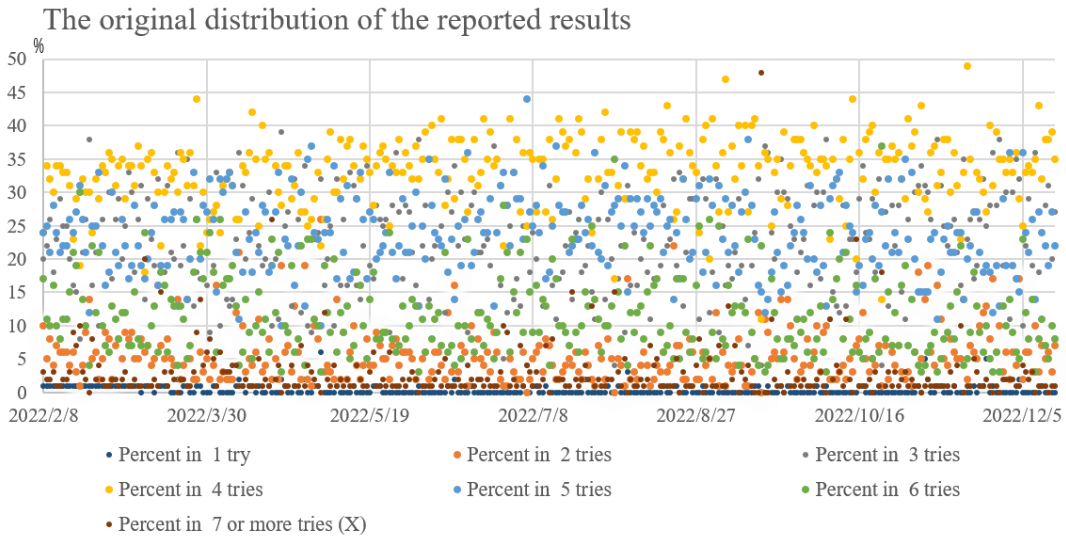


Figure 4.2.4.1

Our prediction for the word EERIE is [0.12，4.09，20.35，33.14，25.24，12.88，4.18]

It shows that the majority of players (over 80%) will make four to six guesses to win the game, indicating that this word is relatively easy.

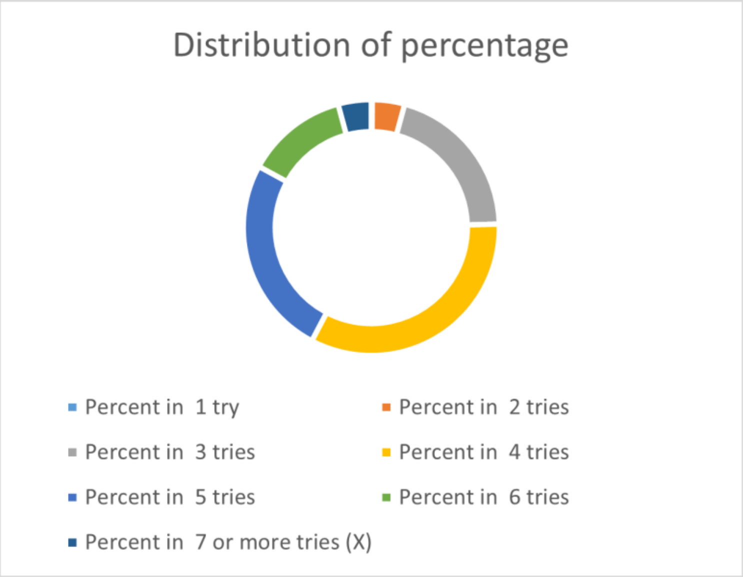


Figure 5.1.5.2: The prediction of the word “EERIE”

4.3 Difficulty Criteria Inspired by Word Embedding and Association Rules

To create a model to predict how hard the word in *Wordle* is, it is imperative to develop a connection model that relates the various attributes of a word to its difficulty level. We divide this process into two parts: constructing word feature vectors and establishing corresponding backpropagation (BP) neural network models.

4.3.1 The Extraction of Word Feature Vectors

Step 1, Defining the criteria for word difficulty:

Our approach is to use the percentage of players who guessed the word in six and more than seven tries as a proxy for the difficulty level, with higher percentages indicating harder words and lower percentages indicating otherwise, our definition is as follows:

- We use $sample_i, (i = 1, 2, \cdots N)$ to indicate the i-th sample word;
- And $sample_i[x_j], (\mathrm{j} = 1, 2, \cdots 7)$ represents the percentage of players who guessed the correct word or didn't win the game in j times;
- N is the total number of words in the sample;

Thus, we can get the moderate difficulty coefficient:

$$B = A / N \tag{10}$$

Where $A = \sum_{i=1}^{N} \sum_{j=6}^{7} sample_i[\mathrm{x_j}]$

The difficulty factor of any sample word is defined as follows:

$$C_i = \sum_{j=6}^{7} sample_i[\mathrm{x_j}] / B \tag{11}$$

Step 2, Defining the feature vector:

We employed the technique of word embeddings[3] to convert words into vectors.

To identify words with higher difficulty coefficients, we borrowed the definition of lift in association rules[2]. Specifically, we investigated the lift for features such as letters with low or high frequency in difficult words, and the frequency of repeated letters in difficult words.

Moreover, to identify suitable feature items, we set a lift threshold to identify the significance of word, if its lift exceeded, it is more important.

Ultimately, we select the following four features for each word:

$$\boldsymbol{x} = [x_0, x_1, x_2, x_3, x_4] \tag{12}$$

Where

x_0 is the bias term;

x_1 is the ratio of the number of vowels to consonants;

x_2 is the minimum frequency among the five letters in the word;

x_3 is the average frequency of the five letters;

x_4 is the maximum number of repeated letters.

4.3.2 The Establishment and Solution of Backpropagation Neuron Network

As demonstrated in Figure 5.3.3.1, our model is a 5-5-1 neural network[4], which outputs our difficulty coefficient.

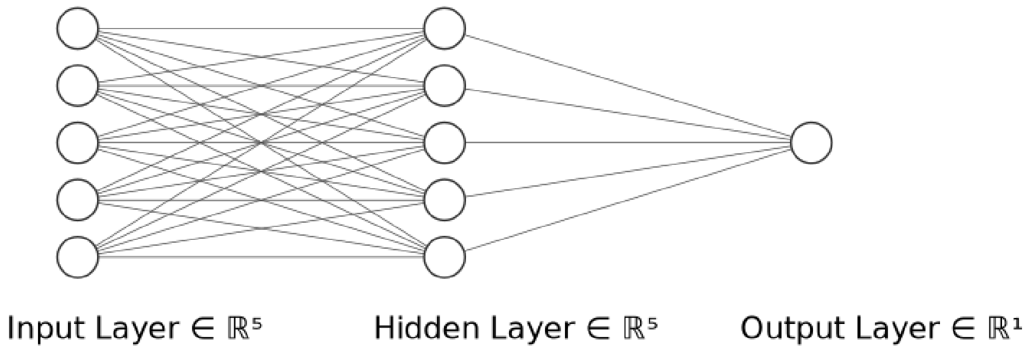


Figure 4.3.2.1

The corresponding mathematical expression is:

$$Y = f(\mathrm{W}^{\mathrm{T}}\mathrm{X}+\mathrm{b}) \tag{13}$$

Other features of our model is as follows:

- To preprocess the difficulty rating matrix Y, we performed range normalization so that each element was less than or equal to 1.
- We employed the sigmoid function as the activation function, as its range is limited to values between 0 and 1.
- we utilized stochastic gradient descent (SGD) with momentum-based learning rate changes to minimize errors caused by local optima.
- Since we set the moderate difficulty coefficients as 1, we multiplied the final output Y value by the rating range. In this way, the out put is a one to five real number, corresponding to our definition of difficulty.

Finally, the difficulty factor of the word 'EEIRE' is:4.79.

In order to verify the accuracy of the model, we randomly selected some words and calculated their difficulty coefficients. The comparison graph of word sizes based on the difficulty coefficients is as follows:

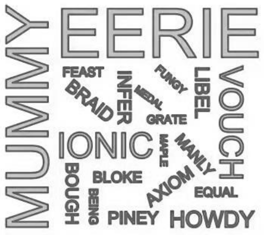


Figure 4.3.2.2

4.4 Other Intriguing Features

We visualize all the reporting results and the number in hard mode. There is a significant decline in the number of reported results on February 13 and November 30. These dates have special social significance. The Los Angeles Rams won the Super Bowl championship on February 13 and U.S. men's soccer team beats Iran to advance in World Cup on November 30. Understandably, people focused more on these big events and few played *Wordle*.

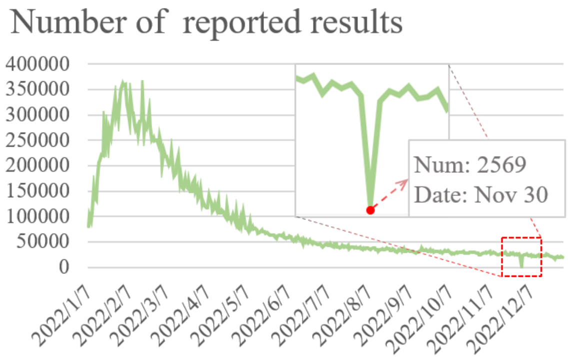


Figure 4.4.1

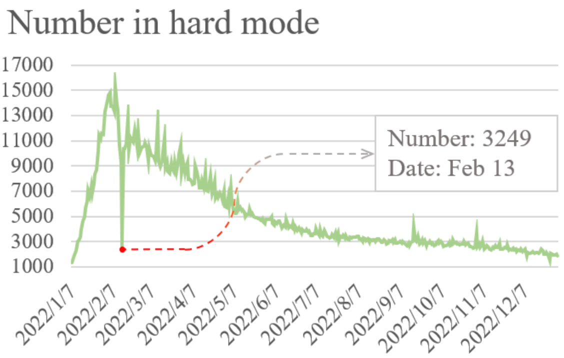


Figure 4.4.2

Figure 4.4.3: 2022 FIFA World Cup

Figure 4.4.4: Super Bowl LVI

5 Conclusion

5.1 Strengths and Weaknesses

Strength: Prophet algorithm can automatically identify relevant time series components such as holidays, which can be time-consuming. It also provides detailed information on each time series component, which can help users understand the drivers of the forecast.

> **Weakness:** Prophet algorithm has a wide range output, making it difficult to predict trends when data is inadequate.

Strength: Ridge regression shrinks the coefficients towards zero, thus reducing overfitting and improving generalization performance. It is also easy to implement with only a few modifications, thus guaranteeing the efficiency of modeling.

> **Weakness:** However, the coefficients produced by ridge regression are not directly comparable to the coefficients in a regular linear regression model, which can make interpretation difficult.

Strength: Backpropagation Neural Network can fit non-linear relationships between input and output data, making it suitable for solving modeling questions.

> **Weakness:** The internal workings of Backpropagation Neural Network are not easily interpretable, making it difficult to understand how the network arrives at its output. During actual testing, there was still a chance of encountering errors (about 3/10 probability) caused by local optima.

5.2 Further Discussion

For the ridge regression model, we did not consider the correlation between the feature items and the percentage distribution, which resulted in the selected features lacking distinctiveness, leading to underfitting during the fitting process of the ridge regression model. To improve this model, two points can be considered: first, select more features that are correlated with the output to compensate for the errors caused by underfitting; secondly, strengthen the screening of data to exclude more data that is irrelevant to the factors under consideration.

For the neural network model for the third question, we did not consider the correlation between the data adequately and used confidence instead of correlation. However, confidence cannot completely replace correlation because there may be cases where confidence is high but correlation is low. Secondly, our fitting model is a biased linear model, which may cause large errors for high-dimensional input data. Thirdly, the neural network model itself has the problem of local optima. Although we used stochastic gradient descent with momentum for learning rate changes, there is still a probability that the final result will fall into a local optimal solution, causing significant deviation from the expected result.

References

[1]. Tom's Guide. (2022, January 5). *Wordle* just announced rule changes: What you need to know. Tom's Guide. https://www.tomsguide.com/news/wordle-just-announced-rule-changes-what-you-need-to-know

[2]. Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In Proceedings of the 20th international conference on very large databases (pp. 487-499).

[3]. Taylor, S. J., & Letham, B. (2017). Forecasting at scale. The American Statistician, 72(1), 37-45.

[4]. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0

[5]. Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). Forecasting: methods and applications (3rd ed.). Wiley.

[6]. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

[7]. Holt, C. C. (1957). Forecasting seasonals and trends by exponentially weighted moving averages. Operations Research, 8(5), 663-675.

[8]. Bramley, N. R., Gürerk, Ö., & Möller, M. (2022). Wordle reveals the importance of strategy in solving complex problems. Nature, 602(7899), 300-304.

Appendices

Appendix 1

Introduction: Ridge Regression

```
X = ones(359,7);
for i = 1:359
    for j = 1:6
        X(i,j+1) = tezheng(i,j);
    end
end
[standard,ps] = mapstd(X);
loss = zeros(1,1000);
I = eye(7);
for i = 1:1000
    for j = 1:7
        lambda = i;
        model = inv(standard'*standard + lambda*I)*standard'*data3(1:end,j+5);
        y_pred = standard * model;
        loss(1,i) = mean((y_pred - data3(1:end,j+5)).^2) + loss(1,i);
    end
    loss(1,i) = loss(1,i)/7 ;
end
xvhao = min(loss);
lambda_opt = find(loss == xvhao);
for i = 1:7
    theta{i}= inv(standard'*standard+lambda_opt*I)*standard'*data3(1:end,i+5);
    y_forecast{i} = standard*theta{i};
end
word = [1,5,5,18,9,5,4];
for i = 1:7
    forecast(i) = word*theta{i};
end
sum = 0;
for i = 1:7
    sum = forecast(i) + sum ;
end
forecast = forecast/sum;
```