Predicting Wordle Results

Abstract

Wordle is a recent popular puzzle offered daily by the New York Times. It is a word-guessing game that has a lot of features that can be mined because of its unique gameplay mechanics. This article analyzes the possible relationship between multiple attributes of words and difficulty based on the dataset. And a forecast is also made for the number of reported results in two months. In addition, we also mine features based on the dataset to predict the associated percentages of (1,2,3,4,5,6, X) for a given word, and also divide the words by difficulty.

For problem one, we first clean the data, correct or eliminate abnormal numbers and words, and then use the ARIMA model and **LSTM** model to fit the number of reported results in the dataset to predict future trends. For the ARIMA model, we use the **ARIMA (3,1,5)** model to do the prediction. The optimal parameter p,q is determined based on the AIC criterion. And the model passes the white noise test. The predicted value is **17079**. For the LSTM model, a prediction sequence of length 75 is established, and the parameters are adjusted so that the previous part of the trained prediction sequence coincides with the data as much as possible. This not only ensures that the model effectively remembers the information of the previous data but also ensures the accuracy of the prediction data, and the final prediction result is **18913.** After several calculations, we calculate the prediction interval as **[17000, 19000].**

In addition, we list as many word attributes that may be relevant to the result as possible, express the word difficulty as a percentage-weighted average, and then conduct correlation analysis, and the analysis showed that the repetition rate of letters in the word is significantly correlated with the word difficulty. For other attributes characterized by frequency, such as initials frequency, we convert them into information and then analyze the correlation with word difficulty, and we find that the **Pearson correlation coefficient** between a single attribute and difficulty is about 0.2 on average (significant).

For problem two, after analyzing the correlation between each attribute and the difficulty coefficient, we construct a **decision tree** as a model for predicting the associated percentages of (1,2,3,4,5,6,X) for each word. Then, the associated percentages of (1,2,3,4,5,6,X) are used as a label to train, then a decision tree model that can predict the associated percentages of (1,2,3,4,5,6,X) by word features is obtained. Taking 'EERIE' as a sample for multiple predictions (Buffon's needle experiment), taking prediction sum in legal range as a scatter, it is found that the deviation between the range of real prediction and the expected prediction reaches 12% at most, so we also have at least 88% confidence that the model is accurate for rare words.

For problem three, we find that the associated percentages of (1,2,3,4,5,6,X) distribution for each word approximate the Gaussian distribution, and the **Pearson correlation coefficient** between expectation and the difficulty coefficient is as high as 0.965 (significant). So we take the expectation of each word's Gaussian fitting function as the classification basis, and use the **K-means algorithm** to divide the words into 5 categories according to the expectation range from small to large, namely: **easy**, **normal**, **medium**, **hard,** and **ultimate**. Then we use the model of the second question, to predict EERIE 1000 times and eliminate the abnormal prediction value, taking the average of the legal value. Then we perform Gaussian fitting, get the expectation, evaluate its difficulty according to the interval where the expectation is located, and finally divide EERIE into the **hard** level. After analysis, we believe that the accuracy of our model is 85%. After the sensitivity test, it is found that the classification effect hardly can be affected by the fluctuation of input values.

For problem 4, in this paper, we tried to mine many features in the original dataset, such as the percentage of the number of difficult patterns in the number of reports HR, the expectation of the score distribution, i.e., the difficulty coefficient D, the percentage of less than j attempts to pass the game U_j, the position-coding of words P, the repetition Mul, the number of vowels VN, the vowel position coding VP, the total letter frequency coefficient FS, the initial letter frequency coefficient FF, the double letter group frequency The features with high correlation with difficulty coefficients were finally retained to train the machine learning model.

Catalog

1 Introduction

1.1 Background

The New York Times offers one-word puzzle per day, and participants are considered successful if they guess the word six times or less. Each guess must be an actual English word, otherwise, it does not count as a valid guess. Each guess will provide feedback, and the color of the letter will change. Gray means the letter is not in the word at all. Yellow means the letter is in the word but in the wrong position. Green means the correct position of the letter in the word is guessed. With tens of thousands of people sharing their guesses on Twitter every day, wordle has created a word-guessing craze.

1.2 Restatement of the Problem

• Use a model to explain the trend in the number of reported results and to give a prediction interval for the number of reported results shared on Twitter on the date of March 1, 2023. Speculate whether other features affect the number of times people guess the correct word in the hard mode.

• Build a model that, when given a word in the future, predicts the associated percentage of (1, 2, 3, 4, 5, 6, X) for this word. Describe the prediction for the word 'EERIE' for the date March 1, 2023. Describe the uncertainty of the model.

• Cluster the words by difficulty level. Make predictions about the difficulty of the given word **EERIE** and talk about the accuracy of the model.

• Discuss some other interesting features of the dataset.

1.3 Our work

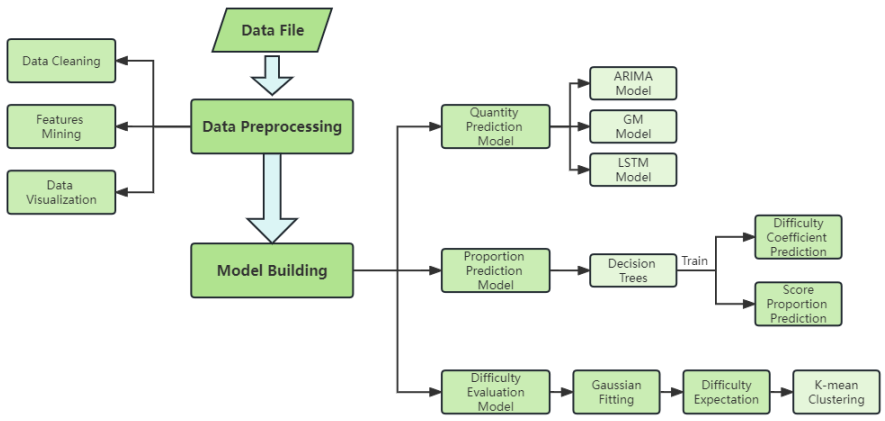

Figure 1. Model structure and construction process

2 Assumptions and Explanations

2.1 Assumptions

1. The difficulty of the puzzle is related only to the character of the word itself and can be measured using the associated percentages of (1,2,3,4,5,6,X)as a criterion, regardless of other influences.
2. No need to consider factors such as cheating or using auxiliary programs that make 1 try over-represented.
3. The rounding of the percentages of (1,2,3,4,5,6,X) in original dataset has a negligible impact on the results.
4. The number of reported results will not surge due to some factor.

2.2 Notation

Table 1 shows the notations that we use.

Table 1. Notation

Symbol	Description
O_i	the sum of the proportion of (1,2,3,4,5,6,X) in record i
HR_i	the percentage of **Number in hard mode** in **Number of reported results** in record i
D_i	the difficulty coefficient of the puzzle in record i
U_{ij}	the percentage of games with less than j tries in record i
Mul_w	repetition of word w
VN_w	number of vowel letters in word w
VP_w	number of all vowel positions in word w
P_{wj}	position coding of the j^{th} letter in word w
FS_w	frequency factor of the word w determined by the frequency of occurrence of the letters in each position
FF_w	frequency factor of the word w determined by the frequency of occurrence of the initial letter
FE_w	frequency facto of word w determined by the frequency of the word itself
FB_w	frequency factorof the word w is determined by the frequency of all diacritical marks in the word

3 Preparation

3.1 Data cleaning

3.1.1 Fix the Words

According to the restrictions of the game, the Word field in the dataset must be a word consisting of 5 lowercase letters, and we found that some records in the dataset did not meet this condition. So we found 5 words with errors by traversing the dataset and corrected the data according to the historical puzzle [1] given on the official website of the Wordle game.

- Delete one extra space after favor (Contest number=207)
- change tash to stash (Contest number=314)
- clen to clean (Contest number=525)
- na?ve to naive (Contest number=540)
- rprobe to probe (Contest number=545)

The above modification is feasible because it refers to the real data and ensures the data quality while avoiding the hazards of deleting records, including the impact of small samples on the robustness of the model and data discontinuity.

3.1.2 Add new attributes

The first row of the original dataset **"Problem_C_Data_Wordle.xlsx"** is deleted and the wrong words are corrected and saved as **"data0.csv"**, which represents the initial dataset. To better explore the information in the dataset, we first add the following fields to the dataset.

- **under_j** is the field that indicates the percentage of attempts with less than j attempts, abbreviated as **U**j. Specifically, we call U_7 to be **Overall**. the formula is:

$$U_j = \sum_{k=2}^{j} t_k \, , j \in [2,7] \tag{1}$$

- **Hard_rate** is the ratio of the number of difficult modes selected to the reported results, abbreviated as H, and the formula is:

$$H = \frac{Number \quad in \quad hard \quad mode}{Number \quad of \quad reported \quad results} \tag{2}$$

3.1.3 Handle abnormal data

By looking at the dataset, we were able to visually and clearly find two abnormal data, namely the record with Contest number = 281 (its Overall is

126%), and the record with Contest number = 529 (its Hard Rate is as high as 93%). We choose to remove these abnormal records. Then we drew box plots of **Number of reported results** and **Number in hard mode** as **Figure 2** and **Figure 3** to detect and correct the abnormal data points by inheriting the value of the previous point of the sequence. The modified relationship diagram is shown in **Figure 4**, **Figure 5**.

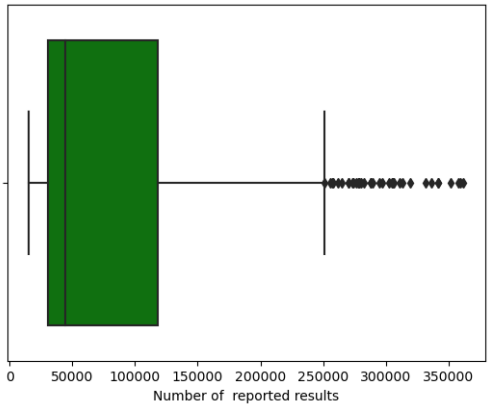

Figure 2. Boxplot of Number of reported results

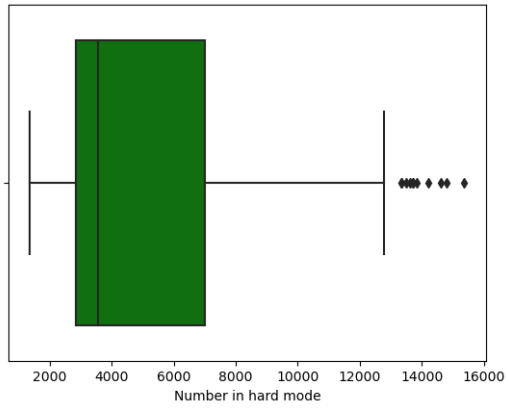

Figure 3. Boxplot of Number of reported results

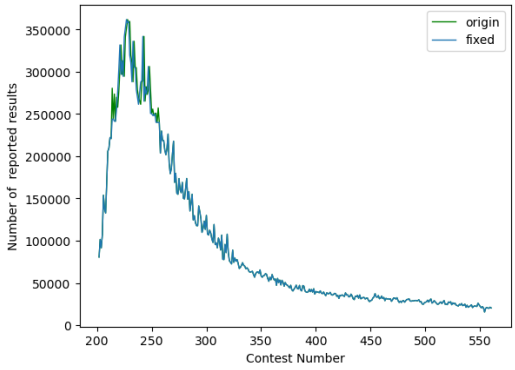

Figure 4. The relationship chart between Contest Number and Number of reported results

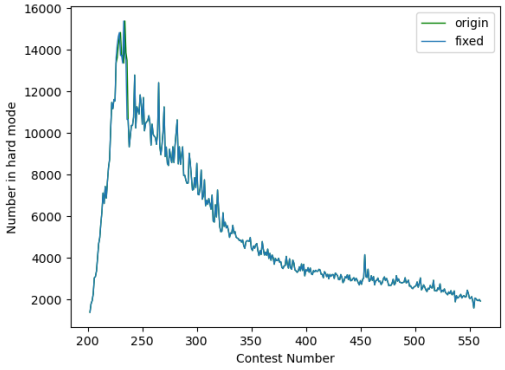

Figure 5. The relationship chart between Contest Number and Number in hard mode

The following set of **Figure 6** shows the relationship between **Contest Number** and **Hard rate**. The left graph represents the original dataset, and the middle graph represents the result after one cleaning, it looks like there are still two points where the **Hard Rate** looks weird, but the dataset as a whole is useful. We then limit the range of **Hard Rate** and get the right graph after the second cleaning. We save the new dataset as "data1.csv". At this point, the data cleaning is almost complete.

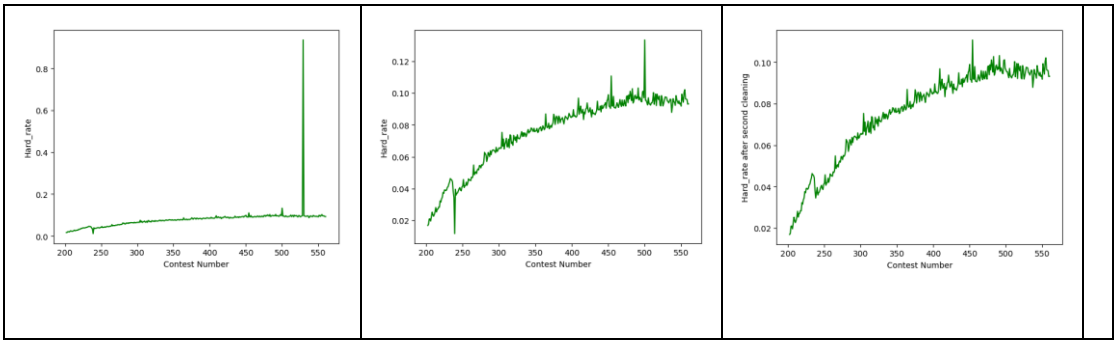

Figure 6. The relationship between hard rate and content number before and after cleaning abnormal data

3.2 Possible attributes of words

Although we can infer the approximate difficulty of a puzzle by counting the percentage of different attempts and getting the difficulty factor D, in the end, the difficulty of a puzzle is determined by the nature of the word itself. For example, words that are less used by people in daily life have a lower probability of being guessed. In addition, words with repeated letters are more difficult to guess, which is determined by the game mechanism. Suppose the word consists of two e's, and at first you only fill in the e. Even if it is filled in the correct position, it is difficult to deduce that the other e is needed to be filled in, not to mention that filling in the wrong position will only return a yellow square.

Therefore, we create a new dataset "**word_data.csv**" for each feature of the word and the corresponding difficulty factor and analyze it to determine the contribution of these features to the difficulty.

3.2.1 **Letter frequency**

The frequency of occurrence of letters in each position affects the difficulty of the whole word. For example, according to Wikipedia [6] it is stated that the most frequent occurrence of letters in English is 'e', followed by 't', 'a', 'o ' ..... However, the highest frequency of the initial letter is 'a'. Based on this information we can infer that words that start with 'a' and have 'e' or 't' are relatively easy to guess. However, in Wordle this rule is not so applicable, for example, 'tion' as a suffix in many words will greatly affect the frequency of the letter 't', and in wordle, which is a five-letter word, it is almost impossible for 'tion' to appear. It is almost impossible for 'tion' to appear. Therefore, we refer to all datasets available as answers and guessable words in Wordle on the Kaggle website [5] and data mining codes and obtained a more reliable frequency factor FS by calculating the following formula.

$$FS_{word} = \sum_{s \in word} \frac{count(s, guess)}{sum(guess)} + \frac{count(s, solution)}{sum(solution)} \quad (3)$$

where count(a,b) denotes the frequency of occurrence of the letter a in

dictionary b, and sum(b) denotes the frequency of occurrence of all letters in dictionary b.

Among the letters in each position, the initial letter frequency is the feature that we need to pay extra attention to. We obtain the frequency coefficient FF of initial letters in a similar way, calculated as follows.

$$FF_{word} = \frac{count(word_0, guess)}{sum(guess)} + \frac{count(word_0, solution)}{sum(solution)} \quad (4)$$

The $word_0$ represents the initial letter of the word.

3.2.2 **Bigram frequency**

It is obviously not reasonable to define the frequency coefficient only in terms of each letter, let's say that 'st' occurs more frequently than a syllable like 'ee', which consists of two high-frequency letters, so we also have to consider the frequency of syllables that occur in the word. To simplify the problem, we consider only diphthongs and define all combinations of adjacent letters in a word as diphthongs (although some cannot be called syllables) and calculate the sum of the frequencies of all combinations in the word, FB, with the following formula.

$$FB_{word} = \sum_{bigram \in word} \frac{count(bigram, dictionary)}{sum(dictionary)} \quad (5)$$

where count(a,b) indicates the frequency of occurrence of adjacent letter combinations a in dictionary b, and sum(b) indicates the frequency of occurrence of all adjacent letter combinations in dictionary b.

3.2.3 **Multiplicity**

As mentioned earlier, if there are repeated letters in the word, the difficulty of the puzzle will be greatly increased. We define the repetition Mul as the sum of the product of the number of species of each repeated letter in the word and the number of occurrences of the letter, calculated as follows.

$$Mul_{word} = \sum_{s \in word} count(s, word)^2 \quad (6)$$

3.2.4 **Vowel**

Vowels are an indispensable part of words, so the position and number of vowels are factors worth considering. We define VN as the number of vowels in a word and VP as the binary code of all vowel positions in a word, for example, the position of vowels in **'EEIRE'** is coded as [1,1,1,0,1].

3.2.5 **Degree of common use**

We have previously uncovered many potential factors that influence word difficulty based on the nature of the words themselves. From the perspective of daily life, the commonness of words can be a very intuitive representation of word difficulty, as people tend to guess the commonly used words. We define the word commonness factor **FE**, which is obtained by counting the frequency of word occurrences in large texts, and these data can be found on top of **Kaggle[4].**

3.3 Feature analysis

3.3.1 **Repetition of word**

We found that if there are repeated letters in the target word, the number of successful solutions or time cost to solve it will increase. Since all the target words consist of 5 letters, we divide the words according to the number of letter repetitions into cases where each letter appears once, one letter appears twice, one letter appears three times, two letters appear twice each, and so on.

Based on common sense, we exclude cases that do not exist, including, situations where a letter appears 4 or 5 times and where the word is a combination of two letters. We divide words into the following four categories according to the number of letter repetitions.

W1: Each letter appears once

W2: One letter appears twice

W3: One letter appears three times

W4: Two letters both appear twice

The words in the data set are divided according to the above four categories, and the average percentage of their completion times is found for each category, and the four categories are plotted to line graphs, as shown in the figure.

Under the assumption that the participants' word reserves do not differ greatly, we regard the percentage of the four types of word categories as the average percentage of the amount of the four types of words in the participants' human brain reserves. The larger the percentage, the higher the probability of guessing the target word. Then we can derive the difficulty ranking of the four categories

DW1<DW2<DW4<DW3.

This hypothesis can also be approximated from the line graph. We assume that the higher the percentage of (1,2,3,4), and the lower the percentage of (5,6,X), the easier the words are. Then we can also derive the above difficulty ranking by sorting from left to right.

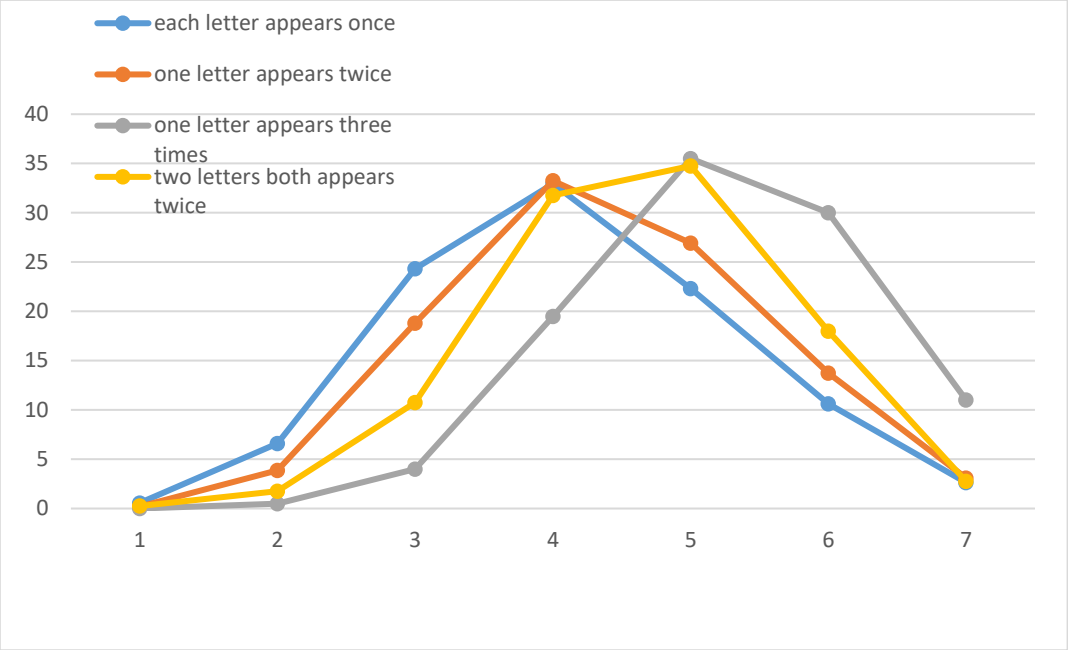

Figure 7. Associated percentages of (1,2,3,4,5,6,X) under word repetition rate

3.3.2 **Attributes characterized by frequency**

We then used the information content formula to calculate the information content of the words under each attribute, such as FE, FS, FF, FB, after which we calculated the Pearson correlation coefficient of each attribute with difficulty separately. The results obtained were statistically significant and showed a low correlation overall. The formula for calculating the information content is as follows.

$$I_{i,j} = -log_2 P_{i,j} \quad i = 1,2,3,4; j = 1,2,3,\cdots,257;\ P_{i,j}\ is\ the\ frequency \tag{7}$$

We analyze that due to the large number of optional attributes of the words and as much as they cannot characterize the word well independently, it can be demonstrated that each of the selected attributes has a low correlation with the outcome distribution shown as **Table 2**. Therefore, we consider that the attributes obtained by weighting the four attributes will have a strong correlation with the outcome distribution.

Table 2. Correlation coefficient

attribute	Pearson correlation coefficient	Significance testing
FS	0.268	Significant
FB	0.173	Significant
FF	0.173	Significant
FE	0.249	Significant

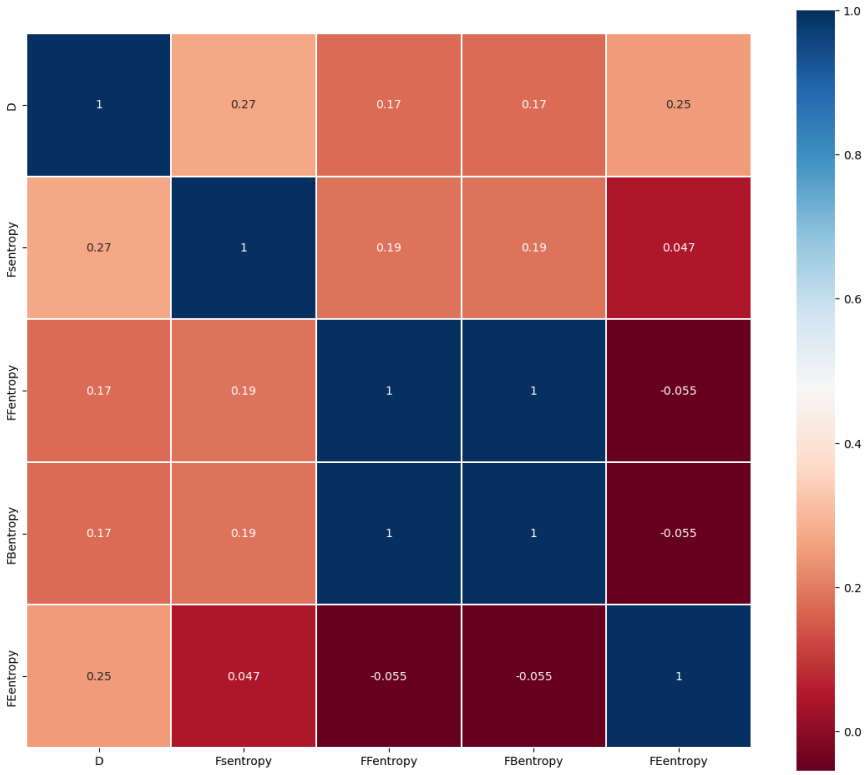

Figure 8.Correlation heatmap

4 Quantity Prediction Model

For the solution of problem one, to obtain the prediction interval, we use the ARIMA model and the LSTM model to predict the number of reported results. The results predicted by the two models are used as two endpoints to form the prediction interval

4.1 Prediction based on the ARIMA(3,1,5) model

To analyze the future trend of the number of reported results, we use the ARIMA model to do the prediction on 2023/3/1 according to the requirements of the question. The ARIMA model is a time series forecasting model that is based on both AR model and MA model. The time series is first differentiated, eliminating its characteristics such as trending seasonality, so that the differentiated series is a stationary time series. At this point, the transformed sequence can be considered as an ARMA sequence for further study.

4.1.1 **ARIMA model introduction**

For a zero-mean smooth sequence $\{X_t\}$,t=0,1,2,…, if it can be expressed as a weighted sum of the first p terms and the sum of zero-mean smooth white

noise, as follows:

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \varphi_3 X_{t-3} + \cdots + \varphi_p X_{t-p} + \theta_t \quad (8)$$

and by introducing the lag operator:

$$B^m X_t = X_{t-m} \quad (9)$$

and the arithmetic polynomial:

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_p B^p \quad (10)$$

where θ_t is a smooth white noise with zero mean variance of σ_θ^2. Then X_t is said to be an autoregressive series of order p, denoted as an AR(p) series. The model can be rewritten as:

$$\varphi(B) X_t = \theta_t \quad (11)$$

If X_t satisfies

$$X_t = \theta_t - \alpha_1 \theta_{t-1} - \alpha_2 \theta_{t-2} - \cdots - \alpha_q \theta_{t-q} \quad (12)$$

Then call X_t a sliding average series of order q, denoted as MA(q) series. As above, the model with the introduction of the lag operator can be rewritten as

$$X_t = \alpha(B) \theta_t \quad (13)$$

If X_t satisfies

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \theta_t - \alpha_1 \theta_{t-1} - \alpha_2 \theta_{t-2} - \cdots - \alpha_q \theta_{t-q} \quad (14)$$

Then call X_t an autoregressive sliding average series of order p, q, denoted as ARMA (p, q) series, and the model can be rewritten after introducing the lag operator as

$$\varphi(B) X_t = \alpha(B) \theta_t \quad (15)$$

The smoothness condition of ARMA model is for the equation $\varphi(B)\, X_t = 0$,all its roots fall outside the unit circle, and the reversibility condition is for $\alpha(B) = 0$,all its roots are outside the unit circle. The two properties are very important in theoretical and practical problems

Therefore, the time series we want to predict is different to eliminate its trend seasonality and other characteristics, and then it can be considered as a smooth time series, after which the ARMA model is used to fit the prediction. the specific steps of the ARIMA prediction model are.

Step1: Calculate the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the time series we want to predict, and determine whether it obeys the ARMA model by judging whether it is truncated or tailed. if at least one condition is not met, it means that the original sequence is a non-stationary sequence, then it is differentiated in the first order, and the ACF and PACF are judged until the differential sequence is a stationary sequence.

Step2: Determine the parameters in ARIMA (p, 1, q): use the AIC criteria to fix the order and select the optimal parameters p, q

Step3: Validation of the model and prediction: the white noise test is performed to judge the reasonableness of the model, after which the model is used for prediction.

4.1.2 **Model building and prediction**

First, we calculate the ACF and PACF for **the number of reported results** and perform the smoothing test, which does not pass. After performing the first-order difference, the smoothing test is performed again and the logical output is obtained as shown in the following **Table 3**.

Table 3. ACF and PACF

Adf	**1**
Kpss	0

The ACF and PACF of the time series after first-order differencing are shown in **Figure 9** and **Figure 10** below.

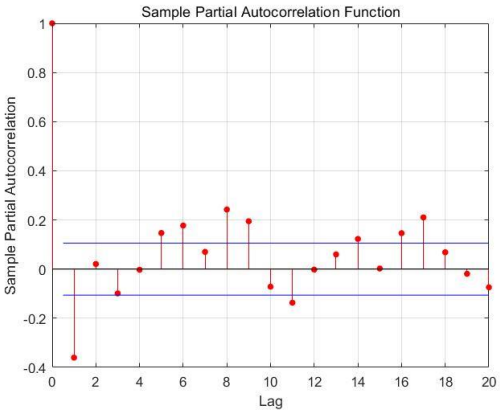

Figure 9.Sample Partial Autocorrelation Function

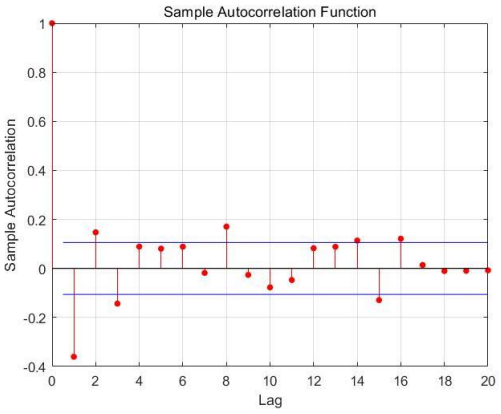

Figure 10.Sample Autocorrelation Function

Afterward, we perform parameter sizing to determine the model as ARIMA (3, 1, 5) time series forecasting model and perform a white noise test on the model to obtain its standardized residuals line plot, histogram, ACF plot, PACF plot, and QQ plot. The standardized residual plot shows that the residuals are randomly distributed around 0. Analysis of the QQ plot shows that the majority of the points fall on the red line. The Durbin-Watson test (D-W test for short) of the obtained errors yields the DW statistic DW0 = 2.0066, which is extremely close to 2 and yields no autocorrelation of the residuals.

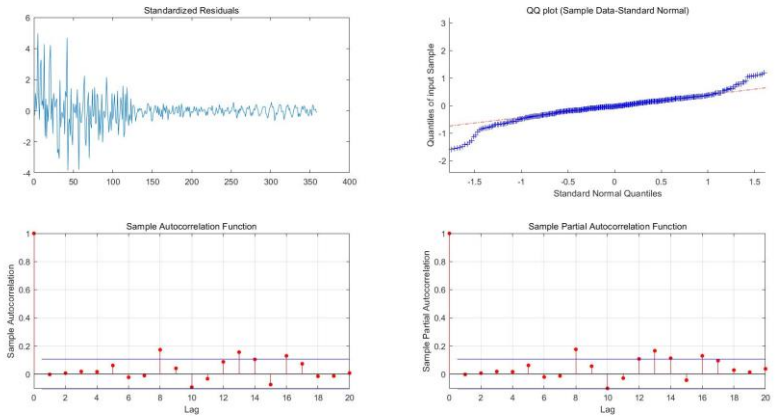

Figure 11. Residua, QQ, ACF, and PACF plot

The final model obtained:

$$Y_t = -84.842 + 0.18896Y_{t-1} + 0.26874Y_{t-2} - 0.44161Y_{t-3} + \theta_t - 0.68302\theta_{t-1} - -0.06505\theta_{t-2} + 0.44715\theta_{t-3} + 0.015029\theta_{t-4} + 0.21154\theta_{t-5} \quad (16)$$

Finally, we use the model to predict forward 60 days from January 31, 2022, so we obtain the data X_2.=17079 for March 1, 2023, which means that **the number of reported results** on that day is approximately 17079.

4.2 LSTM Model

LSTM(Long-Short Term Memory) is a special variant of recurrent neural networks with a “gate” structure. The LSTM has three gates, namely the forgetting gate, the input gate, and the output gate. which determines whether the information is remembered or forgotten at each moment, the input gate determines how much new information is added to the cell, the forgetting gate controls whether the information is forgotten at each moment, and the output gate determines whether the information is output at each moment.

These gates control how data flows into and out of the cell and how much of the previous content is retained in memory. By controlling the flow of data, the LSTM units can learn and memorize sequences over long time intervals. The output of each cell is fed to the next cell in the sequence, allowing the model to learn the dependencies between successive data points. Thus LSTM can be used for time series prediction, thus solving the problem of predicting the number of reported results in the future. Our group tried to use LSTM to predict the time series of reported results data, and the results are shown in **Figure 12**.

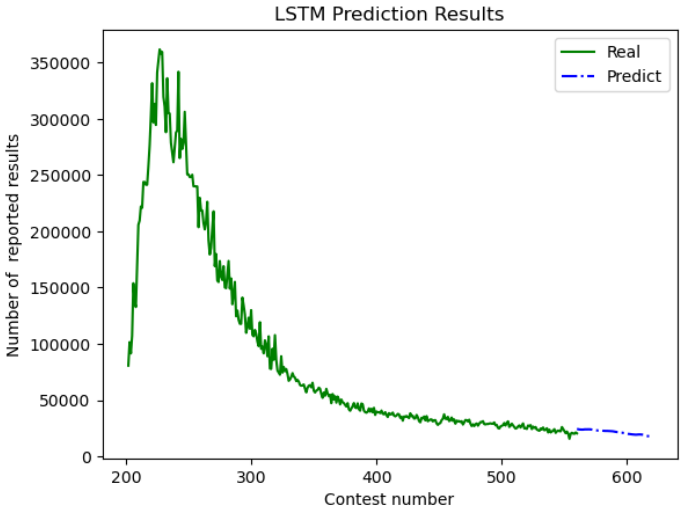

Figure 12. LSTM Prediction Results

From the graph above, it is easy to see that the trend of **Number of reported results** is fitted very approximately using the LSTM model and performs reasonably well on the prediction task.

5 Proportion Prediction Model

5.1 Preparatory work

For the attributes mined above (mentioned in Chapter 3), we use a heat map to observe the correlation between features, especially with difficulty coefficients and then convert some difficulty frequency coefficients into information entropy.

To get an accurate model for predicting word difficulty, we put all the features mined in the alphabet into the dataset and save them as "word_data.csv", the preview is shown in **Figure 15**. After normalizing all the features, we use the difficulty coefficient **D** as the label **y** and the others as the input features **X**. We use these features to train the machine learning model to derive a predicted difficulty coefficient **y_pred**. before training, we removed the Word field and replaced it with five separate positional encodings $\mathbf{pos_i}$. this allowed us to handle data that were not legitimate in training while preserving as much word information as possible.

	Contest number	Word	D	pos_0	pos_1	pos_2	pos_3	pos_4	Mul	VN	VP	FS	FF	FB	FE
0	202	slump	4.13	s	l	u	m	p	1	1	4	0.482079	0.158099	0.039525	0.000013
1	203	crank	4.22	c	r	a	n	k	1	1	4	0.524498	0.085529	0.021382	0.000032
2	204	gorge	4.64	g	o	r	g	e	4	2	9	0.585285	0.049676	0.012419	0.000028
3	205	query	4.43	q	u	e	r	y	1	2	12	0.497670	0.009935	0.002484	0.000452
4	206	drink	3.77	d	r	i	n	k	1	1	4	0.464095	0.047948	0.011987	0.000423
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
352	556	condo	4.45	c	o	n	d	o	4	2	9	0.505771	0.085529	0.021382	0.000093
353	557	impel	4.15	i	m	p	e	l	1	2	18	0.557381	0.014687	0.003672	0.000001
354	558	havoc	4.40	h	a	v	o	c	1	2	10	0.465923	0.029806	0.007451	0.000024
355	559	molar	4.14	m	o	l	a	r	1	2	10	0.622068	0.046220	0.011555	0.000009
356	560	manly	4.34	m	a	n	l	y	1	1	8	0.510910	0.046220	0.011555	0.000018

Figure 15. A preview image of the dataset ‘word_data.csv’

5.2 Training result and model comparison

After dividing the training and test sets, we selected several common models to train and compare the models using the accuracy of validation as the criterion, including Decision trees, Bayesian classifiers, Logistic regression, K-neighborhood models(KNN), Support Vector Machines(SVM), Perceptron Machines, Stochastic Gradient Descent(SGD), and Support Vector Machines(SVM). All accuracy results and the comparison results are shown in **Figure 16** and **Table 2**.

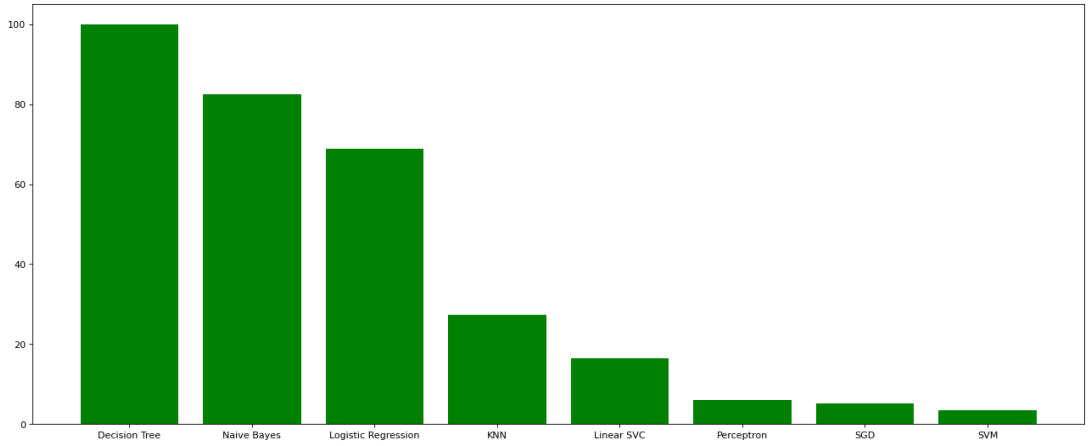

Figure 16. Bar plot of accuracy results for each model

From **Table4** we see that models like Perceptron, SGD, and SVM are not suitable for this type of problem (of course I do not deny that there are reasons why our models are not well-tuned, but all models are called directly from the sklearn library); while decision trees have a very impressive performance on this dataset, bearing in mind that despite the small number of samples Using 80% of the samples to train and thus passing 100% of the remaining 20% is also quite an impressive performance. Therefore, we decided to use decision trees to solve the next problem.

5.3 Use decision tree to predict score proportion

Previously, we verified the feasibility of the decision tree interpretable model and trained a decision tree model that can be used to predict the difficulty coefficient D. However, for the problem of predicting the proportion of scores (1,2,3,4,5,6,X), D alone cannot be used for backward inference because D is derived from the expectation of the scores. According to the same principle, we can train the model with the ratio of each score instead of the difficulty coefficient as the label. After 7 repetitions, we can predict the proportion of each score for each word. We can verify the validity of the prediction by summing the proportions of (1,2,3,4,5,6,X) in a word to get O. We define that the data of O is valid in the interval [97.5,102.5], and the average of multiple predictions gives a stable proportion of scores.

Table 4. model accuracy

	Model	Score
7	Decision Tree	100.00
3	Naive Bayes	82.46
2	Logistic Regression	68.77
1	KNN	27.37
6	Linear SVC	16.49
4	Perceptron	5.96
5	SGD	5.26
0	SVM	3.51

5.4 How difficult ‘EERIE’ is

Our steps in **5.1** obtain other features of the word **'EERIE'** and put them into the model for prediction. The normalized inversion gives a difficulty coefficient of 5.02, while the percentage of scores is (1%,3%,4%,28%,28%,21%,15%). This difficulty prediction placed in 'data1.csv' means that it is more difficult than 348 words in 355 valid data. Words that are more difficult than them are all rare.

We believe that while this result seems reasonable, the small sample size makes it impossible to train a model that takes these types of words into account, and there are few words with as much "personality" as 'eerie' in the original dataset. This greatly hinders the judgment of the model in machine learning. Our group attempted to repeat the prediction 1000 times for this sample, equivalent to Buffon’s needle experiment, in which 168 guaranteed available (**O** in [97.5,102.5] is reasonable) data could form a range distribution on a two-dimensional plane about the proportion of predicted scores. The farthest of these points differed from the training difficulty factor at that time by only 0.78, and this difference divided by the length of the representation range [0,7] of the difficulty factor yielded an accuracy of at least 88% for our model. In summary, for words that are commonly used or have a small distance from the sample feature space, we believe that the model has more than 90% accuracy to distinguish the difficulty, while for words like 'EERIE' that are not sufficiently trained for features, we are only about 88% confident in the model.

6 Difficulty Evaluation Model

For the division problem based on word difficulty, we perform cluster analysis based on the K-means algorithm on the associated percentages of (1, 2, 3, 4, 5, 6, X). We divide the difficulty of words into 5 levels and obtain the associated percentages of (1, 2, 3, 4, 5, 6, X) of the word ‘EERIE’ based on the model established in the second question. Finally, we come up with the difficulty level of the word EERIE by bringing the data into the established classification model.

6.1 Identify clustering metrics

In the second question, we trained the data through machine learning and predicted the associated percentages of EERIE's results. To divide words by difficulty, we first need to identify the clustering metrics first

By analyzing the data provided by the question, we come up with the most intuitive data that shows the difficulty of the word as the associated percentage of (1, 2, 3, 4, 5, 6, X). However, it is much more complicated to directly use

percentages as a basic indicator, because the data is 7 dimensions, and the characteristics of these 7 dimensions data are different. For the percentages of (1, 2, 3), we can roughly think that the lower the proportion, the higher the difficulty is of the word, but to be more specific, we also need to determine the different contributions of (1, 2, 3) to the difficulty; For the percentages of (4, 5, 6, X), we can roughly think that the higher the proportion, the higher the difficulty is of the word, and we also need to calculate their respective contributions. Such an analysis, if directly used as a division basis indicator, requires many complicate parameters to determine and optimize. By reviewing and analyzing the data, we propose a method to identify the clustering metrics: the associated percentages of (1, 2, 3, 4, 5, 6, X) of each word are fitted to a Gaussian function, and the expectation of the Gaussian function is taken as the clustering metric.

By observing the associated percentages of (1, 2, 3, 4, 5, 6, X), we conclude that the associated percentages of (1, 2, 3, 4, 5, 6, X) of almost every word can be approximated as a Gaussian distribution, and the expectation of the fitted Gaussian function has a certain relationship with the difficulty of the word. To preliminarily explore the existence of this relationship, we first select three words with large differences in the associated percentages of (1, 2, 3, 4, 5, 6, X), and then do the Gaussian fitting and plot the three fitting functions in the same figure. By the nature of the Gaussian function: the expectation of the Gaussian function is the axis of symmetry of its function image, and the variance represents the trend of change in its image. Our analysis shows that the variance difference among the three fitting functions is small, that is, the difference of the changing trend among the curves is small; The main difference is that the difference among the fitting function's expectation is large, and the word with higher associated percentages of (1,2,3) corresponds to the smaller expectation of the fitting function.

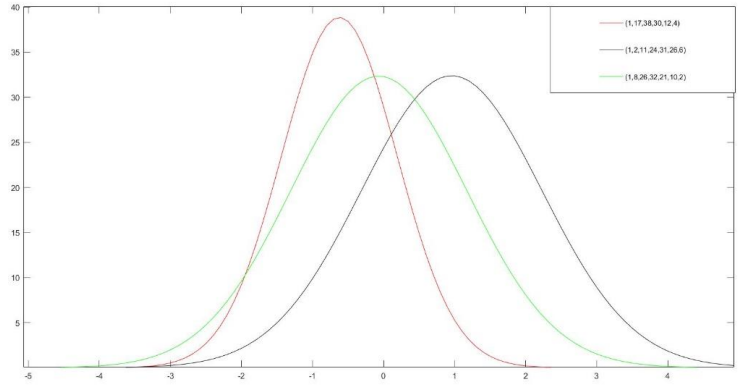

Figure 17. Gaussian fit for different associated percentages

Based on the above, we perform Gaussian fitting on all words to obtain the i-th word's expectation μ_i and variance σ_i of its Gaussian function

The fitting function we use is：

$$y = a \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right) \tag{17}$$

Then we get the expectation and variance of the Gaussian fitting function for the associated percentages of each word. Some of the data is shown in the following **Table 5**.

Table 5. The outcome of the Gaussian fitting

Word	Expectation	Variance
slump	0.06892	1.452
crank	0.1769	1.829
gorge	0.7404	1.805
query	0.5015	1.738
drink	-0.3959	1.451
favor	0.6747	1.898
abbey	0.67	1.7
…	…	…

After that, we test the correlation coefficient of expectation and difficulty coefficient, and the correlation coefficient is as high as 0.965 and passes the significance test. So we conclude that there is a strong positive correlation between expectation and difficulty factor. So we determine to use expectations as clustering matric.

6.2 Clustering model based on K-means algorithm

6.2.1 Model Introduction

We used the K-means algorithm to cluster 257 expectations, with different categories corresponding to different difficulty levels. The K-means clustering algorithm is an unsupervised classification algorithm. It takes the mean of the divided cluster as the center point of the cluster, and can automatically calculate and update the center point of each cluster by continuously iteratively dividing the data set under the premise of uncertain division

The main steps of the algorithm are:

- ✓ Step 1: Select the number of cluster categories k and select k initial center points
- ✓ Step 2： For each sample point, find the closest center point to it and draw it to the cluster represented by the center point
- ✓ Step 3: Calculate the mean of the cluster to obtain a new center point
- ✓ Step 4: Recalculate the distance between the object and the center point and redivide it. If the clustering result changes, return step three; If there is no change, clustering results are returned.

6.2.2 **Model building and solving**

We use the K-means algorithm to cluster the 257 expectations, choose 5 the number of cluster categories, select 5 initial center points, and arrange the 5 clusters according to expectations. Finally, we successfully divide them into 5 clusters, which we called five difficulty levels, namely: easy, normal, medium, hard, and ultimate.

Table 6. Five difficulty clusters

Level	Expectation range	Number
Easy	(-1.016, -0.2587)	54
Normal	(-0.2782, 0.1166)	127
Medium	(0.1201, 0.5524)	113
Hard	(0.5599, 1.262)	57
Ultimate	(1.513, 2.942)	6

After that, we use the model built into the second question to predict the associated percentages of the word EERIE. Since the result distribution obtained each time is different, and the sum of the result distributions of some data is quite different from 100, we use our decision tree model to predict the result distribution of EERIE 1000 times and save the data with the total percentage lying in (98.5, 102.5) to finally obtain 166 valid data, and then we average the 166 results to obtain a more reliable associated percentage for EERIE

$$Result = (0.5697,8,4,32,32.697,21,1)$$

Gaussian fitting is then performed on the data, and the expectation and variance of the fitted function are as follows (95% confidence interval in parentheses)：

$$\mu_{eerie} = 0.7511\ (0.3176,1.185)$$
$$\sigma_{eerie} = -1.486\ (-2.1,-0.8708)$$

The visualization result is shown in **Figure 18**, and the dots of different colors in the figure represent different clusters, and the difficulty from top to bottom is ultimate, hard, medium, normal, and easy. The points indicated by the red boxes represent the position of EERIE. As we can see, the difficulty of the EERIE is hard.

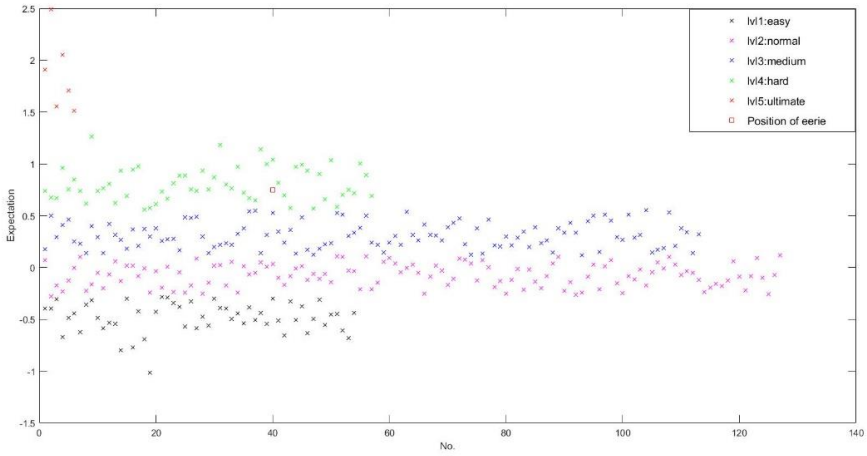

Figure18. Visualization result

6.2.3 **Model accuracy discussion**

The accuracy of the model is mainly related to the error of the data set and the word eerie when Gaussian fitting is performed, and the accuracy of the prediction model in problem two. As mentioned earlier, we have tested the correlation with the difficulty coefficient and obtained a significant correlation of R=0.965. So we take R=0.965 as one of the evaluation coefficients of our model. It has been verified that due to the large interval of the classification, the error generated by the expectation obtained by Gaussian fitting is negligible in the calculation of the effect on the accuracy of the classification. And since we use the prediction results of the second-question model as our input, we need to take the accuracy of the second-question model into account when considering the accuracy of the model. Finally, we get the accuracy of the classification model: A2=R*A1; Resulting in A2=84.92%. Therefore, we believe that the accuracy of our model is about 85%

6.3 Sensitivity tests for the difficulty evaluation model

Suppose that some influencing factors cause the percentage of (2) to fluctuate by about 100%, and the percentage of (3, 4, 5, 6) decreases on average. In this case, we get the expectation of 0.7724, and the classification is still hard. Similarly, if the share of (3) rises by 100%, the percentage of (4, 5, 6) decreases on average, in which case we get an expectation of 0.6781 and the classification is still hard. If the proportion of (4) fluctuates by 20%, the percentage of (2,3) decreases on average, and the expectation is 0.683, and the classification is still hard. We conclude that perturbations in any reasonable case on any one scale have an effect on the expected value, but have almost no effect on the classification results. Therefore, we conclude that our model is stable for classification.

7 Discussion of other features

In addition to the various attributes we found in the data mentioned earlier, we also found some interesting features.

Of the five difficulty levels of the words, there are only 6 words at the ultimate level, and there are exactly two words watch and catch, which all end in tch. This is funny. From this, we can make such an inference that there are relatively few words in English that end in tch; Most of the words at easy level are words with strong life attributes, such as aloud, there, etc.

From 300000 to 20000 reports, the proportion of hard mode always fluctuated around 10%. We believe that the reason why the proportion of hard mode always remains at the floating level of around 10 is the wordle game mode. Since each person can only play once a day, in order to be able to guess the answer, 90% of the crowd always choose not to take risks. If New York Times dose not limits the try number of games, we think the share of hard will increase very quickly.

Another interesting thing is that 6% percent of people hit the word ‘train’ on their first try.

8 Model Evaluation

8.1 Strengths

(1) Multiple models are used to make predictions in the Quantity Prediction Model, which ensures the accuracy of the prediction interval while comparing the fitting and prediction effects.

(2) The concept of difficulty coefficient is introduced, which is directly expressed as the expectation of the grade distribution in the original data set. And in fact, the difficulty of the puzzle is determined by the properties of the words themselves, so it serves as a label for predicting the difficulty in the training task, enabling us to perform supervised training on the features of the words.

(3) The attributes of the words themselves are fully considered, not only mining the attributes that can be inferred from the original dataset, such as the number and position of vowels and the number of repetitions of letters but also obtaining more features by consulting data on the Internet on the frequency of words and letters used, making the training process more accurate.

(4) The decision tree model is easy to understand and interpret and is suitable for small data sets given by similar topics.

8.2 Weaknesses

(1) The prediction of the number of Reported Results starts only from the previous number itself and does not consider the influence of other external factors.

(2) Difficulty Prediction Model needs a larger dataset to support it. Because the small sample size means that many potential attributes of words are not trained, the model will be prone to make misjudgments once it encounters a rare word.

(3) The number of features that can be mined is quite large but does not play a decisive role in the difficulty, which makes the model redundant and complex, and expensive to train.

9 References

[1] “Wordle-The New York Times.” The New York Times, 2022. Accessed December 13, 2022 at https://www.nytimes.com/games/wordle/index.html.
[2] “Wordle-The New York Times.” The New York Times, July 21, 2022.
[3] “Wordle Stats.” Twitter, July 20, 2022.
[4] Dataset of English Word Frequency at https://www.kaggle.com/datasets/rtatman/english-word-frequency
[5] Dataset of Wordle Valid Words at https://www.kaggle.com/datasets/bcruise/wordle-valid-words
[6] Letter frequencies as recorded in Wikipedia at https://zh.wikipedia.org/wiki/%E5%AD%97%E6%AF%8D%E9%A2%91%E7%8E%87

10 Letter for the Puzzle Editor of the New York Times

Dear Puzzle Editor of the New York Times,

We were fascinated by the daily guessing game wordle that you provided. Our research team was all excited about guessing the words in the fewest number of steps. In addition, we mined the data behind the Wordle game, analyzed the effects between the data, and made a mathematical model that might be helpful to you. Below we explain what we did and how the model was built.

We referenced data from January 7 to December 31, 2022, and found that

the number of people sharing their results on Twitter was gradually changing each day. We built a model to explain this change and hopefully give an interval for the number of people sharing their wordle results on Twitter at a future date, and we used common time series forecasting methods like Arima and LSTM to fit and predict the number of future shares on Twitter and give a prediction range.

We not only consider the nature of the words themselves but also refer to external data such as word frequencies. We selected and calculated features such as letter frequency, diacritic frequency, letter repetition, number of vowels and consonants contained, word commonness, letter code on each bit of the word, etc. We trained and compared several models using machine learning methods, and finally chose decision trees to predict the proportional distribution of the number of attempts for different words. For example, for the word 'eerie' given on March 1, 2023, we predicted the percentage of attempts (1%, 3%, 4%, 28%, 28%, 21%, 15%). This prediction translates into a difficulty factor we define (i.e., a weighted average of the number of attempts) of 5.02, which is equivalent to saying that this is harder to guess than the vast majority of words!

We designed a model to classify the difficulty level based on word characteristics. First, the distribution of the number of attempts of each word in the data set was fitted to a Gaussian function, and the expectation was calculated as the index of the K-mean clustering algorithm, and the words were classified into five difficulty levels by clustering, (54,127,113,57,6). The word 'eerie' was predicted to be at the hard difficulty level. And according to the method we proposed in the paper, we can calculate the accuracy of this classification as 84.92%.

In the process of designing the model, one of the main problems we encountered was that there was no way to find a dominant factor to determine whether the puzzle was hard or not. We had to draw on some literature to calculate the word frequency impact, the initial letter impact, the letter frequency impact, the two-letter group frequency impact, etc. for each word in the original dataset ..... Of course, in addition to these features, there are many valid features and relationships between features that can be mined in the original dataset. For example, the percentage of people choosing difficult patterns is slowly increasing, and our procedure shows that this is related to some of the word features as well. A word can be broken down into features such as the number of vowels, vowel position, letter repetition, etc. By mining these features, we can make our training model more accurate.

Last but not the least, I am writing to express my appreciation for your puzzles. They are always challenging and engaging, and I look forward to seeing a new puzzle every day. I think that playing Wordle puzzles is a great way to exercise my mind and learn something new every day.

Sincerely,
three college students