Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2310767

Uncover the Puzzle of Words : Evidence from Wordle

Summary

Using up to six guesses, players are asked to predict a five-letter word in Wordle, which provides feedback after each guess. We all know that different words have various characteristics that could influence how well a player performs. In this regard, our team modeled and evaluated the provided data to arrive at some intriguing conclusions.

For task 1, after data preprocessing, we first use the ARIMA model to forecast the number of reported results and find that it can only capture the linear part. Second, we take the LSTM model to capture the information of the non-linear part. Finally, we combine them to form **ARIMA-LSTM model**, which yields more accurate predictions with an RMSE of 0.0432. We finally arrive at a prediction interval of **[9614,43109]** for March 1, 2023. Subsequently, we define five word attributes, such as syllable count and entropy, and analyze their correlation with the percentage of people reporting scores in the difficult mode through the **Spearman correlation coefficient**, and find them to be significantly correlated.

For task 2, we use **five** preferred word attributes and competition number to predict the distribution of results using a **stacking model** that combines linear regression models (Ridge regression, Lasso regression) and tree models (XGBoost, LightGBM). We find that the stacking model improves the goodness of fit of the prediction results to **83.77%**. Moreover, the MSE, RMSE, and MAE all indicate that the stacking model has a better capacity for learning. On March 1, 2023, the anticipated distribution of "EERIE" will be [1,2,3,4,5,6,X]=**[0,0,9,18,26,37,10]**.

For task 3, we select seven additional word attributes to measure the words' difficulty level, and then downscale the metrics by **principal component analysis (PCA)**. We then use **Gaussian Mixture Model(GMM)** to cluster the words into three categories: difficult, moderate and easy. To get the true difficulty of the words, we calculate the expected number of tries for each word, which is used to compare with the classification results. It demonstrates that our model has a **67% accuracy rate**. We explore interesting findings on the properties of the given words associated with each classification from three different perspectives: entropy, number of letters and frequency. Finally, we classified "EERIE" into the category of "**difficult**" based on its attribute.

For task 4, we perform a visual analysis of the provided dataset and discover some intriguing properties in three areas: (1) the number of reported results and the percentage of players who try the hard mode; (2) the distribution of tries; and (3) the frequency of letters in each position. These characteristics provide some interesting and feasible ideas for players to solve the problem.

We also conduct sensitivity analysis, which shows how different samples affect the word difficulty clustering model. And then the strengths and weaknesses of our model are summarized. Finally, a letter to the editor of the New York Times presenting the overall ideas and results of our paper is written in the end of paper.

Keywords: ARIMA-LSTM Model; Correlation Analysis; Stacking; PCA; GMM; Wordle;

Contents

1 Introduction

Wordle is a daily puzzle offered by The New York Times that requires players to guess a five-letter real word within six attempts to solve, and is now available in more than 60 languages. Players receive feedback for each guess, specifically that the tile will change color to give feedback after the word is submitted. Players can play in either regular mode or hard mode. Hard mode requires that once the player has found the correct letters in a word, those letters must be used in subsequent guesses, which adds to the difficulty of the game.

As Wordle grew in popularity, The New York Times wanted us to analyze and model the data it provided.To address these issues, our team will take the following steps.

- Perform outlier processing on the data provided by COMAP official.
- An ARIMA-LSTM model was built to predict the number of reported results.
- Construct 12 word attributes and indicators reflecting the difficulty of the words.
- Develop models that can predict the distribution of reported results based on the given words at future dates using Stacking model fusion algorithms.
- Calculate mean square error (MSE), mean absolute error (MAE), and R_2 as evaluation metrics to verify the accuracy of the model.
- Classification and identification of solution words based on GMM clustering using the constructed word difficulty metrics.
- Describe the number of player guesses, the position of the 26 letters appearing in the word, and other interesting properties we found.
- Write a letter for the editor of the New York Times that includes our prediction results and interesting findings.

Our modeling framework is shown in Figure 17.

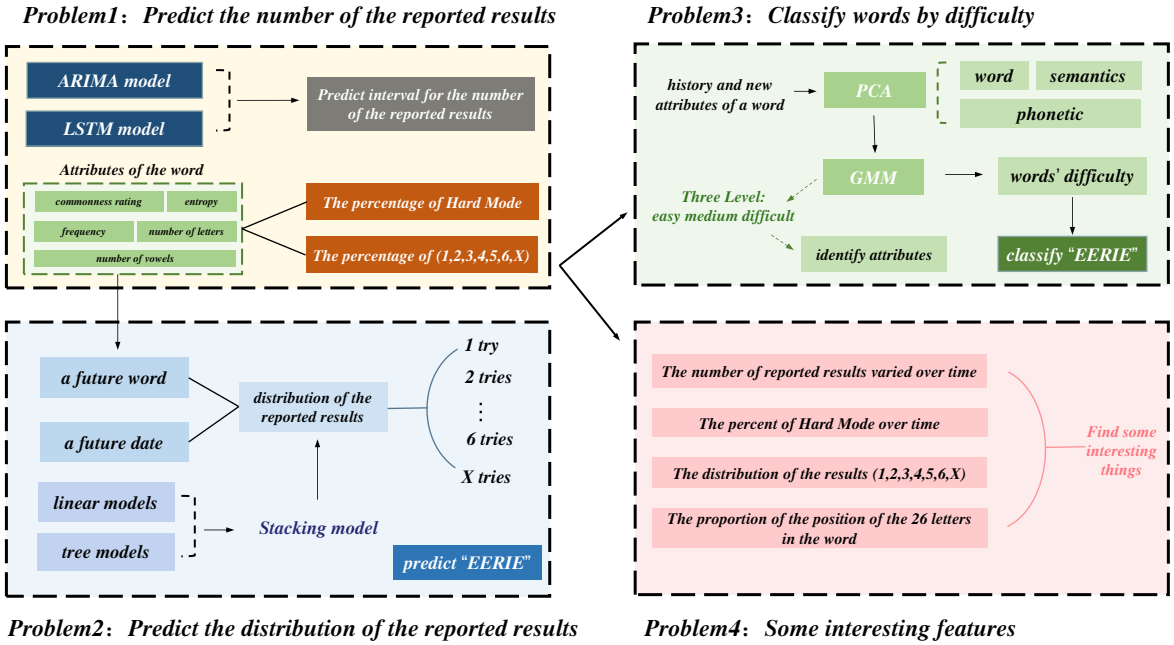


Figure 1: Workflow

2 Assumptions and Symbols

2.1 Model Hypothesis

To simplify our problem, we make the following basic assumptions, each of which is sufficiently reasonable.

- **Assume that on January 1, 2023, March 1, 2023, there are no unexpected factors that will change the data substantially.**
  Since we can only use the data in the additional data file, which is up to December 31, 2022, we ignore the heat that this topic may bring to Wordle later or other influencing factors that may cause drastic fluctuations in the data.
- **Assume that the scores that users report on Twitter are true and reliable.**
  If users report their scores on Twitter as false, the distribution of reported results in the dataset may be difficult to predict using only the properties of a given word for a given date.
- **The disturbance term is assumed to follow an independent normal distribution.**

2.2 Symbols and Definitions

Table 1: Notations

Symbols	Description
r_t	The number of results reported on day t
$H(X)$	The information entropy size of word X
ρ	Spearman rank correlation coefficient
f_r	Word attribute: a measure of common usage in the coca dataset
f_n	Word attribute: word frequency of SUBTLEX-US corpus
f_{tn}	Word attribute: number of lexical labels
f_d	Word property: consonant doubling
f_p	Word property: phoneme length
f_{vow}	Word property: number of vowels
f_c	Word attribute: specificity rating
λ	GMM model parameters
$p_k(d_i)$	Gaussian component density function
μ_k	Mean vector of the kth Gaussian component

3 Data Pre-processing

Since we can only use the official COMAP dataset 'Problem_C_Data_Wordle.xlsx', and the given data is obtained by mining Twitter, there is a possibility of data anomalies, so we pre-processed this part of the data before building the model.

- Fill: We replace the outliers in Number of reported results with the average of the before and after data.

- Reject: We remove the entire data where the sum of the distribution of the reported results deviates from 100%.
- We remove the entire word that has a number of letters not equal to 5, including "clen" and "tash".

4 Task 1: Interval Prediction & Correlation Analysis

4.1 Prediction for the Number of Reported Results

4.1.1 Autoregressive Integrated Moving Average Model

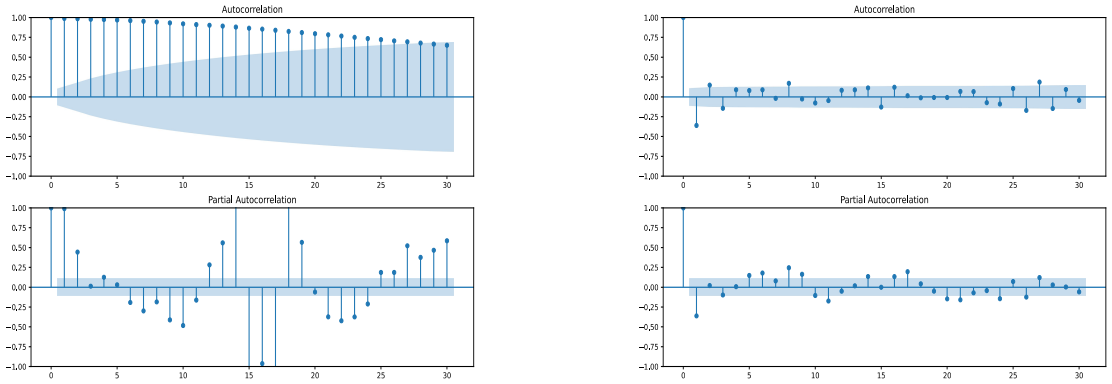
Since the data are time series and the amount of data is small, after considering various prediction models, we first chose the ARIMA model to predict the number of reported outcomes.

4.1.2 Building Predictive Models

In the ARIMA (p, d, q) model, AR is the autoregressive, p is the autoregressive term; MA is the moving average, q is the number of moving average terms, and d is the number of differences made when the time series becomes stationary. The model is based on the principle of conver$r_t$ing a non-stationary series r_t into a stationary series r_t by differencing of order d. The regression is then performed with r_t as the dependent variable and the lagged term of r_t and the lagged terms of the random error terms at and at as the independent variables. For simplicity of writing, the latter denotes the number of reported results in terms of r_t sequence.

Step 1. Sequence smoothing (determine the parameter d)

First of all, we perform a smoothness test on the time series r_t of Number of reported results. As can be seen from figure2 below, the two time series have a clear trend and the autocorrelation coefficients decay relatively slowly. Also, we performed a unit root test and found that the series has a unit root.



(a) Sequence r_t (b) Sequence $\bar{r}_t$

Figure 2: Autocorrelation plot and Partial Autocorrelation plot of rt sequence

Therefore r_t is a non-stationary series and needs to be further smoothed.

Let $\bar{r}_t$ = r_t-r_{t-1}. After taking the first order difference, the smoothness test is performed on the differenced r_t. From the following figure3, we can observe that the series after difference always fluctuate randomly around some value and there is no significant trend. The autocorrelation coefficient decays rapidly, and only the closely spaced series values have a significant effect. Also, the p-values of the unit root tests all converge to 0. Therefore, there is no unit root. r_t is already a smooth series. Since we use the first-order difference method to obtain the smooth series, $d = 1$.

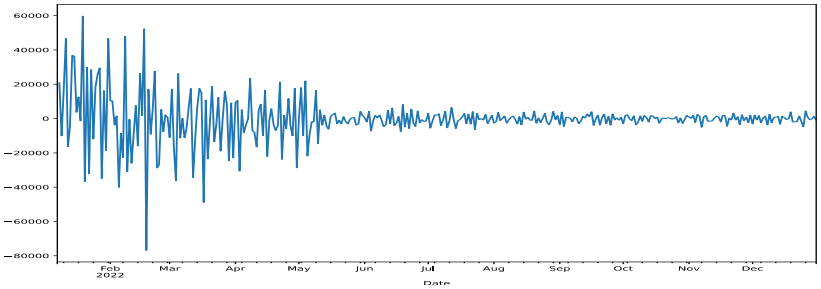


Figure 3: Time series plots of sequence $\bar{r}_t$

Step 2. determination of p and q order.

The ARIMA(p, d, q) model takes the form:

$$\bar{r}_t = r_t - r_{t-1} \tag{1}$$

$$\bar{r}_t = \phi_0 + \sum_{i=1}^{p} \phi_i \bar{r}_{t-i} + \varepsilon_t - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2}$$

where ε_t is a white noise sequence and p and q are non-negative integers. The Bayesian information criterion (BIC) is commonly used to determine the optimal order of the model, which is constructed based on the likelihood function. Based on historical data, the BIC values of the model at different orders are calculated by computer programming loops to find the order p and q that minimizes the BIC, i.e., the optimal order of the model. After determining the optimal order, we perform parameter estimation and test that the data are found to be statistically significant.

Step 3. residual test

To determine the validity of the model, a residual test is also required, in which a white noise verification of the residual series ε_t is required. If the residuals are randomly normally distributed and not autocorrelated, the residual series approximates the white noise series, indicating a good model fit. We use the Ljung-Box statistic $Q(m)$ to test the proximity of white noise:

$$Q(m) = T(T+2) \sum_{l=1}^{m} \frac{\hat{\rho}_l^2}{T-l} \tag{3}$$

When the p-value of the test is greater than 0.05, it means that the residual series ε_t passes the test at 5% confidence level, i.e., the residual series is white noise. The test of the residual series of the data in this paper finds that it is not white noise, which means that there is still useful information in the residuals and the model needs to be modified to extract further information.

The ARIMA model's prediction of the reported outcome quantities lacks a residual term that is not accurate enough, which is especially evident in the prediction outside the original data. However, the ARIMA model is still able to capture the trends in the reported number of outcomes well, which means that it can predict the linear part of the reported number of outcomes well.

4.1.3 ARIMA-LSTM Model

The prediction results of the ARIMA model are not very volatile due to the fact that the residual part of this result is not well predicted reasonably well, so this paper further improves the prediction model by combining ARIMA with LSTM neural network.

LSTM neural network nonlinear prediction model Long short-term memory network (LSTM) is a modified recurrent neural network that can solve the long-term dependence problem and remembering information for a long time is actually their default behavior.LSTM adds a memory unit to each neural unit in the hidden layer based on RNN: the information transmission band called "cell state "The LSTM uses structures such as forgetting gates, input gates, and output gates to control the memorized information on the time series. In this way, LSTM can dig deeper into the underlying patterns between the data, making the prediction more accurate and reliable.

Therefore, in order to compensate the shortcomings of ARIMA model and further improve the prediction accuracy, we use a combined linear and nonlinear model for prediction. To this end, we firstly, based on the ARIMA prediction results and the actual number of reported results on the residual series of number of reports, which is used as the expected output of the LSTM neural network; secondly, the phase space reconstruction of the original data is performed, and the optimal number of reports is finally determined as 18; thirdly, the data reconstructed in the optimal order is used as the LSTM input; fourthly, the training set is input into the LSTM neural network , the learning modeling and prediction of the residual series test set is performed to obtain the ARIMA residual series prediction value; finally, the prediction results of ARIMA and LSTM neural network models are summed to obtain the final prediction results of the reported number of outcomes.

4.1.4 Analysis of Results

The prediction results are shown in Figure 4. The predicted value curve and the actual value curve still basically match, and the fluctuation trend remains consistent, and the model can accurately predict the inflection point, and the prediction results are better. And compared with the single model prediction results, the results are more volatile and more realistic in terms of the number of results reported in real life.

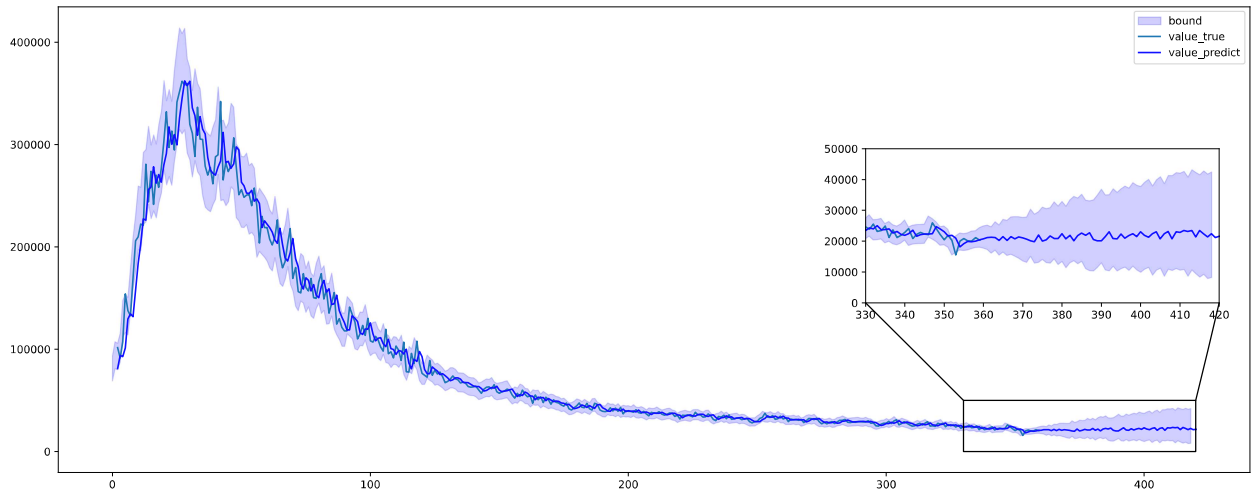


Figure 4: ARIMA-LSTM Model - The number of results reported forecast results

In this section, the ARIMA model was used to predict the number of reported outcomes in the interval, and the LSTM model was used to predict the residual series to correct the short comings of the ARIMA model, and the final predicted value of the number of reported outcomes on March 1, 2023 was calculated to be 22577, with a prediction interval of [9614,43109].

4.2 Construction of Word Attributes

To explore whether any attributes of words affect the percentage of people reporting scores in the difficult mode, we first looked for attributes of a number of representative words. The following are the definitions of the attributes.

Number of syllables

The number of syllables reflects the number of constituent elements of a word. In this paper, we choose the method of counting syllables to measure word length. A syllable is the smallest unit that an individual can produce in a single breath and usually contains a vowel or vowels plus one or more consonants.

Word Class

A word class is a group of words that display the same formal properties. The data set given in the question contains seven main word classes: nouns, verbs, adjectives, adverbs, pronouns, conjunctions, and prepositions. Since pronouns, conjunctions and prepositions account for a small percentage, we categorize them as other.

Vocabulary usage index

Vocabulary usage index. Generally speaking, the more words are used in daily life, the easier it is for players to answer the puzzle, and conversely the rare words will increase the difficulty for players to pass the game.

The Corpus of Contemporary American English (COCA) is a collection of the most frequently used words in the English-speaking world. It is extracted from a large corpus of words. A big data approach was used to automatically generate a word frequency list from various genres (most representative US newspapers, magazines, fiction, academic, and spoken language from 1990-2012), which is considered to be the most accurate word frequency list available today. Therefore, with the help of COCA,

we defined the lexical common usage index as

$$f_i = ln(p_i) \tag{4}$$

where f_i the lexical common usage index and p_i is the order in COCA, and COCA is ordered by word frequency from most frequent to least frequent, the further back the word is, the less common it is.

The number of letters

The words given in the question are all composed of 5 letters, and having more than one repeated letter in a word can reduce the complexity of the word, which may reduce the difficulty of passing the game; it may also be due to the player's inertia that the word will be composed of five different letters, and the repeated letters will increase the difficulty of the game instead.

Word frequency

We consider the utilization of vocabulary on the web, where we use the number of relevant information on Google. With the help of Brysbert New's study of the SUBTLEX-US corpus, we obtained frequency counts for all the words in the dataset.

Information entropy

Information entropy can be used to describe the uncertainty of the source. In the game, Wordle will feedback the result of player input by changing the color of the tiles. The probability of each possible pattern to multiply the information content conveyed by this pattern to get the expected information content that a word can bring is also known as information entropy, and the higher the information entropy also means the higher the information content brought by this word in various situations. The statistics of the frequency of occurrence of 26 letters in English text from the COCA dataset are shown in the following figure5.

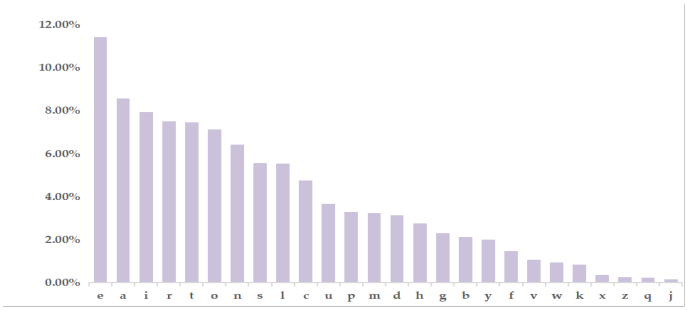


Figure 5: Frequency of 26 letters in the COCA dataset

When making a guess, the player will try to get the most information from each guess. Therefore an attempt will be made to cover the most high frequency letters in the first two attempts. For example, the combination of other + nails will cover 10 of the 11 most frequently occurring letters, and with a bit of luck some letters will be identified. Therefore, we use information entropy as a property of a word to describe the size of the information contained in the word. The greater the uncertainty of the word, the more information it contains, and the greater the information entropy. The specific formula is as follows.

$$H(X) = -\sum_{i=1}^{n} p\left(x_i\right) \log p\left(x_i\right) \tag{5}$$

where X denotes the word as a random variable, $P(x)$ denotes the output probability function of the word, and $H(X)$ denotes the information entropy size of the word X.

4.3 Correlation Analysis

4.3.1 Spearman Correlation Coefficient

We applied the method of correlation analysis to the five word attribute indicators constructed above and the percentage of scores reported that were played in Hard Mode, as measured by the spearman rank correlation coefficient, with the following idea.

The Spearman correlation coefficients of variables x and y were actually calculated using the rank order of the two columns of numbers. By arranging the variables $x = \{x_1, x_2, \cdots, x_n\}$ in ascending or descending order to obtain the sorted series $a = \{a_1, a_2, \cdots, a_n\}$, the position of each element x_i within the variable x in the series a is denoted as r_i , which is called the element x. By arranging the variables $y = \{y_1, y_2, \cdots, y_n\}$ in the same way, we get the rank series $y = \{y_1, y_2, \cdots, y_n\}$, and then we get the rank series $y = \{y_1, y_2, \cdots, y_n\}$. The rank series s corresponding to the variable y. The rank difference series $d = \{d_1, d_2, \cdots, d_n$
is obtained by subtracting the series r and each element in the series s corresponding to each other, and then substituting it into the Spearman rank correlation coefficient formula.

$$\rho = 1 - \frac{6 \sum_{i=1} d_i^2}{n\left(n^2 - 1\right)} \tag{6}$$

where: n is the number of samples, corresponding to the total amount of data involved in the ranking; ρ is the Spearman rank correlation coefficient between the variables of word frequency and the number of letters.

4.3.2 Correlation Coefficient Heat Map

The following graph depicts the relationship between the above metrics and the percentage of scores reported that were played in Hard Mode.

As can be seen from the figure6, most of the indicators do not correlate significantly with the proportion of players who choose the difficult mode, and only the number of vowels has a strong positive correlation with it.

Few attempts: 1 try, 2 tries, 3 tries Many tries: 5 tries, 6 tries, 7 or more tries(X)

The lexical commonness index, number of vowels and information entropy have a strong negative correlation with the proportion of guesses by few tries and a strong positive correlation with the proportion of multiple tries , with the opposite correlation for word frequency. This is consistent with our guess that the more common the word is in daily life and the more often it occurs, the easier it is to guess it by fewer attempts. The more informative a word is the less likely it is to be guessed within three attempts. Words with a higher number of vowels are more complex to pronounce and less likely to be game cleared after a small number of guesses.

The number of letters has a strong positive correlation with the proportion of in few attempts and a strong negative correlation with the proportion of multiple attempts,

probably due to human inertia that repetitive letters make the game more difficult and therefore more difficult to guess after a small number of attempts.

There is a smaller correlation between word lexicality and number of guesses, probably because players consider word lexicality less when guessing.

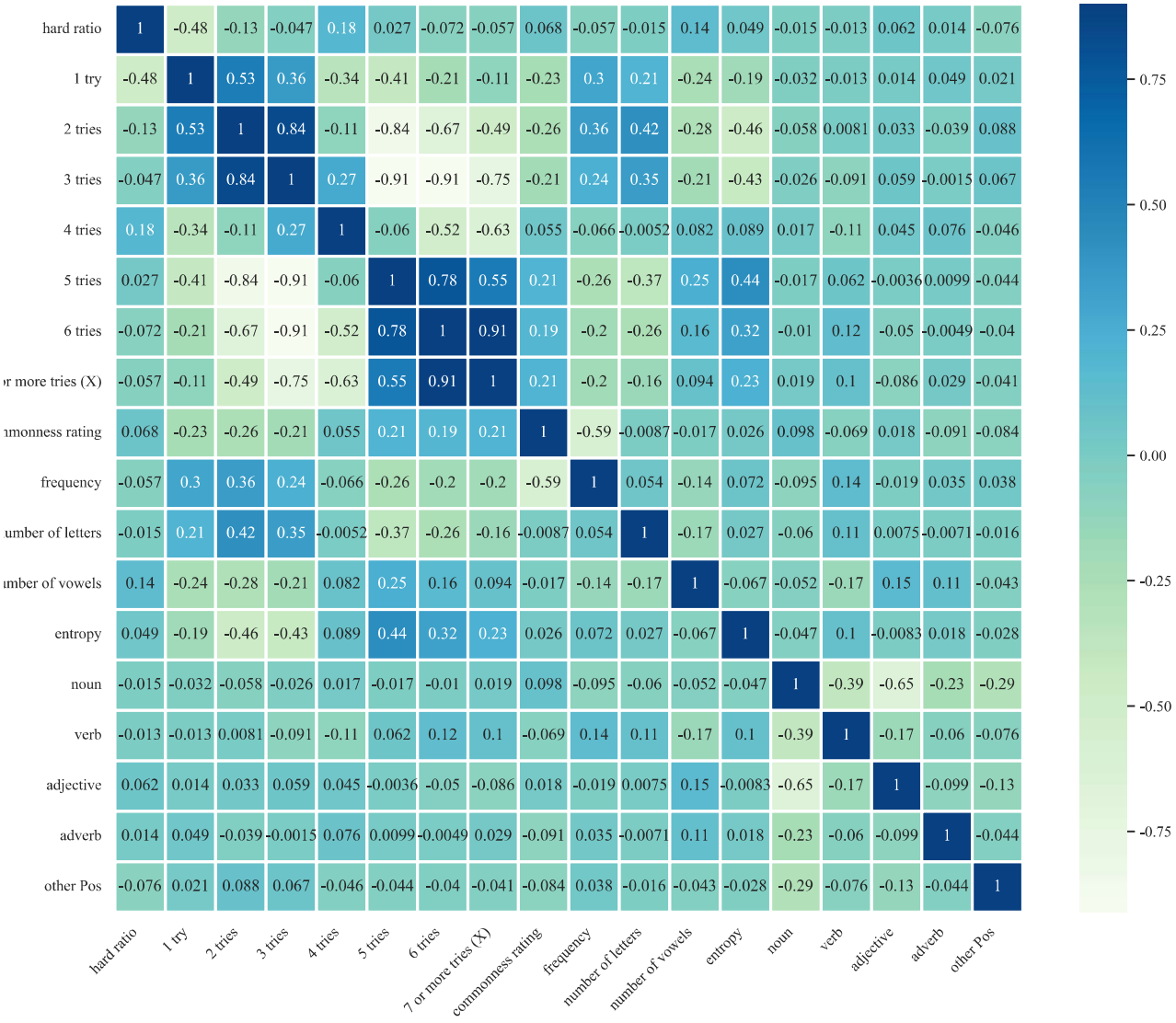


Figure 6: Heat Map of Correlation

4.3.3 Significance Analysis

For space reasons, we only give the correlation coefficients and significance levels of the indicators that were significantly correlated with the percentage of scores reported that were played in Hard Mode.

From the above table, we can see that there is a significant correlation (significant in the two-tailed test) between the five indicators for the number of syllables, the number of letters, common word index, word frequency and information entropy and the proportion of guessed or unsolvable puzzles (X) in one attempt, two attempts, three attempts, five attempts and six attempts, which is consistent with the results of the heat map analysis. In contrast, none of the four indicators was significantly correlated with

the proportion of guesses made at four attempts. The proportion of guesses at four attempts can be seen as the transition point, and often these four indicators are opposite in sign to the correlation coefficients between the proportion of guesses at less than four attempts and the proportion of guesses at more than four attempts.

Table 2: Spearman's correlation coefficient for each variable

x	com.rating	frequency	num. of letters	num. of vowels	entropy
$1try$	-0.231***	0.298***	0.208***	-0.243***	-0.192***
$2tries$	-0.264***	0.355***	0.424***	-0.276***	-0.464***
$3tries$	-0.206***	0.242***	0.347***	-0.207***	-0.428***
$4tries$	0.055	-0.066	-0.005	0.082	0.089
$5tries$	0.208***	-0.261***	-0.368***	0.247***	0.443***
$6tries$	0.189***	-0.202***	-0.263***	0.163***	0.317***
$Xtries$	0.212***	-0.201***	-0.164***	0.094***	0.231***

5 Task 2: Prediction of Result Distribution Based on Stacking Algorithm

In order to predict the distribution of reported results at a given future date and future word, we train seven stacking models to predict the specific proportion of each number of attempts (1,2,3,4,5,6,X) separately, and then normalize their corresponding predicted values as the final distribution prediction.

We built two linear regression models, Ridge and Lasso regression, and two tree models, XGBoost and LightGBM, based on the preferred word attributes (including ranking of commonness, word frequency, number of letters, number of vowels, and information entropy) and competition number, respectively, for comparison analysis, and The models were fused by integrating primary learners using the Stacking method, and the fused results have more accurate and efficient prediction effects.

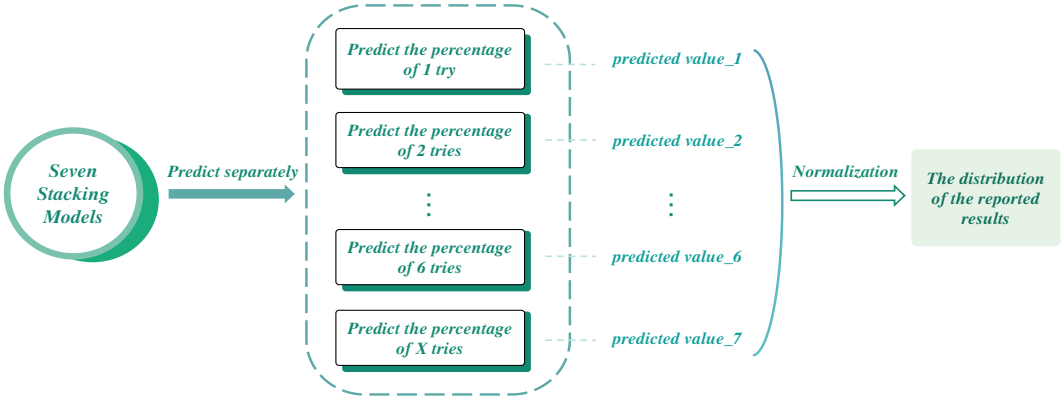


Figure 7: Distribution prediction flow chart

5.1 Stacking Method Model Fusion

The Stacking method is designed to reduce the generalization error of the models and is essentially a hierarchical structure of "stacking", which has the advantage of

improving the accuracy of the prediction results. In this paper, a two-stage Stacking method is applied, and the steps are as follows.

Layer 1.

1) In this paper, we split the data provided by the question into a training set and a test set in the ratio of 4:1, containing 286 and 71 data, respectively. 2) The training set is trained using Ridge regression, Lasso regression, XGBoost algorithm and LightGBM algorithm. 3) Predict the data provided by the question using the four models completed in Step 2 and save the results.

Layer 2.

1) The four prediction result datasets from step 3 of the first layer are used as new training set features, and together with the actual features of the data provided by the topic, the XGBoost algorithm with higher single model evaluation criteria in the first layer is selected as the metamodel for secondary training prediction. 2) The final results are obtained by predicting the validation set of the data provided by the topic with the trained XGBoost algorithm.

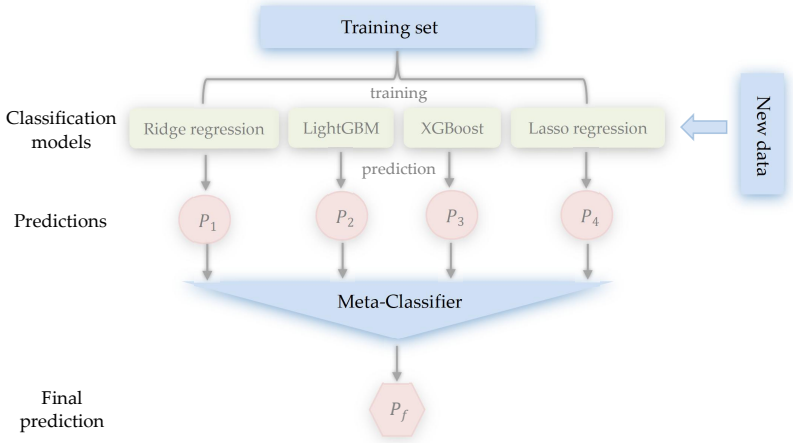


Figure 8: Stacking Model Fusion Flowchart

5.2 Introduction to Predictive Models

5.2.1 Linear regression model

Ridge regression adds the least squares of the second-order regular term to the loss function, also called L2 parametrization, which has the effect of dimensionality reduction, and also limits the matching of the model parameters to the abnormal samples and deals with the highly correlated data sets, thus improving the fitting accuracy of the model to most normal samples. Our team used RidgeCV to adjust the regularization strength alpha to achieve a better fit at alpha 14.

Lasso regression is similar to the above mentioned Ridge regression in that it also deals with the co-linearity of the feature variables by constructing a penalty function. However, compared with Ridge regression, Lasso regression can compress the coefficients of relatively insignificant characteristic variables to zero by changing the penalty term from L2 to L1 parametric, so as to eliminate the variables; whereas Ridge regression only compresses the coefficients of characteristic variables to a certain extent and

retains all variables of the regression model.

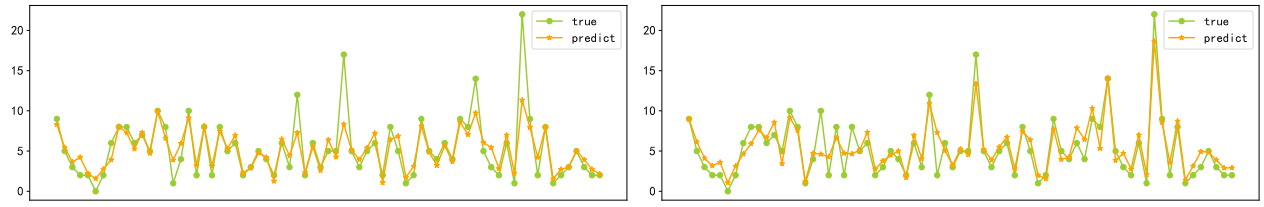
5.2.2 Tree Model

The core idea of XGBoost regression model is to calculate the information gain to reflect the degree of information uncertainty reduction of each feature variable. It is an optimized distributed gradient boosting library designed to be efficient, flexible and portable.

The LightGBM model is a decision tree algorithm that traverses the data to find the optimal splitting point based on the discrete values of the histogram. Like the XGBoost algorithm, it is an efficient implementation of GBDT, and is similar to XGBoost in that it uses the negative gradient of the loss function as a residual approximation to the current decision tree to fit a new decision tree.

5.3 Prediction Results and Analysis

The reported results of the five models selected in this paper are shown in Fig9. The reported results of the linear model are shown in Lasso regression, and the regression results of the tree model are shown in LightGBM algorithm.



(a) Lasso regresion result (b) LightGBM result

Figure 9: Average Rating Value over the years

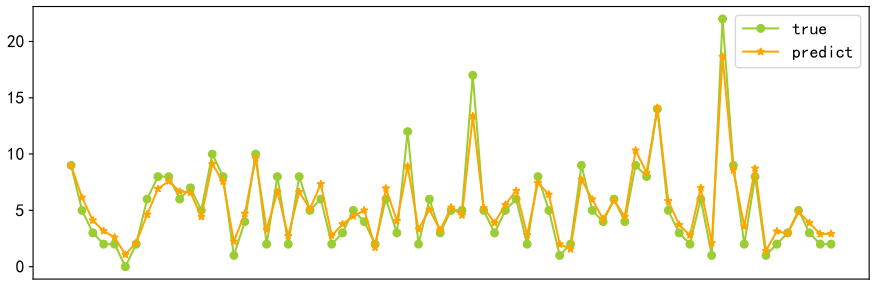


Figure 10: Stacking model result

The prediction results show that the linear model is more accurate in predicting data around the mean, while the tree model is more accurate in predicting the distribution of extreme values, and the Stacking fusion model retains the advantages of both models and compensates each other's shortcomings. The model prediction accuracy is higher. In conclusion, the forecasting effect of the Stacking fusion model is better. The ratio of predicted EERIE on March 1, 2023 is obtained as [1,2,3,4,5,6,X] = [0,0,9,18,26,37,10], respectively.

5.4 Model Evaluation

In this study, the cross-validation method was used to verify the accuracy of each model, and the mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and R2 generated from the cross-validation results were used as evaluation indicators for the accuracy of the estimated and validated models. The larger the R2 corresponding to the estimation model, the smaller the MSE, RMSE and MAE indicate the higher the prediction accuracy of the model. The following table shows the model prediction results of various models with four indicators: (with "2 tries" as the dependent variable)

Table 3: Spearman's correlation coefficient for each variable

	MSE	RMSE	MAE	R_2
$Lasso$	2.3771	1.5417	1.0727	0.6847
$Ridge$	2.1292	1.4592	0.9909	0.7151
$XGBoost$	1.9289	1.3888	0.9186	0.7856
$LightGBM$	1.8562	1.3642	0.9045	0.8062
$Stacking$	1.6768	1.2949	0.8623	0.8377

Compared with the single model above, this paper finds that the R2 of the Stacking model improves, and MSE, RMSE, and MAE all decrease in varying degrees. However, due to the variability and unpredictability of player behavior, as well as the possibility of unexpected events or external factors affecting player participation and performance, even the Stacking fusion model still has shortcomings in predicting the distribution of reported outcomes.

6 Task 3: Gaussian Mixture-Based Classification Model

To solve the third problem, we first selected seven English word attributes and features to measure the difficulty level of the words. The English words in the dataset were classified by visualization analysis and clustering of GMM through principal component analysis (PCA) for dimensionality reduction of the metrics. Finally, the classification results were compared with the real difficulty of the words, and the results showed that the real classification results overlapped highly with our GMM clustering results, confirming that the word features selected in this paper have a close relationship with word difficulty.

6.1 Word Attribute and Difficulty Index Establishment

Dimension 1: Words

Frequent use (fr) Commonness is used to measure how often a word is used in daily life. This indicator is calculated by taking the logarithm of the ranking order of the common words of the word in the coca dataset, with higher values indicating a lower ranking, i.e., less common. Word frequency (fn) We considered the frequency of word usage and obtained frequency counts for all words in the dataset with the help of Brysbert New's study of the SUBTLEX-US corpus. Number of letters contained (fd)

We consider that a word is made up of several letters, and letters that appear multiple times are not counted repeatedly.

Dimension 2: Phonology

Phoneme length (fp) Many English words contain unaccented consonant letters, thus making them more difficult to remember. Therefore, we collected the phonetic length of each word to express the number of silent letters that may increase the difficulty of word memorization. By applying the CMU pronunciation dictionary data from NLTK, we obtained the phonetic sounds of all words.

Number of vowels (fvow) Beinborn et al used the ratio of vowels to consonants to find that words with very high and very low vowel ratios are more likely to cause spelling errors and are more difficult to spell than words with moderate vowel ratios. We directly introduce the number of vowels as a second phonological feature.

Dimension 3: Semantics

Concreteness rating (fc) Concreteness ratings measure the degree to which the concept represented by a word is associated with a perceptible entity. According to Paivio's dual-coding theory, concrete words are easier to remember than abstract words because they activate perceptual memory codes in addition to verbal codes, thus making concrete words relatively less difficult. We refer to Brysbaert's findings to give the concreteness ratings of all words in the dataset given by the question.

Number of lexical labels (ftn) A word may have several lexical properties, which means that it can be used in a variety of contexts, and the more widely it is used the less difficult it is. NLTK, the Natural Language Toolkit, is what we use to label lexical properties.

6.2 Classification Model of Gaussian Mixture

In order to classify solution words, we built a word classification model based on Gaussian mixture model (GMM).

Gaussian mixture model (GMM) is a general probabilistic model. In general, as long as the Gaussian number is large enough, it can effectively model the continuous probability distribution of multidimensional vectors, and is therefore suitable for characterizing the semantic distribution of words and thus classifying them.

A Gaussian mixture model is a weighted combination of a series of Gaussian distributions. A Gaussian mixture density function consisting of M Gaussian components is a linear weighted sum of M Gaussian density functions.

$$p\left(d_i \mid \lambda\right)=\sum_{k=1}^{M} w_k p_k\left(d_i\right) \tag{7}$$

In the above equation λ is the GMM model parameter ,$p_k(d_i)$is the Gaussian component density function,w_k is the weight of each Gaussian component.

$$p_k(d_i)=\frac{1}{(2\pi)^{\frac{D}{2}}\left|\Sigma_k\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\left(d_i-\mu_k\right)^{\mathrm{T}} \Sigma_k^{-1}\left(d_i-\mu_k\right)\right\} \tag{8}$$

Here μ_k is the mean vector of the kth Gaussian component , $\sum_k$ is the corresponding

covariance matrix , and D is the dimension of the eigenvector . Thus, the GMM model can be represented by the following set of parameters :

$$\lambda = \{w_k, \mu_k, \Sigma_k\}, \quad k = 1, 2, \cdots, M \tag{9}$$

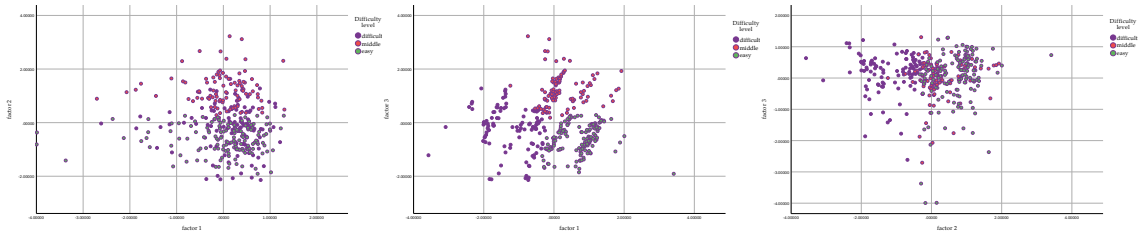
The use of GMM to classify words by word feature distribution is based on two starting points: 1) the Gaussian component of GMM can describe the distribution of certain word vectors; 2) the linear weighted Gaussian density function can approximate the probability distribution of arbitrary shapes, and we are uncertain about the probability distribution of the data, so GMM is chosen to classify words.

6.3 Visual Analysis of Single Word Features

For the features introduced in the previous section, we first perform power operations on the features such as consonant doubling fd, tag length ftn, phonetic fp, and specificity rating fc to obtain a better representation. Similarly, we take logarithms for the raw data of frequency fn. In addition, for word-to-vector features, we use Word2Vec's Gensim implementation to convert all words into a 100-dimensional vector as fvec. Thus, we combine all 7 features into a 107-dimensional feature vector (100 dimensions from word-to-vector feature fvec + 7 dimensions of 7 features, fr...fc).

$$Difficultyfeatures = (f_r, f_n, f_tn, f_d, f_p, f_vow, f_c, f_vec) \tag{10}$$

For intuitive analysis, we first downscaled the features to visualize the relationship between all features and difficulty. Therefore, principal component analysis (PCA) was used to reduce the dimensionality of the f features from 107 to 3 dimensions in Figure 12.



((a)) factor1 and factor2 ((b)) factor1 and factor3 ((c)) factor2 and factor3

Figure 11: Two-dimensional graph of clustering results

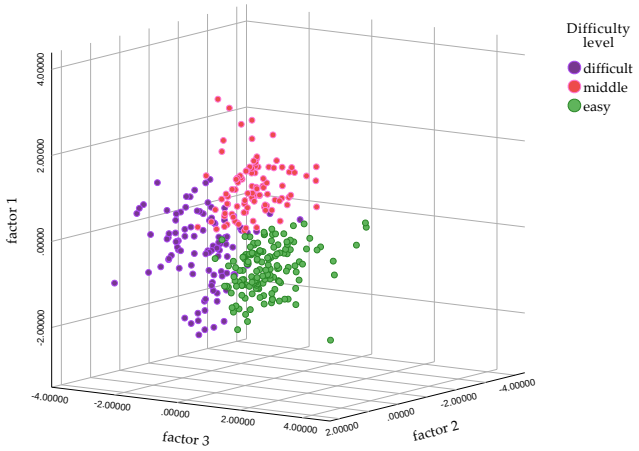


Figure 12: Three-dimensional graph of clustering results

In this figure, the three main axes represent the three common factors we obtained using factor analysis. The different colors indicate the different difficulty levels. Interestingly, Mummy, a word that appears a lot in everyday life, is classified in the difficulty category, which we believe is due to the presence of three identical letters "m" (which is hard to think of) making the word much more difficult. Although the first three principal components of the feature retain only about 72.8% of the variance, we can easily find that almost all words are grouped based on their difficulty level. This confirms that the features we have chosen are indeed closely related to the difficulty level.

6.4 Analysis of Classification Effect of Gaussian Mixture Model

To further classify the data, we used GMM to cluster the words. We called the Python scikit-learn package for GMM to train all the data and plotted the covariance of the data by the mean and covariance values. The results show that the covariances cover most of the data in their corresponding clusters. The total word counts of the GMM clusters for difficult, medium and easy words were 105, 100 and 150, respectively.

To explore the clustering effect in more detail, we defined the difficulty of each word based on the expected number of guesses needed to hit the word and compared it with the classification results, and the true difficulty of the word was defined as follows.

$$E(diff_i) = \sum_{j=1}^{n} j * p_j \tag{11}$$

where $E(diff_i)$ is the expected difficulty of the word, j is the number of times the word is guessed after j attempts, and if it is not guessed, we set the value of j to 7; p_j is the proportion of guesses made in j attempts, and similarly p_7 is the proportion of puzzles that cannot be solved. Finally, we sorted the expected difficulty in descending order, taking the trichotomies of difficulty and classifying them into three categories: difficult, moderate, and easy, to facilitate comparison with our prediction results.

In the three leftmost bars of the figure 16, we can find that the number of words with difficulty level 1 is the highest in this cluster. Accordingly, we named it "predicted difficulty level 1", which corresponds to the green area at the bottom right of the

figure above. The rightmost set of bars in Figure 2 has the highest number of words in difficulty level 3, and therefore corresponds to the purple area at the bottom left of the figure above, which is "predicted difficulty level 3". For the middle set of bars, since this cluster overlaps with the other two clusters, the prediction is not as good as for the difficulty level and the easy level, and we only get an accuracy of 60% for the clustering of words at the medium level of difficulty, i.e., the number of words with the same prediction level as the difficulty label as a percentage of the total number of words in each cluster. The middle bar corresponds to the red area at the top of the above figure13.

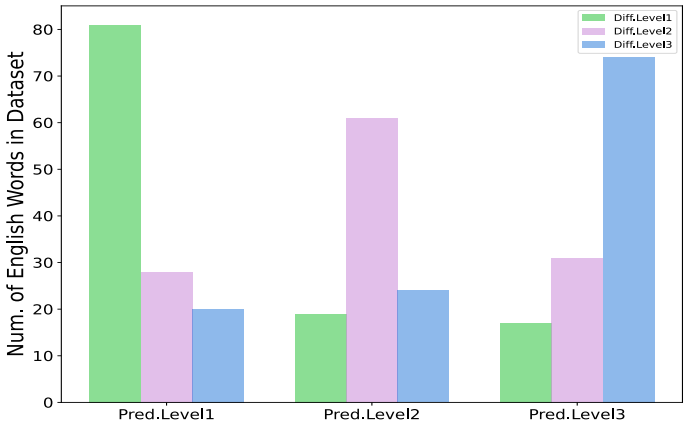


Figure 13: Distribution of prediction clusters with different difficulty levels

6.5 Identification of Word Attributes Associated with Classification Results

To identify the attributes of a given word associated with each category, we explored whether word attributes such as information entropy, The number of letters, and word frequency differed at different prediction difficulty levels.

Information entropy perspective

As shown in the figure14, the information entropy contained in words with difficulty level one is much smaller than that of words with higher difficulty levels, i.e., the higher the difficulty level of a word, the greater the information entropy and the more information it contains. This conclusion is consistent with the results of our previous analysis. The higher information entropy also means that the word brings higher information content in various situations, and therefore the higher the difficulty of the word.

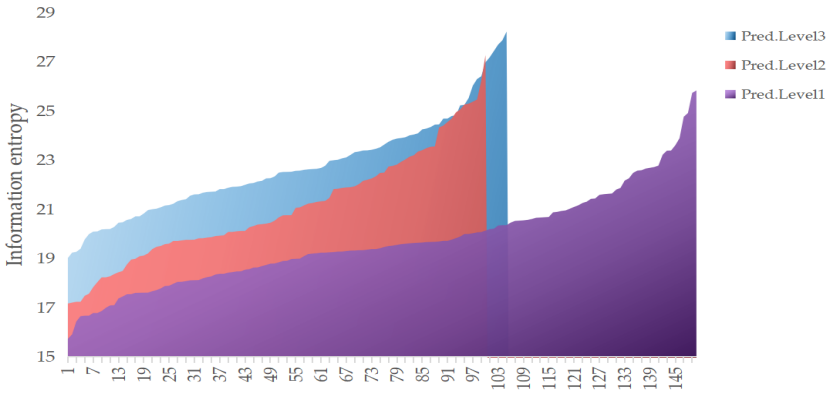
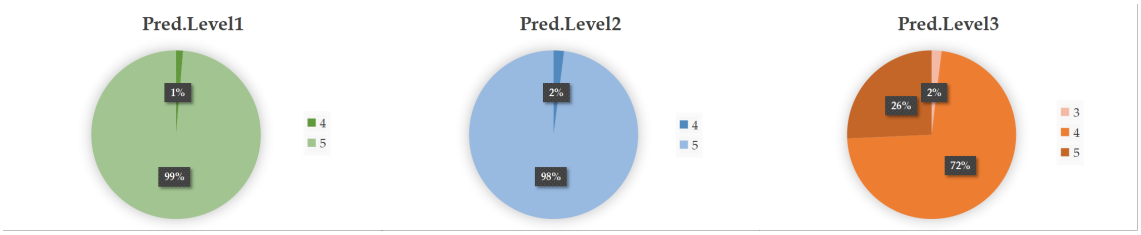


Figure 14: Information entropy of different difficulty levels

The number of letters perspective

As can be seen from the figure15, difficulty level one and difficulty level two contain words with five non-repeating letters, while difficulty level three contains 26% of words with two repeating letters and even 2% of words with three repeating letters, indicating that the presence of repeating letters in words does have an increasing effect on the difficulty of words.



((a)) 1 ((b)) 2 ((c)) 3

Figure 15: The number of letters by Difficulty Level

Word frequency perspective

The figure16 shows that words in difficulty level one appear more frequently in the database, while words in difficulty level three mostly appear between 0 and 7000 in the database, which shows that the more frequently words appear in daily life, the easier they are to be recognized, and the opposite is more difficult. This is consistent with our prediction results.

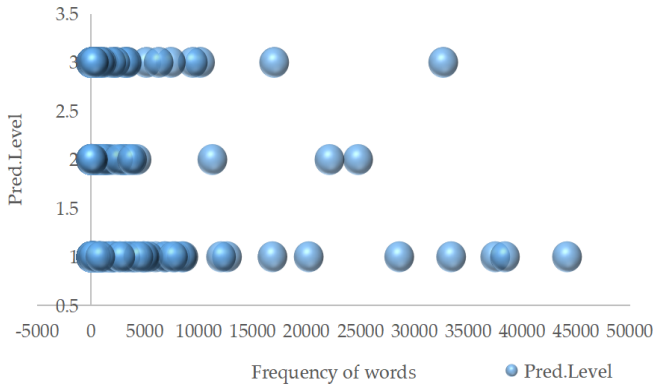
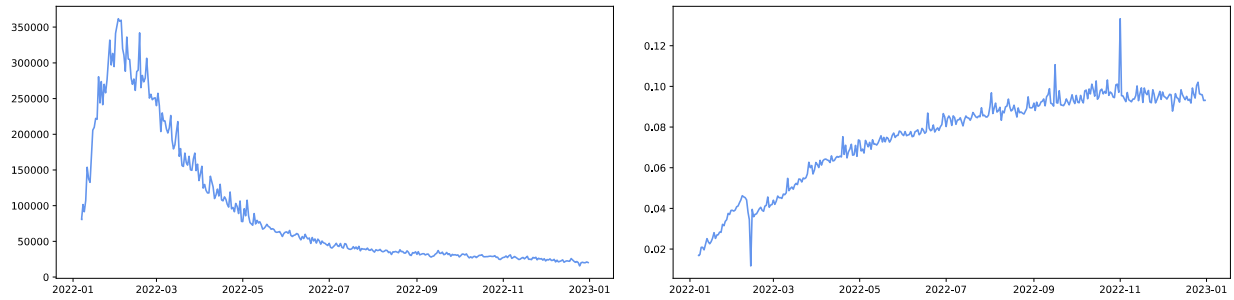


Figure 16: Word frequency of different difficulty levels

7 Task 4: Interesting Features

From the figure17, we find that the number of reported results shows a rapid growth trend in January and February 2022, and from March onwards the heat of Wordle slowly decreases, and the number of reported results shows a decreasing trend of recession type. After August 2022, the number of reported results does not change much, showing a fluctuating decline. The number of players who are willing to try the difficult mode tends to increase over time, with three large fluctuations, illustrating the

unpredictability and unpredictability of player behavior, as well as the unpredictable impact of unexpected events or external factors on players' participation in the game.



(a) Number of reported results (b) Hard Mode ratio

Figure 17: Player information time series trend

We plotted a line graph of the percentage of guesses or the percentage of puzzles that could not be solved (X) after 1-6 attempts and found that the percentage of guesses after 4 attempts was usually higher than the other percentages, followed by the percentage of guesses after 3 attempts, while the percentage of guesses in one attempt was almost equal to 0 (the percentage of results with 0 guesses in one attempt was 61.21% of the total results), and a few people were able to guess on the second (the average percentage of second guesses was 5.84%). In addition, the percentage of people who could not solve the puzzle (X) was more volatile, with a mean value of 2.80%, but a maximum value (word "parer") of 48%, indicating that there are words in the data that are above the ability level of most people, which makes the Wordle game more challenging.

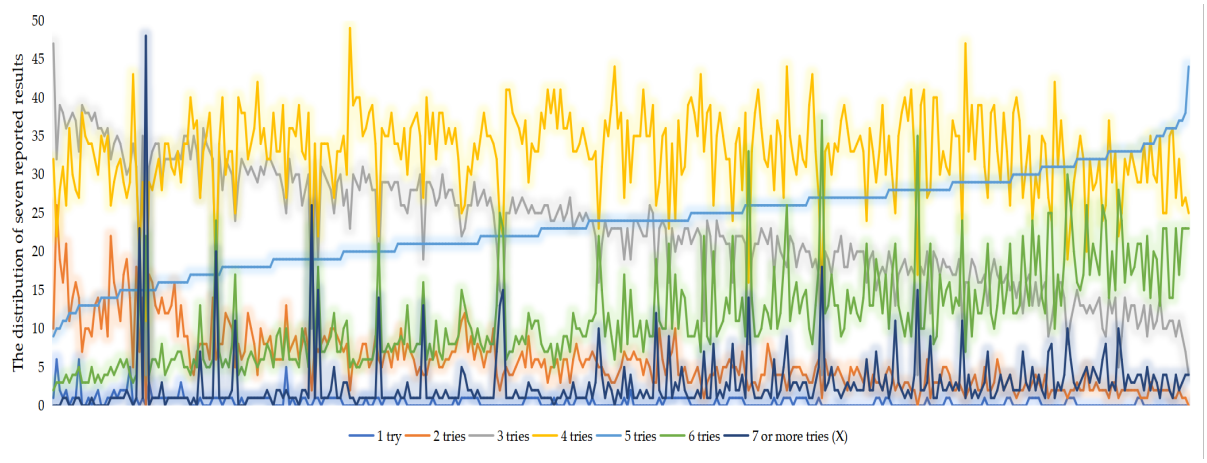


Figure 18: Proportion of correct guesses or unable to solve the puzzle after 1-6 attempts (X) (in ascending order of proportion of correct guesses in five attempts)

We explored the proportion of the 26 letters appearing in the positions of the words in the dataset, and the results showed that the five letters b, f, j, q, and s appear more frequently as the first letters of words, while d, l, t, and y appear more frequently at the end of words, and q appears only at the beginning and end of words. s appears most frequently in the position of the first letter, d appears most frequently at the last letter, and the middle three letters are n, o, and e appear most frequently. This can provide some interesting and feasible ideas for players to guess words and letters that are placed in words but misplaced.

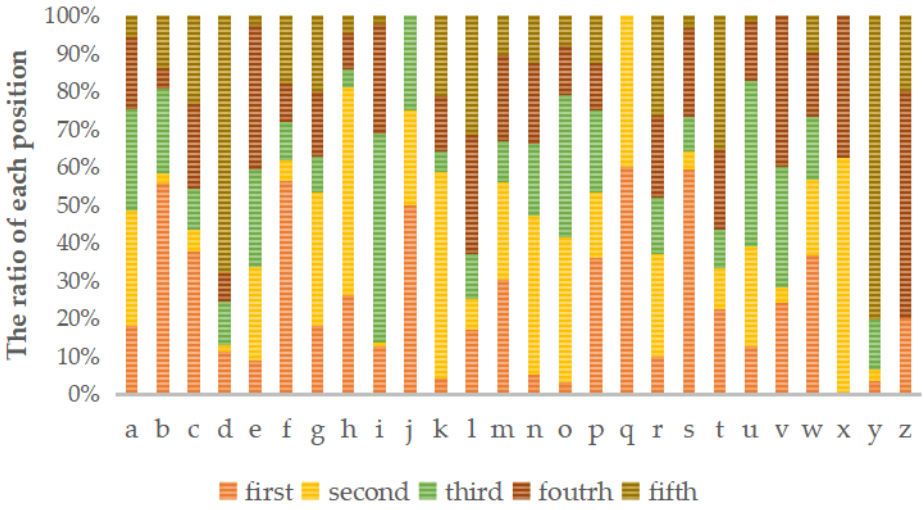


Figure 19: Proportion of the 26 letters appearing in words

8 Sensitivity Analysis

To test the reasonableness and generalizability of our GMM classification model, we expand the training samples to observe the changes in classification accuracy. The training sample in this paper comes from a word set W, which has 11998 English words that have been classified into three difficulty levels: easy, medium and difficult. Each difficulty level contains about 4000 words. Therefore we selected different training sets for words with letter numbers 4, 5, 6, 7, and 8. The number of words in each training set fluctuates up and down from 400, consistent with the amount of data provided by the question. Thus we get a total of 25 accuracies for different training sets of different words. We compared the difficulty labels calculated by the clustering model with the difficulty labels of the word set W. The accuracy rates were calculated according to the method described above.

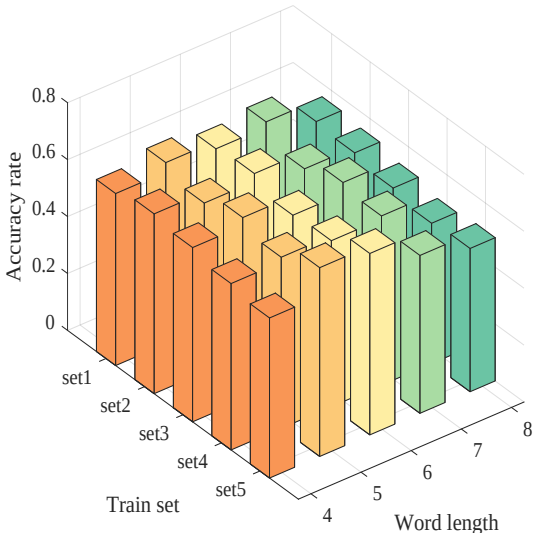


Figure 20: Accuracy of clustering for words containing different number of letters

As can be seen from the figure20, since the clustering model in this paper is trained on data containing words with five letters, the five training sets of words containing

five letters show better classification results, followed by the training sets of words containing four or six letters. The classification accuracy decreases as the number of letters of words increases, especially for the training set 5 of words with 8 letters, which has an accuracy of only 49.31%. However, the accuracy of words with the number of letters around 5 all fluctuate slightly above and below 60%, indicating that our GMM classification model is robust and suitable for classifying words containing 5 letters with difficulty.

9 Model Advantages and Disadvantages

9.1 Advantages

- By combining the ARIMA model with the LSTM model, both linear and non-linear cases are taken into account, thus focusing on both trend changes and volatility changes in the number of reported outcomes.
- The ARIMA-LSTM model can simply use the historical data of the number of reported outcomes themselves to predict their future trends and give the size of the prediction interval, which is more in line with real-life fluctuations than traditional mathematical and statistical methods.
- Based on Stacking model fusion algorithm combines multiple models to predict the distribution of results, which can give full play to the advantages of each model, has stronger learning ability and gets better prediction results, and provides a new idea for modeling.
- According to the results of sensitivity analysis, the GMM clustering model we established has excellent classification and recognition of the difficulty of words with 4-6 letters, and the model is applicable to a wide range of objects and has a high accuracy rate.

9.2 Disadvantages

- There are more computational indicators, which are tedious, and the model running speed needs to be improved.
- Due to time constraints, our description of word attributes and difficulty metrics may not be comprehensive enough, and more word attributes and larger English word difficulty data sets can be used for training in the future.

References

[1] Brysbaert, M., Warriner, A. B., Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. Behavior Research Methods, 46(3), 904–911.

[2] Paivio, A. (2013). Dual Coding Theory, Word Abstractness, and Emotion: A Critical Review of Kousta et al. (2011). Journal of Experimental Psychology: General, 142, 282–287.

[3] Lisa Beinborn, Torsten Zesch, and Iryna Gurevych, "Predicting the Spelling Difficulty of Words for Language Learners," Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 73–83, 2016.

[4] Brysbaert, M., New, B. (2009). Moving beyond Kucera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. Behavior Research Methods, 41(4), 977–90.

[5] A survey of trust and reputation systems for online service provision[J] . Decision Support Systems . 2005 (2).

[6] Hunter M. Breland, "Word Frequency and Word Difficulty: A Comparison of Counts in Four Corpora." Psychological Science, vol. 7(2), pp. 96–99, 1996.

[7] Edward Loper, and S. Bird, "NLTK: the Natural Language Toolkit," ETMTNLP '02 Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics, vol. 1, pp.:63–70, 2002.

[8] gensim models.word2bec-Word2vec embeddings, url: https: //radimrehurek.com/gensim/models/word2vec.html.

[9] scikit-learn Machine Learning in Python, url: http://scikit-learn.org/ stable/.

[10] Douglas Reynolds, "Gaussian mixture models," Encyclopedia of biometrics, pp. 827-832, 2015.

Dear Puzzle Editor,

Thank you for providing us with the file of the daily results of the interesting puzzle Wordle so that we could analyze and model the results of Wordle, which allowed us to discover some interesting conclusions about the given dataset. At the request of your journal, we are pleased to have the opportunity to present to you our findings and conclusions, which we hope will be of interest to you.

- **Results of interval prediction & correlation analysis**

We predicted the number of reported outcomes by constructing an ARIMA-LSTM model, which yielded a prediction interval of [9614,43109] for March 1, 2023. Subsequently, we defined five word attributes such as syllable count and word class, which were found to be significantly correlated with each other by Spearman's correlation coefficient analysis. Among them, lexical commonness index, number of vowels, and information entropy, had strong negative correlations with the proportion of guesses in one attempt, two attempts, and three attempts, and strong positive correlations with the proportion of guesses in five and six attempts or the inability to solve the puzzle (X); word frequency and letter richness were the opposite.

- **Distribution of results & EERIE distribution prediction**

After that, we developed the Stacking model fusion algorithm to predict the distribution of the results and came up with the predicted distribution of EERIE on March 1, 2023 as [1,2,3,4,5,6,X]=[0,0,9,18,26,37,10]. We also used R2 and others as evaluation metrics for estimating the accuracy of the model and validating it, and proved that the performance of our model is optimal with a goodness-of-fit of 0.8377.

- **Classification & EERIE Difficulty Prediction**

In addition, we classify words into difficult, medium and easy categories according to their difficulty, and classify them into the category of "difficult" according to the attribute value of "EERIE". From three different perspectives of information entropy, letter richness and word frequency, we found that the information entropy of words in difficulty level 1 is much smaller than that of words in higher difficulty levels, and they appear more frequently in the database, and the words in difficulty level 3 contain more repeated letters.

- **Some interesting features**

Finally, our visual analysis of the data revealed that: most people guessed the puzzle after 3 or 4 attempts; very few were able to guess the puzzle in one go; and the percentage of people who could not solve the puzzle (X) fluctuated more, up to 48%. Among the words in the dataset, b, f, j, q, and s appear more frequently as the first letter of the word, while q appears only at the beginning and end of the word, and s, n, o, e, and d are the most frequently occurring letters.

We appreciate this opportunity to predict Wordle results and analyze the interesting findings in the dataset. Feel free to contact us for further information about the article.

Yours sincerely

MCM 2023 Team