Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2300348

Revitalizing a Classic Game: Uncovering the Secrets of Wordle through Data Analysis

Summary

In the digital era, language is often conveyed through abbreviations, emojis, and voice messages. However, the Wordle game, provided by the New York Times, offers a chance to return to the basics of language. Thus, we conducted a data analysis of the results yielded by Wordle.

Firstly, we established a **GRU Prediction Model** to predict the number of reported results on March 1, 2023. The model uses the effective Gated Recurrent Unit (GRU) algorithm. Therefore, predictions made by the training set to the testing have **the relative error rate of 2.1569%, and the relative RESE of 6.4957%, indicating a good accuracy of the model predictions.** The predicted interval for the number of reported results on March 1, 2023 is **20367 ± 2.01569%**.

Secondly, we conducted a **data analysis** on the attributes of words and $score$ defined by the percentage of scores. Then, we defined **four attributes** of the words: word frequency, sum of letter frequencies,repetition patterns of letters(2/3 or none), and main part of speech. For the first two, we performed regression analysis with the variable "score". **The Pearson correlation** coefficient between f_{word} and $score$ is **-0.3165**, and f_{letter} and $score$**-0.4005**.rep and pos can be used to categorize the words.The box plot results showed that the **Median difference** of the box plot for rep was **0.13004**, while pos was only **0.05973**.**Therefore, we believe that** f_{word}**,** f_{letter}**, and** rep **can affect the percentage of scores, while** pos **can not.**

Thirdly, we have developed **GSRF Prediction Model** to predict the percentages of 1 to X for EERIE on March 1, 2023. The Grid-Search Random Forest (GSRF) algorithm is an improved random forest algorithm by using **the best combination of hyperparameters.** We selected the three parameters, f_{word}, f_{letter}, and rep as inputs for the model. The model's training results show a **MSE of 20.70641 and a MAE of 3.24388,indicating good predictive performance.**(Table 10) The predicted results for EERIE are **(1,7,23,30,23,13,3)**. In addition, we conducted **sensitivity analysis** by adding Gaussian noise to f_{word} and f_{letter} separately, and the results showed that the model has **low sensitivity** and is thus highly stable.

Fourthly, **Difficulty Rate Classification Model using the K-Means++are conducted**. We first defined the **difficulty date** δ of each word first. The difficulty rate of EERIE is **0.35916** by predicted distribution. Then, we use **K-Means++**, to analyze the δ of each word and obtained **five levels** of difficulty (Table11). **EERIE was classified into the third level.** Finally, we compared the model's classification with the manul difficulty ratings for a subset of sampled words and achieved **a match rate of 93.33%, confirming the accuracy of the model.**

Finally, we explored **two other data features**.Afterwards, a **letter** supported by our stable models has been written for the Puzzle Editor of the New York Times.

Keywords: GRU;Regression Analysis;Box Plot Analysis;GSRF;K-Means++

Contents

1 Introduction

1.1 Problem Background

In this digital age, we have become accustomed to using abbreviations, emojis, and voice messages to communicate. However, sometimes these modes of communication can strip away the beauty and depth of language itself. But a word-guessing game, Wordle allows us to return to the essence of language in a fresh way. By feeling the rhythm and meaning of each letter and word, we can gain a deeper understanding of the magic of language and appreciate its inherent charm.

Wordle is a popular daily puzzle that the New York Times currently provides, which challenges players to guess a secret five-letter word in six or fewer tries. At the beginning of each round, the system randomly selects a word, and the player must use deduction and logic to guess the answer within the allotted number of guesses. With each guess, the system indicates which letters appear in the word and whether they are in the correct position. What's more, players can play in "Hard Mode", which requires that once a player correctly guesses a letter in the answer, they must continue to use that letter in all subsequent guesses.

Numerous users, although not all, share their scores on Twitter. Wordle Stats, a Twitter robot developed by Benjamin Leis, can be used to track and analyze daily score reports of Wordle. By tracking and analyzing the daily data reports from Wordle Stats, we may be able to improve our word-guessing skills and gain a better understanding of the patterns and usage of the English language.

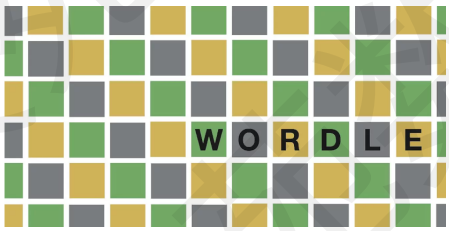


Figure 1: Wordle[2]

1.2 Restatement of the Problem

We need to analyze the data provided by the New York Times on Wordle and answer the following questions:

1. Develop a prediction model to forecast the number of reported results for a future day and provide a prediction interval.
2. Analyze whether word attributes have an impact on the percentage of scores reported that were played in Hard Mode, and how they impact.
3. Build another prediction model to forecast the percentages of (1, 2, 3, 4, 5, 6, X), and use EERIE on March 1, 2023 as an example to make specific predictions. Model evaluation is also required.

4. Develop a classification model to classify the difficulty level of words and obtain the difficulty level of the word "EERIE". Model evaluation is also required.

5. Explore other possible interactions in the data and see if any interesting features can be found.

1.3 Data Cleaning

Observing the provided data, some problems and corresponding processing are found as follows.

Referencing the Wordle Status and we discovered that there were some erroneous data in the attachment. These included incorrect word lengths, spelling errors, and incorrect values in Number of reported results, among others. For instance, the word at 314 was mistakenly written as "tash". Additionally, the Number of reported results at 529 was erroneously recorded as 2569. To address these issues, we conducted data cleaning to minimize errors , and to enhance the quality and accuracy of our data . The following are the results of our data cleaning process.

Original data	marxh(473)	tash(314)	clen(525)	rprobe(545)	2569(529)
Data after cleaning	marsh	trash	clean	probe	25569

Table 1: Data cleaning

During our data inspection, we identified instances where the sum of the proportion of attempts for certain days did not equal 100% due to statistical errors. To address this issue, we recalculated the proportions for each day so that the sum would be exactly 100%. By doing so, we aimed to reduce errors associated with the same variable on different days and to improve the accuracy of our model.Our processing result is $\sum_{i=1}^{X} p_i = 100\%$.

1.4 Our Work

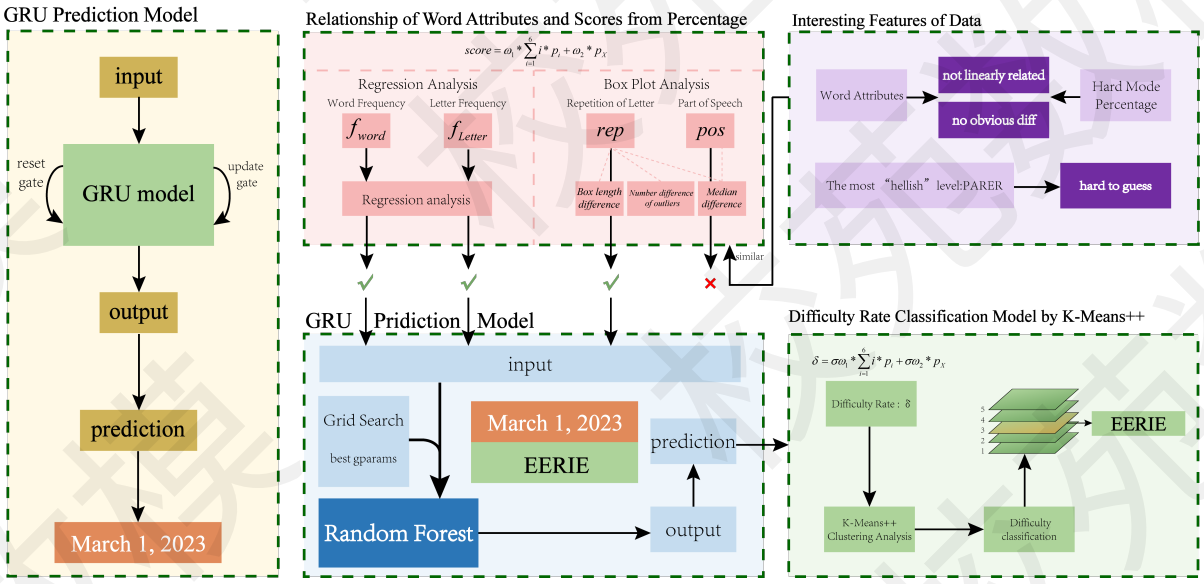


Figure 2: Our work

2 Assumptions

To simplify our modeling, we make the following assumptions:

- **Assumption1** The percentage of scores reported that were played in Hard Mode can be represented by the percentages of (1, 2, 3, 4, 5, 6, X). Because Wordle status had previously compiled the values of (1, 2, 3, 4, 5, 6, X) for both the overall and hard mode in Wordle 207, we found that the two sets of values differed by only 1%[1].This is beneficial for our subsequent analysis.
- **Assumption2** No matter how many tries a player takes, the likelihood of sharing their Wordle results on Twitter is equal. This assumption can lead to more accurate data analysis.
- **Assumption3** Every player does not know the answer before playing Wordle and does not cheat during the gameplay. This ensures that the percentage of scores is both objective and accurate.

3 Notations

Symbol	Decription
$score$	the combination of the percentage of scores
f_word	the word frequency
f_letter	the letter frequency
rep	the repetition of word
pos	the part of speech of word
σ	the difficulty rate
MSE	the Mean Squared Error
MAE	the Mean Absolute Error
d	the Euclidean distance
SSE	the Sum of Squared Errors

Table 2: Key notations used in this paper

4 GRU Prediction Model

The number of reported results varies from day to day, and analyzing the changes in this data can to some extent reflect the trends of active Wordle users. By studying historical data and trends, it may even be possible to make informed predictions about future reported results. So in this

section, we applied the Gated Recurrent Unit(GRU) algorithm to perform machine learning on the provided number of reported results, and ultimately made a prediction for the number of reported results on March 1, 2023.

4.1 Description of GRU Algorithm

GRU (Gated Recurrent Unit) is a type of recurrent neural network (RNN) that is commonly used for time series analysis. It has similar properties to the LSTM (Long Short-Term Memory) architecture but is generally faster to compute.

The main idea behind the GRU architecture is to have two gates, a reset gate and an update gate, which control the flow of information through the network. The reset gate decides how much of the previous hidden state should be forgotten, while the update gate decides how much of the new input should be added to the current hidden state.

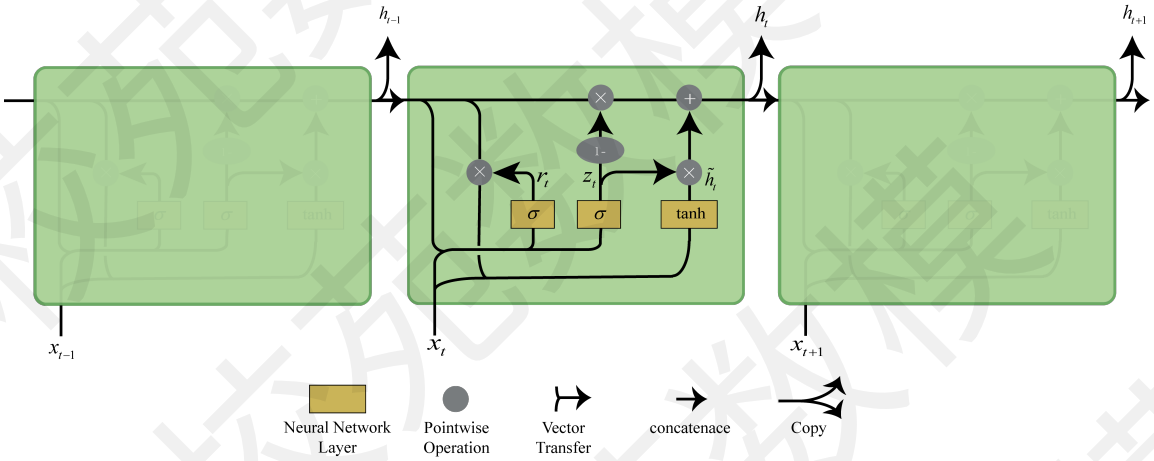


Figure 3: An Overview of the Algorithm Flow of GRU

The update equations of GRU are as follows:

$$\begin{aligned} r_t &= \sigma\left(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}\right) \\ z_t &= \sigma\left(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}\right) \\ \tilde{h}_t &= \tanh\left(W_{in}x_t + b_{in} + r_t * \left(W_{hn}h_{t-1} + b_{hn}\right)\right) \\ h_t &= (1 - z_t) * \tilde{h}_t + z_t * h_{(t-1)} \end{aligned} \tag{1}$$

where:

Symbol	Decription
z_t	the update gate
r_t	the reset gate
h_t	the hidden state at time t

Symbol	Decription
h_{t-1}	the hidden state at time $t-1$ or at time 0
$\tilde{h}_t$	the new candidate hidden state
σ	the sigmoid function
$*$	the Hadamard product
x_t	the input
$W_{ir}, W_{hr}, W_{iz}, W_{hz}, W_{in}, W_{hn}$	the parameters that need to be trained
$b_{ir}, b_{hr}, b_{iz}, b_{hz}, b_{in}, b_{hn}$	the parameters that need to be trained

Table 3: Notations used in Equation1

In this approach, the historical data of a time series is fed into a GRU model to learn the patterns in the sequence, which can then be used to predict future data points.

4.2 Prediction on March 1,2023

With the support of Python's extensive library, we have opted to use the GRU model provided by PyTorch. PyTorch is a machine learning library based on Python, and its distinctive feature is dynamic computational graph, which is different from static computational graphs. The dynamic computational graph can be changed during runtime, which means the model can be modified according to our needs. This is very useful when dealing with variable-length sequence data and is well-suited for predicting the number of reported results that we need to predict. In PyTorch, we can utilize the torch.nn.GRU[3] class to easily construct and train GRU models, as well as use the model for making predictions.

We used 80% of the daily "Number of reported results" time series data from January 7, 2022 to December 31, 2022 as the training set and the remaining 20% as the test set for our GRU model. The visualization of the prediction results on the test set is shown in Figure4:

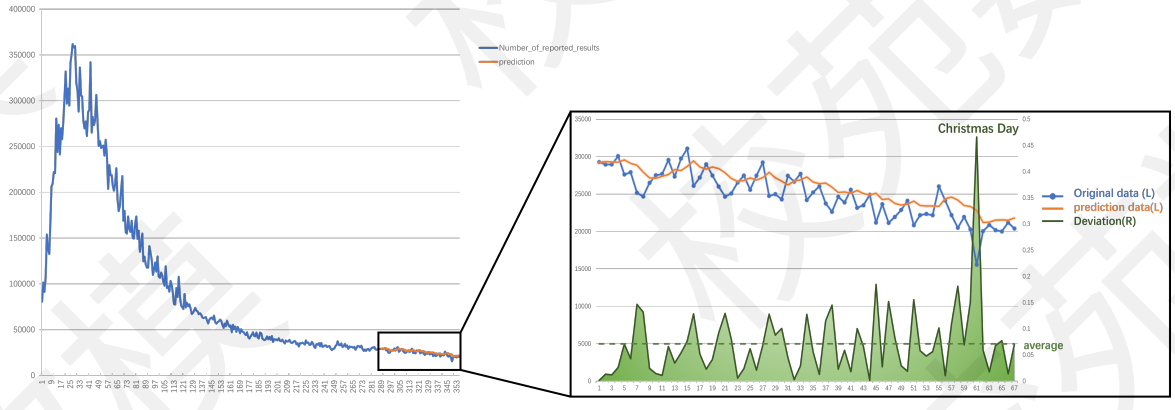


Figure 4: The prediction results on the test set."Deviation" represents relative error rate of between prediction data and original data.

From Figure 3, it can be seen that the average deviation is around 0.06, indicating that the prediction data deviate from the original data by only about 6%. At the same time, an interesting phenomenon is observed: the deviation for December 25, 2022, Christmas Day, is close to 50%, which is due to a sharp drop in the number of reported results on this day in the original data. It is easy to associate this with the fact that it is a major holiday. Such a result belongs to the outlier in time series analysis, and there is enough reason to remove this value and calculate the mean and root-mean-square error(RMSE) of the remaining data. And the calculation method for RMSE is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_t - \hat{x}_t)^2}{n}} \tag{2}$$

As shown in Table 2 below, the relative error rate between the prediction data and the original data is 2.1569%, indicating a small deviation between the two values. Meanwhile, the relative RMSE is calculated by dividing the RMSE by the original data, and in statistics, an error is considered small when this value is less than 10%. Therefore, it can be concluded that the GRU model has good predictive performance for the number of reported results, and we select the relative error rate of 2.1569% as the error interval for model prediction.

$\bar{x}_t$	$\bar{\hat{x}}_t$	Relative error	Relative error rate	RMSE	Relative RESE
24956.1667	25494.4569	538.29019	2.1569%	1621.0881	6.4957%

Table 4: Statistical analysis of prediction data and original data.

By leveraging the pre-trained GRU model, we can make predictions for the following 60 days and obtain a forecast interval for the number of reported results on March 1, 2023:

$$x_{March1,2023} = 20367 \pm 2.01569\%$$

5 Relationship of Word Attributes and Scores from Percentage

In this section, we conducted a data analysis on four attributes of words and $score$ which is derived from the percentage of scores. Through regression analysis, we found that word frequency and letter frequency of words are linearly correlated with $score$. Through boxplot analysis, we found that $score$ of words with different letter repetition patterns has some differences, while the score of words with different parts of speech does not show significant differences. The overview of the results are shown in Figure5

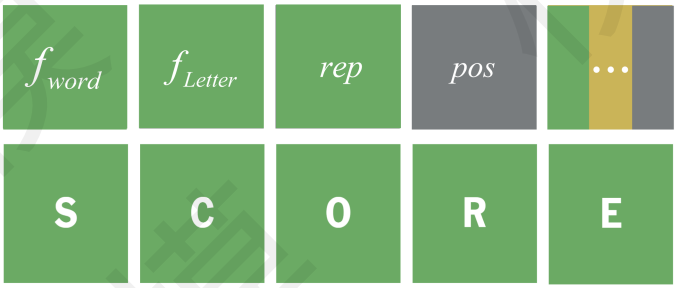


Figure 5: The overview of the results in Section 4

5.1 $score$ Defined by Percentage

The percentage of daily attempts is the percentage of scores on that day. However, when investigating the relationship between word attributes and percentages of scores, it is difficult and not conducive to analysis the relationship between a word attribute and the distribution of the percentage of scores composed of seven numbers. Therefore, we processed the seven percentage values into a single number, defining it as the $score$.

$score$ is composed of two parts. One part is the weighted average of tris from 1 to 6, and the other part is the percentage of "X". The reason for considering the percentage of "X" is that Wordle allows only one attempt per day, so the percentage of "X" actually represents the failure rate of guessing on that day, which cannot be ignored. However, since "X" does not have a specific number of tries, it cannot participate in the calculation of the weighted average, so we have divided the $score$ into two parts and assigned weights to them using the Entropy Weighting Method.$score$ is defined as follows:

$$score = \omega_1 * \sum_{i=1}^{6} i * p_i + \omega_2 * p_X \tag{3}$$

where:

- p_i denotes the percentage of i tries(try) and $i \in \{1, 2, 3, 4, 5, 6, X\}$
- ω_1 and ω_2 denote the weights of two parts of $score$ respectively. And we set ω_1 as 0.5 and ω_2 as 0.5 by Entropy Weighting Method.

5.2 Regression Analysis

5.2.1 f_{word}:Word Frequency

When attempting Wordle puzzles, words that are more commonly used in everyday language are often easier for people to recall, such as "study" and "train." Therefore, when the solution word has a high frequency of use, it is likely that the percentage distribution of tries will be skewed towards fewer tries, leading to a decrease in the value of $score$. To address this issue, we first used

a combination of website[4] and Python to obtain reliable usage frequencies f_{word} for each word. Then, we conducted regression analyses on both the f_{word} and $score$.

The regression analysis results of word frequency and $score$ are shown in Figure6(a) and Table3. The Pearson correlation coefficient is -0.3165, and the Spearman correlation coefficient is -0.2956, indicating **a certain linear correlation between word frequency and** $score$. Also, by observing the scatter plot in Figure6(a), it can be seen that there is no data distribution in the lower left and upper right corners of the image. This suggests that situations where a word has a high frequency but is difficult to guess and thus results in a low $score$, as well as situations where a word has a low frequency but is easy to guess and thus results in a high $score$, are unlikely to occur. Therefore, it is reasonable to conclude that there is a certain linear correlation between word frequency and $score$.

(a) Word frequency (b) Letter frequency

Figure 6: Regression analysis between word attributes and $score$

Regression with $score$	Pearson	Spearman
Word frequency	-0.3165	-0.3256
Letter frequency	-0.4238	-0.4005

Table 5: Regression analysis between word attributes and $score$

5.2.2 f_{Letter}:Letter Frequency

Each letter has its own usage frequency. When Wordle players use words with higher letter frequencies for guessing, they may have a higher chance of hitting the letters in the answer and thus receive more clues, reducing the number of tries.

Letter frequency,f_{Letter} is obtained by adding up the usage frequency of each letter in the words. The data on letter frequency is obtained from the Google Books Ngram Viewer[4], which includes a corpus of books and other publications from 1500 to 2008 with high reliability. For example,

the frequency of the letter "e" is 11.1607%, while the frequency of the letter "z" is only 0.0772%. f_{Letter} is defined as follows:

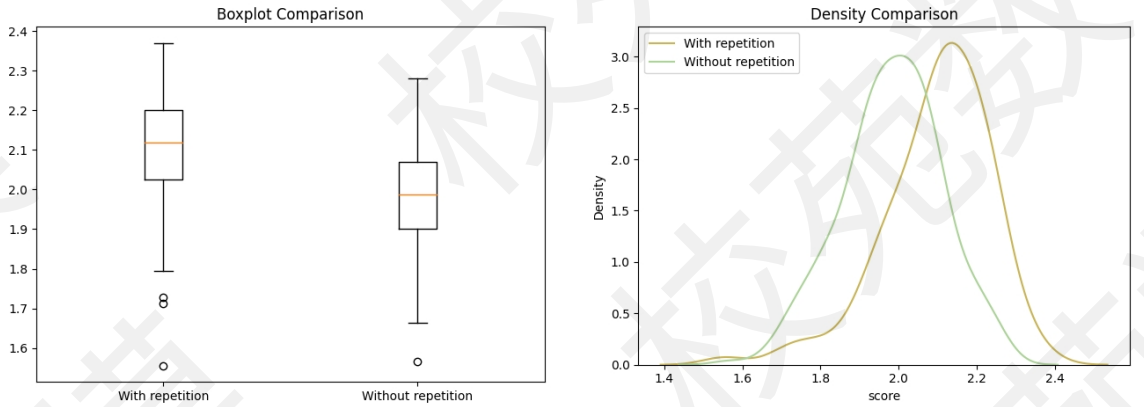
$$f_{Letter} = \sum_{1}^{5} f_{a,b,c...} \tag{4}$$

We conducted a regression analysis between f_{Letter} and $score$, similar to word frequency. The regression analysis results of letter frequency and $score$ are shown in Figure6(b) and Table3. The Pearson correlation coefficient is -0.4238, and the Spearman correlation coefficient is -0.4005, indicating **a certain linear correlation between letter frequency and** $score$**(even more than word frequency)**. Also, by observing the scatter plot in Figure6(b), it can be seen that there is no data distribution in the lower left and upper right corners of the image. This also suggests that situations where a word has a high frequency but is difficult to guess and thus results in a low $score$, as well as situations where a word has a low frequency but is easy to guess and thus results in a high $score$, are unlikely to occur. Therefore, it is reasonable to conclude that there is a certain linear correlation between letter frequency and $score$.

5.3 Box Plot Analysis

5.3.1 rep:Repetition of Letter

When there are repeated letters in the answer word, there may be a greater chance of hitting the letter when guessing and obtaining a hint, reducing the number of attempts, given a fixed word length. Analyzing the words from January 7, 2022 to December 31, 2022, we found that there are cases where a letter was repeated twice or three times(very few cases) in a word. Therefore, we divided the words into two categories: with or without repeated letters. We wanted to know if there was a significant difference in the $score$ of words with and without repeated letters. To this end, we used box plot analysis:



(a) Boxplot of repetition (b) Density of repetition

Figure 7: Boxplot analysis between Repetition of Letter and $score$

Median difference	Box length difference	Number difference of outliers
0.130037	0.00556	2

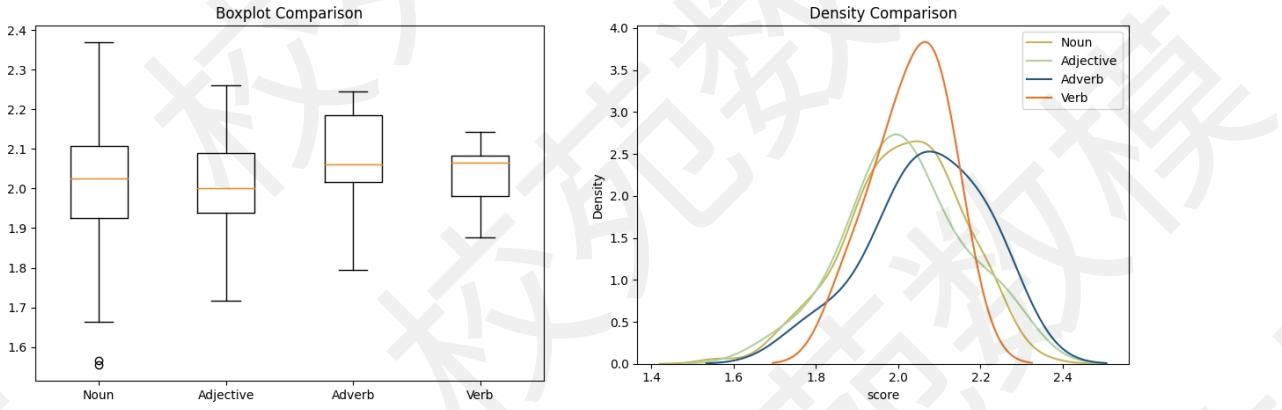
Table 6: Boxplot analysis between Repetition of Letter and $score$

From Figure7(a) and Table6, the median difference was 0.130037 and the box length difference was 0.00556. From Figure7(b), it can be seen that there is a significant difference in the distribution of the two categories of data. Therefore, we believe that **there is a certain difference in the** $score$ **situation between words with and without repeated letters.**

5.3.2 pos:Part of Speech

Part of speech(POS) is also an important attribute of a word. We are interested in whether there are differences in $score$ among words with different POS.

We used the popular Python natural language processing toolkit Natural Language Toolkit[5], which contains various text processing and language analysis tools, including Part-of-Speech Tagging. Using this tool, we performed Part-of-Speech Tagging on the words from January 7, 2022 to December 31, 2022. Words were mainly divided into four categories: Nouns, Adjectives, Adverbs, and Verbs. Subsequently, we conducted a box plot analysis of the "score" of these four types of words:



(a) Boxplot of POS (b) Density of POS

Figure 8: Boxplot analysis between POS and $score$

Median difference1	Median difference2	Median difference3
0.024991	-0.05973	-0.05972

Table 7: Boxplot analysis between POS and $score$

From Figure8(a) and Table7, the median difference between Noun and Adjective is 0.024991, the median difference between Noun and Adverb is -0.05973 and the median difference between Noun and Verb is -0.05972. From Figure8(b), it can be observed that the central tendency of the four categories of data is quite similar. Therefore, we conclude that **the $score$ of words does not show significant differences across different POS.**

6 GSRF Prediction Model

In Section 4, we processed the percentages of (1, 2, 3, 4, 5, 6, X) into a single parameter $score$. While a single parameter can capture some overall characteristics of these seven numbers, the specific meanings of these seven numbers cannot be expressed individually and important details and meanings may also be lost.

Unlike the time series prediction in Section 4, the distribution of the reported results theoretically should be determined by the attributes of the answer words on that day rather than the time series. Therefore, we chose the Grid-Search Random Forest (GSRF) algorithm to determine the best strategy for predicting the distribution of the reported results using the three attributes of the words themselves. With the best strategy obtained from the model trained on existing data, we predicted the distribution of the reported results of EERIE on March 1, 2023, and achieved good prediction performance.

6.1 Description of GSRF Algorithm

The GSRF algorithm consists of Random Forest and GridSearchCV.

GridSearchCV is a parameter optimization algorithm commonly used to fine-tune hyperparameters in machine learning models to optimize their performance. It iterates through a specified parameter grid and trains and evaluates the model for each possible parameter combination, ultimately outputting the best parameter set and corresponding model performance metric.

Random Forest is a machine learning algorithm that is an ensemble of multiple decision trees. The training process of Random Forest is based on multiple decision trees, with the algorithm randomly selecting a subset of features for training in each decision tree. During prediction, Random Forest aggregates the predictions from each decision tree by averaging or voting to obtain the final prediction.

Different from the ordinary Random Forest algorithm, the GSRF algorithm can train and predict the Random Forest model using the best hyperparameter combination, which significantly improves the model's performance and avoids issues such as overfitting or underfitting.The algorithmic flow of GSRF is shown Figure9.

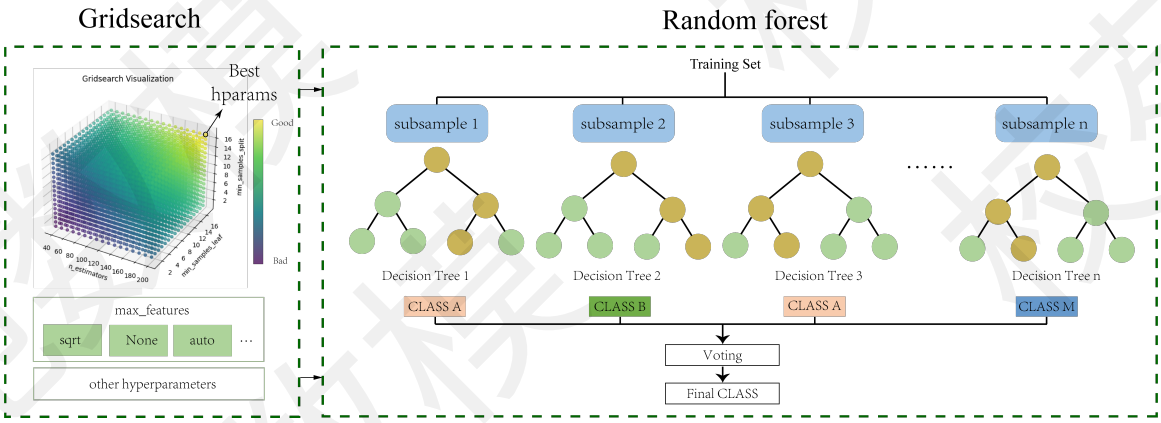


Figure 9: The algorithmic flow of GSRF

6.2 Prediction for EERIE on March 1,2023

In Section 4, we performed data analysis on the relationship between word attributes and the percentage of scores. We found that the word frequency f_{word}, letter frequency f_{letter}, and repetition of letters rep have some influence on the percentage of scores. Therefore, we used these three attributes of the word itself as input parameters for the Random Forest model to predict the distribution of the reported results:

$$(f_{word}, f_{letter}, rep) \xrightarrow{\text{GSRF}} (1, 2, 3, 4, 5, 6, X) \tag{5}$$

We take the word "eerie" that we want to predict for example, and its three input word attributes are shown in Table8. It should be noted that we set the rep value to 1.5 for words with repeated letters, and set the rep value to 1 for words without repeated letters. This is reasonable because GSRF normalizes the input data before calculation. Since rep has only two possible values, it will be processed as 0 and 1.

f_{word},	f_{letter},	ref
0.00023%	0.418799	1.5

Table 8: Word attributes of EERIE

Next, we trained the GSRF model using the word frequency, letter frequency, and repetition information of each word from January 7, 2022 to December 31, 2022, in order to learn the distribution of their reported results. To achieve this, we utilized the RandomForestRegressor algorithm from the scikit-learn (sklearn) machine learning library's ensemble module [6], as well as the GridSearchCV algorithm from the sklearn.model_selection module [7]. With the trained GSRF model, we made predictions on the distribution of reported results for the word "eerie" using its three word attributes. The results are presented in Table9 and Table10:

max depth,	max features,	min samples leaf	min samples split	n estimators
10	'sqrt'	2	10	200

Table 9: The best hyperparameter combination by GridSearchCV

1	2	3	4	5	6	X
1.07508%	6.51769%	23.30265%	30.29725%	23.13552%	13.27519%	2.40859%
MSE: 20.70641	MAE: 3.24388					

Table 10: Prediction for the word EERIE on March 1, 2023

6.3 Prediction evaluation analysis

When using machine learning algorithms for prediction, two commonly used evaluation metrics are the Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE is the average of the squared differences between the prediction and origin data, while MAE is the average of the absolute differences between the prediction and origin data. They are calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

where:

- n denotes the number of samples, y_i denotes the origin data, and $\hat{y}_i$ denotes the prediction data.

According to the data presented in Table10, **the GSRF Prediction Model had a very good prediction performance for the distribution of reported results for "EERIE" on March 1, 2023, with a MSE of 20.70641 and MAE of 3.24388.** This indicates that the model was able to accurately predict the reported results for the word based on its attributes, demonstrating the effectiveness of the GSRF algorithm for this task.

Considering that a word's attributes are not limited to the three we applied, other unexplored attributes could also influence the distribution of the reported results. Incorporating these additional attributes as inputs to the model could potentially improve its predictive accuracy. Moreover, the time when a word appears might also have an impact on the distribution, as was the case with the special instance of "Christmas Day".

- Regarding the issue of incomplete word attributes, our model has already attempted to comprehensively consider the properties that are beneficial to the prediction.
- As for whether time is a factor, it remains uncertain.

Based on the above analysis, we consider our prediction model to be a comprehensive and accurate one.

7 Difficulty Rate Classification Model by K-Means++

In this section, we define the difficulty rate δ of words using the distribution of reported results and calculated δ for "EERIE" and the words from January 7, 2022, to December 31, 2022. Then, we used the K-Means++ algorithm to perform clustering analysis on the difficulty rate of words and obtained a scientific classification of the difficulty rate. EERIE was classified in level 3. Finally, we randomly sampled words, manually labeled their difficulty levels, and used the K-Means++ model for clustering analysis. The results show that our classification model is relatively accurate.

7.1 δ: Difficulty Rate

The difficulty level of a word can be directly determined by the percentages of (1, 2, 3, 4, 5, 6, X). It is observed that when a word is particularly difficult or hard to guess, the percentage of larger tries (e.g. 5 or 6) increases.

Similar to the $score$ in Section 4, the difficulty rate δ is still composed of two parts. One part is the weighted average of the percentages of (1, 2, 3, 4, 5, 6), and the other part is X. However, unlike the X component in $score$ which is calculated as a percentage, the difference in magnitude between X and the other part is relatively larger when calculating δ. As X reflects the percentage of people who failed to answer the question on a given day, it should be given more weight when measuring the difficulty of a word. To balance the two parameters of δ, we use the Sigmoid function to normalize these two data and then assign weights using the Entropy Weighting Method to obtain the difficulty rate δ:

$$\delta = \sigma\omega_1 * \sum_{i=1}^{6} i * p_i + \sigma\omega_2 * p_X \tag{8}$$

where:

- p_i denotes the percentage of i tries(try) and $i \in \{1, 2, 3, 4, 5, 6, X\}$
- σ denotes the sigmoid function
- ω_1 and ω_2 denote the weights of two parts of $score$ respectively. And we set ω_1 as 0.5 and ω_2 as 0.5 by Entropy Weighting Method.

7.2 K-Means++ Clustering Analysis

K-Means clustering algorithm is a common unsupervised machine learning algorithm used to divide data into several categories. It pre-specifies the initial number of clusters and the initial

cluster centers, and divides the sample set into different clusters according to the size of the distance between the samples. The Euclidean distance is used as a measure of the similarity between data objects, with similarity being inversely proportional to the distance between data objects. The larger the similarity, the smaller the distance. Based on the similarity between the data objects and the cluster centers, the position of the cluster centers is continuously updated, and the sum of squared errors (SSE) of the clusters is continuously reduced. When the SSE no longer changes or the objective function converges, the clustering ends and the final result is obtained.

The Euclidean distance formula between data objects and cluster centers in space is:

$$d\left(X, C_i\right) = \sqrt{\sum_{j=1}^{m} \left(X_j - C_{ij}\right)^2} \tag{9}$$

where:

- X denotes data objects
- C_i denotes the i_th cluster center
- m denotes dimensionality of data objects
- X_j and C_{ij} denote the j-th attribute values of X and Ci, respectively

The formula for calculating the SSE of the entire dataset is:

$$SSE = \sum_{i=1}^{k} \sum_{X \in C_i} \left|d\left(X, C_i\right)\right|^2 \tag{10}$$

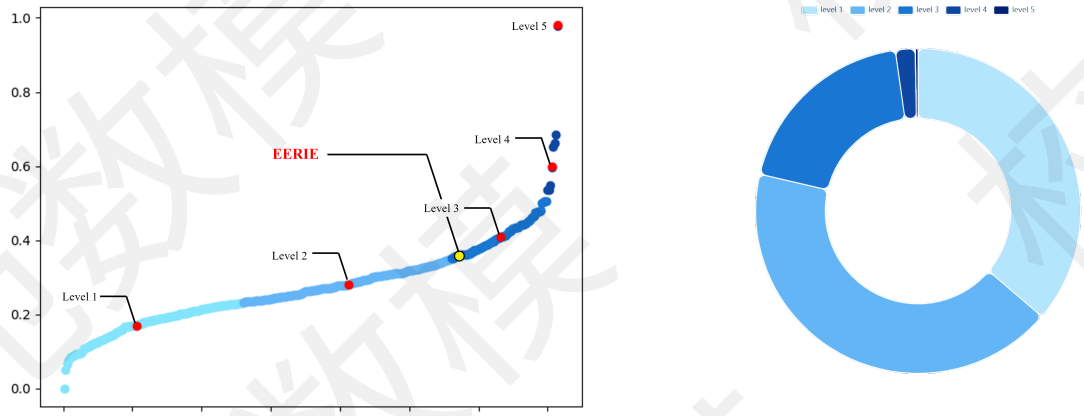
where:

- k denotes the number of clusters

In traditional K-Means algorithm, the initialization of cluster centers is usually randomly selected from k sample points. However, this random selection method is prone to local optima, leading to poor clustering results. The probability selection process introduced by K-Means++ algorithm can make the cluster centers more dispersed, making it easier to find the global optimal solution, and thereby improving the quality of clustering results.

7.3 Difficulty classification for EERIE

Using the K-Means algorithm in the scikit-learn library of Python[8] with the "k-means++" initialization parameter set, we can perform a clustering analysis on the calculated difficulty rate δ of each word. The analysis divides the difficulty coefficients of the words into five levels, ranging from level 1 to level 5, with a higher level indicating a greater difficulty of the word. The calculated difficulty rate δ of the word "ERRIE" is 0.35916, which corresponds to level 3 in the difficulty hierarchy.The results are shown in Figure10 and Table11 as follows:



(a) (b)

Figure 10: Difficulty rate classification

Level	Number	Percentage	Valuable Percentage
1	131	36.2%	36.2%
2	152	42.4%	42.4%
3	69	19.1%	19.1%
4	7	2%	2%
5	1	0.3%	0.3%
Sum	6.51769%	23.30265%	30.29725%

Table 11: Results of clustering analysis

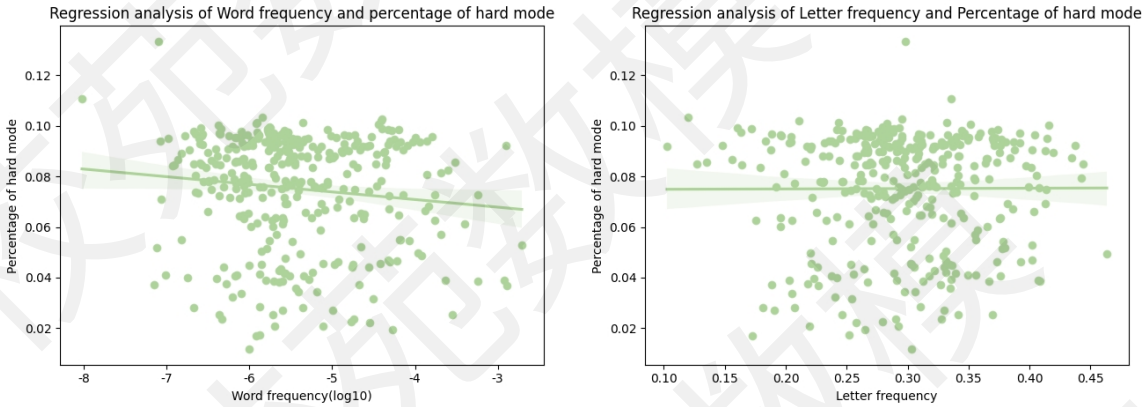
7.4 Accuracy Discussion of Classification Model

Finally, we randomly sampled 30 words, manually labeled their difficulty levels, and used the K-Means++ model for clustering analysis. Comparing the manually labeled difficulty levels of sample words with the difficulty levels obtained from the clustering analysis, 28 out of 30 data points matched, resulting in a **93.33%** match rate. This indicates that the difficulty level classification obtained by the model is consistent with our subjective judgment of word difficulty. Therefore, we believe that **our classification model is relatively accurate.** The data for the word samples are presented in the AppendixA.

8 Interesting Features of Data

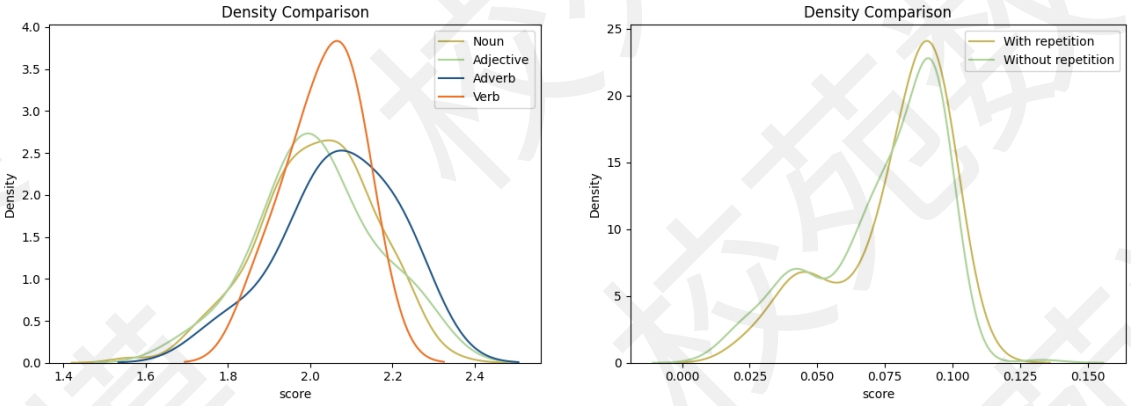
8.1 Feature 1:Relationship of Word Attributes and Hard Mode Percentage

We are interested in whether the properties of words are related to the proportion of users who choose the Hard Mode daily. Since the properties of words are to some extent related to difficulty, if the words are too difficult one day and increase users' frustration, users may not choose the Hard Mode the next day. Therefore, we conducted data analysis on the four attributes of words and the daily percentage of Hard Mode, including regression analysis and boxplot analysis. The results show that the percentage of Hard Mode is not linearly related to word frequency and letter frequency. There is also no significant difference in the percentage of Hard Mode among words with different letter repetitions and different parts of speech. The results are shown as follows:

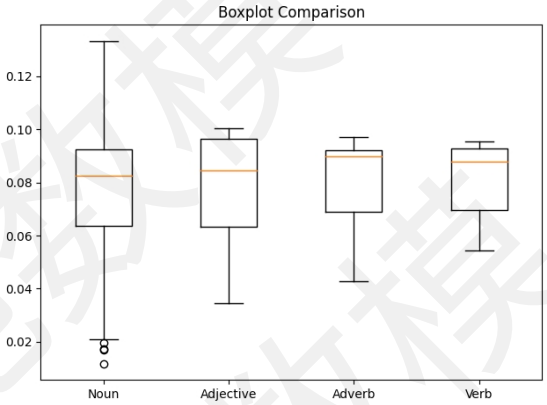


(a) Word frequency (b) Letter frequency

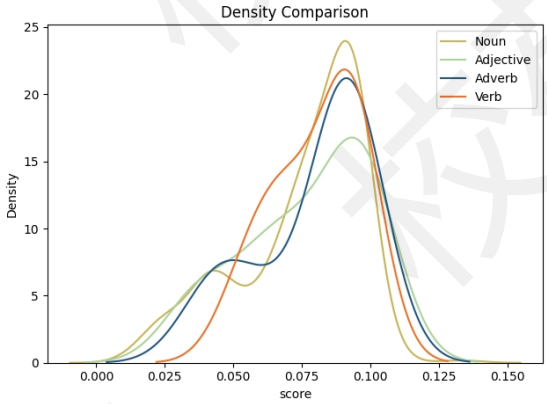
Figure 11: Regression analysis between word attributes and Hard Mode peercentage



(a) Boxplot of repetition (b) Density of repetition



(c) Boxplot of POS

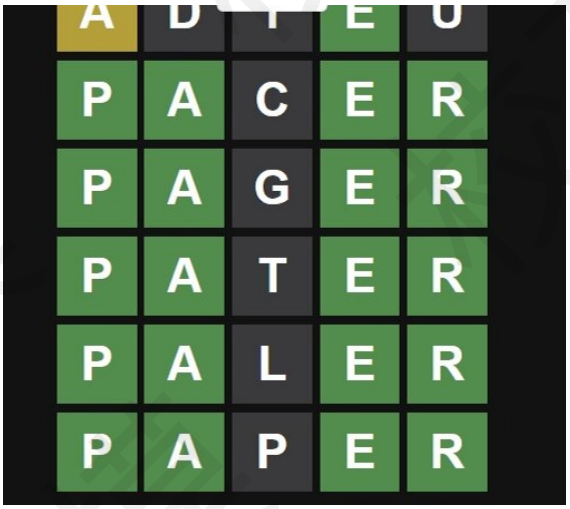


(d) Density of POS

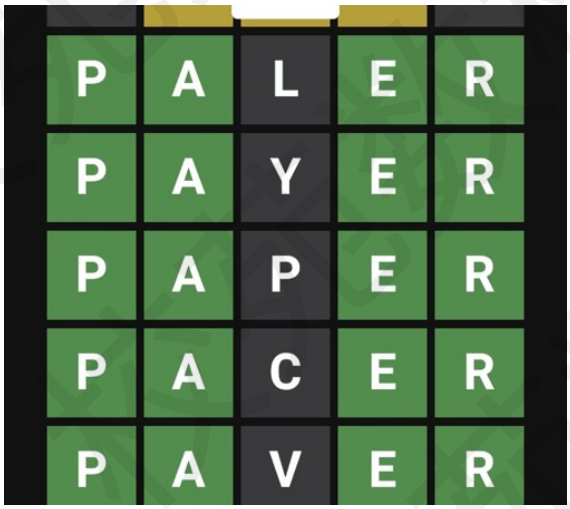
Figure 12: Boxplot analysis between POS and Hard Mode peercentage

8.2 Feature 2:Why "PARER" Has The Most "Hellish" Level of Difficulty

In Section 8, we found that the difficulty date of the word "parer" is as high as 0.98, far exceeding the second-place word "mummy" with a score of 0.69. Looking back at the original data, we are surprised to discover that on the day of "parer", the "X" percentage was as high as 48%, indicating that 48% of players were unable to guess the answer. While browsing the user comments on Wordle Stats, some comments provided a possible explanation.(Figure13) It suggested that when the word to be guessed has many similar words (i.e. with similar letter composition or positions), and those similar words have a higher frequency of everyday use than the solution word, it becomes difficult to guess the correct word.



(a) User comment1[9]



(b) User comment2[10]

Figure 13: User comments

9 Model Sensitivity Analysis

We conducted a sensitivity analysis on the input parameters of the GSRF Prediction Model to test its sensitivity to changes in the input parameters in predicting the distribution of the reported results. The specific method is to add Gaussian noise to the model's input parameters. However, for the word repetition level, there are only two types of data, and the noise added to rep will be directly eliminated by the model when normalizing the input data. Therefore, we only conducted sensitivity analysis on word frequency and letter frequency. Gaussian noise is defined as follows:

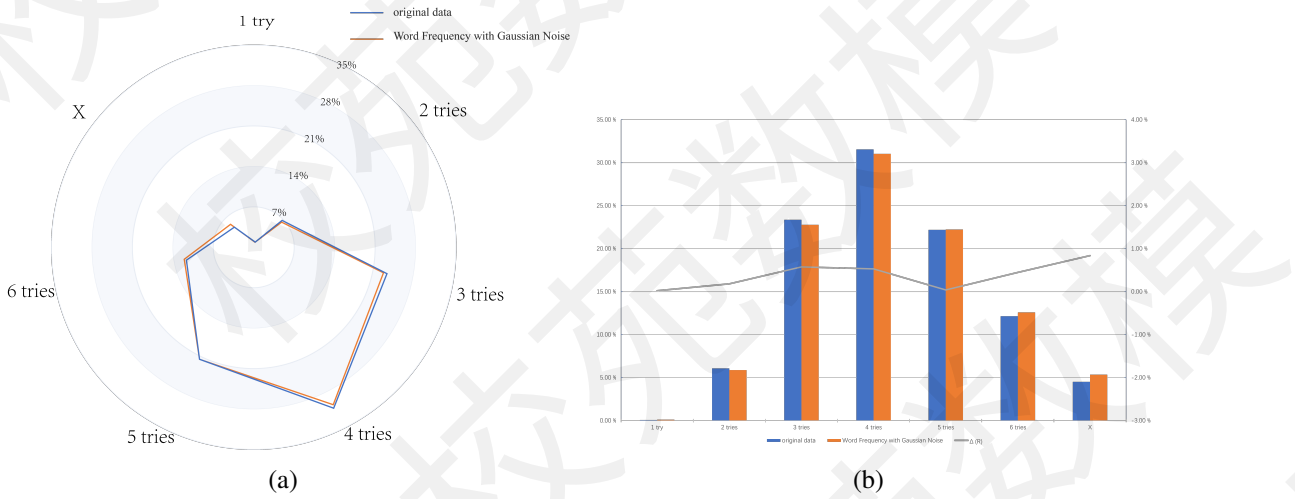
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{11}$$

where:

- x the amplitude of a random signal, μ denotes the mean, σ denotes the standard deviations

9.1 Sensitivity Analysis for f_{word} in GSRF Prediction Model

After adding 1 Gaussian noise to the word frequency, the results are shown in Figure14:



(a) (b)

Figure 14: Sensitivity of word frequency

The visual results from sensitivity analysis indicate that the GSRF Prediction Model exhibits low sensitivity to changes in word frequency, indicating a high level of stability.

9.2 Sensitivity Analysis for f_{letter} in GSRF Prediction Model

After adding 1 Gaussian noise to the letter frequency, the results are shown in Figure15:

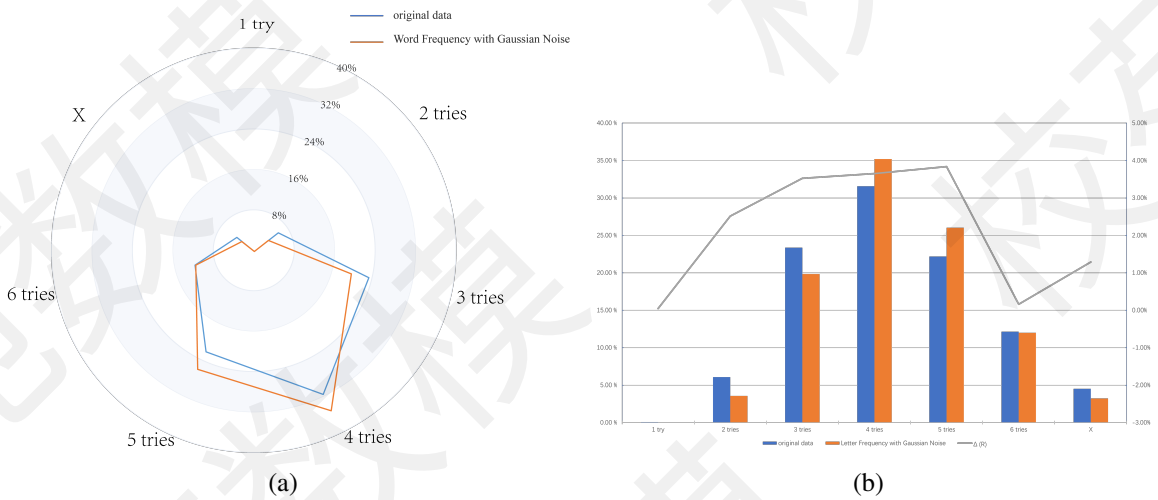


Figure 15: Sensitivity of letter frequency

The visual results from sensitivity analysis indicate that the GSRF Prediction Model exhibits relatively low sensitivity to changes in letter frequency, indicating a high level of stability.

10 Model Evaluation and Further Discussion

10.1 Strengths

1. **GRU**: The GRU algorithm in Section 4 has several advantages when used for prediction tasks. GRU can effectively handle sequential data with variable-length inputs, making it well-suited for time series prediction. And GRU typically requires fewer parameters to train compared to other recurrent neural networks, such as LSTM, making it computationally more efficient and easier to train. Therefore, the relative error rate of our prediction result is only 2.1569.

2. **GSRF**: Different from the ordinary Random Forest algorithm, the GSRF algorithm in Section 6 can train and predict the Random Forest model using the best hyperparameter combination, which significantly improves the model's performance and avoids issues such as overfitting or underfitting.

3. **K-Means++**:In traditional K-Means algorithm, is prone to local optima, leading to poor clustering results. The probability selection process introduced by K-Means++ algorithm Section 7 can make the cluster centers more dispersed, making it easier to find the global optimal solution, and thereby improving the quality of clustering results.

4. **K-Means++**Our analysis in the word attribute analysis in Section 5 and data feature mining in Section 8 is relatively comprehensive and constructive.

10.2 Weaknesses And Further Discussion

1. There are more attributes of words, such as vowels and consonants.

2. Time factor may be incorporated into the prediction of the distribution of the reported results.

3. Our model uses a large number of machine learning algorithms. More advanced methods and data training techniques can be selected. For example, the K-Means algorithm has variations such as Mini-Batch K-Means and Genetic K-Means algorithms. The most suitable algorithm from each variation can be selected for analyzing the current data.

References

[1] https://twitter.com/WordleStats/status/1481687496241164291

[2] https://www.marca.com/tecnologia/2022/02/14/620a2ee522601da7288b4599.html

[3] https://pytorch.org/docs/stable/generated/torch.nn.GRU.html

[4] https://books.google.com/ngrams/

[5] http://www.nltk.org/

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[9] https://twitter.com/WordleStats/status/1571182427007221764

[10] https://twitter.com/s_harmony_/status/1570624760530477057

11 Letter

Dear Puzzle Editor of the New York Times.

I hope this message finds you well.I am writing to share the results of our analyses of Wordle data.

Firstly, we established a GRU Prediction Model to predict the number of reported results on March 1, 2023. The model uses the effective Gated Recurrent Unit (GRU) algorithm. Therefore, predictions made by the training set to the testing have the relative error rate of 2.1569%, and the relative RESE of 6.4957%, indicating a good accuracy of the model predictions. The predicted interval for the number of reported results on March 1, 2023 is 20367 ± 2.01569

Secondly, we conducted a data analysis on the attributes of words and score defined by the percentage of scores played in Hard Mode to explore their relationship. Then, we defined four attributes of the words: word frequency, sum of letter frequencies, repetition patterns of letters (2/3 or none), and pos: main part of speech. For f_word and f_letter, we performed regression analysis with the variable "score". The Pearson correlation coefficient between f_word and score is -0.3165, and f_letter and score -0.4005. rep and pos can be used to categorize the words. The box plot results showed that the Median difference of the box plot for rep was 0.13004, while pos was only 0.05973. Therefore, we believe that f_word, f_letter, and rep can affect the percentage of scores, while pos can not.

Thirdly, we have developed GSRF Prediction Model to predict the percentages of 1 to X for EERIE on March 1, 2023. The Grid-Search Random Forest (GSRF) algorithm is an improved random forest algorithm by using the best combination of hyperparameters. We selected the three parameters, f_word, f_letter, and rep as inputs for the model. The model's training results show a MSE of 20.70641 and a MAE of 3.24388, indicating good predictive performance. (Table 10) The predicted results for EERIE are (1,7,23,30,23,13,3). In addition, we conducted sensitivity analysis by adding Gaussian noise to f_word and f_letter separately, and the results showed that the model has low sensitivity and is thus highly stable.

Fourthly, Difficulty Rate Classification Model using the K-Means++ are conducted. We first defined the difficulty date δ of each word first. The difficulty rate of EERIE is 0.35916 by predicted distribution. Then, we used the improved clustering analysis algorithm, K-Means++, to analyze the δ of each word and obtained five levels of difficulty (Table11). EERIE was classified into the third level. Finally, we compared the model's classification with the manul difficulty ratings for a subset of sampled words and achieved a match rate of 93.33%, confirming the accuracy of the model.

Finally, we found that the four word attributes we analyzed do not affect the percentage of users selecting the hard mode. We also investigated why "parer" is the only word in the fifth difficulty level, with a high difficulty coefficient of 0.98. It turns out that the difficulty of a word may be linked to the word's similarity to other words and the frequency of use of these similar words.

You can read the full text of our model if you would like to learn more detailed information.

Sincerely,

Team 2300348

A Samples

Word	Manual	Model	Word	Manual	Model	Word	Manual	Model
atoll	3	3	foyer	5	5	twang	4	4
train	1	1	flock	4	4	bloke	4	4
madam	5	5	hairy	3	3	primo	5	5
peach	2	2	other	3	3	depth	2	2
admit	3	3	knoll	5	5	brine	3	3
trait	4	4	buggy	5	5	class	4	5
recap	3	3	favor	5	5	natal	5	5
carry	3	5	happy	1	1	atone	2	2
found	5	5	aphid	4	4	thyme	2	2
molar	4	4	bough	5	5	wacky	5	5

B Difficluty Rate of Partial Words

Word	value	group	Word	value	group	Word	value	group
stein	0.139130856	1	treat	0.0894603	1	cloth	0.173900245	1
aloud	0.108252609	1	dream	0.097017869	1	poise	0.139130856	1
today	0.196054096	1	panic	0.151548495	1	glory	0.183128816	1
stair	0.064625023	1	doubt	0.12793931	1	caulk	0.310999245	2
grate	0.174870226	1	solar	0.173476571	1	infer	0.323441505	2
happy	0.203454774	1	choke	0.178493799	1	movie	0.317155672	2
metal	0.17185657	1	tepid	0.117904854	1	donor	0.23639953	2
tiara	0.168933189	1	begin	0.22372847	1	bluff	0.308941723	2
hoard	0.181350828	1	thyme	0.205706341	1	piney	0.349255486	2
avert	0.226157463	1	robin	0.208597166	1	beady	0.285658995	2
Word	**value**	**group**	**Word**	**value**	**group**	**Word**	**value**	**group**
cynic	0.246710978	2	showy	0.327805763	2	eject	0.360750519	3
lofty	0.313559582	2	cargo	0.289792582	2	gully	0.475589662	3
unfit	0.258769444	2	blown	0.253373134	2	sever	0.435166278	3
flock	0.263786672	2	glyph	0.233683305	2	vivid	0.407453933	3
carry	0.293076879	2	nasty	0.236784568	2	comma	0.373391992	3
condo	0.339162962	2	creak	0.303318231	2	wedge	0.376850239	3
sweet	0.304997854	2	shard	0.292839653	2	motto	0.355507495	3
soggy	0.308474029	2	elope	0.293112002	2	droll	0.362206003	3
flood	0.271312514	2	howdy	0.38417086	3	mummy	0.685643564	4
story	0.23657413	2	gamer	0.368070525	3	parer	0.978421619	5