Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2307946

Words Behind *Wordle*: Puzzle Game Analysis Using Machine Learning and Time Series Theory

Summary

Wordle is a popular puzzle currently offered daily by New York Times. Players try to solve the puzzle by guessing a five-letter word in six tries or less, receiving feedback with every guess. Making full use of relative information can effectively help editors to improve operational performance.

Firstly, to explain the variation and predict the future value, a **time series model** based on the number of reported results is introduced. After determining the optimal groups of orders, **ARIMA**(0,1,1) model is used to forecast the prediction interval of the number of reported results on March 1, 2023, which is **[10139.23, 30808.07](80% confidence)**. To find out if any attributes of the word affect the hard mode percentage, a words attributes system and a **LightGBM** model are introduced. The results show that there are some lag attributes that have some but less effect than lag Hard Mode percentage itself.

Secondly, to predict the associated percentages of (1, 2, 3, 4, 5, 6, X), two models are established based on **GBDT** and **MMoE**. The results show that the MMoE model significantly outperforms the GBDT model, with MSE of 145. Then, we attemptd to improve the model by using **data augmentation** and **feature engineering** methods. The former leads to a large amount of noise, which fails to achieve the expected effect, and the latter slightly improves the model performance. The prediction of the final model for the word EERIE is **(0.649, 7.579, 26.298, 32.614, 20.930, 9.63, 2.298)**.

Thirdly, **K-means** model is introduced to cluster the samples into **4 groups** with the distribution of attempt times as the features by difficulty. In order to determine which features of the words are associated with the classifications, we used the classification as the output feature and all the attributes of the words as the input feature to establish a **LightGBM** model for training. The accuracy of the test set reaches **70%**. The importance of the output features is sorted. Finally, the model is used to predict the category of the EERIE word, and the prediction result is Group 2.

Finally, some interesting features of the dataset are found in dataset. The characteristics of large frequency words, the shape of distribution of attempt number and the correlation of the word features are discussed.

In addition, we evaluated the advantages and disadvantages of the model and proposed some suggestions, and carried out a sensitivity analysis of the model to the commission rate, thereby proved the reliability and stability of the model.

Keywords: Wordle ; ARIMA; LightGBM; MMoE; data augmentation; feature engineering; K-means; sensitivity analysis

Contents

1 Introduction **3**

1.1 Problem Background 3

1.2 Restatement of the Problem 4

2 General Assumptions and Model Overview **4**

2.1 Assumptions 4

2.2 Model Overview 4

3 Model Preparation **5**

3.1 Notations 5

3.2 Data Preprocessing 5

4 Model I: Time-Series Forecasting Model **7**

4.1 The concept of Time Series 7

4.2 Stationarity of time series 7

4.3 Model Building 8

4.4 Forecasting Results 9

5 Extraction and Analyse Attributes of the Word **9**

5.1 Extraction of Word Attributes 9

5.2 Word Attributes Overview 10

6 Model II: Explaining Hard Mode Percentage Using LightGBM **11**

6.1 Introduction of LightGBM 11

6.2 Data Description and Preprocessing 12

6.3 Model Results and Evaluation 13

7 Model III: Multiple Input - Multiple Output Regression Model **15**

7.1 White Noise Verification 15

7.2 Model Introduction 16

7.3 Model Refinement 17
7.3.1 Data Augmentation 17
7.4 Feature Engineering 17
7.5 Model Results and Evaluation 17

8 Model IV: LightGBM Classifier based on K-means Clustering Model 18
8.1 Concept of K-means Clustering 18
8.2 Clustering Model Building 19
8.3 Evaluation of Clustering Result 20
8.4 Identification of Important Attributes 20
8.5 Classfication Result and Evaluation 20

9 Other Interesting Features of the Data Set 21

10 Sensitivity Analysis 22
10.1 Sensitivity Analysis for Question 1 22
10.2 Sensitivity Analysis for Question 3 23

11 Strengths and Weaknesses 23
11.1 Strengths 23
11.2 Weaknesses 24

12 A Memorandum to the New York Times Puzzle Editor 24

1 Introduction

1.1 Problem Background

At the beginning of 2022, a simple but novel game gained great popularity on Twitter. This is the web word game ***Wordle*** written by Josh Wardle and published by the New York Times Company.

The game was fairly unknown at the very beginning, but after Wardle creatively added a function that allows players to copy the results into a grid of colored square emojis to share, it immediately attained public attention. As of mid-January 2022, there have been more than 2 million people have played and more than 1.2 million *Wordle* results have been posted on Twitter.

In *Wordle*, players have to guess a word with five English letters within six chances in one day. After each attempt, the players may get three types of feedback: green if the letter is in the correct position; yellow if the answer contains the letter while the letter is in the wrong place; gray if the answer does not have the letter at all. The gameplay is similar to games like Mastermind, but *Wordle* will clearly indicate which letters were guessed correctly. [1]

Apart from that,*Wordle* has another game mode. On the basis of the above regular rules, the "Hard Mode" requires once a player has found a correct letter in a word, those letters must be used in subsequent guesses.

In fact, there is a profound mathematical mechanism behind the seemingly simple game. We can't help wondering what mechanism affects the efficiency of players to make correct guesses, and what laws exist behind the constantly changing number of reported results on Twitter. On what basis do players choose Hard Mode?

We expect to solve the above problems through mathematical modeling to effectively predict the future operation of the game and provide Puzzle Editor of the New York Times with business suggestions.

Figure 1: NY Times Wordle

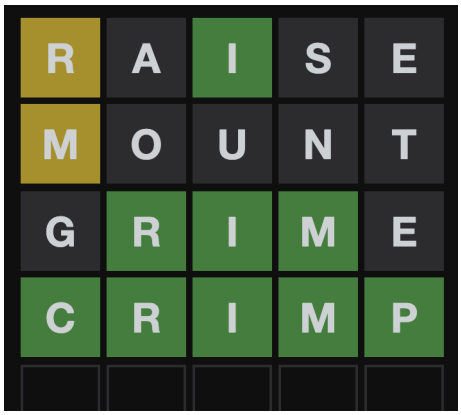

Figure 2: Example of solution

1.2 Restatement of the Problem

As we have a data set containing the date, contest number, word of the day, the number of people reporting scores that day, the number of players on Hard Mode, and the distribution of the reported results. We need to build mathematical models to solve the following problems for New York Times Company:

Question 1:

1. Develop a model which explains the variation of the reported results number, then make a prediction of this number for Match 1,2023 using the developed model.
2. Find out the possible attributes of the given word which may influence the percentage of scores reported that were played in Hard Mode, and give the inherent mechanism of the influence.

Question 2: Develop a model that forecasts the distribution of the reported results of a given word on a day to come.Then discuss the uncertainties and the accuracy of the prediction model.

Question 3: Adopt a mathematical model to classify solution words by difficulty, identify the attributes of a given word that link with each classification as well as evaluate the accuracy of the classification.

Question 4: Discuss and find other features within the data set.

2 General Assumptions and Model Overview

2.1 Assumptions

To simplify the problem, we make the following basic assumptions, each of which is properly justified.

1. The number of reported results on Twitter can effectively represent the total number of players on the day, and the percentage of scores reported that were played in Hard Mode is the same as that of all players.
2. The distribution of the reported results recorded in the dataset is completely accurate.
3. There are correlations and differences between the associated percentages of 1 try, 2 tries,$\cdots$, X.
4. The word difficulty is proportional to the average number of tries to guess the result.

2.2 Model Overview

In summary, the whole modeling process can be shown as follows:

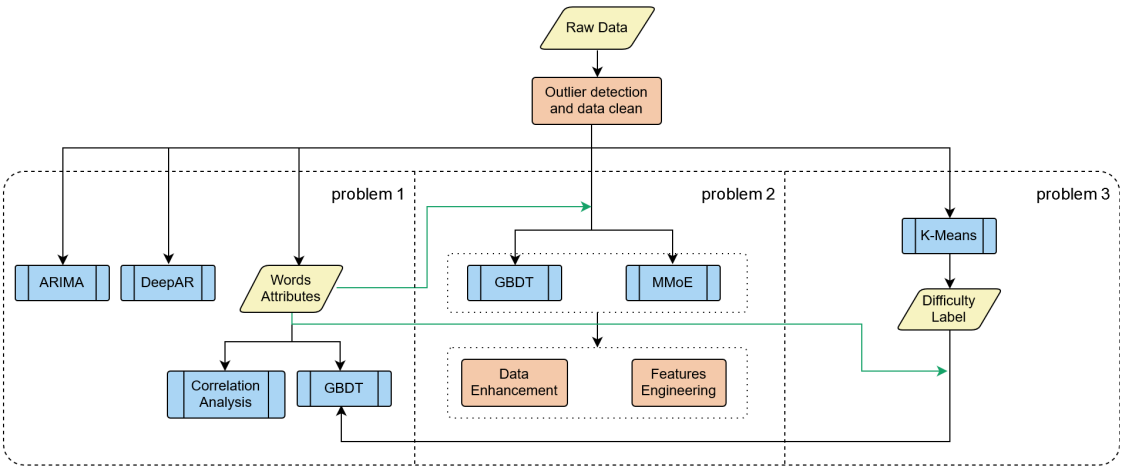

Figure 3: Model Overview

3 Model Preparation

3.1 Notations

Important notations used in this paper are listed in Table 1,

Symbols	Definitions
y_t^T	The number of reported results at day t
$y_t^{T'}$	The first-order difference sequence of y_t^T
$y_t^{T''}$	The first-order difference sequence of y_t^T (after May 16, 2022)
y_t^H	Number of reported results in Hard Mode at day t
pct_t	The percentage of scores reported that were played in Hard Mode at day t
$distribute_i$	The percentage that guessed the word in i tries

Table 1: Notations

3.2 Data Preprocessing

The data we use includes the data files given as **Problem C Data Wordle.xlsx**

This file gives nearly all the information we need for solving the problem. But before using it, the data needs to be preprocessed.

Firstly, we need to exclude outliers in the data set, that is, remove the points where the *percentage of scores reported that were played in Hard Mode* (hereinafter referred to as *Hard Mode Percentage*) is too far away from the other data. When making a scatter plot with a smooth line of the *Reported results number*, *Number in hard mode*, and *Hard Mode Percentage* concerning time, we can easily find the existence of outliers, which are marked with red dots in the following figures.

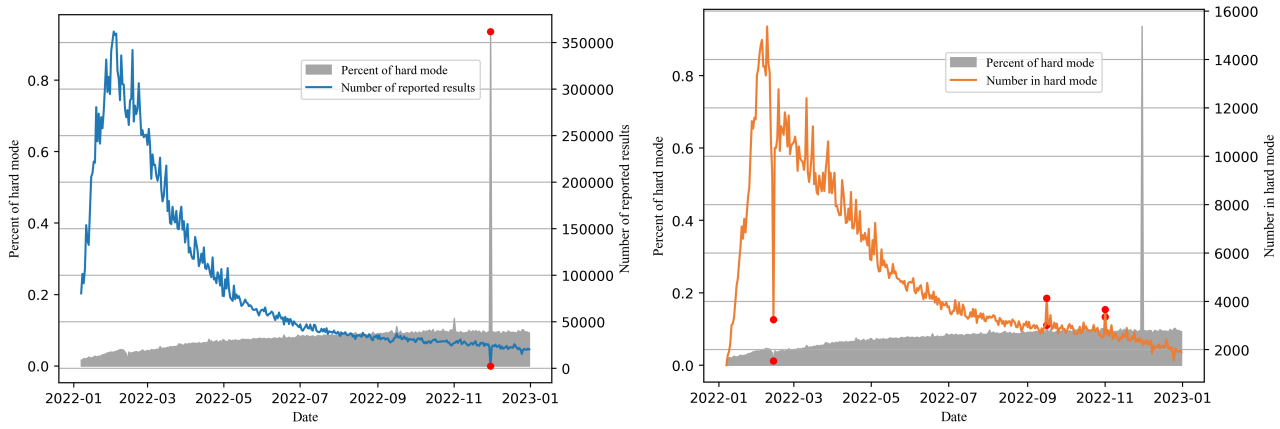

Figure 4: Scatterplots of the three Time Series in the Dataset

Using the method of rolling boundary statistics, select the *Hard Mode Percentage* for 30 days before and 30 days after a certain day as the sample group, then construct the rolling interval according to the following equation:

$$Means \pm 3 \times (Standard\ Deviation) \tag{1}$$

If the *Hard Mode Percentage* for that day falls outside this interval, it is an outlier. According to this method, we tested the data set and obtained 4 outlier points. After eliminating the outliers, we used the linear interpolation method to complete the original data to obtain a more stationary data set.

In addition, we also found that there are spelling errors in the data set. For example, some given words only contain four letters, which contradicts the game's rule of 5 letters. We removed the sample data with misspelled words to ensure the validity of the data.

All data samples to be processed and processing results are shown in the table below.

Table 2: Data Preprocessing Checklist(Outliers and Misspelling)

Date	Error location	Adjustment	Adjusted number
2022/11/30	Number of reported results	interpolation	10.37%
2022/11/26	Word	Deletion	N/A
2022/11/01	Number in hard mode	interpolation	9.48%
2022/10/05	Word	Deletion	N/A
2022/09/16	Number in hard mode	interpolation	8.15%
2022/04/29	Word	Deletion	N/A
2022/02/13	Number in hard mode	interpolation	3.54%

4 Model I: Time-Series Forecasting Model

4.1 The concept of Time Series

A **time series** is a sequence of numbers listed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. **Time series analysis** includes methods for analyzing time series data to extract meaningful statistics and other characteristics of the data. **Time series forecasting** is the use of a model to predict future values based on previously observed values.

4.2 Stationarity of time series

If we want to make a time series forecast, we must first ensure its **stationarity**. A stationary time series should have no trend and seasonality, that is to say, its mean and variance are constant.

However, from Figure 4, we can easily derive that the time series of the number of reported results has an obvious trend and isn't stationary. To ensure our judgment, we then adopted the **ADF (Augmented Dickey-Fuller) test**. The null hypothesis of this test is that the time series is not stationary. [2] The resulting significant test statistic is -1.9608 while the p-value is 0.5934, which cannot reject the null hypothesis. This confirms our judgment.

In view of the non-stationarity of the data, we performed first-order difference processing on the data. We set the time series of the number of reported results as y_t^T, and constructed its first-order difference sequence $y_t^{T'} = y_t^T - y_{t-1}^T$. Made a scatter plot with a smooth line, as shown in the figure below:

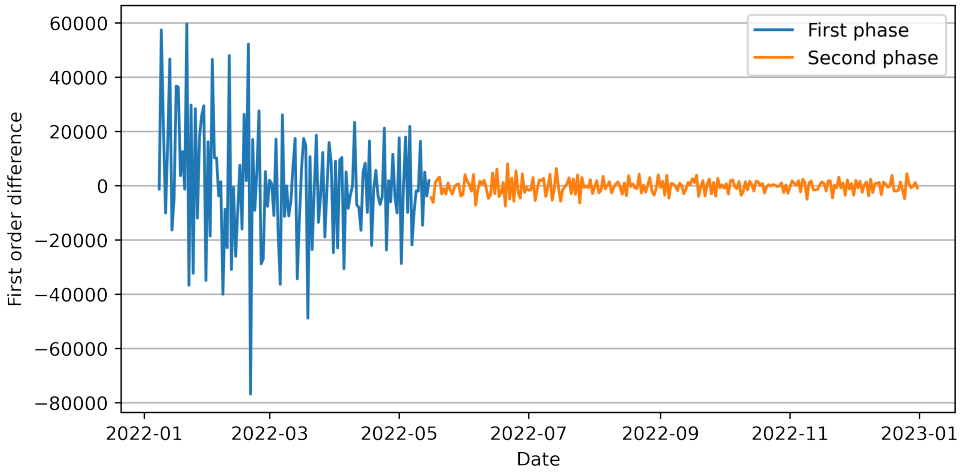

Figure 5: Time Series of y_t^T After Difference Process

It can be seen from the figure that the time series of the number of reported results being processed has a large variance before May 16, 2022, with a small variance after that, so the data after May 16 is selected as the new time series to predict the future value,

recorded as $y_t^{T''}$.

After that, we carried out the ADF test for $y_t^{T''}$ again, this time p-value is 0.01 which rejects the null hypothesis, indicating that the time series after the first-order difference is stationary.

4.3 Model Building

After ensuring the stationarity of the time series, we can use the **ARIMA (Autoregressive Integrated Moving Average Model**) model for time series forecasting. [3] The general expression of the model has the following form, ARIMA(p,d,q):

$$(1-\sum_{i=1}^{p}\alpha_i L^i)(1-L)^d y_t = \alpha_0 + (1+\sum_{i=1}^{q}\beta_i L^i)\epsilon_t \tag{2}$$

Its essence is to combine difference(d), autoregressive model(AR(p)), and moving average model(MA(q)). p is the order of the autoregressive model, d is the order of difference, and q is the order of the moving average model. By drawing the **autocorrelation (ACF) diagram** and **partial autocorrelation (PACF) diagram** of the time series $y_t^{T''}$, it is found that ACF is first-order truncated and PACF is tailed, so we can choose ARIMA(0,1,1)as the target model.

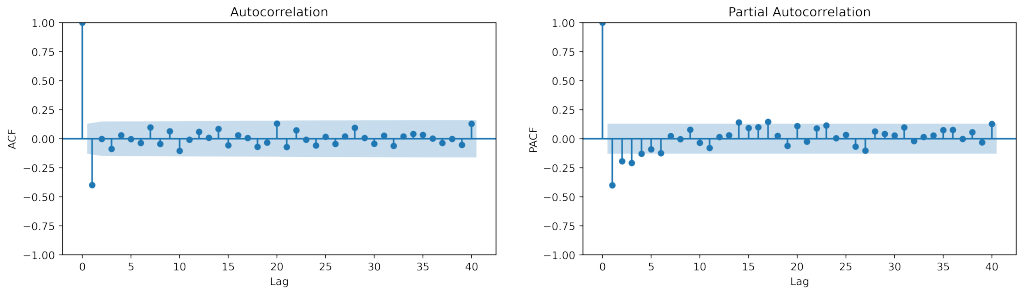

Figure 6: Autocorrelation and Partial Autocorrelation

At the same time, the best ARIMA parameters can also be obtained by using the *auto.arima* command in R language, resulting ARIMA(1,1,1) and ARIMA(3,1,1).

In order to determine the best of all obtained ARIMA parameters, we use the **information criterion (Akaike Information Criterion)** to aid judgment. The **AIC values** of the three optional models are 4200.897/4202.825/4205.466, and ARIMA(0,1,1) has the smallest AIC value, so it is selected as the final model.

To determine the validity of the ARIMA(0,1,1) model, we further made a white noise residual test. Test result shows that the p-values of the LB statistics are all above the threshold of 0.05, so the model passed the test and we can use ARIMA(0,1,1) to explain the variation of the number of reported results.

4.4 Forecasting Results

Using the ARIMA(0,1,1) model and substituting the date, the forecast results are shown as the following graph:

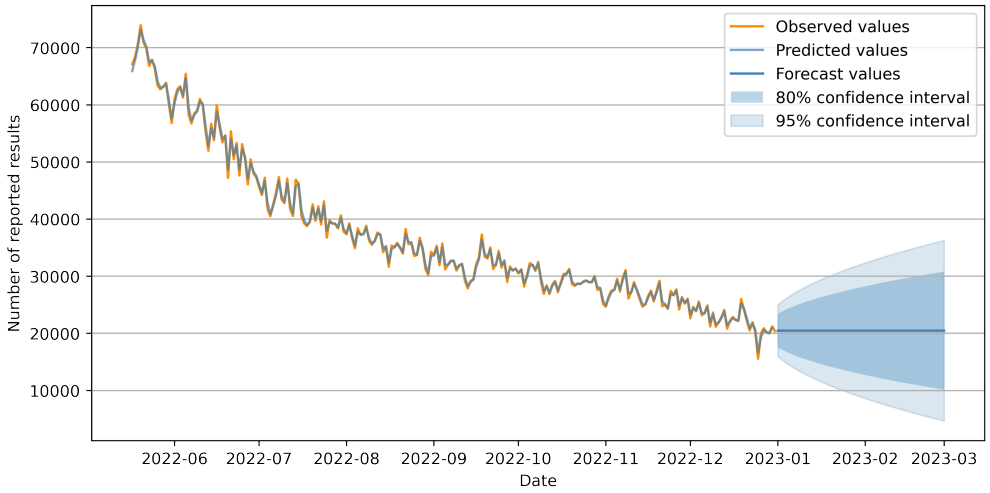

Figure 7: Forecasting Results

For March 31, 2023, the prediction interval of the number of reported results with a confidence level of 80% is [10139.23, 30808.07], the prediction interval with a confidence level of 95% is [4668.516, 36278.78], and the predicted expected value is 20473.65.

5 Extraction and Analyse Attributes of the Word

5.1 Extraction of Word Attributes

The difficulty of word-quiz games like *Wordle* is closely related to the attributes of the given word. In view of the characteristics of the *Wordle* game rules, the difficulty of it mainly depends on the memorability of the given word and the difficulty for the player to recall this word. Many educators and linguists have studied the factors that affect students' vocabulary memorization. For example, Laufer, B. (1990) [4] listed many word attributes that influence vocabulary learning performance. For *Wordle*, in order to get the answer, in addition to learning and memorizing words, it is also essential to infer the result based on known information, which difficulty is affected by many attributes of the word as well.

When selecting the attributes of the word, referring to existing research such as Jakub Jagoda & Tomasz Boiński (2018) [5], we can get some common word attributes that affect quiz difficulty, such as word frequency, part of speech, number of vowels, number of repeated letters and the word's emotional tendency. In addition, the research of psycholinguists also provides us with many innovative indicators, such as the number of orthographic neighbors that a word has, the number of syllables in the main pronunciation, and so on.

5.2 Word Attributes Overview

Our attributes data mainly originate from **The English Lexicon Project** [6], a linguistics project jointly participated by several universities, which aims to provide standardized behavioral and descriptive datasets for 40,481 words and 40,481 non-words. The data can be obtained by visiting *elexicon.wustl.edu*. In addition, some Python packages and algorithms can also provide us with data about word attributes.

Other research results we have used include the SUBTLEX language library, Bysbaert et al. (2014) research on Concreteness Rating [7], Hoffman et al. (2013) research on Semantic Diversity [8] and DeDeyne, et al. (2018) on Association Frequency Research [9].

It is worth noting that word attributes **in different time periods** may have different effects on game difficulty. For example, actually, players do not know what exactly is today's given word before they start playing, so we can naturally guess that the word attributes of today's word have limited influence on the difficulty of today's game. But at the same time, the word attributes of the past may significantly affect the difficulty of the game that players assume. If the word of the past few days is difficult to infer, then the players of the day may be less inclined to choose Hard Mode, and even lose the game because of a lack of confidence. Research on the **hysteresis effect of past word attributes** will become one of the focuses of the modeling below.

The following table lists all the word features extracted in our research, their meanings, and data sources:

Table 3: Word Attributes Overview

Attribute Notation	Meaning of the Attribute	Data Sources
word_freq	How often the word is used in everyday life	Database from Kaggle
num_vowel	Number of vowels in the word	Counted using python
num_repeat	Number of repeated letters	Counted using python
part of speech	Such as nouns, pronouns, etc.	Python package NLTK
sentiment	Sentimental attributes of words[1]	Package vaderSentiment
Ortho_N	The number of orthographic neighbors that a word has	English Lexicon Project
Phono_N	The number of phonological neighbors[2] that a word has	English Lexicon Project
OG_N	The number of phonographic neighbors[3] that a word has	English Lexicon Project
BG_Sum	The sum of the bigram count for a particular word	English Lexicon Project
NPhon	The number of phonemes in the main[4] pronunciation	English Lexicon Project
SUBTLCD	The SUBTLEX contextual diversity[5]	SUBTLEX
OLD	The mean of the closest 20 LD neighbors for the orthograph	English Lexicon Project
PLD	The mean of the closest 20 LD neighbors for the phonology	English Lexicon Project
Concreteness_Rating	The mean of the Concreteness Ratings	Bysbaert et al. (2013)
Semantic_Diversity	The Semantic Diversity	Hoffman et al.(2013)
Assoc_Freq_R123	Number of Times Word is one of first three associates	DeDeyne, et al.(2018)
NMorph	The number of Morphemes	English Lexicon Project

[1] Such as positive, negative, etc.
[2] This statistic excludes homophones
[3] This statistic excludes homophones
[4] The diphthongs /aI/, /aU/, /OI/, and the affricates /tS/ and /dZ/, each count as single phonemes
[5] % of films containing the word

6 Model II: Explaining Hard Mode Percentage Using LightGBM

6.1 Introduction of LightGBM

LightGBM is a novel **GBDT (Gradient Boosted Decision Tree)** algorithm proposed by *Ke* in 2017 *(Ke et al., 2017)* [10]. GBDT has the functional characteristics of Gradient Boosting and Decision Tree, and has the advantages of good training effect and not easy to overfit. Its advantages include fast training speed, high accuracy, low memory usage,

and support for parallel computing. It can be used to solve the problems encountered by GBDT in massive data processing.

One of the characteristics of LightGBM is the use of a **Histogram-based decision tree algorithm**, which first discretizes the continuous eigenvalues into k values, and then generates a histogram with a width of k. When traversing samples, it uses the discretized value as an index. After a traversal, the histogram accumulates the required statistics and then traverses to find the optimal segmentation point through the discrete value of the histogram.

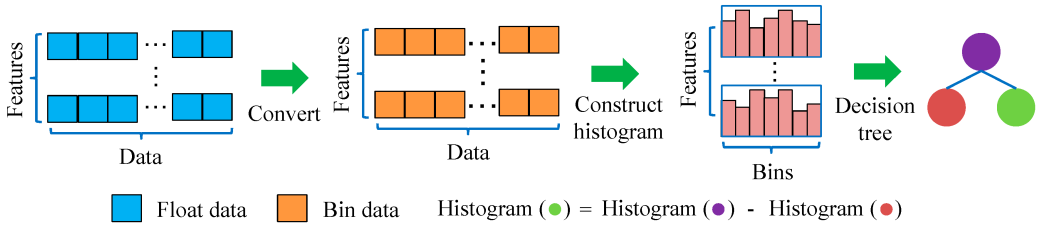

Figure 8: Histogram-Based Decision Tree Algorithm

Another feature of LightGBM is to adopt a more efficient leaf growth strategy, namely the **leaf growth strategy with depth limitation (Leaf-wise)**. Before splitting, this strategy first traverses all the leaves in the tree, and then finds the leaf with the largest splitting gain to split again, and repeats this operation. Experiments have proved that under the same number of splits, Leaf-wise can get higher accuracy, and a maximum depth limit to prevent over-fitting has been added to Leaf-wise.

The leaf-wise leaf growth strategy is shown in the figure below, where the white and black dots represent the leaves with the maximum and non-maximum split gains, respectively:

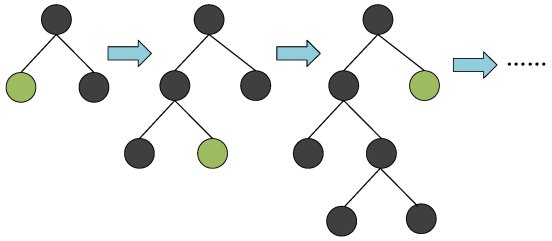

Figure 9: Schematic Diagram of Leaf-Wise Tree Growth

6.2 Data Description and Preprocessing

The main purpose of this model is to identify whether the attributes of words affect the **Hard Mode percentage**, as well as explore the mechanism of influence. Therefore, the **label** of the model is the Hard Mode percentage of the current period, denoted as pct_t.

Considering the hysteresis effect of the previous word attributes and label itself on the label of the current period, our **feature** sequence includes the lag terms from one period*(1 day)* to five period *(5 days* of all the attributes indicators involved in Table 3, as well as the current value of Hard Mode percentage and one period to five period lag terms of it. If all the attributes of the given word in period t are denoted as X, then the model input contains:

$$pct_{t-1}, pct_{t-2}, \cdots, pct_{t-5}; X_t, X_{t-1}, \cdots, X_{t-5} \tag{3}$$

In order to ensure the validity and reliability of the model, the data set needs to be preprocessed. A very important step is to normalize the value of each attribute and map the data to [0, 1]. The method is listed below:

$$v_s = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{4}$$

Among them, v_s is the standardized value, v is the original value, v_max, and v_min represent the maximum and minimum values of the attribute respectively. The normalized data are collectively referred to as $Data_{input}$.

6.3 Model Results and Evaluation

We utilized the above data, and input the feature sequence to train the model with the ratio of $training\ set : test\ set = 4 : 1$. Then the model would assign different weights to different features by making a large amount of historical data correlation calculation, and output the estimated (or predicted) value of Hard Mode Percentage using these weights.

For the purpose of verifying the reliability of the LightGBM model, we used MSE, RMSE, and SMAPE of the prediction results as the measurement indicators of the model accuracy. After parameter optimization, the calculation results of each indicator within the training set and validation set are as follows:

Table 4: Model Accuracy Test Results

	MSE	**RMSE**	**SMAPE**
Training set	3.623×10^{-6}	0.0019	0.0253
validation set	3.382×10^{-5}	0.0058	0.0690

As all the indicators are apparently small, we can easily assert that the model result fits the real data very well. This conclusion can also be drawn from the scatter plot of the predicted value and the observed value below.

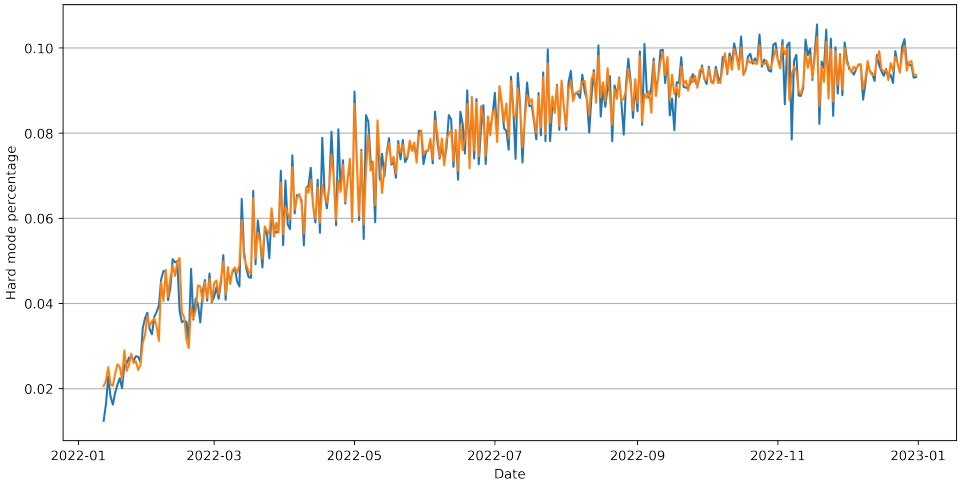

Figure 10: Comparison of Predicted Results of LightGBM and Observed Value

Due to the reliability of the model, we can use the weight of each feature to judge whether the attributes of the word affect the Hard Mode Percentage. It can be concluded from the weight map of the feature that time lag items of the label itself are the most influential, while the time lag items of the word attributes are the next. Therefore, the Hard Mode Percentage of the current day's game is related to the given words' attributes of the previous periods and has almost nothing to do with the word attributes today, despite the ratio being mainly affected by its own lag terms.

As mentioned in subsection 5.2, if the words in the previous periods are difficult, the player will be more inclined not to choose Hard Mode for the current period. What's more, since the player **does not know** the difficulty of the current period of words before choosing, the word attributes of the current period should not affect the player's choice which is consistent with the predictions of our model.

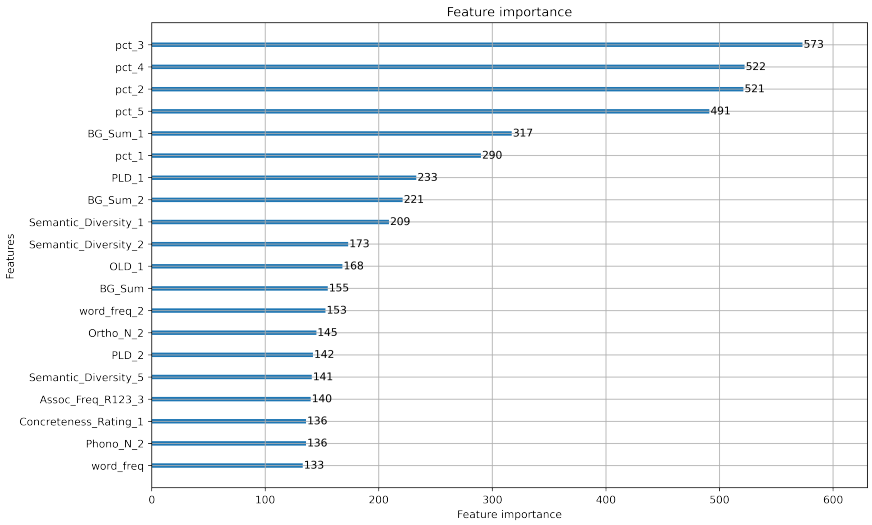

Figure 11: Relative Importance of Features

7 Model III: Multiple Input - Multiple Output Regression Model

Multiple input - multiple output regression model is mainly used to solve the regression problem involving predicting two or more values when given an input example. Generally, there are two types of ideas - one is to train multiple regressors based on the machine learning model, and the other is to modify the number of output layers based on the deep learning model. Experience has shown that due to the large amount of neural network parameters, when the data amount is small, using feature engineering and ensemble learning method may have better prediction results than directly using deep learning models.

7.1 White Noise Verification

Firstly, we ran a **white noise test** [1]on each of the associated percentages of guess attempts for a future date[2]. If the data passed the white noise test, it means that it is not affected by the time trend. Therefore, we can use the cross-sectional data of word attributes and distribution for subsequent modeling and predictions.

Take date as the horizontal axis and the percentage of 1 to 7 or more tries as the vertical axis to make a smooth-lined scatter plot. It can be seen from the figure that the trend of 2-6tries is relatively stationary, while the percentage of 1 and 7tries have some less stationary values. The reason may be that there are extremely easy and extremely difficult questions on certain days.

At the same time, it can be seen from the ACF diagram that the autocorrelation coefficients of 2-7 tries are relatively small, while the autocorrelation coefficient of 1 try is slightly larger.

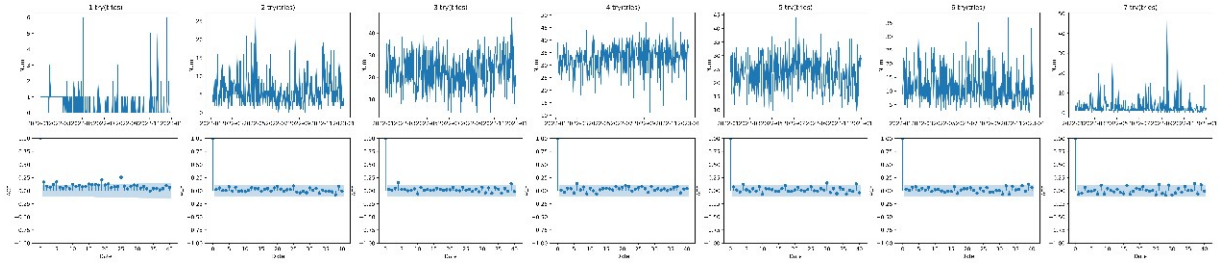

Figure 12: Scatted Plot and ACF Diagram For the Distribution of Reported Results

We conducted the **Box-Pierce White Noise Test** on these 7 percentages, and found out their p-values were all higher than 0.05, which passed the test.

[1]white noise means a purely random sequence, with no relationship between yesterday's and today's values

[2](1 try, 2 tries, 3 tries,$\cdots$, 6 tries, and X)

7.2 Model Introduction

Therefore, our analysis tried to compare the effects of the following two models on this data set, then adopted data augmentation, feature engineering and other methods to improve the model effect.

- Multiple output regression model based on GBDT algorithm:

The idea of this model is to train multiple regressors based on the GBDT algorithm, respectively fitting the seven dependent variables of 1 try to 7 or more tries (X), with no correlation between the regressors.

- Multiple output regression model based on MMoE:

Multi-task learning refers to the method of training multiple objective functions at the same time. Its main advantage is that it can improve the learning efficiency and quality of each task. In addition, it can effectively overcome the shortcomings of large task noise, insufficient training samples, high data dimensionality, and sparse data sets.

The framework of multi-task learning widely adopts the shared-bottom structure, which means the hidden layer at the bottom is shared between different tasks. This structure can essentially reduce the risk of overfitting, but the effect may be affected by task differences and data distribution.

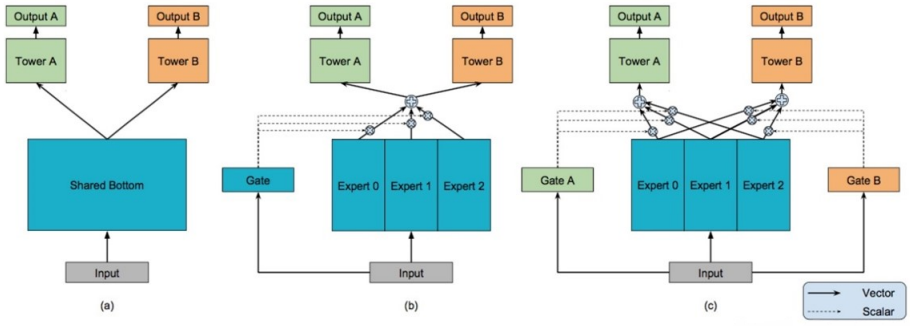

Figure 1: (a) Shared-Bottom model. (b) One-gate MoE model. (c) Multi-gate MoE model.

Figure 13: MMoE's Advantage Shown by Comparison

Therefore, a content recommendation team at Google proposed a **Multi-gate Mixture-of-Experts (MMoE)** multi-task learning structure. The improvement of MmoE is that, compared with the basic shared-bottom structure, it captures the differences in tasks without significantly increasing the requirements of model parameters. Compared with all tasks sharing a gating network (One-gate MoE model, as shown above), each task in MMoE uses separate gating networks. The gating networks of each task realize the selective utilization of experts through different final output weights. Gating networks of different tasks can learn different modes of combining experts, so the model takes into

account the relevance and differences of captured tasks, which is very suitable for predicting the number of player attempts in this problem.

We input word attributes and ran the two models described in Section 7.2 to predict the distribution of the reported results, but the prediction results still need to be optimized.

7.3 Model Refinement

7.3.1 Data Augmentation

For multi-input and multi-output problems, neural networks are generally used for modeling processing, but neural networks or deep learning require a large amount of data to be used for training. As the total sample size in this question is 359, it is difficult to get sufficient data for training the neural network. However, it may be possible to expand the data set through data augmentation.

CTGAN is a collection of **Deep Learning** based synthetic data generators for single tabular data, which is able to learn from observed data and generate synthetic data with high fidelity [11]. It should be noted that the data generator cannot limit the relationship between variables, the sum of percentage from 1 try to 7 tries in the generated sample may be quite different from 100%. So we used CTGAN to generate new samples, filtered out the samples that meet the conditions and merge them into the original data set. **Finally, our sample size was about 4000.**

After obtaining the enhanced samples, the above two models were retrained. However, we found that the training results did not improve significantly, possibly because the original sample size was too small.

7.4 Feature Engineering

Generally speaking, data and features determine the upper limit of machine learning, and models and algorithms only approach this upper limit. In addition to optimizing the model from the perspective of data, it is also possible to extract features from the original data to the maximum extent, that is, **feature engineering**.

OpenFE is a new framework for automated feature generation for tabular data. [12]Using OpenFE as a tool, we perform feature engineering on the unaugmented and augmented datasets separately to generate datasets containing new features. After this process, we retrained the above two models using the new data set. Luckily, the results have been significantly improved.

7.5 Model Results and Evaluation

Our prediction for the word *EERIE* on March 1, 2023 is as follows:

The predictive effectiveness of the two types of models shows obvious differences. **The multiple output regression model based on GBDT** has a large variance in total **MSE** between the training set and the verification set. This indicates that the machine learning model has poor model generalization performance when the data set is small. The MSE

of the training set and validation set when using **the multiple output regression model based on MMoE** are relatively close. What's more, they are apparently lower than the former model, showing an obvious advantage.

Unexpectedly, data augmentation led to a significant decline in the performance of both types of models, indicating that the new data brings greater noise to model training; feature engineering has no effect on the former, however brings a small improvement in the performance of the latter. Therefore, we finally chose **the multiple output regression model based on MMoE** plus **features engineering** mode. After parameter optimization, the prediction results of the word EERIE on March 1, 2023 are as follows:

Percent in	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 or more tries (X)	Total
Number	0.649	7.579	26.298	32.614	20.930	9.63	2.298	99.998

Table 5: Prediction Result of Distribution

8 Model IV: LightGBM Classifier based on K-means Clustering Model

The purpose of this model is to classify solution words by difficulty. When making classification, We first use the **K-means clustering model** to divide the word samples in the data set into several groups according to the average number of puzzle-solving attempts. After that, use the **LightGBM** algorithm to classify them into corresponding groups by using the attributes of the given words.

8.1 Concept of K-means Clustering

Distance-based clustering often uses measurement methods, such as K-means, K-medoids, etc. Among them, the current most popular heuristic method is the K-means algorithm. So we used the **percentage that guessed the word in one to seven tries** given in the data set as clustering basis and ran the K-means clustering model.

Given a set of data points and the number of clusters required, the K-means algorithm moves the data points into each cluster domain iteratively according to a specific distance function, the general implementation steps are as follows:

1. Given a sample word data set with a size of n, let the number of iterations be R, and randomly select k words as the initial cluster centers according to the specified number of clusters k, labeled as $C_j(r)$, j= 1, 2, 3,$\cdots$,k; r=1, 2, 3,$\cdots$, R.

2. Calculate the similarity distance $D(X_i, C_j(r))$ between each data object in the sample and the initial cluster center; among them, i=1, 2, 3,..., n, then form a cluster W_j, if it satisfies the formula (5)

$$\sum_{i=1}^{n} \left| D\left(X_{i+1}, C_j(r)\right) - D\left(X_i, C_j(r)\right) \right|^2 < \varepsilon \tag{5}$$

Then$X_i \in W_j$,X_i is denoted as w, where ϵ is any given positive number.

3. Calculate k new cluster centers, the calculation formula is as follows:

$$C_j(r+1) = \frac{1}{n}\sum_{i=1}^{nj} X_i^{(j)} \tag{6}$$

The formula for calculating the value of the clustering criterion function is as follows:

$$E(r+1) = \sum_{i=1}^{k}\sum_{w\in W_j} |w - C_j(r+1)|^2 \tag{7}$$

4. To judge whether the clustering is reasonable, the discriminant formula is listed below:

$$|E(r+1) - E(r)| < \varepsilon \tag{8}$$

If the result is reasonable, the iteration terminates; while if it is not, return to steps 2 and 3.

8.2 Clustering Model Building

In order to determine a reasonable **number of clusters k**, we first made k=1, 2, 3...,10 as the experimental number of clusters, then used Python to run the K-means clustering model. Finally, calculate the sum of squares due to error **(SSE)** of the results with different k values. With the number of clusters k as the horizontal axis and SSE as the vertical axis, a scatter diagram with a trend line is made as follows:

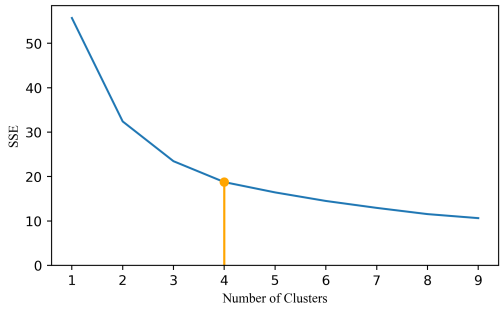

Figure 14: Determine the Number of Clusters

We had to make a balance between the minimal value of SSE and the smallest possible amount of the number of groups. It can be derived from the figure that when the k value is less than 4, as the number of clusters increases, the SSE decreases rapidly. On the contrary, when k is greater than 4, the downward trend is not obvious. So the most suitable k value Which achieves the aforementioned balance is 4.

8.3 Evaluation of Clustering Result

In order to verify the effectiveness of clustering, we calculated the mean difficulty value for each group, which is represented by the average number of attempts a player has to make to guess the word. The more tries a player made, the more difficult the given word is. The calculation formula is as follows:

$$Average\ Number\ of\ Tries = \sum_{i=1}^{7} distribute_i \times i \tag{9}$$

The number of words in each group, the average number of tries to guess the word as well as Skewnesses in each group are shown in the table below.

Table 6: Clustering Result Evaluation

Group	0	1	2	3
No.of Words	34	127	125	57
Average Difficulty	3.574	3.951	4.326	4.800
Skew	0.320	0.246	0.0918	-0.181

The data in the table shows that the average number of tries and skewness of each group are significantly different, indicating that our Clustering is relatively effective.

8.4 Identification of Important Attributes

We have divided words into groups according to difficulty in the analysis above. Next, we would further make an in-depth analysis, hoping to link the classification of word difficulty with its own attributes, and solve the problem of "why is this word more difficult"?

We again used the LightGBM model, taking the 4 clustered groups obtained by the clustering algorithm as labels, and used the word attributes listed in subsection 5.2 as input features to train the lightGBM model. The training result(Figure 15) reflects the importance of each attribute of the word in terms of affecting the word's difficulty level. What's more, we can utilise the model to classify a given word into clustered groups above.

Therefore, most of the attributes of the word are related to classification, while the most relevant attributes are BG_sum, num_repeat, $Semantic_diversity$, PLD, and $Phono_N$. The impact of word attributes on difficulty also mainly comes from these five indicators.

8.5 Classfication Result and Evaluation

Referring to the result of 7.4,the key attributes of the word "EERIE" are:

We input these attributes into the LightGBM classification model, and the output showed that this word belongs to the second category, which is the second lowest level of difficulty.

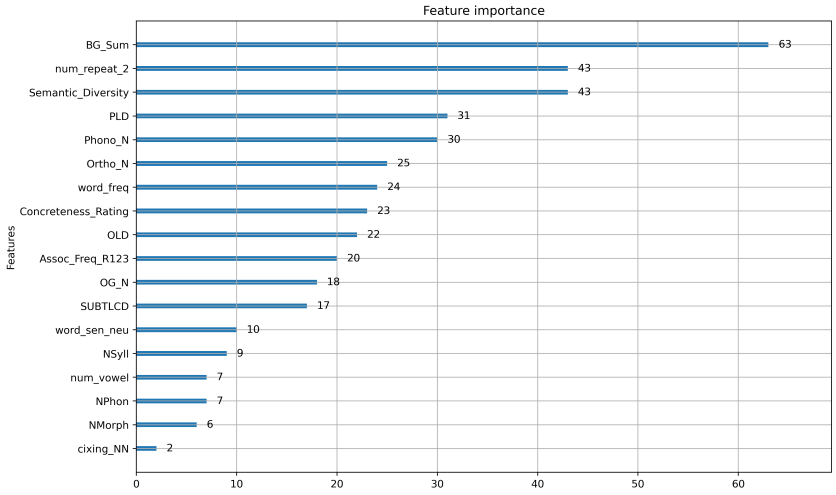

Figure 15: Importance of Features in Classfication

Table 7: Part of key attributes of EERIE

Word	Phono_N	BG_Sum	PLD	Semantic_Diversity	num_repeat
EERIE	8	11159	1.4	1.487	4

We used a variety of indicators to measure the accuracy of the LightGBM classification model. The results are shown in the table below. By analyzing the results, we could assert that our model made a quite accurate classification regarding the words in the data set.

Table 8: Mesuring Accuracy of Classification

	Accuracy	Precision	Recall	F1-score
Validation set	71.13%	71.85%	66.54%	69.09%
Traning set	98.28%	98.82%	98.79%	98.28%

9 Other Interesting Features of the Data Set

1. It can be seen from Figure 4 that the number of users of the game rose rapidly in a short period of time, while then declining rapidly, reflecting the short-term popularity of the game. In addition, the percentage of people who chose the Hard Mode gradually increased over time, indicating that users gradually mastered the game.

2. Figure 16 is a word cloud diagram drawn according to the frequency of use. Most of the high-frequency words are pronouns and auxiliary verbs, followed by nouns.

3. Observing the distribution of the number of user attempts in Figure 17, the kernel density function of less difficult words is flat and skewed to the right; while the kernel density function of more difficult words has a higher peak value, mostly concentrated at 4 times.

Figure 16: The Word Cloud

4. It can be seen from Figure 18 that, among the various word attributes, the correlations between $Ortho_N$, $Phono_N$, and word frequency are strong.

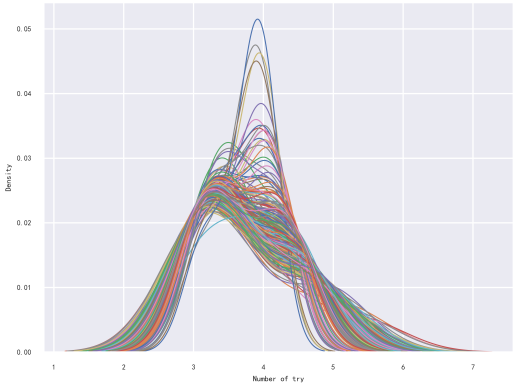

Figure 17: Kernel Density Function Plot

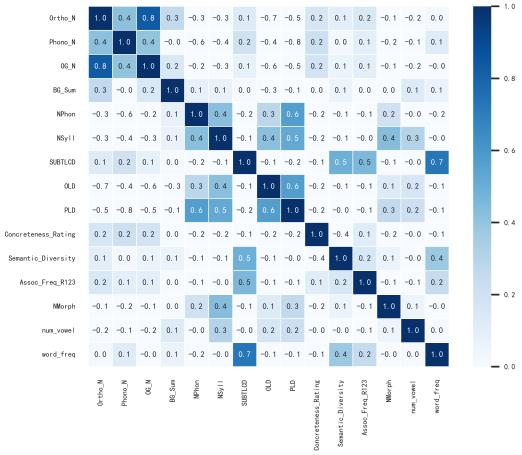

Figure 18: Correlation of Attributes

10 Sensitivity Analysis

10.1 Sensitivity Analysis for Question 1

By constructing the LightGBM model, we solved the problem of what factors (contemporaneous or lagged) influence the current Hard Mode percentage. In order to test the sensitivity of the model, we applied random perturbations to the top five influential attributes derived in 8.4. Then observed whether there was a significant change in the relative importance of attributes. The results are shown in the table below:

Within it, the disturbance range p means that for the original value x, a random number within the range of $[(1-p)x, (1+p)x]$ is used to reassign x. No matter which number is drawn, it will not cause a great impact on the importance of attributes regarding predicting Hard Mode Percentage.

Table 9: Sensitivity Analysis for Model of Question 1

Attribute Name	Disturbance range p
BG_Sum_1	When p is less than 0.15, the ranking of word attributes remains unchanged
BG_Sum_2	Ranking of word attributes with the second-largest to fifth-largest influence keeps steady when p is less than 0.1
Semantic_Diversity_1	
Semantic_Diversity_2	
OLD_1	

It can be derived from the table that, to a certain extent, small-scale data disturbance will not affect the importance ranking of word attributes, which means that the model passed the sensitivity test.

10.2 Sensitivity Analysis for Question 3

In order to analyze the sensitivity of the classification model in this paper, we repeatedly deleted x randomly selected samples from the total samples. We subsequently analyzed whether the classification results had changed significantly, which is shown in the table below:

Table 10: Sensitivity Analysis for Model of Question 3

heightx	The number of times among 100 repeated tests when the classification result didn't change
1	100
2	100
3	98
4	95
5	90

We can derive from the table that after 100 times of random deletion of 1-5 samples from the overall data set, the number of times among 100 repeated tests when the classification result didn't changes still reaches more than 90. That means, to a certain extent, small-scale sample changes will not have a significant impact on the classification result. The model for question 3 passed the sensitivity test as well.

11 Strengths and Weaknesses

11.1 Strengths

1. We have fully exploited a variety of word attributes that adequately reflected the information content of words as much as possible.

2. The model fully considered the data of the current period and lagged terms.

3. We tried a variety of machine learning and deep learning models as well as made an in-depth comparison of their performances.

11.2 Weaknesses

1. We didn't try to adopt a time series prediction model based on deep learning.
2. We didn't fully overcome the problem of poor prediction accuracy caused by a small amount of data.

12 A Memorandum to the New York Times Puzzle Editor

Dear Editor:

We are a professional business analysis team from MCM. It's a great honor to be invited to answer your operation questions.

We are very delighted to see the game *Wordle* achieved great success in 2022. But to ensure long-term steady operation, you also need to have a deep understanding of the game's operating mechanism. At your request, we analyzed the data mined from Twitter and we are here to answer your questions with confidence:

First of all, we adopted a time-series forecasting model which effectively explained the variation of the reported results' number. Using this model, we are 80% sure that the number of reported results on March 1, 2023, should be between [10139, 30808]. Then, by adopting a machine learning model, we found out that the word attributes from the past have a limited impact on the percentage of scores reported that were played in Hard Mode, whereas word attributes of today's word almost have no impact at all.

Then, we used a refined deep-learning model to predict the distribution of the reported results. For instance, for the word EERIE on March 1, 2023, the forecasted probability distribution should be [0.649,7.579,26.298,32.614,20.930,9.63,2.298] (%)from 1 to 7 or more tries. This result passed a set of accuracy evaluations so it could be trusted.

After that, we established a difficulty classification model which is capable of classifying any given word by analyzing its attributes. When considering the word EERIE, the classification model infers that it should belong to the second easiest group of all 4 groups classified by difficulty. What's more, we derived the top 5 attributes that are associated with classification.

Lastly, we further explored other interesting features of this data set, including word cloud map, kernel density classification map, etc.

By analyzing the recent operational performance of *Wordle*, we further made several suggestions on *Wordle*'s future operation:

1. There is a downward trend of the number of players, so new playing mode should be introduced to attract new players.
2. The hard mode pct is increasing steadily. In order to improve the gaming experience, you should consider improving the difficulty of certain days' game.

3. In the future, you can consider setting 4 different game difficulties according to the classification result we acquired above.

Hope you find our suggestions helpful and wish Wordle become better and better!

Yours, Sincerely

MCM Team # 2307946

References

[1] W. contributors, "Wordle. in wikipedia, the free encyclopedia," *Wikipedia*, 2023.

[2] C. R. Nelson and C. I. Plosser, "Trends and random walks in macroeconmic time series: Some evidence and implications," *Journal of Monetary Economics*, vol. 10(2), pp. 139–162, 1982.

[3] T. Jakaša, I. Andročec, and P. Sprčić, "Electricity price forecasting — arima model approach," in *2011 8th International Conference on the European Energy Market (EEM)*, pp. 222–225, 2011.

[4] B. Laufer, "Ease and difficulty in vocabulary learning: Some teaching implications," *Foreign Language Annals*, vol. 23, pp. 147–155, 1990.

[5] J. Jagoda and T. Boiński, "Assessing word difficulty for quiz-like game," pp. 70–79, 2018.

[6] D. Balota, M. Yap, and t. Hutchison, K.A., "The english lexicon project," *Behavior Research Methods*, vol. 39, p. 445–459, 2007.

[7] M. Brysbaert and V. Warriner, A. B.and Kuperman, "Concreteness ratings for 40 thousand generally known english word lemmas," *Behavior Research Methods*, vol. 46(3), p. 904–911, 2014.

[8] P. Hoffman, M. Lambon Ralph, and T. Rogers, "Semantic diversity: A measure of semantic ambiguity based on variability in the contextual usage of words," *Behavior Research Methods*, vol. 45, p. 718–730, 2013.

[9] S. De Deyne, D. Navarro, A. Perfors, and et al., "The "small world of words" english word association norms for over 12,000 cue words," *Behavior Research Methods*, vol. 51, p. 987–1006, 2019.

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," vol. 30, 2017.

[11] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *NeurIPS*, 2019.

[12] T. Zhang, Z. Zhang, Z. Fan, H. Luo, F. Liu, W. Cao, and J. Li, "Openfe: Automated feature generation beyond expert-level performance," 2022.