WORDLE GUESSING

**Abstract**

This paper investigates how the New York Times predicts the number of reported results based on data from the Wordle puzzle game, and establishes a method for predicting the distribution of results and evaluating the difficulty of puzzles based on the solution word.

For problem 1, we first clean the data, correct or eliminate abnormal data, and then build a Quantity Prediction Model to fit the number of reported results in the data set and predict the future trend.

For problem 2, the difficulty coefficient of the puzzle is defined in the original dataset as a weighted average of the performance distribution, and what we need to do is to mine the potential features of the words themselves to predict the future performance distribution. After analyzing the correlation between each feature and the difficulty coefficient, several features related to the score distribution were identified. After that, in this paper, Decision Tree,Naïve Bayes,KNN,Logistic Regression,Linear SVC,Percepton,SGD,SVM are used as reference models and the difficulty coefficients are used as labels to train the word features and observe the performance of these models. The final results showed that decision tree was the best choice among them, using 80% of the data on the original dataset for training and the rest as the test set, which predicted almost 100% of the difficulty coefficients. After that, the percentage of each performance interval was used as a label to train and get a decision tree model that could predict the performance distribution. The 'EERIE' was used as a sample for several predictions, and the average result was taken as the output, and it was found to have about 12% bias, which is considered to achieve a good prediction result.

For problem 3,

For problem 4, this paper tries to mine many features in the original dataset, such as the percentage of the number of difficult patterns in the number of reports HR, the expectation of the score distribution i.e. the difficulty coefficient D, the percentage of less than j attempts to pass the game Uj, the position coding of words P,repetition Mul, the number of vowels VN, the vowel position coding VP, the total letter frequency coefficient FS, the first letter frequency coefficient FF, the double letter group frequency The final effective features are retained to train the machine learning model.

\*\*\* Translated with www.DeepL.com/Translator (free version) \*\*\*

Catalogue

[1 Introduction 3](#_Toc127759059)

[1.1 Background 3](#_Toc127759060)

[1.2 Restatement of the Problem 3](#_Toc127759061)

[1.3 Our work 4](#_Toc127759062)

[2 Assumptions and Explanations 4](#_Toc127759063)

[2.1 Condition Assumptions 4](#_Toc127759064)

[2.2 Notation 5](#_Toc127759065)

[3 Preparation 5](#_Toc127759066)

[3.1 Data cleaning 5](#_Toc127759067)

[3.1.1 Fix the Words 5](#_Toc127759068)

[3.1.2 Add new attribute 6](#_Toc127759069)

[3.1.3 Handle abnormal data 6](#_Toc127759070)

[3.2 Feature analysis 7](#_Toc127759071)

[3.2.1 Correlation analysis between features 7](#_Toc127759072)

[3.2.2 Attributes that affect in hard mode 8](#_Toc127759073)

[4 Quantity Prediction Model 10](#_Toc127759074)

[4.1 ARIMA Model 11](#_Toc127759075)

[4.1.1 Principle and implementation 11](#_Toc127759076)

[4.1.2 Forecast Results 12](#_Toc127759077)

[4.2 GM Model 14](#_Toc127759078)

[4.2.1 Principle and implementation 14](#_Toc127759079)

[4.2.2 Forecast Results 15](#_Toc127759080)

[4.3 Other Model 15](#_Toc127759081)

[4.3.1 Polyfit 15](#_Toc127759082)

[4.3.2 LSTM 16](#_Toc127759083)

[5 Difficulty Prediction Model 17](#_Toc127759084)

[5.1 Possible attributes of words 17](#_Toc127759085)

[5.1.1 Letter frequency 17](#_Toc127759086)

[5.1.2 Bigram frequency 18](#_Toc127759087)

[5.1.3 Multiplicity 18](#_Toc127759088)

[5.1.4 Vowel 18](#_Toc127759089)

[5.1.5 Degree of common use 18](#_Toc127759090)

[5.2 Preparatory work 19](#_Toc127759091)

[5.3 Training result and model comparison 19](#_Toc127759092)

[5.4 How difficult ‘EERIE’ is 20](#_Toc127759093)

[6 Model Evaluation 20](#_Toc127759094)

[6.1 Strengths 20](#_Toc127759095)

[6.2 Weaknesses 20](#_Toc127759096)

[7 References 21](#_Toc127759097)

[8 Letter for the Puzzle Editor of the New York Times 21](#_Toc127759098)

[9 About the Authors 22](#_Toc127759099)

[10 Appendix 22](#_Toc127759100)

# Introduction

## Background

The New York Times offers one puzzle per day, and players are considered successful if they guess the puzzle six times or less. Players are asked to guess from a list of 13,000 recognized words, and each guess must be an actual English word. Each guess will provide one feedback, and the color of the letter will change from gray, which means that there is no letter in the word, to yellow, which means that there is a letter in the word but in the wrong position, to green, which means that the correct position of the letter in the word has been guessed.

## Restatement of the Problem

•Use a model to explain the trend in the number of reported results and to predict the number of reported results on the date of March 1, 2023. Speculate whether there are other features that affect the number of times people guess the correct word in the HARD model.

•Build a model that predicts the distribution of the number of times it takes to guess the word when given any date later in the day. Predict, predict the distribution of the number of times needed to guess the word ERRIE. What are the uncertainties in your model?

• Identify the possible attributes to classify words according to their difficulty. Make predictions about the difficulty of the given word ‘EERIE’ and talk about the accuracy of your classification model.

• List some other interesting features in the dataset

## Our work

For question 1, we used multiple models to explain and predict the number of reported outcomes, including the ARIMA model, the GM model, and other models, of which our group believes that the ARIMA model best explains this trend, while the polynomial function does not fit such a trend. Based on multiple model tests, we believe that the reported results on March 1, 2023 will be between [6000,17500]. Through data visualization analysis, we concluded that some attributes of words, including letter repetition and word frequency, affect the percentage of scores in the difficult mode.

For question 2, we first defined the expectation of the score distribution as the most direct representation of the difficulty of the puzzle, and then analyzed the potential attributes in the solution word, expanded the original dataset to one that could represent multiple features in the word, and trained it with a number of machine learning models. Comparing the accuracy of each model, we found that the decision tree was the best predictor, followed by KNN models, etc., while rejecting some models that were clearly not applicable to the problem. With a sample size of 357 left, the decision tree had a prediction accuracy of 100%, so we also used it as a percentage of the score for predicting the puzzle 'EERIE' on March 1, 2023. We drop this sample into the model multiple times, train with each score share in the original dataset as a label, predict and take the average score, and finally predict the score share as (1%,3%,4%,28%,28%,21%,15%) with a difficulty factor of 5.02. And using the difficulty factor we defined as a label to take the training, the final prediction of ' EERIE' has a difficulty coefficient of 4.24, and based on this error, we are 88% confident that the model's prediction is reasonable.

For problem 3, we follow the model of problem 2 and only start from the attributes of the words themselves to classify the difficulty of solution words. Through correlation analysis, we find that the attributes of letter frequency, syllable frequency, number of vowels, vowel position, and commonness of words in the Wordle game are all related to the difficulty grading of solution words, but find We could not find any potential attributes that could largely classify the difficulty of solution words, and almost all of them were the result of the combined effect of these factors. For the word 'EERIE', we classified it as a 'harder' type, which is harder than about 90% of the words in the original dataset. For this prediction result, our classification model achieves an accuracy of 65%.(这些结果是临时糊上去的，不知道够不够时间做实验了)

For problem 4, our group first processed anomalous data on the original dataset, performed data cleaning, and analyzed the relevance of each feature in the dataset. In Section 5.1, we defined several useful fields as mining features for the dataset, while in Section 7.1, we mined the potential attributes of the words themselves. These features contribute significantly to both model building and validation.

# Assumptions and Explanations

## Condition Assumptions

1. the difficulty of the puzzle is related only to the nature of the word itself and can be measured using the distribution of the number of user attempts as a criterion, regardless of other influences.

2. the data on one-time guesses are allowed, although the existence of many people guessing correctly at once on the same day is considered to be the presence of players with other information or advanced game strategies.

3. the percentage of the data set given has a negligible impact on the results and, after rounding, does not have much effect on the training effect.

4. the assumption that the number of wordle participants will not be influenced by other factors to urge over time.

5. the model built by analyzing the given dataset is sufficient to explain the trend of the Reported results, without considering other factors.

## Notation

|  |  |
| --- | --- |
|  |  |

**Table 1** shows the notations that we use.

Table 1. Notation

|  |  |
| --- | --- |
| Symbol | Description |
|  | Record the sum of the proportion of (1,2,3,4,5,6,X) in i |
|  | Record the percentage of Number in hard mode in Number of reported results in i |
|  | Record the difficulty factor of the game in i |
|  | The percentage of games with less than j passes in record i |
|  | Repetition of word w |
|  | Number of vowel letters in word w |
|  | Number of all vowel positions in word w |
|  | Position coding of the jth letter in the word w |
|  | Frequency factor FS of the word w, determined by the frequency of occurrence of the letters in each position |
|  | Frequency factor FF of the word w, determined by the frequency of occurrence of the initial letter |
|  | Frequency factor FE of word w, determined by the frequency of the word itself |
|  | Frequency factor FB of the word w, determined by the frequency of all diacritical marks in the word |
|  |  |
|  |  |

# Preparation

## Data cleaning

### Fix the Words

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

• Delete one extra space after favor (Contest number=207)

• change tash to stash (Contest number=314)

• clen to clean (Contest number=525)

• na?ve to naive (Contest number=540)

• rprobe to probe (Contest number=545)

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

### Add new attribute

The first row of the original dataset "Problem\_C\_Data\_Wordle.xlsx" is deleted and the wrong words are corrected and saved as "data0.csv", which represents the initial dataset. In order to better explore the information in the dataset, we first add the following fields to the dataset.

• under\_j is the field that indicates the percentage of attempts with less than j attempts, abbreviated as Uj. Specifically, we call U7 Overall. the formula is:

|  |  |
| --- | --- |
|  |  |

•Hard\_rate is the ratio of the number of difficult modes selected to the reported results, abbreviated as H, and the formula is:

|  |  |
| --- | --- |
|  |  |

### Handle abnormal data

Next, using the Contest Number (or time) as the x-axis and the Number of reported results, Number in hard mode, and Hard Rate as the y-axis to make the relationship graphs, Figure 1, Figure 2, and Figure 3 respectively, we can clearly find two We can clearly find two abnormal data, which are the record with the Contest number = 281 (its Overall is 126%), and the record with the Contest number = 529 (its Hard Rate is 93%). We choose to delete these abnormal records to get Figure 4.

|  |  |
| --- | --- |
| 图表  描述已自动生成 Figure 1. The relationship chart between Contest Number and Number of reported results | 图表, 散点图  描述已自动生成  Figure 2. The relationship chart between Contest Number and Number in hard mode |

|  |  |
| --- | --- |
| 图表  描述已自动生成  Figure 3. The relationship chart between Contest Number and Hard rate before cleaning up abnormal data | 图表, 散点图  描述已自动生成  Figure 4. The relationship chart between Contest Number and Hard rate after cleaning up abnormal data |

(If you have time to add to this place, first make three boxplots to indicate the noise data, and then replace them with interpolation or other processing methods)

Although it seems that the Hard Rate of two points still looks weird, we think it is still manageable. We think the current data set is available overall, and we will save the new data set as "data1.csv". At this point, the data cleaning is basically complete.

## Feature analysis

### 3.2.1Correlation analysis between features

In python, the heatmap of seaborn library is a convenient tool for feature correlation visualization, which indicates the degree of positive correlation with cool colors and the degree of negative correlation with warm colors, thus allowing us to better discover the correlation between features. We try to use heatmap for data0, and the result is shown in Figure 5

图表, 树状图

描述已自动生成

Figure 5. The heatmap obtained after removing the Word and Date fields in data0

（好像这个还不太好分析，所以后面看能不能改改吧，或者做Difficulty的热图）

### 3.2.2Attributes that affect in hard mode

Attribute 1: Number of word repetitions: Our group found that if there are repeated letters in the target word, the number of successful solutions or solution time will increase. In other words, since most of the individual's knowledge base are words without letter repetitions, we will rarely use words with repeated letters as target words in our questions. Since the target words are all 5 in length, we divide the words according to the number of letter repetitions into cases where each letter appears once, one letter appears twice, one letter appears three times, two letters appear twice each, and so on.

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

W1: One occurrence of each letter (no word repetition)

W2: One letter appears twice

W3: One letter appears three times

W4: Two occurrences of each of the two letters.

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

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

DW1<DW2<DW4<DW3.

This hypothesis can also be approximated from the line graph, where we assume that the higher the number of answers from 4 times and earlier, and the lower the number of answers from 5 times and later, the easier the words are, so we can also derive the above difficulty ranking by sorting from left to right.

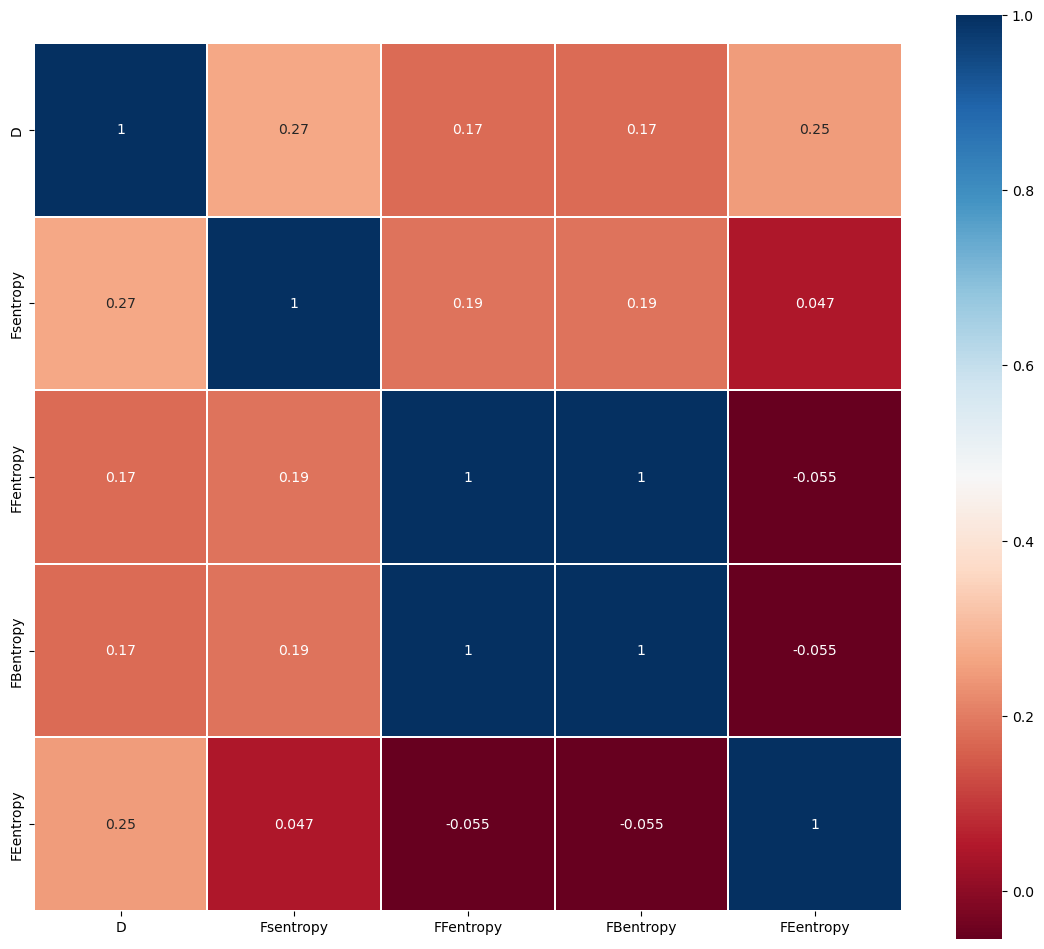
Next, we quantify the difficulty using the average score, which is a weighted average of the seven attempts.

The final ranking is the same, so we can conclude that there is a strong correlation between the repetition rate of letters in a word and the score percentage, i.e., the repetition rate of letters in a word has a strong influence on the result.

Attribute 2: After that, we determined four other attributes by reviewing relevant information, the frequency of occurrence of each letter of FS; the frequency of occurrence of the initial letter of FF; the familiarity of FE words; and the frequency of diphthongs in FB double words. We then used the information content formula to calculate the information content of the words under each attribute, after which we calculated the Pearson correlation coefficient of each attribute with difficulty separately. The results obtained were statistically significant and showed a low correlation overall. The formula for calculating the information content is as follows.

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

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



# Quantity Prediction Model

To solve problem 1, we tried to use ARIMA, LSTM model, and GM model to predict the number of reported outcomes in the future, and through our experiments, we found that ARIMA model is relatively more suitable prediction model and can give a reliable prediction result. After our group's experimental analysis, the number of reported outcomes on March 1, 2023 is predicted to be in the interval [6200,17200].

## ARIMA Model

We have already analyzed the trend of the number of reports over time, and in order to analyze the future trend of the number of reports, we use the ARIMA model to predict the number of reports in the next 60 days, and to derive the interval of the number of reports on March 1 as required by the question. a basic method of the ARIMA model is the difference method, that is, the time series is differenced to eliminate its trend seasonality and other characteristics The transformed series is a smooth time series. At this point, the transformed series can be assumed to be an ARMA series for further study.

### Principle and implementation

For a zero-mean smooth sequence {X\_t,t=0,1,2,---}, if it can be expressed as a weighted sum of the first p terms and the sum of zero-mean smooth white noise, as follows:

|  |  |
| --- | --- |
|  |  |

and introduce the backward shift operator:

|  |  |
| --- | --- |
|  |  |

and the arithmetic polynomial:

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

If satisfies

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

If satisfies

|  |  |
| --- | --- |
|  |  |

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

|  |  |
| --- | --- |
|  |  |

The smoothness condition of ARMA model is φ(B) X\_t=0 all roots fall outside the unit circle, and the reversibility condition is α(B) = 0 all roots are outside the unit circle. The two properties are very important in theoretical and practical problems

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

Step 1: Calculate the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the time series we want to predict, and determine whether it obeys the ARMA model by judging whether it is truncated or trailing, and if at least one of them is not, the original series is non-stationary, then it is differenced to the first order, and the ACF and PACF are discussed until the differenced series is a stationary series

Step 2: Determine the parameters in ARIMA (p, 1, q): use the AIC and BIC criteria to fix the order and select the optimal parameters p, q

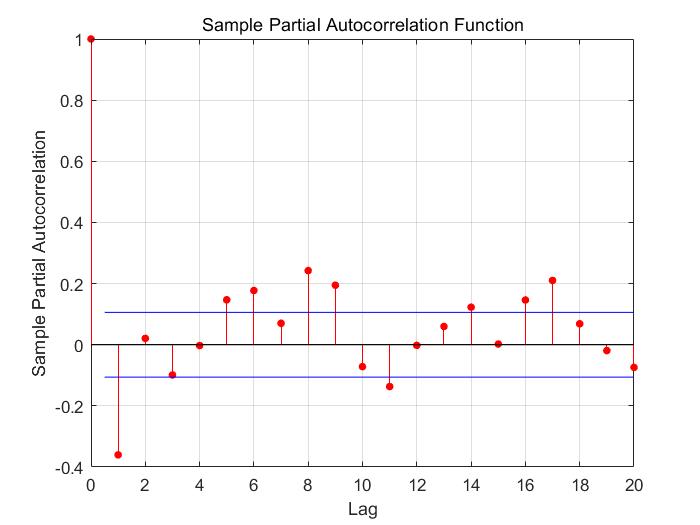
Step 3: Validation of the model: the residual test is performed to judge the reasonableness of the model, after which the model is used for prediction.

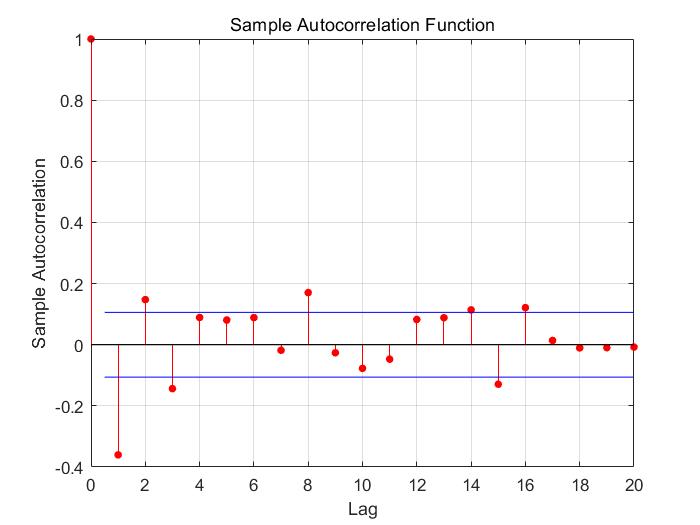
### Forecast Results

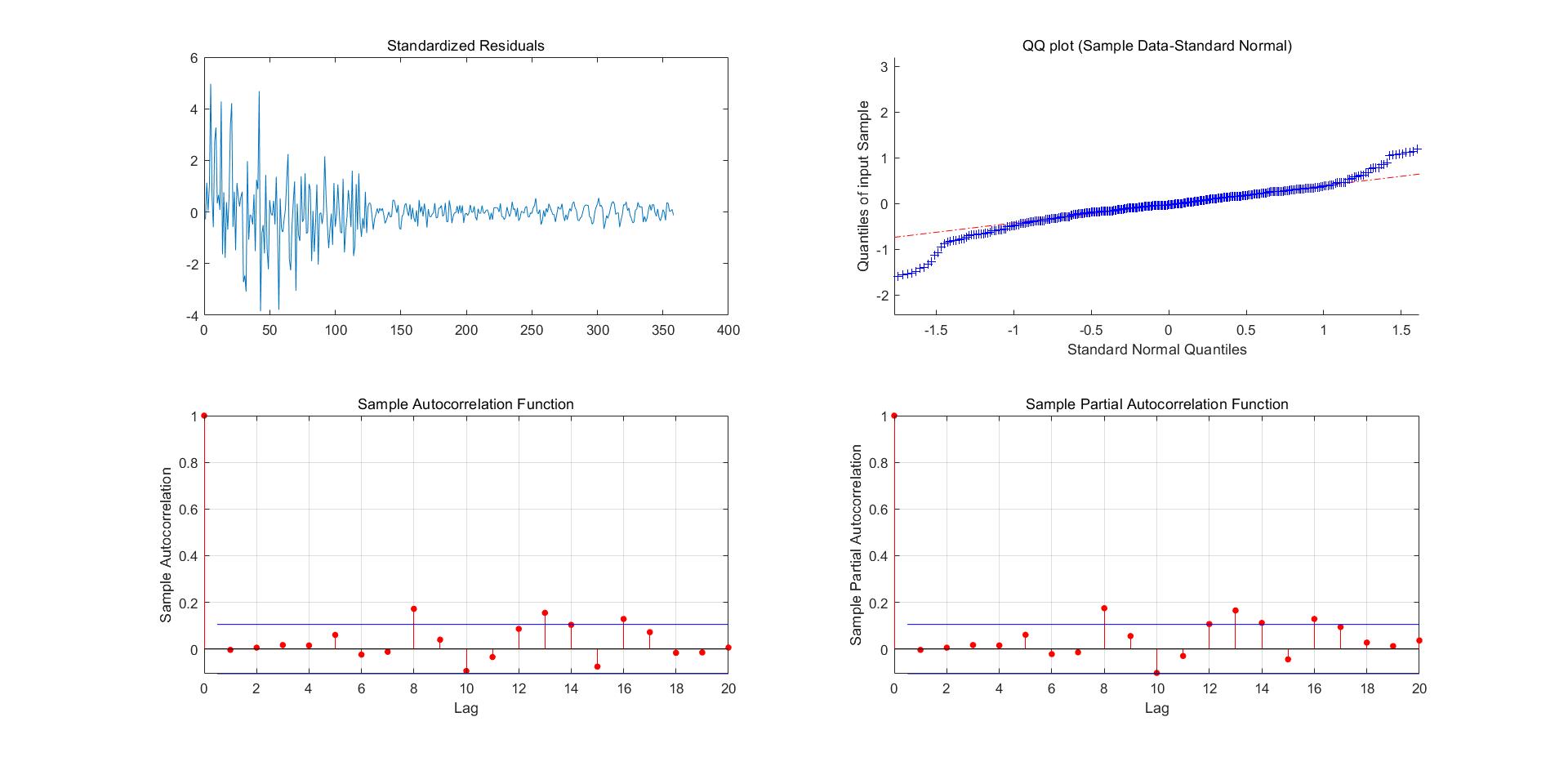
First, we calculated the ACF and PACF for the reported number of people and performed the smoothing test, which did not pass. After performing the first-order difference, the smoothing test was performed again and the logical output was obtained as shown in the following table.

|  |  |
| --- | --- |
| **Adf** | **1** |
| **Kpss** | 0 |

The ACF and PACF of the time series after first-order differencing are shown in Figure X below





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

The final model obtained:

|  |  |
| --- | --- |
| Finally, using the model to project backward 60 days for January 31, 2022, we obtain the data .=17079 for March 1, 2023, which means that the number of people reported on that day is approximately 17079. |  |

## GM Model

Gray forecasting is a forecasting method that predicts gray systems. The gray system is between white (completely known) and black (completely unknown). Gray forecasting is done by identifying the degree of dissimilarity of development trends among system factors, i.e., correlation analysis, and generating the original data to find the pattern of system changes, generating a data series with strong regularity, and then establishing the corresponding differential equation model to predict the future development trend of things

### Principle and implementation

Here we use the GM (1, 1) grayscale model, the GM (1, 1) bit first-order differential equation model containing only one variable. If the number of original samples is known to be n nn,here first define the tolerable coverage interval Θ as follows:

|  |  |
| --- | --- |
|  |  |

Define the column order ratio λ(k) as follows:

|  |  |
| --- | --- |
|  |  |

If the following conditions are met:

|  |  |
| --- | --- |
|  |  |

Then we say that x ( 0 ) x(0) x(0) is predictable by grayscale as GM(1,1) data.

We write the predicted value as x 0 ( k ) :

|  |  |
| --- | --- |
|  |  |

At the end of modeling, residuals and level difference tests are also required, and the residuals ρ are defined as

|  |  |
| --- | --- |
|  |  |

Define the level difference ξ as :

|  |  |
| --- | --- |
| ) |  |

### Forecast Results

图表, 直方图

描述已自动生成。

It is easy to see from the figure that the grayscale prediction model is not as accurate a fit as the ARIMA model. Because the grayscale prediction model is suitable for predicting shorter time series, the results of the grayscale prediction model have a tendency to be small in this data.

## Other Model

### Polyfit

We tried to fit this trend using a simple polynomial and made a polynomial of order 1 to 6 to fit the data as shown in Figure X.

图表, 折线图

描述已自动生成

Figure X. figure of results fitted using polynomial function

From the figure, it is easy to see that the polynomial function is not suitable for fitting such a changing trend. When the number of times is small, there is no way to calculate the results as close as possible to the scatter point; when the number of times increases, the function, although closer to the location of the triple point, will appear overfitting phenomenon, that is, the fitted sample performs well while in the predicted sample there is a significant deviation. Let's say that the graph has a sharp upward trend at the end of the fifth equation and a sharp downward trend at the end of the sixth equation, which are not what we want.

### LSTM

LSTM (Long-Short Term Memory) is a special variant of recurrent neural network with a "gate" structure. The LSTM has three gates, namely the forgetting gate, the input gate and the output gate, which determine whether information is remembered or forgotten at each moment, the input gate determines how much new information is added to the cell, the forgetting gate controls whether information is forgotten at each moment, and the output gate determines whether information is output at each moment. Our group tried to use LSTM to predict the time series of Reported Results data, and the results are shown in Figure X.

图表, 直方图

描述已自动生成

From the result graph, it is easy to see that the trend of Reported\_results is fitted very approximately using the LSTM model and performs reasonably well on the prediction task.

# Difficulty Prediction Model

## Possible attributes of words

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

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

### Letter frequency

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

|  |  |
| --- | --- |
|  |  |

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

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

|  |  |
| --- | --- |
|  |  |

The word0 represents the initial letter of the word word.

### Bigram frequency

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

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

### Multiplicity

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

|  |  |
| --- | --- |
|  |  |

### Vowel

Vowels are an indispensable part of words, so the position and number of our vowels are factors worth considering. In addition, recording the number of vowels also taps into the effect of the number of consonants in a word. We define VN as the number of vowels in the word and VP as the binary code of all vowel positions in the word, for example, the position of vowels in 'EIRE' is coded as [1,1,1,0,1].

### Degree of common use

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

## Preparatory work

For the above mined attributes, we use heat map to observe the correlation between features, especially with the difficulty coefficients, before we transform some difficult frequency coefficients into information entropy. The results are shown in Figure X图表, 树状图

描述已自动生成

Figure X.

To get an accurate model for predicting word difficulty, all the features mined in Section 6.1 are put into the dataset and saved as "word\_data.csv", and the preview is shown in Figure X. The normalized difficulty factor D is used as the label Y, and the other fields are used as the input features X. We use these features to train the machine learning model to derive the difficulty factor. Before training, we remove the Word field because it belongs to the numeric type. Instead, it is replaced by the positional encoding posi, the encoding of the letters on each bit of the word that corresponds to the ranking of the letter frequencies. This allows to handle the data that are not legal in the training while preserving the Word information.表格

描述已自动生成

Figure X. Preview image of the dataset word\_data

## Training result and model comparison

After dividing the training and testing sets, we chose several common models to train and use the accuracy of validation as a criterion to compare the models, including Decision Tree, Naïve Bayes, Logistic Regression, KNN, Linear SVC, Perceptron, SGD, SVM, and the results are shown in Figure X.

|  |  |
| --- | --- |
|  | 图表, 条形图  描述已自动生成 |
| Figure X. | Figure X. |

Figure X.List of training results for each model

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

## How difficult ‘EERIE’ is

Our following the steps in 6.1 to obtain other features of the word 'EERIE' and put it into the model for prediction resulted in a predicted label value of 107 and a difficulty factor of 4.32 after normalized inversion, this difficulty put in 'data1.csv ' means that it is more difficult than 241 words out of 358 words. The words that are more difficult than it are basically rare words, but despite this, we believe that the actual difficulty of 'EERIE' is greater, mainly because it is very rare and has a high letter repetition.

We believe that this error is due to the small sample size, which makes the trained model not take into account this type of words very well, and the fact that there is no word in the original dataset that has a vowel letter repeated three times, which can greatly hinder the model's judgment in machine learning. All in all, for words that are commonly used or have a small spatial distance from the sample, we believe that the model has more than 90% accuracy to distinguish the difficulty; while for words that do not capture the main features in the sample, our prediction rate is only guaranteed to be around 70%.

# Model Evaluation

## Strengths

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

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

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

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

## Weaknesses

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

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

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

# References

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

# Letter for the Puzzle Editor of the New York Times

Dear Puzzle Editor of the New York Times:

最近我们总是在社交媒体上被大批莫名奇妙的格子图猜谜游戏wordle令我们十分着迷，我们都想着是否能用最少的次数解出谜题。每天都有数以万计的人们在推特上分享他们的猜测结果，形成一股猜词风潮。语言学家、统计学家、计算机学者也纷纷开始研究wordle，运用他们领域的知识来分析谜题，求解单词。

我们参考了2022年1月7日到12月31日的数据，发现每天在推特上分享他们成果的人都在逐渐变化，我们建立一个模型解释这种变化，并希望能给出未来某一天在推特上分享他们wordle结果的人数区间，我们使用了Aria模型、GM模型、LSM三个在机器学习领域常见的模型来进行时间序列预测。时间序列预测是通过对按照一系列时间间隔排列的一组数据的分析，来进行预测，发现现象和规律，比如这里的每天在推特上分享他们wordle答案的人数。通过三种模型，我们得出预测区间大概（）

（词语的属性影响报告在困难模式的百分比）

（预测各猜中次数区间）

同样的，在实际做题中，我们总会感慨今天的词好难，今天的词比较容易。我们知道（As we all know）wordle词库中只有12937个，我们是否能过通过大量数据真的对每个单词给进行难度分类，判断今天遇到的单词是否真的是颇具难度的。我们同样地采用2022年1月7日至12月31日的数据来进行预测。在这里我们考虑单词本身的性质，选取了字母频率、双音节频率、字母重复度、含有元辅音个数、单词常见度、单词每一位上的字母的编码个可以通过分析单词能得出的特征。在这里需要特别注意的是单词常见度这一特性，生活中越常见单词就越好猜，人们也往往会猜测自己较为常用的单词，我们通过大文本统计了wordle词库中较为常见的单词，这些数据可以在kangol中查到。

我们将挖掘到的所有特征存入数据集,将难度系数D标准化的结果作为标签Y,其他的字段作为输入特征X。我们选用了大量的机器学习模型来进行训练，这些模型包括Decision Tree,Naïve Bayes,Logistic Regression,KNN,Linear SVC,Perceptron,SGD,SVM。在划分好训练集和测试集后，我们选用了常见的几个模型训练并以验证的准确率作为标准来比较模型的优劣。我们惊人的发现，决策树训练效果甚至达到了百分之一百，而perception、SGD、SVM等几个模型，则只有百分之几的准确率。

我们选取了eerie这个具体的单词，希望能获得关于他的难度分类。向上面一样我们获取这个单词的特征，并放入难度模型中进行预测，我们得到的预测标签值为107，标签化反转后的结果为难度系数4.32，这个难度放在’data1.csv’里意味着在358个单词中它比241个单词更难。而比它更难的基本上都是生僻词，尽管如此，我们仍认为’EERIE’的实际难度是更加大的，主要因为它非常罕见而且字母重复度很高。

I am writing to express my appreciation for your puzzles. They are always challenging and engaging, and I look forward to seeing a new puzzle every day. I think the puzzles are a great way to exercise my mind and learn something new every day.Thank you for your hard work in producing these puzzles each day. I look forward to seeing what new challenges lie ahead(最后写点客套话)

Sincerely,three college students

# About the Authors

# Appendix