Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2307787

Wordle serves as a popular puzzle and continues to grow in popularity. Have been asked to do an analysis of the data provided, we modeled according to the tasks.Several models are established: Model I:Interval Prediction Based on ARIMA Model; Model II:Gradient Descent Optimization Model; Model III:Associated Percentage Prediction Model; Model IV:Words Difficulty Classification Model. With varieties of **visualization methods** applied, most of our results are displayed in figures, making it **intuitive** to be shown.

Model I:As the daily number of reported results(from 7,Jan,2022 to 31,Dec,2023) is given, we firstly set out to build the **ARIMA(p,d,q) model**. Then, based on the verification of the unit root test and white noise test, the validation of using ARIMA model is fully proved. Next, the 250 previous data is used to predict the rest 109 data. we select the **strongly predictive** time series which matches the predicted known data best and determine d=1. With the introduction of ACF and PACF, the parameter p and q of ARIMA is identified 1 and 0. Finally, according to the ARIMA(1,1,0), The **interval prediction** of number of reported results is shown in Figure 9.

Model II: To simplify the problem, we split the words into letters and define the type and number of the letters as the attributes of the word, regardless of the sequence of the order. For each single word, its attribute is represented by a 1*26 vector. By combining all vectors of sample words, we got a matrix. In this step, we can **quanlify** the attributes of the words, since the number of letters in one word is five. The results are shown in the Figure 10,which demonstrates that attributes of the word affect the percentage of scores in hard mode greatly.

Model III: This model is actually a supplement to Model Ⅰ. In the Model 1, the predicted data only considers the factor of date and there is a necessity to consider the factor of a given word. And in this section, what we focus is the factor of word, as the factor of date has been discussed in Model 1. By collecting letters' data and converting them into a matrix, we can finally quantify the features of the words. The result is displayed in Figure 12.

Model IV: Based on the Model Ⅱ, we can score the difficulty of all letters through creating a **"Grading System"**. Meanwhile, we add up all the scores of letters in one word. Therefore, we are able to score each word and **classify words** by the difficulty. And we use scatter diagram to plot the different classifications in different colors in order to clearly present the relationship between each sample.The result is given in Figure 13.

In addition, we list and describe 2 interesting features of the data set : One is we find that the **strong relativity** of number in hard mode and reported results, which is displayed directly in Figure 15. The other is that the try of different words can show different amount of information and the result is visualized in Figure 17.

Finally, as for the **sensitivity analysis**, we randomly choose 50% of the samples, use them to create 2 matrix and calculate their personal relativity with the result shown in Figure 18. Afterwards, a letter of results has been written for Puzzle Editor of the New York Times.

Keywords: Algorithm,Word2Vector, Python,Forecast,Time series prediction, ARIMA

Content

1 Introduction

1.1 Problem Background

“Turning a few early yellows into greens sometimes leaves me a bit stumped when everything else is grey—now what? Luckily for me, I'd stumbled upon the right sort of greens in the right places this time, and they left me with little guessing room for anything other than today's answer.”said by an wordler. Recently, word fans all across the world have come to love playing wordle, which has been the most appealing crossword game available right now. The players’ daily task is to solve the same puzzle and they can also post the game results on social media without giving away the solution to others.

[1] Here we list the basic rules below:

- Players have to guess the Wordle in six goes or less.
- Every word players enter must be in the word list. There are more than 10,000 words in this list, but only 2,309 are answers to a specific puzzle.
- A correct letter turns green.
- A correct letter in the wrong place turns yellow.
- An incorrect letter turns gray.
- Letters can be used more than once.
- Answers are never plurals.
- Hard mode forces players to play any correct letters on subsequent guesses. So if players have a green S at the start, players have to keep playing S at the start from then on. And if players have a yellow R in the word, all subsequent guesses would have to include R somewhere too.(hard mode)

1.2 Restatement of the Problem

Considering the background information and restricted conditions identified in the problem statement, we need to solve the following problems:

- Build a mathematical model to explain how the number of reported results vary daily and determine the prediction interval for reported results on March,1,2023. Based on the model, analyze and find out whether the attributes affect the percentage of scores in hard mode.
- For a specific word, develop a model to predict the distribution of the reported results for a future date. Take the word EERIE on March 1,2023 as an instance.
- Classify solution words by difficulty based on an improved model.
- List and describe some other interesting features of this data set.
- Considering the results summarized above, prepare a one- to two-page letter to the Puzzle Editor of the New York Times.

1.3 Literature Review

Generally speaking,for natural language processing to classify sentences, the core is to split sentences into words, and convey the number of words to a matrix， which is called “word2vector”.This technology was first published in 2013 [2], and got popular soon after.With the help of it, Neuron Network got many improvements[3]. What’s more,there are many more new models comes from it,like CBOW(Continuous Bag-of-Word)and SkipGram. In “Machine Learning In Action”[4], it’s seen as a very splendid and easy way to deal with speech publishing filtrating.

In this “worldle” case, the propose is to find the attributions in every word, so by parity of reasoning, we split words into letters, and convey the indexes and numbers of letters to a matrix.

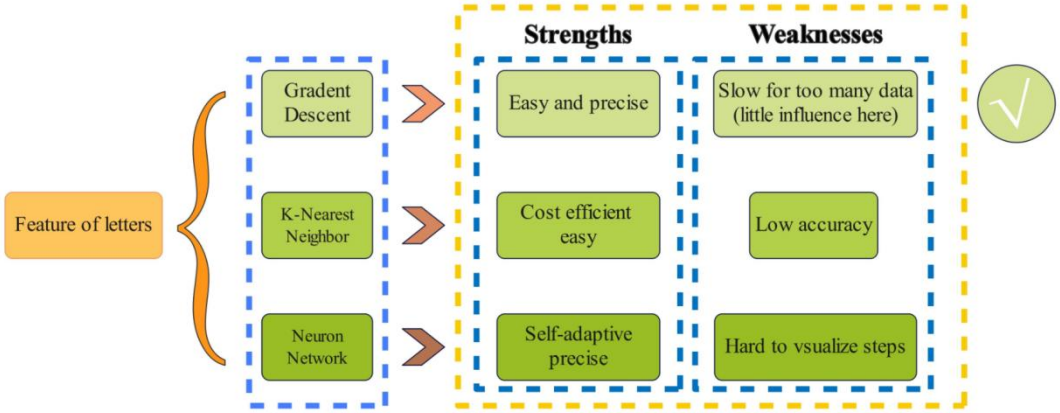

Figure 1 : Choosing The Fittest Algorithm

1.4 Our Work

The problem requires us to use the data provided to analyze and predict the number of the reported results, percentage of scores(1,2,3,4,5,6,X) for a given future solution word on a future date and classify solution word by difficulty. Our work mainly includes the following:

1) Based on the data provided, a prediction ARIMA model of number of reported results is established.

2) Considering the date factor and word factor, the associated percentage scores of the(1,2,3,4,5,6,X) is given for a future date. And we take the word "EERIE" on 1,March,2023 as an instance.

3) Based on the Gradient descent optimization model based on natural language processing, this article effectively classifies solution word by difficulty and demonstrates great validity and applicability.

4) Describe some other interesting features of this data set.

In order to avoid complicated description, intuitively reflect our work process, the flow chart is shown in figure:

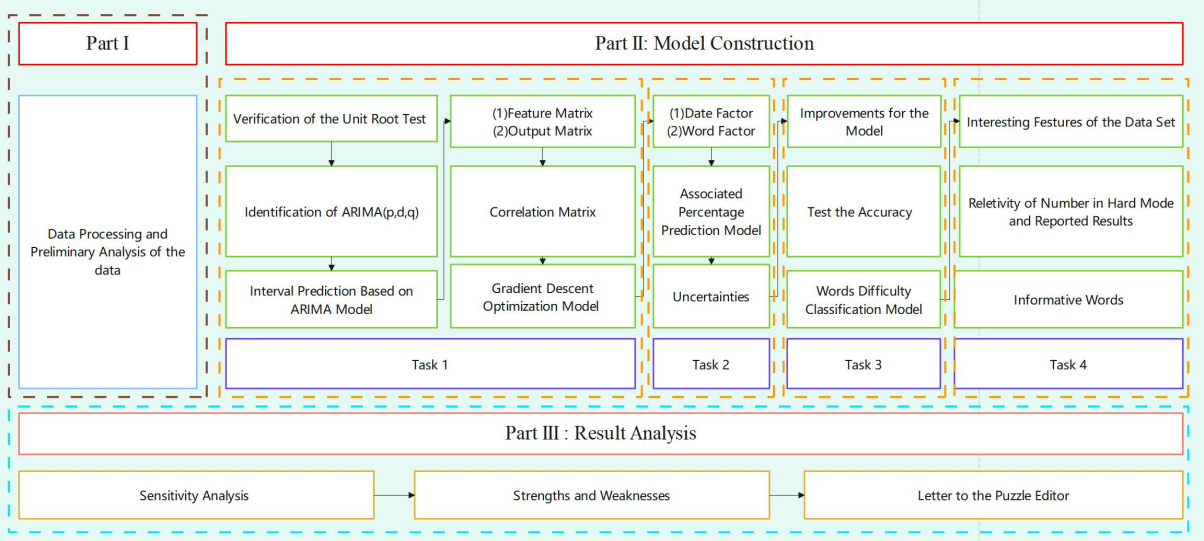

Figure 2 : Flow Chart of Our Work

2 Assumptions and Explanations

When players are solving the Wordle game, the internal factors will have a big impact on their performance. To simplify the problem, we need to make reasonable assumptions, and each hypothesis is followed by its corresponding explanation:

- **Assumption 1: No players cheat or directly share their results with others revealing the actual letters.**

Explanation: As players solve the same Wordle daily, it can be easy for players to compare their answers with those of others. We assume the behavior of cheating is ignored in the process of playing Wordle for two reasons. The number of the cheater is relatively small and the cheating scores have no reference value, so the cheating behavior in the process of playing Wordle can be ignored.

- **Assumption 2: The ratio of the number of reported results to the number of total results equals to a constant.**

Explanation : The ratio may change daily for unpredictable accidental factors. In a relatively short term, the proportion of the reported results vary little. Therefore, it is assumed that the ratio remains unchangeable.

- **Assumption 3 : In hard mode, the percentage of players solving the puzzle in two, three or four guesses will approximately reduce by 10%. The percentage of players solving the puzzle in five, six or more guesses will approximately increase by 10%.**

Explanation : The percentage that guesses the word in one try, two tries, three tries, four tries, five tries, six tries or more in hard mode is not clearly given in the problem. So we make such assumption above, since the actual impact of the hard mode is very complicated and it is tough to calculate the accurate percentage of scores in hard mode. Besides, the rules

in hard mode make no difference to the percentage of players solving the puzzle in one try.

- **Assumption 4 : Identify the attributes of words based on the number and type of letters, regardless of the sequence of the letters in the word.**

Explanation : [2]There are a total of 12,986 five-letter words in the Wordle word list, and in the list only 2,309 words are the answers for the puzzle. We find out that when the number and the type of the five letters are determined, the word is largely determined as well. Since the order of the letters is minimal, we exclude the existence of the sequence of the letters.

- **Assumption 5: Assume the research data is accurate.**

Explanation : We assume that number of reported results, number in hard mode and percentage of scores do not show obvious measurement deviation, so we establish a more reasonable model based on it.

Additional assumptions are made to simplify analysis for individual sections. These assumptions will be stated and justified at the appropriate places.

3 Notations

The key mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Symbol	Description
p	The sequence value lags by order p
d	The minimum differences required to transform the time series into smooth series
q	The error term lags by order q
n	Number of output features
m	Number of samples
$x_{\mathrm{m\,n}}$	For the m-th sample,n-th Feature
$\mathrm{h}\left(x^{(\mathrm{i})}\right)$	Predicted value of sample(i)
$\mathrm{y}^{(\mathrm{i})}$	True value of sample(i)
R_n	Relativity of the n-th feature
r	Relativity as matrix elements
ξ_k	Chi-Squre Distribution:Probability Density Function Value For x=7-k

Note: There are some variables that are only used in one section and are better suited to be defined in the corresponding place.

4 Data Processing and Preliminary Analysis

Since the provided Data File.Problem C Data Wordle.xlsx is large and not intuitive, we visualize some data for display. We can see the daily variation of the number of the reported results from the figure below.

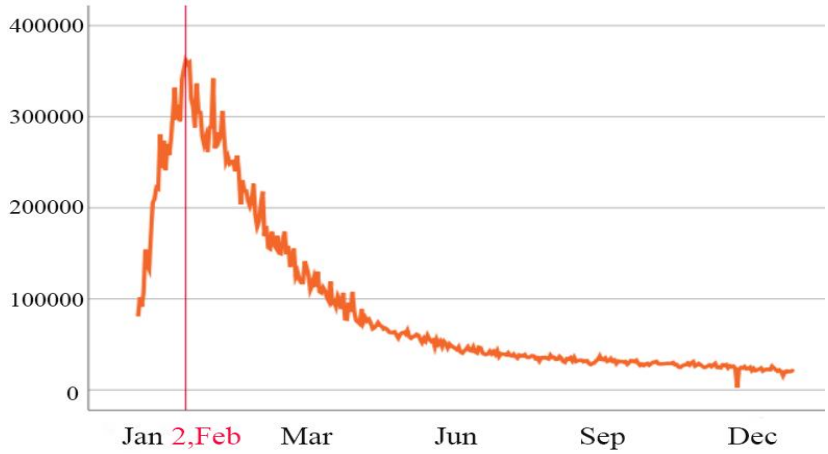

Figure 3 : The Daily Variation Of Number Of Reported Results

The number of the reported results can been seen above: from January 2022, march of players got actively involved in the Wordle puzzle. The peak number of reported results on 2,Feb,2022 impressively reached to 361,908. Between February 2022 and June 2022, the number of the reported results dropped rapidly. Since June 2022, the number of the reported results maintained a flat downward trend.

5 Interval Prediction Based on ARIMA Model

5.1 Unit root Test for time series forecasting

From the time series graph (Figure 1), we can easily find that the number of the reported results is non-stationary time series, which indicates the original data needs to be differenced. So, we use the diff function in SPSSPRO[5] to display first-order and second-order curves and the results are displayed in the following figure.

Figure 4 : Zero to Second-order Difference

Unit root test methods include ADF(Augmented Dickey-Fuller) test, DF(Dickey-Fuller)

test, PP (Philipps-Perron) test, etc. The DF test can only be use for the first-order case,and when there is lagged correlation of higher order in the series, the ADF test can be used, which is an extension of DF test. The advantage of the ADF is that it excludes the influence of autocorrelation by including the first-order downward differential term at the end of the fall. All taken into consideration, we choose to use the ADF test, which is more applicable corresponding to the problem.

Table 2 : ADF Test List Of Number Of Reported Results (d=0,1,2)

ADF test list							
Variable	Difference order	t	P	AIC	Critical value		
					1%	5%	10%
Number of reported results	0	-3.867	0.002***	7203.313	-3.45	-2.87	-2.571
	1	-4.242	0.001***	7195.208	-3.45	-2.87	-2.571
	2	-10.663	0.000***	7176.608	-3.45	-2.87	-2.571
Note：***、**、* represent the significance level of 1%、5%、10%, respectively							

The results of the unit root test show that p=0.002, 0.001, 0.0001<0.005. (The result of the unit root test shows p<0.05, indicating that the data is consistent with the characteristic of a smooth time series.). It illustrates that the original series data, the series after first-order difference and the series after second-order difference are all smooth time series. In order to find out the best time series, the following tasks are carried out.

From the problem, a total of 359 data on daily reported results are provided. We set out to use the previous 250 consecutive data to predict the rest 109 data. By examining the prediction performance of time series of varied differential orders, we will select the strongly predictive time series[6]. The results are shown in the figure below:

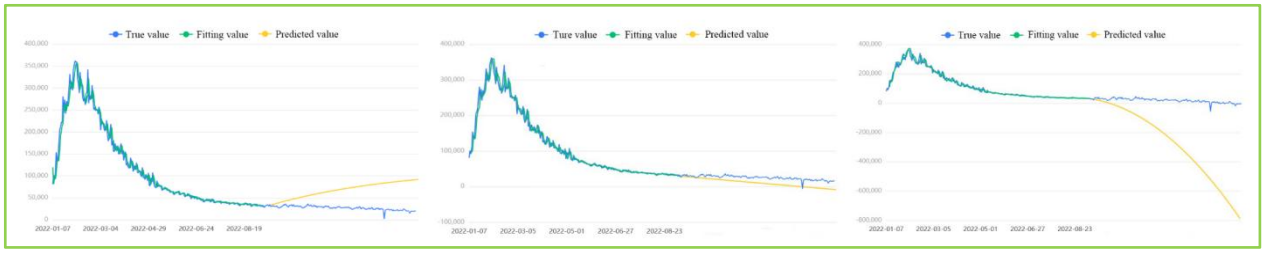

Figure 5 : Time Series Prediction of Number of Reported Results (d=0,1,2)

The figure clearly shows that the predicted results in second-order time series are more closely matched to the actual data. Due to the apparent differences in results and the limited pages, what we need to focus on is selecting the best time series, so we directly choose to use and research the second-order time series in the following questions.

5.2 ACF, PACF Function and Identification of ARIMA (p, d, q)

The original series data becomes smooth after first-order difference processing.The ARIMA model was initially determined to be ARIMA (p, 1, q) as d=1. To identify the rest

parameter p and q, we need to plot autocorrelation and partial autocorrelation of smooth series.

Autocorrelation function, which is known as ACF, can reflect the time series correlation of adjacent observations in the data. The formula of ACF is below:

$$\mathrm{ACF(m)} = \frac{\mathrm{Cov}(X_i, X_{i-m})}{\mathrm{Var}(X_0)} = \frac{\frac{1}{n-m}\sum_{i=m+1}^{n}(x_i-\bar{x})(x_{i-m}-\bar{x})}{\frac{1}{n}\sum_{i=m+1}^{n}(x_i-\bar{x})^2} \quad (1)$$

In SPSSPRO, use ACF function which is based on formula (1) to plot autocorrelation. The result will be shown in the following figure 2.

Partial autocorrelation function, which is known as PACF, can measure the correlation between y(t) and y(t-k) after removing the effect of k-1 intermediate values of the random variable.The formula of PACF is below:

$$\varphi_{kk} = \begin{cases} \rho_1 , & k = 1 \\ \dfrac{\rho_k - \sum_{j=1}^{k-1}\varphi_{k-1,j}\,\rho_{k-j}}{1 - \sum_{j=1}^{k-1}\varphi_{k-1,j}\,\rho_{k-j}} , & k > 1 \end{cases} \quad (2)$$

In SPSSPRO, use PACF function which is based on formula (2) to plot partial autocorrelation. The result are also directly displayed in the figure 2.

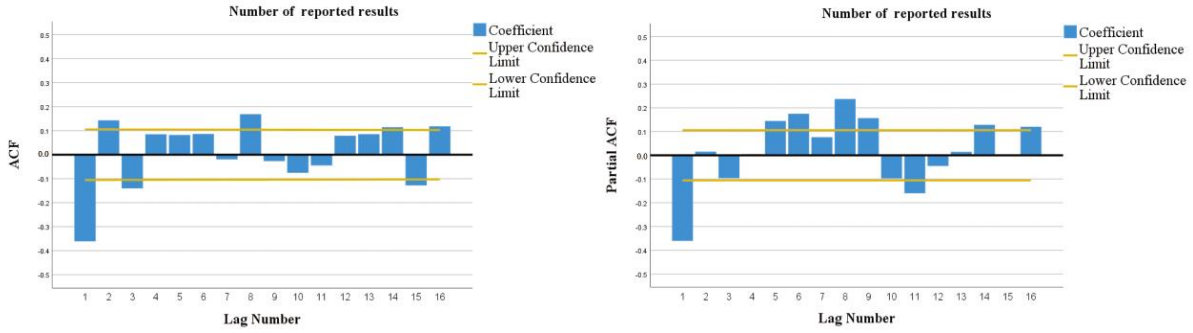

Figure 6 : ACF And PACF Of Number of Reported Results (d=1)

From the figure 2 above, we can initially determine the values of p and q by observing the trend of autocorrelation coefficients in the graphs. From the autocorrelation graph of the series, we can see that the series is truncated at the zero order, and the preliminary judgement is q=0. From the partial autocorrelation graph of the series, we can see that the series is truncated at the second order, and we can make a preliminary judgement that p=1.

5.3 White Noise Test On the ARIMA Model

By plotting the Q-Q diagram of the residual series(figure 8), we can see that the

residuals approximately falls in a straight line, and we can initially judge that the series is random normal distribution.

Model Statistics

Model	Number of Predictors	Model Fit statistics: Stationary R-squared	Model Fit statistics: R-squared	Ljung-Box Q(18): Statistics	Ljung-Box Q(18): DF	Ljung-Box Q(18): Sig.	Number of Outliers
Number of reported results	1	.132	.982	83.507	17	.000	0

ARIMA Model Parameters

					Estimate	SE	t	Sig.
Number of reported results	Number of reported results	No Transformation	Constant		20512.343	24737.515	.829	.408
			AR	Lag 1	-.363	.049	-7.351	.000
			Difference		1			
	Date	No Transformation	Numerator	Lag 0	-1.444E-6	1.726E-6	-.837	.403

Autocorrelations

Series: Noise residual from Numberofreportedresults

Lag	Autocorrelation	Box-Ljung Statistic: Std. Error[a]	Box-Ljung Statistic: Value	Box-Ljung Statistic: df	Box-Ljung Statistic: Sig.[b]
1	.006	.053	.015	1	.902
2	-.024	.053	.228	2	.892
3	-.088	.052	3.060	3	.382
4	.086	.052	5.772	4	.217
5	.178	.052	17.376	5	.004
6	.138	.052	24.392	6	.000
7	.082	.052	26.843	7	.000
8	.202	.052	41.870	8	.000
9	.006	.052	41.881	9	.000
10	-.127	.052	47.837	10	.000
11	-.055	.052	48.954	11	.000
12	.123	.052	54.583	12	.000
13	.195	.052	68.805	13	.000
14	.134	.052	75.549	14	.000
15	-.067	.052	77.242	15	.000
16	.110	.052	81.764	16	.000

a. The underlying process assumed is independence (white noise).
b. Based on the asymptotic chi-square approximation.

Figure 7 : Data From The Time Series Prediction (AR(1) , d=1)

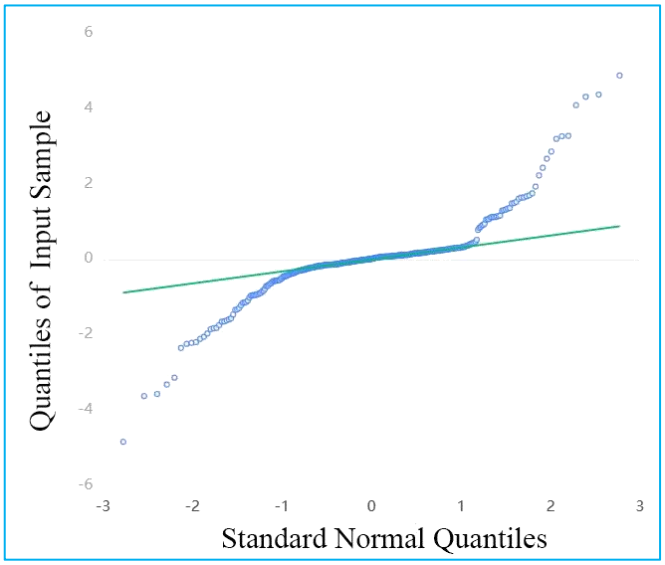

Figure 8 : Q-Q Diagram Of Residual Error

From the Figure 8, as the SNQ(Standard Normal Quantiles) increases from 2 to 3, QIS(Quantiles of Input Sample) increases rapidly. It illustrates great uncertainty.

5.4 Results in ARIMA Prediction And Analysis

Following the above theory, we apply ARIMA with parameter vector (p,d,q)=(1,1,0) to predict the number of the reported results after 120 days. The first part of the problem has been solved, and our result of interval prediction is shown as the following figure.

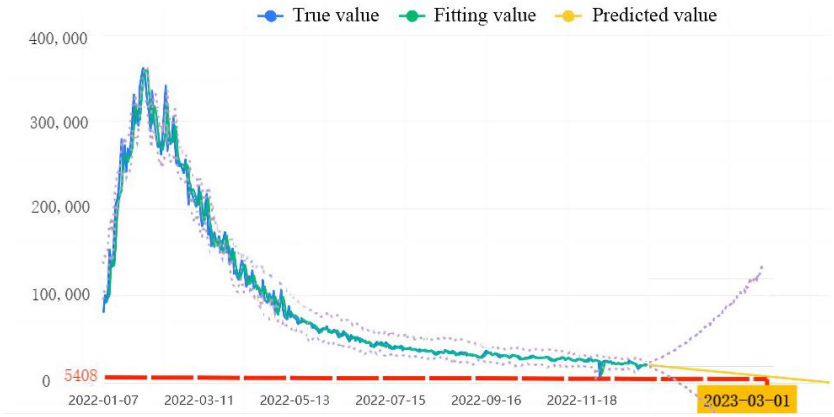

Figure 9 : Predicted Number Of The Reported Results With Predicted Interval

It can be seen from the figure that on 1,March,2023 the predicted number of the reported results equals to 5,408 and the predicted interval is between 0 and 670101. Furthermore, with time going by, the number of the reported results is predicted to maintain a flat downward trend.

6 Gradient Descent Optimization Model

6.1 Inspiration From the Natural Language Processing

The work of the predecessors has given us some inspiration.Enlightened by **Natural Language Processing** and **Naive Bayes Mode，**which are very commonly used to understand and analyze dialogues, we develop a similar model to quantify the elements that influence the distribution of trying times.

As the problem has not clarified what the attributes of the words refer to, we can transform the question into a more intuitive way: Does the constitution of one word have a strong correlation with the difficulty of guessing it out?

6.2 Feature Matrix and Output Matrix

In order to simplify the analysis, we make the hypothesis that once we get a five-letter word and know each number of the letters, the single word is consequently determined. For example, for the word “shape”, once we got one “s”,one “e”, one “p”, one “a” and one”h” in total, we are very sure to say that the world is “shape”, regardless of the sequence.

It’s very important to quantify all the letters in a word, so for a single word , we create a vector(1 row, 26 columns) to record all the features of the word, numbers,sequence, and position.1 row represents it is one word, and one sample as well. 26 columns represents the number of words in the Alphabet, from “a” to “z”.The initial values for all the words are zero, representing not a letter at certain position.For each letter, we add 1 at the same index the letter and vector element share, for example a word has an “a”, we add “1” at the first column, an “z”, add “1” at the 26th column[7]. The vector is defined as follow:

$$[x_1 \quad x_2 \quad x_3 \cdots x_{25} \quad x_{26}]$$

So for all the 359 samples, we design 359 vectors as above, and here comes a matrix **X** with 359 rows and 26 columns, which is called our **Feature Matrix.** Meanwhile, we can create a matrix with 359 rows and 7 columns **Y**. We call it an **Output Matrix,**meaning that for all the 26 same-calss features it has an impact on 7 same-class features ,which is our Output distribution of trying times.

6.3 Fit Linear Regression Model

We can tell there are relativity and connections between the two matrix, here I'd like to quantify the Relativity of each letter do to the distribution of trying times, by means of using Normal Equation in **Linear Regression**.

$$\mathrm{f}(\boldsymbol{x}) = \boldsymbol{\omega}^{\mathbf{T}}\boldsymbol{x} + \mathrm{b} \tag{3}$$

For hard mode , the distribution of trying times that are less than or equal to 4 might drop, percentage of more trying times will go up. For anyone who plays hard mode, we will make the dot product of it with a vector. We assume it's a Chi-Squre Distribution,k=2, at a reversed order, that is, for a bigger trying time, the percentage will multiply a larger number,for a smaller one, the percentage times a smaller number.

$$(y_1 \quad y_2 \quad \cdots \quad y_6 \quad y_7) \cdot (\xi_1 \quad \xi_2 \quad \cdots \quad \xi_6 \quad \xi_7) = \left(y_1^{'} \quad y_2^{'} \quad \cdots \quad y_6^{'} \quad y_7^{'}\right) \tag{4}$$

$$\widehat{\omega}^{*} = (X^{T}X)^{-1}X^{T}y \tag{5}$$

The formula(666) represents single variable, and here we apply this into a matrix, so as we have to count constant **b**, we have to extend the Feature Matrix to below:

$$X = \begin{pmatrix} x_{1\,1} & x_{1\,2} & \cdots & x_{1\,\mathrm{n}} & 1 \\ x_{2\,1} & x_{2\,2} & & x_{2\,\mathrm{n}} & 1 \\ \vdots & & \ddots & \vdots & \\ x_{\mathrm{m}\,1} & x_{\mathrm{m}\,2} & \cdots & x_{\mathrm{m}\,\mathrm{n}} & 1 \end{pmatrix} \tag{6}$$

Where the last column is combined with ones, so that **b** is now taken into consideration when we multiply two matrices. And the whole formula in matrix format is like this:

$$\mathrm{f}(\boldsymbol{x}) = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_{\mathrm{n}} x_{\mathrm{n}} + \mathrm{b} \tag{7}$$

Here we got our **Correlation Matrix**, the shape of which is 27 rows and 7 columns, representing the correlation between each letter and the distributions:

$$\begin{bmatrix} r_{1\,1} & r_{1\,2} & \cdots & r_{1\,6} & r_{1\,7} \\ r_{2\,1} & r_{2\,2} & & r_{2\,6} & r_{2\,7} \\ \vdots & & \ddots & \vdots & \\ r_{26\,1} & r_{26\,2} & \cdots & r_{26\,6} & r_{26\,7} \\ r_{27\,1} & r_{27\,2} & & r_{27\,6} & r_{27\,7} \end{bmatrix} \tag{8}$$

For row 1 to 26, it represents the relationship between each letter and the distribution of trying times, for row 27, it is a basic constant to improve the precision of the model. Now we plot the scatter diagram as follows:

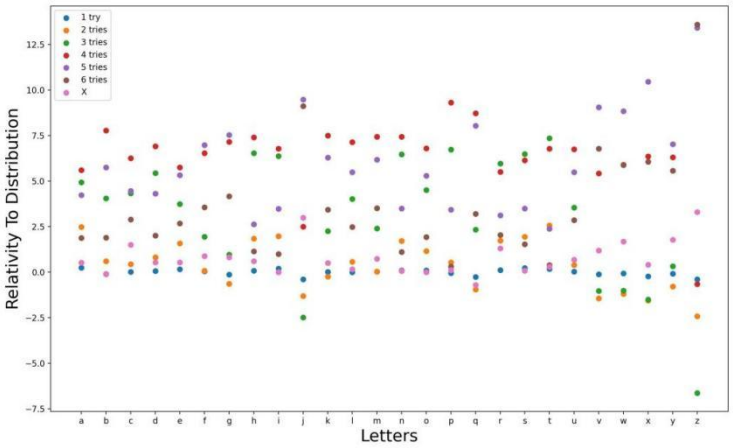

Figure 10 : Relativity Between Letters And Trying Distribution

From the figure: different letters surely will affect the percentage of scores, and furthermore, different words will have an influence as well.

7 Associated Percentage Prediction Model

In this section, we build a model, which is divided into date factor and words factor, to predict the distribution of the percentage. Compared to the interval prediction model, we add the factor of words to be considered. As analysis of the first part of the model is similar to the previous analysis interval prediction, what we discuss will focus on the factor of words.

7.1 Date Factor

Due to the similarity to model 1 analysis process and the page constraints, the intermediate process is a little complicated and will not be redundantly given in this section. The results which only considers the date factor are displayed directly as figure.

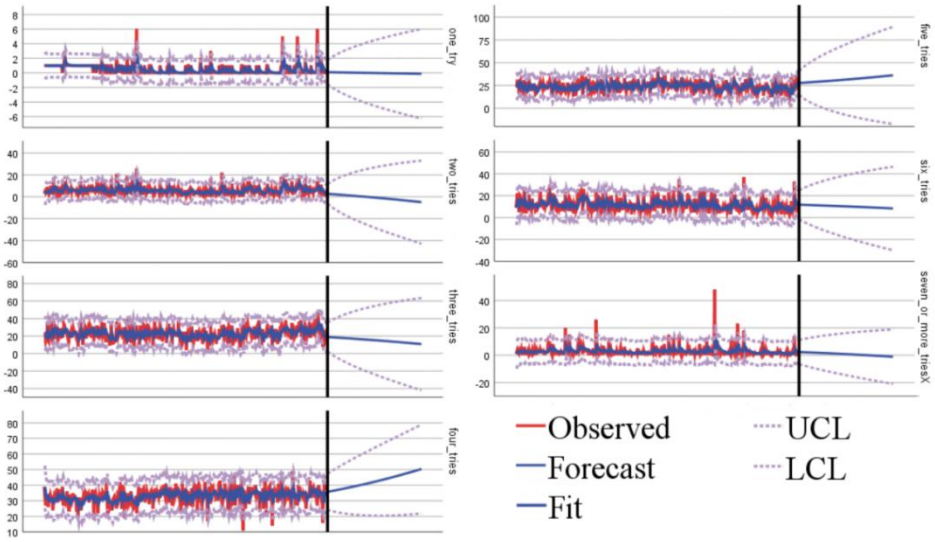

Figure 11:Prediction And Fitting Graph Of Trying Attribution

7.2 Word Factor

Considering that the impact of words on the results is difficult to be analyzed comprehensively, we make some simplifications to the problem:

From the given dataset, we find the solution word “mummy”. The percentage scores of “mummy” is provided and the solution word “mummy” is similar to the word “eerie”, which constitutes of three same letters. Therefore, we make some definitions additional assumptions below to simplify analysis for the question.

Definitions:

- Define k(i) as the ratio of the value of i try (on the day when the answer is MUMMY) to the average value of i try (from January 7 to December 31, 2022).
- Define g(j) as the ratio of the predicted value of j try (on the day when the answer is EERIE) to the predicted actual value of j try (on March 1).

Assumptions:

- **Hypothesis 1: Assume the impact of the given word “eerie” is almost closely to the impact of the word “mummy”.**

Explanation: The given word “EERIE” is similar to the word “MUMMY”, which constitutes of three same words. Additionally, the frequency of the both words is also close. Therefore, the predicted result will not be badly affected. As the percentage scores of “MUMMY” is provided , the analysis of the problem can be better simplified.

- **Hypothesis 2: Assume that the value of k(i) approximately equals to the value of g(j).**

Explanation: As the value of i try(from Jan.7 to Dec.31) is averaged, it can represent the intermediate level of the scores to some degree. So the k(i) reflects the level relative to the average. Based on the hypothesis (1), we can establish a connection between k(i) and g(j) and when k(i) equals to g(j), we can get the percentage scores of “EERIE” more accessibly.

Based on the analysis above, we have developed a model which considers both date

factor and word factor. And the prediction of percentage scores for the word “EERIE” on 1,March,2023 is shown below, as figure.

(series 1: the predicted percentage scores result which only considers date factor)

(series 2: the predicted percentage scores result which consider date factor and word factor)

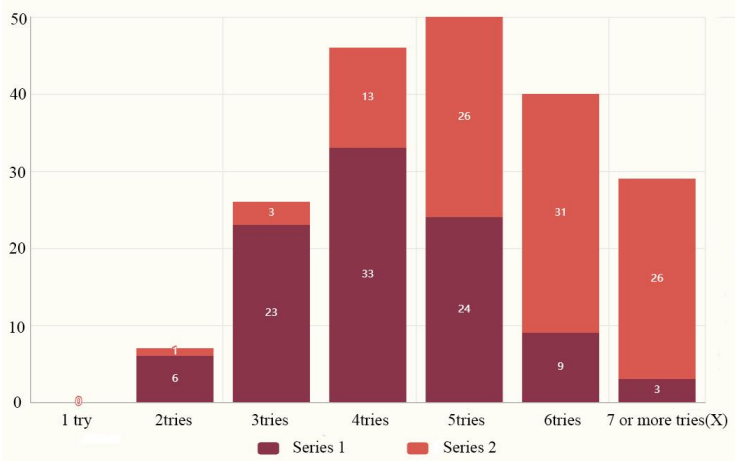

Figure 12 : "EERIE" Percentage Forecast

From Figure 13,it can be seen that:

After considering the word factor, the percentage scores of 2,3,4 decreases, while the percentage scores of 5,6,7 or more tries(X) increases.

Uncertainties:

- **The impact of the given word “EERIE” is not the exactly same with that of the word “MUMMY”.**
- **Directly assuming the value of k(i) equals to the value of g(j) will also cause error.**

With data factor and word factor both scrupulously considered and analyzed, our model is relatively refined. However, we think although the hypothesis (1) greatly simplified the problem, it will result in a certain amount of error which might make the results inaccurate. Therefore, to enhance the accuracy of result and extend the applicability of our model, the following task is carried out.

8 Words Difficulty Classification Model

8.1 Model Precision

Since the first question has already discussed the correlation between letters and difficulty, we will further classify solution word by difficulty. In this section, we split the

words into letters and score all the letters in order to get a concrete "Grade" to measure the difficulty of the solution words.

To ensure the high precision of the model, we will test the accuracy first.

As we know that the difference between the real value and the prediction value is called error, or cost in Machine Learning.But for each sample, the average cost represent the average bias between prediction and reality. Here is the **Average Cost Function**[8]**:**

$$\mathbf{C} = \frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}} \left(\mathrm{h}\left(x^{(\mathrm{i})}\right) - \mathrm{y}^{(\mathrm{i})}\right)^2 \tag{9}$$

By summing up all costs of samples and divide the sample number "m", we got the average cost, or average square error between model and reality. But in order to calculate the accuracy, we have to compare it with the scale of square sample value by dividing the two functions, which is the **Error Rate Function**:

$$\mathbf{R} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \left(\mathrm{h}\left(x_{\mathrm{j}}^{(\mathrm{i})}\right) - \mathrm{y}_{\mathrm{j}}^{(\mathrm{i})}\right)^2}{\sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \left(\mathrm{y}_{\mathrm{j}}^{(\mathrm{i})}\right)^2} \tag{10}$$

Obviously,we know an answer is either right or wrong, so for the accuracy "$\boldsymbol{\alpha}$"

$$\boldsymbol{\alpha}=1-R$$

According the formulas above, We divide all 359 samples into 251 training data, 54 test data, 54 validation data and use the model to predict the accuracy, which is 0.947, very satisfying for this model.We can easily predict α of our model, which is 0.947, very satisfying. With the great accuracy of our model , we can create a grading system based on the correlation we get from gradient descent.

8.2 Grading system

Here comes our "Easiness Score" grading formula:

$$\boldsymbol{S_I} = \sum_{n=1}^{7} \left((8-n)R_n\right) \tag{11}$$

For all letters, we give the highest weight to trying successfully in one time, and for the following results, we sort them at a descending order . Although for the first two tries, the results probably reflects more luck than easiness, since the percentage is very low and evenly spread, it will not make a big difference to the results.

This model is to highlight the “easiness” of all letters, that is to easy, easier a letter is , more score it will get.At the same time, I’d like to carry out my hypothesis: as we have 359 samples with no bias, the number of letters should be presented in an abundant number, which means our prediction is not influenced by too less data for some specific letter.

8.3 Validation

Further, people are reluctant to try letters repeatably, but for 5 letter words, the space for variable changes is not that big,once some of the letters turns green or yellow, people will be willing to try repetitive letters for their idea. So the multiplicity doesn’t make huge influence, since a word with three repetitive letters is very rare.

By adding all five letters in one word, we can get the “Easiness Score” of it.

Next, we calculate all 359 words in the data set, and set up a classification of three labels: “Easy”,”Intermediate”,”Hard”. Most of words will lie in “Intermediate”, only the extremely easy and hard ones will be sorted out.

Based on the above, the words difficulty classification model has been established, refined and perfected. Affected by the limited space, all the calculation results will not be released here. The “Easiness Score” distribution of all 359 words will be displayed through the following visual drawing:

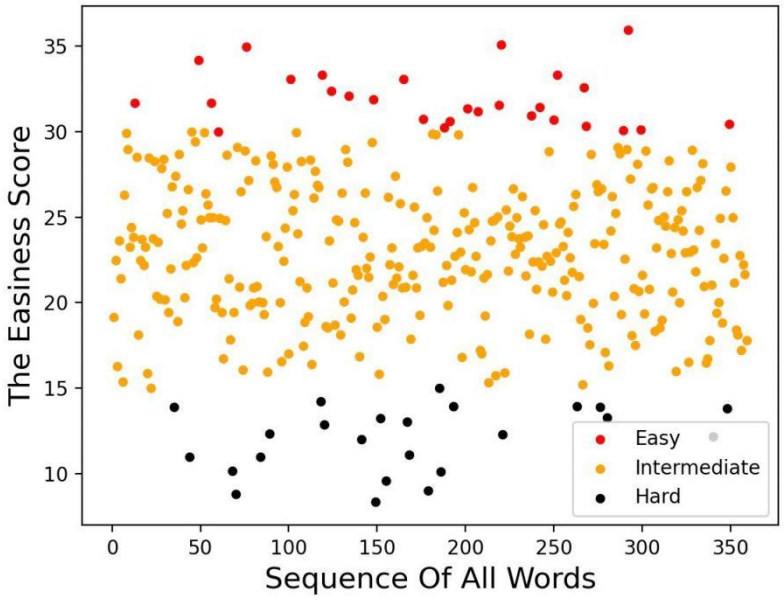

Figure 13 :Classification Of All Samples

As for the word “EERIE”,it’s a quite commonly used word. Meanwhile，“E”, ”R”, and “I” are very frequently used, therefore the given score is very high,it should be classified to “**Easy**”,and if all three or two letters are tried once, it’ll be very easy to guess it out.

However, we all know that for 5 letters, it’s very rare to think of “E” can take up 60% of all position, making it a little harder to imagine. As such unique feature we didn’t previously think about, we should validate our result using a previous model:

$$(x_1 \quad x_2 \quad x_3 \cdots x_{26} \quad x_{27}) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{16} & r_{17} \\ r_{21} & r_{22} & & r_{26} & r_{27} \\ \vdots & & \ddots & \vdots & \\ r_{261} & r_{262} & \cdots & r_{266} & r_{267} \\ r_{271} & r_{272} & & r_{276} & r_{277} \end{pmatrix} = (y_1 \quad y_2 \quad y_3 \cdots y_6 \quad y_7) \tag{12}$$

We got the distribution of “EERIE”,and we choose a “Easy” sample, putting them into the **Correlation Matrix**[9]**,** and I got prediction for two output vectorsY1,Y2, indicating their own distributes of trying times, and take Y1-Y2(the unit of which is percentage):

$$\boldsymbol{\Delta Y}=(0.4 \quad 2.9 \quad -1.1 \quad -4.4 \quad -0.2 \quad 1.14 \quad 1.0)$$

From the results we clearly find out that “EERIE” is more likely to be guessed out in early times, and more likely to be guessed out in late times. Therefore, it is less likely to be guessed out in average time. But the difference of each feature is quite small, their features are very similar too.

In conclusion, the **dispersion degree** of “ERRIE”is lower, but it’s still an “**Easy**” word.

9 Additional interesting features of the Dataset

9.1 Relativity of Number in hard mode and Reported Results

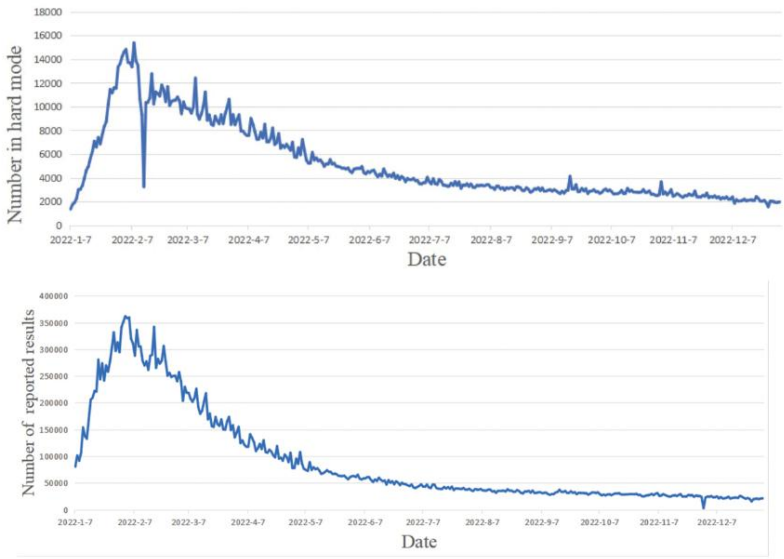

Figure 14 :Image Of Number In Hard Mode And Reported Results

From the figure shows, the curve 1 is similar with the curve 2. In order to dive deeper

into the connectivity, we create a **Neuron Network** model, which contains 20 Hidden layers to help improve the accuracy of the model, meanwhile, we divide the total 359 samples into 251 for training set, 54 for test set and 54 for validation set.And we train the model with **Levenberg-Marquardt** algorithm, the results are as follows:

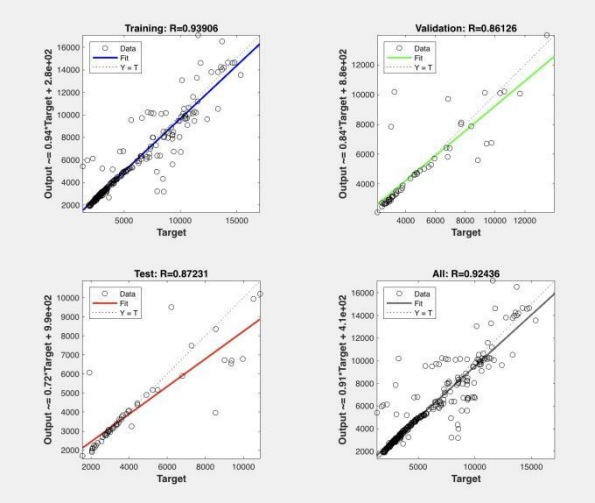

Figure 15 :Relativity Of Numbers In Hard Mode And Reported Result

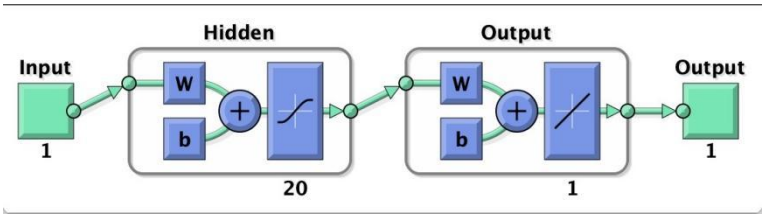

Figure 16 : Structure Of The Neuron Network

According to the figure, the number of the reported results and the number in hard mode have strong connection. We find out that people who can guess the solution word out in hard mode, tend to be willing to post their results on the media. We guess the reason is that passing the barrier of hard mode can boost a sense of pride which makes people tend to post the results online.

9.2 Informative Words

Through calculation, we find out that the try of different words will show different amount of information. Guessing the word “TREAT” can more likely help players enhance the possibility of solving the puzzle. Due to the limited pages, we directly give the feature in the following Figure 17:

Figure 17 : Visualization of Informative Words

10 Sensitivity Analysis

As we know that our input has 26 features, so generally 359 samples might not be that abundant, which means for the correlation matrix, the prediction result can be not comprehensive, and not precise as well.

So we take 180 samples at random,that is half of the whole and all 359 samples, for each we calculate a correlation matrix **C1,C2**,and use Matlab to calculate the Pearson Correlation, here is our result (The leading diagonal represents the relativity of distributions between two correlation matrices.) :

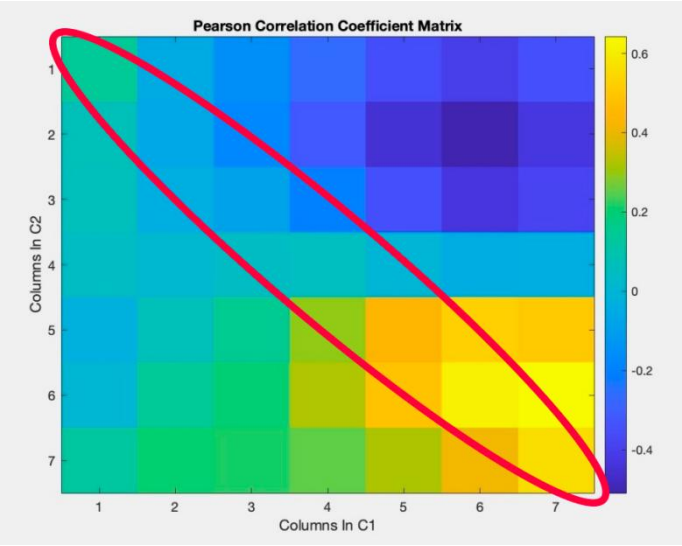

Figure 18 :The Relativity Of Two Correlation Matrices

From the figure we can tell that the relation between C1 and C2 is quite small. From our point of view, there are much randomness since for 27 features, it's a very huge scale that might need more samples to help fit the model to a perfection.

11 Strengths and Weaknesses

Strengths:

- ✧ The visualization work is done very well by us, such as choosing the fittest algorithm, Q-Q Diagram of Residual Error, relativity between letters and try distribution, the informative words demonstration, the relativity of two correlation matrices. Boring data may be able to reflect the law, but not as intuitive as varieties of images.
- ✧ The interval prediction based on ARIMA model is highly scientific and reasonable. Not only do we verify the unit root test and white noise test, but also select the strongly predictive time series.
- ✧ For the "Grading System", we uses gradient descent to calculate the correlationship between inputs and outputs. It is very explicit and has a good comprehensibility.Unlike neuron network, we can check pilot process with ease.
- ✧ The ARIMA model can be applied to a wide range of different time series data, both seasonal and non-seasonal, and therefore has wide applicability.Meanwhile, The autoregressive and sliding average terms in the ARIMA model can be used to analyze trends, periodicity, and noise components in time series data, making the results more intuitive and explainable. Therefore, Our model effectively finishes all of the tasks with great applicability and accuracy.

Weaknesses:

- ● ARIMA model requires data to meet certain stationarity and correlation conditions, otherwise, the model performance will be affected.
- ● The start and end times of each period have a large impact on the parameter update, therefore the model will be difference when the start or end time changes.

12 Letter To Puzzle Editor

Dear Puzzle Editor:

We are honored to inform you our achievements after proceeding the data analysis and building the percentage scores prediction model based on words difficulty classification.

Our team conducted a modeling analysis of the interactions between number of reported results and date (from 7.Jan.2022 to 31.Dec.2023). Based on the data provided, we establish a ARIMA model for predicting the future reported results. Meanwhile, the gradient descent optimization model based on natural language processing is built. In this section, we split the words into letters. Therefore, the type and each number of the letters are regarded as the attributes of the words. Furthermore, based on the previous analysis, the words difficulty classification model is established to reveal the correlation between letters and difficulty. With the model we develop, we obtain the following valuable results from two prospectives:

From the macroscopic prospective, we build a number of reported results prediction model, which can predict the percentage scores(1,2,3,4,5,6,X) for a specific future date and a given word. With the verification of the unit root test and white noise test, our model demonstrates great stability and accuracy. Furthermore, as the time goes by, our model can still be applied with high scalability.

From the microcosmic prospective, we create a “grading system” for all sample words, in order to quantify the difficulty. As long as we got a 5-letter word, we can predict the “score” of its easiness, finally come up with a grade for it:”Hard”,”Intermediate”,or “Easy”. Furthermore, constitution of words not only influences the difficulty, it also carries weight on the distribution of trying times, for a word that might contains more commonly used, repetitive letters, the distribution is likely to be more evenly spread; however,for those words that contains infrequent and repetitive letter, the distribution of trying times will gather at multiple times. By comparing predicted data and real data, we are overjoyed to see the accuracy of which has reached 95%, the prediction has reached a reliable level.

Here we list an interesting figure, which displays that the try of different words can show different amount of information. The larger the word is, the more informative will the try be.

Finally, we hope that our model is enlightening and sincerely hope that Wordle will be attracting more players in the future!

Yours sincerely,
Team # 2307787

References:

[1]"What is Wordle and how to play — everything you need to know ", Tom's Guide (www.tomsguide.com),6,December,2022

[2]Tomas Mikolov,"Efficient Estimation of Word Representations in Vector Space",Conference on Neural Information Processing Systems (NIPS),2013

[3] Rong Xioang, "word2vec Parameter Learning Explained",Computer Science Technical Reports In Standford University,2014

[4]Peter Harriton,Machine Learning In Action,2016

[5] Mike Crowson,ARIMA modeling in SPSS: model identification, ARIMA modeling in SPSS:model identification,YouTube,2019

[6] Mike Crowson,ARIMA modeling in SPSS: Estimation and diagnosis ARIMA modeling in SPSS: Estimation and diagnosis,YouTube,2019

[7] Peter Harriton,"Training Algorithm,Calculate Probabilities From A Word Vector",Machine Learning In Practise,pp60-62,2016

[8] Andew Ng,"Cost Fuction",Introduction to Machine Learning, 2016, https://www.bilibili.com/video/BV164411b7dx/?spm_id_from=333.337.search-card.all.click&vd_source=71201e283a16a266a47aed8a294b6db0

[9] Zhihua Zhou,"Linear Regression",Machine Learning,pp55,2016

Appendices

Appendix 1

Introduce: “Easiness Scores” Of Each Letter (Higher, Easier)

A:6.30716392	F: 0.92101053	K: 0.64361923	P: 2.38043935	U:2.31878389
B:1.50773264	G:1.81924063	L: 3.91690349	Q: 0.13131923	V: 0.4928043
C:2.8144034	H:1.95370748	M:1.85672123	R: 5.69085606	W:0.46386411
D:2.9563529	I: 7.2937834	N: 6.19387965	S: 7.321409512	X:0.1380089
E:8.1890507	J: 0.07737525	O: 4.71747741	T: 5.9579784	Y:0.89281001
Z: 0.07837438				

Appendix 2

Python Code

```
def convert_to_numbers_matrix(letters_list):

        stripped_letters_list = [list(map(str.strip, row)) for r
ow in letters_list]

        numbers_matrix = [[0 for _ in range(len(alphabet))] fo
r _ in range(len(stripped_letters_list))]

        for i in range(len(stripped_letters_list)):
                for j in range(len(stripped_letters_list[0])):
                        letter = stripped_letters_list[i][j]
                        if letter in alpha_dict:
                                index = alpha_dict[letter]
                                numbers_matrix[i][index] += 1
        return numbers_matrix
```