Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2322441

Combined Multi-model Predictive Analytics Model

Summary

Wordle is a puzzle game launched by The New York Times, which has received enthusiastic attention from players at home and abroad since its launch. After a long time of players' attempts, a large amount of background-related data has been generated, so it is of great significance and value to conduct data mining and analysis. In this paper, we focus on, the analysis of the change in the number of reported results, the prediction of the percentage distribution of the number of guesses for a word under any attribute and the determination of which difficulty category the word EERIE belongs to.**A time series analysis forecasting model, a comparison model with mean ratios of random samples,a random forest forecasting model based on historical statistics,and a comprehensive model combining mean cluster analysis, coefficient correction equations, and support vector machines were developed,** Solving for parameters via Python programming.

For problem 1, first, we built a time series analysis prediction model based on the data after pre-processing. By building the model and solving it using Python programming, we first obtained the results of the change in the number of reported results from the data visualization, and found that the number reached the highest value in February and started to decrease thereafter, which we judged might be related to the freshness of players. Second, we obtained the prediction interval for the number of reported results on March 1, 2023 by adjusting the parameters**(8312, 8684)**.Finally, we compared the overall mean with the sample mean using a comparison model with a random sample of mean ratios, yielding the percentage of any attribute of the word that would affect the score played in the difficult mode.

For problem 2, we build a random forest prediction model based on historical statistics. First, we build a first-order historical statistical analysis model and a second-order historical statistical analysis model, and get an important conclusion that word difficulty has a certain relationship with time. Secondly, we used the random forest regression algorithm to predict the percentage of EERIE related to word difficulty in the time period from the end of 2022 to March 1, 2023, when the word difficulty is likely to remain easier. Finally, the percentages for each count were obtained by programming the solution in Python and were**0.53, 7.43, 26.7, 33.6, 19.67, 6.87, and 1.9,**respectively.

For problem 3, we developed a comprehensive stepwise prediction model based on problem 2 with a combination of mean cluster analysis, coefficient correction equation, and support vector machine. First, we classified **five difficulty categories** by mean cluster analysis. Secondly, we found an interesting phenomenon that the proportion of words whose data contained both e and a was high, with a total of 57 occurrences, accounting for 16%, so with and without e and a we used coefficient correction. Then, we used the support vector machine algorithm to predict the classification of the word EERIE and concluded that the word belongs to **the super simple class**.Finally, a sensitivity analysis was performed and all of our forecasts were within a clearly reasonable range.

Keywords: **Time Series Analysis** **Random forest regression prediction** **Means clustering analysis** **Support vector machines**

Contents

1 Introduction

1.1 Background

Wordle is the popular word guessing game of 2022, with an original gameplay style that attracts and challenges people. Players have six chances to fill in a word made up of six letters and are given feedback for each guess. The game is divided into normal and hard modes, with normal mode requiring correct guesses but incorrect positions to be yellow, correct positions to be green and both incorrect to be grey, and hard mode requiring a correct guess (green or yellow) to be used on the next attempt. If all six attempts are incorrect, the challenge fails.

Many users reported their scores on Twitter, whereby the MCM generated a daily results file for the period 7 January to 31 December 2022 (see Annex 1).

1.2 Requirements

The New York Times asked us to analyse the data in Attachment 1to address the following questions.

1. Develop a model to explain the changes in the number of reported results and use your model to create a prediction interval for the number of reported results on March 1, 2023. Explore whether any attributes of the word affect the percentage of scores reported that were played in Hard Mode. If it will affect, how to deal with? If it does not affect, explain why.

2. For a given future solution word on a future date, develop a model that allows you to predict the distribution of the reported results and use the model to predict the percentages associated of (1, 2, 3, 4, 5, 6, X) for the word EERIE on March 1, 2023. Discuss what uncertainties there are in the model and predictions.

3. Develop and summarize a model to classify solution words by difficulty. Identify the attributes of a given word that are associated with each classification. Using our model to assess how difficult the word EERIE is. Discuss the accuracy of our classification model.

4. List and describe some other interesting features of this data set.

Finally, write a one-to-two page letter to summarize our results to the Puzzle Editor of the New York Times.

1.3 Our Works

- First, we preprocess the data set to check the validity of the data.

- For problem 1, we build a time series analysis model, visually analyze the selected data through Tableau, and predict the number of EEIRE reported results on March 1, 2023.

- For problem 2, we establish a second-order historical statistical model and a random forest regression prediction model, and take EERIE as an example to make predictions.

- For problem 3, we establish a mean cluster analysis model, a coefficient correction model and a support vector machine prediction model, combine them into a comprehensive stepwise prediction model, divide all words into five categories according to difficulty, and then obtain the category to which EEIRE belongs.
- Finally, we are looking for other interesting features of the dataset.

Here we use a flowchart to express our ideas more intuitively.

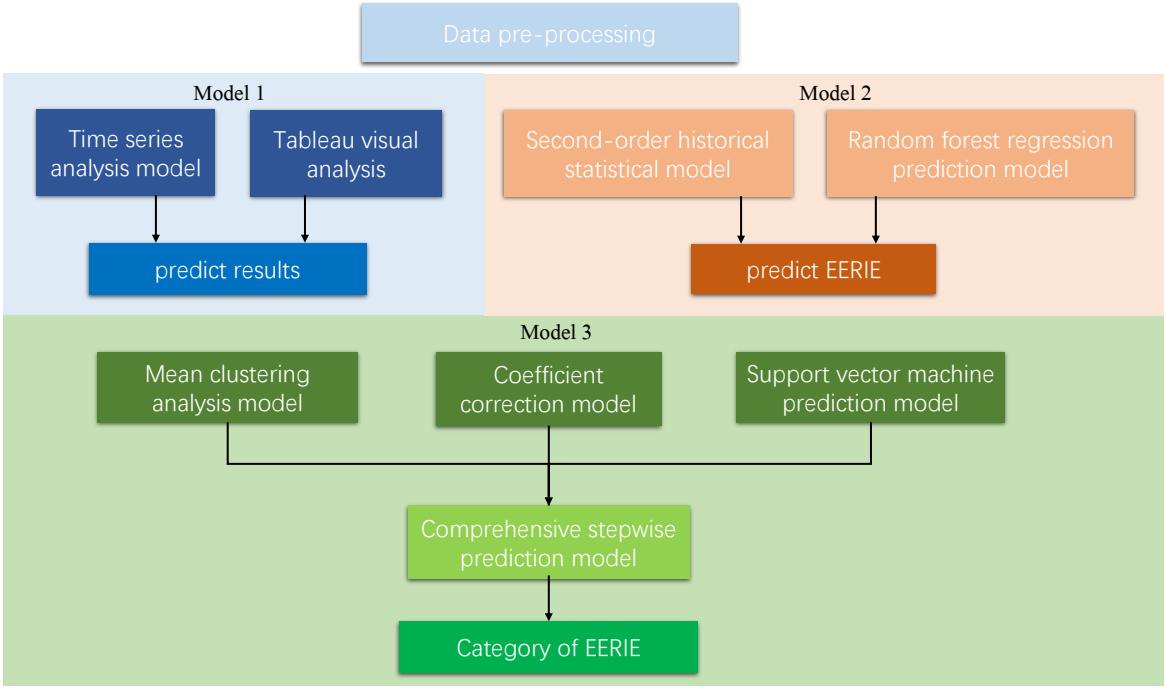

Figure 1: Flow Chart of Our Work

2 Assumptions and Description of Notations

2.1 Assumptions

1. It is assumed that the attached data is true and valid after rounding and has not been modified.
2. We assume that Wordle Games does not make changes to the game rules or game thresholds until March 1, 2023.

2.2 Notations

Table 1: Description of notations

Notations	description	Unit
$S_t^{(1)}$	Smoothed value of the primary index	\
$S_t^{(2)}$	Smoothed value of the secondary index	\
Y	Average number of attempts	\
A_i	Historical mean column matrix	\
H_i	First-order historical statistical analysis model matrix	\
HE_i	Second-order historical statistical analysis model matrix	\

3 Data Pre-processing

According to the dataset (Problem_C_Data_Wordle) in Annex 1, there are six broad categories of data, including Date, Contest number, Word, Number of reported results, Number in hard mode, Percent in (1 try, 2 tries, 3 tries, 4 tries, 5 tries, 6 tries, 7 or more tries (X)). Preliminary judgment is that there are 359 data for each category. Overall, the amount of data is not very large. The above is the basic situation of the original dataset.

First, to check the data health of the dataset, we used Python programming to retrieve the data for missing, duplicates, etc. The result of the search was that the dataset did not have any missing, duplicates, etc., indicating that the dataset was complete and valid. Two anomalies were found, namely on 13 February and 30 November. 13 February showed a large number of reports but very few people choosing the difficult mode, while 30 November showed a small number of reports and a small difference in the number of people choosing the difficult mode. Eventually we made the decision to replace it both based on the stability of the numbers around the time to make a reasonable substitution. According to the rules of the game, the required number of letters is five, but we found words with a four-letter count, such as the CLEN word for Contest number 525, so we eliminated it. To ensure that there must not be any duplicate words, we used the word cloud programming technique again for visual analysis of the data to check once again, and the graph below shows that the words are all the same size, indicating that there are no duplicate words. This was done to ensure that the data set we used was healthy and valid.

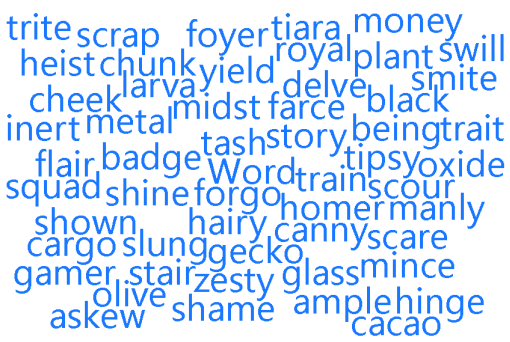

Figure 2: Wordcloud Map

Next part, we subjected Number of reported results, Number in hard mode and Percent in i tries (i=1,2,3,4,5,6,7) to correlation analysis using SPSS software and found that they were all related, ensuring the accuracy of the predicted results in the later section. Here is their heat map:

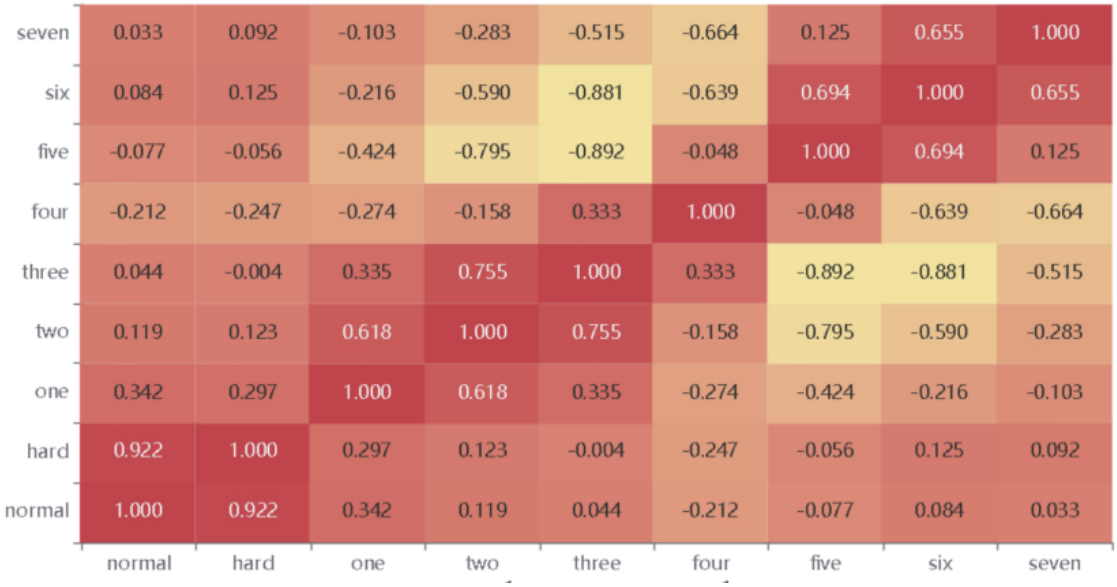

Figure 3: Heap Map

Then, according to the rules of the game, it takes seven or more times before it is judged to be already part of a secondary game or multiple games, which we designate as X tries.

In addition, data processing may continue on a case-by-case basis, as we will explain further.

4 Modeling and Solving Problem 1

The question asks for a model to explain why and how the number of reported outcomes changes from day to day based on the attached data. And use the model to create a prediction interval for the number of results reported on March 1, 2023. In addition, we need to explore whether any attributes of the words affect the percentage of scores played in hard mode.

4.1 Time Series Analysis Models

The individual data times for the dataset are continuous from 7 January 2022 to 27 December 2022, so it is reasonable to use a time series analysis model for analysis and forecasting.

Time series analysis is a method used to analyse and forecast time series data. Time series data is data collected in a chronological order, such as stock prices, temperature changes, sales data, etc. It usually shows certain trends, cycles, seasonality and randomness. Time series analysis can help us to understand the patterns and trends in these data and to predict future trends. Now, we will build a time series analysis model step by step.

The calculation and derivation process is as follows.

Step 1 : Create a quadratic smoothed exponential equation
Exponential smoothing is a special type of weighted moving average method.

$$\begin{cases} S_t^{(1)} = \alpha y_t + (1-\alpha) S_{t-1}^{(1)} \\ S_t^{(2)} = \alpha S_t^{(1)} + (1-\alpha) S_{t-1}^{(2)} \end{cases} \tag{1}$$

where $S_t^{(1)}$ is the smoothed value of the primary index and $S_t^{(2)}$ is the smoothed value of the secondary index.

Step 2 : Build a linear trend model
When the time series y_t has a linear trend from a certain period, a linear trend model can be built

$$\hat{y}_{t+m} = a_t + b_t m, \quad m = 1, 2, \cdots \tag{2}$$

$$\begin{cases} a_t = 2S_t^{(1)} - S_t^{(2)} \\ b_t = \frac{\alpha}{1-\alpha}\left(S_t^{(1)} - S_t^{(2)}\right) \end{cases} \tag{3}$$

Let m = 1 , which gives

$$\hat{y}_{t+1} = 2S_t^{(1)} - S_t^{(2)} + \frac{\alpha}{1-\alpha}\left(S_t^{(1)} - S_t^{(2)}\right) \tag{4}$$

Step 3 : Perform differential operations
Difference operation is a very convenient and effective method for the deterministic information extraction.

$$\nabla^d X_t = (1-B)^d X_t = \sum_{i=0}^{d} (-1)^i C_d^i X_{t-i} \tag{5}$$

Step 4 : Establish the ARIMA model
The smooth series obtained from the difference operation can be fitted with an ARIMA model.

$$\begin{cases} \phi(B)\nabla^d X_t = \theta(B)\varepsilon_t \\ E(\varepsilon_t) = 0, \quad \mathrm{Var}(\varepsilon_t) = \sigma_\varepsilon^2, \quad E(\varepsilon_t \varepsilon_s) = 0, \quad s \neq t \\ E(X_s \varepsilon_t) = 0, \quad \forall s < t \end{cases} \tag{6}$$

4.2 Visual Analysis of Changes in the Number of Reported Results

Although the annexed data is small, we can initially know whether the numbers are increasing or decreasing on a daily basis. However, in order to have a more technical analysis and to be able to present it visually, this paper visualises the data using the time series analysis model that has just been developed for the full number of reported results. This is shown in the figure.

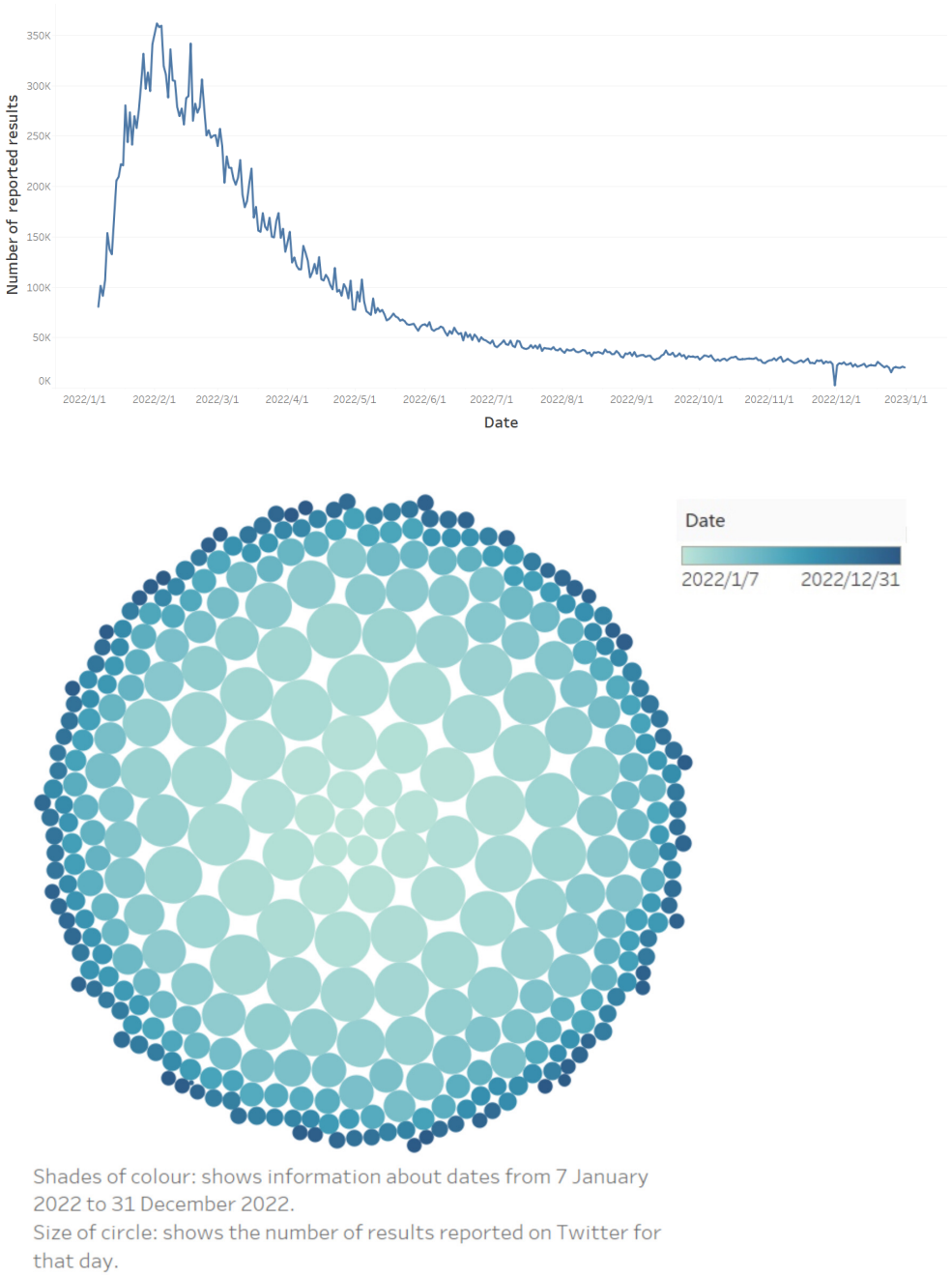

Figure 4: Visualisation of Changes in the Number of Reported Results

Data from Date and Number of reported results in the dataset in Attachment 1 were selected for visual analysis by Tableau.

In terms of trends, the number of reported results on Twitter grew rapidly from January 7, when the original gameplay of the Wordle guessing game sparked curiosity and avid love from the public. After a month of publicity, the number of reports peaked on 2 February and has since fallen back, with the number of reported results stabilising between 25,000 and 35,000 throughout February. From the start of March until December, the number of reports on Twitter decreased as the novelty of the game wore off for users.

In addition, the exact number of changes in the trend also fluctuates due to differences in the difficulty of the solution words.

4.3 Prediction Interval

This forecast for 2023 will use the time series analysis model we built. In order to confirm the accurate prediction interval, this article will make multiple predictions by modifying Python code parameters.

4.3.1 Parameter Description

Here we continue to refine our time series analysis model. We introduce two parameters. One is the p-order autoregressive model (AR), whose partial autocorrelation function (PACF) should be zero after the p-order, saying that it has truncation; In addition, the autocorrelation function (ACF) cannot be zero after a certain step, but decays in the form of a sine wave, which is said to be tailed. The other is the Q-order Moving Average Model (MA), which is the opposite of AR.

4.3.2 Outcome

Prediction results obtained through Python programming:

A. Demonstration of autocorrelation and partial autocorrelation

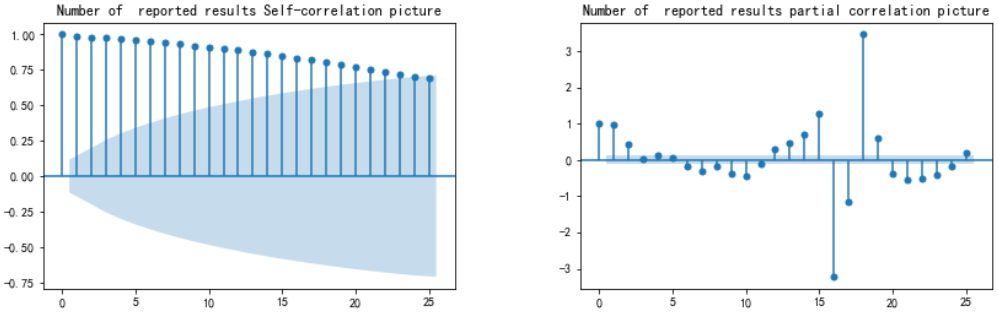

Figure 5: Demonstration of Autocorrelation and Partial Autocorrelation

B. Presentation of forecast results

Table 2: Presentation of Forecast Results

p	d	q	forecast results
2	1	0	8373
3	1	0	8312
4	1	0	8684

C. Forecast interval

The prediction interval derived by the model is (8312,8684).

4.4 A Comparative Model of Mean Ratios for Random Sampling

To address the question of whether any attribute of a word affects the percentage of scores played in hard mode, this paper develops a comparative model of the mean ratio of a random sample to solve the problem. Here we will define that model.

Step 1 : Establishing property indicators

For the characteristics of the words, we have selected an attribute indicator, which is the number of letters, and the presence or absence of repeating letters. Where the subscript 1 indicates the presence of repeating letters and the subscript 0 indicates the absence of repeating letters.

Step 2 : Conduct a random sample

Using Python programming to perform random sampling, the same number of words with and without duplicates will be taken separately as our sample.

Step 3 : Mean comparison

For a more accurate comparison of means, we first find the overall mean of the number of guesses for each of the original attachment data. Then the mean of our sample was calculated. Finally a comparison was made, where we considered that any attribute of the word would not affect the percentage of scores played in hard mode if the difference between the sample mean and the overall mean was within an acceptable range, and where the difference between the sample mean and the overall mean was within an unacceptable range, we considered that any attribute of the word would affect the percentage of scores played in hard mode.

4.4.1 Overall Mean

Table 3: Overall mean

1 try	2 tries	3 tries	4tries	5tries	6 tries	more tries(X)
0.47	5.54	22.73	32.93	23.64	11.56	2.8

Since the number of first, second and seven and more guesses is relatively small, using we decided to combine them and denote them by O.

Table 4: Results of the solution

O tries	3 tries	4 tries	5 tries	6 tries
8.81	22.73	32.93	23.64	11.56

4.4.2 Discussion

We first discuss the simple case, using the number of letters and the presence or absence of repeated letters as samples. Here are the results after solving

Table 5: Results of the solution

With repeated letters	Frequency(%)	O tries	3 tries	4 tries	5 tries	6 tries
No repeated letters	Frequency(%)	8.81	22.73	32.93	23.64	11.56

The resulting frequencies of the sample means are as follows.

Table 6: The resulting Frequencies of the Sample Means

		O tries	3 tries	4 tries	5 tries	6 tries
Sample Means	Frequency(%)	4.5	14	34.5	32.5	14.5

4.4.3 Analysis of Results

Ultimately, we compared the overall mean and the sample mean and found that their results differed somewhat. So we conclude that any attributes of the words may affect the percentage of scores played in hard mode.

5 Modeling and Solving Problem 2

Question 2 asks us to open up a model to predict the distribution of reported outcomes and to show what uncertainties there are in the model and the predictions. We were also asked to use the word EERIE as an example to predict the percentage of correlations for (1, 2, 3, 4, 5, 6, X) at a future date under a difficult model. We will use a combination of a historical statistical analysis model and a machine learning model to create a more accurate and intelligent hybrid model.

5.1 Historical Statistical Analysis Model

The historical statistical model is an efficient model based on mean analysis and rational statistical analysis based on patterns in the attached data. We found an important pattern in the data in that the number of reported results increased and then decreased, so we assumed that there had been some shift, perhaps in the difficulty of the words. We then de-staged the temporal pattern to build multiple models of historical statistical analysis for comparison.

5.1.1 First-order Historical Statistical Analysis Model

We have divided a significant time period according to the increasing and decreasing number of reported results. Ultimately, this paper chose a time period from 7 January to 30 April with a subscript of 1 and a time period from 1 May to 31 December with a subscript of 2.

• **Model Building**

The first-order historical statistical analysis model matrix is

$$H_i = [A_1 A_2 A_3 A_4 A_5 A_6 A_x], i = 1, 2. \tag{7}$$

Where A denotes the historical mean column matrix.

• **Model Solving and Presentation**

Table 7: Presentation of Results

	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	more tries(X)
H_1	0.8	6.3	22.7	31.8	23.3	12.6	3.1
H_2	0.3	5.6	22.7	33.7	23.8	11	2.7

• **Analysis of First-order Historical Statistical Analysis Model**

As can be seen from the results, there is a big difference between the first, fourth and sixth guesses. So we guess that the words guessed in the second stage became a little harder.

5.1.2 Second-order Historical Statistical Analysis Model

The second-order historical statistical analysis model is based on the time period from May 1st to December 31st. From the results of the data visualisation in question one, we find another important point in time, which is that the rate of decline before September is not the same as the rate of decline after September. It is clear that the rate of decline is faster for the former and slower for the latter. So we go on to divide these two phases and carry out a comparison of the historical statistical analysis models. We have a subscript of 1 for the period from 1 May to 30 August and a subscript of 2 for the period from 1 September to 31 December.

• **Model Building**

The second-order historical statistical analysis model matrix is

$$HE_i = [A_1 A_2 A_3 A_4 A_5 A_6 A_x], i = 1, 2. \tag{8}$$

where A denotes the historical mean column matrix.

• **Model Solving and Presentation**

Table 8: Presentation of Results

	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	more tries(X)
HE_1	0.28	5.1	21.4	33.6	25	12	2.6
HE_2	0.35	6.11	24.1	33.9	22.5	10	2.8

• **Analysis of Second-order Historical Statistical Analysis Model**

As can be seen from the results, the number of guesses has changed relatively significantly, except for the fourth and more times. So we guess that after September the difficulty of the words was adjusted and became relatively easier than before. Ultimately, we have come to the important conclusion that there is a relationship between time and word difficulty, meaning that word difficulty is likely to remain easy until 1 March 2023. This may be an attempt by the development company to attract and retain more players to the game.

5.2 Random Forest Regression Prediction Model

Random Forest Regression is an integrated learning algorithm based on decision trees. It combines multiple decision trees to predict output values by voting. Random forest regression is commonly used in areas such as financial risk prediction, medical cost prediction and so on.

5.2.1 Steps in Building the Model

Step 1 Divide the data set
Step 2 Construct a decision tree
Step 3 Random forest training
Step 4 Random forest prediction
Step 5 Model evaluation and tuning

5.2.2 Forecasting with EERIE as An Example

Using Python programming to solve for the results in the case of three different parameters.

Table 9: Distribution of the Reported Results

	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	more tries(X)
parameter 1	0.38	6.78	25	35.4	22	8	2.1
parameter 2	0.5	7.2	26	36.5	20	7	2.2
parameter 3	0.7	8.3	29	38	17	5.6	1.4
mean	0.53	7.43	26.7	33.6	19.67	6.87	1.9

Note: Some sums are not 100% due to rounding.

5.3 Uncertainty of Forecasts

Due to the somewhat long time span between September and March, it is unpredictable that the development company may have adjusted the game's difficulty level or opened up other models during this period. So the accuracy of our model results is still relatively high.

6 Modeling and Solving Problem 3

The requirement of this question is for us to classify all the words in the attached data into intervals of identified difficulty and then build a predictive model to predict how easy a particular word will be. The discussion of this question in this paper will continue to use the results and conclusions we have drawn above to ensure accuracy throughout.

6.1 Integrated Stepwise Forecasting Model

The integrated stepwise forecasting model we have defined is made up of three models, the first being a mean cluster analysis model, the second a coefficient correction model and the third a support vector

machine forecasting model. The advantage of this model is that it uses both traditional statistical analysis algorithms and machine learning type algorithms, which greatly improves the accuracy of the forecasts. Here, we will step through the modelling of this model.

6.2 Mean Cluster Analysis Model

The mean cluster analysis model is a density-based clustering algorithm whose main idea is to move a data point to the densest region around it until it eventually converges on the centre of the densest region.

•Model Building

We used Excel software to find the average number of attempts required to pass each word, and then averaged the average number of attempts required to pass all words to obtain a value that is the average number of attempts (Y) for the words in the given data. The average can be used to indicate the central location of the relatively high concentration of observations in the data, and the average number of attempts required for each word can visually reflect the level of difficulty of the word we pass by this average.

$$\bar{x} = \frac{\sum_{k=1}^{7} f(x)k}{\sum^{f}(x)} \tag{9}$$

•Solution of the Model

After doing the above, we obtained the mean number of words for each of them. Next, we used Weka software to cluster the mean data. Finally, we decided to cluster it into 4 classes(see appendix for screenshots of clustering results).

The results of the clustering are: 20% in the first category, 11% in the second, 41% in the third and 28% in the fourth.

•Classification of Difficulty

We divided the difficulty intervals based on the clustering results and established five categories: those with mean values less than 3.89 belonged to the super easy category, those with mean values between 3.89 and 4.14 belonged to the easy category, those with mean values between 4.14 and 4.32 belonged to the medium difficulty category, those with mean values between 4.32 and 4.69 belonged to the difficult category, and those with mean values greater than 4.69 belonged to the super difficult category.

6.3 Coefficient Correction Model

The coefficient correction model we have defined is based on a number of identical or different indicators to make corrections to the mean values above in order to reduce our forecast errors.

•Words Containing both 'e' and 'a'

We found it interesting that the proportion of words containing both e and a is relatively high in the data in Annex 1, with a total of 57 occurrences, or 16%. The number of occurrences of words in the medium category is higher, 50%, as detected by Python programming. In all other cases, the percentage is similar.

•Establishment Factor

After the mean (Y) obtained after our mean cluster analysis model calculations, it will continue to be multiplied by the following coefficients as appropriate. We define the final mean equation.

$$FM = Y \times i \tag{10}$$

where i depends on which difficulty interval Y falls into above, i=1 when Y is in the medium category, i=0.97 when Y is in the super easy category, i=0.98 when Y is in the easy category, i=1.02 when Y is in the difficult category, and i=1.03 when Y is in the super difficult category.

•Description of Results

When the coefficient correction equation has been calculated, then go on to classify by the difficulty interval above.

6.4 Support Vector Machine Model

Support Vector Machine (SVM) is a machine learning algorithm for classification and regression analysis, based on statistical learning theory and the principle of structural risk minimisation. The main idea is to find an optimal decision boundary (or hyperplane) that separates the different classes of data points so that the different classes can be classified as correctly as possible. We use support vector machines here to validate the results of the previous two models, and quadratic prediction is one of the strengths of the model.

•Steps in Building the Model

Step 1	Select a suitable support vector machine model, define the loss function, regularisation term and other hyperparameters, and perform model training.
Step 2	The model is evaluated using a test set to calculate the model's prediction accuracy, precision, recall and other metrics.
Step 3	Based on the results of the model evaluation, the hyperparameters of the model are tuned to a chieve better performance.

6.5 Calculation with EERIE as An Example

•Calculation of Mean Cluster Analysis Model

Average obtained according to the different parameters of the solution using Python in 6.2.2.

Table 10: Presentation of the Results of the Average

1 try	2 tries	3 tries	4 tries	5 tries	6 tries	X tries
0.53	7.43	26.7	33.6	19.67	6.87	1.9

Then by calculating the formula

$$\bar{x} = \frac{\sum_{k=1}^{7} f(x)k}{\Sigma^{f}(x)} \tag{11}$$

This gives a mean value of 3.8276. i.e. Y = 3.8276.

•**Calculation of Coefficient Correction Model**
Easily obtained FM = 3.71.

•**Calculation of Support Vector Machine Model**
After solving in Python, it was finally confirmed that EERIE is a super simple class.

•**Analysis of the Results**
Information on the frequency of use of the letter 'E' is the most frequently used letter in the English language, so when people play Wordle they often use the letter they are familiar with first. The word EERIE is also a very common adjective in everyday life, and because people know the word themselves, it is an easy word to pass. This is in line with our model which concludes that the word EERIE is a super simple category. This concludes that the integrated stepwise prediction model is more accurate in determining the difficulty of words.

7 Interesting Features of the Dataset

1. We isolated the words that had repeated letters appearing in them and we found that the percentage of words that were successful in one attempt was 0% and 1%, while no other percentage appeared, for example 2%, 3%, 5% and 6%.

2. We found that the overall number of reported results tends to decrease over time, but the number in hard mode divided by the number of reported results tends to increase over time. This suggests that the number in hard mode is more popular with gamers and that hard mode gamers stay fresher for longer.

8 Sensitivity Analysis

We perform a sensitivity analysis on all the previous prediction results. Sensitivity analysis is a method used to assess the degree of response of a model or system to changes in input parameters.

- **Results for Question 1**
  By adjusting the parameters to the Python programming code, it was found that a prediction interval of (8392, 8590) for the number of results reported on March 1, 2023 would be the best result.

- **Results for Question 2**
  By adjusting the parameters to the Python programming code, it was found that the predicted percentages for each count, 0.50, 7.46, 27, 33.9, 19.61, 7, and 1.75, respectively, were the best results.

- **Results for Question 3**
  By adjusting the parameters of the Python programming code, it was found that the word EERIE should be in the simple class for the best result.

- **Conclusions Added**
  Although the best result after sensitivity analysis has a small error with our previous prediction,

and there may be an unavoidable working error in the middle, we can still say that our prediction conclusion is correct.

9 Strengths and Weaknesses

- **Strengths**

1. The various models in this paper are comprehensive, the logic is complete, the text is accurate and the results are based on valid data.
2. The code in this paper is simple and efficient, and the predictions are relatively accurate.
3. The important modelling in this paper uses the idea of a dual model and the results of the analysis and predictions are accurate.
4. This paper does a good job of graphical analysis, with many beautiful visual images.

- **Weaknesses**

1. The number of random samples in this paper is not very large, which may lead to an unavoidable partial numerical error in the mean ratio.
2. The data set of this paper has some too small values, which may lead to some errors in the predicted values of the algorithm.

10 Conclusion

Wordle is a popular word-guessing game, and its innovative and unique gameplay attracts people to challenge and share their results on Twitter, further promoting Wordle. The following are the findings of our team's research and analysis based on the data set provided by MCM.

1. The number of Twitter reported results for Wordle games changes due to **people's curiosity and challenge** for the game **grows rapidly in January**, the number of reported results stabilizes between 25,000 and 35,000 in February, and the number of reported results decreases starting in March as **people's novelty for the game decreases**. We build a time series analysis model based on the dataset given by MCM to predict the number of reported results on Twitter on March 1, 2023 to be **8312 to 8684**. In addition, we build a comparison model with a random sample of mean ratios to investigate the possibility that multiple attributes such as **the number of letters in a word, the presence or absence of repeated letters, and vowels or consonants may affect** the percentage of scores played in hard mode.

2. We build a historical statistical model and a random forest regression prediction model to fit the data set to obtain an intelligent mixed model that can **predict the distribution of the reported results** (see the table 9).

3. We build a comprehensive step-by-step prediction model consisting of a mean cluster analysis model, a coefficient correction model, and a support vector machine prediction model. Combining traditional analysis algorithms and machine learning algorithms, we classify all words into **five categories** according to their difficulty: **words with a mean number of attempts less than 3.89 belong to the super easy category, those with a mean value between 3.89 and 4.14 belong to the easy category, those with a mean value between 4.14 and Those with mean values between 3.89 and 4.14 were in the medium category, those with mean values between 4.14 and 4.32 were in the difficult category, and those with mean values greater than 4.69 were in the super difficult category**. In addition, we found that **'E' is the most frequently used** letter in the vocabulary, so people often use the letter they are familiar with first when playing Wordle. After solving our model, we found that **EERIE is a super easy category**, a very common adjective in everyday life, because people know the word itself, so this is an easier word to pass.

4. In addition, we found some interesting characteristics of the data set. For example, the percentage of successful attempts in one attempt is 0% and 1%. number of reported results is an overall decreasing trend over time, but the number in hard mode/Number of reported results is an increasing trend over time. This suggests that **the number in hard mode is more popular** with gamers and that **hard mode gamers stay fresher for longer**.

11 References

References

[1] Dexin Ran. A study of Shanghai's GDP based on time series analysis[D]. Dalian University of Technology, 2020. DOI:10.26991/d.cnki.gdllu.2020.002692.

[2] Jian Wang . Energy consumption forecasting in Ningxia based on time series analysis [D]. Ningxia University,2015.

[3] Yang Li. Research on off-road pavement identification algorithm based on random forest model interpretation[D]. Jilin University, 2022. DOI:10.27162/d.cnki.gjlin.2022.000915.

[4] Zhi Ji. An enhanced random forest model based on tree sorting [D]. Lanzhou University,2017.

[5] Yunfeng Shi. Text classification algorithm based on deep learning and support vector machine[D]. University of Electronic Science and Technology, 2022. DOI:10.27005/d.cnki.gdzku.2022.001564.

[6] X. W. Wang. Research on classification models based on rough sets and optimal support vector machines [D]. Northwest Normal University, 2022. DOI:10.27410/d.cnki.gxbfu.2022.002140.

[7] Zhenghong Wen Wu . Deep learning-based news text classification system [D]. Nanjing University of Posts and Telecommunications, 2022. DOI:10.27251/d.cnki.gnjdc.2022.000726.

[8] Xiao Yang. Fuzzy C-mean clustering method for stock portfolio optimization [D]. Dalian University of Technology, 2020. DOI:10.26991/d.cnki.gdllu.2020.001747.

[9] Longfei Yu. Research on classification algorithm based on deep factorization machine [D]. Beijing University of Posts and Telecommunications, 2020. DOI:10.26969/d.cnki.gbydu.2020.002338.

[10] Mingming Le. Research and application of data mining classification algorithms [D]. University of Electronic Science and Technology,2017.

12 Appendix

Results of weka mean clustering

Clusterer output

```
Number of iterations: 17
Within cluster sum of squared errors: 355.99899712836407

Initial starting points (random):

Cluster 0: 2022/7/28,4.05
Cluster 1: 2022/12/5,4.8
Cluster 2: 2022/4/14,4.11
Cluster 3: 2022/4/5,4.57

Missing values globally replaced with mean/mode

Final cluster centroids:
                           Cluster#
Attribute      Full Data          0          1          2          3
                 (359.0)     (71.0)     (41.0)    (148.0)     (99.0)
=============================================================================
Date           2022/12/31 2022/12/24 2022/12/26 2022/12/31 2022/12/29
Ê²³ùÖµ             4.1923     3.6727     4.9339     4.0705       4.44




Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       71 ( 20%)
1       41 ( 11%)
2      148 ( 41%)
3       99 ( 28%)
```

Figure 6: Results of Mean Clustering

Code for Relevant Parameters

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.graphics.tsaplots import plot_pacf,plot_acf

%matplotlib
df=pd.read_excel("Problem_C_Data_Wordle.xlsx",parse_dates=["Date"])
df.info()

data=df.copy()
data=data.set_index("Date")

plt.plot(data.index,data['Number of reported results'].values)
plt.show()


train=data.loc[:'2022/10/19',:]
test=data.loc['2022/10/20':,:]

print(sm.tsa.stattools.adfuller(train['Number of reported results']))


acorr_ljungbox(train['Number of reported results'], lags = [6, 12], boxpierce=True)


acf=plot_acf(train['Number of reported results'])
plt.title("Number of reported results Self-correlation picture")
plt.show()


pacf=plot_pacf(train['Number of reported results'])
plt.title("Number of reported results partial correlation picture")
plt.show()
```

Code for Time Series Analysis

```
        import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm

#Read time series data, assuming the data file is 'time_series_data.csv'
data = pd.read_csv('da.csv', parse_dates=['date'], index_col='date')


# Check the smoothness of the data
result = sm.tsa.stattools.adfuller(data)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

# If the data is not stationary, perform differential operation
if result[1] > 0.05:
    diff_data = data.diff().dropna()
    result = sm.tsa.stattools.adfuller(diff_data)
    print('ADF Statistic after differencing: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print('\t%s: %.3f' % (key, value))
    data = diff_data

# Determine the parameters of the time series model
model = sm.tsa.ARIMA(data, order=(5, 1, 1))

# Fit a time series model
result = model.fit()

# View the performance of the model
print(result.summary())

# Predict data for the future
forecast = result.forecast(steps=64)
print(forecast)

```

```
# Visualize prediction results
plt.plot(data)
plt.plot(forecast(100))
plt.show()
```

Code for the Random Forest Regression Prediction Model

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import pandas as pd

# Load the dataset
df = pd.read_csv('data.csv')

# Split the dataset into features and labels
X = df.drop('target_column_name', axis=1)
y = df['target_column_name']

# Split the dataset into a training set and a test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.2, random_state=42)

# Instantiate a random forest regressor
rf_reg = RandomForestRegressor(n_estimators=100, random_state=42)

# Train the model
rf_reg.fit(X_train, y_train)

# Predict test set results
y_pred = rf_reg.predict(X_test)

# Evaluate model performance
mse = mean_squared_error(y_test, y_pred)
print('Mean squared error:', mse)
```

Code for Support Vector Machines

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# Load the dataset
iris = data.load_iris()
X = iris.data
```

```
y = iris.target

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.3, random_state=42)

# Create an SVM classifier and fit the training data
clf = SVC(kernel='linear')
clf.fit(X_train, y_train)

# Predict test data
y_pred = clf.predict(X_test)

# Output prediction accuracy
print('Accuracy:', accuracy_score(y_test, y_pred))
```