Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2305302

Did You Play Wordle Today? An Data Analyses and Prediction about the Game

Summary

Wordle is a word game that is popular all over the world. Each time the game player has six opportunities to fill in the word, and each time when he or she fills in the word, he or she will get the corresponding feedback in the corresponding position. Players need to get the correct answer words with as few guesses as possible. Our originality lies in the combination of time series prediction model, artificial neural network model and other intelligent algorithm, using data analysis and statistical strategies, and obtained some meaningful results.

Firstly, We try to use **ARIMA** model and **LSTM** model to predict the number and distribution range of the report results on March 1, 2023, and analyze whether some attributes of the word are related to the proportion of the number of reports of difficult modes in the total number of packages. Under the premise of 95% confidence, we obtain a relatively accurate prediction interval and prediction results. At the 90% confidence level, we find that there were four word attributes and the proportion of difficult patterns had significant correlation.

Secondly,we use the neural network model to predict the distribution of the number of attempts through the model training.We feed the number of people participating in the questionnaire predicted by the model proposed in question 1,the word attributes and the number of people participating in the difficult mode that is calculated by statistical distribution model into network model as inputs and get outputs as results.However, the neural network usually needs more data,the data volume in this question is small, and it is easy to be over-fitting. Therefore, we use **few-shot learning** as a multiple regression model, and use **MAML** algorithm to train the model, predicts the point estimate of the distribution of the number of attempts, and then establishes the probability distribution model through the statistical law of the number of attempts to obtain the interval estimate.We compare the traditional neural network training method and the nonlinear regression model with MAML and MAML perform better in accuracy,shortening training time and easing over-fitting in few-shot learning task.

Thirdly, we use the **TOPSIS** model based on entropy weight to comprehensively process the data, obtain the weight of each guess, and implement the difficulty classification of words through **K-means++** clustering algorithm. According to the multiple regression analysis model, we have identified the word attributes which affect classification: word promotion, semantic direction, and attributes that are easy to cause confusion. Through the combination evaluation of random forest based on differential evolution, **kernel density estimation** and improved support vector machine, we find that EERIE is relatively easy with high confidence level.

As a supplement to the previous analysis, we also listed the data attribute information and related analysis that may be useful. Our model can adapt to the changes of data better, and the prediction results are not only comprehensive but also accurate with certain reference value.

Keywords: Wordle Puzzle, Time Series Prediction, Word Analysis, Word Relation Prediction, Word Difficulty Prediction

Contents

1 Introduction

1.1 Problem Background

Wordle, a puzzle game published daily in the New York Times, is figuratively called "word cloud" in a sense. Since its launch in November 2021, it has quickly swept the Internet with a high degree of playability and fun, resulting in the birth of various spin-offs. As the poster child for crossword games, Wordle stands out for its unique mechanics and healthy social nature. On the one hand, it creates a false scarcity by offering only one puzzle one day for the whole world to guess. On the other hand, those who managed to guess the word shared their pride on social media such as Twitter, and those who didn't asked for the correct answer below, giving it a mutually reinforcing social quality.

In addition, some players will unconsciously make guesses according to certain fixed ideas during the game, while some players will actively think about better solutions, such as using the principle of information entropy to make decisions, using the prisoner's dilemma, rational man theory, reverse thinking and other game skills to assist decision-making. This leads to many "best solutions".

The game interface is shown in Figure 1.

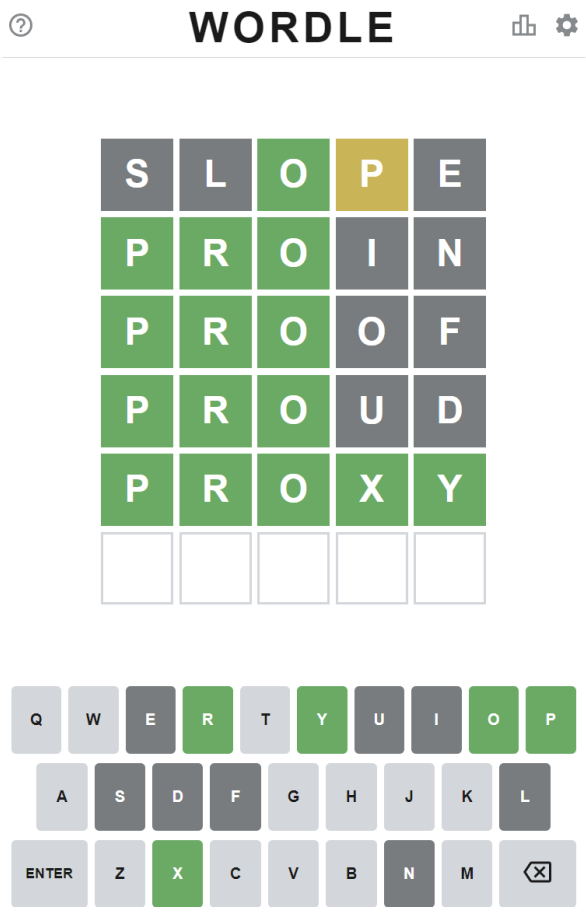


Figure 1: Wordle game interface

1.2 Restatement of the Problem

Through further research and discussion on the background of the problem, combined with the requirements and limitations of the topic, the problem to be solved can be expressed as follows:

- Build a model to explain the changing trend of the number of people reported, and the interval prediction is made for the number of people reported on March 1, 2023.
- Study the relationship between the attributes of words and the percentage of reported scores in the total number of reports on that day under the difficult mode, and explore whether and how the former affects the latter.
- In the premise of knowing the riddle, predict the percentage of guesses on a given date in the future, and analyze the uncertainty of the prediction results. Use ERRIE as an example to predict and assess confidence in the outcome.
- A given word is classified according to its difficulty, and then the attributes of the word are studied according to the category, and a model is built to estimate the difficulty category of the word EERIE.
- Analyze this data set and the interesting findings associated with it.
- To present a summary of the results to the Puzzle Editor of the New York Times.

1.3 Our Work

This problem consists of several relatively independent but internally related problems. We have built corresponding models for each problem. Our work mainly includes the following contents:

- Use the ARIMA model to predict the time series and obtain a wide range is obtained. Then, we used LSTM model to further determine the number of reports and obtain the accurate value, and perform verification and evaluation.
- The number of participants and all attributes of the word were reported as input, and the center value distribution of each trial was output through the few-shot neural network training.
- The entropy weight method is used to assign the weight of guess times and calculate the difficulty score of words.
- SPSS cluster analysis based on K-means++ algorithm is used to classify words into five categories according to their difficulty, and then multiple regression analysis is used to regression fit the difficulty and attribute of words.
- Finally, support vector machine, kernel density estimation and random forest algorithms optimized by differential evolution algorithm were used for learning prediction respectively, and word difficulty was evaluated comprehensively.

The flow chart of this paper is shown below:

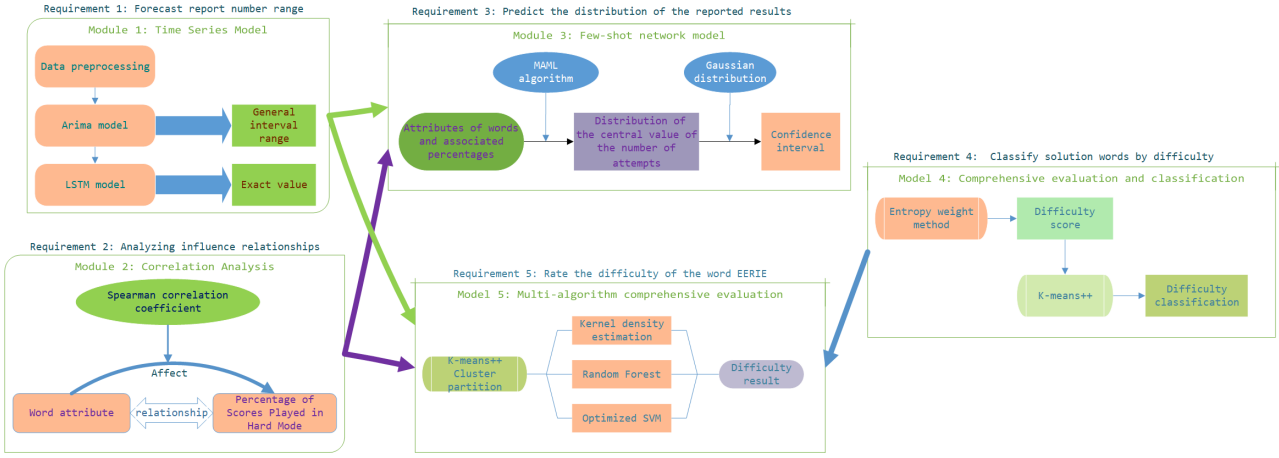


Figure 2: Flow chart of our work

2 Assumptions and Justification

In order to ensure the authenticity and robustness of the modeling, we made the following assumptions:

- The data table used in our analysis is true. Since the only basis for our modeling is the data given in the title, any untrue components will have a huge impact on our modeling results.
- The sampling data in the data table is uniform. That is, the proportion of the number of people who reported the results to the total number of people who played the game and the proportion of the number of people who reported the results to the total number of people who played the game with different times of problem solving remained basically stable over time.
- The age structure and education level of users have no significant change over time. There change of these variables could cause fluctuation in prediction.
- There is no significant change in the distribution of user participation (whether they participate seriously or not) over time. We assume that almost all users keep their playing strategy and do not make huge changes.

3 Notations

4 Time Series Model and Correlation Analysis

4.1 Analysis of the Problem

For the first problem of to create a prediction interval for the number of reported results on March 1, 2023, through the analysis of the topic, our input is the date of a certain day in the future, and the predicted content is the number of people reporting scores that day. This problem can be abstracted as the problem that people pay attention to a hot event over time. We find that this is a time series prediction problem. The independent variable we use is time, and

Table 1: Table 1: Notations used in this literature

Symbol	Definition
r_s	Spearman correlation coefficient
$\phi(x)$	Mapping function
f	MAML model
a	a future date of MAML Model
q	transition distribution of MAML Model
$\mathcal{T}$	lauched task of MAML Model
$\mathcal{L}$	Loss function of MAML Model
H	episode length of MAML Model
W	the weight of MAML Model
b	bias value of MAML Model
v	certain layer of MAML Model
θ	Judge f's performance of MAML

the result/dependent variable is the number of reports submitted, and this is used as an input to the proportional model of the number of attempts in the second question.

It can be seen from the problem that we want to predict the set and confidence interval of the total number of reports on March 1, 2023. We mainly use **LSTM** model for accurate prediction and **ARIMA** model for the confidence interval.

4.2 Calculating and Simplifying the Model

4.2.1 ARIMA Model For Time Sequence Prediction

We use ARIMA model to predict and analyze the wide range. Although this method is traditional, it is practical and easy to use, and can quickly get the prediction range. However, the disadvantage is that the change trend of the early data is different from that of the later data, which will interfere with the prediction.

In ARIMA(p, d, q), AR is "auto regressive" and p is the number of auto regressive items; MA is "moving average", q is the number of moving average terms, and d is the number of differences (order) to make it a stationary sequence:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d X_t = (1 + \sum_{i=1}^{q} \theta_i L^i)\epsilon_t \tag{1}$$

How to difference the data is the the main focus of using ARIMA Model. Figure 3 shows our strategy to difference the data and select proper ARIMA series models and their parameters.

By testing, we finally determine that $p = q = 1, d = 2$, and error analyses are below:

Firstly, The residual diagram of the model is evenly distributed on both sides of the $x$axis, indicating that there are defects, but the data in the early stage is significantly seasonal, while the data in the later stage is not significantly seasonal. So in order to make up for this defect, we adopted LSTM intelligent algorithm.

Secondly, it showed an upward trend in the previous time, with a rapid upward trend; In

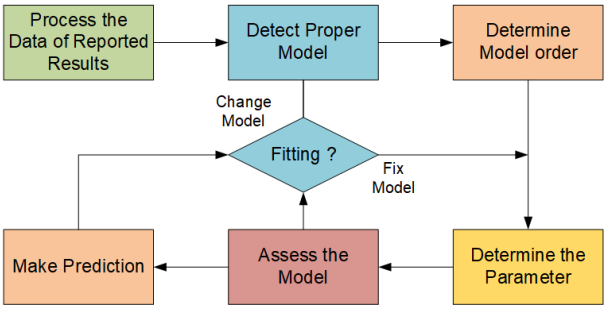


Figure 3: The prediction process of ARIMA Model

the later stage, it showed a downward trend, and the decline rate was slow. What's more, the variable is not white noise and has practical significance.

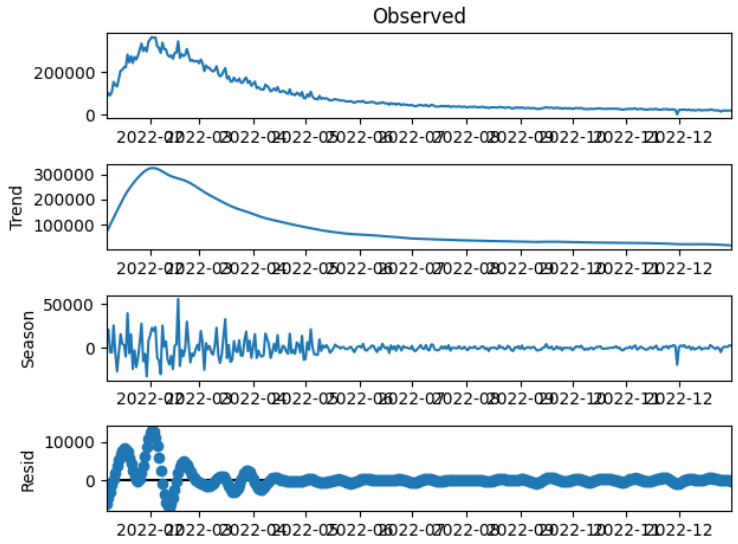


Figure 4: The error analyses

4.2.2 LSTM Model For Analysing the reported results

LSTM model is a time-cycle neural network, specially designed to solve the long-term dependence problem of general RNN (cyclic neural network). All RNNs have a chain form of repetitive neural network modules. In the standard RNN, this repetitive structure module has only a very simple structure, such as a tanh layer. The LSTM model has a large and rich hidden layer, which can deeply mine the internal information of small samples, accurately grasp the change trend of data, and improve the accuracy of data.

We use the method of rolling forward input to advance. In this case, we first read the data in the Excel file and then process the data. Next, we load the previously trained LSTM model and use it to predict future values. Finally, we save the forecast results to a new Excel file.

LSTM algorithm include Forget gate, which determines how much the unit state $c_t - 1$ at the last moment is reserved till the current time c_t, Input gate which determines how much the input x_t of the network at the current time is saved to unit state c_t, and Output gate, controlling unit status c_t that determine the amount of LSTM's current output value h_t.

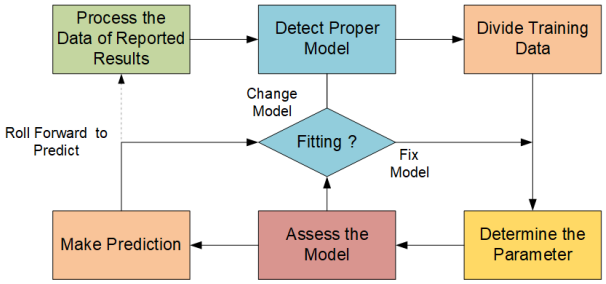


Figure 5: The prediction process of LSTM Model

We optimize the parameter of the LSTM Mode and get the proper parameter of LSTM Model in (Table 2)

Table 2: Parameter Configuration

parameters	values
time step	3
features	1
train size	0.8
lstm units	50
dropout prob	0.2
batch size	32
number of epochs	100
future steps	7

For training data and set test data set, the proportion is 7:3. We find that 7-10 days are the best single period of the prediction with a quarterly window. The test set is the last 100 days of the given data set. Below is the fitting curve (Figure 6).

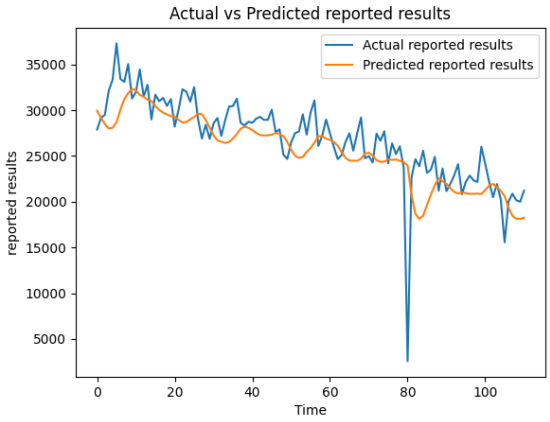


Figure 6: The fitting curve of LSTM Model

And Our focus is on sliding windows, with all data normalized to $[0, 1]$

4.3 The Model Results

Here is out results using ARIMA model. The number pf reported results will probably be in the section of $[16064, 25086]$ $(alpha = 0.05)$. And the precise prediction of March 1, 2023

is 20575 (Figure 7).

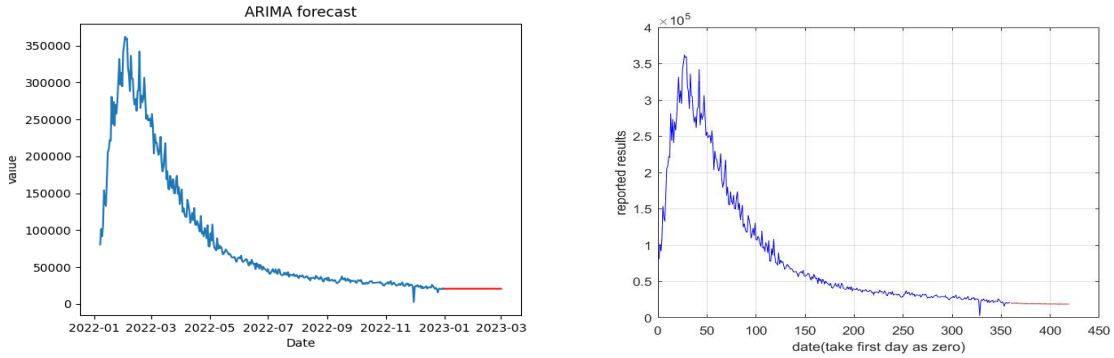


Figure 7: results of ARIMA Model(left) and the prediction curve of LSTM Model(right).

After verifying the feasibility, we forecast the number of reports from January 1, 2023 to March 1, 2023, as shown in the figure. The total number of reports will decrease slightly in fluctuation and finally stabilize in a range. Prediction shows that the final number of reports on March 1, 2023 will be around 18600 (Figure 7).

4.4 Validate and Evaluate the Model

Besides Methods used to access our Models, other evaluation indicators are:Root Mean Square Error (RMSE) Mean Absolute Error (MAE), Mean Absolute Percentage Error(MAPE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2}$$

$$MAE = \sum_{i=1}^{n}\frac{|Y_i - X_i|}{n} \tag{3}$$

$$MAPE = \sum_{i=1}^{i=n}\frac{(X_i - Y_i)}{X_i} \times 100\% \tag{4}$$

We calculated the $MSE, MAE, MAPE$ of two models:

Table 3: ARIMA analyses

indicators	values
RMSE	0.047
MAE	0.0032
MAPE	-

Table 4: LSTM analyses

indicators	values
RMSE	0.119
MAE	0.072
MAPE	3.403%

Both of these meets the two or three indicators . But LSTM can deal with more complicated variables and disturbance.

4.5 Analysis of Word Attributes Affecting Percentage of Hard Mode

This is a question of correlation analysis. Through the analysis of words, we summarized the properties of some words, which are: proportion of vowels, the first letter, the last letters,

the second letter, the third letter, the fourth letter, special structure, the number of different letters, whether the word is easy to cause confusion, parts of speech(noun-0 adjective-1 verb-2 other-3 polysemy-4) range of usage(academic-1 life-2 other-0), Semantic direction, syllable, letter frequency entropy, word frequency entropy, word frequency fraction.

We used Spearman correlation coefficient to analyze the relationship between these word attributes and the proportion of difficult patterns in the total number of reports, calculated the correlation coefficient, and made a correlation test. The results of correlation test are shown in Figure 8.

P-value	Frequency	Vowel Ratio	Number of letters	Number of syllables	Word frequency score	Hardmode percentage
Frequency	1	0. 933323981	0. 37069602	0. 765589832	3. 46E-66	0. 253773006
Vowel Ratio	0. 933323981	1	0. 363183091	0. 358852768	0. 334311923	0. 121115179
Number of letters	0. 37069602	0. 363183091	1	0. 197985038	0. 082954744	0. 102242045
Number of syllables	0. 765589832	0. 358852768	0. 197985038	1	0. 576438926	0. 017197423
Word frequency score	3. 46E-66	0. 334311923	0. 082954744	0. 576438926	1	0. 139817433
Hardmode percentage	0. 253773006	0. 121115179	0. 102242045	0. 017197423	0. 139817433	1

R-value	Frequency	Vowel Ratio	Number of letters	Number of syllables	Word frequency score	Hardmode percentage
Frequency	1	0. 004431022	0. 047383993	0. 015789761	0. 750467142	-0. 060387588
Vowel Ratio	0. 004431022	1	-0. 048131927	-0. 048567407	0. 051100212	0. 081959804
Number of letters	0. 047383993	-0. 048131927	1	-0. 06810041	0. 091633083	-0. 086383429
Number of syllables	0. 015789761	-0. 048567407	-0. 06810041	1	-0. 029578159	-0. 125680748
Word frequency score	0. 750467142	0. 051100212	0. 091633083	-0. 029578159	1	-0. 078078466
Hardmode percentage	-0. 060387588	0. 081959804	-0. 086383429	-0. 125680748	-0. 078078466	1

Figure 8: the Correlation Test(left) and the Correlation Coefficient(right)

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\left(n^2-1\right)} \tag{5}$$

$$r_s\sqrt{n-1} \sim N(0,1) \tag{6}$$

$$H_0 : r_s = 0, \quad H_1 : r_s \neq 0 \tag{7}$$

Formula 5 calculates Spearman correlation coefficient, and Formula (6) (7) verifies the correlation between word attributes and the percentage of scores reported that were played in Hard Mode.

At 90% confidence, we found that the proportion of vowels, the number of different letters, the number of syllables, and the word frequency score have significant correlation. Specifically, the greater the proportion of vowels, the less the number of letters and syllables, the higher the word frequency score, and the higher the proportion of difficult patterns in the total report of the day. As shown in the Figure 8. This is in line with our past experience and common sense.

4.6 Other Factors that Affects Modeling

Since there are a large proportion of Chinese students participating in the American competition, and most of them have not played Wordle games, we must consider the increase in the number of reports brought by students participating in the American competition playing Wordle, which can be regarded as a constant of C.

It is assumed that 30% of Chinese students participating in the American competition will play Wordle, and 50% will report their results. According to the number of participants in the American competition in 2023, the predicted value will rise systematically from the beginning to the end of the competition.

In 2022, the number of participants in the American competition was 80693, 98% of them were Chinese students (from earlier data in 2019), and 30% insisted on playing Wordle within a week, so the short-term fluctuation will be $23723/20380 = 116\%$, which is quite a huge number.

However, with the loss of time, the heat will dissipate quickly, so this kind of disturbance will not affect the prediction of the results on March 1, 2023.

5 Few-shot Learning Words Model

5.1 Analysis of the Problem

For problem 2,we regard the distribution of the number of prediction attempts as a multivariate regression problem. After entering the word, we first calculate the attribute of the word. In order to consider the time factor, we use the distribution of the number of difficult people to calculate the number of people participating in the difficult mode according to the total number of reported people calculated in question 1 as the input of the model, we use the neural network to predict the distribution center value of the number of attempts for trying. The probability distribution model is used to predict the confidence interval. It is worth noting that for this problem, the number of samples is small, and over-fitting is easy to occur using traditional neural networks. Therefore, we adopt the **few-shot learning**, adopt the **MAML** algorithm to train the model, and adopt a relatively simple 5-layer full-connection layer architecture, effectively avoiding over-fitting of the model, and at the same time making the loss relatively small, which has achieved relatively good results.

5.2 Meta-Learning Algorithm for Fast Adaptation of Deep Networks

Deep neural network is powerful to a large number of problems,however,there is a limitation to obtain sufficient samples due to some reasons.So we use **a few-shot learning algorithm**[1] which is specifically for small data.The Meta-Learning Algorithm can effectively prevent over-fitting and can achieve rapid adaptation that is general and model-agnostic.

In order to accomplish the goal of few-shot meta-learning,the model is trained during the period of learning on a data set.We denoted f as the model,which can be seen as a map from the attributes of word to the outputs (1, 2, 3, 4, 5, 6, X) for a future date denoted a. The algorithm is universal and can be use in many tasks,denoted each task $\mathcal{T} = \{\mathcal{L}(\mathbf{x}_1, \mathbf{a}_1, \ldots, \mathbf{x}_H, \mathbf{a}_H), q(\mathbf{x}_1), q(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{a}_t), H\}$ and each task has a loss function $\mathcal{L}$,a distribution of inputs $q(\mathbf{x}_1)$,a transition distribution $q(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{a}_t)$ and an episode length H.In our task of words attributes mapping,we set $H = 1$.The model can generate sample by choosing an output $\mathbf{a}_t$ at time t.The loss $\mathcal{L}(\mathbf{x}_1, \mathbf{a}_1, \ldots, \mathbf{x}_H, \mathbf{a}_H) \to \mathbb{R}$ mapping can provides feedback related to task. Consider a distribution over tasks $p(\mathcal{T})$ that need our model to be able to adapt to.In the K-shot learning data sets,the model use only K samples and want to learning the mapping from q_i and feedback generated by $\mathcal{T}_i$.During training period,a task $\mathcal{T}_i$ is sampled from $p(\mathcal{T})$,the feedback from the corresponding loss $\mathcal{L}_{\mathcal{T}_i}$ from $\mathcal{T}_i$ and then tested results on new samples from $\mathcal{T}_i$.The model f is then improved by considering the test error on

new data from q_i changes associated with parameters.The test error play a role as training error of the meta-learning process.At the end of training,new tasks are sampled from $p(\mathcal{T})$ and after learning from K samples,meta-performance can be measured.

The intuition behind this algorithm is that some internal representations are more transferrable than others.Due to the model will be fine-tuned using gradient-descent rules,the algorithm aim to find model parameters that is sensitive to task,which means a small changes of parameters can make big difference on loss function when altered in the direction of gradient of loss.Sketch Map is showed in Figure.9.

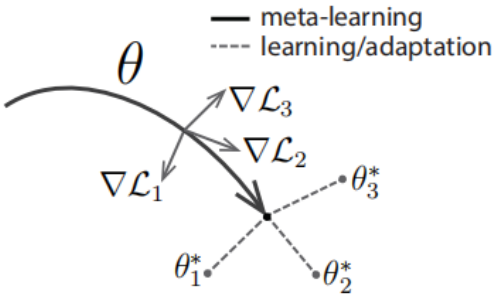


Figure 9: Sketch Map of meta-learning algorithm (MAML), which optimizes for θ that can quickly adapt to different tasks.

In the algorithm,assume that loss function is smooth enough to parameter θ,so gradient-descent can be used.Formally,consider the model is represented by a parameterized function f_θ with parameters θ.Facing a new task $\mathcal{T}_i$,parameters θ becomes θ_i' using gradient descent rules for specific task $\mathcal{T}_i$.The model parameters are trained by optimizing algorithm for the performance of $f_{\theta_i'}$.Concretely,the meta-objective is as follows:

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right) \quad = \quad \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}}\left(f_\theta\right)\right) \tag{8}$$

Since the optimization period is judged by the parameter θ and model is computed by updated parameters θ'.In effect,the meta-learning method can optimize the model with a small number of gradient steps and produce great behavior on the task.The meta-optimization across tasks is performed via stochastic gradient descent.The **MAML** meta-gradient update involves a gradient through a gradient.Besides,it requires an additional backward pass through f to compute Hessian-vector products,and we use pyTorch to achieve the algorithma and the pseudocode is showed in appendix A.

For our regression task,we can define the horizon $H = 1$ and drop the timestep subscript on $\mathbf{x}_t$.The task $\mathcal{T}_i$ generates K i.i.d. input $\mathbf{x}$ from q_i and the task loss is represented by the error between the model's output for $\mathbf{x}$ and the corresponding target values y for that observation and task.We choose mean-squared error(MSE) as loss function,which can be described as followed:

$$\mathcal{L}_{\mathcal{T}_i}\left(f_\phi\right) = \sum_{\mathbf{x}^{(j)}, \mathbf{y}^{(j)} \sim \mathcal{T}_i} \left\| f_\phi\left(\mathbf{x}^{(j)}\right) - \mathbf{y}^{(j)} \right\|_2^2 \tag{9}$$

where $\mathbf{x}^{(j)}$,$\mathbf{y}^{(j)}$ are a pair of data sampled from task.For our K-shot regression,K pairs are provided for each learning task,for a total of NK data points points for N-way classification.Given a distribution over tasks $p\left(\mathcal{T}_i\right)$,these loss function can be modified by each task.

5.3 The Structure of the Few-shot Network Model

Since the number of samples is small and in order to prevent from over-fitting,we can't use models that are too complex and we can not train for too many episodes.

The structure of our five-layer few-shot network model is show in Fig.10.We feed the attributes of words that are selected in problem 1, as well as the number of reported results and number in hard mode predicted by the model proposed in problems 1 As a consideration of the time factor of the date,after five Linear layer with tanh as activation function,the model generate the predicted distribution of number of attempts.Formally,denoted v_{i} as this layer and v_{pre} as the previous layer,W is the weight and b is bias,then the linear can be described as follow:

$$v_{\mathrm{i}} = W v_{pre} + b \tag{10}$$

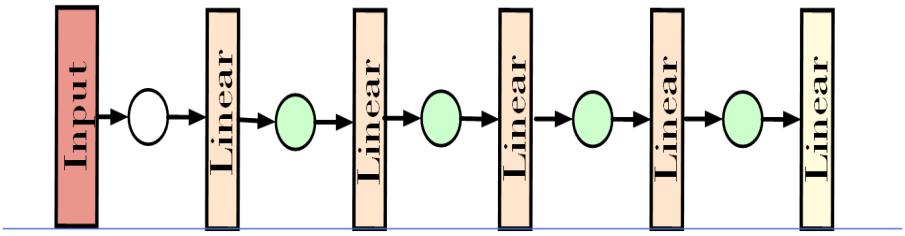


Figure 10: The structure of nested five-layer few-shot network model

5.4 Training Period

5.4.1 Training Method

We use **MAML** introduced in part 5.2 to train the model,which can get good results with only a few samples.In order to evaluate model performance,we divide the data into training set and test set according to the ratio of 7:3. Before feeding data,we use normalization method to process on our data to eliminate the impact of data distribution and make it easier to train,it can be described as followed:

$$x_i = \frac{x_i - \mu_i}{\sigma_i} \tag{11}$$

where μ_i and σ_i are the mean and standard deviation of the i^{th} feature.

5.4.2 Parameters Setting

To prevent from over-fitting,out model is trained for only 3000 epochs and we set k-shot to be 20,which means we sample only 20 samples for each task.After training,we get a great results which will be introduced in the next part.

5.5 The Model Results

5.5.1 Changes of Loss

The change of training and test losses during are showed in Fig.11. As the number of training epochs increases, the training loss continues to decrease. However, when the number of training rounds is more, the test loss will increase instead, and the surface model will be over-fitting. Therefore, the training epochs we choose can not only minimize the loss, but also reduce the over-fitting of the model.

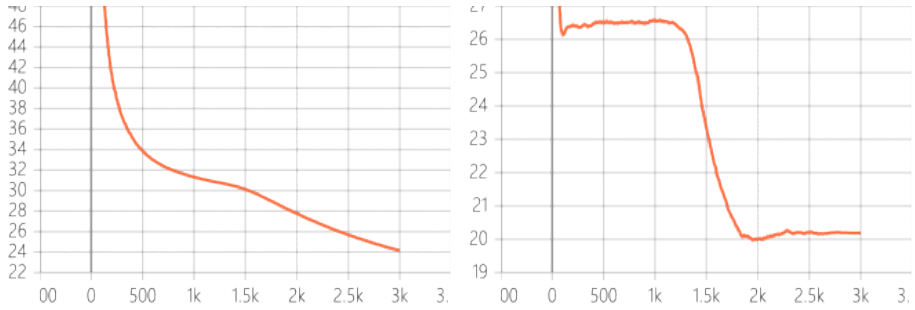


Figure 11: Error change curve of training (left) and testing (right) of MAML training methods.

5.5.2 Prediction of the Distribution of Attempts

After finishing the training of model,when given a solution word on a future date,just compute the number of reported results and number in hard mode by the model proposed in problem 1,and list the attributes of the word as the input of the model and we can get the center value of predicted distribution which can be seen as mean value $\bar{x}$, and confidence interval is obtained by interval estimation of mean value.We carry out a normal test on the distribution of word attempts and found 3 tries and 5 tries pass the normal test at 95% confidence level. Although the number of other attempts has not passed the normal test, according to the large number theorem, the sample size of this data is large, and the number of samples will continue to increase over time. According to the central limit theorem, the distribution of the number of attempts is also similar to the normal distribution and can be expressed as $X \sim N(\mu, \sigma^2)$, where μ is mean value and σ^2 is standard deviation.After calculate,we get the normal distribution mean and variance of the number of attempts and we can calculate the interval length in the followed formula.Since $X \sim N(\mu, \sigma^2)$, so pivot $G = \frac{x-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$,so choose the confidence level α, if we want c and d to meet $P(c \leq G \leq d) = \Phi(d) - \Phi(c) = 1 - \alpha$, so we can get 1- α equivalent confidence interval of μ is:

$$[\bar{x} - u_{1-\alpha}\sigma/\sqrt{n}, \bar{x} + u_{1-\alpha}\sigma/\sqrt{n}] \tag{12}$$

where $u_{1-\alpha}$ is the $1-\alpha$ quantile of standard normal distribution.Distribution of tries times and interval length calculated by formula 12 result are showed in Table .5. As for word EERIE,we

Table 5: Distribution of try times

Number of attempts	distribution	unilateral interval length
1 try	N(0.47,0.61)	0.10
2 tries	N(5.84,16.61)	0.31
3 tries	N(22.72,60.54)	2.38
4 tries	N(23.64,28.67)	4.03
5 tries	N(11.56,35.37)	2.48
6 tries	N(2.80,38.53)	0.93
7 or more tries (X)	N(1.58,16.99)	0.10

feed the date March 1, 2023 into the model proposed in problem 1, we can get the predicted number of reported results N_{total} are 18600 and we calculate the average proportion of people participating in the difficult mode to the total number is about 0.055, then we get use the rate r to calculate the number in hard mode N_{hard} through the formula:

$$N_{\text{hard}} = r * N_{total} \tag{13}$$

We feed the important attributes of word and N_{total} and N_{hard} into the model and we can get the predicted mean value of the distribution of tries times, then we add the interval length calculated in Table 5 by formula 12 and we can calculate the distribution of tries times,inputs of model are showed in Appendix 12 and result of predicting the associated percentages of (1, 2, 3, 4, 5, 6, X) for the word EERIE on March 1, 2023 are showed in Table.6. We choose the confidence level α to be 95% and we show the interval estimation in Table 6, the accuracy of the model is high, so we have great confidence in the predicted distribution.With the probability of making mistakes not exceeding 5%, we can think that the distribution of this prediction is roughly correct.

Table 6: Predicted Distribution Result

tries times	central value	interval estimation
1 try	0.053	[0.00,0.16]
2 tries	3.359	[3.05,3.67]
3 tries	17.599	[28.84,36.90]
4 tries	32.871	[15.22,19.98]
5 tries	27.360	[24.88,29.84]
6 tries	14.056	[13.13,14.99]
7 or more tries (X)	3.675	[3.57,3.78]

5.6 Validating and Evaluate the Model

5.6.1 Model Evaluate

In order to compare the advantages of the MAML algorithm we adopted, we compared the **MAML** method with the traditional neural network training method. The two groups used the same model, the same training data set and test data set, and the losses of the traditional training method shown in the Figure.12.Compare to Figure 11, the stability test loss of the traditional method is 7.1 more than that of the MAML method(detailed losses are shown in Table 7), and the MAML method converges faster, which can effectively save computing resources.Besides,the training error of the traditional training method for this kind of small data set is about 5 higher than the test error, indicating that the model has a certain degree of over-fitting, while the MAML training method is almost the same, indicating that the MAML method effectively alleviates the over-fitting and can achieve good results for small sample learning.

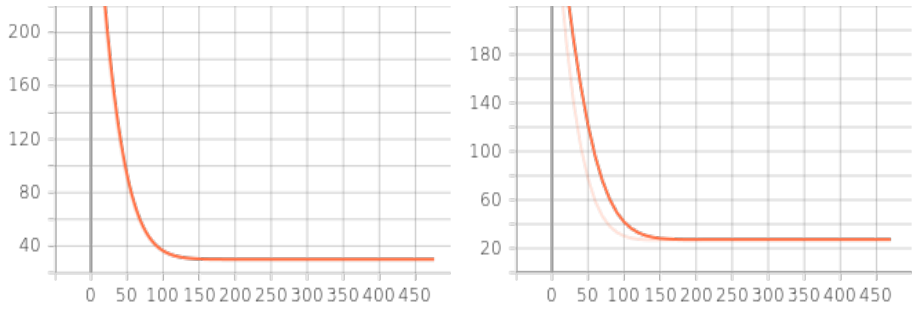


Figure 12: Error change curve of training (left) and testing (right) of traditional training methods.

We also used the sklearn library to test the same training data set and test data set with 8-degree polynomial nonlinear regression. The test loss is 31.2, which is larger than the neural network trained by the two training methods(Table 7), and it takes longer training time, which further proves that our model is more excellent.

Table 7: Stability Loss

method	Stability Loss
Traditional neural network training	27.3
MAML	20.2
Nonlinear SVM	31.2

5.6.2 Uncertainties in Models and Forecasts

Since the model is trained by network, as well as some other factors, there are some uncertainties may affect the performance of model. Concretely, there are: The main contributions are summarized as follows:

1) Due to the parameters and random initialization,model may be different each time.

2) The attributes of some words are subjective, and the frequency of words will gradually change over time.

3) With the accumulation of time, the player's game experience may gradually accumulate and even understand the rules of word change, which reduces the number of attempts. At the same time, the author may also adjust the difficulty of the selected words, and the model must be constantly updated with the passage of time and the change of data.

4) The existence of random factors: due to the gradual change of player's age, country, vocabulary, seriousness of game participation and other random factors such as environment and intuition, the accuracy of model prediction may also be affected.

6 Quantification, classification and prediction of word difficulty

6.1 Holistic Problem Analysis and Model Architecture

To achieve word classification based on difficulty, we first need to obtain a measure of difficulty degree.In this paper, we used **TOPSIS model based on entropy weight** to calculate the weights of different guesses, and the overall difficulty coefficient of words is calculated according to the weights obtained.Then, the **K-means++ algorithm** built into SPSS software was used to realize the cluster analysis based on the difficulty coefficient, and the corresponding difficulty level and corresponding word class were obtained. **Multiple linear regression analysis** and **Spearman correlation coefficient analysis** are used to get the relationship between word difficulty coefficient and attribute.Finally, the **kernel density estimation**, **support vector machine optimized by differential evolution algorithm** and **random forest** are used to learn the data and predict the difficulty level of new words respectively. The difficulty level with the largest number of decision results among the three methods is selected. If the results obtained by the three methods are different, the result with the highest classification accuracy is selected.

6.2 Detailed Introduction of Models

6.2.1 TOPSIS Weight Calculation Model Based on Entropy Weight Method

TOPSIS method is a commonly used comprehensive evaluation method, which can make full use of the original data Information, whose results can accurately reflect the gap between the evaluation schemes. The basic idea is to assume positive and negative ideal solutions, measure the distance between each sample and positive and negative ideal solutions, get the relative closeness degree to the ideal solution (that is, the closer it is to the positive ideal solution and the farther it is from the negative ideal solution), and rank the pros and cons of each evaluation object.

Entropy weight method is an objective weighting method based on the principle that the smaller the variation degree of the index, the less information reflected, and the lower the corresponding weight value should be. Objectively speaking, the data province tells us the weight.

Through the overall analysis of the data, the weight corresponding to each attempt times is obtained, and the overall difficulty score is calculated by weighting. Since the data obtained by direct calculation is generally small, it is multiplied by 100 before further analysis.

Table 8: Number of attempts and corresponding weight

1 try	2 tries	3 tries	4tries	5 tries	6tries	7 or more tries
0.059628	0.104344	0.231523	0.237976	0.193498	0.148193	0.024838

6.2.2 K-means++ Clustering Algorithm Based on SPSS

The K-means++ algorithm is based on the traditional k-means algorithm. By improving the initialization of cluster centers, each cluster center is dispersed as far as possible, so as to effectively avoid the unreasonable position of the initial cluster center that causes the result of Kmeans clustering algorithm to be seriously affected.

Randomly select a sample point from data set χ as the first initial clustering center c_i , calculate the shortest distance between each sample and the existing clustering center, expressed as $D(x)$, and then calculate the probability $P(x)$ of each sample point being selected as the next clustering center:

$$P(x) = \frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2} \tag{14}$$

Select the sample point corresponding to the maximum probability value as the next cluster center, and repeat the above steps until k cluster centers are selected.

Therefore, it can be concluded that the basic principle for K-Means ++ algorithm to select initial clustering centers is that **the distance between initial clustering centers should be as far as possible**. In this paper, the difficulty score is taken as reference, and the built-in K-means++ algorithm of SPSS is used for cluster analysis. The difficulty level is from 1 to 5, and corresponding word classification is obtained.

6.2.3 Regression Analysis

We use multiple linear regression model to analyze the relationship between word difficulty and word attribute. In the previous model, we obtained the quantitative evaluation score and rating of the difficulty of each word. Combined with the previous word attribute classification, the 'OLS + robust standard error's Mode is used to calculate the difficulty coefficient of the word attribute in the game, and the confidence level is obtained.

After analysis, the word attributes that are strongly related to the difficulty of words are the promotion of words, semantic direction and which it is easy to cause fusion.

6.2.4 Kernel Density Estimation

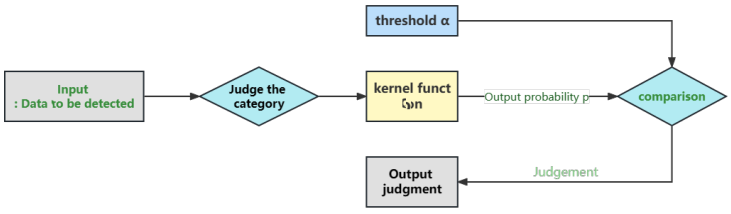
Kernel density estimation (KDE for short) is a non-parametric estimation method, mainly used to estimate unknown density parameters. Different from the parameter estimation method, it does not need to assume that the data obey some distribution, that is, it can estimate the unknown density function according to the data sample without using any prior conditions, so as to achieve the goal of having the minimum mean square integral error between the estimated results and the real results.

The flow of our algorithm is shown in Figure 13: For the input data to be detected, after the cluster division is completed, the kernel function estimation hypothesis test is used to detect outliers according to their categories. The detection principle is as follows: Firstly, the normal data is divided into several categories according to the K-Means ++ clustering method. For each data to be tested, it is also divided into one category according to the K-Means ++ clustering method. For the data in the subcluster, the kernel density estimation method is used to obtain the probability density function $f(x)$ of the data to be tested:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(x - x_i\right) \tag{15}$$

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{16}$$

Finally, the threshold value δ is set and the kernel output probability p is compared with the threshold value. When $f(x) < \delta$, it is considered abnormal; otherwise, it is considered normal.



Kernel density estimation method

Figure 13: Kernel Density Estimation

6.2.5 Random Forest

Random forest algorithm is a kind of integrated learning algorithm based on decision tree. It is a combination classifier based on decision tree with random selection idea and integration idea of set features. It adopts self-help method to carry out put back sampling and generate training subsets to ensure that N random sampling generates N training subsets with the same size.

Each training subset separately constructs its own decision tree, which includes two processes of node segmentation and random selection of random feature variables. Node segmentation compares information attributes based on split rules, and selects the information attributes of optimal comparison results to generate subtrees to realize the growth of decision trees. Random characteristic variables are generated by random selection of input variables and random selection of information attributes for node segmentation. The random selection of the training subset and the random selection of the node attributes ensures the randomness of the random forest and avoids the dilemma of overfitting and local overoptimization of the model. Finally, the average value of N decision tree regression prediction results is selected as the final prediction value.

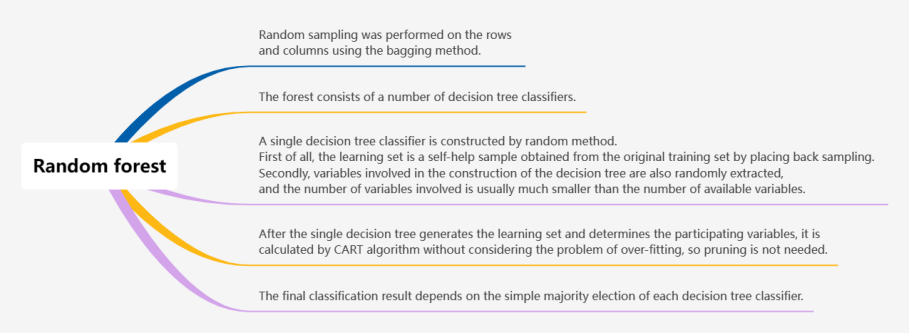


Figure 14: Random Forest Mind map

6.2.6 Differential Evolution Algorithm Optimized Support Vector Machine

The basic idea of support vector machine is to map the input x to the high-dimensional or even infinite dimensional feature space Fthrough a nonlinear mapping $\phi(x)$, and then construct the maximum interval classification hyperplane $w \cdot \phi(x) + b$ in the feature space, which is equivalent to:

$$\min \frac{1}{2}\|w\| + C\sum_{i} \xi_i \tag{17}$$

C is the compromise parameter between classification error and maximum interval, i.e. the penalty parameter, ξ Is a relaxation variable. Use the duality in optimization theory. According to the theory, the final classification equation is:

$$y = sgn|\sum_{s\ v} y_i\alpha_i(\phi(x_i)\cdot\phi(x)) + b| = |\sum_{s\ v} y_i\alpha_i K(x_i, x) + b| \tag{18}$$

In the formula, $K(x_i, x) = \phi(x_i)\cdot\phi(x)$. The kernel function must meet the Mercer condition. The commonly used kernel functions include polynomial kernel function, radial basis function

(RBF), kernel function and sigmond kernel function. Therefore, the kernel parameter and the penalty parameter C are closely related. Finding this parameter is a relatively difficult process. The scope of application of different kernel functions is different. The new classification ability of SVM changes dramatically with the parameters, and there are many extreme points. This requires us to find appropriate parameter selection methods.

Differential evolution algorithm is a real-coded population-based evolution The optimization algorithm of can be used to solve the total Local optimization problems. Without losing generality, global optimization problems can be transformed into To solve the following minimization problem: $\min_{xinD} f(x)$ Where $x = [x_1\ x_2\ \cdots\ x_n] \in D$. R^n is a continuous variable. The objective function $f : D \rightarrow R$ may be discontinuous or non-differentiable Assume that there are P individuals in each generation of k in the DE algorithm, namely The group size is P. The $k$represents x Where $x_{i1}^k\ x_{i2}^k \cdots\ x_{in}^k$ is the i of generation k Individual. The basic DE algorithm mainly includes three operators: mutation Hybridization and selection Select according to the function value, evaluate the newly generated individual i to find its function value, and then decide whether to select the newly generated individual according to the following criteria:

$$x_i^k = \begin{cases} z_i & \text{if } f(z_i) \leq f(x_i^k), \\ x_i^k & \text{if } others \end{cases} \tag{19}$$

In the DE algorithm, there are three control parameters that maintain constant values: step size α, p_c, and P. Parameters α and p_c affects the robustness of the search process and the convergence speed of the algorithm. Their optimal value depends on the characteristics of the objective function and the population size P.

SVM searches for optimal parameters through DE algorithm

The DE algorithm will find the optimal parameters more efficiently with its strong global search ability, and the process is as follows[2]:

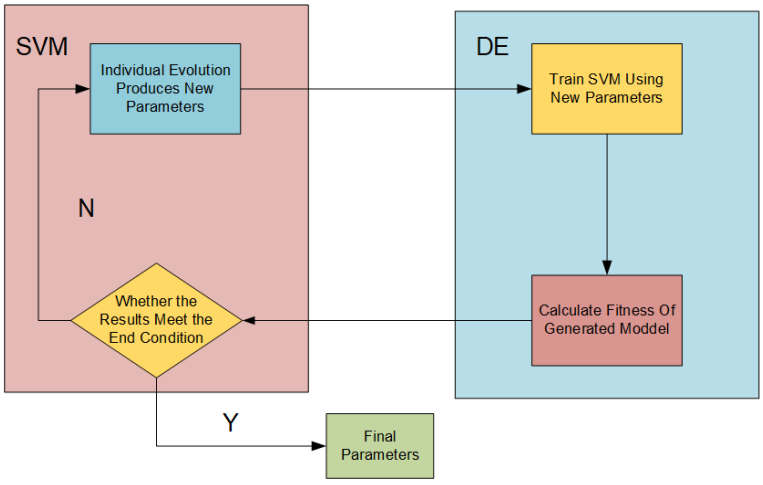


Figure 15: Working Process of SVM using DE Algorithm

The principle of SVM parameter selection using DE algorithm is shown in Figure 15 As shown in, P individuals in DE algorithm evolve in parameter space, each evolutionary generation trains SVM classifiers with the newly generated parameters, calculate the fitness corresponding to each group of parameters, and then judge whether it is satisfied. If stop

conditions satisfied, it will output the optimal parameters; Otherwise, it will continue to evolve and generate new parameters. Because DE algorithm adopts real-value coding, we only need to initialize arbitrarily in the parameter space.

6.3 The Difficulty Prediction of EERIE

In order to ensure the accuracy and objectivity of the prediction results and reduce the random error caused by the chance of a single algorithm, we respectively used random forest, kernel density estimation and SVM optimized by differential evolution algorithm to predict the difficulty of eerie. The results are as follows:

Table 9: The predicted results of three methods

Algorithm	Difficulty score	Difficulty level	Accuracy
Random Forest	28.39628	2	69.64%
optimized SVM	56.96214	4	52.08%
Kernel density estimation	30.10286	2	45.88%

By analyzing the data in the table, we can conclude that the word **EERIE** is most likely to have a difficulty level of 2, with a confidence level of 69.04%.

7 Other Interesting Features of the Data Set

- Through simple linear fitting, we find that the overall level of players is slowly improving.
  - Through simple linear fitting, we find that the overall level of players is slowly improving. The proportion of 1 attempt and 6 attempts is decreasing, the proportion of 2, 5 and unsuccessful attempts is basically unchanged, and the proportion of 3 and 4 attempts is increasing, indicating that the overall level of players is improving, and the level of topics is also improving.

- There are Changes in player psychology hidden in data sets.
  - When faced with choices, people will give priority to the options that are most beneficial to them at present, thus ignoring the collective interests of the team. By analogy, in word guessing games, the general psychology of the players is generally to give priority to the letters that are close to the direction of the last guessed. letters, which may often lead to the idea of "depth first traversal" following a clue to the black, but the way to the black is now impassable, so they have to overturn it.
  - Assuming that players choose strictly according to the strategy indicated by the analysis of absolute data at any time, the answer must be found in a few guesses. However, one reason why the game attracts a large number of players is the limitations and randomness (psychological factors, etc.) of people's choice in the actual process of playing the game, which ensures the fun of the game, People often solve the Wordle puzzle not by systematic training but by intuition formed by their own experience.

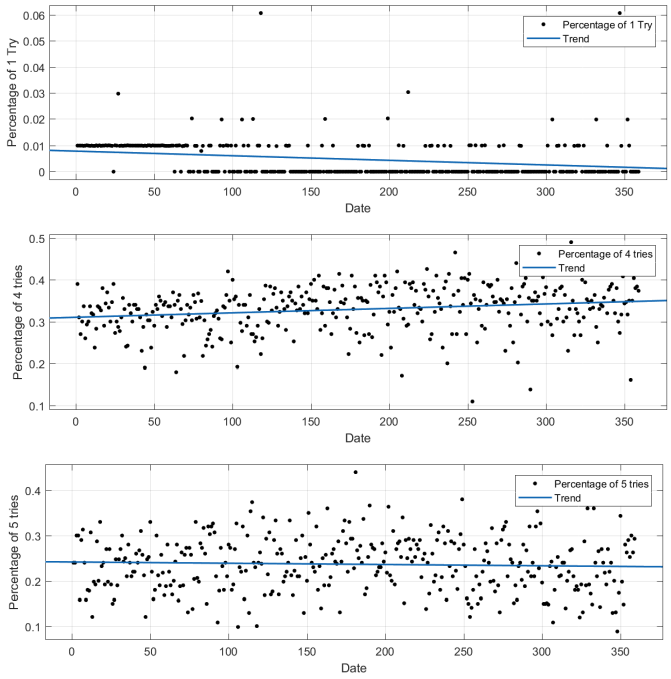


Figure 16: Trend of Tries

8 Sensitivity Analysis & Model Analysis and Test

We have made sensitivity analysis on the change of word classification and the prediction distribution of the result of the attempt under the condition that the different attributes of the word remain unchanged. An example of changes in initial letter is shown in Table 10 The following are the corresponding initial letters, semantic direction, easy to confuse, special structure, and number of letters of each line. The results are in Table 11.

Table 10: A Group of Word that Have different Initial Charater

match	watch	catch	batch	hatch	patch	latch

Table 11: Notations used in this literature

K-means Clustering	Final Division	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 tries
0.57	0.57	0.02	6.22	49.58	15.02	8.05	17.22	4.26
1.80	1.20	0.15	12.02	60.83	18.04	34.98	31.36	6.09
0.13	1.36	0.03	10.21	71.49	26.50	9.07	20.37	5.85
1.07	1.20	0.13	10.40	53.53	7.29	27.41	26.92	4.55
1.07	1.20	0.19	13.13	42.45	15.44	65.28	41.98	6.50

When the given parameter changes by 20%, the corresponding words in 3 attempts and 4 attempts have a relatively large and sensitive change, while the rest have a small change, and the prediction result is relatively stable.

9 Evaluation, Improvement and Extension of The Model

9.1 Evaluation

9.1.1 Strengths

- The requirements of the problem are classified, and the model is established for each small problem to solve, the output of one model as the input of other models, each model follows the principle of "high cohesion, low coupling".
- The neural network model adopts one MAML method – small sample learning to train, which has a good effect on small sample data processing.
- When estimating the difficulty of words, three algorithms are used to analyze and integrate respectively, and the results have high universality and credibility.

9.1.2 Weaknesses

- The data set given in the topic is too small, and it is difficult and risky to use learning algorithms such as neural networks, which may lead to prediction bias.
- Some attributes of words are abstract, so it is difficult to determine a unified measurement standard. Classification is usually accompanied by strong subjectivity, which leads to the deviation of results.

9.2 Further Improvements

- For neural networks, the interpretability is relatively poor compared with the traditional methods. In the future, if the model mechanism can be more accurately modeled from the perspective of graph theory, and more interpretable factors can be added, it will be easier to analyze the data relationship and establish a more accurate model.
- In the future, we can add some attributes that are more likely to reflect the difficulty of guessing words to the model from the perspective of psychology and game theory, which may further improve the accuracy of the model.
- The user's nationality, preference, age, education, and experience accumulated over time, as well as the level of serious participation in the game, can affect the prediction of the number of attempts in question 2 to a certain extent. If we can obtain user information in the future, we can get a more accurate and comprehensive model by modeling users. However, in this problem, due to the current information acquisition and platform, we cannot obtain more information about users and can only treat it as a random variable, which limits the improvement of model accuracy to a certain extent.

10 A Letter to the Puzzle Editor of the New York Times

Dear Sir or Madam: We've been informed that you need analyses for the results of Wordle Game in the past year. According to our analysis of number of reported results, number in hard mode and distribution of tries, we summarize three model to solve the problems, which are Time Sequence Model using ARIMA or LSTM Model to the predict number of reported results in

March 1, 2023, Few-shot Learning Words Model to research distribution of the reported results, and synthesis of K-means++ clustering, Kernel density estimation, Random forest, difference evolution for SVM to score and rank a given word. We also use Spearman correlation coefficient and multiple regression method to analyse word attributes affecting results of hard mode and distribution of tries, TOPSIS weight calculation model based on entropy weight method to process attributes of words. Below is some useful information.

First of all, we use ARIMA model and LSTM model to fit the curve of number of reported results, drawing a conclusion that the number of reported results will be about 18600 , in the range of $[16064, 25086]$. Attributes of words that significantly affect the percentage of scores reported that were played in Hard Mode include proportion of vowels, the number of different letters, the number of syllables, and the word frequency score.

Second, We use the neural network to predict the distribution of the number of attempts under the specified word and date by multiple regression of the word attribute and the number of participants predicted according to the time information. For small samples, we use the **few-shot learning** method, use the **MAML algorithm** to train the neural network, and select a relatively simple model structure to effectively alleviate the over-fitting and obtain a small loss.Compared with traditional neural network training methods and multiple nonlinear regression, this model has excellent effects on small sample data in terms of link over-fitting, shortening training time, improving accuracy, etc. The point estimation and inter-area estimation (95% confidence level) of the distribution of the number of attempts predicted by the word ERRIE on March 1, 2023 are shown in the Figure.17.

Figure 17: Point estimation (left figure) and interval estimation (interval estimation) for the distribution of the number of attempts predicted by the word ERRIE on March 1, 2023.

Third, We use TOPSIS model based on entropy weight method to comprehensively process the data, obtain the weight of each guess number, and realize the difficulty classification of words by K-means++ clustering algorithm. According to the multiple regression analysis model, we determined the word attributes significantly correlated with the classification: the promotion of words, semantic direction and which it is easy to cause fusion. Through the combination evaluation of random forest, kernel density estimation and support vector machine based on differential evolution, the most likely difficulty level of EERIE is 2, and the confidence level is 69.64%.

In the end, we find that the data shows a game between Wordle words and gamers, while getting correct answer in one guess may be more difficult, gamers are train to be more skillful to solve a word in less guesses in general.

We hope that our solutions might help.

Best regards, MCM team # 2305302

References

[1] Finn C, Abbeel P, Levine S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks[J]. PMLR, 2017.

[2] Lin Lianlei, Jiang Shouda, Liu Xiaodong SVM parameter selection based on differential evolution algorithm [J] Journal of Harbin Engineering University, 2009,30 (02): 138-141+159

[3] N. De Silva, "Selecting Optimum Seed Words for Wordle using Character Statistics," 2022 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 2022, pp. 1-6, doi: 10.1109/MERCon55799.2022.9906176.

Appendices

Appendix A MAML Algorithm

Algorithm 1 Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Require: $p(T) : distributionovertasks, \alpha, \beta : stepsizehyperparameters$

randomly initialize θ
while not done **do**
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
end while
for all $\mathcal{T}_i$ **do**
    Sample K datapoints $\mathcal{D} = \left\{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\right\}$ from $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(f_\theta\right)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$
    Compute adapted parameters with gradient descent:
    $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(f_\theta\right)$
    Sample datapoints $\mathcal{D}_i' = \left\{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\right\}$ from $\mathcal{T}_i$ for the meta-update
end for
Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right)$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$

Appendix B Inputs of Few-shot Learning Words Model

Table 12: Inputs of Few-shot Learning Words Model

name	N_{total}	N_{hard}	word frequency	proportion of vowels	initial	part of speech
value	18600	1024	2.44	0.8	4	1