Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2303707

A Wordle Analyzer in Multi Aspects

Summary

This paper mainly establishes the S-curve regression model, Spearman correlation coefficient analysis model, probability distribution prediction model, and k-means cluster analysis model to solve the related problems of Wordle.

First, we preprocess the attachment data, including removing misspelled words (we can't look them up anywhere); removing the abnormal number of reported results; reordering data by date.

For the first question, we use the data after Wordle's popularity is stable for analysis. We use an S-curve regression model to fit the number of reported results on the given date and then use the model to predict the number of reported results on March 1,2023. The results of ANOVA show the reliability of our predicted data. We looked up several attributes of words, and then conduct Spearman correlation tests with each of these word attributes and the percentage of scores reported that were played in Hard Mode. We use the significance test to verify the reliability of the Spearman correlation analysis.

For the second question, we construct the training flow WordleNet by combining meaningful and intuitive embeddings together, and then putting them into an improved MLP network, which is more robust against overfitting and vanishing gradient problems. Addition, we introduce two techniques, Monte Carlo Dropout and Deep Ensembles, to create two variations of one prediction. With the variations we can effectively quantify the uncertainties, and transfer them to two confidence indicators. The overall framework was named as SOUL, which prove to be effective and confident of its predictions.

For question 3, we use the k-means cluster analysis model to classify the solution words. In order to improve the accuracy of k-means clustering analysis as much as possible, we also use the elbow algorithm and K-means ++ algorithm to find the optimal number of clusters and the optimal initial cluster center respectively. We conduct a sensitivity analysis of the k-means clustering analysis model and found that the randomness of the k-means algorithm has little impact on the clustering results, which indicates that our classification model is relatively stable. We use the usage frequency of words and the number of syllables studied in question I to analyze the word features of different categories, and find that different categories of words have relatively obvious differences in frequency and number of syllables, which further verifies the rationality of our classification.

For question four, we count the frequency of each letter appearing in all words, which decides on what we think of first. We also perform *Case Processing* on the number of reported results to get an intuitive understanding of the attributes of the data.

Keywords: S-curve Regression; Spearman's Correlation Test; K-means Cluster Analysis; Multi Layer Perceptron, Dropout

Contents

1 Introduction 3
1.1 Previous research 3
1.2 Restatement of the Problem 3
1.3 Our Work 4
2 Assumptions and Symbol Description 4
2.1 Assumptions 4
2.2 Notations 4
3 Problem Analysis and Model Overview 5
3.1 Analysis of the problem 1 5
3.2 Analysis of the problem 2 5
3.3 Analysis of the problem 3 6
3.4 Analysis of the problem 4 6
4 Model Establishment and Solution 6
4.1 Data preprocessing 6
4.2 Time series Forecasting Models based on nonlinear regression 6
4.3 SOUL: MLP Ensembles of Multi Levels 11
4.4 K-means clustering analysis model 17
4.5 Other features of this data set 22
5 Model Evaluation and Improvement 23
5.1 Model Strengths 23
5.2 Model Weaknesses 23
5.3 Model improvement 23
6 Conclusion 23
7 Letter to the Puzzle Editor 24
References 25

1 Introduction

1.1 Previous research

The topic proposes a variety of problems such as time sequence analysis, correlation analysis, machine learning prediction, clustering analysis.

For the problem of time series analysis, the predecessors gave countless examples. For example, Qin Pan built a Multi-Att-LSTM prediction model based on multi-head self-attention mechanism and multi-characteristic variables and combined with LSTM to train and learn the concentration of industrial pollution NOX in the future moment[1]. Yating Gao established the maximum related entropy self -regression model for stock price prediction and so on[2]. In the study of correlation issues, NJ Gogtay[3] et al. clearly explained the principle and usage scenarios of Pearson's correlation coefficient and Spearman's correlation coefficient in their article, which laid a good foundation for the application of this paper.

Additionally, paper[4] carefully explains the characteristics and mathematical theory of BP neural network algorithm, and discusses the shortcomings and improvements of BP algorithm. The paper[5] concludes the benefits and costs of using Monte Carlo dropout in neural networks through experiments and analysis. And the paper[6] systematically describes the theoretical background of deep ensemble learning, traditional methods and recent new methods in detail, and reviews the application of deep ensemble models in different fields. All are important references for the main model in this paper.

For the clustering problem, DT Pham[7] et al. mentioned the K-means algorithm in their paper and proposed the method of selecting the appropriate K, which provides a very good reference for the application of this paper.

1.2 Restatement of the Problem

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

- Interpreting and forecasting the number of reported results is a typical time series analysis and forecasting problem. A suitable regression model is needed to fit the sequence and predict some data in the future. To explore the influence of word attributes on the percentage of scores reported that were played in Hard Mode is also a correlation analysis problem, which requires us to find out the word attributes that are correlated with the percentage of scores reported that were played in Hard Mode.
- Predicting the distribution of the reported results is similar to a regression task. Given just a word like *EERIE*, we need to find its embedding with sense of word and intuitive features. Then we should use a function, typically a neural network, to fit the mapping relationship between embeddings and target distributions. To find the uncertainties of the model, we should find the uncontrollable or unknown factors causing the inherent noise of data. Finally, a strategy should be proposed to define the confidence of a prediction and work it out.
- Since no specific difficulty labels are given, this is a typical unsupervised classification problem, and the key is to find the data that reflects the difficulty of the solution words.

We also need to identify the attributes of the words in each category, and compare the attributes of words in different categories. Finally, we will discuss the accuracy of the classification model and analyze the specific effect of classification.

- Finally we need to dig deeper and analyze the dataset to find patterns and features that we haven't explored before.

1.3 Our Work

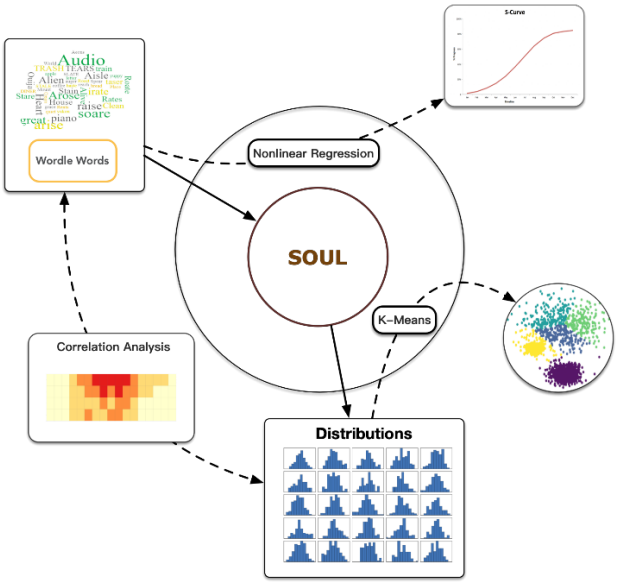

Figure1: The sketch map of our work

2 Assumptions and Symbol Description

2.1 Assumptions

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

1. **In the future, the rules of the game will not change.** The change of gameplay or game planning activities will widely affect the willingness of players to play games. This unpredictable event will affect our prediction of future data.

2. **The change in the number of players in this puzzle game is similar to that of general stand-alone games,** that is, the number of players will increase rapidly in the early stage, and then gradually decreases, and finally tends to stabilize. This excludes the possibility of a fierce change in the number of players in the future, which can ensure the rationality of our prediction data.

3. **The factors not mentioned doesn't influence the data distribution directly.** This assumption helps to reduce the complexity of our model.

2.2 Notations

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

Table 1: Notations used in this paper

Symbol	Description
x	Date
y	number of reported results
δ	MSE of Regression fitting
r^s	Spearman correlation coefficient
R^2	Correlation coefficient of regression equation

3 Problem Analysis and Model Overview

3.1 Analysis of the problem 1

Forecasting the number of reported results on March 1, 2023 is a time series analysis problem. Generally, linear or nonlinear functions are used for fitting, so as to predict the future data. Visualizing the Number of reported results over time in the appendix shows that in the early days, the number of players and the number of results obtained increased rapidly as the game became more popular. Subsequently, with the boredom of a large number of players, the number of results obtained each day showed a significant decrease trend, and finally had a trend of nearly stable. The topic requires us to give a prediction value of the future time. It can be seen from the data chart that only a small part of the data in the early period increases sharply, while the majority of the data in the later period shows a monotonic trend. The data with a large change range in the early period has no positive reference value for the prediction of the future data, and should be ignored in the analysis. Based on the above considerations, we can first truncate the data before the peak value, and then build a nonlinear regression time series analysis model, so as to predict the target data. According to the characteristics of the data, this paper uses S-curve regression model to fit the data.

Explore whether there are certain attributes of words that can affect the percentage of scores reported that were played in Hard Mode. This is clearly a correlation analysis problem. We analyzed the frequency of word use, the number of syllables, the number of different letters, and the number of vowels, and correlated these attributes with the percentage of scores reported that were played in Hard Mode. If there is a correlation, it would indicate that the attributes of the words we find affect the percentage of scores reported that were played in Hard Mode to some extent. spearman's correlation coefficient was used for analysis.

3.2 Analysis of the problem 2

For the word dataset, it’s hard to find effective embeddings from single source, either intuitive embeddings or meaningful embeddings prove to be incomprehensive. So we construct the training flow WordleNet, combining embeddings together and putting them into an improved MLP network, which is trained robustly against overfitting and vanishing gradient problems. In Addition, we introduce two techniques, Monte Carlo Dropout and Deep Ensembles, to create two variations of one prediction.With the variations we can effectively quantify the uncertaintess, and transfer thems to two confidence indicators, separately for the two techniques. The overall framework was named as SOUL, which prove to be effective and confident of its predictions.

3.3 Analysis of the problem 3

When dividing the solution words by difficulty, because the problem does not give the specific difficulty label and the number of classes, only provides the percentage of times used to complete the puzzle, so this problem needs to use the unsupervised clustering learning method, and the percentage of times used to complete the puzzle is used to represent the difficulty feature of the solution word. We adopt the most common k-means clustering algorithm to classify the answer words. We also use the elbow algorithm to select the optimal number of clusters, and use the K-means ++ algorithm to select the optimal initial clustering center, and then carry out the centralization iteration to divide the sample set into several disjoint clusters as the final classification result.

Further, we compare and analyze the mean value of attributes of words in each category. The attributes used here are the ones we found in the first question. Finally, we evaluate the accuracy of k-means clustering model by calculating its CH.

3.4 Analysis of the problem 4

We need to mine some other interesting features from the given data. There are many options. We can count the frequency of letters in words, so we can use this as a basis to determine which letters should be used more and which letters should be used less when playing games. We can also analyze the concentration dispersion of the data using case processing.

4 Model Establishment and Solution

4.1 Data preprocessing

In observing the overall situation and preliminary processing of the data, we found that the data has error or abnormal presence. We have done the following processing:

1.We found that there are very few words in the case of spelling errors and cannot be used for subsequent analysis, so deleted.

2.The data given is based on the edge of the time, which is not conducive to analysis, so we sort the data in the order of time.

3.For very individual data (such as data corresponding to Study), there are obvious abnormalities from the observation of adjacent data groups, and we remove it as an abnormal value.

The total number of final data is 355.

4.2 Time series Forecasting Models based on nonlinear regression

4.2.1 Model Description

We need to forecast the number of reported results on March 1, 2023. The first is the choice of data. As mentioned earlier, the figure below gives evidence:

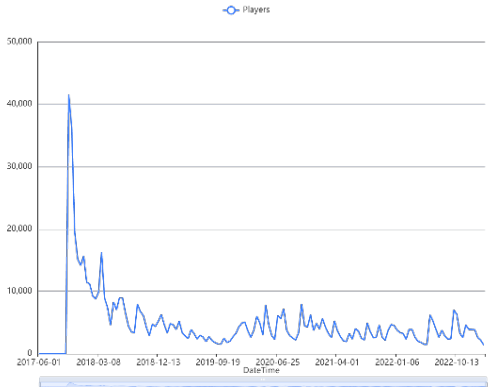

Figure 2: Players of Assassin's Creed Origin

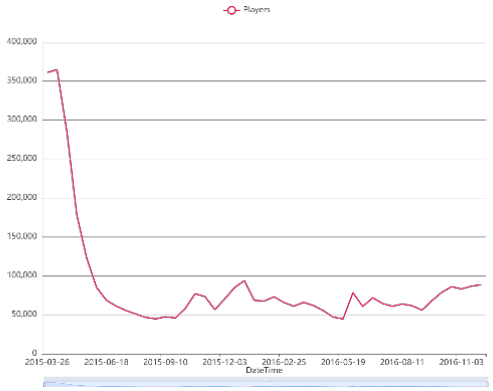

Figure 3: Players of Grand Theft Auto V

The above data comes from SteamDB. The number of players in the early days of the game is affected by the game's release and the changes in the game's heat. To make a reasonable prediction of future data, the data in the stable period is required.

From the perspective of data images, the data is not linear shape, so non -linear functions should be used for fitting. The common regression equations are: secondary curves, composite curves, growth curves, pairing curves, S curves, index curves, reverse functions, and power functions. Here we use the s-curve to fit according to the characteristics of the image.

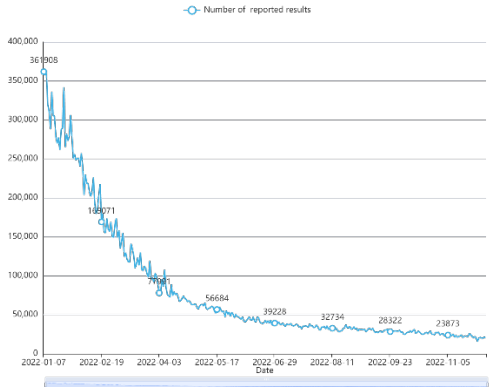

Figure 4: Cut data

For the attributes of the words that can affect the game, the first is the usage frequency of words. If a word is almost not used, it is obviously more difficult to be remembered in the game. Our data of the usage of words comes from Google Books Ngram Viewer. We calculated the average frequency of words from 2000 to 2019. Followed by the number of different letters in words. The length of the word of the game is fixed to 5. If there is a duplicate letter, it may affect the search scope of the player when playing. Then there is the number of syllables. For a word, the word with two syllables is much more than the number of words with only one syllable, and the difficulty is obviously different. In the end, we also considered the influence of vowel letters, and the sensitivity of people's sensitivity to vowel letters and consonants is very different. Therefore, the correlation analysis of the above four attributes and the percentage of scores reported that were played in Hard Mode can obtain the word attributes that affect the percentage of scores reported that were played in Hard Mode.

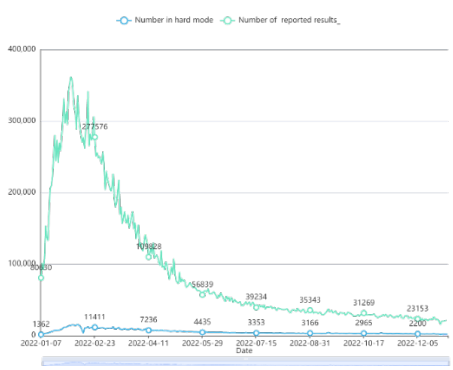

Figure 5: Comparison data

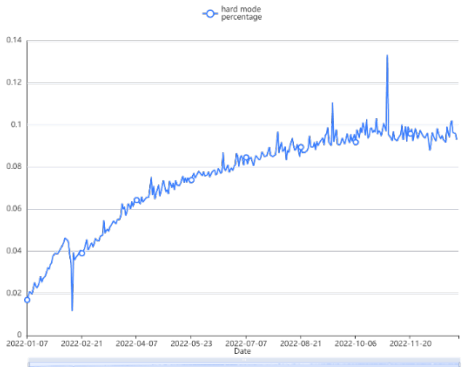

Figure 6: Percentage data

In terms of data choice, it can be seen from the figure that the first half is rising with the popularity of the game, and a large number of players are pouring in. Most of these players have the mentality of trying to play games. They do not completely consider whether they want to choose difficult mode to play, so that they only play ordinary models. This has a great impact on the percentage of scores reported that were played in Hard Mode. If we want to explore the impact of the word attributes on the percentage of scores reported that were played in Hard Mode, the data when the number of players is not stable will be unavailable.

4.2.2 Model Calculation

We used the S-curve regression model to fit the number of reported results.

$$y = e^{\beta_0 + \beta_1 / x} \tag{1}$$

where β_0 and $\beta 1$ are undetermined coefficient of the regression equation. We then transform the variables to convert the nonlinear equation into a linear equation, which is convenient for regression calculation:

$$y_1 = \beta_0 + \beta_1 x_1 (y_1 = \ln(y), x_1 = 1/x) \tag{2}$$

where x_1 and y_1 are respectively the transformation variables of x and y. We take the minimum equity error as the fitting target:

$$\delta = \sum_{i=1}^{n} (\hat{y_i} - y_{1i})^2 \tag{3}$$

$$\frac{\partial \delta}{\partial \beta_0} = 0 \tag{4}$$

$$\frac{\partial \delta}{\partial \beta_1} = 0 \tag{5}$$

where $\hat{y_i}$ is the predicted value given by the regression equation. δ reflects the gap between the regression equation prediction value and the real value. We take the derivative of this equation, so that the derivative equation is equal to 0. If there is a unique solution, the parameter when delta is minimum, that is, when the regression equation is optimal, can be obtained. The solution results are as follows:

$$\beta_1 = \frac{\sum_{i=1}^{n} x_{1i}\, y_{1i} - n\overline{x_1}\,\overline{y_1}}{\sum_{i=1}^{n} x_{1i}^2 - n\overline{x_1}^2} \tag{6}$$

$$\beta_0 = \overline{y_1} - \beta_1 \overline{x_1} \tag{7}$$

where $\overline{x_1}$,$\overline{y_1}$ are respectively the mean value of x_1 and y_1. We substitute the parameters into the regression equation for fitting, and the results are as follows:

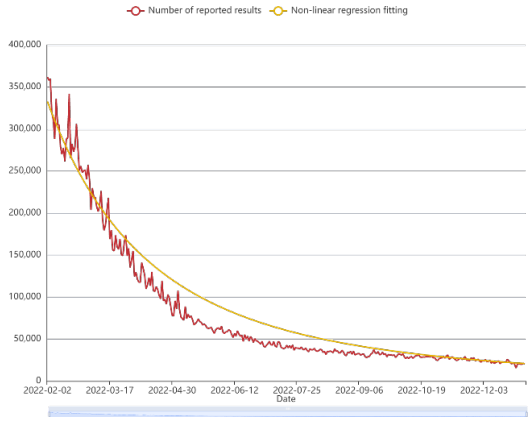

Figure 7: Fit chart

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	209.385	1	209.385	21891.356	0.000
Residual	3.137	328	0.010		
Total	212.522	329			

Figure 8: ANOVA

We take the mean square error as the possible error of the predicted value, and give the range of results obtained on March 1, 2023 as [15403,15431].

We use spearman's coefficient to analyze the correlation between the percentage of difficult players and each attribute of a word. Spearman's correlation coefficient is used to evaluate whether variables tend to change at the same time, that is, to explore whether the percentage of scores reported that were played in Hard Mode increases or decreases when the value of a certain attribute of a word increases. Spearman's correlation[8] coefficient formula is as follows:

$$r_s = 1 - \frac{6\sum s_i^2}{n(n^2-1)} \tag{8}$$

where n denotes the amount of data，s_i denotes the difference between two data's orders：

$$s_i = rg(X_i) - rg(Y_i) \tag{9}$$

where X and Y represent any two sequences of data ，$rg(X_i)$ denotes X_i's order in it's sequence.

First, all the data are used for analysis. The correlation heat map is as follows:

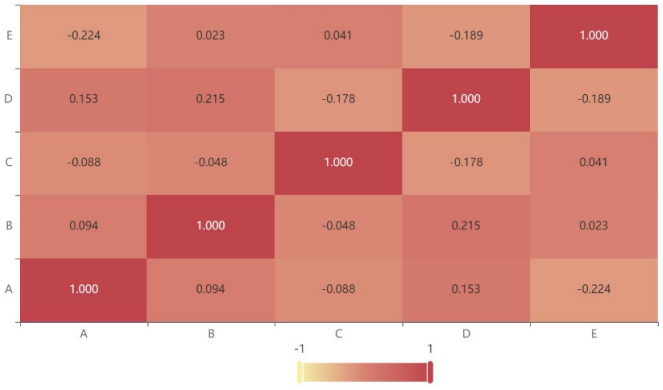

Figure 9: Heat map1

For Spearman's coefficient less than 0.2, we believe that there is no direct relationship between variables. It can be seen that only word usage frequency is related to the proportion of hard mode players. However, no obvious correlation was found for other attributes which had

a certain effect. The data itself is biased, so we should ignore the data when the number of game players is not stable.

The data of the latter half of the game when the number of players is stable is analyzed, and the correlation heat map results are as follows:

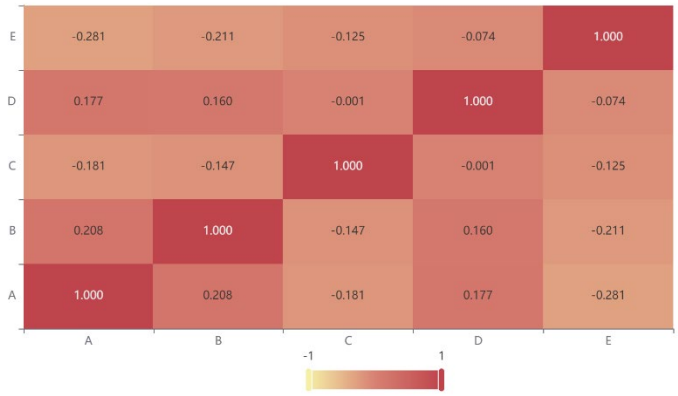

Figure 10: Heat map2

A：the percentage of scores reported that were played in Hard Mode，B：number of syllables，C：number of different letters，D：number of vowel letters，E：usage frequency of words

As can be seen from the chart above, both the number of syllables and usage frequency of words are related to the percentage of scores reported that were played in Hard Mode. We can assume that the number of syllables and usage frequency of words will affect the percentage of scores reported that were played in Hard Mode. Moreover, the percentage of scores reported that were played in Hard Mode is negatively correlated with usage frequency of words and positively correlated with number of syllables. We infer that when words are used more frequently and syllables are less, players are more likely to play in normal mode, and vice versa.

4.2.3 Model Evaluation and testing

We calculated R value and R^2 value to evaluate the fitting results of the S-curve regression model. R value is easily affected by the amount of data, and we took R^2 as the main reference value, which was defined as:

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{10}$$

That is, how much of the variance of the variable is explained by the predicted value, and how well the predicted value fits the true value is measured. The closer it is to 1, the better the fitting effect. The calculation results are as follows:

R	R Square	Adjusted R Square	Std. Error of the Estimate
0.993	0.985	0.985	0.098

Figure 11: R square

Our result is 0.985，indicating a good effect.

In order to evaluate the rationality of spearman's correlation analysis, we conducted significance test, and the results were as follows:

Table 2: Significance test1

	A	B	C	D	E
A	1(0.000***)	0.208(0.023**)	-0.181(0.049**)	0.177(0.054*)	-0.281(0.002***)
B	0.208(0.023**)	1(0.000***)	-0.147(0.110)	0.16(0.081*)	-0.211(0.022**)
C	-0.181(0.049**)	-0.147(0.110)	1(0.000***)	-0.001(0.994)	-0.125(0.177)
D	0.177(0.054*)	0.16(0.081*)	-0.001(0.994)	1(0.000***)	-0.074(0.425)
E	-0.281(0.002***)	-0.211(0.022**)	-0.125(0.177)	-0.074(0.425)	1(0.000***)
Note: ***, **, and*represent the significant level of 1%, 5%, and 10%, respectively					

Clearly, the number of syllables and the usage frequency of words both pass the test, reaching the 5% significance test level, and we are 95% confident that these two attributes are related to the percentage of players in hard mode.

4.3 SOUL: MLP Ensembles of Multi Levels

4.3.1 Overall Framework

Backpropagation (BP) algorithm is an optimization technique commonly used for training artificial neural networks. Combined with different activation functions and loss functions, it's adaptive to different areas including the *natural language process*(NLP) to which our task is partially relevant .

However, the BP algorithm is known to suffer from issues such as overfitting and the vanishing gradient problem, which can limit its performance in some degrees. Due to the intrinsic noise of data, some of which derives from factors like variations of web users' Concerns, problems mentioned above may be magnified and more tough to solve. Additionally, some unreliable solutions may be given by the traditional BP-based neural network, which may contribute to fatal consequences in specific cases.

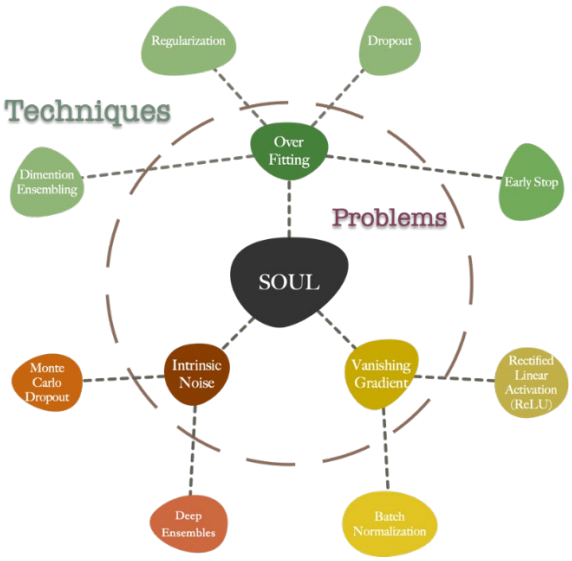

Figure 12: Techniques we used to solve problems existing in MLP training

In this section, we address the aforementioned limitations by proposing a simple yet effective MLP framework, namely, SOUL(enSembles Of mUlti Levels).

To prevent over-fitting, we utilities techniques, Specially, **Dropout**, **Early Stopping**, **Regularization** and **Dimension Ensembles**, which have been proved effective on most neural network framework, including MLP framework. When it comes to vanishing gradient problem, we put **Batch Normalization** into the SOUL framework, and select Activation functions with non-saturation properties (e.g., **ReLU**) to accept the outputs of fully connected(FC) layers.

To make the model more reliable under the noise existing in the given data, firstly, we give some assumptions about potential factors causing performance degradation of classic model and unavoidable noise problem, which can be concluded into **Epistemic Uncertainty** and **Aleatoric Uncertainty** separately. Following the assumptions, We verified that the resulting damages do exist, by displaying the correlation between the recorded distributions(labels of model outputs) and some irrelevant or uncontrollable factors. Finally we use two techniques, Monte Carlo Dropout and Deep Ensembles, to combining the model outputs from multiple forward passes, which can precisely quantify the uncertainty associated with our model and predictions, with which we can give a confidence indicator accompanied with a prediction.

4.3.2 WordleNet: An Effective Training Flow

The BP algorithm consists of two phases: the forward pass and the backward pass. In the forward pass, the input is propagated through the network, and the output is computed. In the backward pass, the error signal is propagated backward through the network, and the weights and biases are adjusted to reduce the error.

Here we give the training flow WordleNet, The overall framework is illustrated in Figure 14.

Constructing the Embeddings. We transfer all of recorded words into embeddings, including Frequency Embeddings and Meaning Embeddings.

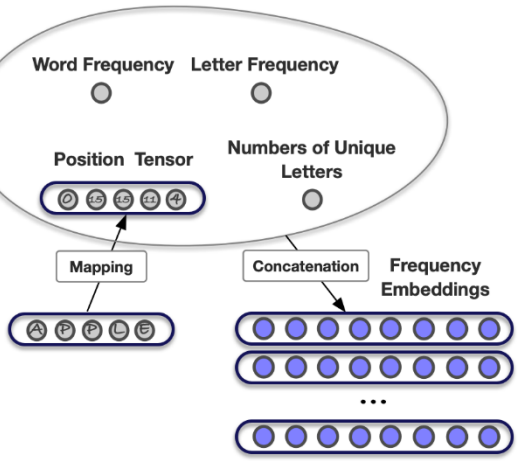

Figure 13: Embeddings

Intuitively, the difficulty of a word puzzle game is closely related to the usage frequency of given words. In more depth, the usage of frequency, in terms of a word, can be divided into frequency of the word and letters it contained. In addition, the number of unique letters in the word can be another determinant, resulted from the game mechanics that a correctly guessed

letter will be indicated in green or forced to be used in the next guess. A 5-dimension tensor indicating letters' positions are also added on. We put the attributes mentioned above into 8-dimension Frequency Embeddings.

$$FreqEmb\left(word^{(i)}\right) = f^{(i)}, \tag{11}$$

where $dim\left(f^{(i)}\right) = 8$.

To make our embeddings more comprehensive, we import the pre-trained words embeddings GloVe as Meaning Embeddings, which have been proved effective on Capturing lexical semantics and Handling of rare words.

$$MeanEmb\left(word^{(i)}\right) = m^{(i)} \tag{12}$$

We then concatenate the embeddings accessed from two aspects into a longer tensor, followed by a normalization operation. By this way, we get intuitiveness-generalization balanced embeddings for the following training process:

$$e^{(i)} = cantenate\left(m^{(i)}, f^{(i)}\right), \tag{13}$$

$$e^{(i)} = \frac{e^{(i)}}{\max\left(\left|\left|e^{(i)}\right|\right|_{\mathrm{p}}, \epsilon\right)}, \tag{14}$$

where p is the exponent value in the norm formulation, which is set as 2 in this paper, $e^{(i)}$ represents the initial embedding of i-th word.

Forward Propagation. Once the embeddings of words are constructed, we can calculate the hidden layer output H of the embeddings. Then we feed the output into a activation function and dropout module. Finally the 7-dimension outputs can be calculated by:

$$\boldsymbol{H_0 = e} \tag{15}$$

$$\boldsymbol{H_{l+1} = f}(H_l W_l), \boldsymbol{l} = 0,1,\ldots,\boldsymbol{L} \tag{16}$$

$$\boldsymbol{H_k = Dropout}(\boldsymbol{H_k}, \boldsymbol{prob}), \tag{17}$$

$$\boldsymbol{H = H_L}, \tag{18}$$

where H_l represents the output of the l-th layer, L represents the numbers of layers, f represents the rectified linear Unit(ReLU), W_l represents the weights of the l-th FC layer, k represents the special layer of which we randomly zeroes the outputs with probability $prob$. It is noteworthy that no activation function is added during the calculation of the last layer output H_L. Here we set $L = 2, k = 1, prob = 0.01$ as default.

Predicting the distributions. Finally we will use Softmax to re-scale outputs so that the elements lie in the range [0, 1] and sum to 1.

$$H^{(i)} = \left(H_1^{(i)}, H_2^{(i)}, \ldots, H_7^{(i)}\right) \tag{19}$$

$$d^{(i)} = Softmax\left(H_c{}^{(i)}\right) = \frac{exp\left(H_c^{(i)}\right)}{\sum_j exp\left(H_j^{(i)}\right)}, c = 1,2,\ldots,7 \tag{20}$$

Calculating the losses. When we finish predicting the distributions d for the words set, we

can calculate the KL-divergence loss between the results d and the recorded distributions on try times, y by:

$$L = \{l^{(i)}, \dots, l^{(N)}\}^{\top}, \tag{21}$$

$$l^{(i)} = -\sum_{c=1}^{7} y_c^{(i)} \left(log\ y_c^{(i)} - log\ d_c^{(i)} \right), \tag{22}$$

$$l(d, y) = \frac{\sum_{i=1}^{N} l^{(i)}}{N}, \tag{23}$$

Backward Propagation. With the losses calculated, we can calculate the gradients of the losses l with respect to the parameters $W_l, l = 0,1, \dots, L$, and use SGD optimizer to update the parameters W_l. The detailed process is implemented by pytorch AutoGrad, of which you can refer to https://pytorch.org/tutorials/beginner/introyt/autogradyt_tutorial.html for more information.

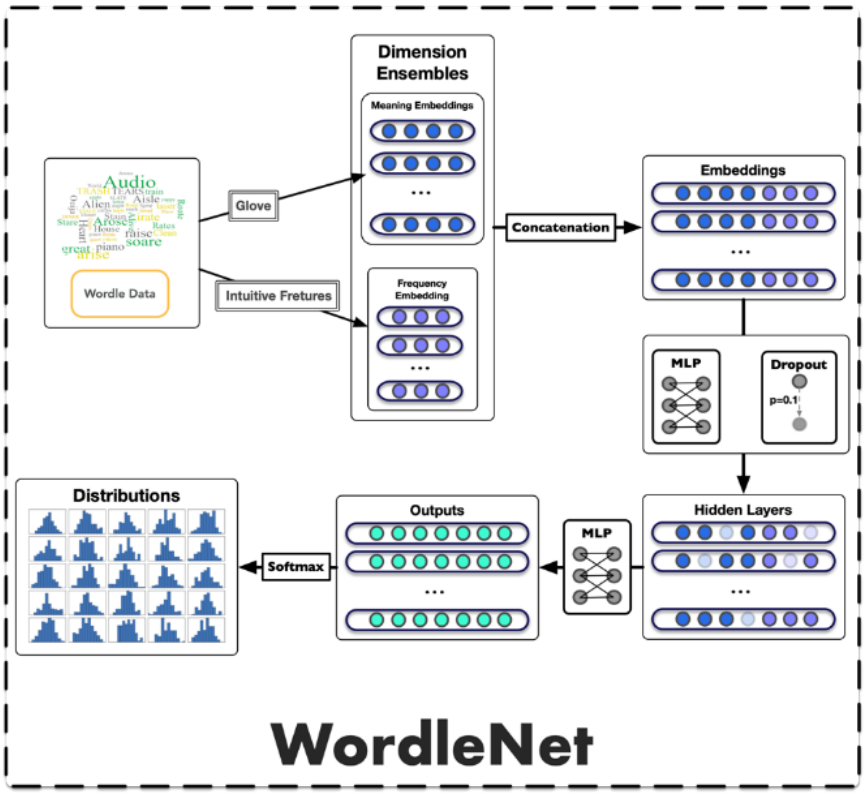

Figure 14: Framework of WordleNet. In the flow we add Dimension Ensembles, Dropout, etc. to prevent performance degradation caused by limitations mentioned in Figure 12

4.3.3 Uncertainty Calculator

A machine learning model's uncertainty can be divided into two types: **Epistemic Uncertainty** and **Aleatoric Uncertainty**, shown as Figure 15

Epistemic uncertainty is the uncertainty represented in the model parameters. This type of uncertainty is mainly caused by rare and over generalized training data. Embodied in our dataset, just over 300 samples recorded in a year are too thin to make the model convincing enough.

Aleatoric uncertainty captures inherent noise caused by unknowable or unobservable influencing factors. As mentioned in Figure 6, we find that the percentage of people playing Hard Mode was decreasing over time. The reason may be that as the game's popularity waned over time, most players chose to quit(showed in Figure 5), and only the minority who want to challenge themselves remain to play the Hard Mode. However, these are not our concerns, and it's hard to get the information about the correlations.

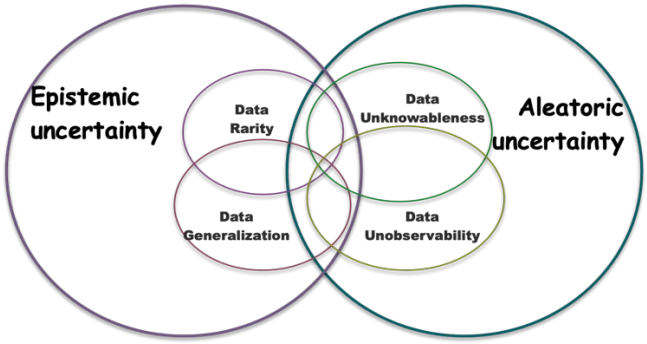

Figure 15: model's uncertainty

To quantify the uncertainty of our model WordleNet, we introduce two techniques, Monte Carlo Dropout and Deep Ensembles, to create variation of the model's outputs. We then calculate the sum of KL divergences between each output with the variation, respectively for the two techniques. Finally we transform these two indicators into interval 0~1, which represents the confidence of the model for the predicted results.

1. **Monte Carlo Dropout.** In the WordleNet, dropout layers are used as a regularization technique during training, which make the model more robust against overfitting. Usually, these dropout layers are disabled when evaluating on a new sample or new dataset. But for MC Dropout, whale we've been finished training the WordleNet, the dropout layers are still remained(shown in Figure 16), meaning neurons may still be randomly dropped out. By this way, we can create a variation of the outputs of the model.

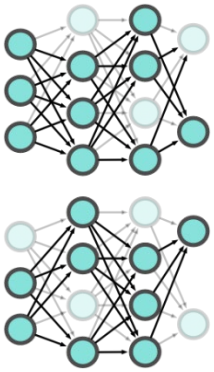

Figure 16: Monte Carlo Dropout

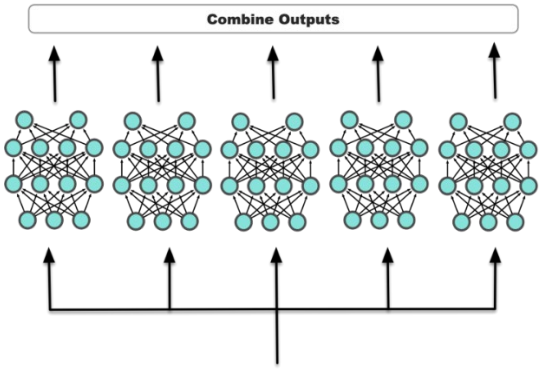

Figure 17: Deep Ensembles

2. **Deep Ensembles.** Deep Ensembles is another way to create variation of the model's outputs. Instead of training only one model and predicting multi times with it, Deep Ensembles aims to create multiple models with the same structure, initialize their weights randomly and train them on the same dataset repetitively. The framework of Deep Ensembles is shown in Figure 17.

3. **Uncertainty Calculation.** With the variable outputs combined, either by Monte Carlo Dropout or Deep Ensembles, we can calculate the uncertainties of the prediction through getting sum of KL divergences between each output with the variation. Here we give necessary notations on the combination and the variation of the output:

$$\boldsymbol{C}(\boldsymbol{h}) = \{\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \dots, \boldsymbol{h}^{(\boldsymbol{M})}\}^{\top}, \tag{24}$$

$$V(h) = \frac{\sum_{m=1}^{M} h^{(m)}}{7}, \tag{25}$$

where $h^{(m)}$ represents the m-th output in the combination $C(h)$.
Then we can formulate the **Uncertainty** and **Confidence** as:

$$U(h) = \sum_{m=1}^{M} KL(h^{(m)}, V(h)), \tag{26}$$

$$Confidence(h) = e^{-U(h)}, \tag{27}$$

Where $Confidence(h)$ is the re-scaled indicator which ranges from 0 to 1, meaning that the model is confident of its output when the $Confidence(h)$ is close to 1 and uncertain about its prediction when the $Confidence(h)$ is closer to 0.

4.3.4 Solution Given by SOUL

After Constructing the modules **WordleBP** and **Uncertainty Calculator**, we can combine them into a larger workflow SOUL, which can predict the distribution of try times accompanied with two confidence indicators, using just one word as input! The whole framework is shown in Figure 18.

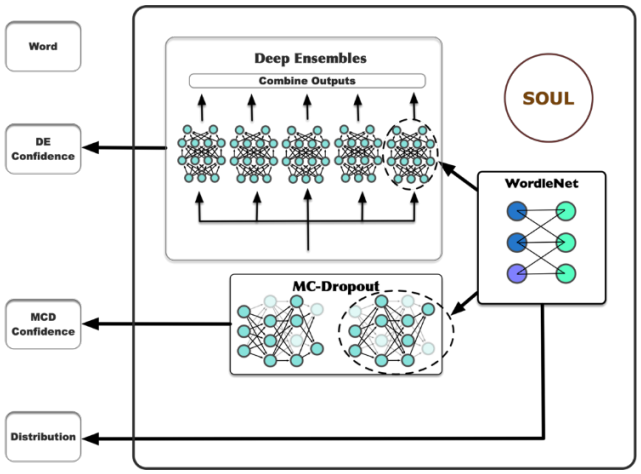

Figure 18: The overall framework

When we put the word EERIE into SOUL, we finally get predicted distribution:

(1, 6, 23, 31, 23, 12, 3),

and the confidence given by two calculators:

MC-Dropout Confidence: 93%

Deep Ensemble Confidence: 94%

The code can be accessed on https://github.com/LvAoAo/SOUL.

4.4 K-means clustering analysis model

4.4.1 k-means model construction

This question needs to classify the solution words according to the difficulty. We chose to classify them using the k-means unsupervised clustering algorithm. The main idea of the k-means algorithm is to divide the samples with close distance into a cluster, and at the same time make the distance between the samples of different clusters as far as possible.

We regard the percentage of guesses as 7-dimensional data to represent the difficulty of each solution word. In other words, we directly use the 7 percentages of guesses in the given data set to represent a sample point in the k-means clustering division.

First, randomly select k initial cluster centers $C_i(1 \leqslant i \leqslant k)$ from the data set, calculate the Euclidean distance between each sample and the cluster center C_i, find out the cluster center closest to the target sample, and assign the sample to the corresponding cluster. Then, calculate the mean value of the samples in each cluster as the new cluster center, and continue to iterate. Finally, when the cluster centers no longer change, the algorithm ends[9].

The Euclidean distance calculation formula between the sample and the cluster center is:

$$d(x, C_i) = \sqrt{\sum_{j=1}^{n} \left(x_j - C_{ij}\right)^2} \tag{28}$$

x is the sample data,C_i is the i-th cluster center, n is the data dimension of the sample, x_j and C_{ij} is the j-th attribute value of x and C_j respectively.

The cost function of the k-means algorithm is the sum of error squares (SSE), which is calculated as follows:

$$SSE = \sum_{i}^{k} \sum_{x \in C_i} |d(x, C_i)|^2 \tag{29}$$

The value of SSE indicates the quality of clustering results, and k is the number of clusters. In general, the k-means algorithm divides samples into k classes by iteration to minimize the sum of the distances between each sample and the center of its class.

4.4.2 Find the optimal number of clusters and the optimal initial cluster center

The selection of the k-value and the point of the initial cluster center will have a great impact on the convergence speed and clustering results of the k-means algorithm. We use elbow algorithm and k-means++ algorithm to obtain the optimal k value and the optimal k initial clustering centers.

According to the results of the elbow algorithm, draw the relationship diagram of the sum of squared errors(SSE) and the number of clusters k:

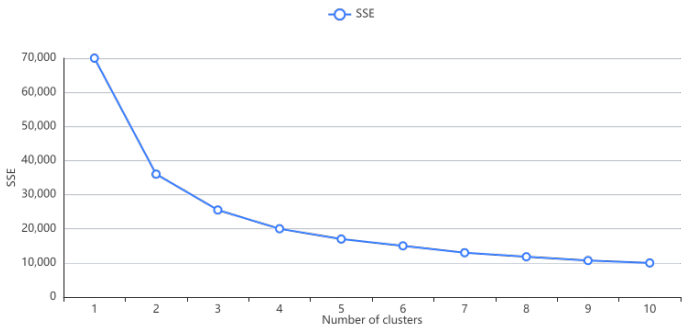

Figure 19: SSE-K

It can be seen from the above figure that as the number of clusters k increases, the SSE gradually decreases, and the degree of aggregation of each cluster gradually increases. When k is less than the optimal number of clusters, the increase of k will greatly increase the aggregation degree of each cluster, so the SSE will be greatly reduced. When k is greater than the optimal number of clusters, as k increases, SSE decreases slowly and tends to be flat, indicating that the improvement of classification accuracy brought by increasing the value of k at this time is very small. Intuitively, the relationship between SSE and k is in the shape of an elbow, and the k value corresponding to the elbow is the optimal number of clusters.

For the classification of solution words, we choose 4 as the optimal number of clusters after comprehensively considering the aggregation degree of clusters and the convergence speed of the algorithm.

When calculating k optimal initial cluster centers, first randomly select a sample as the initial cluster center. Then, gradually select the sample points that are far away from the determined cluster centers as new cluster centers, and repeat this process until k cluster centers are determined. The more dispersed the selected k initial cluster centers, the greater the difference between the final k clusters, which is beneficial to enhance the classification accuracy of the model.

4.4.3 Analysis of k-means clustering results

Because of the randomness of k-means algorithm, each clustering result may be different. Therefore, we repeated 500 experiments to compare SSE values. The final clustering result corresponding to the minimum SSE value is as follows:

Table 3: Cluster center

	Central 1 try	Central 2 tries	Central 3 tries	Central 4 tries	Central 5 tries	Central 6tries	Central 7 or nire trues(X)
Cluster1	0.371	6.220	26.742	35.902	21.311	8.015	1.280
Cluster2	0.286	2.714	12.018	24.714	28.536	22.571	9.089
Cluster3	0.241	3.509	18.009	34.371	28.112	13.371	2.448
Cluster4	1.404	13.000	33.981	31.154	14.808	5.000	0.808

We calculate the average value±the standard deviation of each variable in each cluster, and use F-test to detect whether there is significant difference between each variable in each cluster. We first assume that each variable has no significant difference between clusters, and

then use the F-test to calculate the significance P value of each variable. If the P value of a variable is less than 0.05, the original hypothesis is rejected, indicating that there is a significant difference between the categories classified by cluster analysis. The results of the difference analysis are as follows:

Table 4: Difference analysis

	Cluster category (mean ± standard deviation)				P
	Cluster1 (n=132)	Cluster3 (n=116)	Cluster2 (n=56)	Cluster4 (n=52)	
1 try	0.371±0.485	0.241±0.45	0.286±0.456	1.404±1.376	0.000***
2 tries	6.22±1.895	3.509±1.717	2.714±1.806	13.0±3.961	0.000***
3 tries	26.742±2.931	18.009±2.864	12.018±4.127	33.981±3.416	0.000***
4 tries	35.902±3.455	34.371±3.992	24.714±4.241	31.154±3.089	0.000***
5 tries	21.311±2.269	28.112±2.945	28.536±6.033	14.808±2.352	0.000***
6 tries	8.015±2.111	13.371±2.909	22.571±4.062	5.0±1.597	0.000***
7 or more tries (X)	1.28±0.911	2.448±1.274	9.089±7.403	0.808±0.627	0.000***

Note：***、**、* represent the significance level of 1%, 5% and 10% respectively

The result of the difference analysis shows that the significance P value of the seven guess times variables is significantly less than 0.05, which is significant at the level, indicating that the seven variables have significant differences between the categories classified by the cluster analysis, and the classification is effective.

In order to further evaluate the accuracy of k-means clustering analysis, we calculated the CH (Calinski-Harbasz Score) of the clustering results. The calculation method is:

$$CH = \frac{SS_B}{k-1} / \frac{SS_W}{N-k} \tag{30}$$

k is the number of clusters, N is the number of all data, SS_B is the between-class variance, and SS_W is the intra-class variance.

Calculate SS_B:

$$SS_B = tr(B_k) \tag{31}$$

B_k is the between-clusters dispersion mean:

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{32}$$

n_q is the total number of data points of class q,c_q is the mass point of class q, and c_E is the center point of all data.

The name Calculate SS_W:

$$SS_W = tr(W_k) \tag{33}$$

W_k is the within-cluster dispersion:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \tag{34}$$

C_q is the set of all data in class q, and c_q is the particle of class q.

CH measures the compactness (denominator) within a class by calculating the square sum of the distance between each point in the class and the center of the class, and the separation (numerator) of the dataset by calculating the square sum of the distance between the center points of all classes and the center point of the dataset. The larger the CH is, the better the clustering accuracy is. The CH value of our k-means model is 305.857, with good clustering accuracy.

In order to show the clustering results more intuitively, we further use the principal component analysis (PCA)[10][11] to reduce the dimensions of the original sample data. Using the first two principal components after dimensionality reduction to draw a scatter map, which can reflect the clustering effect to a certain extent:

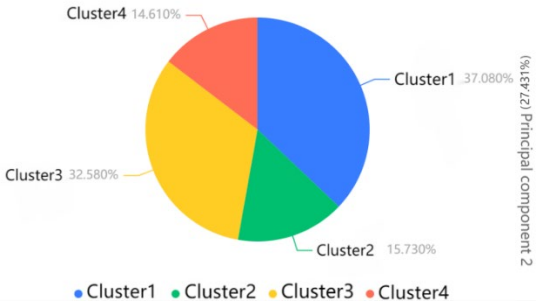

Figure 20: Percentage of clusters

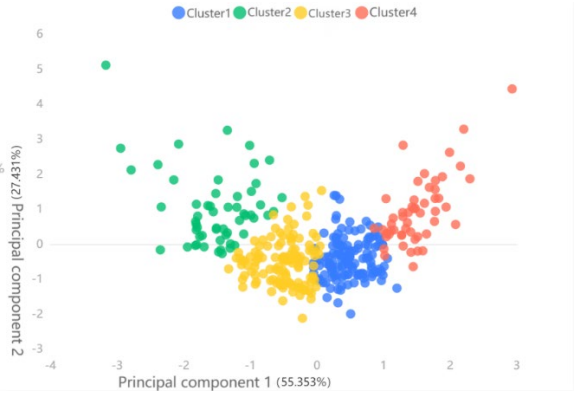

Figure 21: Cluster scatter

4.4.4 Sensitivity analysis of k-means model

Since the samples we use are constant, the sensitivity of the k-means model only depends on its randomness. We use the Jaccard similarity coefficient to measure the sensitivity of the k-means model. The calculation formula of Jaccard similarity coefficient is as follows:

$$J(A,B) = \sum_{i=1}^{K} \max_{j=1\ldots k} \frac{A_i \cap B_j}{A_i \cup B_j} \tag{35}$$

A. B represents different clustering results, and A_i represents the i-th subclass of division result A. J (A, B) can be used to measure the similarity of clustering results. Obviously, $0 \leq J(A,B) \leq 1$.

We first conducted 500 k-means cluster analysis experiments, then calculated the Jaccard similarity coefficients of every two experimental results in these 500 experiments, and finally calculated the average of all Jaccard similarity coefficients.

If the average value of the Jaccard similarity coefficient is close to 0, it means that the results of the 500 k-means clustering analysis experiments are quite different, indicating that the k-means model is highly sensitive and not stable enough.

If the average value of the Jaccard similarity coefficient is close to 1, it means that the similarity of the 500 k-means cluster analysis experiments is relatively high, indicating that our k-means model is relatively stable.

We calculated that the average value of the Jaccard similarity coefficient of the k-means model is 0.96, which shows that the randomness of the k-means algorithm has little influence on our clustering results, and our k-means clustering analysis model is very stable.

4.4.5 Cluster attribute analysis

In order to quantify the difficulty of the puzzle, we assume that the number of guesses that successfully solve the puzzle is 1, 2, 3, 4, 5, and 6, and the corresponding scores are 6, 5, 4, 3, 2, 1, and the scores for the rest is 0. Define the average score of each puzzle as:

$$score = 6 * P_1 + 5 * P_2 + 4 * p_3 + 3 * P_4 + 2 * P_5 + P_6 \quad (36)$$

$P_i(1 \leq i \leq 6)$ represents the percentage of the number of guesses i times. The higher the score, the lower the difficulty of the game.

We define 4 levels of difficulty according to the number of clusters, from low to high, they are 'simple', 'normal', 'hard', and 'master'. Since the difficulty of each category can be reflected by its cluster center, we can directly calculate the average score corresponding to the cluster center of each cluster, and then use the average score to judge the difficulty of each category. The difficulty judgment results are as follows:

Table 5: Difficulty analysis of categories

	The average score	Degree of difficulty
cluster class 1	2.986363636	normal
cluster class 2	2.171428571	master
cluster class 3	2.637327586	hard
cluster class 4	3.374230769	simple

Given that the number of syllables and usage frequency of a word have a certain degree of influence on the difficulty of the puzzle, we further analyzed the characteristics of the number of syllables and frequency of use in each cluster. We calculate the average number of syllables and the average usage frequency of all words in each cluster, and the results are as follows:

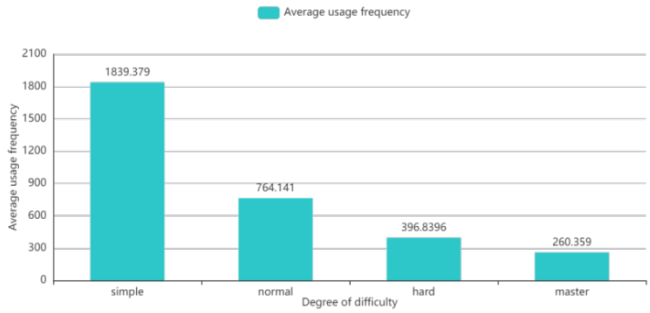

Figure 22: Average usage frequency

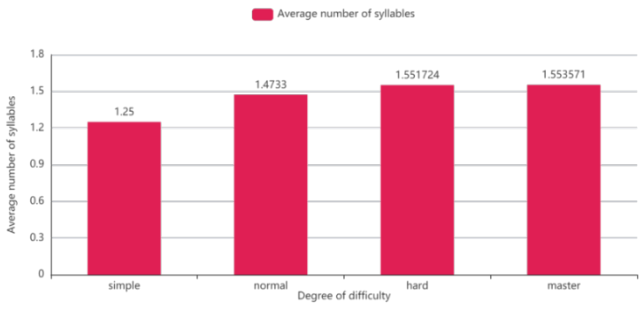

Figure 23: Average number of syllables

It is easy to see from the above two figures that as the difficulty of words in a cluster

increases, the average usage frequency of words in a cluster decreases, and the average number of syllables in a word increases. Explain that the more difficult the word puzzle, the less frequently the word is used, and the more syllables the word has. We believe that the smaller the word usage rate, the stranger the player is to the word, the more syllables in the word, the more possibilities the player has to consider when solving the puzzle, and the more guesses the player has to make when solving the puzzle. Our conclusions are in line with objective laws.

Combined with our answer to the first question, we can infer that there is a certain degree of positive correlation between the difficulty of the puzzle and the score percentage of the hard mode, that is, the more difficult the puzzle is, the less the total number of people trying to solve the puzzle, but the number of people playing the hard mode is relatively stable, which makes the score percentage of the hard mode higher.

Using the percentage predictions for (1,2,3,4,5,6,X) of 'EERIE' from the second question, calculate its distance from each cluster center. After calculation, it can be seen that 'EERIE' is the closest to the cluster center of the 'hard' difficulty category, so its difficulty is 'hard'. According to relevant data, the usage frequency of 'EERIE' is 42.57, and its number of syllables is 2. From these two attributes, the difficulty of 'EERIE' is indeed relatively high, which proves the rationality of our classification of 'EERIE'.

4.5 Other features of this data set

4.5.1 Frequency of each letter

As shown in Figure 24, the letter e gets the highest frequency, and the subsequent a, o, r, t, l, and i also exceed 100 times. J, z, q, x get the smallest number of times, and even less than 10. When guessing words, you may wish to consider words containing letters of higher frequency, and try to avoid guessing words that contain letters of lower frequency.

4.5.2 Case processing

We conduct case processing on the number of reported results and obtain the following results:

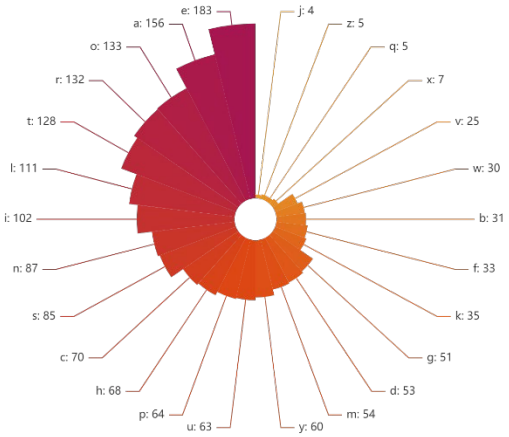

Figure 24: letter frequency

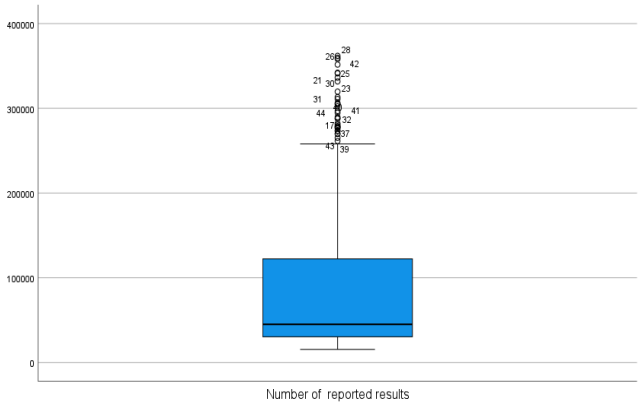

Figure 25: Case processing

It can be seen that most of the data are concentrated below 100000, and there are a large number of "outliers" beyond the upper bound. Moreover, from the overall shape of the figure, our data is not centralized enough and relatively scattered.

5 Model Evaluation and Improvement

5.1 Model Strengths

1. The time series analysis model based on nonlinear regression established in the first question has strong mathematical theory, so the fitting result is very good, the error is very small, and the prediction for the future is also accurate enough.

2. The proposed framework SOUL can calculate the distribution with high accuracy and confidence. Specially, the SOUL effectively quantify the confidence of its predictions, and prove itself to be reliable.

3. With all modules mentioned above, we have formed a coherent work flow, which can output multi-indicators like distribution, confidence, difficulty and other attributes, with just single word accepted.

5.2 Model Weaknesses

1. In the process of modeling, for example, the judgment of the end of k-means iteration and the selection of k-value can also be further optimized in combination with the topic information.

2. Limited to rarity and noisy data, the model can't be generalized to a larger input space. The model can be improved by subsequent Online Learning.

5.3 Model improvement

1. In the first question, if we use model of strong biochemical theory may be able to fit well on the data affected by the early game popularity.

2. The third problem is to use the k-means algorithm to realize the difficulty classification of the answer words. We can consider using more classification models to solve it, and compare the effect of different classification models to choose the best result.

6 Conclusion

In our work, we reasonably predict the number of reported results in the future. We found that word's usage frequency and number of syllables affected the percentage of scores reported that were played in Hard Mode, that is, the more frequently the word was used, the fewer syllables the word had, the smaller the corresponding percentage of scores reported that were played in Hard Mode. Subsequently, we built models SOUL to predict the distribution of try times for a word, accompanied. We then used the k-means model to categorize the answers by difficulty, and further analyzed the attributes of the words in each category. We found that the more difficult the solution word is, the less frequently it is used and the more syllables the word has.

7 Letter to the Puzzle Editor

Dear Puzzle Editor of the New York Times:

We analyzed the relevant data of the Wordle you designed and drew some interesting conclusions. We are happy to share these conclusions with you.

We analyzed the changes of Number of reported results and Number in hard mode over time, and found that as time went by, Number of reported results first rose rapidly, then decreased, and finally stabilized, while Number in hard mode was relatively stable, changing very small. This may be due to the fact that in the early days of Wordle's release, a large number of people were driven by curiosity to solve the puzzles. But as time goes by, Wordle cannot bring more freshness to people, most people will not play Wordle again, only those who like Wordle will continue to play. Most of the people who play hard mode should be people who like Wordle, this kind of players will insist on playing Wordle, so Number in hard mode will be more stable. In addition, we found that the usage frequency of word and the number of syllables have an effect on the percentage of hard mode scores, that is, the less frequently used a word is, the more syllables the word has, and the higher the percentage of hard mode score.

We designed the SOUL model, which is a neural network model aggregated in multiple dimensions, which can output the distribution of try times according to the word. We quantified the uncertainty of the SOUL model, and we can intuitively observe that SOUL has a high degree of confidence, which can ensure that the predicted distribution results are reliable. The distribution of try times can intuitively reflect the difficulty of the puzzle. If you want, we can tell you the distribution of try times for the designed words, so you can adjust the difficulty of the puzzle.

We also classify the solution words according to their difficulty, and further study the relationship between the attributes of words and the difficulty of solution words. We found that the more frequently a word is used, the lower the number of syllables in the word, the lower the difficulty of the word, and the lower the Hard Mode Score percentage. Considering that the number of people playing Hard Mode is relatively stable, a lower Hard Mode score percentage means a higher total player count. Therefore, you can appropriately reduce the difficulty of the puzzle to attract more players.

We hope our conclusions will be helpful to your future puzzle designs.

Sincerely,

Team # 2303707

References

[1] Qing Pan. Study on prediction model of industrial pollutant gas concentration based on deep learning and time series[D].Zhejiang Gongshang University,2022.DOI:10.27462/d.cnki.ghzhc.2022.000562.

[2] Yating Gao. Research on Time Series Trend Prediction Method and Its Application in Stock Market[D].Shandong Business College,2022.DOI:10.27903/d.cnki.gsdsg.2022.000041.

[3] Gogtay N J, Thatte U M. Principles of correlation analysis[J]. Journal of the Association of Physicians of India, 2017, 65(3): 78-81.

[4] Li J, Cheng J, Shi J, et al. Brief introduction of back propagation (BP) neural network algorithm and its improvement[C]//Advances in Computer Science and Information Engineering: Volume 2. Springer Berlin Heidelberg, 2012: 553-558.

[5] Seoh R. Qualitative analysis of Monte Carlo dropout[J]. arXiv preprint arXiv:2007.01720, 2020.

[6] Ganaie M A, Hu M, Malik A K, et al. Ensemble deep learning: A review[J]. Engineering Applications of Artificial Intelligence, 2022, 115: 105151.

[7] Pham D T, Dimov S S, Nguyen C D. Selection of K in K-means clustering[J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2005, 219(1): 103-119.

[8] Werchao Zhang. Overview of the research on correlation coefficient[J]. Journal of Guangdong University of Technology,2012,29(3):12-17.

[9] Saroj,Kavita.Review:study on simple k mean and modified K mean clustering technique[J].International Journal of Computer Science Engineering and Technology,2016,6(7)：279-281.

[10] Wang Wei, Zhao Ming. Application of principal component analysis in the construction of classification index system of aviation materials [J]. Ship Electronics Engineering, 2019, 39 (01): 118-120+151.

[11] He Xiaoqun. Multivariate statistical analysis. Beijing: China Renmin University Press, 2012.