| Problem Chosen | 2023 MCM/ICM Summary Sheet | Team Control Number |
|---|---|---|
| C | | 2316712 |

# A Comprehensive Study on Wordle game and Wordle game word


## Summary

In the Wordle game, the computer randomly chooses a 5-letter word and the player has six chances to guess the word. After each guess, the computer provides feedback indicating how many letters are correct and whether they are in the right position. A critical question throughout the problem is which kind of word is difficult to guess. What makes a certain word difficult to guess?

We can solve the problem by a series of outstanding and explainable features of a word. In addition to some regular attributes of a word like frequency and repeated letter times, **we proposed a new attribute called "Wordle Word Perplexity"**. It's a static and determined metric for a word. This attribute is computed to measure a word's similarity with other words. But it also considers the different importance of different letters and the frequency of the confused word. The weight is found by **simulated annealing**. In our **ablation experiments**, we find that this "Wordle Word Perplexity" can **increase 13% percent** of the effect.

For the first question. For the prediction of the number of reports submitted on twitter for problem 1, we captured the temporal characteristics of the data for the time series, and captured the general laws of game propagation to construct a dynamics model and named it ***PNP*** model, respectively. We gave the prediction results and evaluated the model effects based on the different models. For the second part, we use **Pearson similarity** to measure the linear correlation, **Spearman's rank correlation coefficient** to measure the monotonic correlation, and draw the image to observe other correlations. Finally, we conclude that there doesn't exist any correlation.

For the second question. We train a **Multilayer Perceptron(MLP)** with the attributes from the feature engineering and compare its effects with **other machine learning models including DecisionTree and RandomForest**. For training fairness and effectiveness, we use **K-fold cross-validation** to examine our results and do many detailed experiments. There is a difference of **2.23**%(which is percentage) from the actual data and predicted on validation fold data-set on average.

For the third question regarding the measurement of difficulty, We used the **Kolmogorov-Smirnov test** to check the frequency distribution of the trial attempts in the data and accepted that the distribution conforms to a normal distribution. We then used the mean, skewness, and kurtosis of the normal distribution as clustering indicators for the **k-means algorithm** and used the **silhouette coefficient** to select the best number of clusters as four, which were ranked in increasing difficulty. We used the features above to train another MLP classifier, which achieved good classification results. We believe that the EERIE word has a difficulty level of 3, and we have provided a reasonable explanation. Additionally, we considered the impact of the difficult mode on the data skewness and used the **naive Bayes approach** to adjust the skewness of the normal distribution, making the model more reasonable.

For the fourth question, we find five interesting items about the date, players' abilities, and others.

# Contents

# 1 Introduction

## 1.1 Restatement of the Problem

- For the first question,the first requirement is making a model for prediction of the number of reported results. The second requirement is to do feature engineering for the word and explore the relationship between the attribute and
- For the second question, train a model to predict the seven percentage of certain word. The attributes of words in the second question is the same as the first question.
- For the third question, train a model to classify the words by difficulty. The attributes of words in the third question is the same as the second question.

## 1.2 Notations

The primary notations used in this paper are listed in Table 1.

Table 1: Notations

| Symbol | Definition |
|---|---|
| $\mu$ | the average tries time of all words in giving excel |
| $\beta$ | the rate coefficient that non-players become players |
| $k$ | the relative impact of non-posting players compared with posting players |
| $p$ | the percentage of players who don't post Twitter |
| $\omega$ | the rate that non-players become players who post |
| $\omega'$ | the rate that non-players become players who don't post |
| $\gamma$ | the rate that posting players quit playing Wordle |
| $\gamma'$ | the rate that non-posting players who don't post |
| $q$ | the percentage of keen players who always post results |

# 2 Data exploration

## 2.1 Data cleaning

There are some wrong data in the excel. The first category is the word spelling mistakes. Like the 15-th line, the word is "probe". Other spelling mistakes including words "clean", "trash", "favor","and marsh" have all been modified. The second category is the numeric mistake. Like the 31-th line, the number of reported results definitely is false. It can be changed to a number that is larger than 20000.

## 2.2 Data Preprocessing

1. Some data needs to change to the proportion which is larger than zero and smaller than one. The number in hard mode should be divided by the current number of reported results. The percentage of the number of tries should be divided by 100.

2. When it comes to the input of the following model, the data of the network model should be on the same scale. So the min-max scale processing is processed on the columns of "Number in hard mode", "Number of reported results" and "Contest number".

# 3 Preparation of the Models

## 3.1 Feature engineering of words

For this Wordle game analysis, the extraction of words is critical. That is, which word is easier to be guessed. We summery some important attributes of the word which maybe influence the times of the Wordle game.

### 3.1.1 The CEFR level of the certain word

In fact, the people's level of familiarity with the word is the best metric. But it's hard to know the level. The CEFR level of a word is similar to the frequency of the word. It contains six levels: A1, A2, B1, B2, C1, and C2, with A1 being the most commonly used in life and C2 being the least commonly used. The level should be modified into a float number so that the model can take it as input. In the experiment, the level is mirrored to a number from 0 to 1. The A1 level is mapped to 0.125. The C2 level is mapped to 0.875. The word that is not in the CEFR list will be recognized as unfamiliar words whose value is 1.0.

### 3.1.2 The appearance times of the one certain letter matters

If a word contains letters that occur frequently (e.g. e, a, o, etc.), then these letters will be easy to guess and the game will be less difficult. Conversely, if the word contains letters that are rarely found, such as q, z, x, etc., then these letters become less likely to be guessed and the game becomes more difficult. For example, "enjoy" and "judge" are familiar, but they take more time to guess. It's even much more difficult that there is the same letter in one certain word like "mommy", or "excel". So the appearance times of one letter is deserved to be considered.

### 3.1.3 The Wordle Perplexity

This is defined by the inspiration that "epoxy" and "proxy" is difficult to guess because their average attempt times are more than others. These two words look similar. It may be the situation in which someone knows "oxy" appears and their position. But it's difficult to know the final answer directly. So we propose the Wordle Perplexity that measures the degree the word looks like other

words. The following is the perplexity design pattern.

$$Perplexity_n\ of\ A = \sum_{w \in W} ((2 - CEFR_w) * \prod weight(letter_{replaced}) * weight(letter_{replacement}))$$

The n refers to the level of perplexity. In the real guessing process, there are three important information categories.

- Known letters with the corresponding position.
- Known letters without knowing the right position
- Known letters not in the word

The W set contains the words in Wordle answer lists which differ n $(n \leq 3)$ letter from the original word A. The weight of a specific letter is determined by the degree of certainty. The letter with higher frequency will have higher certainty because these letters are either composed of the word or not always known by the player. For example, the letters "a", "e" and "s" will have a very high degree of certainty. Most players will guess the first word with these three letters. So they will have a lower weight because these words similar to origin word won't be such confused. The specific number is in Appendix.

It can be proved that the perplexity of 2 degree has a relatively strong linear positive correlation with $\mu$ which is global average guessing times. It can be measured by Pearson similarity.

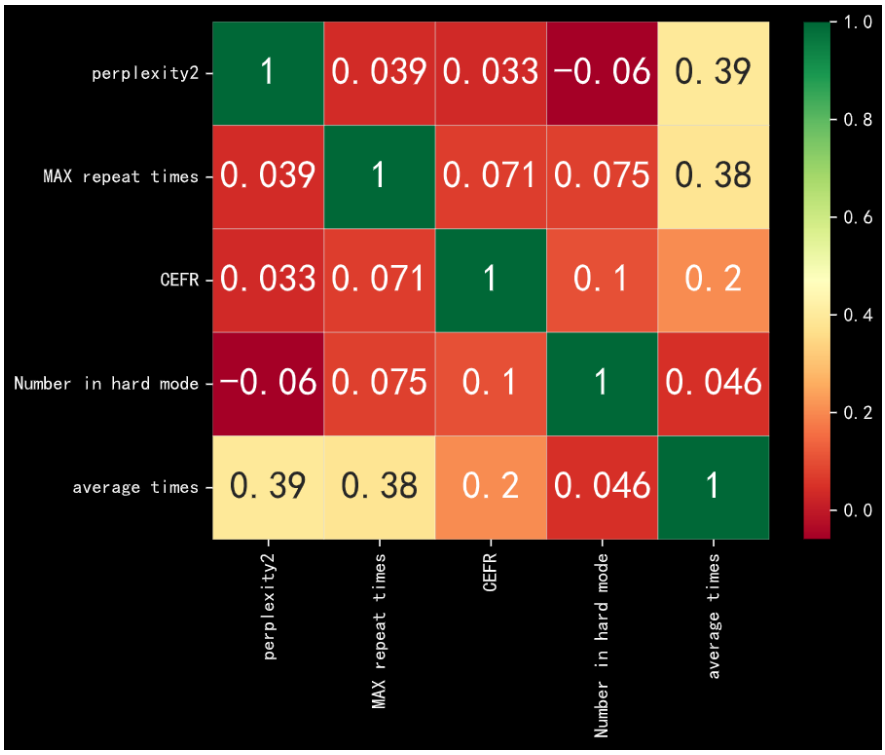



Figure 1: The result of Pearson similarity among several important columns

This reveals that perplexity completely computed by the above algorithm can dig a positive attribute with a correlation bigger than 0.39. The correlation coefficients of perplexity among other

attributes include "MAX repeat times", "CEFR", and "Number(percent) in hard mode" which are all smaller than 0.05.

The perplexity of 3 degrees and perplexity of 1 degree have a relatively lighter linear positive correlation with $\mu$ whose correlation coefficients are around 0.2.

# 4 Analysis and Modelling

## 4.1 Question 1

### 4.1.1 Prediction for number of reported results on March 1, 2023

For Question1 we need to predict the posting players' number of Wordle on March 1, 2023 based on the number of players who have posted their individual game results on twitter in 2022. Wordle was released in October 2021. According to the data we got for 2022, we can see from the graph below that the number of posting players reached its highest on February 2, 2022, at 361,908. And since mid-February, the number of reported results also gradually dropped back, tending to a steady downward trend in the following time.

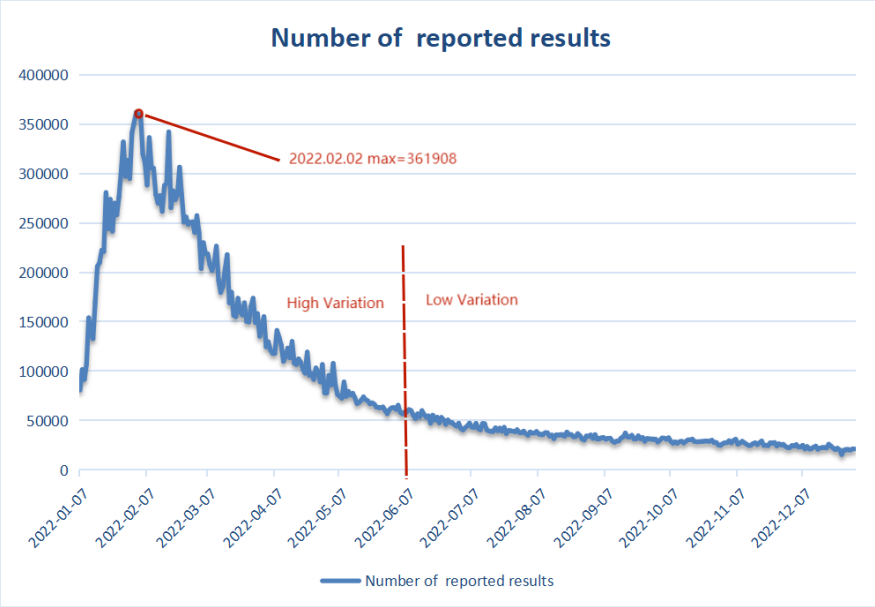



Figure 2: Visualization of data

For the given data, we first note that time, as the independent variable, can be predicted using real series models. Here, we have chosen three models: ARIMA WITH DRIFT, PROPHET and PROPHET with XGBOOST. Later we will describe in detail the training, diagnostic and forecasting process of the three models.

However, the prediction using the time series model can only use data up to after the peak retreat. Because it is predictable that after the peak of the game's online traffic, the number of people submitting daily results on twitter tends to decline due to the defined rules of the single-player game and the absence of special in-game marketing activities. Therefore, from a communication

perspective, we should build a dynamics model to simulate the change in player numbers. Here, we refer to the SEAIR model used to describe infectious diseases and independently create a model called ***PNP***. Figure.2 is the prediction result of the time series model and the ***PNP*** model for predicting the number of players who will submit reports on March 1, 2023, followed by the details of the model.

**Time Series Model: ARIMA WITH DRIFT PROPHET and PROPHET with XGBOOST** Firstly, to avoid misleading model fitting by the peak data of the first 5 months, we only keep the data of the last 184 days as the prediction basis of the time series model. We train the model on a month-by-month basis and use cross-validation to improve the prediction accuracy. The following figure visualizes the training set.

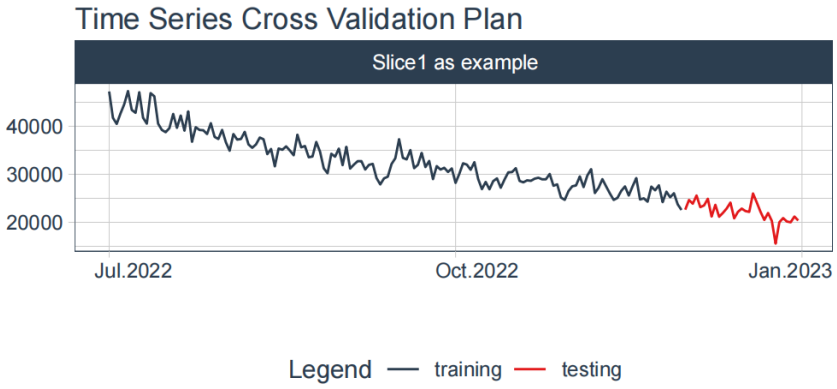



Figure 4: Answer for Q1 predicted by 4 models

Automatic models are usually modeling methods that have been automated. This includes the "Auto ARIMA" functions from the forecast package of R and the "Prophet" algorithm from the prophet package. "algorithm from the prophet package. These algorithms are integrated into the modeltime package. The setup process is simple.

Model selection: Use canonical functions to initialize the algorithm and key parameters; Engine**:** Train the model using the engine available for model selection; Fitting the model: Fitting the model to the training data

The ARIMA model is a classic time series forecasting model that can capture the trends, seasonality, periodicity, and randomness in the data. It is known for its simplicity, interpret-ability, and forecasting accuracy. By fitting the historical data, it can predict future values with reasonable accuracy. On the other hand, the Prophet model is a more flexible and nonlinear model that can

| Model | Date | Value | conf_low | conf_high |
|---|---|---|---|---|
| ARIMA(0,1,2)(2,0,0)[7] WITH DRIFT | 2023/3/1 | 12158.59 | 8884.396 | 15432.8 |
| PROPHET | 2023/3/1 | 9989.23 | -4490.074 | 24468.5 |
| PROPHET with XGBOOST | 2023/3/1 | 8880.24 | -5837.819 | 23598.3 |
| PNP | 2023/3/2 | 10305.73 | NA | NA |

Figure 3: Visualization of cross validation

capture complex patterns in the data, including seasonality, trend changes, and holiday effects. It is highly interpretable and comes with built-in visualization tools to showcase the model's fitting and prediction performance. It can also incorporate custom holidays and other events to improve the forecasting accuracy.

In addition to ARIMA and Prophet, we have also used a hybrid time series model that combines Prophet and XGBoost. This hybrid model aims to take advantage of the strengths of both models and improve the forecasting accuracy by incorporating additional features.

The Prophet model captures the seasonality, trend changes, and holiday effects, while the XGBoost model focuses on the non-linear relationships between the target variable and the additional features. By combining the outputs of both models, the hybrid model can better capture the complex patterns in the data and produce more accurate predictions.The hybrid model also allows for feature engineering and selection, which can further improve the model performance. By selecting the most relevant features and optimizing the hyperparameters of XGBoost, the hybrid model can achieve even better forecasting accuracy. Overall, the hybrid time series model that combines Prophet and XGBoost provides a powerful and flexible approach to time series forecasting, and can be particularly effective for datasets with complex patterns and multiple variables.

Here is the forecasting plot of three time series model.

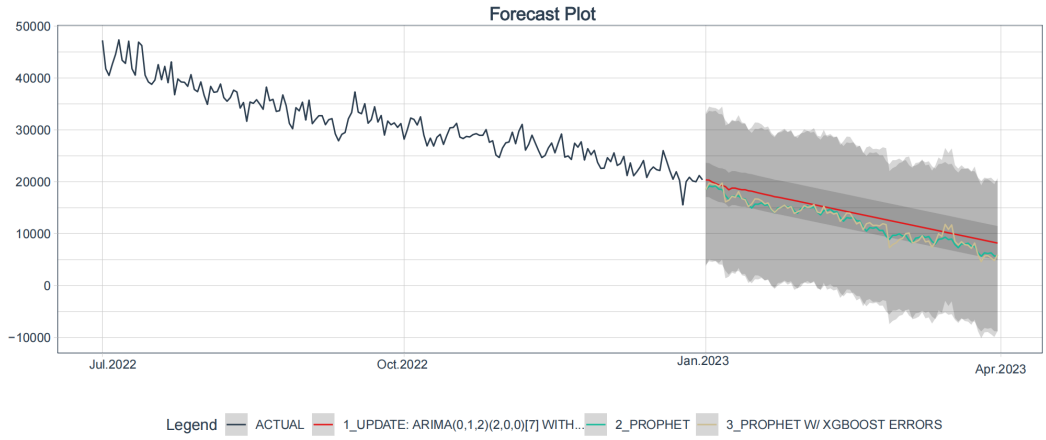



Figure 5: Forecasting plot

**Self-built model: *PNP*** We notice that the data given in the question is not the number of players who actually participated in the game, but the number of players who participated in the game and sent the results to twitter. Therefore, we should adapt the model to the actual situation and build it from the perspective of game data analysis. This way, we can make more use of the data and known information to build a more realistic model compared to the traditional time series model. Of course, the prediction effect of the specific model will be tested by time.

First, we restore the overall situation of the participating games. There are six important groups. The relationship between them are as follows and corresponding parameters are listed in ***Section1.2.***

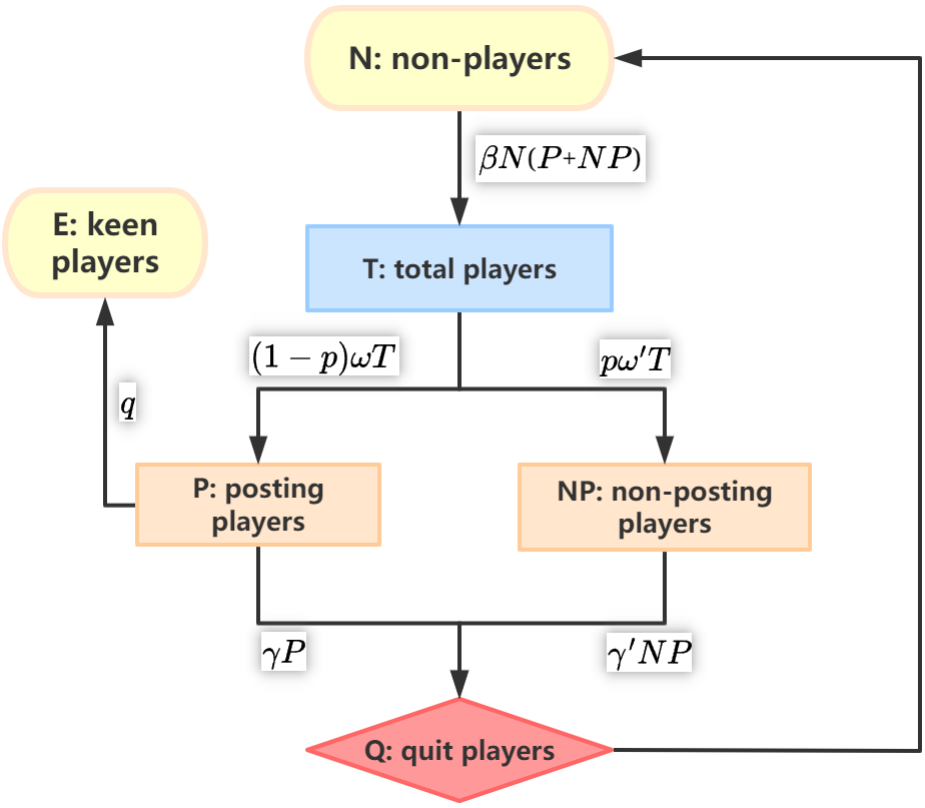



Figure 6: *PNP* Model for players

**N**: non-players. They are players who do not participate in the game and are potential users. This value implies an upper bound, because the number of potential players of the game is limited. **T**: total players. This is the total number of people involved in the game. **P**: posting players. This is the number of players who participate in the game and also report the results on twitter. This is our important data and the variable we need to predict. **NP**: Unlike posting players, this is the number of players who participated in the game but did not share their results on twitter. **Q**: This is the number of players who quit after playing the game for a while. **E**: keen players. This is the part of the game that is loyal and will keep reporting on twitter. The reason for including this group in the model is that we have observed a gradual plateau in the number of hard mode selections, which indicates the existence of such a group of players who have always stuck with the game.

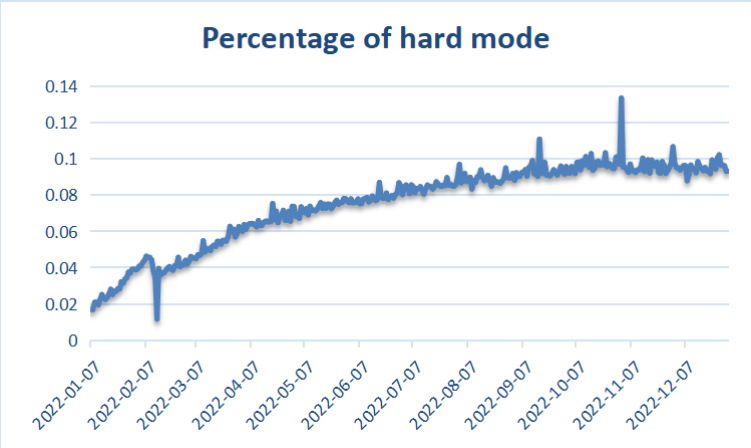



Figure 7: How percentage of hard mode change

It is important to note that our predictions for the data are all calculated in days. According to the relationship between these groups, we establish the following ordinary differential equation:

$$\frac{dN}{dt} = -\beta N\left(P + kNP\right)$$

$$\frac{dT}{dt} = \beta N\left(P + kNP\right) - \left(1-p\right)\omega T - p\omega' T$$

$$\frac{dP}{dt} = \left(1-p\right)\omega T - \gamma P - qP$$

$$\frac{dNP}{dt} = p\omega' T - \gamma' NP$$

$$\frac{dQ}{dt} = \gamma P + \gamma' NP$$

$$\frac{dE}{dt} = qP$$

Based on these six differential equations, we construct a dynamical system, which can incorporate the transformation between different populations into the model and enhance the prediction. We used package(desolve) in R language to perform the simulation, and used annealing algorithm to iterate the parameters to finally obtain the model as shown in Fig8.

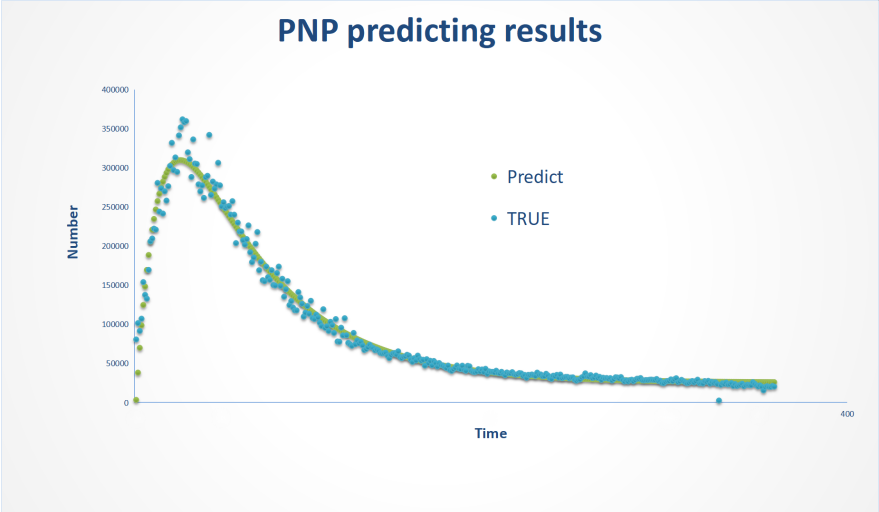



Figure 8: *PNP* prediction results

### 4.1.2 Which attribute of word influences the percentage of hard mode

For the relation of the percentage of hard mode and other attributes, we analyse it from linear relation and nonlinear relation.

1. For the linear relationship analysis, we use the **Peason similarity**.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

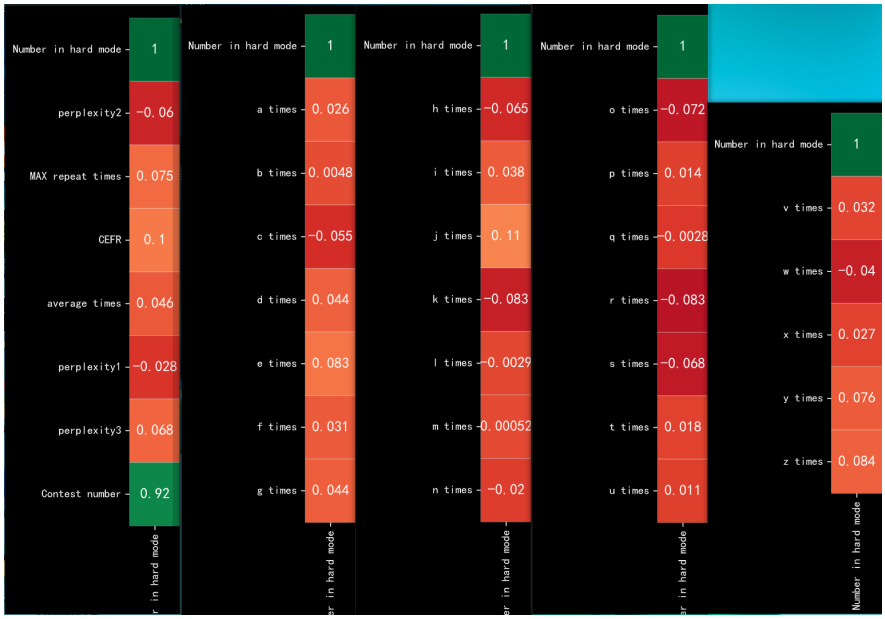



Figure 9: The heatmap of correlation(The hard mode data has been converted to percentage forgetting change the column name))

The above picture reveals all of the words attributes Person similarity is lower than 0.11, and most of them are around 0. The contest number is a strong positive correlation with the percentage

in hard mode. So it concludes that there doesn't exist linear correlation. 2. For monotonic analysis, we use the Spearman correlation coefficient.

$$\rho_S = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

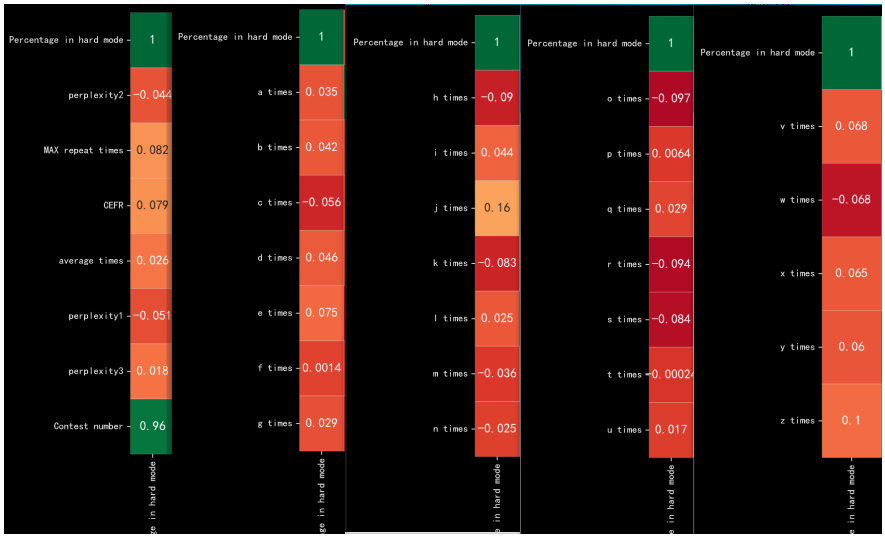



Figure 10: The heatmap of spearman correlation

This also demonstrates that there doesn't exist any monotonic correlation between the percentage of hard mode and another word.

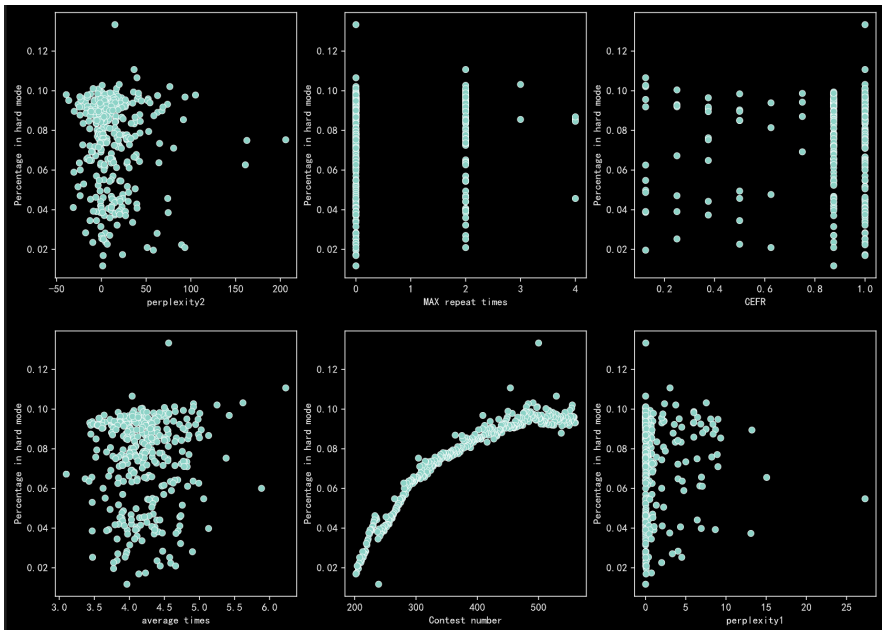



Figure 11: The scatter plot of percentage of hard mode with other words

We still can't find some non-monotonic correlation between the percentage of hard mode and another word.

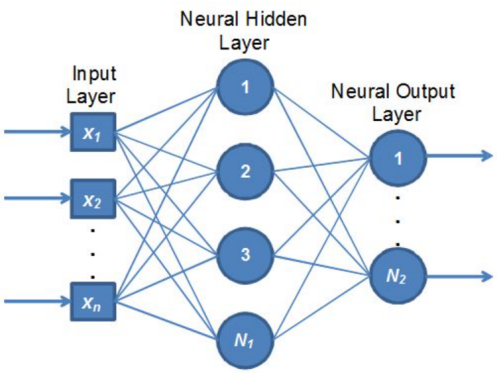
**So we conclude that there are not any attributes of the word affect the percentage of scores reported that were played in Hard Mode**

## 4.2 Question2:Machine learning models for the associated percentages of (1, 2, 3, 4, 5, 6, X)

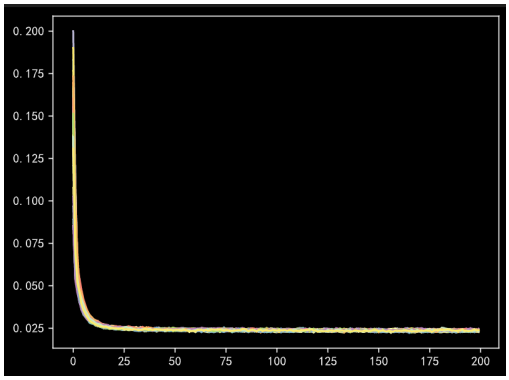
After we establish a series of features for a certain word, it can combine the features with other information like time and hard mode percentage to one vector. These vectors can be taken as the input. MLP, short for Multilayer Perceptron, is a type of feedforward neural network that is commonly used to solve classification and regression problems. It consists of an input layer, at least one or more hidden layers, and an output layer, each layer being composed of multiple neurons with weights that are fully connected between adjacent layers. To evaluate the effect of the model, we use K-fold cross-validation to examine each machine-learning model. In our experiment, we take k = 8. We divide the original dataset into 8 non-overlapping subsets. For each subset, use it as the validation set, use the other K-1 subsets as the training set, train the model, and evaluate the performance on the validation set to obtain a model performance metric. Calculate the average of the 8 performance metrics as the model's average performance metric. In order to offset the randomness of grouping, we will take the 8-fold cross-validation for 5 times. We use the Adam gradient optimizer.

### 4.2.1 Implement detalis

For the implementation of different machine learning models. We construct our MLP with PyTorch. The number of layers for this data scale takes 2 or 3. The loss function for PyTorch we choose M1loss, which mean $L_1 = \sum_{i=1}^{n} |y_i - \hat{y}_i|$ We construct other models including DecisionTreeRegressor, RandomForestRegressor, XGBRegressor, and LGBMRegressor by sklearn package. These four models are set to compare with the MLP model.




(a) MLP schematic diagram




(b) LOSS decreasing diagram

### 4.2.2 experiment

There are several advanced sets up for these experiments. To reach the fairness of these experiments, the below models only train with no more than 5 minutes. The unit of the value of all the columns is a percentage.

**All the performance represents the validation set which won't be seen by the training model in 8-fold cross-validation in my experiments. For real prediction, I will use all 359 rows to train the model and train with more time.**

Table 2: Different models with their performance

| model | average percent MAE loss | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | $\leq$ 7tries |
|---|---|---|---|---|---|---|---|---|
| MLP(2 layers, no relu, no softmax, 64 hidden) | 2.25 | 0.17 | 2.00 | 3.74 | 3.11 | 2.89 | 2.93 | 1.24 |
| MLP(2 layers, no relu,softmax, 64 hidden,) | 2.25 | 0.09 | 1.65 | 3.73 | 3.08 | 2.75 | 3.16 | 1.26 |
| MLP(2 layers,leakyrelu(0.8),no softmax, 64 hidden) | 2.24 | 0.12 | 1.72 | 3.7 | 3.11 | 2.86 | 2.95 | 1.21 |
| MLP(2 layers,leakyrelu(0.8),softmax, 64 hidden) | 2.23 | 0.09 | 1.66 | 3.7 | 3.04 | 2.76 | 3.08 | 1.26 |
| MLP(**3 layers**,leakyrelu(0.8),softmax, 64 hidden) | **2.20** | 0.09 | 1.64 | 3.58 | 3.0 | 2.74 | 3.08 | 1.29 |
| MLP(3 layers,**relu**,softmax, 64 hidden) | 2.46 | 0.08 | 1.68 | 4.1 | 3.36 | 3.12 | 3.48 | 1.38 |
| DecisionTree Regressor | 4.14 | 0.40 | 3.47 | 6.93 | 4.95 | 5.58 | 5.34 | 2.35 |
| RandomForest Regressor | 2.70 | 0.09 | 2.16 | 4.95 | 3.14 | 3.69 | 3.59 | 1.31 |
| LGBMRegressor | 2.88 | 0.1 | 2.76 | 5.19 | 3.15 | 3.99 | 3.55 | 1.44 |
| XGBRegressor | 2.56 | 0.12 | 2.02 | 4.51 | 3.13 | 3.48 | 3.3 | 1.38 |

From above table, it denotes that the best model is just 2 layers fully connected or its extended models. Given an input vector $z = [z_1, z_2, \cdots, z_k]$, the Softmax function converts it into a probability distribution $\hat{y} = [\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_k]$,where each element $\hat{y}_i$ represents the probability that the sample belongs to the $i$-th class. Leaky ReLU is a variant of the rectified linear unit (ReLU) that has a small slope $\alpha$ for $z < 0$. When using relu activation function, it will become unstable in this small scale of data.

$$\text{LeakyReLU}(z) = \begin{cases} z, & \text{if } z > 0 \\ \alpha z, & \text{otherwise} \end{cases}$$

From the experiments, the Softmax layer and nonlinear activation layer won't decrease loss significantly. The scale of the data is too small for the neural network. But the deep learning method is

still stronger than the traditional machine learning method in this problem. **In prediction, considering effect and speed, we use MLP(2 layers,leakyrelu(0.8),softmax, 64 hidden) for the word eerie.**

### 4.2.3 Ablation study

In order to prove the CEFR, MAX repeated time and Perplexity is effective for the prediction, we drop each of them to make a further study.

Table 3: Different models with their performance)

| model | average percent MAE loss | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | $\leq$ 7tries |
|---|---|---|---|---|---|---|---|---|
| MLP(2 layers,leakyrelu(0.8),softmax, 64 hidden) | 2.23 | 0.09 | 1.66 | 3.7 | 3.04 | 2.76 | 3.08 | 1.26 |
| MLP(no perplexity) | 2.57 | 0.17 | 2.00 | 4.04 | 3.51 | 3.03 | 3.60 | 1.61 |
| MLP(no MAX repeated times) | 2.49 | 0.13 | 1.99 | 4.41 | 3.10 | 3.26 | 2.31 | 1.24 |
| MLP(no CEFR level) | 2.43 | 0.08 | 1.68 | 4.05 | 3.38 | 2.89 | 3.56 | 1.39 |

The results of the experiment support that perplexity can decrease 13% of the loss. MAX repeated times can lower 10% of the loss. CEFR level can reduce 8% of the loss. Our Wordle perplexity attribute is useful.

#### 4.2.4 EERIE prediction

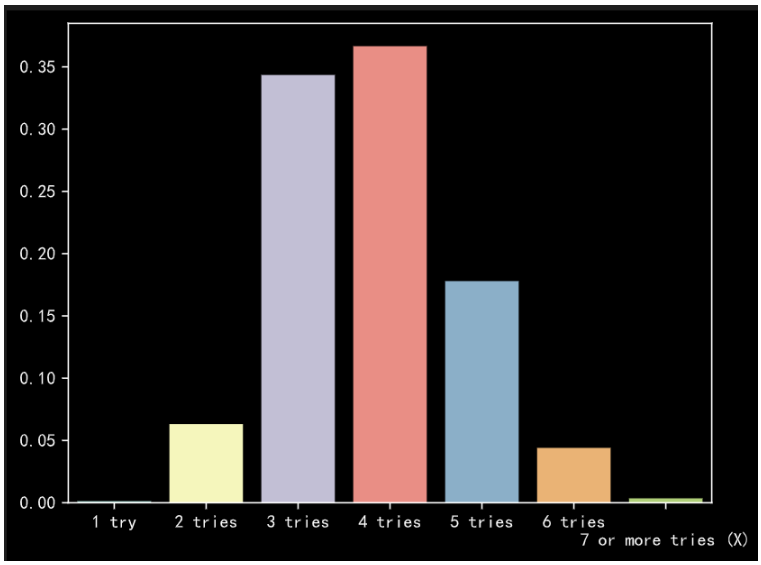



Figure 13: The result of EERIE prdiction

Table 4: The predicted answer

| word | 1 try | 2 tries | 3 tries | 4 tries | 5 tries | 6 tries | $\leq$ 7tries |
|---|---|---|---|---|---|---|---|
| eerie | 0 | 6 | 34 | 37 | 18 | 4 | 1 |

#### 4.2.5 the uncertainties and the confidence of model and prediction

**1.** The first uncertainty is from data. The data is provided by the percentage which is an integer. In fact, the integer is not numeric. And the Wordle game is full of uncertainty and luck plays an important role.

**2.** Another critical uncertainty is the initial weight of MLP. The initial parameters of the MLP model are set random. Especially for this small-scale data, it's easy to fall into local minima. What can we do is to utilize K-fold cross-validation and more experiments to offset the uncertainties.

#### 4.2.6 Confidence of model

It's hard for a deep learning model to measure confidence. But we can use the result on the validation dataset to determine the confidence. For each test, there are 78 percentage that $\left|y_{predict} - y_{real}\right| \leq 0.2 * y_{real}$. In some sense, we can conclude that the MLP model will predict a not bad value(not deviate more than 20%) in 0.78 possibilities.

## 4.3 Question 3

### 4.3.1 Develop and summarize a model to classify solution words by difficulty.

We utilized a clustering algorithm based on normal distribution to categorize the difficulty levels of the Wordle game using measures of mean, skewness, and kurtosis. This approach allowed us to effectively group the game's challenges based on their unique statistical properties, enabling us to better understand the nuances of each level's complexity.

**1.** Verify that the distribution is normal.
Since player levels tend to show a normal distribution, let's assume that the frequency distribution of correct guesses times for each word also corresponds to a normal distribution. We will use the Shapiro-Wilk test to verify whether it is normally distributed.

Shapiro-Wilk test is a statistical test used to assess whether a sample of data comes from a normal distribution. It is based on the calculation of the W statistic, which is a measure of the discrepancy between the sample data and the expected values under the null hypothesis of normality. The null hypothesis of the Shapiro-Wilk test is that the sample data comes from a normally distributed population.

The formula for the Shapiro-Wilk test statistic is:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{1}$$

where $x_{(i)}$ is the $i$th order statistic (i.e., the $i$th smallest value) of the sample, $\bar{x}$ is the sample mean, $n$ is the sample size, and $a_i$ are constants that depend on the sample size and are used to maximize the test's power to detect deviations from normality.

The Shapiro-Wilk test is commonly used in hypothesis testing and in exploratory data analysis to check for normality. If the p-value of the Shapiro-Wilk test is less than the chosen significance level, the null hypothesis is rejected, indicating that the sample does not come from a normal distribution. If the p-value is greater than the significance level, there is not enough evidence to reject the null hypothesis.

After both tests, we do not reject that the distribution is in a normal distribution, so it is reasonable to accept that all distributions are normal.

**2.** Modeling with clustering algorithms.
We can derive from practical experience that if the number of attempts by players to complete a game decreases with the increasing number of players, then the game is simpler, and the corresponding normal distribution will be left-skewed. On the other hand, if more players attempt the game more times, then the game is more difficult, and the corresponding normal distribution will be right-skewed. We do not consider factors such as the ratio of hard mode and time in this discussion. Therefore, we use the mean $\mu$ and skewness of the normal distribution to characterize the difficulty of a specific word. Moreover, kurtosis affects the height of the distribution at the mean, meaning that the area under the curve is larger. A higher value of , skewness, and kurtosis indicates a more difficult word.

Mean: $\mu$ Skewness: $\gamma_1$ Kurtosis: $\gamma_2$ The formula for the normal distribution is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

where $\sigma$ is the standard deviation. The skewness and kurtosis of the distribution can be calculated as:

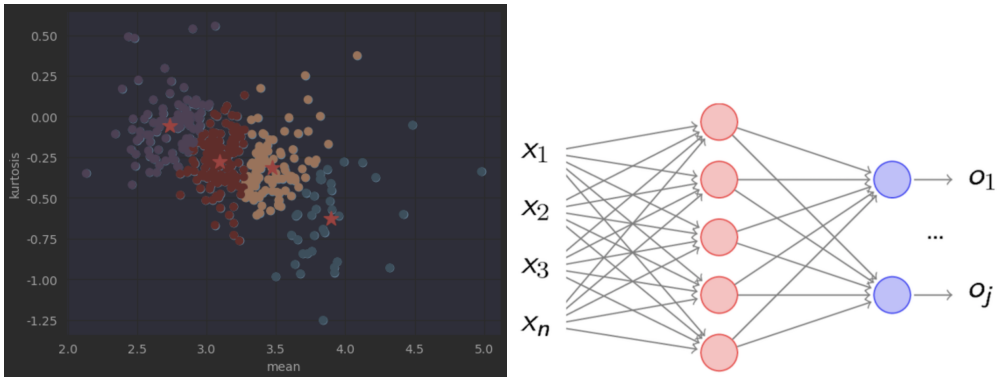
$$\gamma_1 = \frac{E[(x-\mu)^3]}{\sigma^3}$$
$$\gamma_2 = \frac{E[(x-\mu)^4]}{\sigma^4} - 3$$

where $E$ is the expected value operator. A higher value of $\mu$, $\gamma_1$, and $\gamma_2$ indicates a more difficult word.

We perform K-means clustering using the mean, skewness, and kurtosis of a normal distribution, and obtain four clusters representing different levels of difficulty.

K-means clustering is a commonly used unsupervised machine learning algorithm that aims to partition a dataset into a predetermined number of clusters, where each data point belongs to the cluster with the closest mean.

For ease of visualization and interpretation, we use only the mean and kurtosis for clustering, and sort the resulting cluster centers by the first dimension (mean). We use the same clustering results for 3D.




(a) Cluster result (b) MLP classifier

Figure 14: Two images

**3.** Difficulty Level Classifier.
We use the MLP classifier, with the same features as the predicted distribution in the second question, to train the classification of word difficulty levels. And using k-fold cross-validation is to get the error rate of the final classification. As shown in the figure.

**4.** Discuss the accuracy of your classification model. This table displays the accuracy scores for each of the folds in a k-fold cross-validation procedure. In particular, the table shows the fold number and the corresponding accuracy score for that fold. The k-fold cross-validation procedure involves dividing the data into k equally sized subsets, or folds, and then training the model on k-1 of these folds while using the remaining fold for testing. This process is repeated

Table 5: Accuracy Scores for Cross-Validation Folds

| Fold | Accuracy Score |
|---|---|
| 1 | 0.708 |
| 2 | 0.778 |
| 3 | 0.778 |
| 4 | 0.750 |
| 5 | 0.831 |

k times, with each of the folds used exactly once as the test set. The accuracy score represents the proportion of correctly classified instances in the test set. The table suggests that the model achieved varying levels of accuracy across the different folds, with the highest accuracy score of 0.831 obtained in fold 5, and the lowest accuracy score of 0.708 obtained in fold 1. And the mean score is 0.7694. Overall, the accuracy scores suggest that the model performed reasonably well in this cross-validation procedure.

#### 4.3.2 Identify the attributes of a given word that are associated with each classification.

We mainly use Confusing Word Number, Maximum Word Repetition, and Word Difficulty Level to subjectively measure the difficulty of the word, and we use the difficulty level we just classified by clustering to group the words and look at the average of each attribute of the word in turn.

Table 6: Comparison of Difficulty Attribute Levels

| Level | Confuse Num | Max Repetition | CEFR |
|---|---|---|---|
| 1 | 2.26 | 0.16 | 0.80 |
| 2 | 1.81 | 0.43 | 0.88 |
| 3 | 1.44 | 0.90 | 0.92 |
| 4 | 2.16 | 1.2 | 0.91 |

This table provides a comparison of difficulty attribute levels based on three different criteria: edit distance, maximum repetition, and Common European Framework of Reference (CEFR). The four levels (1-4) are listed in the first column, and the corresponding values for each attribute are shown in the following columns.

The first attribute is the number of confusing words, which represents the number of words that have an edit distance of less than 2 from common words. Edit distance is a measure of the similarity between two pieces of text, which is the number of single characters that need to be changed, inserted, or deleted to transform one word into another. A lower edit distance indicates greater similarity between words, while a higher value indicates greater differences between them. In the wordle game, words that are harder to guess should have more similar words, and therefore more confused words. This table shows that words at difficulty level 4 have the highest number of confused words.

The second attribute, maximum repeat rate, refers to the highest number of times a single character is repeated in a word. For example, the word "vivid" has a repeat rate of 4. Since there are only five blank spaces in the wordle game, people usually will not sacrifice two or more spaces

to guess the same word. As shown in this table, the maximum repeat rate gradually increases with the difficulty level, with words at difficulty level 4 having the highest repeat rate.

The third attribute, CEFR, is a widely used framework for describing language proficiency. The values listed in this column correspond to the CEFR level that corresponds to the text for each difficulty attribute level. The CEFR levels range from A1 (beginner) to C2 (proficient), with higher levels indicating greater proficiency. The values in the CEFR column indicate that text at difficulty level 1 is prepared for CEFR A1 level, while text at difficulty level 3 is prepared for CEFR B2 level, which is considered to be an intermediate to high level. It can be seen that the higher the difficulty, the greater the CEFR value.

It can be seen that this classification model has high interpretability.

### 4.3.3 Using your model, how difficult is the word EERIE?

The final predicted difficulty level of 3, which falls within the moderately difficult category, is highly reasonable. Firstly, this word is a rare word in the CEFR and probably most people don't use this word often and will have trouble remembering it. However, there are few similar words, and the confusion count is very low. Although the word contains three identical letters, it has a special structure. Many people can find the word more easily through search engines and other aids. Also, the normal distribution corresponding to this difficulty level is roughly similar to our predicted distribution in Section2. Hence, a difficulty level of 3 is easily comprehensible.

## 4.4 Question 4

1. The game Wordle launched to great fanfare and popularity in early 2022, but interest has waned in recent months, with fewer than 25,000 players logging in daily. However, among the remaining players, there has been a growing trend towards selecting the difficult mode. This is not surprising, as players who perform better in the game tend to find it more engaging and addictive.

2. While holidays may bring in more players overall, they are unlikely to significantly increase the number of people attempting the challenging levels.

3. As time goes on, the distribution of players has become increasingly skewed to the right, indicating a greater number of skilled players.

4. Difficult words in the game tend to feature high repetition of certain letters, rare vocabulary, and many words that look similar to each other.

5. Notably, on November 1, 2022, there was a sudden surge in players attempting the difficult mode. Some speculated that this was due to the fact that it marked the 500th Wordle competition.

# Memorandum

**To:** Puzzle Editor,
**From:** Team 2316712
**Date:** February 21th, 2023
**Subject:** A study on Wordle game

We are very happy to be exposed to Wordle, a word-guessing game, through the US competition. Guessing Wordle's word of the day has become an important spice for us during the intense competition process and heavy modeling tasks. The game has a very interesting setup that makes you want to play it. In addition to its fun, of course, the statistical ideas embedded in the game mechanics are well worth studying.

Our task is aimed to answer the question of what makes a certain word difficult to guess in the game. We proposed a new attribute called "Wordle Word Perplexity" to measure a word's similarity with other words, while also considering the different importance of different letters and the frequency of the confused word. The weight was found using simulated annealing, and the results showed that this attribute can increase the game's effect by 13%. It definitely right that the data quality determines the effect of your learning models.

The study also tackled three other questions: predicting the number of reports submitted on Twitter for problem 1, measuring the correlation between different attributes of the game, and measuring the game's difficulty. For each question, we used various techniques such as the PNP model, machine learning models, and the Kolmogorov-Smirnov test to provide comprehensive answers.

To predict the associated percentages of (1, 2, 3, 4, 5, 6, X) for a future date, it's recommended that you utilize the output of feature engineering and take it as input to a Multilayer Perceptron(MLP) model which performs best among machine learning model. To be honest, it supports why deep learning models can lead the innovation of our time.

For the problem of measuring the game's difficulty, it's recommended to use clustering indicators for the k-means algorithm and features above to train an MLP classifier. We concluded that the "EERIE" word has a difficulty level of 3, and we provided a reasonable explanation for their findings. Additionally, we considered the impact of the difficult mode on the data skewness and used the naive Bayes approach to adjust the skewness of the normal distribution, making the model more reasonable.

Overall, the study provides valuable insights into what makes a certain word difficult to guess in the Wordle game. It also offers new approaches and techniques that can be applied to further analyze and improve the game's mechanics. I believe this study will be of great interest to your readers, especially those who are passionate about puzzle games like Wordle.

Thank you for considering this letter for publication.

Sincerely,

# References

[1] *S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2010, pp. 63-67, doi: 10.1109/IITSI.2010.74.*

[2] *Citation: Tian Liu, Zeyu Zhao, Menglei Yao, et al. Establishment and application of SEIAR model[J]. Dis Surveill, 2020, 35(10): 934-938. doi: 10.3784/j.issn.1003-9961.2020.10.014*

# Appendix A: the weight of perplexity

"j":3.899006,"q":2.121849,"x":3.609389,"z":3.800280,"v":1.809270,"k":1.575361,

"w":2.142675,"g":2.683015,"h":0.799648,"m":1.152375,"a":0.304510,"e":-0.524403,

"r":0.872481,"s":0.738144,"o":0.842527,"i":-1.028403,l":1.222733,"t":-0.444665",

"n":0.469933,"u":1.132885,"b":0.285132,"c":0.763612,

"d":2.094804, "f":1.590406, "p":0.794989, "y":1.838979,

This weight is optimized by simulated annealing. It implies that the letter with higher frequency will get a lower weight.