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

**Title**

**Summary**

**Key words:**

Contents

[1. Introduction 2](#_Toc127745535)

[1.1 Restatement of the Problem 2](#_Toc127745536)

[1.2 Our Work 2](#_Toc127745537)

[2. Model Assumptions 2](#_Toc127745538)

[3. Notations 2](#_Toc127745539)

[4. Construct Word Features 3](#_Toc127745540)

[4.1 Words’ Natural Attributes 3](#_Toc127745541)

[4.1.1 Word Frequency 3](#_Toc127745542)

[4.1.2 Letter Frequency 3](#_Toc127745543)

[4.1.3 Word Gini Index 4](#_Toc127745544)

[4.1.4 Repeated and Consecutive Letters 4](#_Toc127745545)

[4.1.5 The Part of Speech 5](#_Toc127745546)

[4.2 Words’ Wordle Features 5](#_Toc127745547)

[5. Trend for The Number of Reported Results 7](#_Toc127745548)

[5.1 Modeling the Number of Reported Results with Breakpoint Regression 7](#_Toc127745549)

[5.2 Modeling the Percentage of Hard Mode 11](#_Toc127745550)

[6. Model the Results Distribution 14](#_Toc127745551)

[6.1 Modeling Distribution Using Static and Dynamic Features 15](#_Toc127745552)

[6.2 Evaluation of Fitting Model 16](#_Toc127745553)

[6.3 Rolling Prediction for Future and Uncertainties 18](#_Toc127745554)

[7. The Difficulty of Wordle Puzzle 20](#_Toc127745555)

[8. Interesting Insights of Dataset 22](#_Toc127745556)

[9. Conclusion and improvement 23](#_Toc127745557)

[10. Reference 24](#_Toc127745558)

1. Introduction
   1. Restatement of the Problem
   2. Our Work
2. Model Assumptions
3. Assume that
4. Notations

|  |  |
| --- | --- |
| Symbol | Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Construct Word Features

In order to analyze the target word in the subsequent modeling process, we hope to convert the string of each word into a numerical vector (i.e., features) to represent the characteristics of the word, including the structure of the word, the difficulty of guessing the word, etc.

We put forward two ways to construct the features of words. On the one hand, we consider the “nature” of words, such as the composition, the frequency of occurrence in the corpus, which we call the words’ natural attributes; On the other hand, starting from the Wordle game itself, we measure the uncertainty of the system from the perspective of entropy, and inspired by the information gain of the decision tree, we define the gain of Wordle word to describe how difficult it is to guess a given word, which is called words’ Wordle attributes.

* 1. Words’ Natural Attributes
     1. Word Frequency

For a guessed word , intuitively, if the word is used more frequently in people's daily life, the player will be most likely to associate these words after receiving certain feedback. According to the network, we found that the Wordle game contains nearly 13,000 legal words (we call it the valid word set ), and only about 2,300 of them will be the answers to the Wordle game (we call it the answer word set ).

Based on Google Web Trillion Word Corpus, we counted the frequency of each word  in valid word set and normalized to make the sum of frequencies equal to 1.

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

* + 1. Letter Frequency

We now focus on each letter  that constitutes a word . We will count the number of occurrences  of each letter in each position in valid word set , in which, , and then use two methods to normalize the frequency.

First, we fix the position  and count the frequency of 26 letters (formula [(4.2)](#formula4_2)). Second, we can fix the letter  and count the frequency of each letter in 5 positions (formula [(4.3)](#formula4_3)).

|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |
|  |  | (4.3) |

For a given word , we can construct a letter frequency vector with 10 elements in it as some attributes of  using letters .

|  |  |  |
| --- | --- | --- |
|  |  |  |

* + 1. Word Gini Index

For a discrete distribution , we can define the Gini index of  as:

|  |  |  |
| --- | --- | --- |
|  |  | (4.4) |

Gini index is like entropy and can be used to indicate the dispersion of a probability distribution. If a probability distribution is more dispersed, the Gini index is larger. As shown in [(4.5)](#formula4_5), we count the frequency of letters in  and use it to calculate the Gini index  of a word:

|  |  |  |
| --- | --- | --- |
|  |  | (4.5) |

For one of the most “complex” words, that is, the word does not contain duplicate letters, the Gini index  takes the maximum value:

|  |  |  |
| --- | --- | --- |
|  |  |  |

For those words that contain repeated letters, the Gini index will be less than 0.8, for example . The Gini index must be greater than 0, because no word containing five letters is all composed of the same letter.

* + 1. Repeated and Consecutive Letters

Both words “APPLE” and “FEVER” have repeated letters (“P” and “E”), so they have the same Gini index . However, the repetition is different. The two letters “P” in word “APPLE” appear continuously, and an attribute  is defined to indicate whether the word  contains repeated and continuous letters, as shown in [(4.6)](#formula4_6).

|  |  |  |
| --- | --- | --- |
|  |  | (4.6) |

* + 1. The Part of Speech

We assume that part of speech, as an attribute of a word, may also affect the difficulty of the Wordle game. We use 0-1 vector  to encode the part of speech of the word .

* 1. Words’ Wordle Features

Next, we construct the attribute of the word  from the rules of the Wordle game. We hope this attribute can measure the difficulty of guessing the word. The basic idea is to use entropy to measure our current uncertainty.

For a given word , assuming that there are total  answers and  valid guessing words in Wordle, the guess strategy for a player may follow as:

* At the beginning, we have no information about the guessed word . The uncertainty of the system is the largest, and the entropy of the system is :

|  |  |  |
| --- | --- | --- |
|  |  |  |

* In the first round, for any guess , players will receive a feedback containing an array of five elements (each has 3 color states), which provides us with information to reduce the uncertainty of the system.
  + In short, the size of the feasible state space of the system will be reduced from  to . Notice that  is related to guess  and target word .

|  |  |  |
| --- | --- | --- |
|  |  |  |

* + Inspired by the decision tree and we define the gain  brought by guess  as the reduction of system uncertainty as shown in formula [(4.7)](#formula4_7):

|  |  |  |
| --- | --- | --- |
|  |  | (4.7) |

* + If we try each word in the valid word set  in turn, we can define the possible average gain  of guessing for the word  in the first round. In formula [(4.8)](#formula4_8), the weight  represents the possibility that players choose  from the valid word set , which can be estimated by word frequency.

|  |  |  |
| --- | --- | --- |
|  |  | (4.8) |

* + Notice that  various with word . For a difficult target word , the possible gain  from trying all guesses is also small, and more guesses are needed to obtain more information in the next few rounds.  can be used as an attribute to describe the difficulty of the word.
* The above process can be deduced to the next few rounds. We search all possible guesses in the subset in turn, and then calculate the average gain brought by these guesses. There, we can define such as , the subscript represents the round of guess. This can be easily achieved through a recursive algorithm. Notice that the computational complexity of the recursive algorithm increases exponentially with the number of rounds. The time required to calculate  of one word will exceed several hours.

[Table 1](#table1) shows the recursive algorithm of calculating gain  for a given word  and a required round  ([Algorithm 1](#algo1)).

**Table 1**: The recursive algorithm of calculating gain.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 1**: | | | |
|  | **Input:** target word , search rounds , feasible word set , word frequency , valid word set  and answer word set | | |
|  | **Output:** Average gain of word  after  rounds of guessing | | |
| **1** | Initialize the set used to save the gain,  and | | |
| **2** | Initialize the set used to save the weights, | | |
| **3** | **for** guess word  **in** | | |
| **4** |  | Calculate the pattern  obtained when the target is  and the guess is | |
| **5** |  | , | |
| **6** |  | **if**  **and** | |
| **7** |  |  |  |
| **8** |  | **else** | |
| **9** |  |  |  |
| **10** |  | **end if** | |
| **11** |  | Add word frequency | |
| **12** | **end for** | | |
| **13** | Normalize the weights , | | |
| **14** | Calculated the average gain , | | |
| **15** | **return** | | |

We will supplement the implementation details of [Algorithm 1](#algo1). Because the calculation of gain requires a lot of search and recursion, it will take a lot of time. In the algorithm:

* given the target word  and the player guess , to calculate the pattern  that the player will receive (line 4), and
* for a given guess , find which answers can get the same pattern  to from constructing a new feasible set  (line 5).

These two steps can greatly be accelerated by storing the pre-calculated results in a hash table. In this way, only  time complexity is required for table lookup operation, without recalculation each time. [Figure 1](#Fig1) shows the calculation principle of gain .

|  |
| --- |
|  |

**Figure 1**: Recursive diagram of gain calculation.

1. Trend for The Number of Reported Results

In this section, we will discuss the modeling of the number of reported results and the percentage of scores reported that were played in Hard Mode, both of them can be regarded as time series analysis problems.

We introduce the breakpoint detection regression algorithm and decompose the number of reported results changing into four parts for modeling according to the change characteristics. In addition, we find that the attributes of the word in the previous few days will affect the percentage of Hard Mode scores in the future through an  process, which is intuitive. We will discuss the phenomenon in detail in section 4.2.

* 1. Modeling the Number of Reported Results with Breakpoint Regression

Visualize the change trend for the number of reported results  over time , we find that the change of  presents different trends at different stages.

* At the beginning, Wordle received people’s attention.  increases rapidly.
* After the  rose to the peak, the popularity of the game gradually decreased and  declined rapidly.
* Wordle games entered a stable attenuator in the second half of 2022, and  was still declining, but the decay rate was very slow.

We design a breakpoint detection regression algorithm ([Algorithm 2](#algo2)) to split the time series into different parts, and then model each part separately. For a time series , the core idea of [Algorithm 2](#algo2) is to find a breakpoint  to divide the series into two sub-series  and , then we run OLS regression for two sub-series with respect to time index . Assume that the square error of OLS on the two series is  and  respectively, the algorithm adjusts  to minimize  through dichotomy search.

[Algorithm 2](#algo2) is very efficient due to the use of dichotomy search. There is no need to iterate on every possible breakpoint and the time complexity of the algorithm is , in which,  is the length of the time series.

[Figure 2](#Fig2) shows the two breakpoints obtained by using breakpoint detection regression. Next, we will start from 2022-01-07 and divide the time series into four parts for modeling. The assumptions and corresponding models of each part are as follows:

* Part I: (From 2022-01-07 to =2022-02-01) Assumption of fast rising stage,  can be expressed by the following linear model:

|  |  |  |
| --- | --- | --- |
|  |  | (5.1) |

* Part II: (From =2022-02-01 to =2022-07-26) Assumption of fast exponential decay stage,  can be expressed by the following model. Assume the decay rate satisfies the following differential equation:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Then  has the following expression:

|  |  |  |
| --- | --- | --- |
|  |  | (5.2) |

* Part III: (From =2022-07-26 to 2022-12-31) Assumption of slow exponential decay stage,  can be expressed by the following model. Assume the decay rate satisfies the following differential equation:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Then  has the following expression:

|  |  |  |
| --- | --- | --- |
|  |  | (5.3) |

* Part IV: (After a long time on 2022-12-31, **Not Observed**) Assume there is a stable number of reported results  when time index  is large enough. Then  satisfies:

|  |  |  |
| --- | --- | --- |
|  |  | (5.4) |

The Part IV is the assumption of the future change trend for . The model parameters of the Part I, II, III models are estimated by minimizing the square error.

|  |
| --- |
| 图表, 折线图  描述已自动生成 |

**Figure 2**: The two breakpoints found by the breakpoint detection regression algorithm. It can be seen in the figure that the three curves obtained by breakpoints have significantly different change trends.

|  |
| --- |
|  |

**Figure 3**: The fitting result of the three models (Part I, II, III) to the data.

[Figure 3](#Fig3) shows the estimated model and the real data. The curve fits the trend of data change well. Test the residuals . The residuals  can pass the normality test at the significance level of 10%, but from the ACF plot of the residuals, the 1-Lag tail can be observed. Therefore, the residuals series do not conform to the independence assumption, which can be improved by establishing an  model for the residuals . For the sake of simplicity, we will continue the main clue of modeling and do not discuss it in detail here.

As for the prediction of future , we assume that the change trend of  will continue to maintain the assumption of Part III model (formula [(5.3)](#formula5_3)) until March 1, 2023. According to the estimated model parameters, we can obtain the point estimate of :

|  |  |  |
| --- | --- | --- |
|  |  | (5.5) |

Then we use the historical data to calculate the realized residual . Next, we can estimate the standard deviation  of the residual. Finally, for the given confidence level , We can construct confidence interval from point estimation  as shown in formula [(5.6)](#formula5_6):

|  |  |  |
| --- | --- | --- |
|  |  | (5.6) |

In which,  is the CDF of standard normal distribution.

After calculation, the model predicts that the number of reported results of March 1, 2023 is , and the prediction interval with 95% confidence level is .

**Table 2**: The breakpoint detection regression algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 2**: | | | |
|  | **Input:** time series array , the number of iterations | | |
|  | **Output:** the best breakpoint | | |
| **1** | Initialize the search range, , . Initialize | | |
| **2** | Initialize the set used to save the square error, | | |
| **3** | **while** | | |
| **4** |  | Split the array into two parts, , | |
| **5** |  | Run OLS for  and  respectively to obtain the square error , | |
| **6** |  | **if** | |
| **7** |  |  |  |
| **8** |  | **else** | |
| **9** |  |  |  |
| **10** |  | **end if** | |
| **11** |  | Save square error, , update iteration index | |
| **12** | **end while** | | |
| **13** | **return** | | |

* 1. Modeling the Percentage of Hard Mode

We use  represents the word attributes of the word when the date is , and we use  to describe the percentage of scores reported that were played in Hard Mode. Notice that  has nothing to do with  because  is an "afterthought attributes". It is impossible for players to see the word  to guess today in advance (otherwise, they can obtain information  about it), and then decide whether to play the Hard Mode.

In fact, our assumption is that at date , the word attributes for the past few days, i.e.  will have an impact on . The intuition of this conjecture is that:

* If the words in the past few days are easy to guess, more people will choose the Hard Mode to challenge themselves the next day.
* If the words in the past few days are difficult, most people will choose to give up the Hard Mode in the next day.

Next, we will verify the correctness of the above assumption through time series analysis.

First, it is observed that  has a trend of growth over time. The growth rate gradually slows down and finally approaches to a steady state. We assume that  meets the model,

|  |  |  |
| --- | --- | --- |
|  |  | (5.7) |

and use the exponential model to model the trend term  as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (5.8) |

The above model has the following two intuitive realistic explanations:

* The number of players who join in the fun gradually decreases.
* The percentage of hardcore players among the remaining players continues to rise, and finally reaches a stable state.

[Figure 4](#Fig4) shows the fitted result of trend term . The next step, we move to check the residuals time series . [Figure 5](#formula4_5) shows the histogram, time varying characteristic, ACF and PACF plot of residuals. After checking we notice that:

* After normality test,  does not obey the normal distribution.
* After ADF test,  is not a stationary time series.
* In the ACF,  has a serious lag,  is autocorrelated.

|  |
| --- |
|  |

**Figure 4:** The fitted trend term model for the percentage of Hard Mode.

|  |
| --- |
|  |

**Figure 5:** The (a) histogram, (b) time series, (c) ACF and (d) PACF plot of residuals. It can be clearly observed that the residual time series is not stationary, and there is a serious lag in the ACF.

|  |
| --- |
|  |

**Figure 6**: The (a) histogram, (b) time series, (c) ACF and (d) PACF plot of residual changes. At this time, the time series becomes normal and is stationary, we can see an obvious AR(1) characteristics in ACF.

We carry out first-order difference for residual  and define sequence  and :

|  |  |  |
| --- | --- | --- |
|  |  | (5.9) |

As shown in [Figure 6](#Fig6), the same test is performed on  again. The sequence after difference can pass the normality test and stationarity test. It is observed from ACF that  still has first order autocorrelation.

We assume that the random variable  can be explained by two parts of information, one is about the auto-regression of  (i.e. an  process), and the other is the word attribute of the previous day . We build the model as shown in formula [(5.10)](#formula5_10):

|  |  |  |
| --- | --- | --- |
|  |  | (5.10) |

notice that we use the historical word attributes , so there is no lookahead bias in [(5.10)](#formula5_10).

Now we run an OLS to fit residual changes . Filter out most of the insignificant word attributes, and the final fitted model statistics are shown in [Table 3](#table3). From the results,  is most relevant to the information of its 1-order lag . Only Gini index  is significant (level 5%) in the word attributes. The significance of  (whether there are repeated and continuous letters in ) and (2-round gain) is close to 10%. The other attributes have no explanatory power. The whole model adjusted R-squared is 0.219. According to the F-statistic, the regression model is significant.

Attributes (especially ) do have an impact on , combine [(5.9)](#formula5_9) and [(5.10)](#formula5_10), we have:

|  |  |  |
| --- | --- | --- |
|  |  | (5.11) |

We substitute the above model into the expression [(5.7)](#formula5_7), and the final model is:

|  |  |  |
| --- | --- | --- |
|  |  | (5.12) |

in which, . Therefore, over time, the impact of the target word attributes  in the previous few days on the current percentage  decreases exponentially. In our model,  is about 0.354, so the influence of the word attributes five days ago  on the current  is only close to 1%.

**Table 3:** The OLS results of residual changes fitting.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OLS** | Adj. R-squared: | 0.219 | P (F-statistic): | 3e-18 \*\*\* |
| **Features** | | Coef | std,err | p-value |
| **Lag Term** |  | -0.3540 | 0.050 | 0.000 \*\*\* |
| **Word**  **Attributes** |  | -0.1229 | 0.062 | 0.049 \*\* |
|  | -0.2945 | 0.185 | 0.113 |
|  | -0.1744 | 0.115 | 0.131 |

To sum up, from [(5.12)](#formula5_12), we can see that the word attributes of the past few days are superimposed through an  process, thus affecting the Hard Mode percentage  at the time of date , which will decay exponentially in several days. The above discussion verifies our conjecture at the beginning of this section. At the same time, we get from the regression model [(5.10)](#formula5_10) that the Gini index has the greatest impact on the  of all defined word attributes.

1. Model the Results Distribution

In this section, we will discuss the Wordle results distribution modeling. There are 7 possible states for each day’s results (1, 2, 3, 4, 5, 6 and X). [Figure 7](#Fig7) shows the distribution various with time. We use a linear function to model the contribution of the features to the labels, and then use the softmax function to convert the output into a probability distribution. Finally, the model parameters are obtained by maximum likelihood estimation (MLE).

|  |
| --- |
|  |

**Figure 7:** The distribution of daily results changes over time. This change may be related to the difficulty of Wordle words and the level of players.

* 1. Modeling Distribution Using Static and Dynamic Features

For each possible state , each word  and time index , assume the distribution we want wo predict is :

|  |  |  |
| --- | --- | --- |
|  |  | (6.1) |

we define the logits , i.e., the contribution of the features to the distribution:

|  |  |  |
| --- | --- | --- |
|  |  | (6.2) |

in which,  is the corresponding coefficient of state .

*  is the static feature of the model.  contains the word attribute of the word , In other words, we have . Every time the word  is given,  is determined.  is only related to the word and has nothing to do with the player's level.
*  is the dynamic feature of the model, which is related to players and date . We design the following three features:
  + Current time index  is used to capture the trend term in model.
  + Hard Mode percentage  discussed in Section 5.2. We assume that the more players in Hard Mode, the higher the overall level of the players. On average, players only need to try less guesses to win, so the distribution is skewed to the right.
  + In the past period (taking the window size parameter ), the player’s average performance  is calculated:

|  |  |  |
| --- | --- | --- |
|  |  | (6.3) |

 is the average value of the distribution of the past  days. It can be used to measure the ability (game level) of players in the recent period.

Now, we can define our distribution model using softmax transform:

|  |  |  |
| --- | --- | --- |
|  |  | (6.4) |

we estimate the model parameters  by minimizing the Cross Entropy loss :

|  |  |  |
| --- | --- | --- |
|  |  | (6.5) |

and use KL divergence  to measure the error between the predicted distribution and the real distribution. The smaller the value of , the more accurate the distribution prediction is.

|  |  |  |
| --- | --- | --- |
|  |  | (6.6) |

* 1. Evaluation of Fitting Model

When fitting, we split the whole time series into training set and validation set. This kind of random split ensures that the training set is always in front, and the validation set is behind, so there will be no lookahead bias.

[Figure 8](#Fig8) shows the model coefficient matrix estimated when . The horizontal axis represents the features (29 in total), and the vertical axis represents the components of the states (7 in total). The brighter the color of the part in the heatmap represents the stronger the interpretation ability of the feature to the distribution.

|  |
| --- |
|  |

**Figure 8**: The heatmap of estimated model parameters. The brighter the color is, the more significant the coefficient is, and the more important the corresponding feature is for distribution prediction. The darker the color is, the closer the absolute value of the coefficient is to 0, and the less important the feature is.

We can also get some direct insights from [Figure 8](#Fig8). For example, from the perspective of letter frequency, the frequency of the first letter is more important than the frequency of the letter at the end of the word. The word frequency itself does not play much role. The part of speech of words and the average performance of players in the past few days are very helpful for prediction.

We use cross validation to select the window size parameter . [Figure 9](#Fig9) shows the change trend of KL divergence  of training set and validation set with . Considering bias-variance tradeoff, with the increase of , the  of the training and validation set decreases, the model fitting ability increases. When  continues to increase, the  of the training set continues to decrease, but that of the validation set increases. Currently, the generalization error increases, the model appears overfitting. We think  is a good choice.

|  |
| --- |
|  |

**Figure 9**: The KL divergence of training set and validation set with different window size parameter. When the length of the window is greater than 7, we can observe overfitting.

[Figure 10](#Fig10) shows the comparison between the predicted results and the real distribution of the two samples in the training set and the validation set. The model can grasp the overall characteristics of the distribution and there is a small amount of error in each state space.

We consider the errors  of distribution prediction. [Figure 11](#Fig11) shows the boxplot of the residuals. The estimation of the model in the training set is accurate and the performance in the validation set is slightly biased, but the error distribution is concentrated and acceptable. The mean absolute error of  is 0.0313. The possible reason for this bias is that we do not model the time series trend of distribution changes in detail. We will discuss it in the model improvement section.

|  |
| --- |
|  |

**Figure 10**: Comparison of predicted distribution and real distribution.

|  |
| --- |
|  |

**Figure 11**: The boxplot of distribution prediction error. We can observe that the point estimation of the training set is accurate, and the point estimation of the validation set is slightly biased, but the error distribution is relatively concentrated. The validation set mean absolute error of each state is displayed.

* 1. Rolling Prediction for Future and Uncertainties

For a given date  and a given target word , we can construct the static features  using , and get a rough prediction for Hard Mode percentage  using model [(5.8)](#formula5_8). However, in order to predict the distribution at , model [(6.4)](#formula6_4) needs all the previous  days’ distribution information . This part belongs to future information and cannot be obtained at present. We can only fill in the distribution between the last day  of the dataset and the target date  step by step through rolling prediction method.

For rolling prediction procedure, we start from the last day  of the dataset. Suppose we know the target words  to the Wordle game every day. Then, we can use the word  to construct static features , and use  (the past  days distribution) to construct dynamic features  with predicted Hard Mode percentage . Combined with  and , we can predict  using [(6.4)](#formula6_4). Next, we update the time index to , take the predicted  as the historical distribution, use the same process to predict , and repeat the above rolling prediction procedure until reach the target date .

The above rolling prediction method includes two main uncertainties:

* [(6.4)](#formula6_4) needs true history results distributions to build . We use the predicted value to fill in the missing information about future distribution. This may cause uncontrollable error accumulation, to reduce the reliability of the results of long-term prediction.
* In practice, we don't know the target word list  for each day in the future. Different word list, or words in the same list appear in different order, will make the final prediction result different.

Therefore, for the given date  and word , we have to assume the target word list  between  and . We sample from the unused answer set  of Wordle game to construct the list. The predicted distribution of the date  obtained from each sampling are different. Then we repeat the above process  times through the bootstrap method, and finally take the mean value as the point estimate of the distribution. We can also construct confidence intervals from the prediction results of the  bootstrap predictions to obtain the knowledge about the reliability of the results.

We take bootstrap times , and [Figure 12](#Fig12) shows the prediction for the word “EERIE” on March 1, 2023. From the result, “EERIE” is a difficult word to guess. Only about 5% of players can win the game within three tries, and more than 40% of players need five attempts to win the game. [Figure 12(b)](#Fig12) shows the bootstrap standard deviation, which describes the confidence level of the model for each state prediction. The maximum standard deviation is 0.007, which indicates that the rolling prediction method is stable, the prediction result does not fluctuate too much. The point estimation of distribution prediction and its 95% confidence interval prediction (as a subscript) is as follows:

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |
| --- |
|  |

**Figure 12**: The predicted distribution for the word “EERIE” on March 1, 2023. From the (b), the rolling prediction results are stable, without too large numerical fluctuations, so the prediction results are reliable.

1. The Difficulty of Wordle Puzzle

In this section we’ll discuss the classification of solution words by difficulty and how word attributes influence during the classification. We’ll extract a measure of difficulty for the words based on the daily participants’ performance, and then turn the problem into a multi-classification task by assigning difficulty labels to each word.

We’ll discuss the influence of different assignment strategies on the accuracy of the model and use the best one to classify the difficulty level of the given word. Also, in section 7.2, we will evaluate and get an overall view of the importance level of each word attribute to the classification model.

* 1. Multi-Classification Model Measured by Difficulty

The difficulty of the solution word actually has an implicit relationship, which we’re going to find later in this section, with the attributes we found in former chapters. However, based on the numbers of tries that participants take to solve the wordle, we can extract a measure of the difficulty of the word. Here we use the average number of tries taken to solve the wordle to measure the level of difficulty of the word. The value of the word difficulty indicator can be expressed as:

|  |  |  |
| --- | --- | --- |
|  |  | (7.1) |

Here represents number of tries that participant take, is the number of those who use tries to solve the wordle. What needs to be noticed here is that, for those who takes more than 6 tries or not be able to solve the wordle, we use 7 to represent their number of tries. is the summation of numbers of participants of different tries.

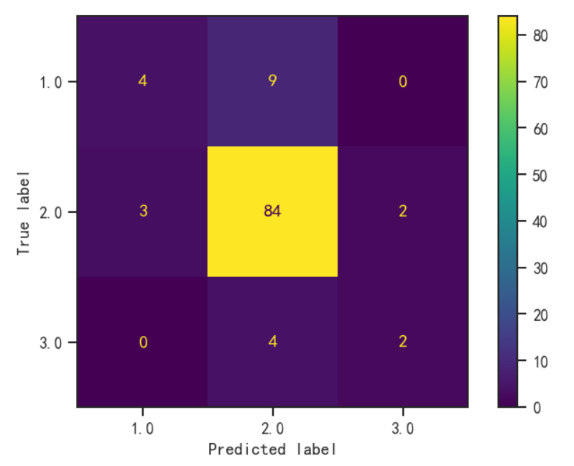
In this way, the difficulty of the word can be represented by a float number. In order to make the results of the classification easier to understand and use, we assign difficulty level indexes 1,2 and 3 to each word according to the value of difficulty indicator we constructed above. We can regard the indexes 1,2 and 3 practically as levels of easy, medium and hard difficulty, respectively. We first split the range of by percentile and percentile and assign difficulty index for in each interval:

|  |  |  |
| --- | --- | --- |
|  |  | (7.2) |

After labeling the words with difficulty indexes, we can do the classification with the word attributes. Considering that it’s a multi-classification problem, we use RandomForestClassifier here to get a better interpretation of the model. The accuracy of the model on the training set after cross-validation is about 0.64 while on the testing set is about 0.564.

We found that the model’s accuracy has a relationship to how we split the range of difficulty value, which is how we assign the labels. Therefore, to further improve the accuracy, we tried partitions of difficulty value by different percentiles to assign the labels and run the model again. The results of grid search shows that choosing percentile and percentile as the split points to assign difficulty indexes will get the best average accuracy score on training set as 0.808 and the accuracy on the test set as 0.8148. This means the best assignment strategy will be:

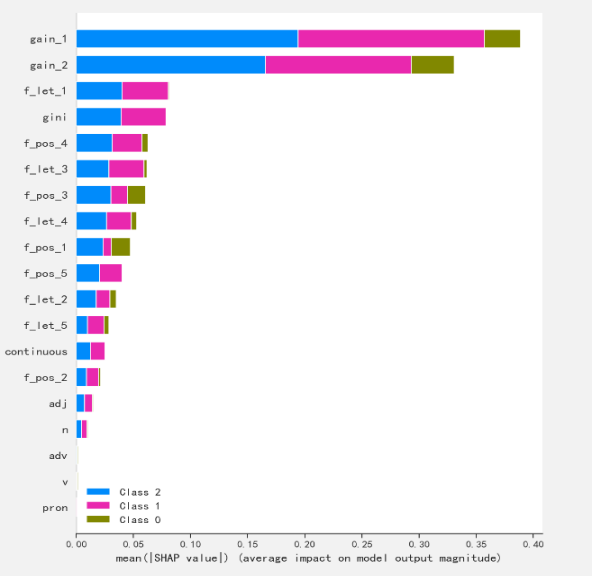
|  |  |  |
| --- | --- | --- |
|  |  | (7.3) |

 To better illustrate the validation and accuracy of our model, we give the visualization result of the confusion matrix of the model below in Figure 13:

**Figure 13**: The confusion matrix of the classification model

From the confusion matrix, we can find that our model displays a good performance of recognizing the exact or close difficulty level of the words since it won’t classify words of easy level as hard ones, nor will it classify the hard words as easy ones. The number of misclassification words is relatively quite small as well.

From the values of feature importance in the model, we find that word attributes that are most associated with the classification are feature gain\_1 and gain\_2.

To get a more explicit illustration of the difference among the influences of the word attributes to the classification, we introduce SHAP value, which is created to help interpretate the outputs of machine learning algorithms. We take the average of the absolute SHAP values for each feature as the measure of their importance to the model and get the bar graph as below in Figure 14. The figure gives the same results as the common values of feature importances.

**Figure 14**: The summary plot of SHAP values for each feature

Based on the summary of the model we get above, we can now construct the features for word EERIE and make classification for it. As the result shows, the classification of the difficulty level for it is XXX. Interpretation:

1. Interesting Insights of Dataset

Inspired by [1], we counted the frequency of letters as the first letter of a word in the valid word set , and counted the distribution of the letter in five positions, shown in Figure X.

* About 30% of all valid words begin with the letters S, E and A, while almost no words begin with the letters X and Q.
* Compared with the average number of guesses of words, it is also found that the higher the frequency of the first letter, the less the average number of guesses required.

Figure X contains a large amount of deep information about the word structure that can be mined, which will affect people's decision-making when playing Wordle.

|  |
| --- |
|  |

We have not intuitively presented the relationship between the difficulty of words and the constructed gain attribute . Figure X shows the strong interpretation ability of feature  to the average number of tries.

|  |
| --- |
|  |

1. Conclusion and improvement
2. Reference