Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2316240

WordleMetrics: Predictive Modeling and Difficulty Classification for Word Puzzles

Summary

With the prevalence of the word game named "Wordle", there has been a craze in solving the word puzzle on the social media around the world. It is of great significance for the game developers and maintainers to analyze and identify patterns and trends in player behaviors.

In order to approach the problem, a comprehensive and in-depth analysis is carried out on the dataset. In particular, by identifying and extracting word attributes, we aim to model the distribution of player scores in a predictive manner, and attempt to classify the difficulty of words based on their intrinsic and social characteristics. Specifically, we start with data preprocessing and exploratory data analysis, which removes noise in the dataset and renders us enlightening insights into the data. Then we adopted **Autoregressive Integrated Moving Average (ARIMA)** algorithm to study the pattern in previous dates. With Shapiro-Wilk Test and Ljung-Box justifying the fit, the model produce the prediction interval of reported number of posts on a future date. Subsequently, a **correlation analysis** is carried out, which turns out denying the existence of a causal relationship between multiple word features and hard mode ratio.

Additionally, we use a **multi-output support vector regressor (MOSVR)** to map the relationships between word attributes and score distribution. In this way, the correlations among output variables are taken into account, so that outputs with higher accuracy may be generated. In order to avoid the limitation of the regressor, the idea of **partial regression** is used. In the difficulty **classification** model, intermediary results from Task 2 are fed into the **Generalized Additive Model (GAM)** and produce outputs with ternary labels. With a sensitivity analysis, the stability and robustness of the classification model are confirmed.

During the establishment of models, interesting features in the dataset are spotted as well. The percentage of 4 tries shows obvious inconsistency with other scores in terms of correlation with neighboring columns, regression fit and monotonicity by word difficulty. Apart from that, we observe that the trend of hard mode ratio goes up while the average player scores stay still with fluctuations.

In the end, we conducted sensitivity analysis, which shows that our model is stable in predicting average scores from input word features. Then we analyze the strengths and weaknesses of our models. In a nutshell, although some assumptions may be somewhat speculative and imperfect, our models demonstrate excellent capabilities in distribution prediction and difficulty classification, with the stability and robustness maintained to a large extent. However, due to the nature of the game, various random factors besides word attributes may impact the predicting effectiveness of our models.

Keywords: ARIMA algorithm; Multi-output SVR; Classification; Partial regression; GAM; Feature extraction; Correlation analysis

Contents

1 Introduction

1.1 Background

A word game named Wordle has gained immense popularity, particularly since it began featuring as a daily word puzzle on the New York Times website. The game challenges players to guess a five-letter word within six tries, with feedback provided after each guess. To increase the difficulty and playfulness for experienced players, the game includes a "Hard Mode" that requires players to use any letters that they guess correctly in subsequent attempts. As many players have posted their scores onto social media, corresponding analysis may be carried out to identify patterns and trends in player behaviors.

1.2 Problem Restatement

In this problem, we are given a data file which aggregates some reported data of user playing Wordle, including dates, solution words, number of reported results, number of results in hard mode and score distribution in percentage for each word. The data set contains change pattern of results number with date, and various attributes within words can be uncovered, thus revealing the relation between score distribution and word features. We are required to only use the data to solve the following problems:

- Analyse the data set in qualitative and quantitative manners, conduct explanatory data analysis to obtain patterns, indicators and invisible relationships across features.
- Based on the data analysis, the following list of problems should be further addressed:
  - Observe and analyse how the number of reported results evolves over time and design a model to explain the variation of number.Create a prediction interval for a solution word on a future date.
  - Construct and extract attributes from words, determine if certain word features may affect the reported percentage of scores in hard mode.
  - Fit a model to describe the relation between word features and score distribution (in percentage), apply the model on "EERIE", then evaluate the robustness and stability of our model.
  - Develop a model to classify words by their difficulty using word features, and apply the model on "EERIE".
  - Discuss other possible attributes of the dataset.
- Write a letter to the Puzzle Editor of New York Times by summarizing the result of our analysis and proposing recommendations.

1.3 Our Work

Throughout the course of model establishment, after preprocessing the data, we use ARIMA model to predict the number of recorded results. An MOSVR model is the core part to partially predict the distribution of different number of tries given an arbitrary word. A GAM is used together with SVR to give the whole prediction. Additionally, intriguing features are discussed in Task 4. The complete picture of workflow is shown in Figure 1.

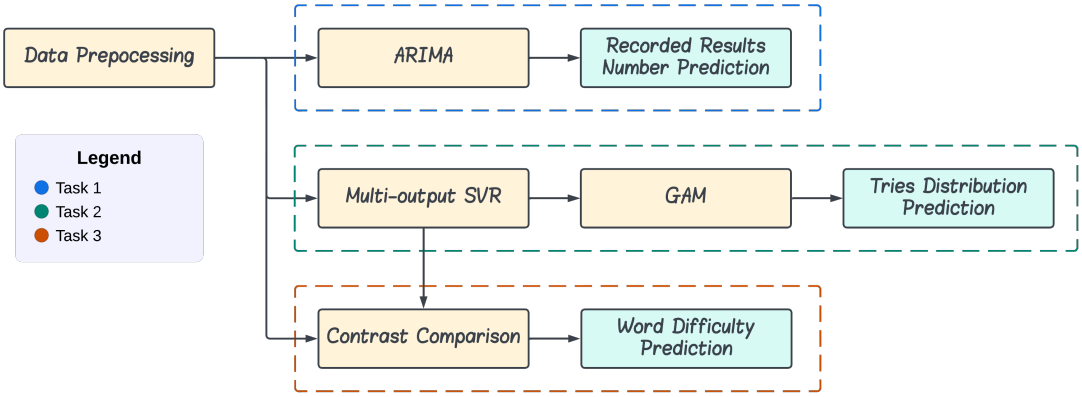

Figure 1: Workflow chart overview

2 Assumptions and Notations

2.1 Assumptions

In order to facilitate model construction, the following assumptions are made:

- It is assumed that the distribution of player scores can generally reflect the difficulty level of the solution word.
- We consider cases where the solution is guessed correctly by players in the first shot matter of luck, since no prior feedback was given.
- The given dataset is considered a sample collection of data randomly extracted from the complete dataset.
- We use the expected score as an indicator for the difficulty of a solution word.
- For a simple linear regression as given below:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, 2, \cdots, n \tag{1}$$

  The following assumptions are made:

  - $E(\epsilon_i) = 0$
  - $Var(\epsilon_i) = \sigma^2$
  - $\epsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$

Symbol	Definition
∇	Differencing operator of a time series
W_t	An arbitrary time series
X_t	Time series of the total number of reported results
Y_t	X_t with log transformation
Z_t	X_t with log transformation and differencing
ϕ_i	Coefficients contributed to AR part in an ARIMA model
θ_j	Coefficients contributed to MA part in an ARIMA model
a_t	A sequence of *i.i.d.* random variables, each with zero mean and same variance
σ_a^2	The variance of a_t
r_k	ACF for a stationary time series at lag k
ϕ_{kk}	PACF for a stationary time series at lag k
F_w	Frequency of a word in a huge text corpus
F_l	Frequency of a letter in the dataset
$\hat{X}_t(k)$	k-day forecast of X_t with forecast origin at t
$\hat{Y}_t(k)$	k-day forecast of Y_t with forecast origin at t
ρ	Pearson's correlation coefficient
D_i	Training dataset of support vector machine.
$\mathcal{K}$	Radial Basis Function
$\boldsymbol{x}^{(i)}$	Feature vector of the i-th sample
$\boldsymbol{y}^{(i)}$	Target vector of the i-th sample
C	Regularization coefficient

2.2 Notations

3 Data Preprocessing and Analysis

3.1 Data Preprocessing

In order to facilitate our analysis and construction of models, we perform data cleaning to the dataset with the following modifications:

3.1.1 Data Cleaning

- For words whose length is not equal to 5 characters, such as "tash", "clen", "rprobe", "favor ", we process identifiable words by amending them, and we remove those with higher uncertainty.
- For words that are not composed by English letters only, such as "naïve", we replace the non-English letters with letters in English alphabet.
- For number of reported results whose values are significantly inconsistent with proceeding or subsequent days, such as "study" with 2569 reported results, we remove the whole record.
- For misspelled words such as "marxh", we remove the whole record.

- Owing to truncation errors, the sums of all scores range from 98% to 102%. In case of this error, we normalize the average scores by dividing their corresponding score sums.

3.1.2 Feature Extraction

To make our models more descriptive and increase their performance, we observed and extracted the following word attributes that may be relevant:

- The frequency of letters in five positions of a word and the frequency of letters in general (sum the values of all positions).
- The number of distinct letters in a word.
- The frequency of letter combinations (specifically, 2 characters) in four positions and in general.
- The number of syllables in a word.
- The frequency of words in daily usage.
- Part of speech of a word.
- Whether a word is a homophone.

3.2 Exploratory Data Analysis

Based on the processed data, we conduct exploratory data analysis with statistical tools. It can be observed from Figure 2 that the trend of reported numbers starts to increase with fluctuations and makes a relatively smooth downturn from the February of 2022.

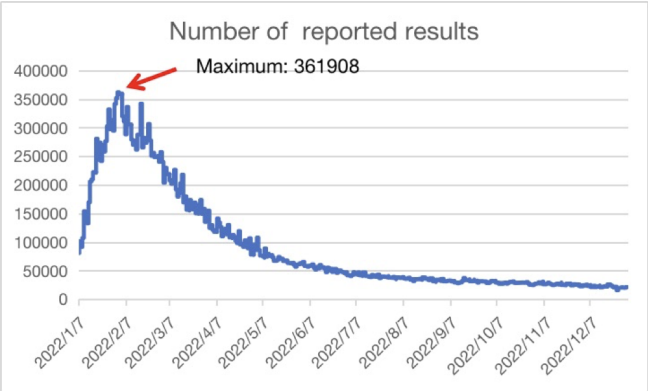

Figure 2: Time series of result numbers

It is generally acknowledged that players tend to take more guesses on difficult solution words. Therefore, we take the weighted average of scores of each word. Figure 3 demonstrates the average number attempts of players by date. It can be seen that while average attempts needed to get the most of words are close, there are some words that seem particularly difficult, which may serve as resources for further study and model establishment.

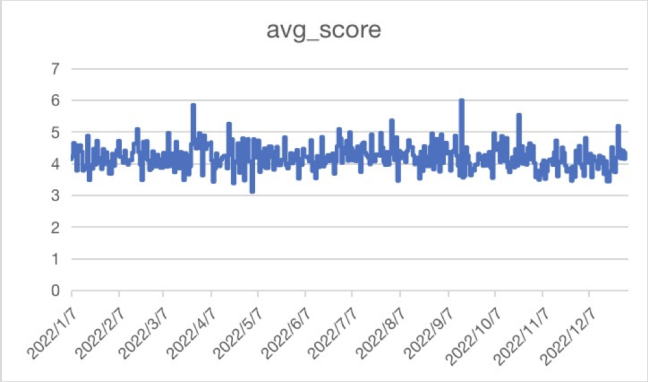

Figure 3: Time series of average number of scores

4 Task 1: Predictive Modeling with ARIMA and Correlation Analysis on Word Features v.s. Hard Mode Ratio

In this section, we will fit the result numbers by ARIMA. With log transformation, the variance of the model is largely stabilized. We thus create the prediction interval for the solution word on March 1st in 2023. Additionally, we will conduct correlation analysis on word attributes and the reported percentage of scores in hard mode, finally determine if word features may affect the portion of reported scores that were played in the hard mode.

4.1 Autoregressive Integrated Moving Average Model on Report Numbers

4.1.1 Model Construction

From the pre-processed data, we may observe that the curve is generally smooth without abrupt changes in value. In order to predict future values while taking past trend into account, we adopted ARIMA (Autoregressive Integrated Moving Average) [6]:

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \tag{2}$$

where

$$a_t \sim WN(0, \sigma_a^2) \tag{3}$$

After the introduction of ARIMA models, we simply fit one ARIMA(1,0,1) model to X_t. The prediction results look decent after we plot the true values and predicted values as shown in Figure 4. However, when we look deeper into the plot of residuals as shown in Figure 5, residuals variances seem to vary significantly, particularly in the first few months in 2022. This is traditionally regarded as heteroskedasticity. A potential explanation of this scenario could be the wellspring growth of players during this period which is approximately the initial stage of the prevalence of Wordle.

To alleviate the heteroskedasticity issue, we take the log-transform of the number of reported results. Plots of the number of reported players before and after log-transformation are shown in Figure 2, 6. For our following analysis, we focus more on the log-transformed data, which is notated as Y_t.

To construct an appropriate ARIMA model, there are three hyperparameters we need to determine, namely p, d and q. Firstly, it is intuitive to observe the non-stationary of mean in the log-transformed

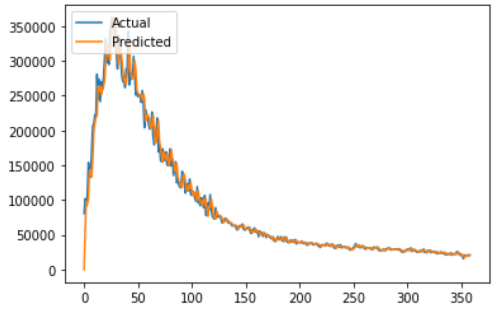

Figure 4: ARIMA(1,0,1) fitted values of X_t

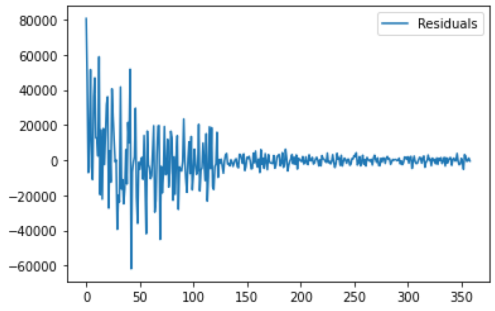

Figure 5: ARIMA(1,0,1) residuals of X_t

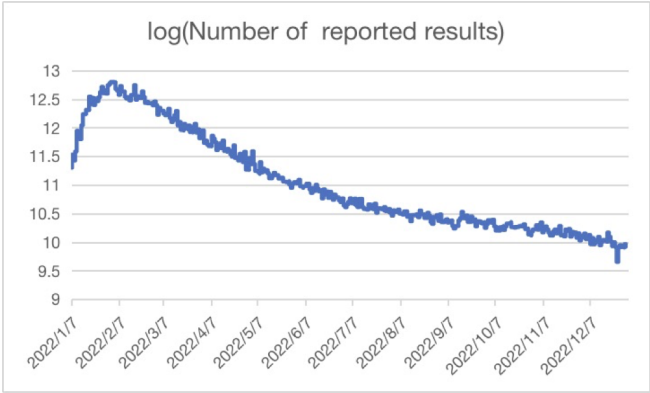

Figure 6: Time series of result numbers after log transformation

data in Figure 6. One possible method is differencing Y_t to get Z_t, which follows the Formula 4.

$$\nabla Y_t = Y_t - Y_{t-1} \triangleq Z_t \tag{4}$$

To verify whether Z_t is a stationary time series, we perform the Augmented Dickey–Fuller (ADF) test whose null hypothesis H_0 is that a unit root is present in a time series sample [7]. In other words, H_0 assumes the time series is not stationary where H_1 states the time series is indeed stationary. If the test statistic is more negative, then we have more confidence to reject H_0 and hence conclude that the targeted time series is stationary. After we conduct the ADF test to both time series, Y_t and Z_t, we found that the ADF statistic for Z_t is more negative with the value of -5.9009 compared to Y_t (-2.8751). Also, the p-value of ADF test of Z_t is far less than 10^{-4}, which indicates that we can even reject H_0 at 0.0001 significance level. Hence, Z_t is assumed to be stationary, which satisfies the stationary assumption of fitting an ARMA model. Therefore, we choose d as 1 in the ARIMA model. Additionally, the results of ADF test are shown in Table 1.

To further determine the values of p and q in the ARIMA model, we plot the autocorrelation function (ACF) [5] and partial autocorrelation function (PACF) [6]. For a stationary time series W_t, the ACF (r_k) is defined as:

$$r_k = \frac{Cov(W_t, W_{t-k})}{\sqrt{Var(W_t)Var(W_{t-k})}}, \ k = 1, 2, \cdots \tag{5}$$

The PACF (ϕ_{kk}) is defined as the correlation between W_t and W_{t-k} after removing the effect of the

	Y_t	Z_t
ADF Statistic	-2.8751	-5.9009
p-value	0.0483	2.7801×10^{-7}

Table 1: Table of ADF test results

intervening variables W_{t-1}, W_{t-2}, ···, W_{t-k+1},

$$\phi_{kk} = Corr(W_t - P_{1,W_{t-1},W_{t-2},\cdots,W_{t-k+1}}W_t, W_{t-k} - P_{1,W_{t-1},W_{t-2},\cdots,W_{t-k+1}}W_{t-k}),\ k = 1, 2, \cdots \quad (6)$$

where

$$P_{1,W_{t-1},W_{t-2},\cdots,W_{t-k+1}}W_t = \widehat{\beta}_0 + \widehat{\beta}_1 W_{t-1} + \widehat{\beta}_2 W_{t-2} + \cdots + \widehat{\beta_{t-k+1}} W_{t-k+1} \quad (7)$$

and

$$(\widehat{\beta}_0, \widehat{\beta}_1, \cdots, \widehat{\beta_{t-k+1}}) = \underset{\beta_0,\beta_1,\cdots,\beta_{t-k+1}}{\arg\min} E(W_t - \beta_0 + \beta_1 W_{t-1} + \beta_2 W_{t-2} + \cdots + \beta_{t-k+1} W_{t-k+1})^2 \quad (8)$$

Specifically, if ACF decays almost exponentially towards zero after lag q_0, we then suppose q to be q_0. Similarly, if PACF decays almost exponentially towards zero after lag p_0, we then suppose p to be p_0. From and after lag 1, we use Bartlett's approximate confidence interval (CI) to decide p and q. As shown in Figure 7, it is observed that there is a significant ACF out of the light blue 95% Bartlett's CI at only lag 1. Thus, we assume q to be 1. Also, p can be determined as 1 in Figure 8 using the same strategy. Therefore, the ARMA(1,1) should be an attempted model choice for Z_t. Equivalently, it is to say that ARIMA(1,1,1) is one of the potential models for Y_t.

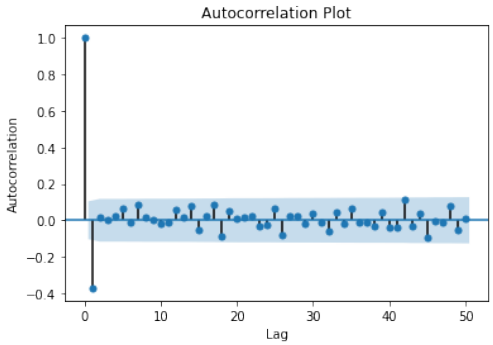

Figure 7: ACF of Z_t

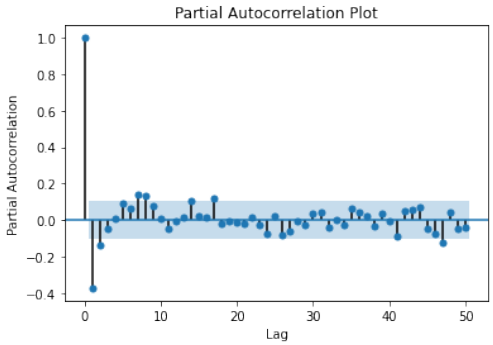

Figure 8: PACF of Z_t

4.1.2 Model Fitting

In this subsection, we first fit the ARIMA models to Y_t. As illustrated in Section 4.1.1, ARIMA(1,1,1) is a good model to start with by direct observation of ACF and PACF plots shown in Figure 7, 8. We have tried all combinations of (p,q) pairs ranging from (0,0) to (2,2). A summary table of coefficients together with other statistics is shown in Table 2.

Model	AIC	BIC
ARIMA(0,1,0)	-696.511	-692.631
ARIMA(0,1,1)	-753.748	-745.987
ARIMA(0,1,2)	-752.919	-741.277
ARIMA(1,1,0)	-747.880	-740.119
ARIMA(1,1,1)	-752.719	-741.078
ARIMA(1,1,2)	-779.495	-763.973
ARIMA(2,1,0)	-752.424	-740.782
ARIMA(2,1,1)	-751.015	-735.493
ARIMA(2,1,2)	-761.315	-741.912

Table 2: Summary of different ARIMA models

In order to quantify the model fitness among ARIMA models with different hyperparameters, we use Akaike information criterion (AIC) and Bayesian information criterion (BIC) [2]. For both standards, negative values with greater absolute values are preferred. From Table 2, we find ARIMA(1,1,2) model outperforms the other models in both AIC and BIC. A more detailed table of coefficients together with some other statistics of ARIMA(1,1,2) is shown in Table 3.

	Coefficient	Std. Error	z	$P > \|z\|$	95% Lower Bound	95% Upper Bound
ϕ_1	0.9918	0.004	250.267	0.000	0.984	1.000
θ_1	-1.5580	0.037	-42.360	0.000	-1.630	-1.486
θ_2	0.6116	0.036	16.794	0.000	0.540	0.683
σ_a^2	0.0065	0.000	15.066	0.000	0.006	0.007

Table 3: Summary of ARIMA(1,1,2) model fitting results

Under the column Coefficient, we find both ϕ_1 and θ_2 are positively related to logarithm of total number of reported results while the θ_1 affects the target variable in the opposite direction. Typically, ϕ_i's are used to model the autocorrelation of the time series. Since ϕ_1 is positive, it indicates that the current observation is positively correlated with the past observations at lag 1. ϕ_i's are used to model the moving average of the time series. θ_1 is positive in our case, it indicates that the current observation is positively influenced by the past forecast errors at lag 2. By contrast, the current observation is negatively influenced by the past forecast errors at lag 1.

Regarding the column $P > |z|$ of Table 3, it shows the p-values of all the parameters in this time series model. All of them are less than 0.001, which indicates that they are significant and hence the model is robust. The fitted model is combined with the original Y_t in Figure 9.

4.1.3 Model Diagnostics with Shapiro-Wilk Test and Ljung-Box Test

It is not uncommon to conduct residual analysis to diagnose whether a model performs well. If the model is correctly specified and the parameter estimates are reasonably close to the true values, then

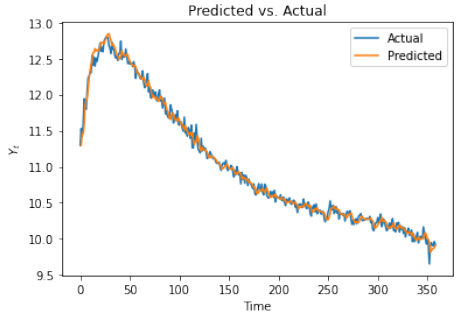

Figure 9: ARIMA(1,1,2) fitted values of Y_t

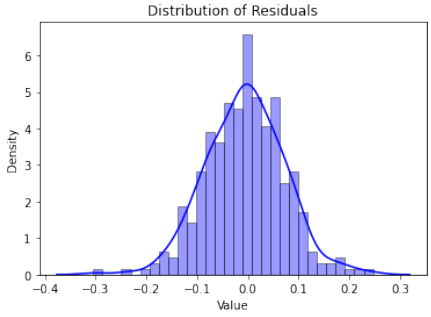

Figure 10: ARIMA(1,1,2) residuals distribution

the residuals should have nearly the properties of white noise, which is assumed in Equation 2, 3. Furthermore, a_t's should behave roughly like independent and identically normal distributed with mean zero and common variances.

Figure 10 shows the distribution of the fitted residuals which seems to follow normal distribution. We apply the Shapiro-Wilk test for normality check where the null hypothesis H_0 states that the tested data indeed is normally distributed [5]. The statistic is 0.9953 and the respective p-value is 0.3762. Hence, since the p-value is much greater than 0.05, we accept H_0 and conclude that the residuals follow a normal distribution.

Ljung-Box test is another statistical test proposed for a fitted ARMA model [3]. It is used to test the ACF of residuals to check if there is lack of fit of a time series model. Equivalently, it tests the null hypothesis H_0 that the error terms have no correlation and thus we are not necessary to build more complex ARIMA models due to the principle of parsimony.

$$Q_* = n(n+2)(\frac{\hat{r}_1^2}{n-1} + \frac{\hat{r}_2^2}{n-2} + \cdots + \frac{\hat{r}_K^2}{n-K}) \overset{H_0}{\sim} \chi^2_{K-m} \tag{9}$$

where $\hat{r}_k$ is the sample estimation of r_k, n is the number of observations, K is the number of lags that should be predetermined, and $m = p + q = 3$ in our case.

In the Ljung-Box test, a smaller value of Q_* is favored by H_0 and it actually correlates with the number of lags we would like to test. In practical, we set K to be in the range of 1 to 10. This means that we would like to compute Q_* of ACF values after the first 10 lags at most. In Table 4, p-values of all the 10 tests are greater than 0.05. Thus, from lag 1 to lag 10, we all accept H_0 at significance level of 0.05 and conclude that ARIMA(1,1,2) model does not exhibit significant lack of fit.

4.1.4 Model Prediction

In previous subsections, we have built the ARIMA(1,1,2) model and use some statistical tests to test its fitness. To fulfill the requirements in this question, we are required to predict the number of reported results on one particular day, March 1, 2023. As the latest data provided is on the last day of 2022, the target is to predict the result after 60 days. This actually introduces much more uncertainty into the confidence of our prediction.

Table 5 summarizes changes in predicted log reported results with their prediction intervals. We observe that the model predicts quite well when our prediction date is not far from the forecast origin. In other words, it is quite confident for us to predict the reported scores on the first few days in 2023.

Lag	1	2	3	4	5
LB Statistic	1.0300	1.3114	3.0717	3.6573	3.6894
p-value	0.3102	0.5191	0.3807	0.4544	0.5949
Lag	6	7	8	9	10
LB Statistic	3.7338	4.6048	4.7106	6.6234	10.307
p-value	0.7127	0.7081	0.7880	0.6763	0.4140

Table 4: Ljung-Box test results

k	$\hat{Y}_t(k)$	Standard Error	95% Lower CI	95% Upper CI
1	9.9072	0.0803	9.7497	10.0646
5	9.8939	0.1149	9.6686	10.1191
10	9.8778	0.1745	9.5358	10.2198
30	9.8200	0.5035	8.8332	10.8067
60	9.7493	1.1228	7.5487	11.9499

Table 5: Summary of ARIMA(1,1,2) model fitting results

When k grows to 60, it is reasonable for $\hat{Y}_t(k)$ to be slightly less than 9.9223 that is the value on the last day of 2022 and our model forecasts it as 9.7493. However, regarding its confidence interval, we are not as certain as before. A 95% confidence interval ranges from 7.5487 to 11.9499. More prediction details can be found in Figure 11.

The last step remained for the forecast is to perform inverse log transformation back to calculate the true number of recorded results. By Equation 10, the expected mean of $\hat{X}_t(60)$ is 17198. Nevertheless, the corresponding confidence interval becomes unstable after this transformation.

$$E[X_t(60)] = e^{\hat{Y}_t(60)+\frac{\sigma_a^2}{2}} \tag{10}$$

4.2 Correlation Analysis on Word Attributes and Hard Mode Ratio

To determine if words attributes may affect the report percentage in hard mode, we conduct a correlation analysis in the first place. For features extracted in **3.1.2**, we normalize some of them by dividing the maximum frequency or total number, so that patterns and associativity are more likely to be observed. Attributes that we will conduct correlation analysis on are as follows:

- $charNum$: The number of distinct letters in the word.
- $charFreq$: The frequency of a letter normalized by dividing the total letter numbers.
- $charPos$: The frequency of a letter at a certain position normalized dividing the total number of words.

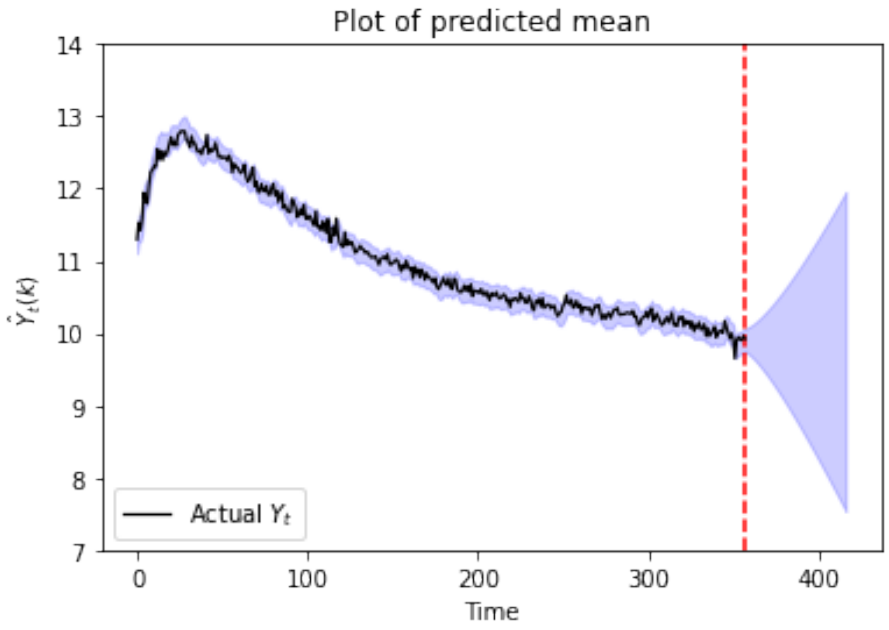

Figure 11: Plot of predicted mean

- $charCombo$: The frequency of letter combinations.
- $slbNum$: The number of syllables in the word.
- $wordFreq$: The frequency of a word normalized by dividing the maximum frequency, and feed the result to a scaled sigmoid function:

$$sigmoid(F_w) = \frac{1}{e^{-3(F_w - 0.5)}} \tag{11}$$

Table 6 contains the correlation coefficients between hard mode ratio and the extracted word attributes. We may observe that all coefficients are small in magnitude. Among all attributes, only the number of syllables appears weak positive correlation with the percentage score of results in the hard mode, while all other attributes show weak negative correlation with the hard mode ratio. It is notable that correlation and causal relationship are distinct concepts, where correlation does not necessarily imply a causal relationship, but a causal relationship implies a correlation. In our case, the correlation between word attributes and hard mode ratio is weak, which tells it is unlikely that the studied word attributes may affect the percentage of scores reported that were played in the hard mode.

Lag	$charNum$	$charFreq$	$charPos$	$charCombo$	$slbNum$	$wordFreq$
ρ	-0.079227	-0.013932	-0.064322	-0.038629	0.117395	-0.100611

Table 6: Correlation coefficient between word attributes and hard mode ratio

5 Task 2: Score Distribution Modeling with Partial Multi-Output SVR

In this section, we use MOSVR to predict the distribution of scores. In addition, in order to cope with the bad fit in regression of 4 tries and 7 tries, we adopted the idea of partial regression. Finally, we use the model to produce a prediction for the word "EERIE".

5.1 Model Overview

In order to better leverage the information given by the data, the insights gained from Section 3 are of great significance. Based on the extracted word features, we would like to design a model taking them as inputs and directly outputting the final distribution of the reported results of a given word on any date. We do not regard any specific day as an influential factor because there does not exist any trend in distribution of reported scores during this time span. As assumed in Section 2.1, we simply take the overall sample mean as the prediction of "1 try" for any given word. For the rest of scores, we may adopt partial regression with multi-output SVR for prediction.

5.2 Partial Regression

Because of the correlations among different tries, it is not necessary for us to fit the other six variables from "2 tries" to "$\geq$7 tries" at the same time. Since all these variables and "1 try" should sum to 100 after normalization, we can therefore model the predictions from "2 tries" to "6 tries" and get "$\geq$7 tries" by subtracting the others. Furthermore, from the distribution of "4 tries", it is observed that there does not exist any upward or downward trend, which makes the SVR model fail to predict "4 tries" solely on the word characteristics in a precise level.

To tackle with the low prediction accuracy of "4 tries", we take the four outputs from the multi-output SVR model as inputs to predict "4 tries" based on GAM. In other words, we divide the distribution of "4 tries" into three disjoint adjacent intervals and fit piecewise linear regression to each part of it. To be more precise, based on different ranges of the sum of "2 tries" and "3 tries", we split the intervals into three parts with breakpoints of 20 and 30 respectively.

In our analysis, we find that although "4 tries" is not strongly linearly correlated with the other recorded tries, it indeed relates to other tries in a non-linear format. Concretely, as shown in Figure 12, when the second and third tries are large and the fifth and sixth tries are small (the rightmost part of Figure 12), the value of "4 tries" tends to be small. The similar scenario occurs exactly when the opposite condition is satisfied (the leftmost part of Figure 12). However, for the rest of the time, the proportion of "4 tries" tends to level off but fluctuates all the time. After the prediction of "4 tries", we can directly calculate "$\geq$7 tries" since all of the seven proportions should sum to 100.

To sum up, as shown in Figure 13, we only use the model to predict the distributions of "2 tries", "3 tries", "5 tries" and "6 tries" given the constructed word features. Regarding "1 try", its sample mean is adopted. While for "4 tries", it is predicted by taking For the last "$\geq$7 tries", given the other six variables have been decided, it is simply calculated by subtracting the sum of the other predictions from 100.

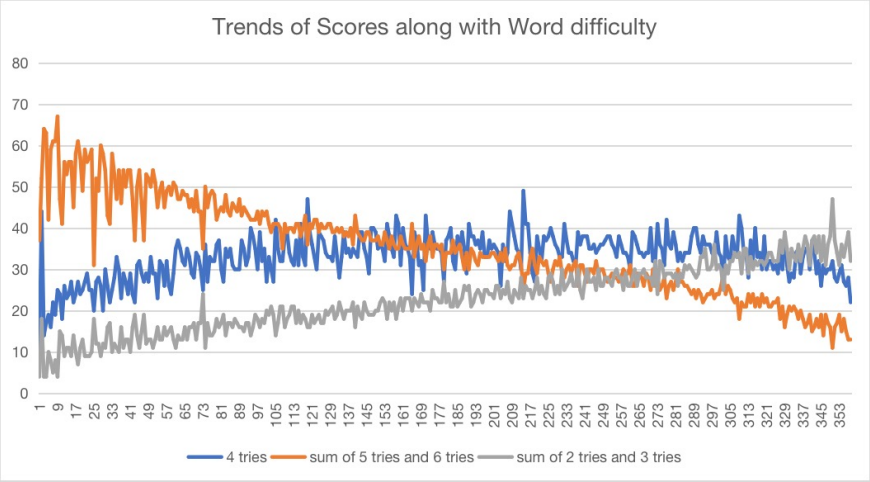

Figure 12: Trends of scores along with word difficulty

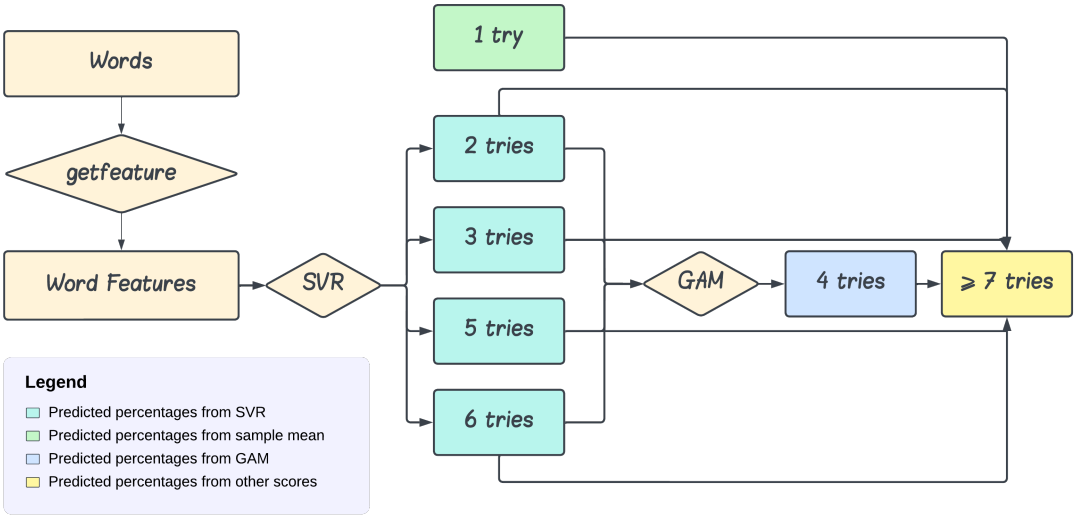

Figure 13: Task 2 workflow chart

5.3 Multi-Output Support Vector Regression

In regression tasks, a linear regression always exhibits lack of fit or high bias because of its linearity in features. Typically in our task, hand-crafted word features are limited and no other features have been provided, which requires us to introduce some non-linearity in our model in order to get better prediction results.

SVR borrows similar ideas from support vector machines that are used to classify targets into different classes. The fundamentals behind SVR is to find the best fit hyperplane that has the maximum number of points. With the aid of kernel functions like Radial Basis Function [12] (RBF), the fitted regression plane are no longer solely linearly dependent with the features.

$$\mathcal{K}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = e^{-\gamma||(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)})||^2} \tag{12}$$

where γ is a free parameter.

Conventionally, SVR is used to determine the mapping between the input **x** and a single output y_i from a training set S_i by finding a regressor $\mathbf{w}_i \in \mathbb{R}^{m\times 1}$ and bias b to minimize the loss function. In accordance to the case of score distribution, where the correlations between outputs may be taken into account, we propose the multi-output SVR method [4]. The multi-output SVR method seeks to minimize the following loss function 13 [1]:

$$\frac{1}{2}\sum_{i=1}^{4}||\mathbf{w}_i||^2 + C\sum_{l=1}^{N} L(\mathbf{y}^{(l)} - (\mathcal{K}(\mathbf{x}^{(l)})^T\mathbf{W} + \mathbf{b})) \tag{13}$$

where the $m \times d$ matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d)$ and $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_d)^T$, m is the dimension of each input sample.

Multi-output SVR model is able to predict the required four outputs and also encode the interaction effects among outputs. This serves the exact expectation in our task.

5.4 Prediction for Score Distribution

In this question, we are required to give the proportion of reported results of the word "EERIE" on March 1, 2023. As assumed in Section 5.1, this particular date does not render much information to predict the respective distributions in our model. Therefore, we will mainly focus on the "EERIE" word itself to give our proportion distribution.

For any given word, we follow the same procedure defined in Figure 13. Firstly, "EERIE" is passed into the *getfeature* function to extract features of this word. A multi-output SVR model is followed up to predict the distributions of "2 tries", "3 tries", "5 tries" and "6 tries" simultaneously. Then, we use linear regression model to further get the prediction of "4 tries". The rest can thus be determined. Overall, the predicted results are shown in Table 7.

1 try	2 tries	3 tries	4 tries	5 tries	6 tries	≥ 7 tries
0.4721	6.3539	19.5208	31.3662	28.2041	13.0922	0.9909

Table 7: Predicted distribution of reported results of "EERIE"

5.5 Evaluation of Confidence

To evaluate how confident we are in our prediction, we have created diagrams which illustrate how close the value of prediction is to the true value. In Figure 14, samples are sorted in the ascending manner by the value of average score (the blue curve). Black lines represent model's prediction while orange lines depict the true values. It can be seen that the black curve roughly follows the orange curve, which indicate the effectiveness of the prediction.

Besides that, correlation analysis was also conducted. From table 8, we can see that all predicted scores are of mildly strong or strong positive correlation with the true values, which means the model is effective in predicting the distribution of player scores.

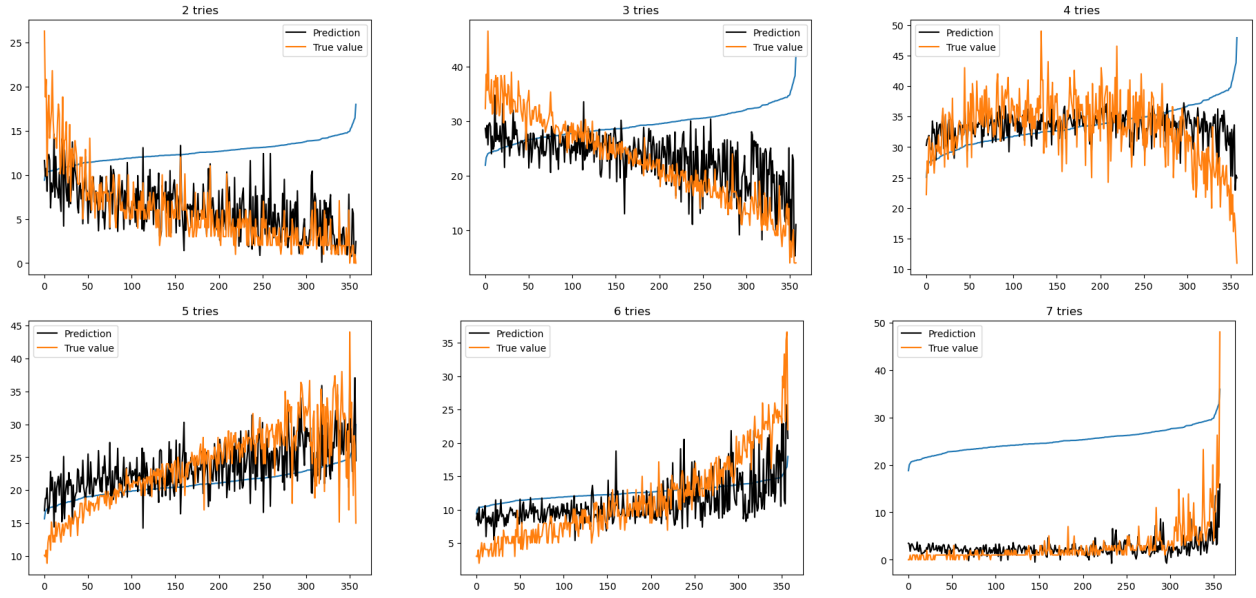

Figure 14: Predicted v.s. True Score Percentage

Coefficient	2 tries	3 tries	4 tries	5 tries	6 tries	≥ 7 tries
ρ	0.6919644	0.6393237	0.4388274	0.6295990	0.5609590	0.5172031

Table 8: Correlation Coefficients for the Score Distribution

6 Task 3: Difficulty Classification with Generalized Additive Models

With the establishment of MOSVR model, we follow a simple strategy of thresholding to decide whether a given word is classified as "easy", "medium" or "hard". This task strongly depends on the previous progress. Also, the predicted class of the word "EERIE" will be elaborated later.

6.1 Model Overview

In section 5, we have developed multi-output SVR to fit the proportion distributions of "2 tries", "3 tries", "5 tries" and "6 tries". Then we use these four distributions to determine the difficulty level of a word. By subtracting the sum of "5 tries" and "6 tries" from the sum of 2 tries" and "3 tries", we compare the contrast with the threshold. If the contrast is greater than the larger threshold, then it is classified as "easy". Conversely, if the contrast is smaller than the threshold, then it is followed by another threshold test which further classifies the word as "medium" or "hard". In our model, threshold 1 is 3.3765 and threshold 2 is -18.0690. The details are shown in Figure 15.

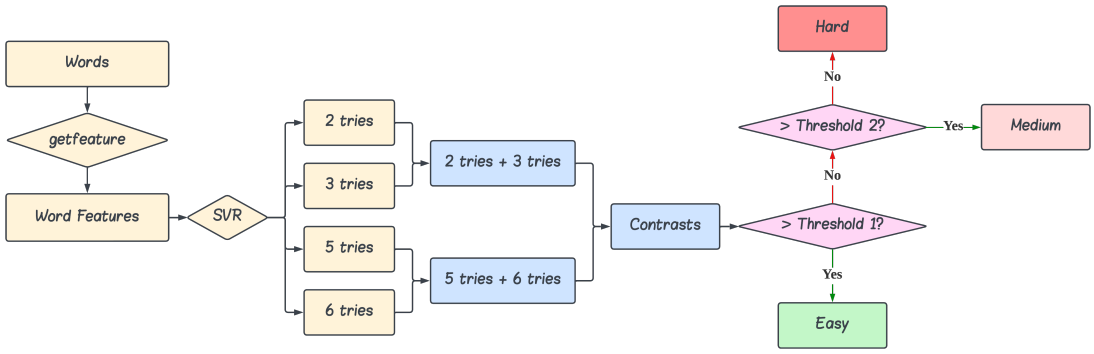

Figure 15: Task 3 workflow chart

6.2 Model Interpretation

The difficulty level of a word can be represented by the average score. By observing Figure 16, ”2 tries” and ”3 tries” decrease as the average scores increase, and ”5 tries” and ”6 tries” increase at the same time. Thus, the contrast of the sum of ”2 tries” and ”3 tries” and that of ”5 tries” and ”6 tries” can be regarded as an indicator of the difficulty level of a word.

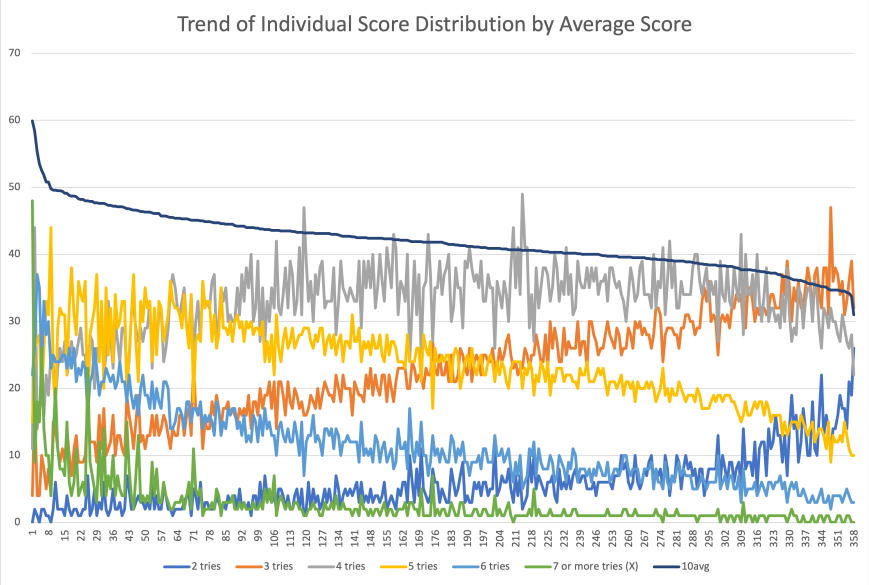

Figure 16: Trend of Individual Score Distributions by Average Score

In our classification task, we would like to divide the sample into three difficulty levels on a comparative basis. To determine the threshold, we first find out the 1/3 quantile of the average scores of all samples. Then we select the words whose average scores do not differ from their 1/3 quantile by more than 0.01. Afterwards, we calculate the average contrast of these words, which is our first threshold. Similarly, we decide the second threshold based on 2/3 quantile and the corresponding average constrast.

6.3 Relationship Between Word Features and Classes

After classifying the words into three difficulty level, we the calculate the average value of different features for each class separately. By comparing the average values, we find out 4 attributes which is tightly correlated with difficulty level: *wordFreq*, *charFreq*, *charCombo* and *slbNum*. The average value of these features are listed in Table 9.

	Easy	Medium	Hard
$wordFreq$	0.2062	0.1897	0.1849
$charFreq$	0.2951	0.2831	0.2668
$charCombo$	0.02168	0.02002	0.01766
$slbNum$	1.3666	1.4677	1.5789

Table 9: Average value of features among different classes

Regarding the *wordFreq* attribute, as we can observe from the table, the greater the *wordFreq*, the easier the word is. A potential explanation can be that people are more impressed with high frequency words, so players are more likely to guess these words in Wordle game.

For *charFreq*, it follows the similar pattern as *wordFreq*. Intuitively, when a player guesses a wrong word, there is a higher probability that the player uses a letter contained in the answer, if the answer has a greater value of *charFreq*.

Turning to the *charCombo* characteristic, if a word contains more frequent letter combinations, it is much easier to be guessed. Again, this is in line with our common sense.

However, for the *slbNum* feature, we find that when there are fewer vowels in a word, it will be easier for the player to complete the game in fewer tries. This is a little abnormal since the appearance frequency of a vowel is higher than a consonant and thus if there are more vowels in a word, players should be more likely to guess it.

6.4 Prediction Results

Given the word "EERIE", we take it as input to the SVR model and calculate the contrast as -15.4217 which lies between the two thresholds. As it is closer to the threshold 2, it is classified as "medium" class, but is very close to the "hard" class. Regarding the accuracy of our classification model, we classified all the words in the sample, finding that 65.8% of them are correctly classified.

7 Task 4: Identification of Interesting Features

Throughout our process of model establishment, we get to gain deeper insights into the data. A few interesting features were spotted and identified. It is worthwhile to discuss some of them, which may enlighten us and assist in model optimization in turn.

7.1 Curse of Four Tries

During the course of we solving the problem, it is always assumed the distribution of player scores generally reflects the difficulty of a solution word, and the percentages of neighboring times of tries should be closer than those with more attempts away. However, while analyzing the dataset, we found that the column "4 tries" appeared patterns that are inconsistent with the other score distribution columns. Since it

7.1.1 Weaker Correlation with Neighboring Columns

It usually agrees with our intuition that the percentage of neighboring times of attempts should be close in value or even correlated to some extent. With this mindset, we attempted to confirm the assumption by conducting correlation analysis among all neighboring columns that represent the distribution of scores. It is notable that the column of one try is dismissed because we generally consider that those who guess the word correctly in the first time were purely lucky given that no prior feedback had been given. It can be seen from Table 10 that 2 and 3 tries, 5 and 6 tries, 6 and 7 (or more) tries all demonstrate strong positive correlation, which indicates that the increase of one will likely result in the increase of the other. However, 3 and 4 tries are mildly positively correlated with coefficient 0.33289725, and 4 and 5 tries are barely correlated.

Coefficient	$2\ and\ 3\ tries$	$3\ and\ 4\ tries$	$4\ and\ 5\ tries$	$5\ and\ 6\ tries$	$6\ and\ 7\ tries$
ρ	0.75527864	0.33289725	-0.04767388	0.69393922	0.65450803

Table 10: Correlation coefficient between neighboring number of tries

7.1.2 Bad Fit in Linear Regression

Apart from correlation analysis, inconsistency of "four tries" was also observed in linear regression during the process of model construction of Task 2. To predict different words' distribution of player scores, all features mentioned in 4.2 are fed into the model as inputs, while player scores (except for one try) serve as dependent variables. After predicting the number of tries with linear regression, we conduct correlation analysis for all six outputs. From table 11, we can see that the predicted "4 tries" is barely correlated with the true value, while other scores all appear stronger correlation with corresponding true values.

Coefficient	$2\ tries$	$3\ tries$	$4\ tries$	$5\ tries$	$6\ tries$	$7\ tries$
ρ	0.4979768	0.4914183	0.0872949	0.4927578	0.3801517	0.2198602

Table 11: Correlation coefficient between prediction and true values

7.1.3 Non-Monotonic Trend by Word Difficulty

As assumed in 2.1, we use the expected number of attempts to measure the level of difficulty of a word. Figure 16 depicts the trends of individual distribution of different scores by the average score of words. In the diagram, average scores have been sorted in the descending order beforehand. We may observe that the percentage of "4 tries", which shows intense fluctuation in the middle, and look close in value on the two ends. Meanwhile, all other trends of scores appear roughly monotonic along with the decrease of average scores.

7.2 Fewer tries Used in the Hard Mode:Anomaly Observed and Possible Explanations

As the level of difficulty is increased in the hard mode, it is expected that the distribution of player scores should be larger in value than that in the regular mode. Figure 17 illustrates that the percentage of reported results is generally increasing by time with some fluctuations over time. However, in Figure 18 we can see that the average player score keeps oscillating with no obvious trend, which goes against normal sense. If we combine the information implied in the two Figures, a possible explanation is that As time goes by, the game is losing a lot of players, and whose who keep playing the game should gradually have a better command of the game. That portion of loyal players may be more open to the hard mode and have better performance in the hard mode.

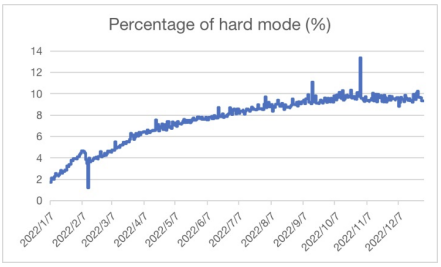

Figure 17: Trend of hard mode ratio by date

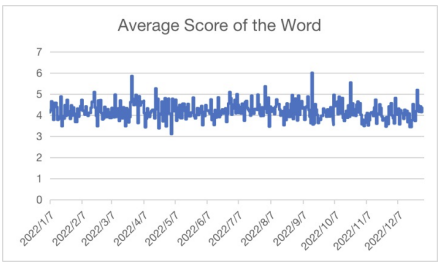

Figure 18: Trend of average score by date

8 Strengths and Weaknesses of the Model:

8.1 Sensitivity Analysis

In order to test the robustness of joint model, we first get the features of word "EERIE". Then, we vary the features values by $\pm 10\%$ in sequence while keeping the other values the same. We get the prediction values from our model and then calculate the average score. The results shown in Table 12 do not change significantly with only a maximum change of 1.9%. Hence, it can be observed that the stability of the parameters is robust.

8.2 Strengths

- In our project, we have constructed many characteristics of the given words, including features of each word and among different words, including but not limited to those introduced in Section 4.2.

	$charFreq$	$charPos$	$charCombo$	$slbNum$	$wordFreq$
Original	4.237	4.237	4.237	4.237	4.237
Decrease by 0.1	4.317	4.193	4.253	4.242	4.248
Increase by 0.1	4.157	4.281	4.221	4.232	4.227

Table 12: Results of sensitivity analysis

They act as critical factors in our model prediction.

- We adopted a partial multi-output regression model, which takes into account the correlation between outputs. Meantime, partial regression avoids the limitation of SVR in predicting the percentage of four tries, thus more reliable.
- Our predictive models are tested as robust after the sensitivity analysis. Thus, the model parameters are not affected by the subtle change of inputs.

8.3 Weaknesses

- The assumption on One Try may be speculative. It is assumed beforehand that with no prior knowledge given, we consider cases where the solution is guessed correctly in the first time as random events. However, we observed that some words such as ”train” and ”slate” may have slightly higher percentage. Therefore, further study is needed for verification.
- Due to the time limit and lack of other related information, not all extracted features were included in our prediction model to achieve better performance. Especially, regarding word features, there may be more informative ones such as homophones and part of speech that we have not experimented.
- It is reasonable that players of the game may have a better command at it over time, which means the expected score could go down gradually. However, when studying the difficulty of solution words, we take the expectation of scores as the indicator of difficulty. Further study may take this factor into account.
- There are multiple factors beside word attributes that can affect the score of each player. This intrinsic nature may set a top limit for the performance of our model.

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

To: The Puzzle Editor of the New York Times
From: Model Development Team
Subject: Predictive modeling on score distribution and difficulty classification
Date: February 21, 2023

Dear Sir or Madam:

Thank you for inviting us to do an analysis on the data file containing the game stats of Wordle posted online. Based on the data file and the features we extracted, we evaluated the relationship between word attributes and the percentage of scores reported that were played in the hard mode. Furthermore, we have established mathematical models to predict the number of reports on a future date, distribution of solution words and the difficulty of words. Our approaches, findings, and suggestions are as follows. To start with, there are various word features extracted from the data file, which are of great significance for training our model. For your reference, below is a list of features that we have trained in our model:

1. The number of distinct letters in the word.
2. The normalized frequency of a letter.
3. The normalized frequency of a letter at a certain position.
4. The frequency of letter combinations.
5. The number of syllables in the word.
6. The normalized frequency of a word.

With the identified features, here are some findings based on our analysis:

- In the past days, Wordle experienced a stage where the number of players booms until February 2^{nd}, and started to decline in player number from then on.
- Although the number of reported scores has been generally shrinking, the prediction interval of reported number given by ARIMA allows for intense fluctuation.
- The recognized word patterns are effective in determining the difficulty level of solution words.
- Players' performance is sensitive to the word attributes. Specifically, when word attributes correspond to a higher difficulty level, the average number of tries will rise.
- The portion of players who play in the hard mode is increasing over time.
- Wordle players are getting more experienced in the game over time.

According to our discoveries, we propose a few recommendations, which may help increase the number of Wordle players.

- Invest in marketing and advertising. Since there are huge room for increase, these two strategies may bring exponential growth in the player number.
- Pay attention to the word attributes, and be aware of the difficulty level of solution words. Try to separate extremely tough words, as hard word can significantly affect the score distribution of players, thus frustrating players.
- Different word sequence can be added in two modes, and words can be categorized by their difficulty. Words with higher difficulty levels can be put into hard mode since those who are open to hard mode are likely to be loyal to the game.

The suggestions and strategies above may give a better understanding in the player preferences, and can hopefully increase the future game players of Wordle.

Yous sincerely,
Model Development Team

References

[1] Borchani, H., Varando, G., Bielza, C., & Larrañaga, P.. (2015). A survey on multi-output regression. Wires Data Mining and Knowledge Discovery, 5(5), 216–233. https://doi.org/10.1002/widm.1157

[2] Cavanaugh, J. E., & Neath, A. A. (2019). The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. WIREs Computational Statistics, 11(3), e1460. doi:10.1002/wics.1460

[3] Lawrance, A. J., & Lewis, P. A. W. (1985). Modelling and Residual Analysis of Nonlinear Autoregressive Time Series in Exponential Variables. Journal of the Royal Statistical Society. Series B (Methodological), 47(2), 165–202. http://www.jstor.org/stable/2345560

[4] Sanchez-Fernandez, M., de-Prado-Cumplido, M., Arenas-Garcia, J., & Perez-Cruz, F. (2004). SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems. IEEE Transactions on Signal Processing, 52(8), 2298–2307. doi:10.1109/TSP.2004.831028

[5] Shapiro, S. S., & Wilk, M. B. (1965). An Analysis of Variance Test for Normality (Complete Samples). Biometrika, 52(3/4), 591–611. https://doi.org/10.2307/2333709

[6] Siami-Namini, S., & Namin, A. S. (2018). Forecasting Economics and Financial Time Series: ARIMA vs. LSTM. doi:10.48550/ARXIV.1803.06386

[7] https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

10 Appendices

10.1 Appendices 1: Codes for Key Features

```
wordfreq = np.asarray(
    [word_frequency(words[i], "en") for i in range(len(words))], dtype=np.float64
)
maxfreq = max(wordfreq)
wordfreq = wordfreq / maxfreq
wordfreq = 1 / (1 + np.exp(-(wordfreq - 0.5) * 3))
```

Figure 19: *wordfreq*

```
charpos = np.zeros((26, 6))
for i in range(len(words)):
    for j in range(5):
        charpos[ord(words[i][j]) - CZERO][j] += wordfreq[i]
        charpos[ord(words[i][j]) - CZERO][5] += wordfreq[i]
```

Figure 20: *charpos*