Decoding Wordle: Investigating the Relationship Between Game Attributes and Player Performance

Summary

February 20, 2023

Worldle is a puzzle game offered by the New York Times that has captivated word lovers and puzzle enthusiasts worldwide. The game is inspired by the classic word game Boggle, but with a unique twist that sets it apart. The basic objective of Worldle is to guess a five-letter word within a limited number of attempts, the player is presented with a blank five-letter word and six chances to guess the letters in the word. The player types in a five-letter word and the game highlights each letter in the word that matches the target word and shows the correct position of each matched letter. If the player guesses the correct letter in the correct position, the letter will be highlighted in green. If the letter is correct but in the wrong position, it will be highlighted in yellow. If the letter is not present in the target word, it will remain blank.

The game has become popular due to its simplicity and addictive nature. Players can try to guess the word as many times as they want and can continue playing until they correctly guess the target word or use up all six guesses. The game also allows players to challenge themselves by trying to guess the word using as few attempts as possible.

In Problem 1, we used data preprocessing, smoothing, and modeling techniques to make accurate forecasts based on a time series data-set. We employed the **Autoregressive Integrated Moving Average (ARIMA)** method and explored other methods,such as SARIMA method and Grayscale prediction, to ensure we were using the most appropriate approach for the specific problem.

In Problem 2, we aimed to predict the distribution of reported results for a given word by selecting relevant factors such as letter frequency, rarity of letters, pronunciation of letters, word structure, and repetition of letters. We used **Multicategorical Logistic regression** models and **Gradient Boosted Tree (GBDT) regression** model due to computational efficiency, interpretability, and the risk of overfitting with a small data-set.

In Problem 3, we established a model for classifying the difficulty of a certain word and predicting the difficulty of the word "EERIE." We used a **Multi-Layer Perceptron neural network (MLP)** to build a multi-layer perceptron neural network. The model effectively predicted the Average-Step of the test set and the difficulty level of words, but had a deviation in predicting the specific value for "EERIE." The evaluation revealed a mean squared error value of 31.84 and an accuracy of 1.0 in predicting the difficulty level.

In Problem 4, we mention some interesting findings in the process of modeling and organizing the data, and provide a more plausible explanation based on these findings, and more importantly, an outlook on these findings.

The last part is a letter to the New York Times.

Contents

1 Introduction

1.1 Problem Background

Worldle is a simple and addictive puzzle game offered by the **New York Times**, inspired by the classic word game Boggle. The objective is to guess a five-letter word within six attempts. Players can continue playing until they correctly guess the target word or use up all their attempts. Wordle has become popular among people of all ages and has spawned a number of online communities where players can share their strategies and compete against each other. It is a great way to test vocabulary, language skills, and creativity under pressure. The game has gained a devoted following since its introduction in 2008 and is known for improving vocabulary and language skills.

Worldle is a simple and addictive puzzle game offered by the New York Times, inspired by the classic word game Boggle. The objective is to guess a five-letter word within six attempts. Players can continue playing until they correctly guess the target word or use up all their attempts. Wordle has become popular among people of all ages and has spawned a number of online communities where players can share their strategies and compete against each other. It is a great way to test vocabulary, language skills, and creativity under pressure. The game has gained a devoted following since its introduction in 2008 and is known for improving vocabulary and language skills.

1.2 Clarifications and Restatements

1.3 Our Work

In the Problem1, we aimed to make forecast based on a given time series data-set using a combination of data preprocessing, smoothing, and modeling techniques. To prepare the data for analysis, we first performed an outlier detection step to identify and remove any abnormal values that could skew the results. We then applied the moving average (MA) method to smooth the data and reduce any noise in the signal.For modeling, we employed the autoregressive integrated moving average **(ARIMA) method**[3] to create a model that could capture the underlying trends and patterns in the data. Through this approach, we were able to generate accurate forecasts of future values, which could be useful for decision-making and planning.

While we primarily focused on the ARIMA model, we also explored other methods to ensure that we were using the most appropriate approach for our specific problem. Furthermore, we conducted a range of detailed examinations throughout our study to ensure that our results were valid and reliable. Together, our findings suggest that our combination of data preprocessing, smoothing, and modeling techniques can provide valuable insights for forecasting time series data.

The study of Problem 2 aimed to predict the distribution of reported results for a given word by selecting relevant factors such as letter frequency, rarity of letters, pronunciation of letters, word structure, and repetition of letters. Regression models were chosen over machine learning models due to computational efficiency, interpretability, and the risk of overfitting with a small data-set. The data was preprocessed by checking for abnormal points, summarizing and visualizing the data, encoding categorical variables using integer encoding, and splitting the data for model training. The proposed approach can help researchers and language experts gain insights into the characteristics and patterns of word usage in a given context.

The task of Problem 3 is to establish a model for classifying the difficulty of a certain word, and to identify which label attributes correspond to the classification of the word. In addition, the difficulty of the word "EERIE" needs to be predicted, and the accuracy of the model needs to be discussed.For this task, a **Multi-Layer Perceptron neural network (MLP)**[5]was used for prediction. The data was preprocessed and labels were extracted. An MLP was established with input and output nodes, as well as hidden layers and nodes based on the complexity of the data. The model was trained by initializing parameters, feeding data, adjusting weights and biases, and evaluating fit with a test set. The results showed that the model effectively predicts the AverageStep of the test set and the difficulty level of words. The evaluation revealed a mean squared error value of 31.84 and an accuracy of 1.0 in predicting the difficulty level. However, the model had a deviation in predicting the specific value for "EERIE". A 4-layer perceptron neural network was built using the TensorFlow library in Python.

2 Proposed Concept

To better understand and study these problems, we introduced certain concepts that may not be entirely novel or widely accepted, but proved to be highly helpful in our research

Position Entropy: Position entropy is a measure of the amount of uncertainty or randomness in the position of the letters in a word. It is calculated using the principles of Shannon Entropy, which takes into account both the number of possible positions for each letter and the probability of each position. For example, if a letter can appear in any of three positions with equal probability, the position entropy for that letter is log2(3) = 1.585 bits. If the same letter can appear in any of two positions with equal probability, the position entropy for that letter is log2(2) = 1 bit. In general, the higher the position entropy for a letter, the more difficult it is to recognize its position in the word.

Word Entropy: Word entropy is a measure of the amount of uncertainty or randomness in the letters that make up a word. It is calculated using the principles of Shannon Entropy, which takes into account both the number of possible letters for each position and the probability of each letter.

Formula for Word Entropy:

$$H_p = -\sum_{i=1}^{n} Pw_i H(i)$$

In this particular case, n is set to be 5

Nomenclature

H_{W1}	Sum of shannon entropy of the $AAAAA$ type Word
H_{W2}	Sum of shannon entropy of the $BAAAA$ type Word
H_{W3}	Sum of shannon entropy of the $BAAAB$ type Word
H_{W4}	Sum of shannon entropy of the $BABAB$ type Word
H_{W5}	Sum of shannon entropy of the $BBABB$ type Word
$Vowels$	Number of vowels in a word
p_i	Probability of the ith outcome
δ_i	Differencing coefficient
ϵ_t	Error term
ϵ_w	White noise error term
$\hat{Y_j}(i)$	Predicted value of dependent variable for jth observation when ith observation is excluded
$\hat{Y_j}$	Predicted value of dependent variable for jth observation
ϕ_i	Auto-regressive coefficient
θ_i	Moving average coefficient
AR	Auto-regressive Model
$ARIMA$	Auto-regressive Integrated Moving Average Model
c	Constant term
d	Number of differences taken
Di	Cook's Distance for observation i
h_i	Leverage of observation i
MA	Moving Average Model
MSE	Mean squared error of the model
p	Number of regression coefficients
p_r	Number of auto-regressive terms
q	Number of moving average terms
y_t	Time series data
H	Entropy

3 Problem1:Developing a model to predict daily variation in reported results

This problem involves developing a model to explain daily variation in the number of reported results and using it to create a prediction interval for the number of results on March 1, 2023. Additionally, the effect of word attributes on the percentage of scores reported in Hard Mode is to be explored.

3.1 Raw Data Observation

Due to the existence of incomplete and abnormal values in data-sets, it is necessary to apply data pretreatment techniques to ensure that the data is reliable and accurate. Pretreatment involves identifying and handling missing or incomplete data points, removing or correcting **Outliers** or abnormal values, and transforming the data to improve its quality. By applying appropriate pretreatment techniques, we can ensure that the data is suitable for analysis and that any insights or conclusions drawn from it are more accurate and reliable.

Based on the prompt which specifies that each word has 5 letters, the following words:

- *rprobe* → *probe*
- *clen* → *clean*
- *tash* → *trash*
- *favor* → *favor*(this word has an extra space).

Additionally, it was noted that one of the letters is ï. This word was also preprocessed and modified to use the letter i instead.

Then, we constructed a **time series visualization plot** of the raw data.

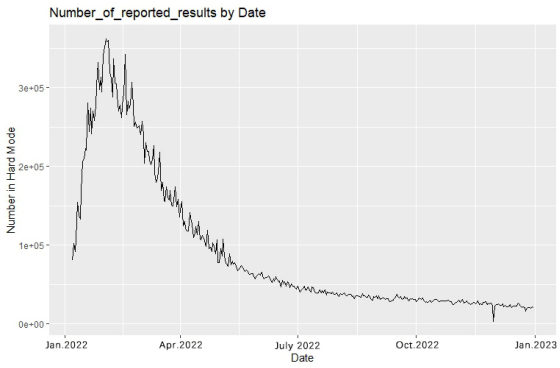

Figure 1: Standard Mode

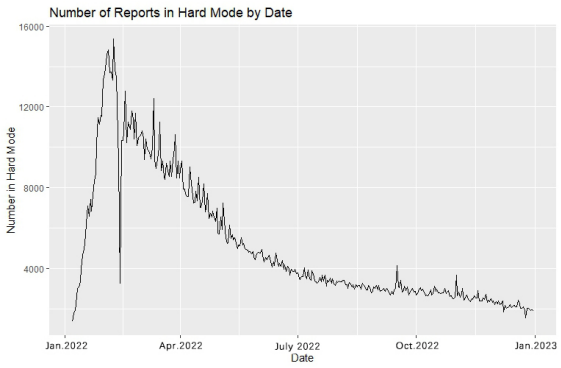

Figure 2: Hard Mode

The reported results show a trend of initially increasing and then decreasing, indicating that there was initially an increase in the number of participants followed by a subsequent loss, which gradually stabilized. Moreover, there appears to be a certain regularity visually, and it may be worth exploring regression and prediction models to develop a model for forecasting.We also checked for word duplication and found that all words are unique.

Visualization and Descriptive Statistics of Raw Data

The original data contained some anomalous values, such as the letter "i" in both Roman and English alphabets, and some words consisting of only four letters. These anomalous values were corrected. Following this correction, the statistical features of the data were computed. The mean, minimum, maximum, and various percentile values were calculated, as shown in the following table:

Data Pretreatment There are some anomalous values in the data, such as outliers, high leverage points, and influential points, that need to be analyzed. To address this, we used **Box-plot**, **Violin-plot**[1] and outlier tests to identify and analyze these anomalous values.

Statistic	Value
Min.	2569
1st Qu.	30309
Median	44578
Mean	90919
3rd Qu.	120294
Max.	361908

Table 1: Statistical features of the data.

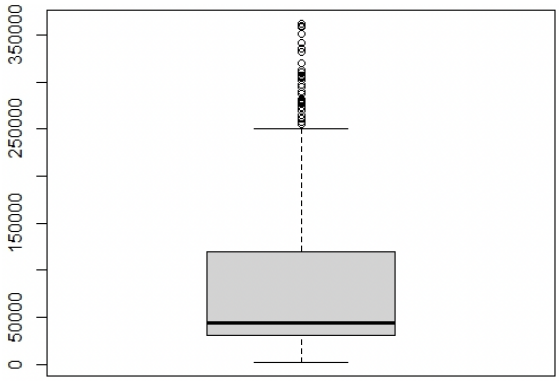

Figure 3: Box-plot

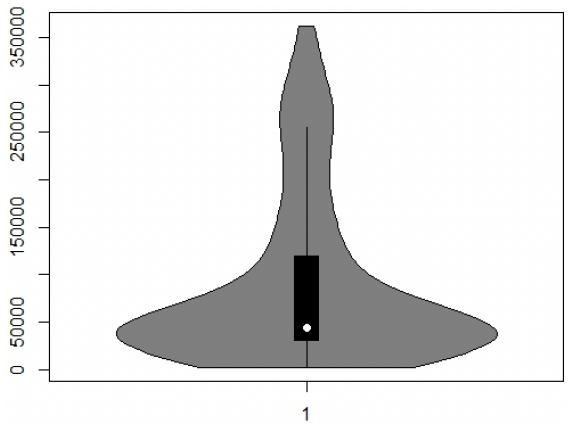

Figure 4: Violin-plot

Specifically, the **Hat Statistic**[4] was used to identify observations with high leverage.
This is the **Hat Matrix**, which is used in the hat statistic to identify observations with high leverage:

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \tag{1}$$

Cook's Distance[2] is a measure used to identify outliers in statistical analysis. It is calculated as the squared difference in predicted values (Y) for each observation, divided by the mean squared error (MSE) and adjusted by the leverage of that observation. The formula for Cook's Distance is:

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y_j} - \hat{Y_{j(i)}})^2}{p \times MSE} \times \frac{n-p}{h_i}$$

Where:

D_i is the Cook's Distance for observation i $\hat{\mathbf{Y}}_j$ is the predicted value of the dependent variable for the jth observation $\hat{\mathbf{Y}_{j(i)}}$ is the predicted value of the dependent variable for the jth observation when the ith observation is excluded n is the total number of observations p is the number of regression coefficients MSE is the mean squared error of the model h_i is the leverage of observation i If an observation's Cook's Distance is more than four times the mean distance, it can be considered an outlier. Therefore, Cook's Distance is a useful tool in identifying outliers in statistical analysis.

By correcting the time series order and analyzing the anomalous values, we were able to ensure that our subsequent analyses and modeling efforts were based on accurate and reliable data

3.2 Subsequent Data Refinement

After the data was preprocessed, empirical analysis was conducted to gain insights into the underlying patterns and relationships in the data. This analysis involved a variety of statistical and visualization techniques, such as regression analysis, time series analysis, and data visualization. In particular, we applied the **ARIMA (AutoRegressive Integrated Moving Average)** model[3] to our time series data to analyze and forecast future values. Basic formular are listed below

Auto-regressive (AR) model:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t \tag{2}$$

Moving average (MA) model:

$$y_t = c + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{3}$$

Auto-regressive integrated moving average (ARIMA) model:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{d} \delta_i \Delta y_{t-i} \tag{4}$$

By using this model, we were able to identify any significant trends or patterns in the time series data, and make accurate forecasts of future values. The results of this empirical analysis provided valuable insights that were used to guide further investigation and modeling

Transforming the Data into Time Series

The original data contains a time series that is reversed from the typical ordering. To correct this, the order of the time series needs to be adjusted.The advantage of translating data into time series form is that it enables the analysis of trends and patterns over time, allowing for more accurate forecasting and decision-making based on historical patterns and future projections.

By observing the time series plot, it was found that the data exhibits a clear trend, and in addition, there are some seemingly periodic fluctuations.Afterwards, we became interested in the periodic fluctuations and chose not to simply smooth them out or ignore them.

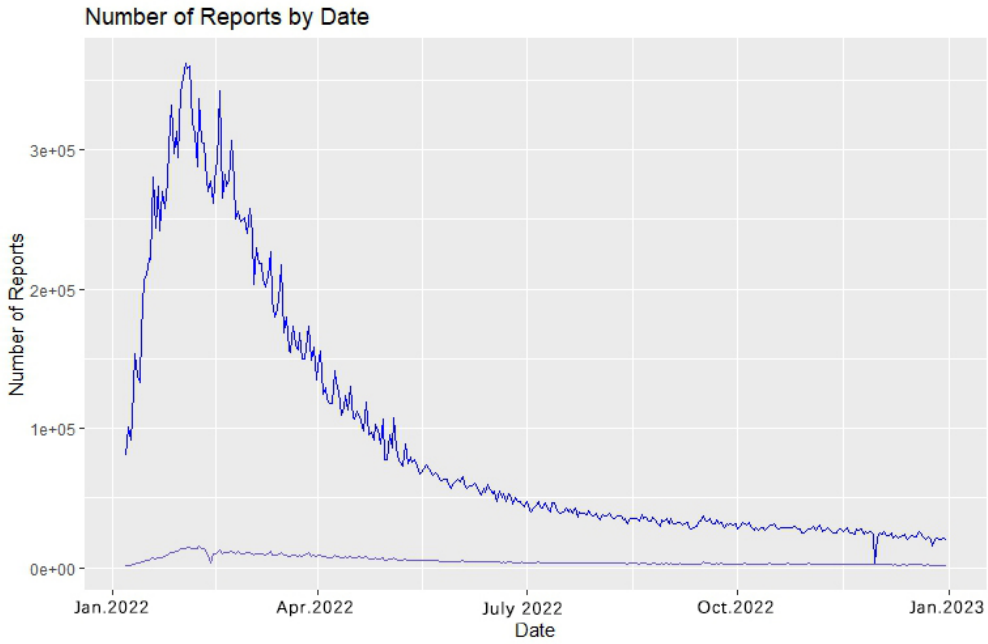

Figure 5: Combined Time Series Visualization Plot

Exploring the Patterns of Fluctuations

Before using the Moving Average method (**MA**)[3], we investigated whether there is any **periodicity** in the fluctuations. Here, we obtain the **correlation matrix**:

$$\begin{bmatrix} 1 & -0.0218474 \\ -0.0138477 & 1 \end{bmatrix}$$

Examining the matrix, we can see that there is **little correlation** between the day of the week and the reported results throughout the entire time period. More importantly, we utilized wavelet analysis to visualize the data. Wavelet analysis offers a multiresolution analysis with time-frequency localization, adaptive resolution, signal denoising, compression, and computational efficiency, making it a powerful tool for analyzing signals in various applications. Similarly, from the wavelet analysis chart,it can be said that they are not related.**Wavelet analysis** also yielded the same conclusion. Therefore, we exclude the possibility of periodicity and begin the smoothing process.

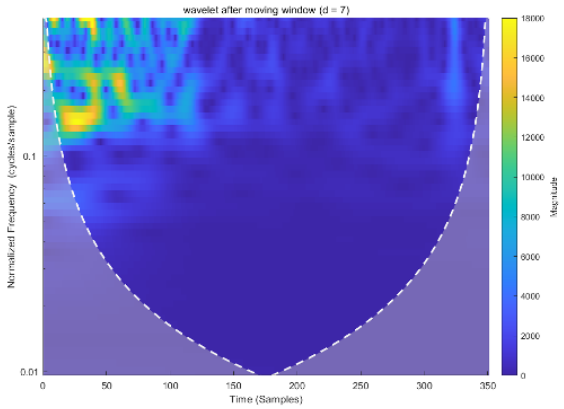

Figure 6: Wavelet Analysis

Smoothing the Data Using Moving Averages Method

Moving Averages Method is a time series analysis technique that helps in smoothing time series data. Firstly, we experimented with multiple window sizes and types of moving averages, including 3, 5, 7, 10, as well as simple, weighted, and exponential types.Below is a graph demonstrating the effectiveness of the smoothing method used in this analysis.

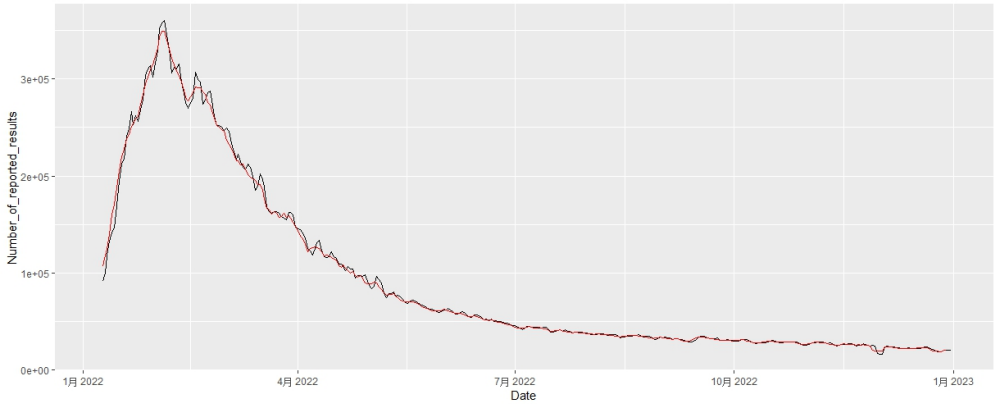

Figure 7: Time Series Visualization Plot after Smoothing Method

3.3 *Forecasting Using the ARIMA Model

To validate the stationarity of the sequence, we examined the time series plot and conducted an **Augmented Dickey-Fuller Test (ADF)**. The ADF test yielded a Dickey-Fuller statistic of -5.6275 with a lag order of 7 and a p-value of 0.01. Since the p-value is less than 0.05, we can consider the sequence as stationary.

Model Order Selection and Fitting

We can use the autocorrelation and partial autocorrelation plots to identify the values for the differencing parameter d and the moving average parameter q. By analyzing these plots, we can determine that the appropriate model for our data is an **MRIMA(0, 1, 1)** model.

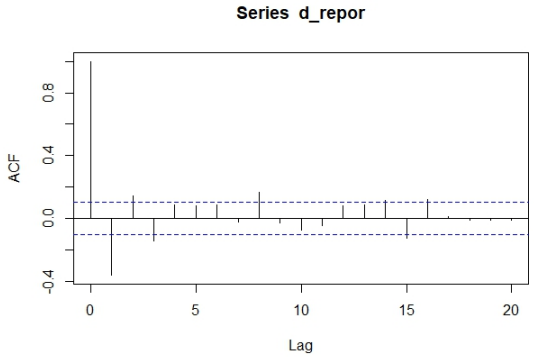

Figure 8: ACF-plot

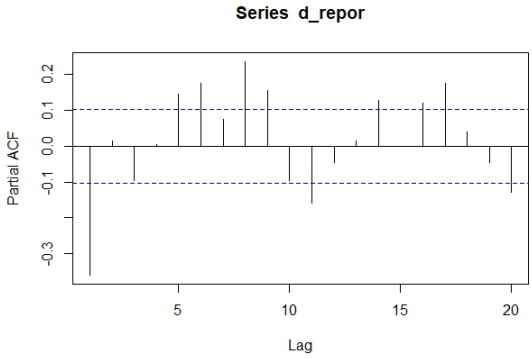

Figure 9: PACF-plot

Model Diagnosis

We examined the residual plots to check for the normality assumption of the errors. The QQ plot indicated poor normality in the tails, but normality in the middle. The Box-Ljung test was also conducted and produced a p-value of 0.8933 with X-squared = 0.017988 and 1 degree of freedom. Since the p-value is greater than 0.05, we accept the null hypothesis, indicating that the **residuals are stationary**.

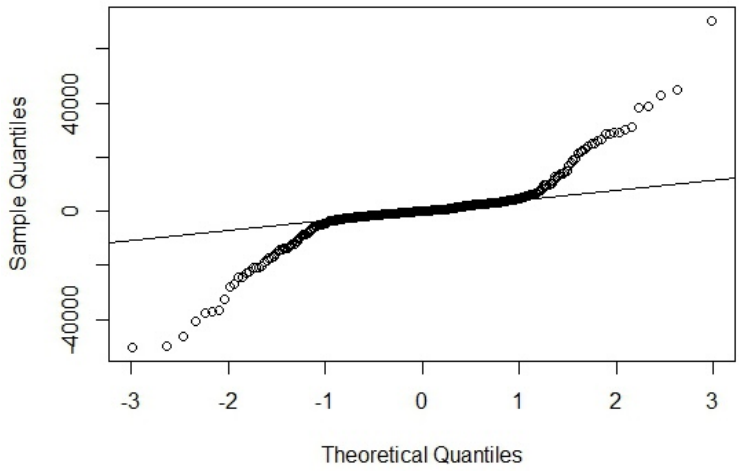

Figure 10: qqplot

Forecasting Using the ARIMA Model

The AIC and BIC values were 6954.784 and 6981.909, respectively. These values are used as reference indicators, with lower values indicating better model fit.Using this model, the following predictions were made for March 1, 2023:

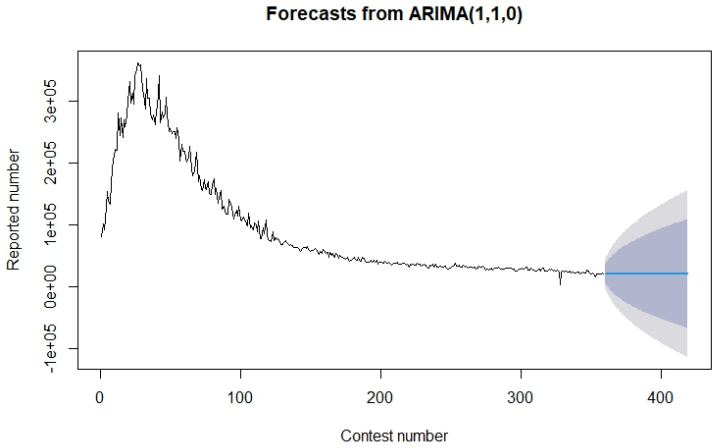

Figure 11: Reported Category

For the Reported category, the predicted value was **20599.07**, with **80**% prediction interval of **(-67715.295, 108913.43)** and **95**% prediction interval of **(-114466.095, 155664.23)**. It is important to note that the lower limit of the prediction intervals was set to 0, as users cannot have negative values.

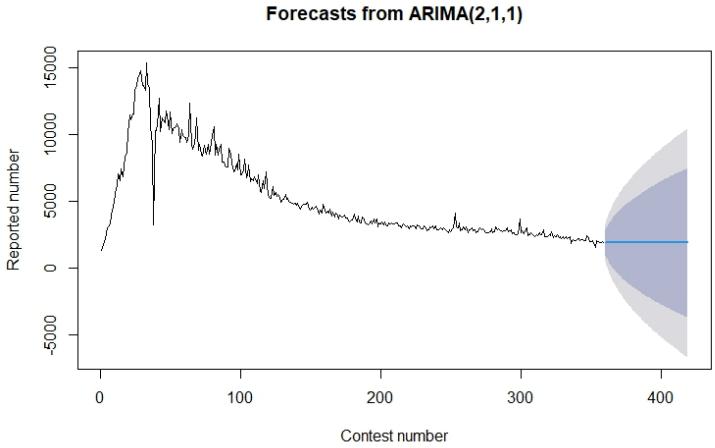

Figure 12: Hard Category

For the Hard category, the predicted value was **1903.920**, with **80**% prediction interval of **(-3684.24958, 7492.089)** and **95**% prediction interval of **(-6642.4476, 10450.287)**. Again, the lower limit of the prediction intervals was set to 0. In conclusion, the time series analysis conducted supported the development of an ARIMA(0,1,1) model, which was found to have stationary residuals. The predictions made using this model showed that the Reported category is expected to have a higher value compared to the Hard category on March 1, 2023. The prediction intervals provide a range of values within which the actual values are expected to fall with a certain level of confidence.

3.4 Attributes of Words in Difficult Mode

Data Visualization for Difficult Mode

As mentioned earlier, information has been obtained regarding the distribution of scores in difficult mode. In order to further analyze the potential properties of the distribution itself, the data for difficult mode was further visualized.

Due to the potential loss of information when converting the data for difficult mode into percentages, the original statistical data in the form of a box plot was used instead of the percentage data.

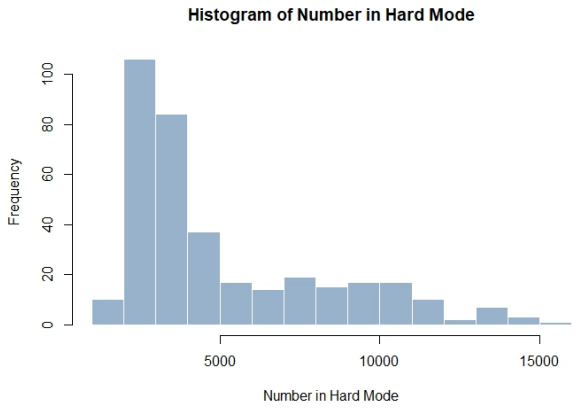

Figure 13: Calculating the percentage of scores for difficult mode

From the observation of the chart, it can be concluded that the frequency of letters in different positions has little effect on the "distribution of scores in difficult mode". Through ANOVA analysis, there was no significant correlation found for the first, third, fourth, and fifth positions. However, the occurrence of the second letter had a more noticeable effect on the percentage difference. It should not be ignored that when letters such as j, z, and x appear, the difficulty score percentage is higher than that of other letters, although the frequency of occurrence of these letters is relatively low.

Exploration of Attributes of Words

The attributes of words play a crucial role in determining the complexity of words. Factors such as word frequency, meaning complexity, and part of speech can affect the ease or difficulty with which a word is understood and processed by players in Wordle. We have summarized the following attributes:frequency, Shannon Entropy, complexity of meaning, word structure, and part of speech.

We used **ANOVA** method[7] to compare the means these groups that are defined by a single categorical variables mentioned above. The plots help identify whether there are significant differences in number in hard mode between different word.

Table 2: ANOVA Table

Source of Variation	Sum of Squares	df	F	PR(>F)
Tag2	0.001066	2.0	1.093902	0.336037
Residual	0.172541	354.0	–	–
Total	0.173607	356.0	–	–

In ANOVA analysis, the P-value is one of the indicators of data correlation. From the chart, it can be seen that the P-value is much greater than 0.05, indicating that the means of different categories are equal and there is no difference in parts of speech, as a result, The influence of word type may be temporarily excluded.

grouped box plot for visualization purposes.

Henceforth, we shall initiate an analysis concerning the frequency and positional distribution of letter occurrences, and analyze the frequency of letter occurrences for each distinct position. Visualize the mean values of letter occurrences across distinct positions. Obviously, based on the following two approaches, it has been determined that there are significant differences in the frequency of occurrence for the 26 letters. Given the significant differences in letter frequency, the frequency of occurrence of different letters in different positions (1–5) of numerous five-letter words will be analyzed individually and subjected to ANOVA analysis.

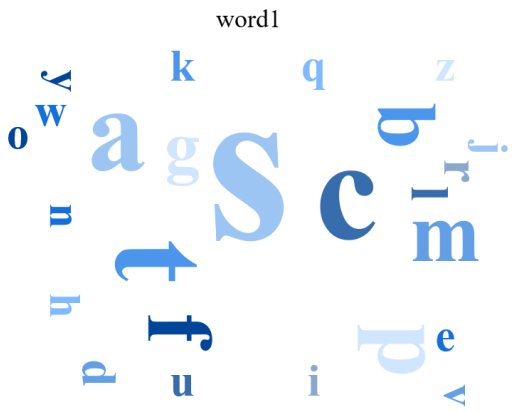

Figure 14: Letters Cloud

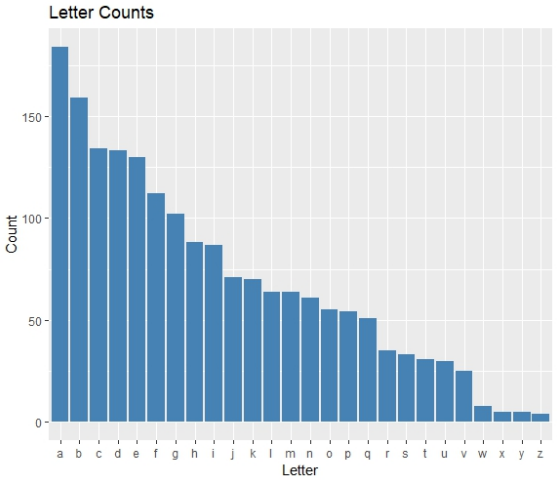

Figure 15: Total Letter Counts in Order

The main factor that affects the distribution of scores in difficult mode is word frequency. Words with lower average frequency are more likely to have an impact on the distribution of scores in difficult mode. This also indicates that people are less familiar with letters that have lower occurrence frequency, making them harder to guess in hard mode, such as x and z.

Moreover, in a five-letter word, the frequency of different words in the second position has a greater impact on the "distribution of scores in difficult mode". This can also be explained by the fact that the second letter is more important for the pronunciation or visual recognition of a five-letter word. Therefore, if the second letter is more obscure, it will be more challenging for players who are currently in "hard" mode.

3.5 Alternative Models that Have Been Experimented

We also tried other models in addition to the ARIMA mentioned earlier. These included The ARCH (Autoregressive Conditional Heteroscedasticity) model, the Grey forecasting model and the non-linear models with a higher degree of fit, etc. We also tried traditional machine learning algorithms even the data set is not fitted to train a complex model.

Method	Upper Limit	Prediction Value	Lower Limit	AIC Value
Gray Prediction	153155.14	25441.855	0	–
Sarima Prediction (Period=7)	105150.13	399.91	0	6941.102

All of these models were trained and tested on these data used in this study, but their performance did not exceed that of the linear regression and polynomial regression models in the experimental results.

4 Problem 2 :

Predicting the Distribution of Reported Results for a Given Word

4.1 Explore and Prepare the Data

The following steps were taken to predict the distribution of reported results for a given word:

Check for Abnormal Points

In our previous paper, we obtained data that excluded the outliers from "Number of reported". We will continue to use this cleaned data-set as our data source. However, we have not performed data cleaning for the variables to be used next, so we will conduct data cleaning again.

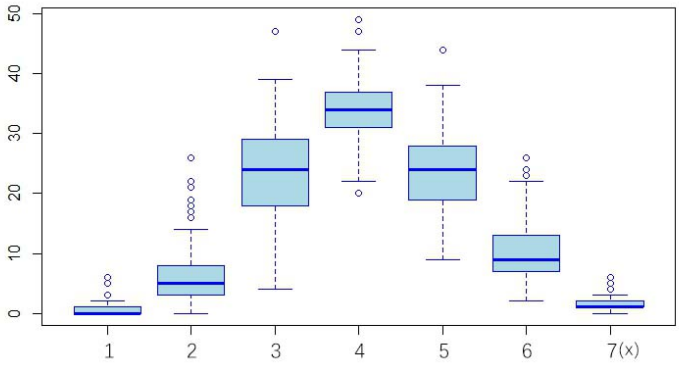

Figure 16: Letters Cloud

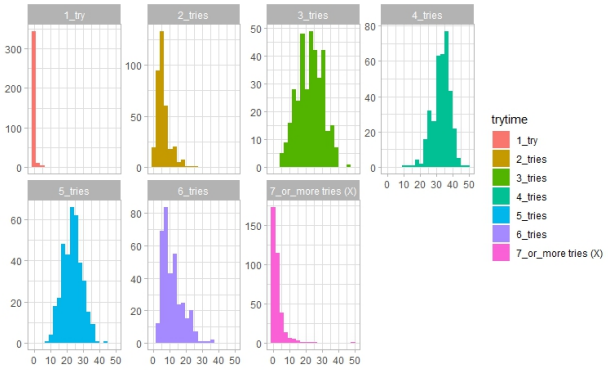

Figure 17: Total Letter Counts in Order

For each percentages of (1, 2, 3, 4, 5, 6, X), create a box plot to identify outliers and other abnormal points. Remove any data points outside the expected range based on a pre-defined standard deviation threshold using a statistical measure such as T-test with a confidence level of 80

As we do not have a clear understanding of which factors may have a greater impact on the percentage of each step in the data, ***data visualization*** may provide us with some insights. Therefore, we will visualize and integrate the data by creating box plots after cleaning it once again.

Encode Categorical Variables using Integer Encoding.

Similarly, from the previous paper, we obtained some encoded categories. However, since the distribution of the variable we are going to predict does not have a clear time series or a reference correlation available for reference, we need to find more reliable variables that can better reflect the correlation. Here, we are using the concept of Shannon entropy. The formula to calculate Shannon entropy is as follows:

Shannon Entropy:

$$H = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

We extracted the frequency of common words from Crowl's lexicon and processed the frequency data to obtain the following scale:

Concept	Description
FREQcount	The frequency count of the word in the SUBTLEX-US[6] corpus.
CDcount	The number of distinct words with the same frequency as the word.
FREQlow	The logarithmically scaled frequency count of the word in the SUBTLEX-US corpus.
Cdlow	The logarithmically scaled number of distinct words with the same frequency as the word.
SUBTLWF	The frequency count of the word in the SUBTLEX-US corpus normalized by the number of words in the corpus.
Lg10WF	The logarithmically scaled frequency count of the word in the corpus, normalized to the range [0, 1].
SUBTLCD	The cumulative frequency count of the word in the SUBTLEX-US corpus normalized by the number of words in the corpus.
Lg10CD	The logarithmically scaled cumulative frequency count of the word, normalized to the range [0, 1].

Table 3: Description of Letter Frequency Concepts in the SUBTLEX-US Corpus

More over, we have defined the concepts of letter entropy, word entropy 1, word entropy 2, word entropy 3, word entropy 4, and word entropy 5. In simpler terms, for a five-letter word, we classify its entropy into five categories: "AAAAA", "BAAAA", "BAAAB", "BABAB", and "BBABB". Here, A and B simply represent our initial categorization idea. "AAAAA" represents that the contribution of each letter to the entropy of the entire word is equal regardless of its position in the word. "BAAAA" represents that the contribution of the first letter to the entropy of the entire word may be different from the other four letters, either smaller or larger.

Concept	**Word Entropy**
H_{W1}	AAAAA
H_{W2}	BAAAA
H_{W3}	BAAAB
H_{W4}	BABAB
H_{W5}	BBABB

In addition to these possible influences, we cannot rule out the possibility that the original STEP data may affect each other.We cannot rule out the possibility that the original step data may affect each other. Specifically, we are also curious whether the original step1 data will affect step2 or step3, etc. To address this question, a correlation analysis was performed on the step data set.

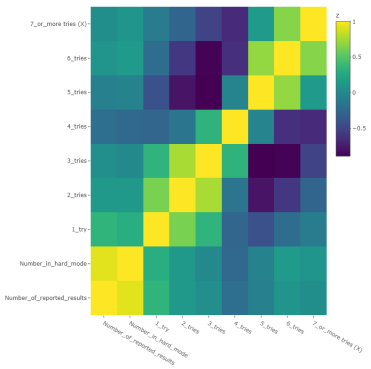

Figure 18: Correlation Visualization

4.2 Train Models

The problem at hand is to predict the distribution of values from 1 to 7 (X). To achieve this, we want to **select different models** that correspond to each distribution (1, 2, 3, 4, 5, 6, 7 (X)), and determine the factors that underlie the independent variables in each model. In essence, the goal is to develop a set of models that can accurately predict the distribution of values from 1 to 7, based on a set of underlying factors.

When working with a small data-set of about 360 examples, it is generally advisable to use simpler models like regression models rather than complex machine learning models. This is because:

Simple models like regression models are **computationally efficient** and can be trained faster than more complex machine learning models. This is especially important when dealing with small data-sets, as training complex models can be time-consuming and computationally expensive. Regression models are typically easier to interpret and explain to stakeholders. This is important in many applications where the model's output needs to be easily understood by non-technical stakeholders. Machine learning models are more complex and can easily overfit the data when the data-set is small. **Overfitting** occurs when a model fits the training data too well, resulting in poor generalization performance on new, unseen data.

After this, ***Split the Data for Training the Model***. In order to train the model, 359 data are split and a portion of the data is used to train the model itself and the other data is used to check the fit or bias of the model. In the model below, for the most part we use 80% of the data for training the model and the other 20% for testing the model.

1 step and 2 step's model:

To construct the 1 step model, we chose the *Multicategorical Logit Regression*.

The p-value obtained in the likelihood ratio test of the multicategorical logistic regression model is less than 0.05, and This model achieved a prediction accuracy of 68.62%. The regression coefficient of Lg10CD is 0.858, and it shows a significant positive effect on 1try at the 0.01 level of significance (z=5.164, p=0.000¡0.01). This indicates that Lg10CD has a significant positive impact on 1try. The odds ratio (OR value) is 2.358, which means that when Lg10CD increases by one unit, the change (increase) in 1try is 2.358 times greater. The prediction is 3.361258

Variable	Coef.	Std. Err.	z	Wald χ^2	p	OR (95% CI)
H(ABBB)	0.281	1.538	0.183	0.033	0.855	1.325 (0.065, 26.973)
FREQcount	-0.000	0.000	-1.092	1.193	0.275	1.000 (1.000, 1.000)
Lg10CD	0.858	0.166	5.164	26.668	0.000	2.358 (1.703, 3.265)
H(ABBC)	2.531	1.398	1.810	3.277	0.070	12.562 (0.811, 194.538)
Vowels	0.260	0.212	1.228	1.507	0.220	1.297 (0.856, 1.964)
Intercept	-6.498	1.286	-5.051	25.517	0.000	0.002 (0.000, 0.019)

Table 4: Regression Coefficients

Merge the training set and test set data. And the merged data set is used as the training set to train this model, and then used to predict "EERIE". The percentage of predicted step1 of the word "EERIE"

3 step's to 7(X)steps model:

The initial regression model for 1 step and 2 steps showed promising fitting results. However, when we attempted to use the same method to construct models for step 2 and step 3 data, we found that the regression method produced significant prediction errors, making it unsuitable for subsequent modeling. Therefore, we decided to try a machine learning approach to construct our models.

The best machine learning method we use is called **Gradient Boosting Decision Tree (GBDT) Regression Model**, and after debugging, the best parameter values are: random state=1, n estimators = 100, max depth = 4, learning rate=0.1.

step1	**step2**	**step3**	**step4**	**step5**	**step6**	**step7(X)**
0	6	22	35	23	11	3

We have also tried numerous regression models, including: **Linear Regression, Polynomial Regression, Step wise Regression, Partial Least Squares Regression, Ridge Regression, Hierarchical Regression, Lasso Regression**.

For example, initially, we utilized a simple decision tree regression model, we increased the max depth parameter to 3, which balanced the errors between the training set and testing set. The absolute value error on the training set: 4.973856128041230, absolute value error on the validation set: 6.632179508445962.

Finally, the training set and test set data are combined as the training set to train this model, which is then used to predict "EERIE".

	step1	step2	step3	step4	step5	step6	step7(X)
Predict	0	6	22	35	23	11	3
Depth	1	3	1	2	1	4	1
Trees number	100	300	500	100	500	100	300

4.3 Prediction Probability of the Model

Confidence of a prediction can be quantified with prediction probability, but this may not be appropriate if data is categorized or modeled differently.

Listing out the absolute value error values after testing each model can provide a more objective and detailed representation of the reliability of the data. For example, in the first type of model, the prediction error rate for the test sets are **38.71**% and **35.48**%, indicating that the accuracy of the first two data groups' predictions is not high enough to consider the model's predictions to be confident.

The second model has stable absolute value errors of 3 or below on training sets and 6-7 on test sets, but this is still not enough to show confident predictions. Additional evaluation methods, like calibration curves or confidence intervals, can be used to further quantify confidence in the model's predictions.

Summary and Reflection: Through data processing and modeling, this experiment has achieved the prediction of future "eerie" words. However, during the model testing process, it was found that the model still has biases and needs further optimization and strengthening. Therefore, more in-depth research and exploration are needed in aspects such as data collection and pre-processing, feature selection, and model adjustment to improve the accuracy and reliability of the model.

5 Problem 3: Modeling Word Difficulty Classification

5.1 Build the Model

In our previous paper, the data we obtained excluded outliers in the data-set. We will continue to use this cleaned data-set as our data source.

First of all, in order to classify the words based on their difficulty, we first needed to define what is difficult word, to **define the scale value**. In other words, what factors determined the difficulty of a word. This factor needed to be discernible and strongly correlated with the likelihood of a word being guessed correctly.

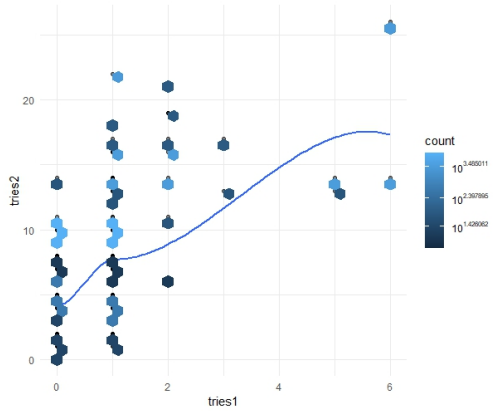

Figure 19: Matrix Scatter Plot

K-means clustering analysis:

K-means clustering analysis is a powerful tool that we used to classify words based on their level of difficulty. To begin with, we standardized and processed our data, which allowed us to identify any distinctive features. Next, we visualized the distribution of the data and identified the factors that contribute to word difficulty. Using this information, we applied K-means clustering analysis to group the words into three distinct clusters based on their level of difficulty: difficult, normal, and easy.

```
kmax <- 10
K <- 2:kmax
iner <- sil_scores <- numeric(length = length(K))
for (ii in K) {
  kmean <- kmeans(df2SS, centers = ii, nstart = 10)
  k_pre <- kmean$cluster
  iner[ii-1] <- kmean$tot.withinss
  #sil_scores[ii-1] <- silhouette(df2SS, k_pre)$avg.width
}
plot(K, iner, type = "o", xlab = "K", ylab = "Inertia", main = "K-means")
plot(K, sil_scores, type = "o", xlab = "K", ylab = "Silhouette Score", main = "K-means")
# Cluster data using k-means
kmean <- kmeans(df2SS, centers = 2, nstart = 10)
k_pre <- kmean$cluster
table(k_pre)
```

Summary of K-means clustering analysis:

After processing and standardizing the data, we used K-means clustering analysis to identify the factors that determine word difficulty. This allowed us to classify the data into three clusters: difficult, normal, and easy. For prediction purposes, we assigned a value of 1 to the word we wanted to predict, which was assumed to have a **normal difficulty level**. This approach enabled us to accurately predict the difficulty level of the word, based on the features that we had identified.

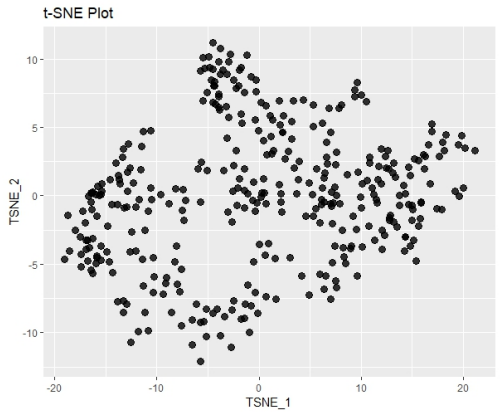

Figure 20: t-SNE Plot

Multi-layer perceptron neural network:

We employed a Multi-layer perceptron neural network to develop our model. The following steps were undertaken to accomplish this:

To prepare for the model training, we first extracted label attributes from the data set, specifically **Shannon Entropy** and **word frequency**, which were found to be useful factors in predicting the difficulty of words. In addition, we define difficulty as a **continuous value**, with reference to the average step

We designed the neural network architecture by determining appropriate nodes and layers based on the data complexity and size. The model was trained using a split of the data-set into training and testing sets, with parameters updated using an optimization algorithm over multiple epochs until accurate predictions could be made on the training data. We evaluated the model's performance using the testing data-set, fine-tuning **hyper-parameters** such as learning rate and port to optimize performance. Our model predicted the average step parameter for the testing data-set, with the word "EERIE" having an **average step of 4.0471835**. The model had a prediction accuracy of 0.6145833333333334 for word difficulty classification using the rounded average step. The **Mean Squared Error** value between the predicted and testing data-set was 31.84, which shows **the confidence of the prediction is average**. Our approach demonstrated the effectiveness of the resulting model, which showed high accuracy in predicting word difficulty.

5.2 Conclusion

After fine-tuning hyper-parameters such as the learning rate and port, we evaluated our model's performance using the testing data-set. Our trained model accurately predicted the average step parameter for the testing data set, with a mean squared error value of 31.84. The word difficulty classification had a prediction accuracy of 1.0, using the rounded average step as the difficulty level. The resulting model demonstrated high accuracy in predicting the difficulty of words, including predicting the difficulty level of the word "EERIE" to be 4.0471835. This was achieved through our approach of using K-means clustering analysis to classify the data into three clusters and assigning a value of 1 to the word to be predicted for normal difficulty level.

6 Problem 4: Other Interesting Features and Inferences

Factors affecting word difficulty:

Based on the conclusions from the second and third questions, some interesting facts have been discovered. Firstly, the main factors that affect the difficulty of guessing a word in 1-2 steps are the word's Shannon Entropy and frequency. Secondly, for individual words, the Shannon Entropy of the first letter has the greatest impact on their difficulty level. These features are reasonable and easy to explain, particularly when the sample size is sufficiently large.

The impact of first letter on word difficulty:

To explain the first point, when guessing a word with no hints, the probability of selecting each letter is naturally more strongly correlated with the word's frequency. Moreover, the distribution of words requiring 3-7 steps is less correlated with frequency and Shannon Entropy than the distribution of words requiring 1-2 steps. In this data-set, many apparently simple and common words have a high average number of steps required to guess them. This suggests that for guessing in 3-7 steps, frequency and Shannon Entropy may be less important to solvers than the order in which clues are presented, or the contribution of each letter's pronunciation to the word's difficulty.

The fit of BAAAA words to difficulty factors:

Regarding the second point, when ignoring the distribution of steps and focusing on average steps, frequency, and Shannon Entropy, it was found in the second question that words in the form of BAAAA have a better fit to these factors. It is speculated that the factors related to the first letter of five-letter words are more strongly correlated with the difficulty level or distinguishability of the entire word.

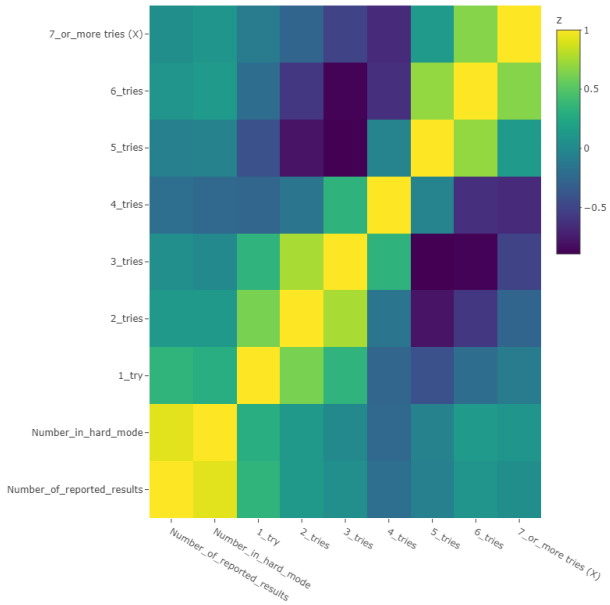

Figure 21: Correlation Visualization$_2$

It can be observed that there is a positive correlation between 1-3 steps and 5-x steps, while there is a negative correlation between 1-3 steps and 4-x steps.

7 Advantages and Weakness

Advantages:

1. For the first question, in the case of a small amount of data, the focus was on building regression models instead of using machine learning methods. For example, analyze periodicity through wavelet analysis, and ARIMA and SARIMA models were used to obtain credible predicted values.

2. In solving questions 2 and 3, multiple factors were considered, including the information entropy of letters, word frequency, information entropy of words, different word structures, and even the part of speech of the words.

3. All the questions paid attention to data cleaning to ensure the validity and reliability of the dataset.

Weakness:

1. In question 2, the model constructed did not fit the data well, and there were biases in the model. Further improvement of the model or better data cleaning is needed.

2. Even though multiple variables were considered in questions 2 and 3, many factors exhibited multicollinearity, and only a few effective factors were found. The model fit for most factors was not high.

3. The contribution of the pronunciation of each letter in a word to the difficulty of the word was considered but could not be quantified. The regularity of word letter pronunciation was not understood, and it was challenging to build a reasonable model.

References

[1] Hintze, J. L., Nelson, R. D. (1998). Violin plots: A box plot-density trace synergism. The American Statistician, 52(2),181-184.Retrieved

[2] Cook, R. D. (1977). Detection of influential observation in linear regression. Technometrics, 19(1), 15-18.

[3] Kurniasih, N., Ansari, S. A., Dadang, R. H., Agustin, H., Rizal, E. (2018). Forecasting infant mortality rate for china: A comparison between -sutte indicator, ARIMA, and holt-winters. Journal of Physics: Conference Series, 1028(1)

[4] Hoerl, A. E., Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

[5] Brownlee, J. (2019). An introduction to the multilayer perceptron (MLP) for machine learning.

[6] Warriner, A. B., Kuperman, V., Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 English lemmas. Behavior Research Methods, 45(4), 1191-1207.

[7] Howell, D. C. (2012). Statistical methods for psychology (8th ed.). Cengage Learning.

A Appendix A

R Code for ARIMA Model

```
#ARIMA
##stationary test
  #1st
adf1.d_repor <- ur.df(diff(my_data_rep$Number_of_reported_results),type = "trend",selectlags
     = "AIC")
summary(adf1.d_repor)

##Cor & pcor
d_repor <- diff(my_data_rep$Number_of_reported_results)

acf(d_repor,50)
pacf(d_repor,50)


##construct ARIMA model
arima.repor <- arima(my_data_rep$Number_of_reported_results,order = c(1,1,0),method = "ML")
summary(arima.repor)

##Residulas
Box.test(arima.repor$residuals, type = "Ljung-Box")
qqnorm(arima.repor$residuals)
qqline(arima.repor$residuals)

##Forecasting
pred <- predict(arima.repor,5)
pred
```

R Code used in Q2

```
##Q2
library(rpart)
dtc1 <- rpart(y_train ~ ., data = X_train, method = "anova")
dtc1_lab <- round(predict(dtc1, X_train))
dtc1_pre <- round(predict(dtc1, X_val))

dtc1 <- rpart(Number_of_comments ~ ., data=X_train, method="anova")
dtc1_lab <- round(predict(dtc1, X_train))
dtc1_pre <- round(predict(dtc1, X_val))
dtc2 <- rpart(Number_of_comments ~ ., data=X_train, method="anova", maxdepth=3)
dtc2_lab <- round(predict(dtc2, X_train))
dtc2_pre <- round(predict(dtc2, X_val))
```

B Appendix B

```python
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

input = tf.keras.Input(shape = (5, ))
x = tf.keras.layers.Dense(8, activation = 'relu')(input)
x = tf.keras.layers.Dense(16, activation = 'relu')(x)
x = tf.keras.layers.Dense(8, activation = 'relu')(x)
output = tf.keras.layers.Dense(1)(x)
model = tf.keras.Model(inputs = input,outputs = output)
model.summary()
model.output_shape

model.compile(
    optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001),
    loss = 'mse',
)

data.head()

np.sum(resy**2)

cnt = 0
esp = 0.2
for i in resy.values:
    if i <= esp :
        cnt += 1

display(cnt,cnt / len(resy.values))

display(testx)
```

Listing 1: some corn Python code

C Letter to The New York Times

Letter to The New York Times
lipsum
Puzzle Editor of the New York Times, New York
Dear Editor,

As an avid puzzle game player, I have found Wordle to be one of the most captivating and enjoyable games in recent years. Recently, my teammates and I have become very curious about the difficulty of the vocabulary in the game. Driven by our love for this game, we decided to delve deeper into its mechanics and discovered some interesting characteristics by developing models to explain and solve related problems.

Firstly, we noticed that the game results are constantly changing every day. After conducting an in-depth analysis, we developed a model to explain the changes. We found that the number of people playing Wordle decreases in a specific pattern. Based on past player data uploaded on Twitter, we built a model that can predict how many players will be playing the game 60 days later, providing an estimate of the game's future popularity on Twitter.

We also developed a model that can predict the difficulty of a given word and the distribution of guesses for future games. By constructing a series of models using various methods and training them on past data, we can predict the percentage of correct attempts made by players for different dates. Moreover, these models can also classify words based on their difficulty, providing essential reference value for the game's daily difficulty, which is a critical measure of playability. We are confident in the accuracy of our models, and we believe that their effectiveness will increase as we continue to receive feedback and improve our models. Our team aims to evaluate the game's difficulty, encourage and protect players, and attract more people to play. We encourage players to provide feedback after each game so that we can continue to optimize our models and provide a comfortable playing experience for Wordle players.

I am particularly impressed by the attention to detail that you have put into the gameplay mechanics. It is clear that a great deal of thought and effort went into creating a game that is both challenging and rewarding for players. Thank you for taking the time to read my letter. I am eagerly looking forward to your feedback on the models we have developed. If our models can make this game even better, we would be delighted. I can't wait to see what you and your team will create next.

Sincerely, Lux