Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2316053

Cracking the Past for Future: Comprehensive Analysis of Wordle Players' Participation and Performance via Time Series, ML, and NLP

Summary

The New York Times introduced a web-based word game called Wordle in 2022. The game is to guess a five-letter word in six tries or fewer. However, the data from January 7, 2022, to December 31, 2022, shows that the number of reported players continued to decline after a short-term increase. Therefore, through historical data analysis, this report discovered some potential patterns for the game to enjoy long-term prosperity.

We first performed data cleaning and adjustments followed by analyzing the relation of attributes and times. We conduct time series analysis and natural language processing methods to retrieve 82 time-series features and 47 word-related features. The goal of our model is to find out the number of reported results' trend with time, the relationship between the word attribute and the distribution of tries, and predict the distribution guessed at a given day. Specifically, the study includes the following parts.

Firstly, to predict the number of reported results, we used GluonTS and ARIMA to make a prediction of March 1st and compared the models; based on those, we successfully provided a prediction interval. We also find a strong correlation between hard mode percentage and the time while the goal is to predict its relationship with word features. Therefore, we picked a less correlated interval, and use different machine-learning algorithms as models.

Secondly, to predict the distribution of the reported results on March 1st with the word "EERIE", we use both time series and word-related features and apply machine learning models containing XGBoost, Random Forest, and Tree Algorithms to train the data. The label of difficulty levels was derived using the K-means algorithm to classify the data into easy, medium, and hard modes. We also apply the machine learning model to predict the classification and we achieved an accuracy rate of 73% and succeed to classify the word "EERIE" as a hard difficulty level, which confirmed our assumption.

Based on analysis, we wrote a letter of suggestions to the New York Times for future game development. Our suggestions include introducing new features to the game, and making the game more consistent, inclusive, comprehensive, and attractive. By using various data processing techniques to analyze the Wordle game's data and discover its features, our predictions and suggestions can help the New York Times improve the game and attract more players.

Keywords: Difficulty classification, Wordle, time series analysis, Natural Language Processing, machine learning, GluonTS, XGBoost, Random Forest, Tree, ARIMA

Contents

1 Introduction

Throughout world puzzle game history, vocabulary games like Crossword puzzles, Scrabble, Spellingbee, and Wordscapes have long had a great vogue, becoming overnight sensations after their release, receiving critical acclaim worldwide. Following their step is Wordle, which is another new online word puzzle game that first came out in Jan 2022. It was invented by Josh Wardle and has quickly gained popularity and has a prevailing advantage during the COVID-19 pandemic, providing users with tremendous pleasure through the mental challenge to guess a hidden five-letter word within no more than six attempts by filling in blanks of a complex matrix of letters as hints [1]. With every guess, the agent receives immediate system feedback regarding if the letters guessed are present in the target word as well as if the position is correct by the system of color. Specifically, grey, yellow, and green respectively indicate if each letter is absent, present but not in the correct position, or presented and in the correct position, as shown in Figure 1 [1]. Each day, hundreds of thousands of players around the world play to guess the same word, excavating the charm of the game.

However, the study finds that Wordle's users have been on the decline since the end of February 2022, as shown in Figure 2; it is essential to derive models to analyze the game via user participation and performance in order to have a lasting influence and generate incentive mechanisms or policies to sustain Wardle's success. The Wordle company has to skirt the boundary between making the difficulty too hard and too easy to achieve that balance can increase customer stickiness engagement, and retention willingness. Many factors affect word difficulty from the linguistic convention and natural language processing perspectives, including characteristics like word frequency, word range, academic language, word recognition norms, contextual distinctiveness, age of ac-

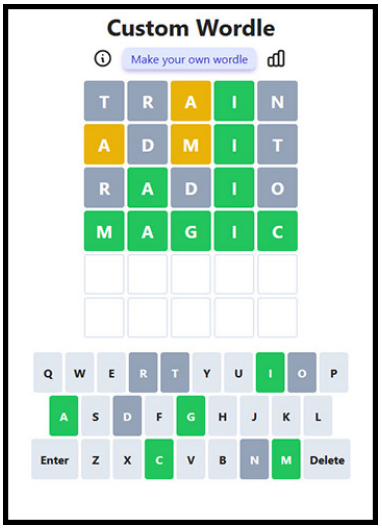

Figure 1: Illustration of Wordle, from February 18, 2023 "Magic".

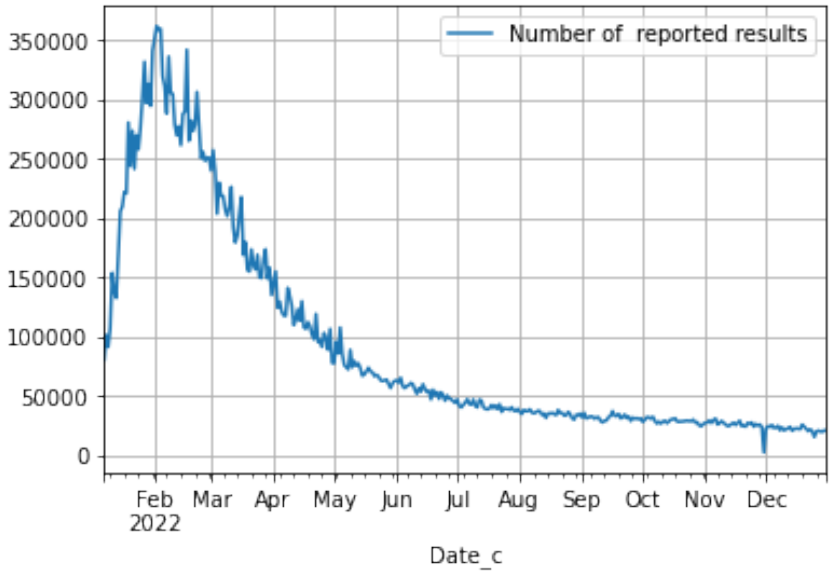

Figure 2: Number of reported results.

quisition, etc. Finding the features that best optimize the difficulty classification accuracy is the focus of this study. All these considerations must be taken into account in developing a successful difficulty mechanism and providing a suitable solution for real-world configurations.

In this study, we facilitate several quantitative methods to analyze the correlation among different data series and present several algorithms for analyzing data patterns and movements of the historical data of Wordle. Then, given the implementation of this algorithm, we simulate the pattern in a given scenario to predict the future interval and implement several correction algorithms to minimize the error.

The approaches we use to realize such goals are:

(1) We use the Pandas Profile repository to discover the interesting feature of each dataset. given the pattern of each feature, we hope to eliminate aspects influencing the word difficulty. Then, we set s series of assumptions that are listed in Section 3.1. After that, we implement K-means to classify the dataset into as fewer clusters as possible. Based on the assumption, we give a temporary prediction on the word "EERIE."

(2) Based on the prediction and limited dataset, We first implement XGBoost to preprocess the time series data, and we implement some NLP models (textstat, nltk) to figure out different features that could affect the word difficulty; finally, we implement a multiple-regression hybrid model to predict the word "EERIE" difficulty and check whether it meets our assumption.

(3) We use the distribution of tries to define difficulty, and we also plot the number of tries over time; we don't see a clear trend of these values over time. Thus, we will use all these values to analyze the difficulty of words. After we define the difficulty, we use the same model XGBoost for classification since we wish to elaborate the word further on their attributes. The loss function we facilitate is $softmax()$.

2 Problem Restatement and Analysis

Past research using Wordle mainly explores winning strategy from user perspectives, yet this study seeks to analyze for agents to provide business implications and difficulty construction through real-world users' frequency and participation data. This research constitutes a relatively new area that has emerged from difficulty classification.

First of all, according to the given question, this study is required to build a comprehensive and predictive model that explores the total reported results with the passing of time. It will incorporate a model that considers distinctive features' relationship with good time applicability. A challenging problem that arises in this domain is incorporating time series analysis with machine learning and natural language processing.

Second, the study reports the associated percentages distribution of attempts for a specific solution on a given future date. It should also provide uncertainties, confidence, and limitation in the model prediction.

Moreover, the study should first classify the difficulty of the dataset to get labels via

clustering algorithms based on the number of attempts. Then it develops and trains a model that identifies word attributes or features that impacts difficulty using features from Natural Languages Processing. We then explore the best features combination and machine learning algorithm to get an accuracy rate. Lastly, the study is asked to classify EERIE into one of the categories of the easy, medium, or hard rating, given the proposed model.

Considering all the requirements above, our goal is to construct a comprehensive analysis of Wordle players' participation and performance via time series, Machine learning, and natural language processing to meet our assumption. In light of the appropriate strategies considered to address this issue, the study attached a letter using plain language for New York Times and it should explain our difficulty selection criteria and the aspects of the judgment.

3 Assumptions

To simplify our model and reduce total complexity, we make the following assumptions in this project. All assumptions will be restated if it is once used in our model. For instance,

- **Reduced dimension due to discovery via correlation matrix.** Given that "1 try" players involve extreme luck and potential cheating, and high-correlation values exist between 2 and 3 tries; and the same for 5,6, and 7 tries [figure 1], the study decides to abandon the "1 try" column, and combine the data of 2 and 3 tries with reasonable weights, and 5,6,7 tries the same operation.
- **Set K-means results as a standard.** When doing classification, we assume the K-means algorithm gives a better clustering standard, and we set the results as a part of training standards for NLP modeling.
- **Exclude cheating.** We assume each player did not collaborate with other players.
- **Players are serious.** We assume every player plays each game seriously; they do not abandon each game or put in the wrong words on purpose.
- **Data Reliability.** We assume that the data we have chosen is true and reliable.
- **Ignore individual differences.** The majority of individual differences between players, e.g., individual intelligent quality and educational level, are ignored.

4 Data Preprocessing

4.1 Data Preprocessing Analysis

Data preprocessing is an essential step in any data analysis which involves transforming raw data into a usable format for analysis, improving the accuracy and quality of the results obtained.

In this paper, we describe the data preprocessing steps taken for our project.

- **Data collection.** The data source is user performance information in Wordle, entailing information on user participation (number of reported results) and performance(number of attempts to succeed), the data also contains the number of hard modes, which will lay the basis of our discussion around the interesting top of hard mode and true difficulty level, as well as future recommendation to New York Times. The data collected is relevant to analyzing our research question and is of high quality as it contains few error terms, and according to our assumption, we believe that the data is reliable despite several discrepancies of words.

- **Data cleaning.** Once the data is collected, it is necessary to clean it by removing any irrelevant or duplicate data. We corrected typos of "marxh" into "march", and fixed formatting errors of addition space – "favor " into "favor". Since the data set should be a collection of two words, we also corrected the four letter word "clen" and "trash" into "clean" and "trash". For the six-letter word "rprobe", we modified it into the original form "probe". It is important to ensure that the data is consistent and accurate, as incorrect data can affect the accuracy of the analysis.

- **Data transformation.** In the natural language processing part, the data needs to be transformed into digits to make it suitable for analysis. This transformation is based on the formatting of words, for example, "EERIE" is in the form of AABCA, thus corresponding to 11231. In the time series analysis for ARIMA, we normalized the data by differencing and converting it to stationary series. The goal of data transformation is to make the data suitable for the analysis techniques that will be used.

- **Feature selection.** Our considerations of NLP feature selection are based upon repetition, vowel number, uncommon structure, rare letter, and cognitive level, more details refer to 6.1 of the article.

- **Data integration.** We reduce the dimensionality of the data from seven categories (try1, tries2, tries3, tries4, tries5, tries6, tries 7 or more) into (tries22, tries 4, and tries567) in the K-means clustering, which reduces into 3 dimensions, and improve the accuracy of the analysis. We made this categorization based on correlation analysis to select the most relevant features.

- **Data spliting.** We also split the data into training, validation set, and testing set. In ARIMA model of the time series forecasting part and machine learning models of question 3, the percentage chosen is 90% training set, 10% testing set. In GluonTS, the percentage is 2/3 of the training set and 1/3 testing set. In question 1B, we run the model firstly for the day index after 220 since the difficulty ratio is more stable, and secondly for the whole dataset.

4.2 Interesting discovery about dataset (Question 4)

First, the total number of players in the game is declining considerably after only 1 month of the game, and after April, the remaining 1/5 users (compared to peak time) keep

stable in the game. This shows that more users are leaving the game. Data visualization of number of reported results can be found under Introduction.

Percentage of different tries visualization is shown in Figure3, demonstrating general bell shaped distribution of attempts from 1 try to 7 tries. Calendar plot of reported result and number of hard mode are shown in Figure 3, 4, 5 respectively, signaling no apprent fluctuation in reported result and hard mode across different week days. Our data analysis found that the difficulty of the game, the distribution of scores, and the game settings are all in conformity with normality. This indicates that the game's design is thoroughly thought out and executed precisely.

Moreover, we discover that "Hard Mode Percentage" show a weak correlation with word difficulty, indicated in Figure 24, which suggests that choosing hard mode has no effective influence on the total genre of word difficulty.

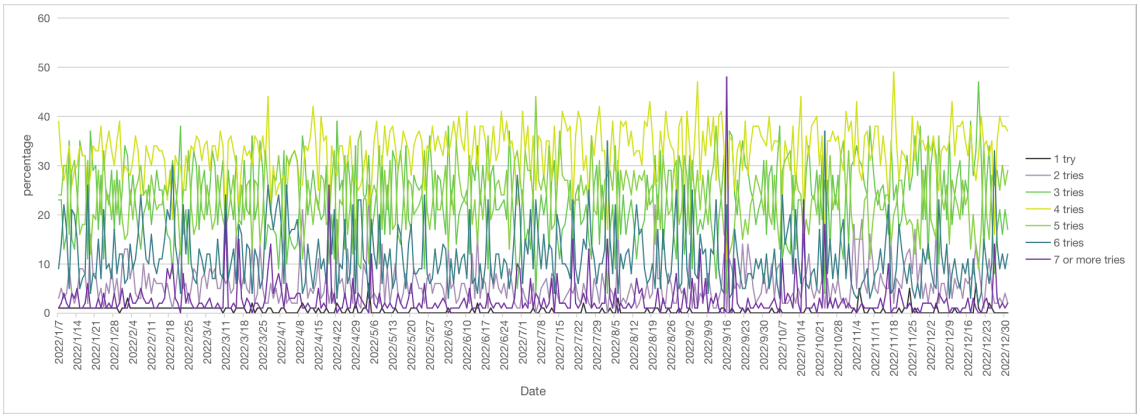

Figure 3: visualization of the percentage of different tries in time series

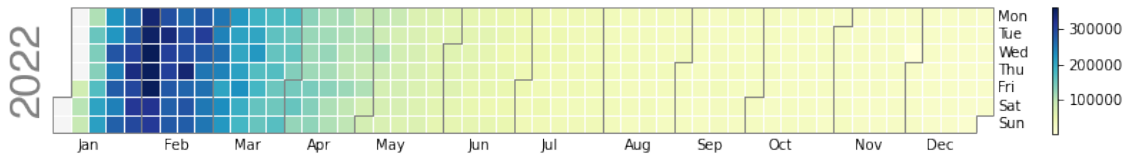

Figure 4: calendar plot of number of reported result

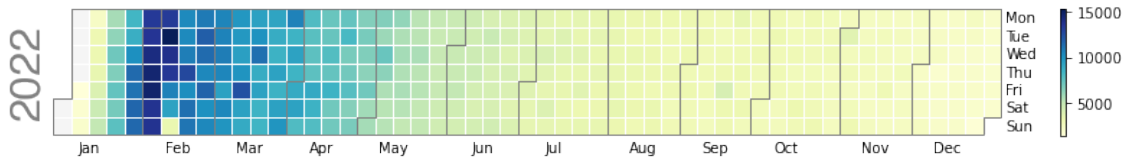

Figure 5: calendar plot of number in hard mode

5 Time Series Forecasting Model (Question 1A)

5.1 ARIMA Time Series Analysis

Autoregressive Integrated Moving Average model (ARIMA) is one of the time series [4] predictive analysis methods [2]. In ARIMA(p, d, q), AR is "autoregressive", p is the number of autoregressive items; MA is "sliding average," q is the number of sliding average items, and d is the number of differences made to make it a stationary sequence (Order)[2]. Although the word "difference" does not appear in the English name of ARIMA, it is a key step. Its principle is to transform the non-stationary time series into a stationary time series and then regress the dependent variable only on its lag value and the present value and lag value of the random error item to establish a model [4].

The ARIMA model with p autoregressive terms, d differencing terms, and q moving average terms consists of three parts. The autoregressive part of the model is represented by

$$(1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p)$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the auto-regressive coefficients. The moving average part of the model is represented by

$$(1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q)$$

where $\theta_1, \theta_2, \ldots, \theta_q$ are the moving average coefficients [13]. The differencing part of the model is represented by

$$(1 - L)^d$$

where d is the order of differencing. Therefore it is defined as:

$$(1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p)(1 - L)^d y_t = (1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q)\varepsilon_t \qquad (1)$$

where L is the lag operator, which shifts the time series back to one time period.

5.1.1 STL decomposition

STL decomposition breaks down time series data into its constituent parts, including trend, seasonal, and remainder components [5]. The method works by applying a locally weighted regression technique (LOESS) to the time series, estimating the trend and seasonal components, and subtracting them from the original time series to obtain the residual component; in the study, STL decomposition results shows in Figure 6.

The trend component represents the long-term behavior of the series. In contrast, the seasonal component captures any regular, repeating patterns that occur over a fixed period of time, such as weekly or monthly cycles. The remainder component represents any remaining variation in the series not accounted for by the trend or seasonal components. Here we use the elements above to deduct STL decomposition.

Let y_t denote the time series, where $t = 1, 2, \ldots, T$, and use the Loess smoothing formula first:

$$\hat{y}_t = S_{t,m} + R_t$$

where $S_{t,m}$ is the value of a locally-weighted regression smoother (loess) with a span of m centered at time t, and R_t is the remainder at time t, given by $R_t = y_t - S_{t,m}$. Here we have seasonal component:

$$\hat{y}_t^{(1)} = m_t + \gamma_1(y_t - S_{t,m})$$

where m_t is the moving average of $\hat{y}_t^{(0)}$ over a window of length L, centered at time t, and $\gamma 1$ is a scalar parameter that controls the strength of the seasonal smoothing. The seasonal component is given by $S_t^{(1)} = \hat{y}_t^{(1)} - R_t$. Here we have a trend component:

$$\hat{y}t^{(2)} = m_t + \gamma_2(y_t - S_{t,m})$$

where m_t is the moving average of $\hat{y}t^{(1)}$ over a window of length L, centered at time t, and $\gamma 2$ is a scalar parameter that controls the strength of the trend smoothing. The trend component is given by $S_t^{(2)} = \hat{y}_t^{(2)} - R_t$. Given Above, we have the remainder component is given by $S_t^{(3)} = R_t$.

Overall, the STL Decomposition formula is:

$$y_t = S_{t,m} + S_t^{(1)} + S_t^{(2)} + S_t^{(3)} \tag{2}$$

5.1.2 Differencing of one means in ARIMA models

From ARIMA, finding the first and second differences of a time series is commonly used in transforming a non-stationary time series into a stationary one, ensuring that statistical properties like mean and variance of the time series model remain constant over time, as illustrated by Figure 7 [2]. Differencing one means involves taking the first-order difference of the time series, the formula for it is,

$$\Delta y_t = y_t - y_{t-1}. \tag{3}$$

Differencing two means involves taking the second-order difference of the time series, which can be derived on top of the previous formula,

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1}. \tag{4}$$

which is,

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1} = y_t - 2y_{t-1} + y_{t-2}. \tag{5}$$

The differencing is crucial for future objectives in implementing ARIMA and ensures that the result derived from our model yields better estimation.

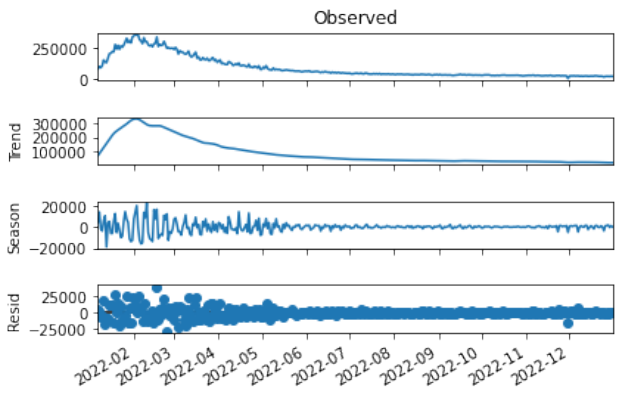

Figure 6: STL decomposition of the number of reported result time series.

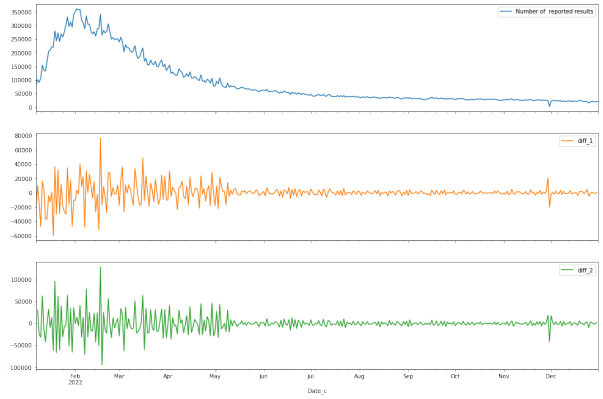

Figure 7: Differencing of the number of reported result time series.

5.1.3 Autocorrelation and Partial Autocorrelation Analysis

Autocorrelation function (ACF) features an ordered sequence of random variables comparing with itself, reflecting the correlation between the values of the same sequence in different time series points.

On the other hand, the partial autocorrelation function (PACF) [5] is a method to characterize the structure of a stochastic process. For a stationary AR(p) model, when the lag k auto-correlation coefficient p(k) is obtained, it is actually not a simple correlation between X(t) and X(t-k). Instead, autocorrelation coefficient p(k) is actually mixed with the influence of other variables on X(t) and X(t-k). After the partial autocorrelation function eliminates the interference of the middle k-1 random variables X(t-1), X(t-1), ... X(t-k+1), the partial autocorrelation function represents the degree of correlation of the effect of X(t) on X(t-k).

ACF also includes the influence of other variables, while the partial autocorrelation coefficient PCAF is strictly the correlation between two variables.

$$ACF(k) = \frac{Cov(y_t, y_t - k)}{Var(y_t)} \tag{6}$$

The j^{th} term in the ACF is denoted as $\phi_{k1}x_{t-1} + \phi_{k2}x_{t-2} + \cdots + \phi_{kk}x_{1-k} + u_t$ Then the k ACF model can be written in the form of the following:

$$x_t = \Phi_{k1}x_{t-1} + \Phi_{k2}x_{t-2} + \cdots + \Phi_{kk}x_{t-k} + u_t. \tag{7}$$

From previous steps, we can deduct the PACF model equation can be written in the form of:

$$\Phi_{k1}x_{t-1} + \Phi_{k2}x_{t-2} + \cdots + \Phi_{kk-1}x_{t-k} + u_t = \Phi_{kk}x_{t-k} + u_t \tag{8}$$

Decision Principle of terms in ARIMA(p,d,q) is demonstrated via Table 2. In Figure 8,

we observe the ACF and PACF we derived the p and q terms of ARIMA to be 4 and 2, respectively.

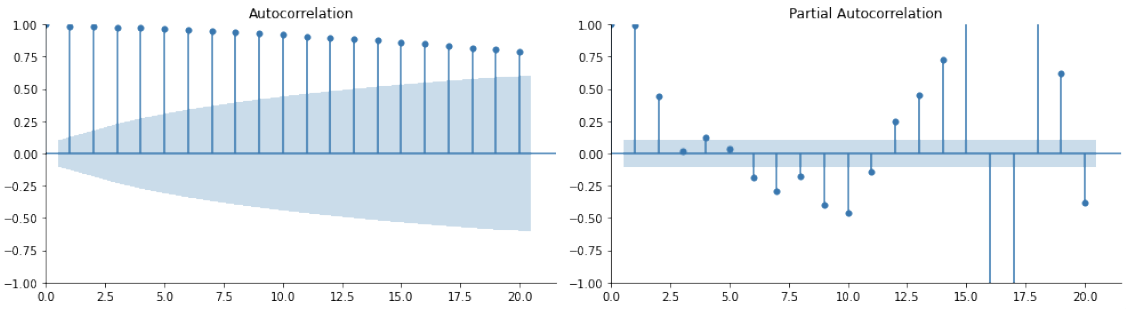

Figure 8: ACF and PACF plots

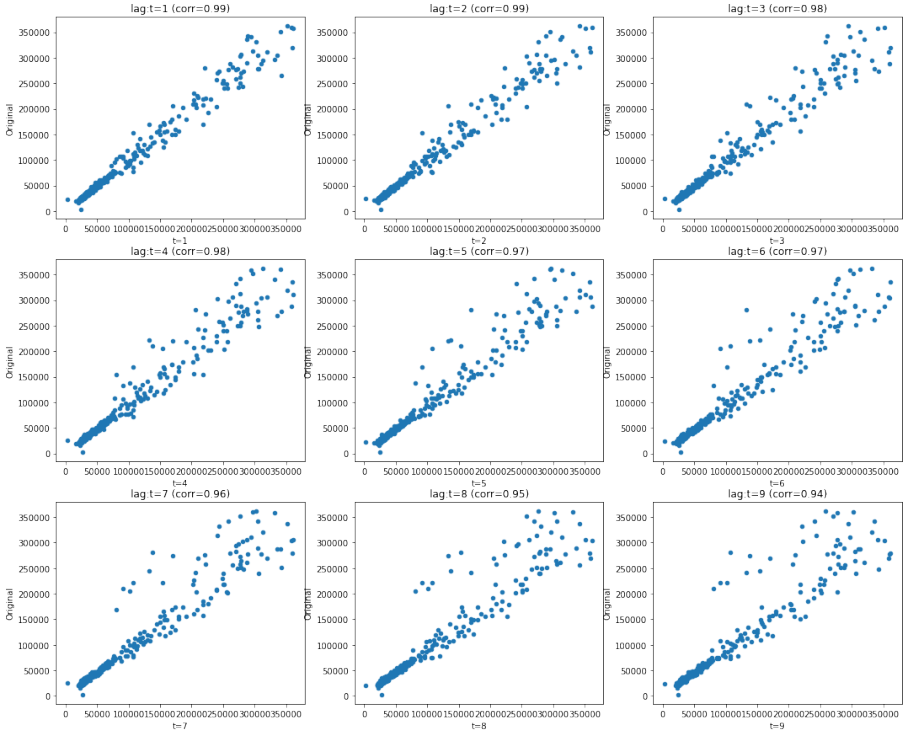

Figure 9: Scattered Plot of ACF and PACF with different lag(t) values (t=1,2...9), from the plots above, intuitively, we could choose t = 1 or t = 2 as candidates for lag value.

5.2 DeepAR Model from GluonTS

This study also utilizes GluonTS, an open-source toolkit developed by Amazon for building and training deep learning models in time series forecasting to complement the previous naive model of ARIMA. GluonTS's based on the deep learning framework MXNet enable it to deal with large-scale data accurately [3]. Specifically, the coding part utilized in the study incorporates the package of DeepAREstimator [6], deep learning algorithm designed explicitly for forecasting. The plot for ARIMA and GluonTS predictions are shown in figure 10 and 11.

Table 1: Decision Principle of terms in ARIMA(p,d,q)

model	ACF	PACF
AR(p)	Attenuation approaches 0	fall within CI after p term
MA(q)	fall within CI after q term	Attenuation approaches 0
ARMA(p,q)	Attenuation approaches 0 after q term	Attenuation approaches 0 after p term

CI in the table stands for confidence interval.

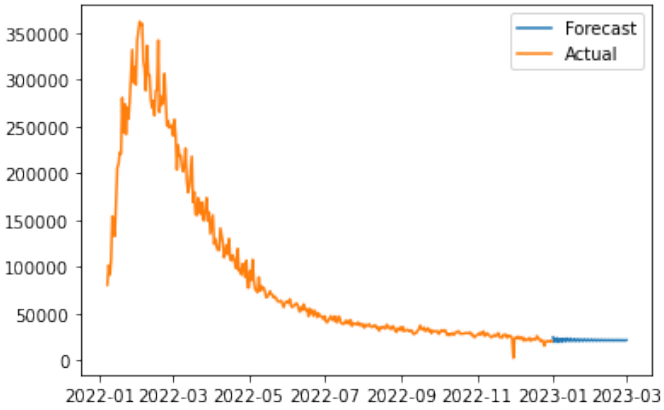

Figure 10: Forecast of ARIMA result

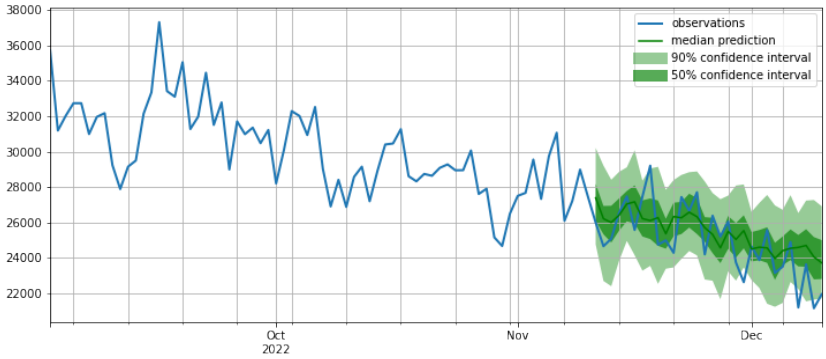

Figure 11: Forecast of GluonTS result

6 Word influence on the percentage of scores reported that were played in Hard Mode (Question1B)

6.1 Prepossessing

In common sense, the percentage of scores reported that were played in Hard Mode is calculated by the ratio of the Number in hard mode and the number of reported results, such that

Table 2: Prediction Value on March 1 2023

model	prediction result	95% confidence interval
ARIMA	21707.364079	(19741, 23641)
GluonTS	21721	(18863, 24323)

$$\text{The percentage of Hard Mode reported} = \frac{\text{Number in hard mode}}{\text{Number of reported results}} \tag{9}$$

However, this study discovers that this ratio is highly correlated with time. The study finds that the ratio increases as the day increases regardless of words. This might be because when time passes, there are more experienced players who tend to play hard mode to challenge themselves. It could also be that more strategies have been developed and players are improving. To better analyze the attribute of the word that affects such a ratio, we expect the ratio more dependent on word attributes instead of time. Therefore, we think of two methods. One is to adjust the ratio according to the time influence. However, the influence of time on the ratio is complex, and this method might add too much additional noise to our ratio and will make our analysis of word attributes less convincing. The second method is to pick up a time interval the ratio doesn't change significantly as time passes. In such a way, we assume that the ratio within this time interval has a low correlation with time, so the influence of word attributes is more significant among such intervals.

To find such an interval, we plot the ratio over time as the Figure 12. The graph shows that for the first 220 days, the ratio increases shapely as time increases. The previous early date data might add too much noise to the analysis of the word since it is greatly influenced by time. Therefore, we only use data when the ratio over time is relatively stable, and we choose the period after day 220.

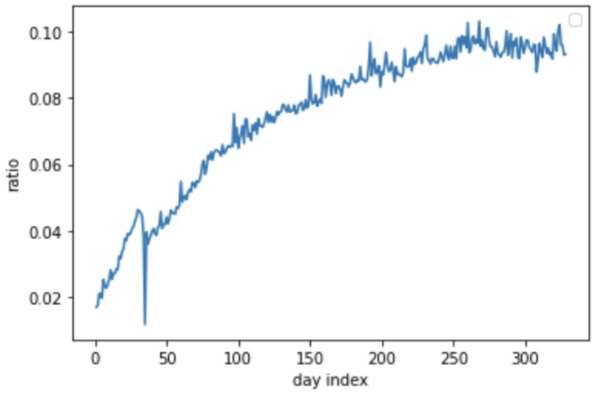

Figure 12: Ratio over time

We analyze multiple aspects of the attributes of the word. We consider mainly three

kinds of attributes: letter constitution, understanding level, and similarities between 5-word corpus. In such a way we retrieve 47 attributes of words.

6.1.1 Letter constitution

The letter constitution represents the structure of a word. It provides the fundamental information of the word in such a Wordle game.

- **Repetition in letter constitution:** We consider the maximum letter repetition of the word. For example, the maximum letter repetition of the word "good" is 2.
- **Syllable count:** We count the number of syllables of the word.
- **Word format:** We analyze the structure of letters in the word. For example, the format for "EERIE" is 11231.
- **Letter of the word:** We analyze specific letters in the word that have some obvious correlation to the distribution of tries and ratio. The letters we choose are: 'f', 'g', 'j', 'm', 'p', 'q', 'v', 'w', 'x', 'z', 'oo', 'ir', 'ph', 'ly', 've', 'wh', 'sk', 'ch', 'ck', 'ng', 'qu', 'th', 'ie'.
- **Understanding level:** The understanding level is the degree to which the meaning can be understood when every single word appears. It shows how likely a person can recall the word.
- **Difficult score:** If the syllable count is 2 or greater, we consider it hard; otherwise, easy.
- **Brown frequency:** The word frequency in the Brown Corpus.
- **Reuters frequency:** The word frequency in the Reuters Corpus.
- **Others:** I-Mean-RT, I-Zscore, I-SD, Obs, and I-Mean-Accuracy are attributes in Data For Word Difficulty Prediction, retrieved from IEEE data port [7].

We assume that the data we have chosen is true and reliable.

6.1.2 Similarities between 5-word corpus

We check the rules for Wordle carefully and find that

> "There are 2,315 possible secret words and the game accepts 12,972 possible words as guesses [1]."

We consider them as two separate word lists: one with 2315 words and another with 12972 words.

We measure the similarities among these words by Levenshtein Distance [9].

We analyze the Levenshtein distance of the word to all the words in each of the list and count the number of Levenshtein distances as 1,2,3,4,5(since there are only 5 letters, the maximum Levenshtein distance is 5). We name the attribute in the 2315 corpus as "short_1", "short_2", "short_3", "short_4", "short_5"; the attribute in the 12972 corpus as "long_1", "long_2", "long_3", "long_4", "long_5".

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{a_i \neq b_i} \end{cases} & \text{otherwise.} \end{cases} \tag{10}$$

6.2 Models

Since we only use 140 days of data. The neural network requires much more data. Otherwise, it is easy for the neural network to overfit such a small amount of data. Moreover, we use tree-based ensemble models better to explain these attributes towards the effect on the ratio. We use XGBoost, which performs best among all the tree-based ensemble models. The study applies RMSE as loss, and we set the l2 constraint to 1 to prevent the over-fitting of our model. We also compare XGBoost with other tree-based methods, such as basic decision trees and random forests. We also compare it with a basic linear regression model and neural network. The comparison is shown in Table 3:

6.3 Result Analysis

The result of the feature importance of our XGBoost model is as Figure 13. We see that Word format, and I Mean Accuracy have the most significant effect on the percentage of Hard Mode reported. Following that is the single-letter constitution of the word.

Since this graph only uses 140 days' data, the data set is relatively small and might cause the problem of overfitting. Many letters of the words currently being considered important might just be because these letters accidentally occur more during these 140 days. To confirm the significance of these single-letter attributes, we use the entire 360 days of data to get an attribute importance graph(see Figure 14) for comparison. In such a way, we can better understand the importance of these attributes.

Although the entire 360 days of data is influenced by time, it still contains important

Table 3: Model comparison and best model selection question1

	model_type	loss	train_time
	Decision Tree	0.0242211	15.5
	linear	0.024519	4.26
the best	XGBoost	0.0234594	4.5
	Neural Network	0.0379846	2.35
	Random Forest	0.0242363	6.72

work-related information. By comparing with the previous Figure 14, we find that the single letter in the word attributes are less significant here, which might support our hypothesis earlier that the single letter attributes might only be significant within the previous time interval. Thus, we think the format of the word and I Mean Accuracy are important attributes of the ratio. The I Mean Rt, I Zscore, I SD, and short 1 attributes are also practical since they ranked high in both figures.

We use the 140 days data model for prediction. The prediction of the percentage of Hard Mode reported of the word "EERIE" is 0.10227332. The training and test RMSE for the 140 days' data model is 0.002495, 0.002649. The two numbers being similar make the result compelling.

7 Word Attribute Enhanced Distribution Forecasting in Time Series (Question 2)

7.1 Time series features

In this problem, we try to combine times series with word attributes. We use word attributes from section 6.1 and develop new time series attributes.

For time series, we take the moving average and weighted moving average of 3,5,7,30 days [11]. The attributes we take moving average include the Number of reported results, Number in hard mode, Percentage of Hard Mode, number of 1 try, 2 tries, 3 tries, 4 tries, 5 tries, 6 tries, and 7 or more tries (X).

The formula for calculating the moving average of a time series can be expressed in

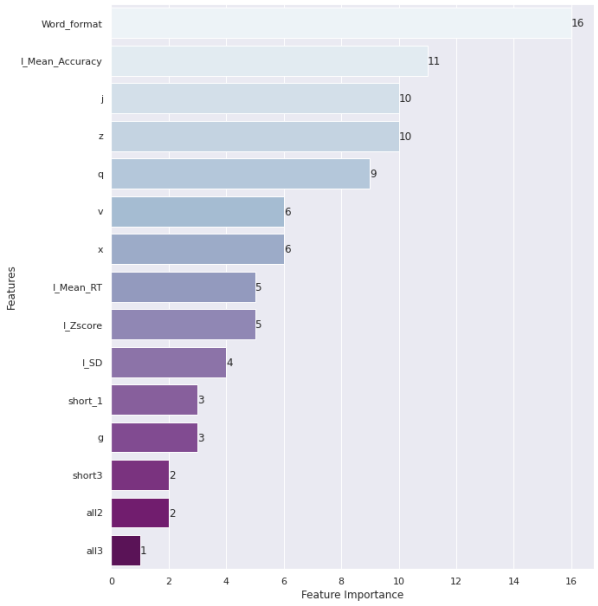

Figure 13: Feature importance for 140 days data.

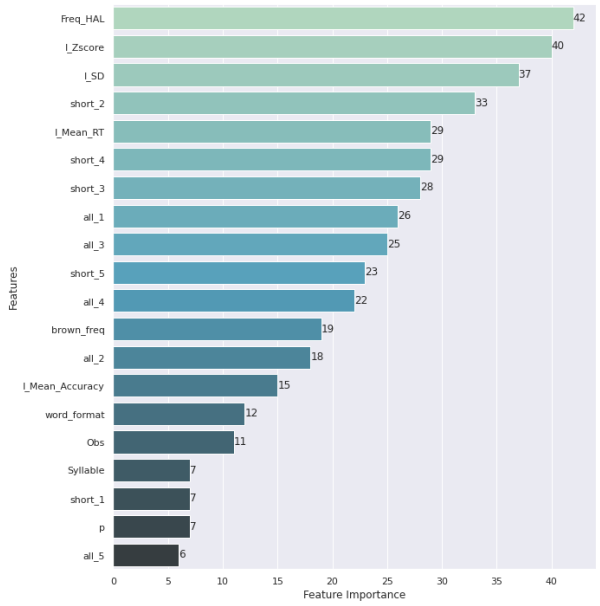

Figure 14: Feature importance for 360 days data.

LaTeX using the following notation:

Let $y_1, y_2, \ldots, y_n$ be a time series of length n, and let k be the window size for the moving average. Then the moving average at time t is given by:

$$\hat{y}_t = \frac{1}{k} \sum_{i=t-k+1}^{t} y_i \tag{11}$$

In this formula, $\hat{y}t$ represents the estimated value of the time series at time t, based on the average of the k preceding values $yt - k + 1, y_{t-k+2}, \ldots, y_t$. The symbol $\sum$ represents the sum operator, and the fraction $\frac{1}{k}$ is used to normalize the sum so that the moving average has the same scale as the original time series.

The formula for the weighted moving average is derived from the simple moving average formula by giving more weight to recent observations. Let $y_1, y_2, \ldots, y_n$ be a time series of length n, and let k be the window size for the moving average. Let $w_1, w_2, \ldots, w_k$ be the weights assigned to the most recent k observations, such that $w_1 + w_2 + \cdots + w_k = 1$ and $w_i \geq 0$ for $i = 1, 2, \ldots, k$. Then the weighted moving average at time t is given by:

$$\hat{y}_t = \sum_{i=t-k+1}^{t} w_{t-i+1} y_i \tag{12}$$

In this formula, $\hat{y}t$ represents the estimated value of the time series at time t, based on the weighted average of the most recent k observations. The symbol $\sum$ represents the sum operator, and $wt - i + 1$ represents the weight assigned to the i-th observation, with greater weight given to more recent observations.

Since the date, we wish to predict is 03/01/2023, the latest data we have is on 12/31/2022. There are 60 days in between. Therefore, our model should be able to predict 60 days later values. We make our training data to predict the distribution of tries 60 days later. Therefore, our weighted average needs to be shifted to 60 days. This means that our prediction will use the moving average from 60 days before not its current date moving average. We also add the day of the week and month of the date information to the dataset. In this way, we retrieve in total 82 time series features.

7.2 Models

With all these time series and word attribute information, we have 129 features in total. With similar reasons addressed in section 6.2. We still use XGBoost for such a smaller dataset and the ability to explain the importance of features better. In consideration of the small dataset and the distinct characteristics of 1 try, and 2 tries..., it will be difficult to predict all of them at the same time. We predict the distribution of tries separately to better predict each category more concisely, we use RMSE as a loss. We also gain more freedom for the model if we predict them separately. We could use different kinds of models and hyper-parameters. Therefore, our model can be understood as 7 separate models each predicting one of the values of the distribution. The separate models can of course have

some drawbacks compared to a single model that could predict the entire distribution at the same time. The separate models lack the ability to consider the prediction as a whole. We alleviate the problem by assigning weight to each of the models and making sure their prediction values sum to 100. To best analyze the importance of features, we still use XGBoost for all of them and present the result in the following section. The comparison for different models is as Table 4. Although Neural Network has lower training loss, it suffers from overfitting. So we still use XGBoost model.

7.3 Results and Analysis

The prediction of tries for the word "EERIE" on March 1, 2023, is The result of the distribution of the prediction is 0.47452372, 4.54758084, 15.71131084, 25.88720217, 28.99295704, 16.25635793, 8.13006746. To make it into the integer format as the provided data, we simply round these values. Therefore, the 1 try, 2 tries, ..., 7, and more tires for "EERIE" on March 1 are 0, 5, 16, 26, 29, 16, 8. The RMSE loss for both our training set and test set are close, we are confident that our prediction will be accurate. The uncertainty related to our prediction is some sudden events happened between 12/31/2022 and 03/01/2023. We don't have data on this interval, and some unexpected events could influence our prediction. However, based on the importance of features for 1,2,..., and 7 tries shown in Figure 16 and Figure 17, our model is more based on word attributes features instead of time series features, this makes it more robust to unexpected change happened during 2/31/2022 and 03/01/2023.

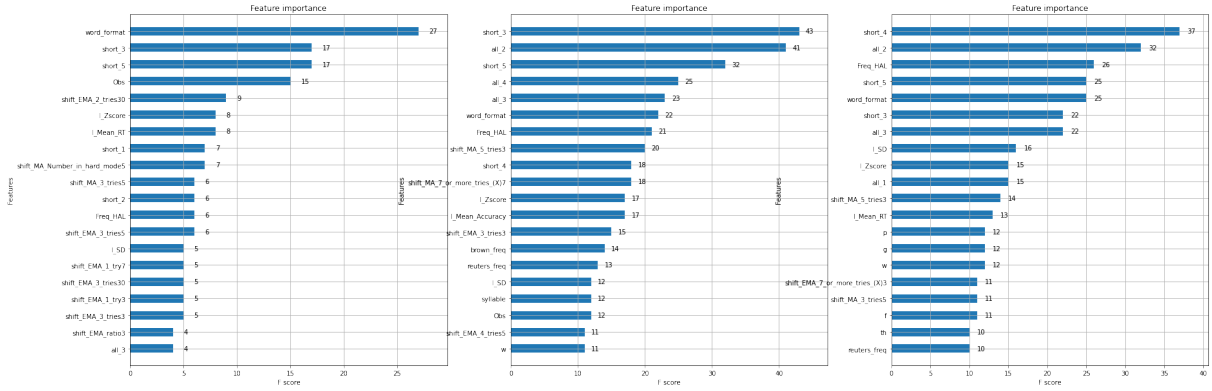

Figure 15: Feature importance for 1 try, 2 tries, and 3 tries

Table 4: Model comparison and best model selection question2

	model_type	loss	train_time
	Decision Tree	1.29408	30.08
	linear	1.0751	8.6
the best	XGBoost	0.95875	14.43
	Neural Network	0.922913	3.47
	Random Forest	0.965535	12.23

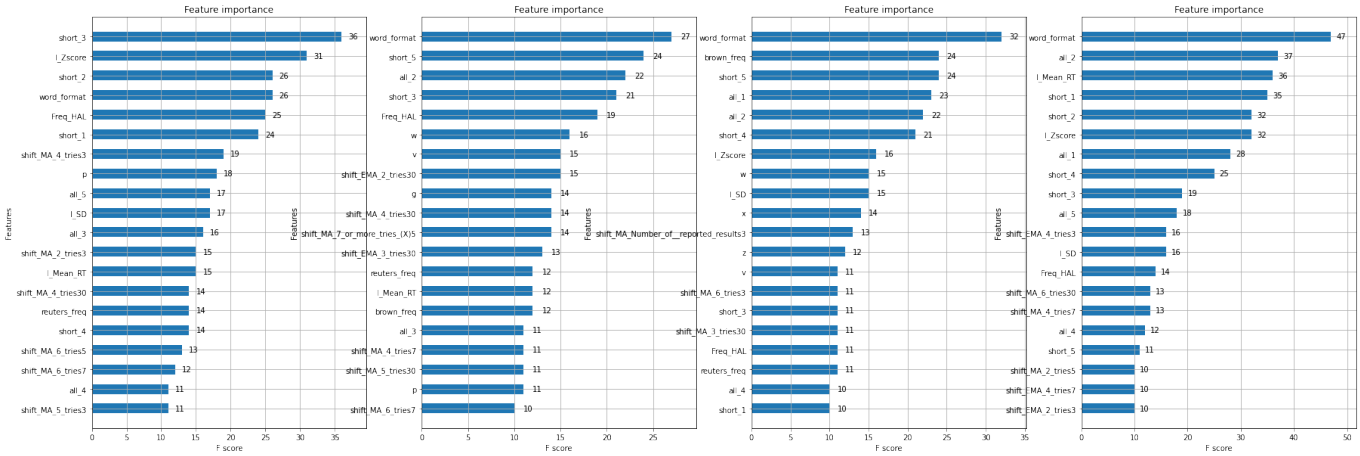

Figure 16: Feature importance for 4 tries, 5 tries, 6 tries and 7 and more tries

8 Natural Languages Processing (Question 3)

8.1 Prepossessing

The level of difficulty can be considered in two aspects: one is the subjective view from the players, and another is the objective results. The former one can use the percentage of hard mode to measure. We assume that if the players think the word is easy, they are more likely to play hard mode. Thus, the higher the percentage of hard mode, the easier the player thinks of the word. As for the objective results, we use the distribution of tries to measure. As we discussed in section 6.1, the percentage of hard mode is highly correlated with time. To classify the difficulty by the word itself, we wish the features we use to determine difficulty have a mere correlation with time. We also plot the number of tries over time, we don't see a clear trend of these values over time as Figure 17-23. Thus, we will use all these values but not the percentage of hard mode to analyze the difficulty of words.

For this question, we first use K-Means Clustering to determine the level of difficulty. Then we use our word attribute features to do classification.

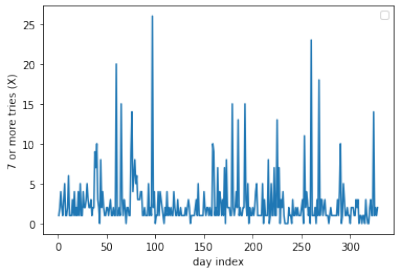

Figure 17: 7 try.

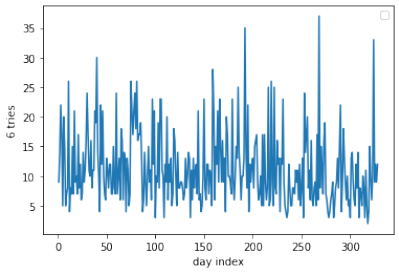

Figure 18: 6 tries.

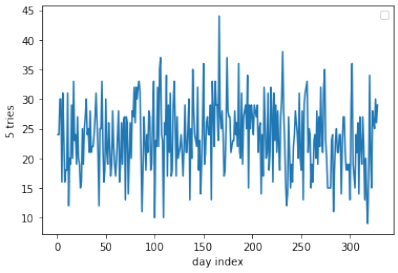

Figure 19: 5 tries.

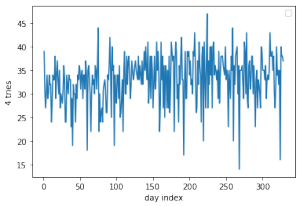

Figure 20: 4 tries.

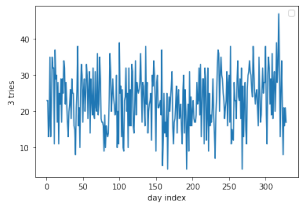

Figure 21: 3 tries.

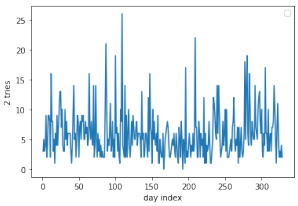

Figure 22: 2 tries.

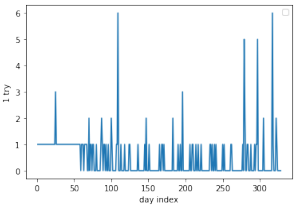

Figure 23: 1 try.

8.2 K-means Clustering for Determining Difficulty

This study uses the distribution of tries to define difficulty, and we cluster them by K-Means. First, it's necessary to determine the dimension on which data points are distributed. Instead of seven dimensions, we project the data point into three dimension space, including 2 tries and 3 tries as the first dimension, 4 tries as the second dimension, 5 tries, 6 tries, and 7 tries as the third dimension. This is because we examined the correlation matrix and discover that 1 try is determined as an outlier and does not display much correlation with the rest of the number of tries. 4 tries behave independently and it can be easily observed that 2 3, as well as 5 6 7 tries, can be separately grouped due to high correlation in the respective groups, referred to Figure 24.

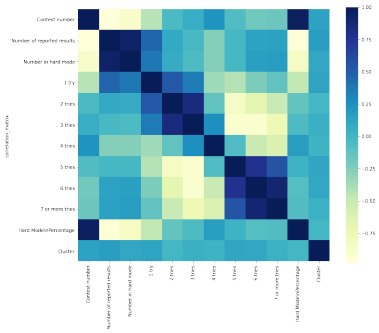

Figure 24: Correlation matrix between given information from Wordle data

Given a set of N data points, $X = x_1, x_2, ..., x_n$, and a pre-specified number of clusters, k, the k-means algorithm aims to partition the data into k clusters in a way that minimizes the sum of squared distances between data points and their assigned cluster centroids.

Initialization: Choose k initial centroids, $C = c_1, c_2, ..., c_k$, randomly from the data points in X. Here in our dataset we choose three centroids that correspond to easy, medium, and hard modes.

Assignment: Assign each data point x_i to the nearest centroid c_i in C, such that:

$$c_i = \underset{j \in \{1,2,...,k\}}{\arg\min} ||x_i - c_j||^2 \tag{13}$$

Update: Recalculate the centroids of the clusters using the mean of the data points

assigned to each cluster:

$$c_i = \frac{1}{N_i} \sum_{x_j \in C_i} x_j \tag{14}$$

for each (x,j) in cluster i, where Ni is the number of data points in cluster i.

Repeat steps 2 and 3 until convergence: Stop when the assignment of data points to clusters no longer changes, or when a pre-specified number of iterations is reached(See Figure 25).

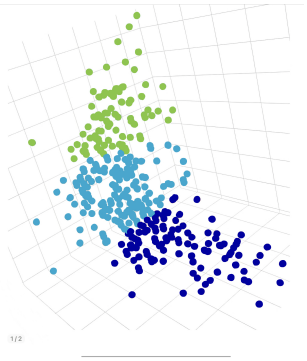

Figure 25: K-means clustering 3-D visualization

8.3 Model

With the same reasons for the easy explanation of features and the relatively small dataset, we still use the XGBoost model for classification. We also compare it with other models. The loss we use is softmax.

$$softmax = P(y = j|x) = \frac{e^{x^T w_j}}{\sum_{k=1}^{K} e^{x^T w_k}} \tag{15}$$

8.4 Result and Anaysis

The accuracy of the XGBoost model on the test set is 0.73 and 0.74 of accuracy on the training set. This model is consistent with the test set and training set. And it classifies "EERIE" as hard. In such a way, we are at least 73% confident in our prediction. If we see more closely at the word "EERIE," which has three repetition letters. For all three repetition letters, our model is able to predict the level of difficulty correctly. If we only consider such a small range, our model might reach an accuracy close to 100%.

The model comparison is as Table 5. Although neural networks have a low loss value, the test accuracy is only 54%. The result is consistent with our understanding that Tree-Based models performed the best and neural networks face the issue of overfitting.

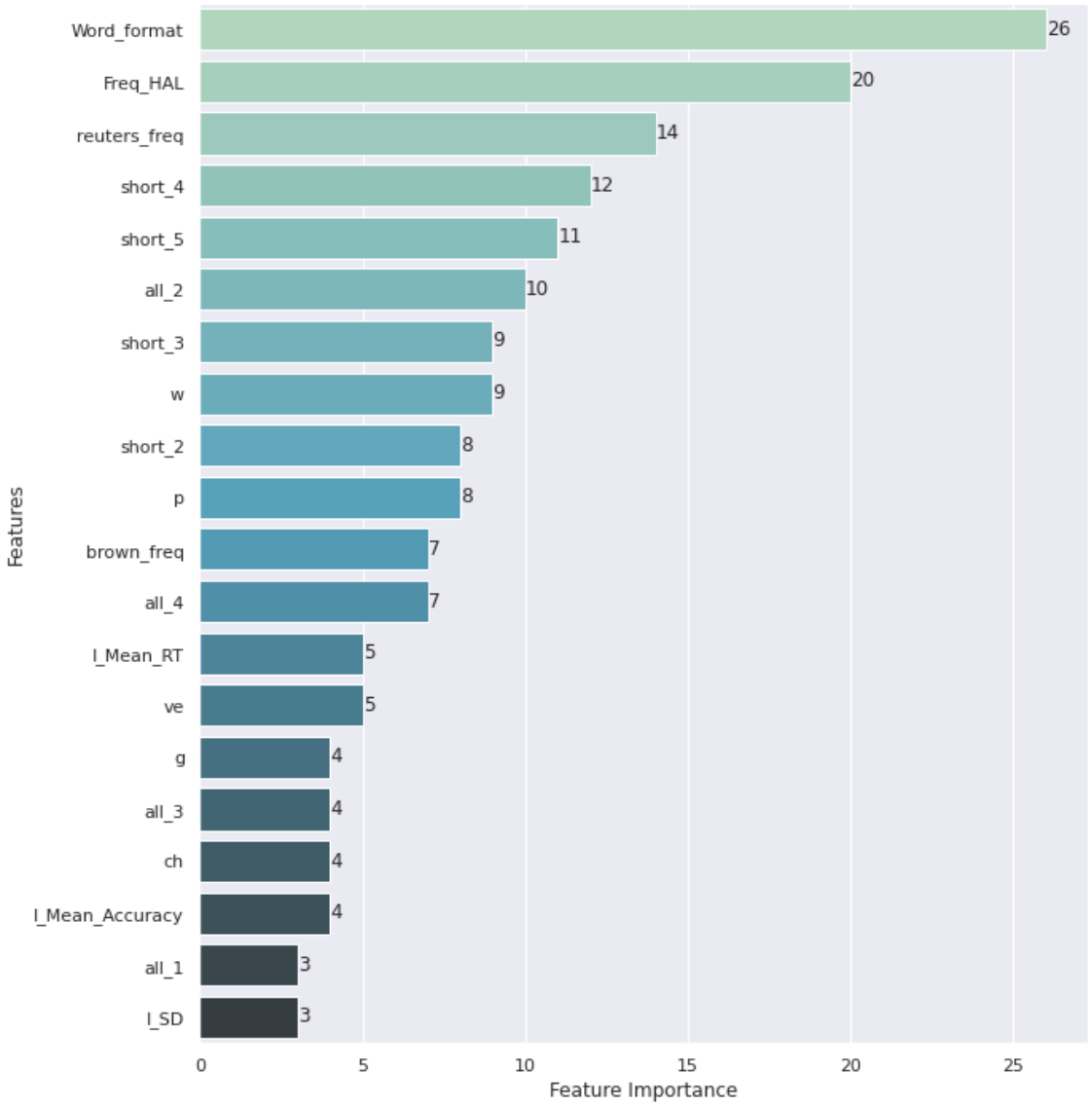

Figure 26: Feature Importance Rankings

Table 5: Model comparison and best model selection question3

	model_type	loss	train_time
	DecisionTree	1.01904	21.39
	linear	0.908394	6.45
the best	XGBoost	0.852454	6.07
	Neural Network	0.881989	2.45
	RandomForest	0.931719	8.91

9 Conclusions

We successfully collected, cleaned, transformed the dataset and find out many exciting parts about the dataset. We mainly use time series and natural language processing(NLP) methods to enrich our features.

Based on the prediction and limited dataset, we first pre-process the data through the time series method and get 82 time-related features. We add more features by analyzing the word by some NLP methods (textstat, nltk) to figure out 47 features that are related to the word; finally, we implement many machine learning models, including XGBoost, Random Forest, and Decision Tree to predict values of the word "EERIE" on March 1st, 2023. It turns out that our results are all consistent with our intuition.

We manage to use the distribution of tries to define difficulty and plot the number of tries over time; we don't see a clear trend of these values over time. Therefore, we will use all these values to determine the difficulty of words. After we define the difficulty, we use the same model XGBoost for classification since we wish to elaborate the word further on their attributes. The accuracy for clarification of our model is 73%.

10 Strengths and Weaknesses

10.1 Strengths

- **Accuracy.** The maximum RMSE of all our models in NLP implemented is 0.05, and the Accuracy for the model is above 70%.
- **Robustness.** Our models are highly consistent between the training set and the test set. The model retrieves features highly correlated to the word itself, which reduces the randomness caused by time.
- **Innovation.** We implement (XGBoost, GluonTS, XGBoost, Random Forest, Decision Tree, etc.) a hybrid model in the Time Series Prediction Model.
- **Generalization.** Our developed NLP and classification model are generalized enough for all other kinds of dataset of similar structure.
- **Well-formed features.** We thought about over 40 word-related features and over 80 time-related features.

10.2 Weaknesses

- **Too many features.** We have over 120 features, and it might make our models hard to catch the most significant features.
- **limited dataset.** Providing dataset length is only less than 400, the training set is not sufficient enough to train an existing model. Therefore, the outcome of our trained model may vary.

11 Letter to New York Times (See Page 22)

Puzzle Editors
The New York Times Building
620 Eighth Avenue
New York, NY 10018

Dear Editor of the New York Times,

We would like to express our sincere gratitude for the Wordle game. It is fun and challenging and helps people to enhance communication with friends and family.

After analyzing the game, we would like to take a moment to praise the team for the scientific and reasonable approach to the game's settings. Our data analysis found that the difficulty of the game, the distribution of scores, and the game settings are all in conformity with normality. This indicates that the game's design is thoroughly thought out and has been executed with precision. However, a few areas could be improved.

Firstly, from the data perspective, we discovered that the "Hard Mode Percentage" does not reflect the actual difficulty level of the game. The "Hard Mode" feature needs a few updates so that it functions properly and provides a genuine challenge for players. In other words. the game should provide real challenges for enthusiastic gamers.

To help address this, we suggest adopting the difficulty classification model that we have designed, quickly and accurately label each vocabulary. We recommend changing the current "One day, one word" to a "One day, three words" policy, with 3 separate words for 3 difficulty levels, making the game more enjoyable for new players and more challenging for more experienced ones. With this change, new players could quickly appreciate the intelligence of the game, while players who love a challenge could increase their sense of accomplishment. These changes could maintain the uniqueness of the game's features and enhance its inclusiveness. We also propose that the game adjusts the difficulty level dynamically according to the historical performance of the individual player. This will allow users to enjoy the game and feel a sense of accomplishment.

The popularity of the game Wordle peaked in mid-February 2022 during the pandemic yet has steadily declined since then. The game must change to maintain its popularity. To attract more players, we suggest adopting marketing strategies like combining the essential holidays in the world, using celebrities, and enhancing promotion ads. Introducing new and unique modes of play also helps, like a timed mode or advanced numbers of letters. These new modes of play would keep the game fresh and exciting for players, as well as add to the variety of the game.

Overall, We appreciate the effort and hard work that goes into creating such a game and look forward to seeing more innovative and engaging puzzles from the New York Times in the future.

Best regards,

References

[1] Benton J.A., Jesse G.M., Finding the optimal human strategy for Wordle using maximum correct letter probabilities and reinforcement learning. *Cornell University*, 2022.

[2] Zhang G P., Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 2003.

[3] Alexandrow A, Benidis K., GluonTS: Probabilistic and Neural Time Series Modeling in Python. *Journal of Machine Learning Research*, 2020.

[4] Box, G.E.P., Jenkins, G.M., Time Series Analysis: Forecasting and Control. *Holden-Day, San Francisco*, 1970.

[5] Cleveland, Robert B and Cleveland, William S and McRae, James E and Terpenning, Iii, STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics, vol.6, no.1*, 1990:3-73.

[6] Salinas D., Flunkert V., Jan G., DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Amazon Research*, 2020: 1181-1191.

[7] Renu B., Kathryn S.M., Danielle S.M., Applying Natural Language Processing and Hierarchical Machine Learning Approaches to Text Difficulty Classification. *International Artificial Intelligence in Education Society*, 2020: 337-370

[8] Bird, Steven and Klein, Ewan and Loper, Edward, Natural Language Toolkit. *https://www.nltk.org/*, 2009.

[9] Levenshtein, Vladimir I, Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics doklady, vol.10*, 1966:707-710.

[10] Bartosz T., Jakub C., Przemyslaw B., Binary codes mljar-supervised: Automated Machine Learning Framework. *The Journal of Open Source Software, vol.5, no.54*, 2020:2593.

[11] Shardlow, Matthew, A simple word difficulty metric which allows for incomplete assessment. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014:778-786.

[12] Bloomfield, Peter, Exponential data smoothing by moving averages. *Biometrics*, 1972:429–435.