Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2320341

Unlocking the Secrets of Wordle: Predicting Game Difficulty and User Experience

Summary

Wordle is a popular online game that has gained widespread popularity in recent years. It is a word-guessing game in which players must correctly guess a five-letter word chosen by the game's computer. The player has six chances to guess the word, and with each guess, the game provides feedback by indicating how many letters of the guessed word match the secret word and are in the correct position.

In this paper, we present the results of our analysis of various aspects related to the game of Wordle. We developed prediction models for game difficulty and user experience predictions. Our models accurately predicted the difficulty level and reported the scores distribution of Wordle words using various machine learning algorithms.

To achieve these results, we derived a **Long Short-term Memory Model (LSTM)** for predicting the interval of the number of reported results on a given date by feeding normalized average reported result numbers within a 7-Day **Rolling Window**. We found that the confidence interval of the number of reported results on March 1, 2023 is in the range of $[22537, 23041]$.

Then, we extracted all possible Wordle attributes and tested their correlation with the percentage of scores reported that were played in Hard Mode with **heatmap**, **ADF Test**, and **Sample T-Test**. Our findings indicate that there are no attributes of Wordle that affect the percentage of scores reported that were played in Hard Mode.

To address the problem of predicting reported scores distribution of a given Wordle, we applied **Multi-Output Regression Chain model** with **Multi-layer Perceptron Regression**, **Decision Trees Regression**, and **Random Forest Regression** to Wordle attributes to explore the specific relationship between Wordle attributes and reported scores distributions. According to the result obtained from **Multi-layer Preceptron** and **PSO Algorithm**, our predictions indicate that the distribution of reported scores for the word "EERIE" will be [0.581, 8.08, 24.729, 31.813, 22.546, 10.162, 1.706]. This means that we estimate 0.581% of players will pass on their first try, 8.08% on their second try, 24.729% on their third try, and so on.

We also used **K-means Clustering** and **RSR** models to build clusters for classifying difficulty levels, and a **CNN** model to predict which difficulty level the Wordle lies in based on its attributes. In this paper, we classify the data set into 5 clusters (k=5). Predictions show that wordle "EERIE" lies in the difficulty level 3 out of 5, which is a medium difficulty level.

Our study's findings have interesting implications for the future development of Wordle and the use of machine learning algorithms in predicting game outcomes. Based on our results, we have written a letter to the Game Editors of the New York Times to explain our findings and highlight interesting features found in Wordle data.

Finally, we conducted a **sensitivity analysis** to verify our models' robustness and result adaptability, demonstrating that our models are accurate, consistent with reality, effective, and practical for predicting future Wordle outcomes.

Keywords: LSTM; Multi-Output Regression; K-means; CNN; PSO

Contents

1 Introduction

1.1 Problem Background

The Wordle puzzle is a popular game offered daily by the *New York Times*, where players attempt to guess a five-letter word in six or fewer tries. Each guess must be an English word, and the game provides feedback for each guess, indicating whether the guessed letters are in the correct location (green), in the word but in the wrong location (yellow), or not in the word at all (gray). The game can be played in regular mode or Hard Mode, with the latter requiring players to use correct letters (yellow or green) in subsequent guesses.

1.2 Restatement of the problem

We are given a data set of daily results from January 7, 2022, through December 31, 2022, which includes the date, contest number, word of the day, number of players, and the percentage of players who guessed the word in each number of tries.

- Develop prediction models to create a prediction interval for the number of reported results on March 1, 2023.
- Determine whether any attribute of Wordle affect the percentage of scores reported that were played in Hard Mode.
- Develop prediction models towards the distribution of reported results to create a specific prediction about the distribution of reported result for the word “EERIE” on March 1, 2023.
- Develop a model classifying solution words by difficulty and determine the difficulty level of Wordle “EERIE”.

1.3 Assumptions

We make the following main assumptions to simplify our model and eliminate the complexity:

- It is assumed that there is homogeneity in players’ familiarity levels with the Wordle game.
- It is assumed that players will consistently and automatically upload their scores post-game, resulting in the retention of all scores.
- It is assumed that players will not employ external resources such as algorithms or illicitly accessing answers while playing Wordle.

1.4 Workflow

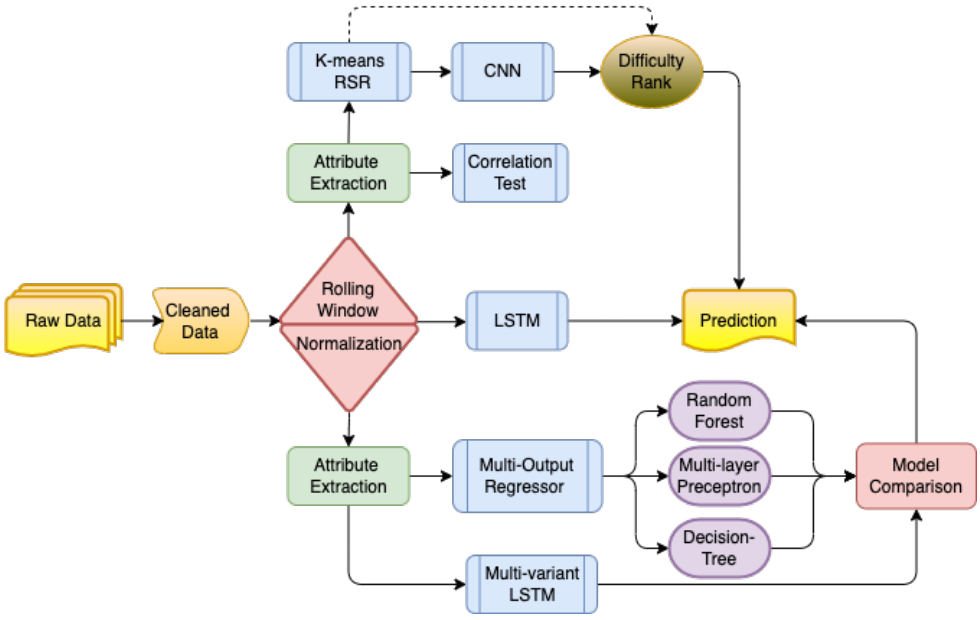

Figure 1: Work Flow

1.5 Notations

Symbol	Definition
C_t	Cell state
U_t	Update filter
O_t	Cell state that is going to output
h_t	Hidden state to be passed on next cell
W_t	Rolling window at time t
g	Activation function
C_i	Cluster i
W	Training input of MLP
b	MLP model parameter
Q_m	Data at node m of DT
t_m	Threshold at node m of DT

Table 1: Symbol Description

2 Data Processing

To commence our analysis, we first pre-processed the given dataset. Upon examining the Wordle data, we discovered several blank rows at the end of the dataset and removed them. A thorough analysis

of the data revealed outliers in the 'Number of reported results' column. We mitigated their impact on our analysis by manually modifying the data to the lower bound ($Q_1 - 1.5 * \text{IQR}$), as recommended by standard statistical practices. By conducting these pre-processing steps, we were able to reduce the dataset to 360 rows, enabling us to conduct our subsequent analyses with confidence in the accuracy and reliability of the data.

To enhance our understanding of the word structure, we have incorporated our analysis results into the dataset. The detailed information about the added columns can be found in Table 2 below. The methods used to derive each data will be introduced in subsequent sections when the variables are utilized.

Variable Name	Description	Possible Value
a	The number of letter 'a' appearing in the word	0, 1, 2, 3, 4, 5
...	...	0, 1, 2, 3, 4, 5
z	The number of letter 'z' appearing in the word	0, 1, 2, 3, 4, 5
vowel_number	The number of vowel letter appearing in the word	0, 1, 2, 3, 4, 5
consonant_number	The number of consonant letter appearing in the word	0, 1, 2, 3, 4, 5
vowel_rate	The number of vowel / Wordle length	(0, 1)
word_vc_structure	The pattern of vowels and consonants in the word	1: "CVCVC" 2: "CVCCV" 3: "VCVCV" 4: "VCCVC" 5: "Other Form"
max_repeat	The maximum number of the same letter appearing in the word	1, 2, 3, 4, 5
familiarity	People's familiarity with the word	(0, 1)
Percentage	The number of Hard Mode / The number of Reported Results	(0, 1)

Table 2: Variable Description

To facilitate the comparison and evaluation of the strengths and weaknesses of our models, we opted to utilize the first 80% of the data as the training dataset and the remaining 20% as the testing dataset. This allowed us to visualize and compare the predicted values against the true values effectively.

3 Predict Reported Result Number Based on Long Short-term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is designed to capture long-term dependencies [10]. The architecture of LSTM typically comprises multiple memory blocks, referred to as cells, which are connected through multiple layers. The cells contain gates that regulate the information stored in the cell and hidden states using activation functions such as sigmoid and tanh. Specifically, the gates take the hidden states from the previous step h_{t-1} and the current input

x_t and perform element-wise multiplication with weight matrices W, followed by the addition of a bias b.

The utilization of gates enables the LSTM to selectively retain or forget information from previous time steps, which is critical for modeling long-term dependencies. Furthermore, the activation functions of the gates are designed to mitigate the vanishing gradient problem that can impede the training of deep RNNs. Thus, the LSTM architecture is well-suited for time series data analysis, and we leverage this framework to predict the number of reported results on March 1, 2023.

3.1 Model Introduction

There are three primary gates in an LSTM model, each utilizing a different activation function:

- Forget Gate: This gate determines which information to discard from the memory cell.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

- Input Gate: This gate decides which parts of the input should be used to update the memory state:

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$
$$U_t = \sigma(W_u[h_{t-1}, x_t] + b_u)$$
$$C_t = f_t \cdot C_{t-1} + U_t \cdot \hat{C}_t$$

- Output Gate: This gate determines the output value based on both the input and memory state.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = O_t \cdot \tanh(C_t)$$

In the above formulas, C_t represents the cell state, U_t refers to the updated filter, O_t is the cell state that is going to output, and h_t stands for the hidden state that will be passed on to the next cell [12].

3.2 Model Adjustment

3.2.1 Rolling Window

In this study, we introduce a rolling window approach to capture temporal dependencies, trend, and seasonality in time series data to make accurate predictions for future values [9]. The rolling window is defined as a vector of length n containing the values of the time series for the n previous time-steps, up to and including time t. We shift the rolling window over time using a step size s to capture patterns and changes in the data. For example, the rolling window with window size n and step size s at time t is defined as [4]:

$$W_t = [y_t, y_{t-1}, ..., y_{t-n+1}]$$
$$W_{t-s} = [y_{t-s}, y_{t-s-1}, ..., y_{t-s-n+1}]$$
$$W_{t-2s} = [y_{t-2s}, y_{t-2s-1}, ..., y_{t-2s-n+1}]$$

Here, W_{t-s} and W_{t-2s} represent the rolling windows at time $t - s$ and $t - 2s$, respectively. For our dataset, which consists of daily time series data, we choose a rolling window size of 7 to capture a week's worth of data at a time as shown in Figure 2. By doing so, we can capture weekly patterns and smooth out noise at the same time, leading to more accurate predictions.

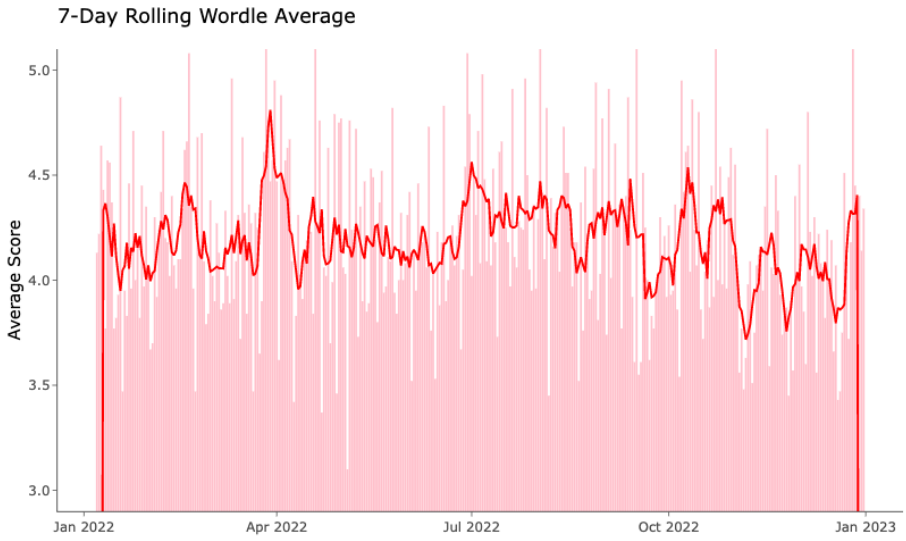

Figure 2: 7-Day Rolling Window

3.2.2 Normalization

Before using the data for training the LSTM model, it is important to normalize the data to improve the efficiency and accuracy of the model. In this study, we used the MinMax Scalar method to scale numerical features between 0 and 1. Normalizing the data helps in faster convergence of the model by speeding up the gradient descent method that is used to minimize the loss function. Additionally, normalization can also help in preventing certain features from dominating others in the model, thus improving the accuracy. The MinMax Scalar method is used to normalize the data as follows: for each feature, we subtract the minimum value from the feature and then divide the result by the range of the feature. The resulting value is a normalized value between 0 and 1. The formula used for MinMax Scalar normalization is as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}.$$

In time series analysis, it can be challenging to perform cross-validation due to the sequential nature of the data. Selecting random samples and assigning them to either the test or train set is not feasible, as it may result in future-looking bias, which is not desirable when training a predictive model. Instead, a common approach is to use a fixed window of time for training and testing. In this study, we employed a rolling window of size 7 to capture weekly patterns and smooth out noise. We used the MinMax Scalar method to normalize the original data, which helped improve the convergence of the gradient descent algorithm during training. To prevent overfitting, we selected a small number of hidden layers (4) for the LSTM model, given the low dimensionality of the input data. We chose a batch size of 1,

which is appropriate for the small sample size in this study. The time step, which represents the lag length between the training and test sets, was set to 7 days. These hyperparameter choices were made based on prior research and experience in time series analysis, and were evaluated using appropriate performance metrics.

3.3 Model Outcome

Figure 3 shows the learning history of of our LSTM model. The loss is minimized and stays stable after 40 epoches of training in both training and testing set. The good performance in training and testing leads to comparatively accurate and robust prediction result. Based on the information presented in Figure 4, it can be observed that the prediction interval for the number of reported results on March 1, 2023 is [22573, 23041], with a corresponding prediction value of 22807 and a 95% confidence interval of 234.

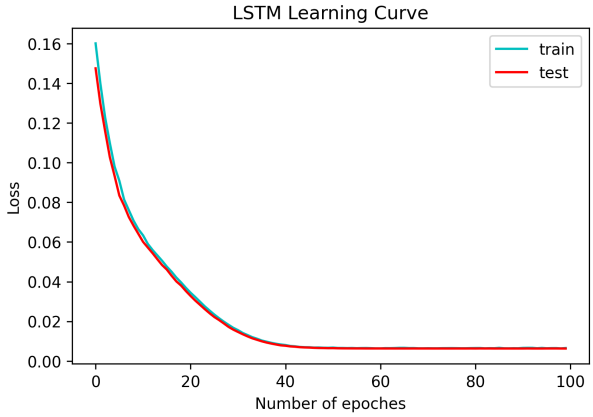

Figure 3: LSTM Model Learning Curve

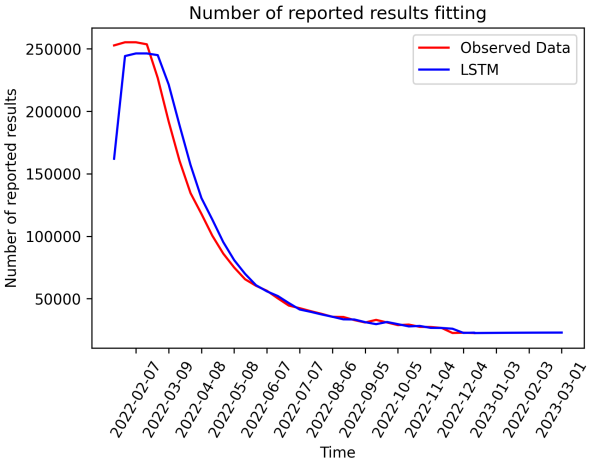

Figure 4: LSTM Model Prediction

Several metrics were utilized to evaluate the performance of the Long Short-Term Memory (LSTM) model, including the root mean squared error (RMSE), mean absolute error (MAE), and coefficient

of determination (R^2). The RMSE of the model was calculated to be 18452.979, indicating that the model's predictions were relatively accurate. The MAE was also computed, resulting in an average error of 8032.43 units and indicating the model's ability to make predictions with moderate accuracy.

The R^2 value, which measures the strength of correlation between predicted and actual values, was also determined to be 0.942, indicating a strong correlation between the model's predictions and the actual values. To further assess the model's performance, a graphical representation of the actual and predicted values was generated in Figure 4. The resulting graph showed that the model was able to capture the overall trend of the data, including the peaks and valleys, as well as the general shape of the data.

In summary, the LSTM model demonstrated relatively accurate predictions for the time series data with an RMSE of 18452.979, a MAE of 8032.43, and an R^2 value of 0.942. However, further model improvements may be necessary to account for extreme events and improve prediction accuracy.

MAPE	RMSE	MAE	R^2
6.745948642492294	18452.979	8032.43	0.9420348362090252

Table 3: Model Summary

4 Wordle Attribute Analysis

4.1 Attribute Extraction

When analyzing the structure of Wordle, vowel rate emerges as a crucial characteristic to consider, as it offers insight into the phonological and morphological properties of the Wordle [6]. Vowels, which convey the most information in a Wordle, have a significant impact on the comprehension of the Wordle. Their presence, absence, and distribution affect how well the Wordle is understood. The vowel-consonant structure of a Wordle further assists in understanding its syllabic structure and distinguishing it from other Wordles. Additionally, we introduce familiarity as a significant factor in Wordle recognition. Familiar Wordles are more comfortable to understand and process than unfamiliar ones. To provide additional information that can help individuals distinguish a Wordle from others, we analyze the maximum repeat time for a Wordle, taking hints from the results of each try.

To calculate vowel rate, we first count the number of each letter in an individual Wordle. From this count, we determine the number of vowels and consonants in each Wordle and derive the corresponding vowel rate. We categorize the vowel-consonant structure patterns into five types, namely, "CVCVC", "CVCCV", "VCVCV", "VCCVC" and "Other Forms". These patterns are labeled from 1 to 5 to facilitate data analysis. To assess the familiarity of each Wordle, we used word frequency data from the "nltk" packages, assuming that people will be more familiar with more frequently appearing words.

4.2 Correlation Analysis

We conducted a correlation analysis on five attributes of Wordle, and the results from Figure 5 showed that there is no strong correlation between any two attributes. The highest correlation coefficient of 0.31 was observed between vowel rate and vowel-consonant structure, indicating a moderate positive correlation. The remaining attributes, including percentage and familiarity, were found to be weakly

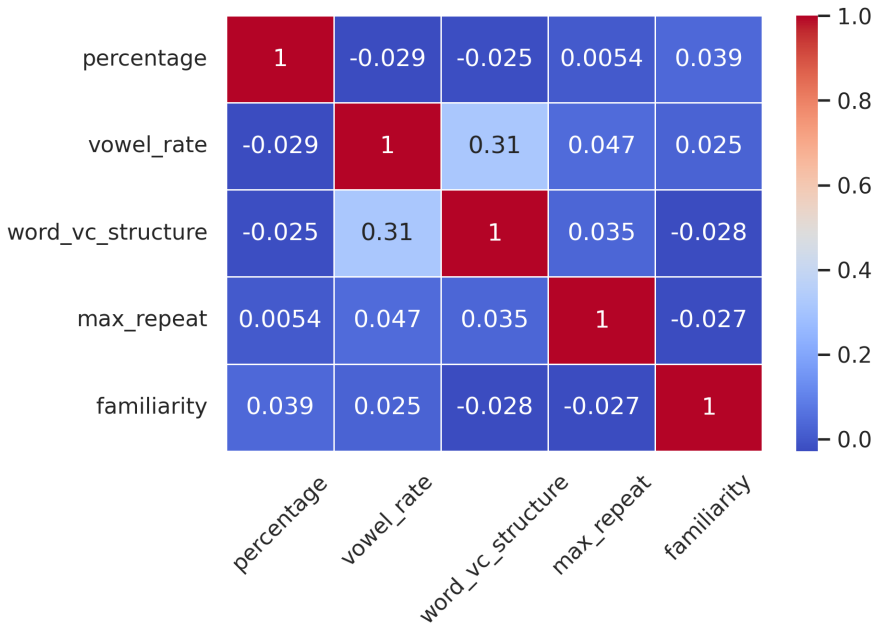

Figure 5: Pentagonal Wordle Heatmap

or not significantly correlated, with correlation coefficients below 0.1. Based on these findings, we conclude that the attributes are relatively independent of each other, as evidenced by the lack of significant correlation between them.

ADF Test							
Variable	Difference Order	t	P	AIC	Critical Value		
					1%	5%	10%
hard pecent	0	-3.738	0.004***	-1088.885	-3.449	-2.87	-2.571
	1	-9.599	0.000***	-1080.933	-3.449	-2.87	-2.571
	2	-9.502	0.000***	-1024.115	-3.45	-2.87	-2.571
Note: ***, **, * denote 1%, 5%, 10% significance level							

Table 4: ADF Test Result

Table 4 shows the result of Augmented Dickey-Fuller (ADF) test conducted on the variable "hard percent" with difference orders. The test is used to determine whether a time series is stationary or not. A stationary time series has constant mean and variance over time, which is necessary for many time series models. The table shows that for all three difference orders, the t-statistic is negative and significant at the 1% level, indicating that we can reject the null hypothesis of a unit root and conclude that the time series is stationary. And the critical values at the 1%,5% and 10% significance levels are also reported.

The one-sample t-test compares the mean of a sample to a specified test value. According to the result from Table 5, the test value is 0.078 with the sample size of 359. The average (AVG) of the sample is also 0.078, with a standard deviation of 0.051. The calculated t-statistic is 0, and the corresponding

Test Value	Size	AVG	SD	t	P
0.078	359	0.078	0.051	0	1.000
Note:***, **, * denote1%, 5%, 10% significance level					

Table 5: One-Sample T-Test Result

p-value is 1. Since p-value is clearly greater than 0.05, we can not reject the null hypothesis that the true population mean is equal to the test value of 0.078. This suggests that there is not enough evidence to support the claim that the sample mean is significantly different from the test value.

Thus, based on the analysis of coefficient above, there is not any attribute of Wordle will affect the percentage of scores reported that were played in Hard Mode.

5 Predict Distribution of Reported Scores based on Multi-Output Regression

5.1 Model Introduction

To provide a more comprehensive analysis of model selection for the Wordle dataset, we investigated Multioutput Regression as a potential approach. Multioutput Regression is a method that predicts multiple numerical properties for each sample, with each property represented as a numerical variable [2]. In the context of Wordle, we used the characteristics of words as input to predict the percentage of each number of trials (1, 2, 3, 4, 5, 6, X) as the output. We explored three different algorithms for Multioutput Regression: Multi-layer Perceptron Regression, Decision Tree Regression, and Random Forest Regression.

5.2 Model Adjustment

While Multioutput Regression with different algorithms allows for the learning of non-linear functions that map inputs to outputs, it may not capture the complex interdependencies between input variables. To address this limitation, we utilized Multioutput Regression Chain. This approach improves the accuracy of prediction by capturing more of the underlying structure of the data through a chain structure that uses the predictions of previous regressors as input features for the next regressor. The efficacy of this approach has been demonstrated in previous studies [7].

Multi-layer Preceptron Regression

Multi-layer Preceptron (MLP) is a supervised learning algorithm that can learn a function training on a dataset. For example, given the training input as $X = x_1, x_2, \cdots, x_m$ and a output Y, it can learn a non-linear function approximator that maps X to Y. The main theme of MLP is that between the input layer and the output layer, there can be one or more non-linear layers called hidden layers. In the context of regression, given a set of training examples $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ where $x_i \in \mathbb{R}^n$

and $y_i \in [0, 1]$, a MLP regressor with one hidden layer one hidden neuron learns the function

$$f(x) = W_2 g(W_1^T x + b_1) + b_2,$$

where $W_1 \in \mathbb{R}^m$ and $W_2, b_1, b_2 \in \mathbb{R}$ are model parameters and output activation function g is the identical function. The MLP regressor applies Mean Squared Error loss function, which is

$$Loss(\hat{y}, y, W) = \frac{1}{2n} \sum_{i=0}^{n} ||\hat{y}_i - y_i||_2^2 + \frac{\alpha}{2n} ||W||_2^2,$$

where α is a non-negative hyperparameter that controls the magnitude of the penalty [8].

Moreover, to enhance the accuracy and generalization ability of the MLP model, we introduce Particle Swarm Optimization (PSO) here to help us find the best combination of hyperparameters that maximize the accuracy of the MLP model on the given data set.

Decision Trees Regression

Decision Tree (DT) is an unsupervised learning method that make prediction of a target variable by learning simple decision rules inferred from the data features. In the context of regression, given training vectors $x_i \in \mathbb{R}^n, i = 1, \cdots, l$ and a output vector $y \in \mathbb{R}^l$, a decision tree recursively partitions the feature of training sample to let samples with same or similar output values are grounded together.

Let data at node m be represented by Q_m with n_m number of samples. For each candidate split $\theta = (j, t_m)$, where j represents feature and t_m represents the threshold at node m. The DT algorithm partitions the data into $Q_m^{\text{left}}(\theta)$ and $Q_m^{\text{right}}(\theta)$ subsets, where

$$Q_m^{\text{left}}(\theta) = \{(x, y) | x_j \leq t_m\}$$
$$Q_m^{\text{right}}(\theta) = Q_m \backslash Q_m^{\text{left}}(\theta)$$

Then, the quality of a candidate split at node m is measured by the formula

$$G(Q_m, \theta) = \frac{n_m^{\text{left}}}{n_m} H(Q_m^{\text{left}}(\theta)) + \frac{n_m^{\text{right}}}{n_m} H(Q_m^{\text{right}}(\theta)),$$

and we select the parameters that minimizes G, which is $\theta^* = \text{argmin}_\theta G(Q_m, \theta)$. Function H represents the Mean Squared Error at node m, which is defined as

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2,$$

where $\bar{y}_m = \frac{1}{n_m} \sum_{y \in Q_m} y$ [11].

Random Forest Regression

Random Forest (RF) algorithm is an extension of DT method [3]. First, a bootstrap sample is randomly drawn from the original data set to grow a decision tree. Second, a randomly selected subset of variables is chosen as candidate variables for splitting at each node of the decision tree.Averaging over trees, due to the randomization used in growing the trees, the model could approximate rich classes of functions with low generalization error.

5.3 Model Outcome

Model	Tag	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	X
MLP	train	1.37	0.724	0.22	0.114	0.148	0.398	0.998
	test	1.421	0.424	0.194	0.124	0.194	0.484	1.079
DT	train	1.678	0.958	0.217	0.115	0.158	0.367	0.777
	test	1.088	0.678	0.224	0.122	0.177	0.392	0.757
RF	train	0.722	0.169	0.12	0.075	0.091	0.255	0.68
	test	1.448	0.548	0.267	0.118	0.233	0.509	1.076

Table 6: Multi-Output Regression Summary

Table 6 provides the performance of three regression models, Multi-layer Perceptron Regression (MLP), Decision Tree Regression (DT), and Random Forest Regression (RF), on both the training and test datasets. Each row corresponds to a specific model, and each column represents the Mean Absolute Error (MAE) of the model's prediction for each of the seven possible scores in the Wordle game (1-6 and X).

From the table, we can see that the performance of the three models varies across the different number of tries. In general, the MLP and DT models perform similarly, while the RF model has the lowest MAE on the training dataset across all number of tries. However, on the test dataset, the RF model generally has a higher MAE than the MLP and DT models, indicating that it is overfitting to the training data.

Additionally, we can see that as the number of tries increases, the MAE generally decreases for all models on both the training and test datasets. This may be due to the fact that as the number of tries increases, the distribution of scores becomes more spread out and therefore easier to predict.

Model	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	X
MLP	0.581	8.08	24.729	31.813	22.546	10.162	1.706
DT	1	10	26	32	21	9	1
RF	0.23	3.513	18.326	31.263	27.5	15.751	2.861

Table 7: Multi-Output Regression Results

It is worth noting that by introducing the PSO algorithm to optimize the hyperparameters of the MLP model, we were able to achieve better performance compared to the MLP model alone. PSO algorithm helps to find optimal weights and biases values for the MLP model, which leads to better accuracy and precision in the prediction of the distribution of reported scores. Therefore, the combination of MLP model and PSO algorithm shows great potential in improving the accuracy and robustness of multioutput regression tasks like this. From Table 7, we can conclude that based on MLP model and PSO algorithm, our prediction for the distribution of reported scores for Wordle "EERIE" is [0.581, 8.08, 24.729, 31.813, 22.546, 10.162, 1.706]. This distribution suggests that most players will complete the game in 4 or 5 tries, with a smaller number of players requiring 3 or 6 tries. The prediction can

be useful for game designers to understand the difficulty level of the game and to adjust the game play accordingly.

5.4 Model Comparison with Multi-variant LSTM

We utilized Long short-term memory (LSTM) architecture to predict the distribution of reported scores for a future date in the context of Wordle. LSTM has been widely recognized as a powerful sequence modeling tool, particularly for tasks involving time series data analysis and prediction. To accomplish this task, we introduced a multi-variant LSTM model that learns from both the distributions over time and the characteristics of Wordles.

This approach involves the use of LSTM to learn the temporal dependencies in the reported scores distribution over time, as well as the features of Wordle words that may influence these distributions. By incorporating these variables into the model, we aim to improve the accuracy of our predictions of future score distributions.

LSTM Data Preparation

In this study, we approach the supervised learning problem of predicting the distribution of reported scores using a combination of Wordle word characteristics and the distribution of reported scores at the prior time step. To achieve this, we define several characteristics of the Wordle word, including the number of letters, number of vowels and consonants, rate of vowels, word structure, number of maximum repeat letters, and word familiarity. We then concatenate the distribution of reported scores at the prior time step with the characteristics of the Wordle word to create an input sample. Specifically, for a given date t, the input has a shape of $(1, 40)$, where the first 7 values indicate the distribution of reported scores on date $t - 1$, and the following 33 values represent the word characteristics. The output has a shape of $(1, 7)$, which is the distribution of reported scores on date t. Prior to training and testing, we apply normalization using the MinMax Scalar method to the distribution of reported scores to accelerate the optimization process and prevent certain features from dominating others in the model.

5.4.1 Model Adjustment

Our model follows a general LSTM architecture, which contains a forget gate, an input gate and an output gate. The forget gate is identical to the one we mentioned previously. In order to prevent the existence of negative numbers in the final output, instead of using the default activation function tanh, we replaced it with a rectified linear unit (ReLU) activation. The RELU activation function maps negative numbers to 0 and others to themselves. Thus, we mathematical formula for input gate and output gate becomes:

- Input Gate:

$$\hat{C}_t = \text{ReLU}(W_c[h_{t-1}, x_t] + b_c)$$
$$U_t = \sigma(W_u[h_{t-1}, x_t] + b_u)$$
$$C_t = f_t \cdot C_{t-1} + U_t \cdot \hat{C}_t$$

- Output Gate:

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = O_t \cdot \text{ReLU}(C_t)$$

Our model has several hyperparameters that require careful consideration. We set the total number of epochs to 50, which determines the total number of forward and backward propagation iterations during training. We choose 32 as the number of hidden layers in the neural network to avoid overfitting, given the relatively low dimensionality of our input data. The batch size is the number of training samples used in a single forward or backward propagation before updating the weights. We choose 5 as the batch size, which is a common factor of the training set (320 samples) and the test set (35 samples). We also experiment with a batch size of 1, but the training results show little difference compared to a batch size of 5. For the time step, which represents the lag length between the training and test sets, we select a value of 1 day, indicating that we consider the overall data with a lag of 1 day. We choose mean absolute error (MAE) as the loss function to be minimized during training. This decision is based on the observation that there are no significant outliers in the training data. MAE can increase the robustness of the model compared to other loss functions, such as mean squared error. To prevent overfitting, we set the dropout rate to 0.2 and the recurrent dropout rate to 0.3. These values are carefully chosen to balance the model's ability to learn from the training data while also generalizing well to unseen data.

5.4.2 Model Outcome

Figure 6 shows the training history of our multi-variant LSTM model. We observe that there is an obvious trend of overfitting: as the training epoch increase, the training loss keeps decreasing but the testing loss stays the same.

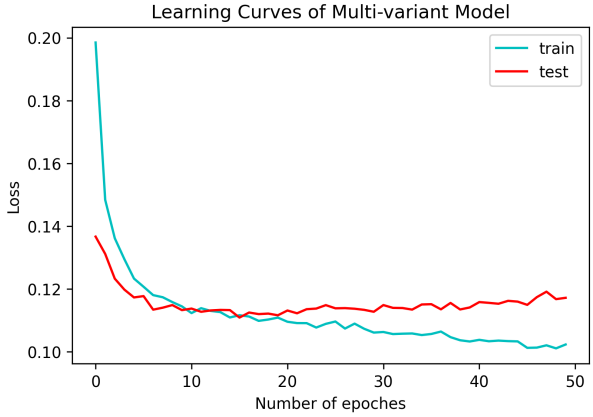

Figure 6: Multi-variant LSTM Model Prediction

The model's MAPE (Mean Absolute Percentage Error) of 0.022 indicates that on average, the model's predictions are within 2.2% of the actual values, which is relatively low and suggests that the model is performing well in terms of accuracy. The RMSE (Root Mean Squared Error) of 0.175 suggests that the model's predictions have an average deviation of 0.175 from the actual values, which

is not too high considering the scale of the target variable. The MAE (Mean Absolute Error) of 0.118 is also relatively low, suggesting that the model's predictions are close to the actual values.

However, the R-squared value of -0.22 suggests that the model's performance is poor in terms of explaining the variance in the target variable using the predictor variables. This indicates that there may be other variables that are important in predicting the target variable that are not included in the model, or that the model is not capturing the relationship between the variables effectively. So, based on the parameters from Table 8, we can not accept the prediction result of the model.

MAPE	RMSE	MAE	R^2
0.022213453	0.17509325	0.11785471	-0.21952703091487963

Table 8: Multi-variant LSTM Model Summary

Reported Scores	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	X
Probability	0.46	5.95	23.07	33.56	24.1	11.77	2.83

Table 9: Prediction Result

Limitation

While Multi-variant LSTM is a good choice to model this type of question, it does not work in this specific task. We noticed an obvious trend of overfitting, even applying with normalization and dropout. When making a prediction, the distribution of reported scores for a previous day is required, which makes the prediction of distribution far away from the data less reliable. Also, one possible reason that leads to this result is that Multi-variant LSTM model can only produce one output. In other words, it is not efficient on predicting the distribution of reported scores .

6 Cluster Wordle by Difficulty Level based on K-means

6.1 Introduction of Model

K-means is a commonly used clustering algorithm in data mining that is used to cluster large sets of data. The method works on the principle of partitioning clustering and involves classifying the given data objects into k different clusters through an iterative process, converging to a local minimum [1]. The resulting clusters are compact and independent.

The algorithm consists of two distinct phases: first, k centers are randomly selected, and second, each data object is assigned to the nearest center based on Euclidean distance, determining the distance between each data object and the cluster centers.

Suppose the target object is x, x_i indicates the average of cluster C_i, criterion function is defined as follows:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - x_i|^2$$

The basic process of k-means algorithm is:

- Randomly select k data objects from dataset D as initial cluster centers.
- Calculate the distances between each data object d_i $(1 \leq i \leq n)$ and all k cluster centers c_j $(1 \leq j \leq k)$ and assign data object d_i to the nearest cluster.
- For each cluster j, recalculate the cluster center
- Repeat two steps above until no changing appears in the center of clusters

The sum of squared errors (E) is a criterion function used in K-means clustering to determine the quality of the resulting clusters. The Euclidean distance is the distance measure used to determine the nearest distance between each data object and the cluster center. It is defined as the square root of the sum of the squared differences between each coordinate of two vectors, which can be expressed mathematically as follows:

$$d(x_i, y_i) = \left[\sum_{i=1}^{n} (x_i - y_i)^2 \right]^{\frac{1}{2}}$$

Here, x and y represent two vectors with n number of dimensions, and x_i and y_i represent the i-th coordinates of the vectors. This distance measure is used to minimize the sum of squared errors (E) between the data objects and the cluster centers in K-means clustering.

6.2 Adjustment of Model

The Rank-Sum Ratio (RSR) model is a regression model that aims to identify the optimal values of predictor variables that minimize the response variable. It is commonly utilized in process control and quality improvement applications to detect sources of variation and enhance process performance. The RSR model establishes a relationship between a response variable and one or more predictor variables. In order to use the RSR model, it is necessary to select positive and negative factors that influence the response variable. To facilitate this process, a heatmap is employed to visualize the relationship between reported score and identify these factors based on their effect on the response variable.

From Figure 7, we can see that there is a strong positive correlation between 1 try and 2 tries (0.618209), 1 try and 3 tries (0.335239), and 2 tries and 3 tries (0.755279). It also appears a strong positive correlation between 5 tries and 6 tries (0.693939), and a positive correlation between 5 tries and 7 or more tries (0.124513). Based on these findings, we can separate factors into two parts: one with 1 try, 2 tries, 3 tries and 4 tries, the other with 5 tries, 6 tries, and 7 or more tries. Since we have already picked positive factors and negative factors, we can easily make cluster analysis based on dataset.

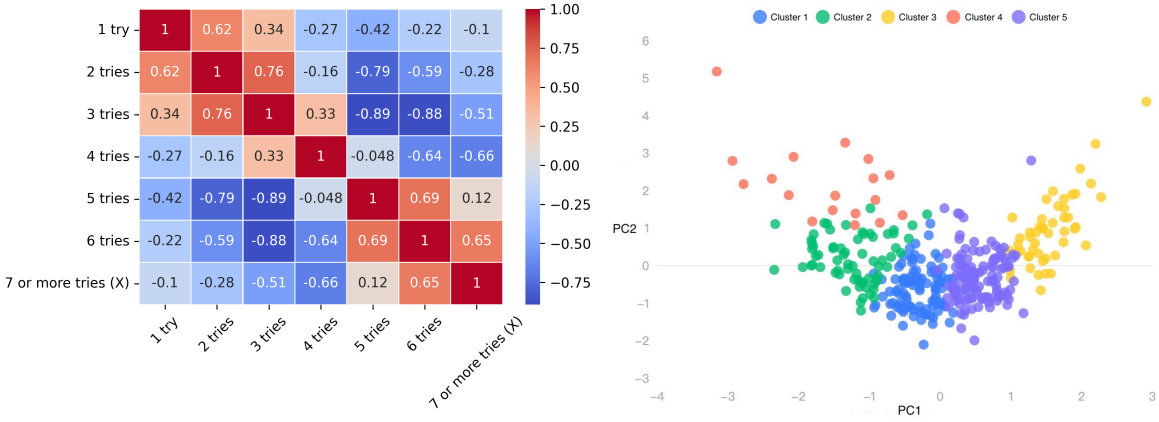

Figure 7: Reported Scores Heatmap (Left) Difficulty Cluster Analysis (Right)

6.3 Model Outcome

The K-means cluster plot shows the result of clustering a data set of reported scores. Figure 7 displays five distinct clusters, each represented by a different color. The x-axis and y-axis represent each principle component. The clusters are based on a K-means algorithm with $k = 5$, where each Wordle is assigned to the cluster with the closest mean. These five clusters each represents Wordles with certain difficult level.

In the context of K-means clustering, the F-score is used to evaluate the quality of the clustering results by comparing the variance between clusters to the variance within the groups. A higher F-score indicates the clustering results are more reliable. In Table 10, we can see that F-scores are strong enough to indicate our clustering results are reliable. Also, from the table, all P-values are reported as 0.000***, indicating that the difference between the clusters are statically significant at a very high level of confidence (less than 0.001). This suggest that the clustering results are highly reliable and meaningful.

In Table 11, the Silhouette coefficient is 0.335, suggesting the overall clustering result is reasonably good. With the DBI score being 0.983 and CH index being 278.63, it indicates that the clusters are well-separated. Overall, based on coefficients and indexes above the clustering results appear to be of good quality.

7 Predict Difficulty Level based on CNN

After we successfully cluster Wordle according to their reported scores, we need to use the attributes of the Wordle to predict a Wordle's difficulty level. So we train CNN (Convolutional Neural Network) model to make predictions on the distribution.

7.1 Introduction of Model

CNN (Convolutional Neural Network) is a deep learning that can be used to automatically extract relevant features from high-dimensional data, such as time series or text data. The convolutional layers of the CNN are applied to the input data, which could be a sequence of values or a document of text, and

	Cluster Tag					F	P
	Cluster 5	Cluster 1	Cluster 2	Cluster 3	Cluster 4		
1 try	0.455	0.225	0.269	1.367	0.278	25.417	0.000***
2 tries	6.65	3.657	2.701	13.531	3.5	216.601	0.000***
3 tries	27.407	19.51	13.075	34.388	13.167	479.718	0.000***
4 tries	35.472	35.814	28.612	30.959	20.611	115.671	0.000***
5 tries	20.74	26.892	31.134	14.388	22.278	349.787	0.000***
6 tries	7.862	11.892	19.627	4.673	23.667	350.812	0.000***
X	1.293	2.098	4.493	0.735	16.5	184.288	0.000***
Note: ***, **, * denote 1%, 5%, 10% significance level							

Table 10: K-means Model Summary

Silhouette coefficient	DBI	CH
0.335	0.983	278.636

Table 11: Cluster Summary

filters are used to identify patterns in the data. The resulting feature maps can be further processed using pooling layers, which can help to reduce noise and redundancy in the extracted features [5]. Once the features have been extracted, they can be passed to fully connected layers for classification or regression tasks. CNNs have been used successfully in various data analysis tasks, such as speech recognition, natural language processing, and anomaly detection in time series data. Overall, the ability of CNNs to automatically learn features from raw input data makes them a powerful tool for data analysis, particularly in domains where traditional statistical methods may struggle with high-dimensional data.

We change a Wordle into a matrix consisting of the appearing time of each letter because it will covers all the possible attribute of a Wordle. And we feed the CNN model with the dataset of Wordle matrix. The CNN consists of three convolutional layers with 8, 16, 32 filters, respectively. Each convolutional layer was following by a max-pooling layer and a batch normalization layer. We use a rectified linear unit (ReLU) activation function for all layers except the output layer, which used a softmax activation function to generate the class probabilities. The CNN was trained using a categorical cross-entropy loss function and the Adam optimizer with a learning rate of 0.001. We trained the CNN for 100 epochs with a batch size of 1.

7.2 Model Outcome

The CNN model achieved a accuracy of 83.9% on the train set and a classification accuracy of 81.2% on the test dataset. From the CNN model prediction result, the probability of being classified as each difficulty level is shown in Table 12. And the difficulty level with the largest probability is the

Difficulty Level	1	2	3	4	5
Probability	0.08068027	0.21517237	0.43974343	0.19988433	0.06451967

Table 12: Result Summary

predicted difficulty level. Thus, from the prediction, EERIE lies in the difficulty level of 3 out of 5, which is the medium difficulty level.

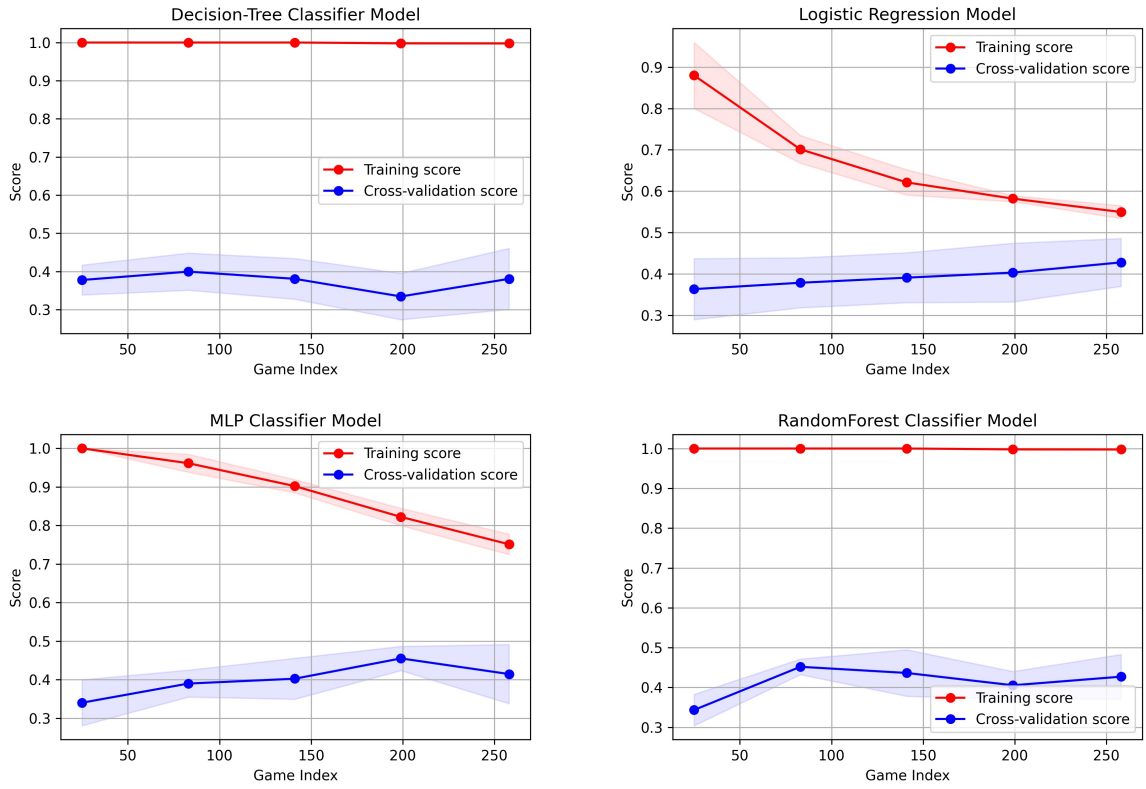

Figure 8: Model Result Comparison

We can also consider other models. However, the models fail to perform well on this dataset. From Figure 8, we can find that the Decision-Tree Classifier model and RandomForest model are overfitting, and MLP Classifier is underfitting. Only Logistic Regression Model looks good. However, after we compare the Accuracy Rate of train set and test set in Table 13, we can find that CNN model is better than all the other models.

8 Sensitivity Analysis

8.1 Method Description

Assuming that we are predicting the distribution of reported scores using the three models (MLP, DT, and RF) and X, we can use statistical metrics to compare the performance of the models when

	Acc (Train)	Acc (Test)	MAE	F-score
Decision-Tree	0.997	0.444	0.722	0.445
RandomForest	0.797	0.500	0.583	0.487
MLP	0.731	0.417	0.722	0.415
Logistic Regression	0.545	0.472	0.667	0.469

Table 13: Model Comparison Summary

changing from a maximum of 6 tries to a maximum of 7 tries.

8.2 Analysis Outcome

First, let's look at the results using a maximum of 6 tries in Table 14

Model	MAPE	RMSE	MAE	R^2
MLP	0.02221	0.17509	0.11785	0.92437
DT	0.06108	0.40413	0.29984	-2.50329
RF	0.10318	0.10295	0.07135	0.82831

Table 14: Result Summary (Max 6 Tries)

As we can see from these results, the MLP and RF models have relatively low errors (MAPE, RMSE, and MAE) and high R^2 scores, indicating good performance. However, the DT model has significantly higher errors and a negative R^2 score, indicating poor performance. Now, let's see what happens when we increase the maximum number of tries to 7 in Table 15

Model	MAPE	RMSE	MAE	R^2
MLP	0.03470	0.30718	0.22252	0.87239
DT	0.06796	0.47542	0.36726	-3.21049
RF	0.02065	0.16423	0.10790	0.44226

Table 15: Result Summary (Max 7 Tries)

We can see that increasing the maximum number of tries to 7 results in higher errors (MAPE, RMSE, and MAE) for all models, indicating worse performance. The R^2 score also decreases for all models, indicating that the models are less able to explain the variation in the data. Comparing the results between the maximum of 6 tries and the maximum of 7 tries, we can see that the relative performance of the models remains largely the same. The MLP and RF models continue to have relatively low errors and high R^2 scores, while the DT model continues to perform poorly. However, all models experience a decrease in performance when the maximum number of tries is increased.

In summary, increasing the maximum number of tries from 6 to 7 appears to have a negative impact on the performance of the models, as indicated by an increase in errors and a decrease in R^2 score. However, the R^2 score is still high enough to support the robustness of the model.

9 Strength and Weakness

9.1 Strengths

- Rolling window technique in time series analysis (LSTM in this problem) provides more accurate predictions and helps to mitigate overfitting by using a moving window of data to update the model at each step.
- Normalization of time-series data improves the accuracy of predictions by ensuring that all features are on the same scale, preventing some features from dominating the model.
- Rank-sum Rate (RSR) wisely utilizes the positive factors and negative factors derived from the heatmap to rank clusters from the K-means model.

9.2 Weaknesses

- Small sample size may lead to difficulty controlling extreme values and potential underfitting.
- Predicting the difficulty level of a Wordle word using a classification model may result in similar predicted probabilities of multiple difficulty levels, making it challenging to determine the word's actual difficulty level.
- Conducting a sensitivity analysis can prove to be challenging due to the nature of the data set, where the variables cannot be modified.

10 Interesting Features

Figure 9 explains four different scatter plots showing the relationship between different variables in a dataset. The upper-left scatter plot shows the relationship between the rate of reported scores and words in a game. The plot suggests that the rate is divided into two groups, with lower rates being associated with fewer tries and no tries, while higher rates are associated with more tries. The upper-right plot shows the number of players of the game over time, indicating a decrease in players from March to October, possibly due to a rumor that the game became more difficult after the New York Times purchased it in January. However, the plot suggests that the rate for each reported score almost remains the same. The lower two graphs depict the frequency of common bigram letter pairs in the dataset. The graphs show that letter pairs such as ER, IN, ST, LO, AL, and AR are more common in the dataset, which is also confirmed by the Bigram Network in the right graph.

11 Conclusion

In conclusion, this paper has successfully developed prediction models for several aspects related to the game of Wordle. Using LSTM modeland a second question model, we were able to make accurate predictions about the number interval of reported results on March 1, 2023, which was found to be in the range from 22537 to 23041. Additionally, we used K-means and RSR to classify solution words into five clusters by difficulty and determined that the Wordle "EERIE" is of a medium difficulty with

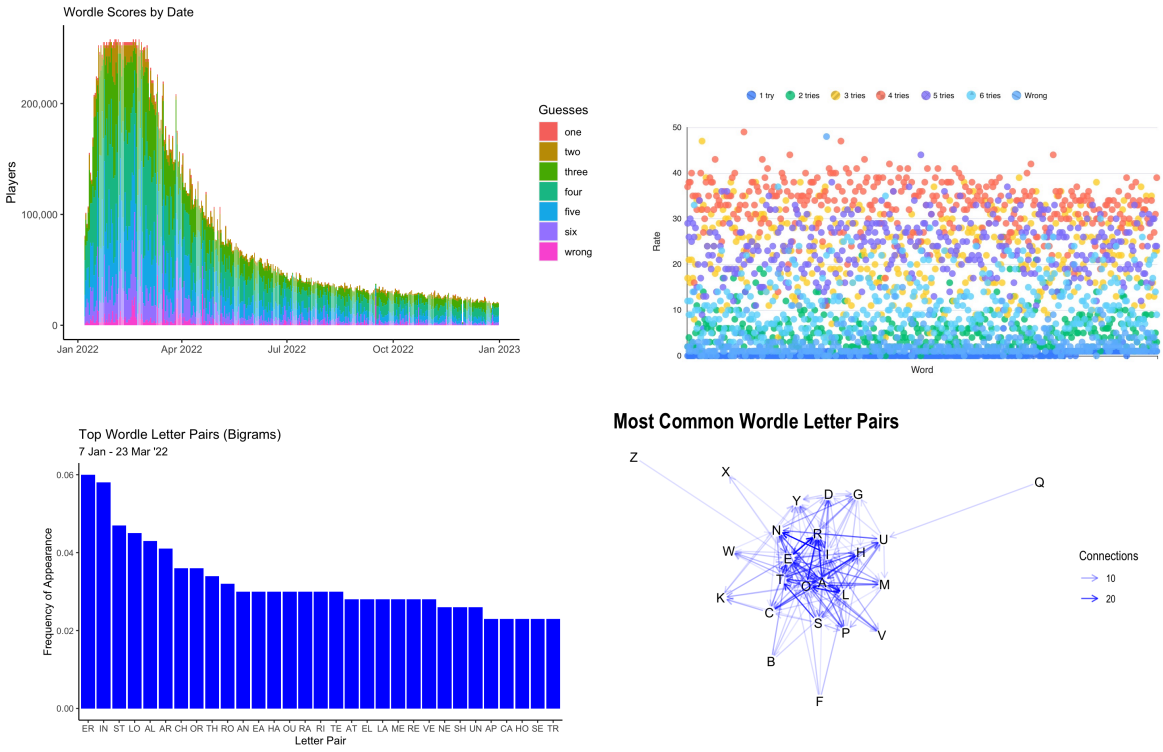

Figure 9: Interesting Features

difficulty 3 out of 5 depending on the prediction from CNN model. We also found that no attribute of Wordle affects the percentage of scores reported that were played in Hard Mode.

Overall, this research provides valuable insights into the game of Wordle and the factors that impact player performance and reporting. By developing these prediction models, we can better understand the game and make more informed decisions about how to approach it. We believe that our findings will be useful not only to players of Wordle but also to game developers looking to improve upon the game's design and functionality. We hope that this research will inspire further study and exploration of this popular game and its many intricacies.

12 Letter

February 20, 2023

Dear Puzzle Editor,

I am writing to share the conclusions of our recent research paper, which aimed to develop prediction models for various aspects related to the game of Wordle. Our study involved analyzing and modeling the game using LSTM, Multi-layer Receptron Regression Chain, PSO, CNN, K-means and RSR techniques.

We are pleased to report that our analysis allowed us to make accurate predictions about the number interval of reported results on March 1, 2023, based on our LSTM Model. Our estimated prediction interval for the number of reported results is between 22,573 and 23,041. We believe this finding could be of great interest to players and game developers alike.

In addition, we investigated whether any attribute of Wordle affects the percentage of scores reported that were played in Hard Mode. However, our findings indicate that there is no such attribute. Thus, we hypothesize that players' choice to play Wordle in Hard Mode is based on personal confidence rather than the difficulty level of the game.

Furthermore, our research successfully utilized Multi-layer Preceptron Regression Chain and PSO algorithm to make predictions on the distribution of reported results based on Wordle. Our predictions indicate that the distribution of reported scores for the word "EERIE" will be [0.581, 8.08, 24.729, 31.813, 22.546, 10.162, 1.706]. This means that we estimate 0.581% of players will pass on their first try, 8.08% on their second try, 24.729% on their third try, and so on.

We also classified solution words by difficulty using K-means and RSR, which allowed us to determine the difficulty level of each Wordle. According to our predictions, Wordle "EERIE" lies in difficulty level 3 out of 5, which is a medium difficulty level.

Overall, our research provides valuable insights into the game of Wordle and the factors that impact player performance and reporting. We believe that our findings will be useful to both players and game developers, and that they will inspire further study and exploration of this popular game.

Thank you for your dedication to providing top-notch puzzles and bringing new and exciting games to your readers. We look forward to solving Wordle and other puzzles in The New York Times for years to come.

Sincerely,

Team 2320341

References

[1] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

[2] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larranaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Chengcheng Chen, Qian Zhang, Mahsa H. Kashani, Changhyun Jun, Sayed M. Bateni, Shahab S. Band, Sonam Sandeep Dash, and Kwok-Wing Chau. Forecast of rainfall distribution based on fixed sliding window long short-term memory. *Engineering Applications of Computational Fluid Mechanics*, 16(1):248–261, 2022.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[6] Ivan Li. Analyzing difficulty of wordle using linguistic characteristics to determine average success of twitter players. 2022.

[7] Yuting Li and Xinhua Zhu. Multi-output regression with high-dimensional output space via ridge regularization. *Journal of Machine Learning Research*, 22(71):1–40, 2021.

[8] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[9] Sreelekshmy Selvin, R Vinayakumar, E. A Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647, 2017.

[10] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[11] Mingxin Sun, Tiegang Wang, and Meng Wang. Decision tree-based parameter optimization for deep learning models. *Neurocomputing*, 398:43–53, 2020.

[12] Teguh Wahyono, Yaya Heryadi, Haryono Soeparno, and Bahtiar Saleh Abbas. Enhanced lstm multivariate time series forecasting for crop pest attack prediction. *ICIC Express Lett*, 10:943–949, 2020.