Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2322627

Summary

Wordle, A puzzle game, offered by New York Times, went viral on social media. Players are asked to guess the correct words within 6 attempts. The game will deliver a corresponding prompt after each guess is completed by the player.

As for task I subsection I, **STL decomposition and Granger causality test** were performed to interpretate the variation in the number of reported results. The STL decomposition explains **the reason for the change in the number of reported results from the time dimension**. Granger causality test analyzes the effect of word difficulty time series on the number of reported outcomes time series. The word difficulty is quantified with a weighted average of the number of attempts. The results of Granger causality test shows that **the variation of word difficulty is another reason for changing in number of reported results**: The more difficult the previous words, the higher the number of results reported in the future. As for subsection II, ARDL model and ARIMA model were applied to predict the number of reported results on 1st March. Considering the variation of reported number was driven by time changing and variation of difficulty, ARDL model was adopted instead of pure time series forecasting models. ARIMA model was applied to forecast the value of difficulty and the prediction value was then input ARDL model to predict number of reported results.

For task II, we take both **LSTM and Deep-learning neural network (DLNN) model** into account and compare their accuracy of predictions about the probability distribution of reported results. The results suggest that **DLNN model performs better than LSTM obviously**, where the former model has a **0.7778 confidence level**. That is, we can eliminate the effect of time-series from the guess, which shows that **the percentages of reported results are independent of date and directly encoding words as feature vectors is an excellent method to contain considerable word attributes**. For task III, **KNN** was used to classify the data using several attributes as variables and the difficulty levels as classes. The observation is that choosing **six attributes**, such as the frequency of the word and the number of consonants in the word as attributes and **three clusters** can have the highest accuracy in both train and test set, both **larger than 0.7**. Using our model, the word "EERIE" is predicted to be **middle-level difficult**.

As for Task IV, the interesting features we found is that the data set can be interpreted by the **SIR infectious disease dynamics model**. If the infection rate, recovery rate, and re-infection rate are set as appropriate constants, the number of infections (that is, the number of people currently playing the game) generated by the SIR model can be approximately regarded as a smoothed curve of the number of reported results. If the machine learning algorithm is used to adjust the parameters of the SIR model according to the word attributes, the model will explain the data better. However, limited by only 300 pieces of data, the algorithm cannot be implemented, but the SIR model is still a comprehensive model with strong explanatory power.

Keywords: PCA, ARDL, SIR, ARIMA, KNN, Deep Learning, CNN

Contents

1 Introduction

1.1 Problem Background

A puzzle game, Wordle, offered by New York Times, went viral on the social media. The Player who guesses the word set in advance within 6 tries win. As long as the word filled in is a real five-letter word, the player will get clues about the answer. If the letter you input also in the answer but in the wrong location, the corresponding tile will be painted yellow. If it is a correct letter in correct place, the tils will be colored green. A gray tile means a wrong letter. This game also allows players to freely choose the degree of difficulty they want. Players who choose the hard mode are required to enter words that contain the correct letters previously found.

1.2 Restatement of the Problem

- **What We Know**:
  1. The solution word on the associated date.
  2. Number of reported results and number of people who chose hard mode.
  3. The distribution of the number of attempts to solve the problem.
- **What We Should Do**:
  1. Interpret the variation in the number of daily reported results.
  2. Forecast number of results reported on March 1, 2023.
  3. Investigate the effect of word attributes on the proportion of hard modes selection.
  4. Predicting the distribution of problem-solving attempts for a specific vocabulary on a future date.
  5. Develop Difficulty Classification Model Using Summarized Word Properties.

1.3 Model Framework

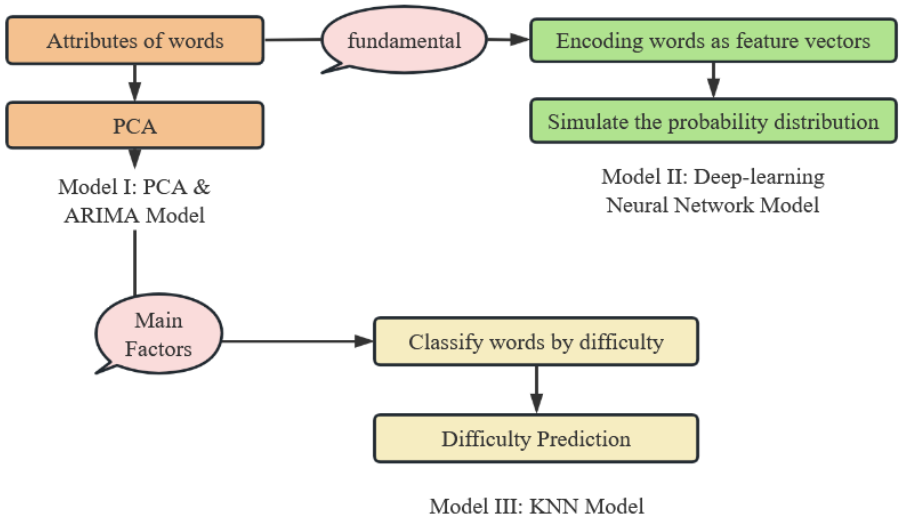

Figure 1: The overall algorithm flow chart

In this paper, we establish three models to solve the problem. Model II, based on the Deep-learning Neural Network model, is established to predict the probability distribution of reported results for a solution word on a future date and finds that directly encoding words as vectors is a fundamental method. Then we develop Model I to take the attributes of words into consideration and perform Principal Component Analysis to build several main factors. At last, we build Model III to classify solution words by difficulty in relation with main factors based on the K-NearestNeighbor Model and link the difficulty with proposed attributes of words in Model II. The overall algorithm flow chart of the model is shown in the Fig. 1.

2 Assumptions and Justifications

To simplify our model and eliminate the complexity, we make the following main assumptions in this paper.

- **Assumptions 1:** There is no difference between the probability distribution of reported scores and that of unreported scores.
- **Justifications:** Since the data contains only the number and probability distribution of reported scores in Twitter, according to some common characteristics of human, we believe that the whole picture of all Wordle players is the same as what is reported. This assumption makes the model better fit the distribution of data points.
- **Assumptions 2:** The default value of X is 7 and the weighted average of tries can represent the difficulty level of the corresponding word.
- **Justifications:** This assumption is to simplify the calculation of the model and remove the influence of unknown factors on the model.
- **Assumptions 3:** Wordle game players are homogeneous in the aspect of knowledge of English.
- **Justifications:** Since the given data only considers the big picture of Wordle players, no individual difference will be included in the paper.

3 Notations

The notations used in the paper are in the Fig. 2.

The notations used in this paper

Symbol	Description
α	Re-infected rate
β	Infection rate
γ	Recover rate
u	Bias
c	Weight of time

Figure 2: Notations

4 Model Derivation and Sensitivity Analysis

4.1 Model Preparation: Data Cleaning

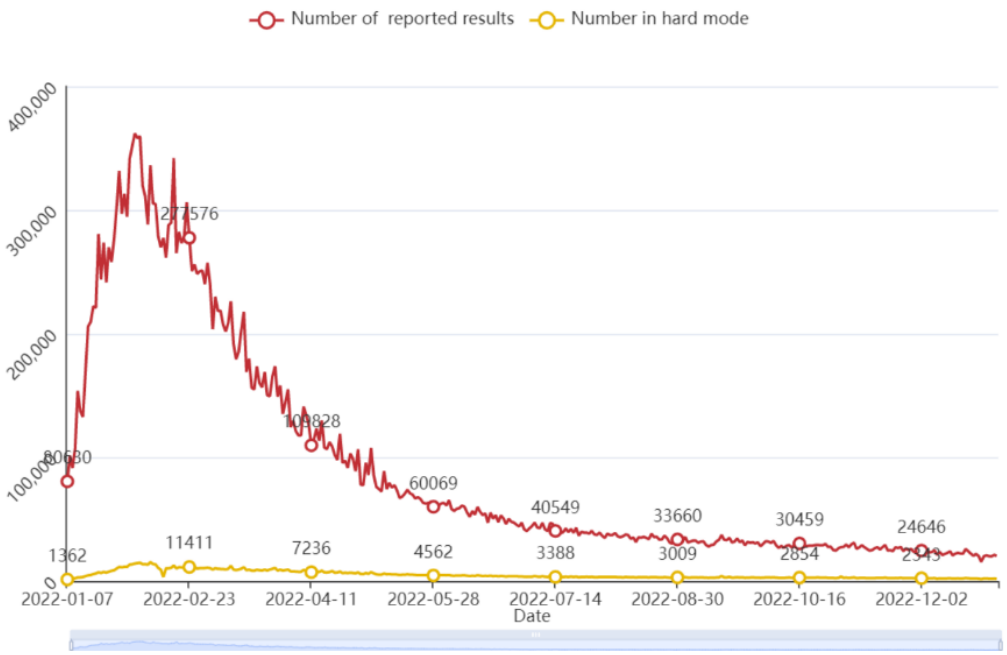

Figure 3: Number of Reported and Number in Hard Mode

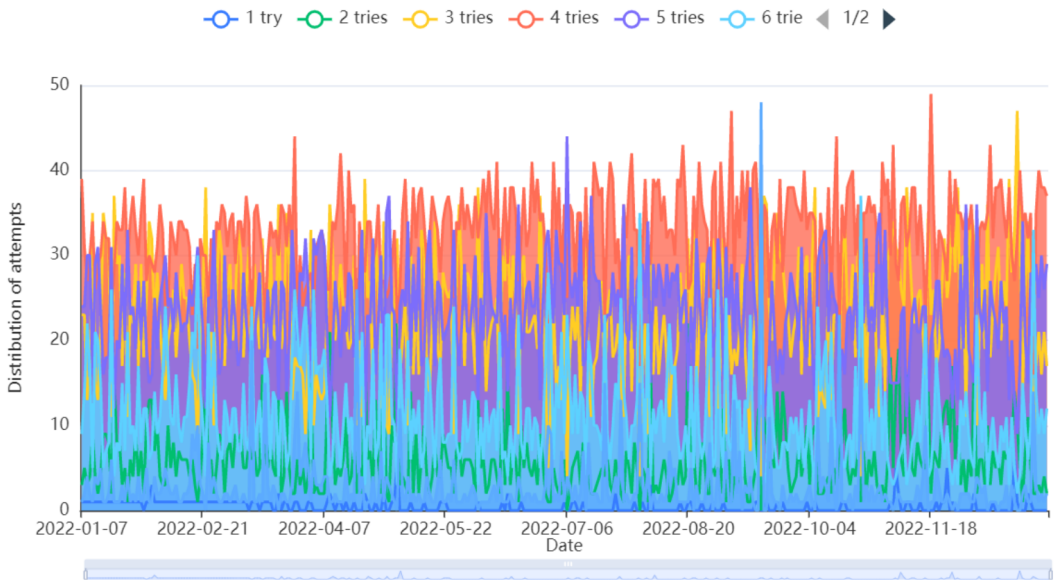

Figure 4: Distribution of Attempts

The total number of reported results on November 30, 2022 is 2569, which is about one-tenth of the number of the previous day. And almost all of the participation chose the hard mode on that day, while there was no significant abnormality in the number of people who chose the difficult mode that day. So, the total number of reported results on 30th Nov. is considered as an outlier and replaced with the rounded average of November 29th and December 1st. The data visulizations are shown in Fig. 3 and Fig. 4.

4.2 Model 1: PCA and Time Series Analysis

In this chapter, there are two sub-questions and they correspond to different models. They are a) Explain the variation on the number of reported results and create a prediction for 1st Mar. b) Determine the attributes of the word and their effect on the percentage of scores reported that were played in Hard Mode.

4.2.1 Variation Explanation and Prediction

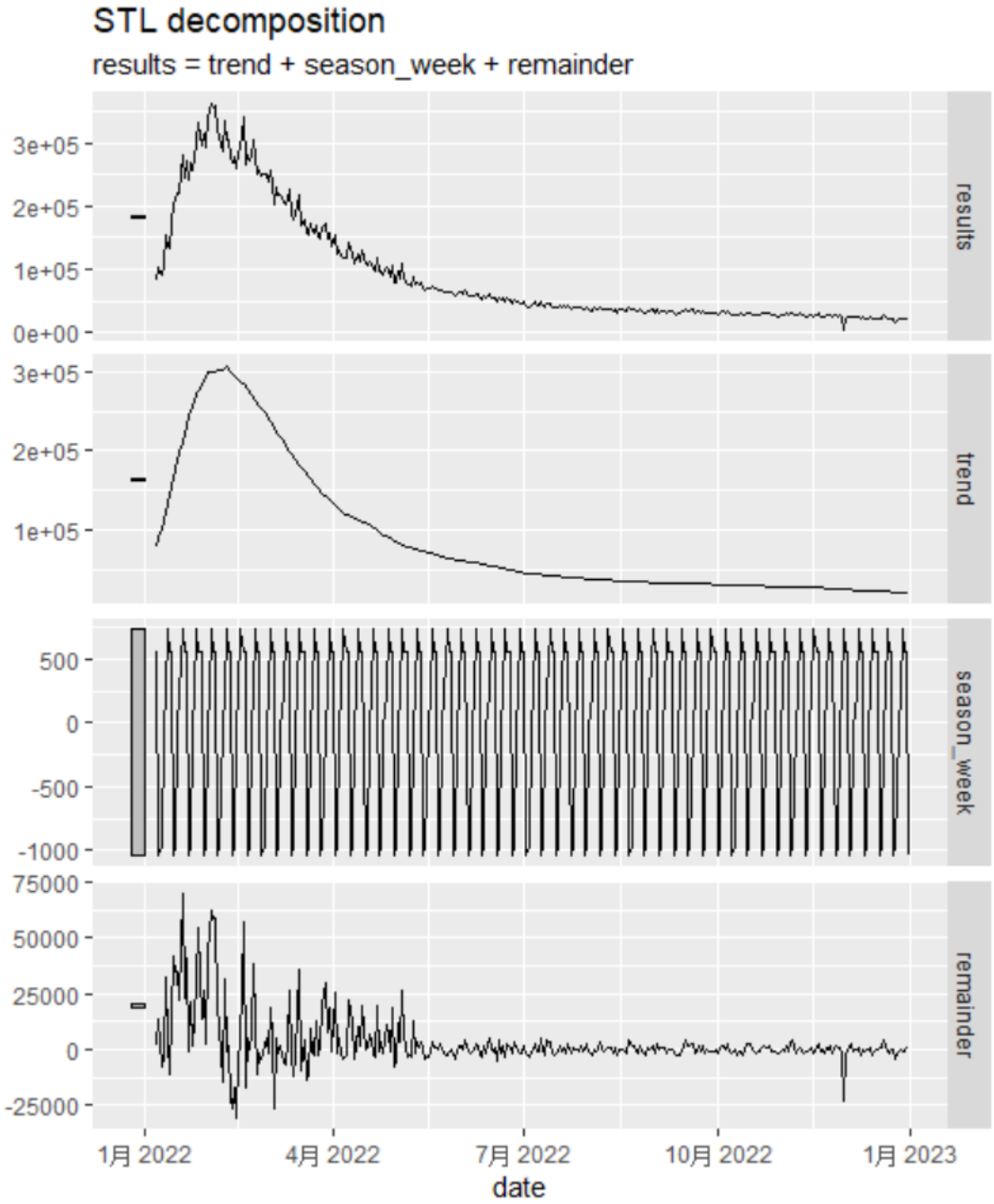

Figure 5: Trend Following Time

First, time series plot was drawn and STL [1] decomposition was performed to determine the underlying pattern. STL is a versatile and robust decomposing method for time series and it can decompose a time series into three components: a trend-cycle component, a seasonal component, and a remainder component. As the following graph shows, the variation of daily number of reported results is mainly driven from a trend, which shows a sharp rise and then a slow decline, and weekly seasonality. The reason for trend is that the game quickly became popular on the Internet when it first came out, resulting in a rapid increase in the number of participants. Afterwards, people gradually lost interest, and the number of participants slowly declined. The rationality for seasonality may be justified by the cyclical nature of people's weekly disposable time and interests in gaming. The trend is shown in Fig. 5.

$$WA = \sum_{i=1}^{7} i * \text{ percentage of players solving the puzzle in i guess}$$

Figure 6: Formula

In addition to the time series, other factors such as the difficulty of words before may also have an impact on the number of reported results. Weighted average attempt is applied as a measurement for the level of difficulty. Words guessed using more times are considered more difficult. Formula shows in Fig. 6.

$$y_t = \sum_{i=1}^{k} \alpha_i \, y_{t-i} + \sum_{i=1}^{k} \beta_i \, x_{t-i} + u_{1,t}$$

Figure 7: Formula

The Granger causality test is used to test whether one set of time series is the cause of another set of time series. If A is the Granger cause of B, it means that the change of A is one of the reasons for the change of B. Formula shows in Fig. 7.

Testing the null hypothesis for the existence of Granger non-causality is: $\beta_1 = \beta_2 \ldots = \beta_k = 0$.

Paired Samples		F	P
WA	Number of reported results	3.723	0.025**
Number of reported results	WA	0.099	0.906

Note: ***, **, * represent 1%, 5%, 10% significance level respectively.

Figure 8: Results

As the result show, the significance P value is 0.025, showing significance, rejecting the null hypothesis, WA can cause changes in the Number of reported results. Results show in Fig. 8.

According to the regression results, the higher the difficulty level of the word in the current day, the more the number of results reported in the next day. This is another explanation for the time variation.

$$y_t = c_0 + c_1 t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=0}^{q} \beta_i' \mathbf{x}_{t-i} + u_t$$

Figure 9: Formula

Considering the number of results reported today is affected by the number of results reported in the past and the difficulty of past questions, ARDL model was applied for prediction. Autoregressive distributed lag models (ARDL) have been used to characterize variable relationships in a single time series equation. Because the co-integration of non-stationary variables is equivalent to an error correction model, and the autoregressive distribution lag model can be obtained after simplifying the error correction model. The formula shows Fig. 9.

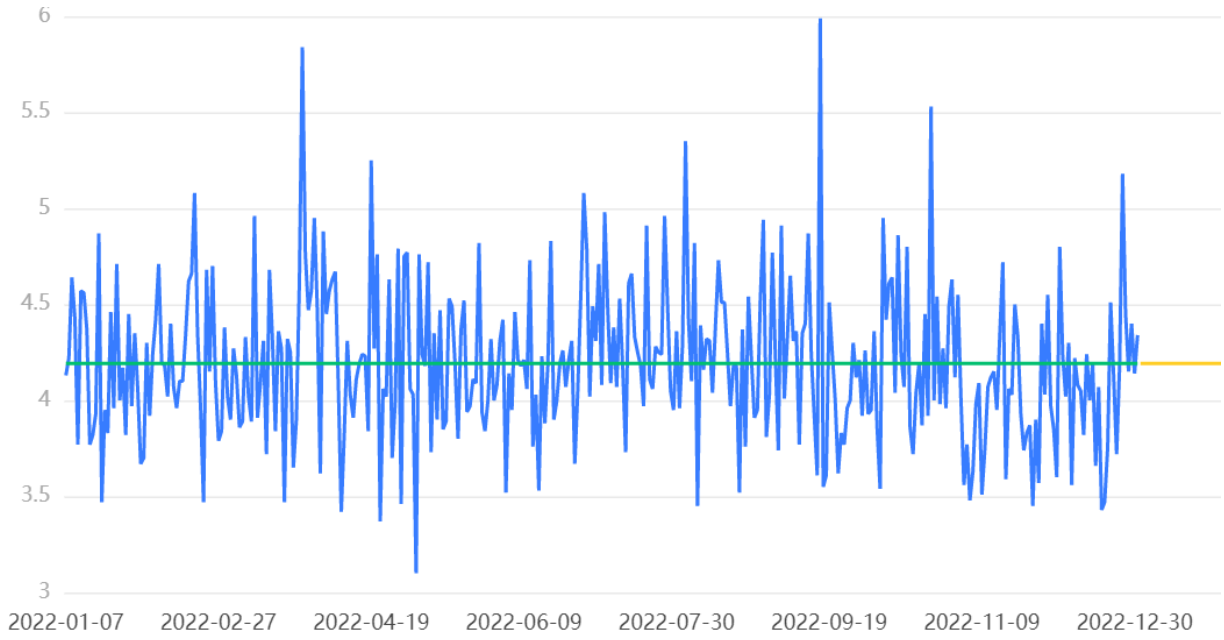

Figure 10: ARIMA Prediction

According to the above figure, it can be seen that predicting the number of report results after 60 days needs to predict WA after 60 days. Referring to the Granger causality test, the number of reported results does not affect the value of WA, so we can only consider the impact of time series when predicting WA. Therefore, the ARIMA model can effectively predict WA. The prediction is shown in Fig. 10.

It can be seen from the figure that the forecast of WA in the next 60 days is a constant, that is to say, it cannot affect the number of reported results. Forecasting the number of reported results can simply use the time series model ARIMA.

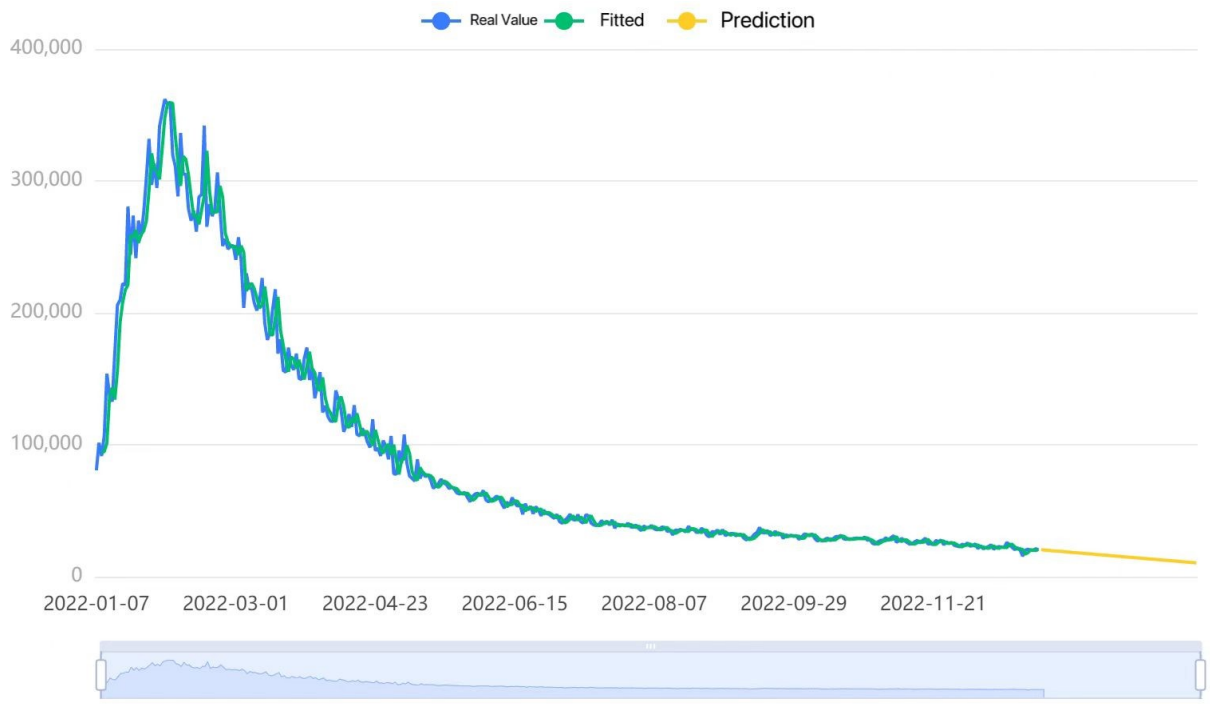

Figure 11: Prediction Interval

According to the result from ADF test, ACF and PACF, the parameter for ARIMA model is ARIMA(1,1,0). At the same time, the goodness-of-fit R^2 of the model was 0.982, the model performed well, and the model met the requirements. The prediction for number of reported results on 1st March,2023 is 10456, 0.95 confidence interval [9411,11501]. The figure is shown in Fig. 11.

Based on the above explanation of the variation on the number of reported results, we employ the ARDL model to predict the number of reported on 1st March, 2023.

4.2.2 Word Attributes and Impact

Referring to linguistic research results and experience [2], a total of 8 words have been summarized and then refined using principal component analysis and decision trees.

Attributes	Quantized value
Frequency	Positive real number
Number of repeated letters	0/2/3
Number of consonants	0/1/2/3/4/5
Connected	0/1
Consonant-middle	0/1
Consonant-first	0/1
concrete	0/1
Number of common letter combinations	Non-negative integer

Figure 12: Results

Frequency characteristics refer to how often words are used in everyday life. Word-freq is a python library that summarizes word frequency by pooling 8 different text domains such as Wikipedia, subtitles, news, books, web texts and twitter. Connected is a

dummy variable which describes whether there are two consecutive and identical letters in a word. For example, for the word “sheep”, the variable “connected” is labeled as 1. “Consonant-middle” is a dummy variable which describe whether the 3rd letter in a 5-letter word is a consonant. According to experience, a consonant in the third letter makes the word more difficult. Words with a consonant in the middle are marked as 1. “Consonant-first” is a dummy to determine whether the first letter is consonant. Words are labeled 0 if the first letter is a consonant. "concrete" is a dummy variable that describes whether the meaning of the word is concrete. According to Cleveland ’s research in 1990, the more specific the meaning of a word, the easier it is to be recalled. For example, apple is concrete while beautiful is an abstract word. The results are shown in Fig. 12.

Considering that there are too many factors extracted above and have strong correlations, we use principal component analysis to reduce the dimensionality of information. Principal component analysis is a multivariate statistical method that seek comprehensive substitutes for related variables through the correlation of original variables, and to ensure the minimum information loss in the transformation process.

KMO test and Bartlett's test		
KMO		0.785
Bartlett test for sphericity	Approximate chi-square	325.469
	df	28
	P	0.000***

Note: ***, **, * represent the significance levels of 1%, 5%, and 10% respectively

Figure 13: KMO and Bartlett’s test

As the results from KMO and Bartlett’s test shown in Fig. 13, principal component analysis is valid, and the data is suitable for principal component analysis.

Principal component weight results

Name	Variance explained rate (%)	Cumulative variance explained rate (%)
Factor 1	23.689	23.689
Factor 2	16.513	40.202
Factor 3	14.96	55.162
Factor 4	11.821	66.983
Factor 5	11.463	78.446
Factor 6	9.668	88.114

Figure 14: Total Variance

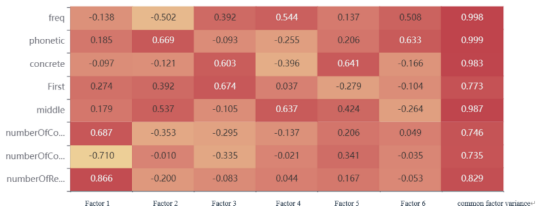

Figure 15: Total Variance

The following table in Fig. 14 and Fig. 15 is the total variance interpretation table, mainly to see the contribution rate of the principal components to the variable interpretation. It can be seen from the table that the contribution rate of the cumulative explanation of

the first six principal components reaches 0.88114, indicating that the use of the first six principal components can well summarize the lexical attributes.

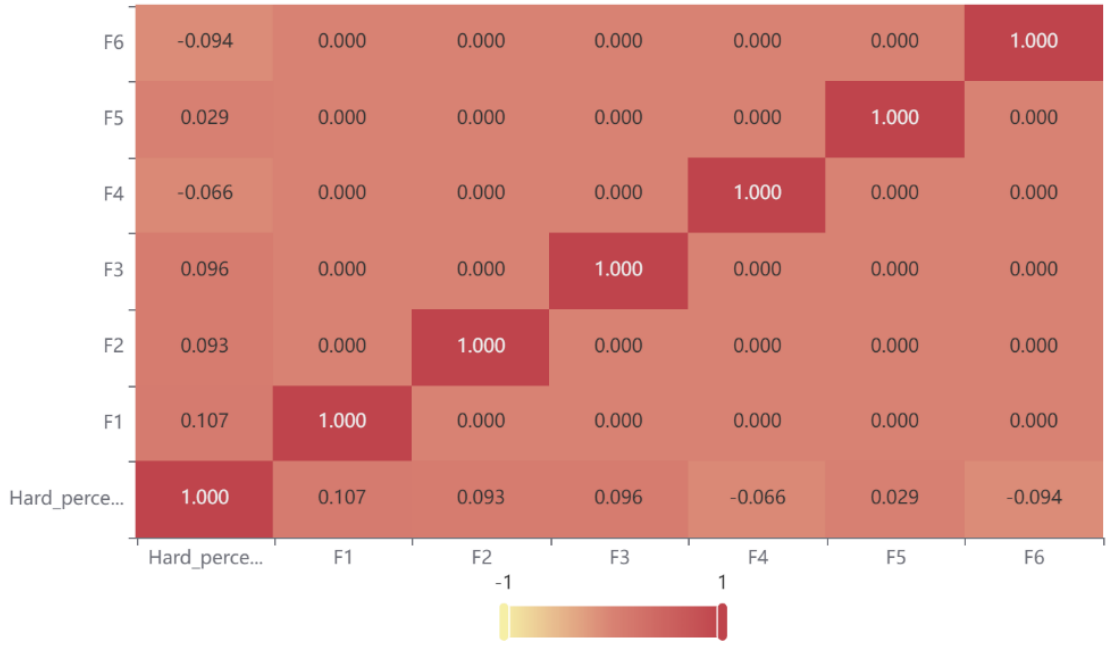

Figure 16: Results from Pearson Correlation Analysis

After finding the 6 principal components, Pearson correlation analysis was performed to judge the relationship between hard-mode choice and word attributes. There was no significant correlation between word attributes and the proportion of difficult mode choices. Intuitively, the player does not know the word when choosing the mode, so the word attribute will not affect the proportion of choosing the difficult mode theoretically. The results from Pearson correlation analysis are shown in Fig. 16.

4.3 Model 2: Deep-Learning Model Based on Artificial Neural Network

4.3.1 Model Establishment

At the first stage, Word2Vec with CBOW algorithm is used for presenting each word in the set of combined solution words with a vector. Therefore, we get a dictionary called "embeddings" containing words as keys and vectors as indexes by training the Word2Vec model with all words in the dataset.

$$min \sum_{i} \sum_{k}^{7} -log(Q(x_{ik})) * p(x_{ik})$$

Figure 17: Fomula of The loss function of categorical cross-entropy

The second stage is to create a deep-learning model with 3 layers of sequential linear artificial neural network, of which the inputs are "embeddings" and outputs are probability distributions of corresponding words. Each layer has some common arguments such as inputs dimension and activation function. The main idea is to optimize(minimize) the loss function of categorical cross-entropy by multiple iterations of resampling in the training set and training. The loss function of categorical cross-entropy is shown in Fig. 17.

In practice, we choose the cross-entropy function rather than the mean squared error (MSE) function due to the faster iteration speed of the former when applying the gradient descending method to iterate the independent variable.

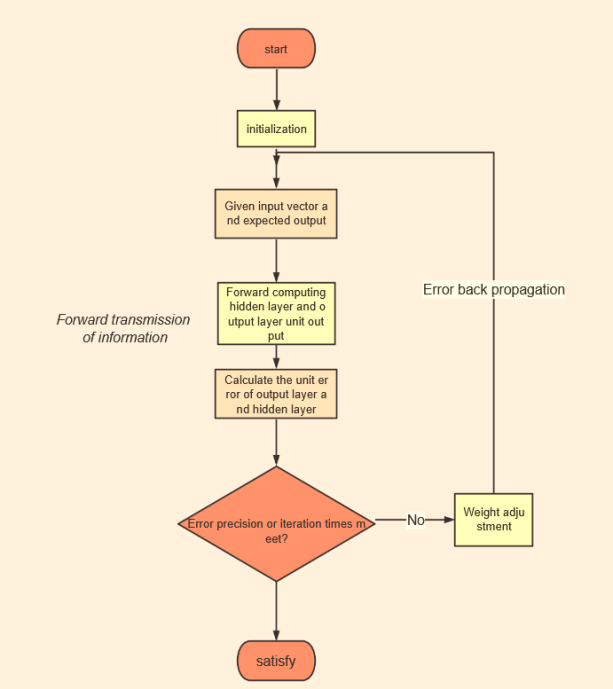

Figure 18: Processing

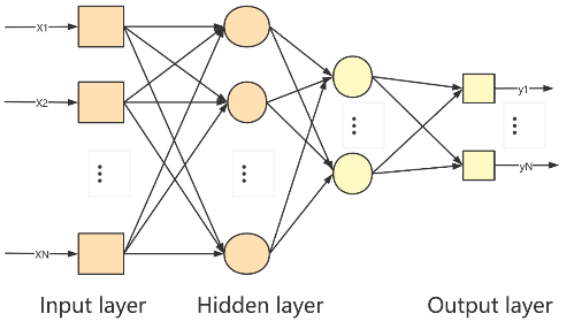

Figure 19: Processing

Finally, we divide the target dataset into 2 separate sets: 0.7 as a training set and 0.3 as a testing set. Given the trained model after the second step, plug the data in the testing set to examine the model accuracy also by optimizing the loss function. The processing figures are shown in Fig. 18 and Fig. 19.

4.3.2 Result of Problem 2

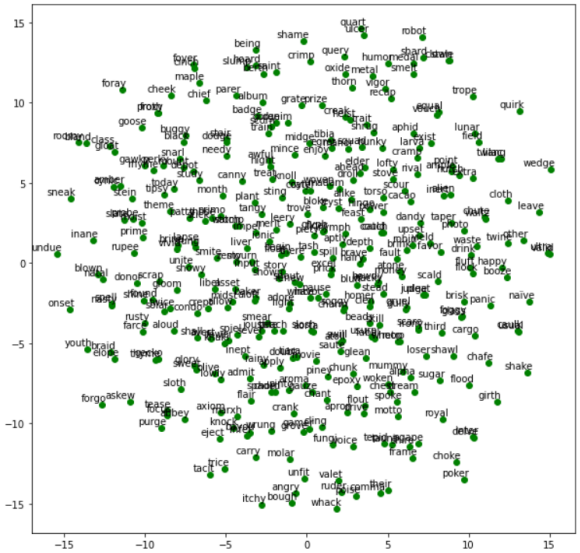

Figure 20: visualization the features of words

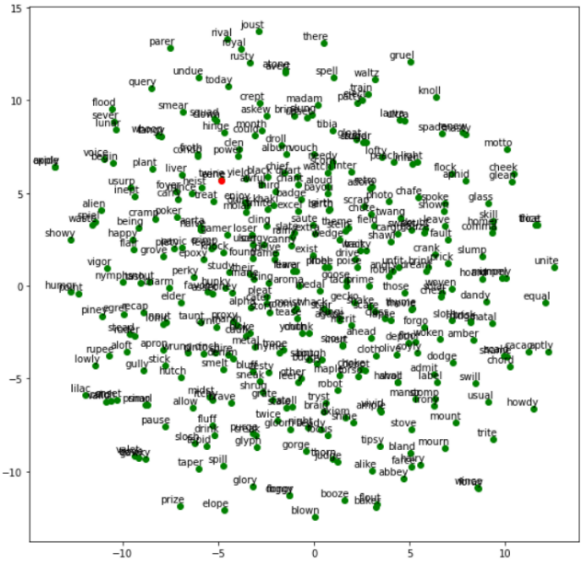

Figure 21: visualize the features of words

To visualize the features of words, we transform the high-dimensional vector into the 2-dimensional vector called through the dimensionality reduction of TSNE. The visulizations are shown in Fig. 20 and Fig. 21.

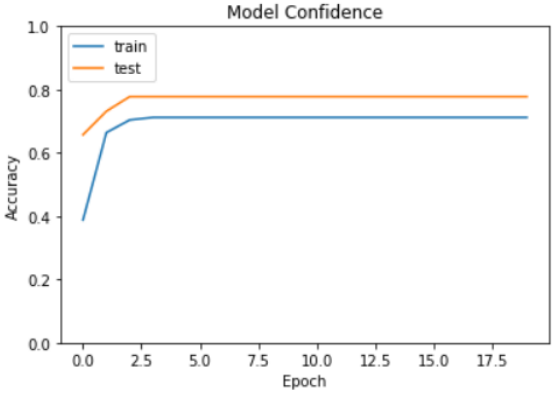

Figure 22: Results

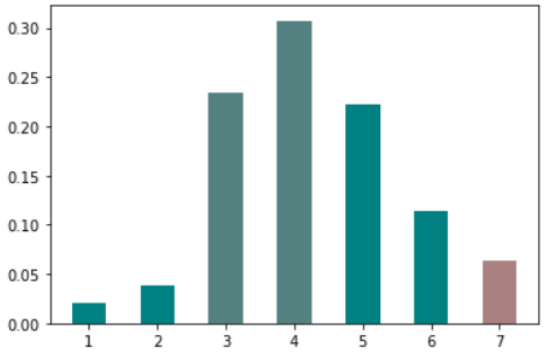

Figure 23: Results

We set 20 iterations for the deep-learning neural network model and record the accuracy for each iteration. Obviously, both 2 accuracy curves tend to be flat and converge to a certain value. Therefore, we conclude that the accuracy of training model is 0.7120 and that of testing model is 0.7778. The figure is shown in Fig. 22.

The probability distribution of the word “EERIE” on 1 March, 2023 is shown in Fig. 23 and the prediction is convincible at a 0.7778 confidence level:

4.3.3 Discussion for the Model

First, the given dataset reflects the number and probability distribution of reported scores on Twitter other than true outcomes, considering people’s aversion to report bad scores out of vanity and many other factors.

Second, the future outcomes may have some correlations with the past, therefore, a certain future date may be influenced by the unknown number of players in the past several days due to time-delay effect.

4.4 Model 3: KNN Classification Model

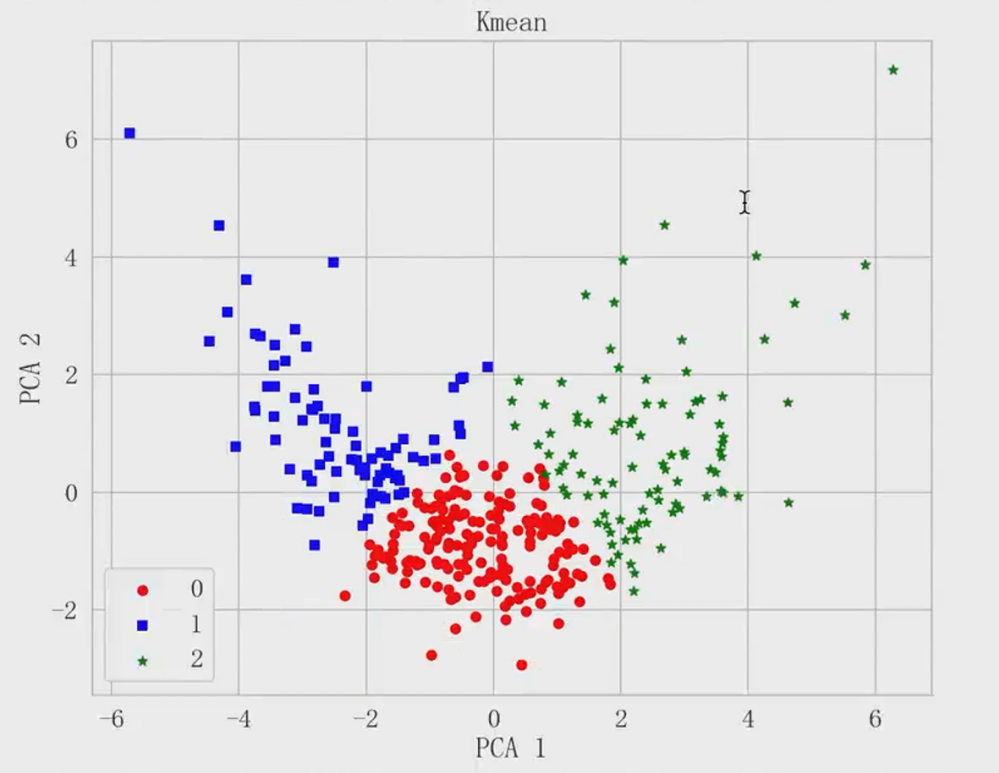

Figure 24: Classification Results

There are clustering models: K-means, KNN, decision trees, and so on. Since we use the weighted average of different tries to represent the difficulty of words, we consider using it as the label of each item. Therefore, we should choose a supervised model. At last, we choose KNN as our model. The classification result is shown in Fig. 24. After inputting the word "EERIE" into the model, the predicted result lies in the red part of this figure, which means it lies in the second part, resulting in the "WA" around 4. According to the resulting probabilities in model 2, the distribution of resulting probabilities is respectively 0.035, 0.052, 0.204, 0.336, 0.230, 0.100, and 0.042. So the result of "EERIE" is 4.15, showing that the result produced by our model is acceptable.

4.4.1 Description of the Model

For KNN, feature extraction greatly influences the accuracy of the model. Firstly, we consider inputting all the related attributes. Then we consider the decisive attributes. The results of choosing different numbers of attributes lead to different results. The result is shown in Fig. 25. When choosing too few and too many attributes, the accuracy of the test set is very low. Therefore, we finally choose 6 attributes: the frequency of the word, the number of consonants in the word, the number of repeat characters in the word, the number of the character appearing in the word “arose”, the number of consecutive characters, and the number of sequences appearing in the frequently seen phonetic.

4.4.2 Parameter Selection

When using KNN, there is a large difference between using percentage to label and using the floor of the number to label. When using percentages, we divide WA into three parts equally. The accuracy of the train set is 0.564, and the test set is 0.612. When using the floor of the number, we set WAs between 3 and 4 to 3, WAs between 4 and 5 to 4, and WAs between 5 and 6 to 5. The accuracy of the train set is 0.729, and the test set is 0.687. The results show that the accuracy of both the train set and the test set of the former is lower than the latter. Therefore, we choose to use the floor of the number to label.

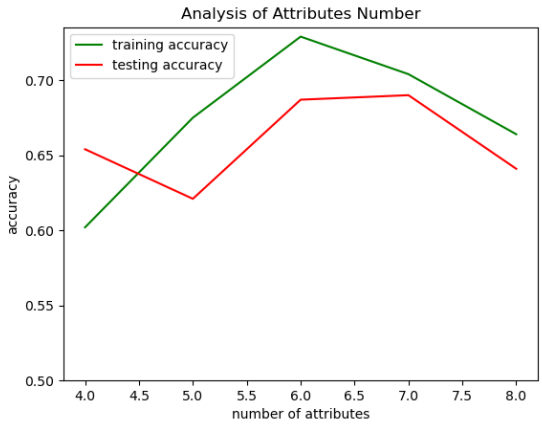

Figure 25: Analysis of Attributes Numbers

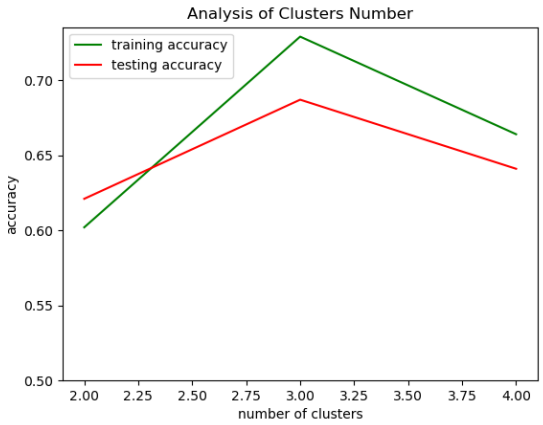

Figure 26: Analysis of Clusters Numbers

The choice of clusters also greatly influences the accuracy of the train set and test set. We consider choosing k to be 2,3,4. The result is shown in Fig. 26. The result shows that k=3 is the best.

4.5 Model 4: Interesting finding-SIR model

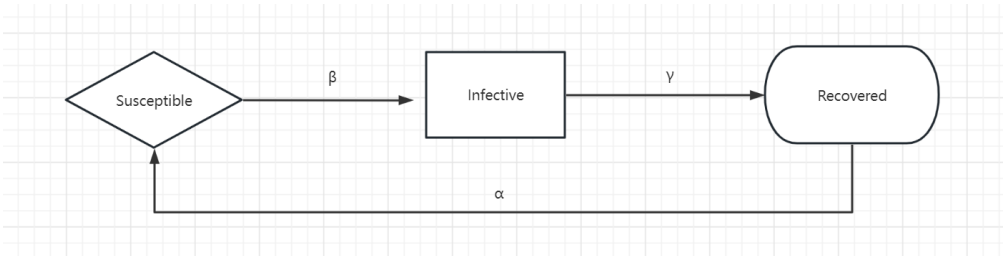

Figure 27: Flowchart

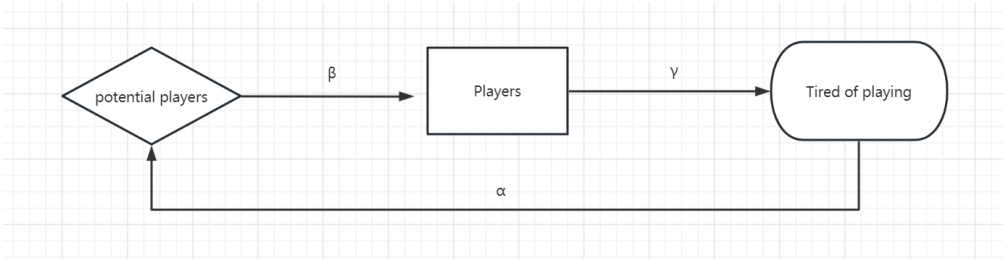

Figure 28: Flowchart

After completing the modeling of the entire process, we found that the data set can be interpreted by a completely different model - the SIR infectious disease dynamics model. β is the infection rate, γ is recover rate and α is re-infected rate. With the same logic, potential players are the susceptible population in the SIR model, the current players are infected group, and players who have played and lost interest are the recovery crowd. The current players send Twitter to infect potential players. Some current players gradually get tired of the game and become recovery crowd. Partial recovery crowd regain interest in the game. Therefore, the spread of games on social media can follow the model of infectious diseases theoretically. The flowchart Fig. 27 and Fig. 28 explain the source of this interpretation.

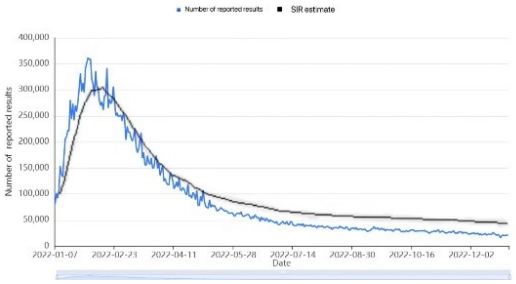

Figure 29: Fitting of the SIR Model

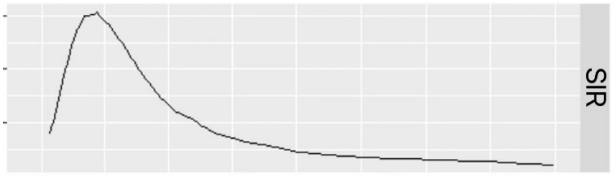

Figure 30: Fitting of the SIR Model

Fig. 29 and Fig. 30 show the fitting of the SIR model to the data of this question.

The prediction generate from SIR model can estimate the Trend component but not explain the excess volatility in Number of reported results. This may be due to the fact that the three parameters(β, γ , α) of the current model are all set as constant. When we incorporate the attributes of words into parameter settings, such as using machine learning to simulate parameter values, theoretically the SIR model can explain the data well. However, due to the limitation of only 300 available data, this task cannot be completed, but we still believe that the SIR model is a more comprehensive model with stronger explanatory power.

5 Conclusion

Wordle game has attracted broad attention in the social media Twitter. In conclusion, the paper has discussed the number of players in the future, effect of word attributes on the percentage of hard mode results, the probability distribution of a five-letter word on a future date and words classification by difficulty related to word contributes. We use the established model to process the given data set and propose many quantitative variables about word attributes.

Model 1 perform STL decomposition on time series data and Granger causality test to test the effect of difficulty of past words on the number of results reported. It was found that the change in the number of reported results per day comes from three parts: 1) Trend: first a rapid rise and then a slow decline 2) Seasonality: seasonal fluctuations in weeks 3) Difficulty of words in the past: the difficulty of words can improve in the future number of reported results. For prediction, ARDL model and ARIMA model were applied to predict the number of reported results on 1st March, 2023. The goodness-of-fit R^2 of the model was 0.982, the prediction model performed well.

Model 2 encodes the word set as high-dimensional feature vectors fundamentally and then set 3 neural network layers to process the data for training and testing. Though Model 2 does not take time series into account, it turns out the results has a high accuracy rate (>= 0.70). We have compared the effect of deep-learning model (Model 2) and LSTM (Long-short Term Model), and the results of model accuracy turn out that LSTM is not better than DL model. After being tested, it proves that this method has a high degree of classification confidence.

Model 3 uses KNN to classify the difficulty level without choosing parameters. This model can make the accuracy of both the train set and the test set larger than 0.70. We consider carefully the construction of characters and words. And we solve the problem from an NLP perspective. We found that the frequency of a word, the number and sequence of the vowel and consonants influence its difficulty level. And the word "EERIE" is classified to the range 4,middle-levell difficulty.

6 Model Evaluation and Further Discussion

- **Time series analysis:**

  a) Limitation: Time series analysis only considers the changes of variables over time, ignoring the influence of external factors on variables.

  b) Advantages: Using statistics to predict the future based on past trends usually conforms to the law of development of things; Besides trend, time series analyze the impact of seasonality and cyclical changes.

- **Principal component analysis:**

  a) Limitation: PCA requires the user to have certain prior knowledge; PCA is the linear sum of the factors so that the pivot components obtained by the PCA method may not be optimal in the case of non-Gaussian distribution.

  b) Advantages: Remove noise and reduce dimensionality of data; There are no parameter restrictions.

- **Deep-learning neural network model:**

  a) Limitation: The available data in the dataset are kind of inadequate for the Deep-learning neural network model to train and test.

  b) Advantages: The model encodes the word set as high-dimensional feature vectors fundamentally and has a high accuracy rate of training and testing model.

- **Clustering analysis:**

  a) Limitation: The accuracy is sensitive to the cluster number and too fewer datas will lead to underfitting, too many features will lead to overfitting.

  b) Advantages: The model needs not to extract the parameters, it will classify implicitly.

A letter to the Puzzle Editor of the New York Times

Dear Sir or Madam:

Kowning that the New York Times has always been interested in puzzle games and had a takeover of Wordle in 2022, we are very glad to share some information about the data of Wordle to give some recommendations.

The number of reported results per day is driven by many factors. First, there is a long-term trend in the number of daily reported results. The game went to a peak around 1st March,2022 and the heat gradually drops. Second, there is a weekly seasonality, which means more people will participate in the game on certain days of the week, which may due to the different game hours available for each day. Third, the difficulty of past words also affects the number of current reported results. The more difficult the past words be, the bigger number of reported results. However, words attributes do not have an effect on the percentage that choose hard mode.

Second, we have discussed several methods to predict the associated percentages of tries for a future date. The conclusion derived from the comparison between methods is that the percentages of tries are only connected with the features of the given solution word, no matter when the word is set as the solution, which is also consistent with our intuition.

Third, we have investigated how to predict the percentage of the hard mode given a word. And the accuracy of our model is relatively high considering the train and test set. And the result can stand the test since we compare the result in model 3 with the predicted result in model 2. One reason that we obtain such a good result is that we consider carefully the construction of characters and words. And we solve the problem from an NLP perspective. Wordle and NLP solutions are both in a frequency analysis approach. And they can both be interpreted in a bayesian probability perspective. We found that the frequency of a word, the number and sequence of the vowel and consonant influence its difficult level.

Considering our observation, to attract more users in Wordle, we have some recommendations. Above all, the decline of the difficulty in the previous day can attract more users in the next day.

We hope that Wordle can really attract more users. In 2016, Puzzle Mania was published. The goal of this crossword puzzle is to reinvent and expand the possibilities of print. And it did it! Nowadays, such kind of games using paper are more and more popular. We believe that Wordle is also one of them. Therefore, we are willing to investigate and dig more wonders under it.

Yours sincerely,

Team 2322627

Tuesday, February 21, 2023

References

[1] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[2] K. Gilhooly and C. Johnson, "Effects of solution word attributes on anagram difficulty: A regression analysis," *Quarterly Journal of Experimental Psychology*, vol. 30, no. 1, pp. 57–70, 1978.

[3] S. Sudholt and G. Fink, "Attribute cnns for word spotting in handwritten documents. ijdar 21 (3), 199–218 (2018)."

[4] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[5] B. J. Anderson and J. G. Meyer, "Finding the optimal human strategy for wordle using maximum correct letter probabilities and reinforcement learning," *arXiv preprint arXiv:2202.00557*, 2022.

[6] D. D. Wickens, R. E. Dalezman, and F. T. Eggemeier, "Multiple encoding of word attributes in memory," *Memory & Cognition*, vol. 4, no. 3, pp. 307–310, 1976.

Appendices

```python
def create_word_embeddings(words, dim = 100, plot = False, highlight_last = False):
  # define the dataset
  data = [[item] for item in words]

  # train the model
  embed_model = Word2Vec(data, vector_size=dim, window=5, min_count=1, workers=4,sg =1)

  # Get the vocabulary and the corresponding embeddings
  vocab = list(embed_model.wv.key_to_index)
  embeddings = embed_model.wv[vocab]

  if plot == True:
    # Use t-SNE to reduce the dimensionality of the embeddings to 2D
    tsne = TSNE(n_components=2)
    embeddings_2d = tsne.fit_transform(embeddings)

    # Plot the embeddings
    plt.figure(figsize=(10, 10))
    for i, word in enumerate(words):
        x, y = embeddings_2d[i, :]
        if i != np.size(words)-1:
          plt.scatter(x, y, marker='o', color='green')
        else:
          if highlight_last:
            plt.scatter(x, y, marker='o', color='red')
          else:
            plt.scatter(x, y, marker='o', color='green')
        plt.annotate(word, xy=(x, y), xytext=(5, 2), textcoords='offset points', ha='right', va='bottom')
    plt.title('Dimension-reduced word scatter')
    plt.show()

  return embeddings
```

Figure 31: Code

```python
model = keras.Sequential([
    keras.layers.Dense(30,activation='relu', input_shape=(embeddings.shape[1],),name='layer1'),
    keras.layers.Dense(10, activation='relu'),
    keras.layers.Dense(7, activation='softmax')
])

# Compile the model with appropriate loss function and optimizer
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

x_train, x_test, y_train, y_test = train_test_split(embeddings, prob_dist, test_size=0.3,random_state = 42)

x_train=np.asarray(x_train).astype(float)
y_train=np.asarray(y_train).astype(float)
x_test=np.asarray(x_test).astype(float)
y_test=np.asarray(y_test).astype(float)

# Train the model on the training set
results = model.fit(x_train, y_train, epochs=30, batch_size=32,
          validation_data=(x_test, y_test))
```

Figure 32: Code