The Prediction Puzzle Problem:
An analysis of Wordle data in 2022

Team Number # 2322072
Problem C

February 20, 2023

Abstract

In this paper two models are presented, one to predict the amount of players who will post their Wordle results to Twitter on a given day and another to predict the average amount of guesses a word will take for users reporting their Wordle results to Twitter. For the prior, a Linear Regression Model, Exponential Decay Model, and Random Walks were implemented in order to predict the amount of players that will upload their results to Twitter on any given day after December 31st, 2022. For the latter we performed thorough analysis of scraped data to determine what factors had a correlation with the average amount of guesses to solve the puzzle, before combining these into a Linear Regression Model. Our models concluded that on March 1st, 2023 with the word "eerie", we will have $17,646$ players reporting their scores on Twitter with an average score of 4.49.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Wordle is an online game, which has gained popularity relatively recently, with the first spike being seen at the start of January, 2022 [10]. In the game Wordle, players try their best to guess a five letter word, in six or fewer tries, with the word changing every day [9].

Once a player has guessed a word the tiles are each assigned a color, with gray indicating that the letter appears nowhere in the correct word, yellow indicating that the letter does appear, but is in a different location, and green indicating that the letter appears and at least one occurrence of it is in that spot, see figure 1.

There is also a hard mode one can optionally select. In hard mode, everything described above still applies, with the added constraint that once one gets a green or yellow tile on a letter, that letter is required to be used in every subsequent guess.

Some players report their results to twitter once they are done. This data was provided to us and is the basis for all following analysis.

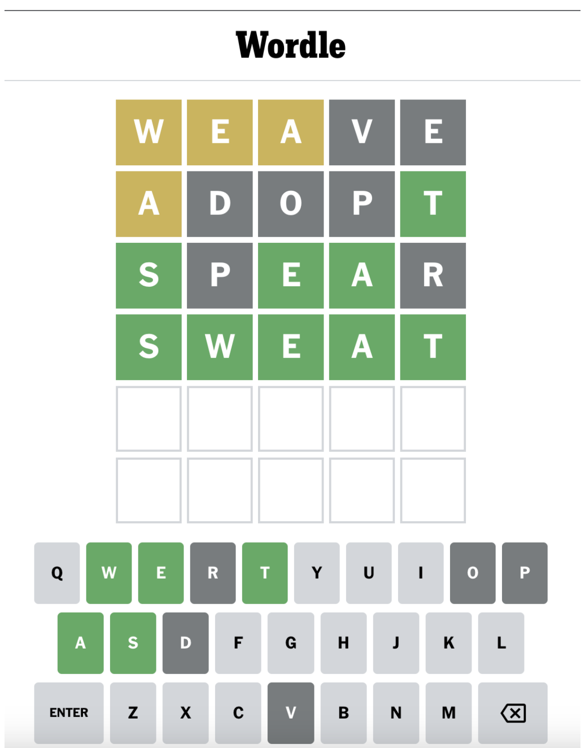

Figure 1: Wordle Demonstration

1.2 Overview

As puzzle games becomes more common to the general public, it becomes more important for developers to have a well defined sense of difficulty for their games. This paper serves to present two models addressing this issue. The first, a model for which to evaluate the difficulty of a word, based on various metrics. The other, to determine the number of players who will submit their Wordle results to Twitter both in general and on March 1st, 2023. These models will allow for both The New York Times and interested players to have more information about the game.

We were tasked with performing four analyses, these being:

(a) Develop a model to explain this variation and use your model to create a prediction interval for the number of reported results on March 1, 2023. Do any attributes of the word affect the percentage of scores reported that were played in Hard Mode?

(b) For a given future solution word on a future date, develop a model that allows you to predict the distribution of the reported results. In other words, to predict the associated percentages of $(1, 2, 3, 4, 5, 6, X)$ for a future date. What uncertainties are associated with your model and predictions? Give a specific example of your prediction for the word EERIE on March 1, 2023.

(c) Develop and summarize a model to classify solution words by difficulty. Identify the attributes of a given word that are associated with each classification. Using your model, how difficult is the word EERIE? Discuss the accuracy of your classification model.

(d) List and describe some other interesting features of this data set.

We will not be doing a deep dive into (a), (b), and (c) here, as they are main subject of this paper and are discussed in great depth later on. However (d) is not discussed elsewhere we will talk about it here. Some interesting features of this data set were:

(1) Wednesdays have the largest total amount of reported results of any day of the week, however are only the largest day of the month a few times.

(2) As the Wordle dictionary is composed of exclusively five letter words, root word analysis is unusable here.

(3) The percentage of hard mode players relative to the total amount of reporting players grows over time. This could indicate that hard mode players are less likely to stop playing Wordle.

2 Parameters

2.1 Assumptions

In order to develop our models in the following sections we made several assumptions, these are listed here:

(1) Players reporting their scores only posted it to Twitter once a day.

(2) Players reporting their scores played to the best of their ability each day.

(3) Their exists a loyal player-base for Wordle.

(4) The Wordle dictionary is accurate and up to date [2].

(5) Date is uncorrelated to difficulty.

2.2 Data Filtering

Before analysis could be performed on this data, we needed to filter and cleanse it. Out of the 360 unique words listed, 6 could not be matched to the Wordle solution set. However, each was a single character shift from an existing word and the correct answer was determined through an internet search. Additionally, 2 number values were determined to be incorrect. On 11/30, the total player count (2569) was an order of magnitude less than any other player count recorded. By appending a 0, the player count is brought in line. Similarly, on 2/13, the number of hard mode players (3249) was roughly a third of the preceding and following days (10343, 9310), while the player count was roughly identical.

Date	Original	Adjusted
12/16	rprobe	probe
12/11	naïve	naive
11/26	clen	clean
10/5	marxh	marsh
4/29	tash	trash
11/30	2569	25690
2/13	3249	13249

3 Score Prediction Model

3.1 Data Trends

To predict the average player score for a given word, we first identified factors that correlated with score. These included ”Letter Score”, ”Chunk Score”, ”Word Frequency”, and ”Unique Letters”, each explained below. No other independent variables were identified that contributed to score. Using these factors, a linear regression model was generated and optimized on the provided data set. Throughout the section, the word ”eerie” will be used as an example and the final estimated score will be predicted.

3.1.1 Letter Score

A correlation was found between the letters in a word and the final score. The average score for words that included a given letter and words that didn't include that letter was calculated and a difference taken. For letters like ”Z” or ”X”, the average word score was significantly lower when included, as shown in the below table. For each 5 letter word, the individual letter scores were summed to find an adjustment for the whole word. For ”eerie”, the total letter score was 0.27.

Letter	Effect on score
z	−0.62
j	−0.41
x	−0.28
s	0.16
i	0.17
t	0.27

3.1.2 Chunk Score

A core piece of our prediction model, the Chunk Score is an analysis of every word available in the Wordle dictionary. The goal is to isolate patterns within words that are correlated with the average amount of guesses these words have taken in Wordle according to the data scraped from Twitter. We start by generating every permutation of every single segmentation of every single word in said dictionary. Next we count the occurrences of each of these permutations among the whole Wordle dictionary. After this is done the permutations and their counts were separated into categories based on length, five and one letter permutations were tossed out of the data set, as they have no meaning not already carried by the word itself or our other metrics.

Now that we have all of our two, three and four letter permutations and their counts, we divide all the counts by the maximum amount of counts observed for the length of permutation. All that is left is to assign each of these permutations as well as their corresponding normalized weights to each word within the Wordle dictionary. Once this is done we sum the normalized counts for each length and assign each of these sums to every word to be passed to the model. For ”eerie”, the chunk scores of length two, three, and four were 1.74, 0.53, and 0.02.

Below is a graph showing the correlation between these permutation scores and the average guesses of words, using Minitab [3].

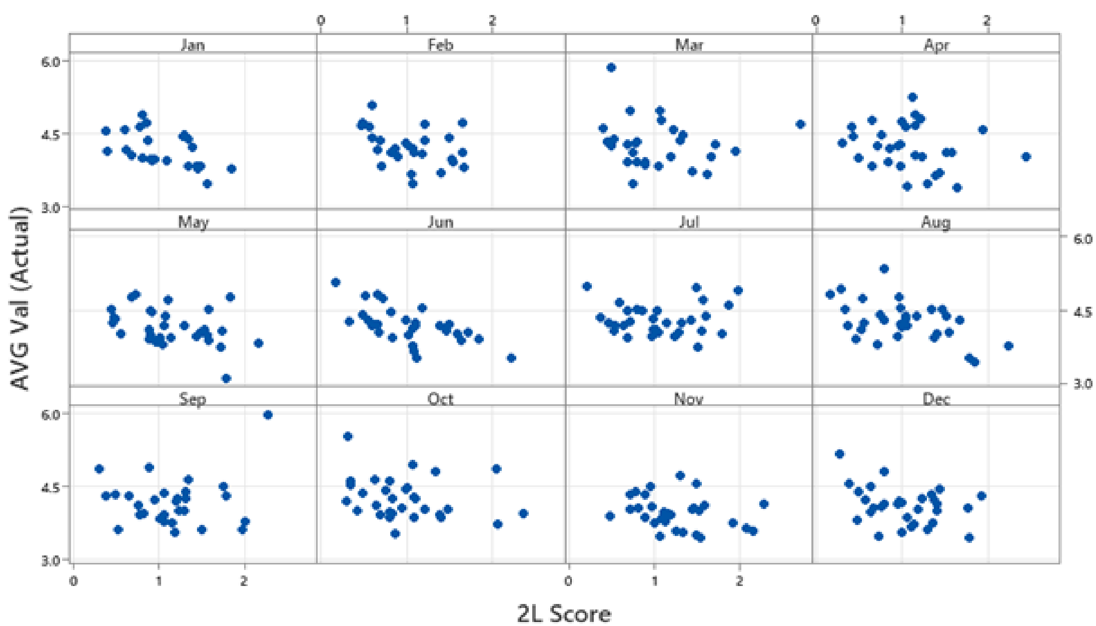

Figure 2: Average word score vs. Two letter permutation sum

3.1.3 Word Frequency

Word Frequency is a measure of the usage of a word. In our case, we used wordfreq to get the frequency of use for a word in the English language. This program scans texts from Wikipedia, Google Books Nrams 2012, OSCAR, Twitter, Reddit, subtitles from OPUS OpenSubtitles, Latex, news reports from NewsCrawl2014, global voices, and a variety of other sources[8].

While there is admittedly low correlation between the frequency of a word and the average amount of guesses said word takes, it does have a impact on words that are either very rare or very common. In order to incorporate this into our model we developed a multitude of strata into which words were placed in accordance with their frequency. We optimized these strata with our model in order to eliminate stratification in the projected vs. actual score plot.

Below is a table with our final decided strata, which serve to model the average human's binning of words based on frequency.

Word Frequency range	Strata Weight
$z < .10^{-5}$	$5 * 10^{-}3$
$z < 5 * 10^{-5}$	$7 * 10^{-}3$
$z < 1 * 10^{-4}$	$8 * 10^{-}3$
$z < 9 * 10^{-4}$	$9 * 10^{-}3$
$z < 5 * 10^{-3}$	$9.5 * 10^{-}3$
$z > 5 * 10^{-3}$	$10^{-}2$

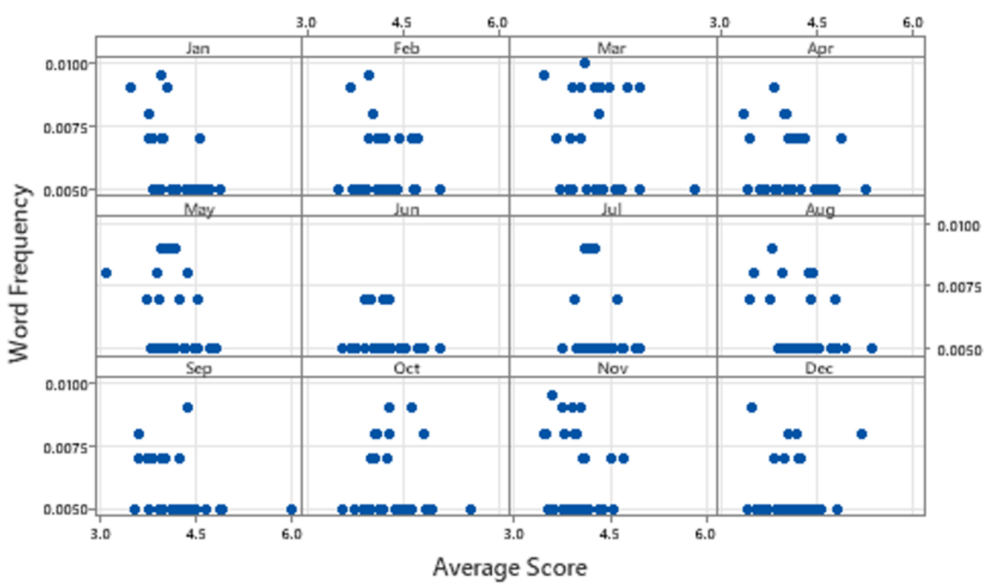

Figure 3: Score vs. Frequency

3.2 Score Prediction Model

Given the above factors, the following equation can be used to predict the average score for a given word,

$$S_P(F, L, C_2, C_3, C_4, U) = (W_1C_2 + W_2C_3 + W_3C_4) \cdot W_4 \cdot F \cdot U + W_5 \cdot L + W_6 \quad (1)$$

Where P_2, P_3, and P_4 are the chunk sub-scores of that respective length, F is the adjusted frequency of the word, U is the number of unique letters in the word, L is the sum of the letter adjustment factors, W_n is the nth coefficient to be optimized, and S_P is the final predicted score.

To create a scoring metric, the above equation was solved for each word in the data set and the absolute value of the difference between the predicted score and the actual score was recorded. By taking the average of these values, the Mean Absolute Error (MAE) was found.

A set of initial coefficients was randomly generated (between -1000 and 1000) and the scoring metric was evaluated before the coefficients were adjusted by a random amount, $-step \leq \delta \leq step$ and the scoring metric was evaluated again. If the new set of coefficients resulted in a lower (less error) score, then the old coefficients were replaced. This process was repeated until it converged and stabilized on a single set of coefficients, with the maximum step size being reduced throughout (starting at 200 and ending at 2). The model reached a minimum after $40,000$ steps in all tested cases.

This process was repeated five times and the optimal set of coefficients in the equation was,

$$S_P(F, L, C_2, C_3, C_4, U) = (38.05C_2 - 20.65C_3 + 13.77C_4) \cdot -2.46 \cdot F \cdot U - 0.224 \cdot L + 5.35 \quad (2)$$

A few values are displayed in the table below (the three highest and lowest scoring words predicted by the model). Generally speaking, the model was quite successful at predicting the actual score based on the six factors. When the predicted and actual score was measured for each word in the data set, the MAE was 0.241 and the Root Mean Square Deviation was 0.327. The mean of the actual scores was 4.19 (SD: 0.404) while the mean of the predicted scores was 4.18 (SD: 0.222). [4, 12, 13, 14]

Substituting the values that we've calculated for "eerie" into this equation gives us,

$$S_P(0.005, 0.273, 1.735, 0.528, 0.023, 3) = 4.49 \quad (3)$$

Word	C_2	C_3	C_4	F	U	L	Actual	Pred.
mummy	0.30	0.04	0.01	0.005	3	−0.27	5.53	4.71
fluff	0.22	0.08	0.01	0.005	3	−0.36	4.98	4.71
vivid	0.48	0.10	0.02	0.005	3	−0.02	4.70	4.68
inter	2.96	1.49	0.56	0.007	5	0.62	4.01	3.27
train	1.78	0.58	0.08	0.008	5	0.70	3.10	3.45
point	1.57	0.35	0.04	0.009	5	0.59	3.47	3.54

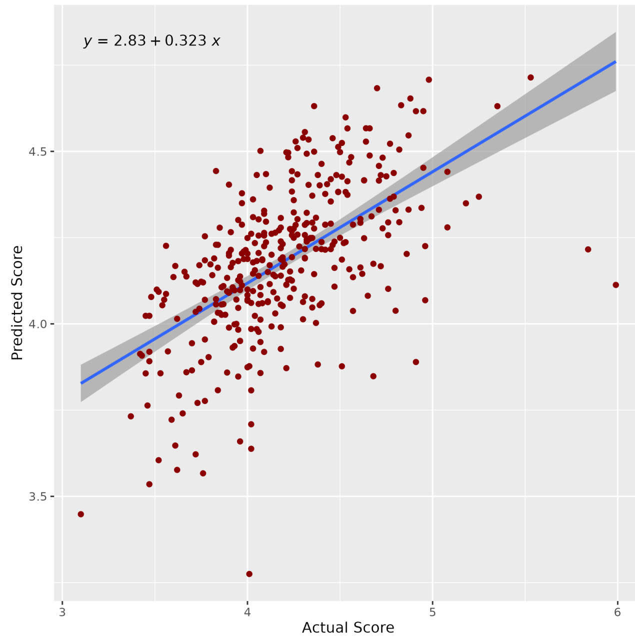

Figure 4: Predicted Score vs. Actual Score

3.3 Distribution Prediction Model

In order to predict the distribution of scores, we relied on the fact that distributions were nearly identical for a given score. For lower scores, the distribution was normal and for higher scores, the distribution was skewed right. Given a score, all words with a similar score ($\pm$.05) were selected and the average of each bucket was calculated. This method does not work for outliers, as there will be less data above the point and more below, skewing the distribution down. For "eerie", there were 26 words distributed evenly within .05 of its predicted score of 4.49. The final predicted distribution is displayed below.

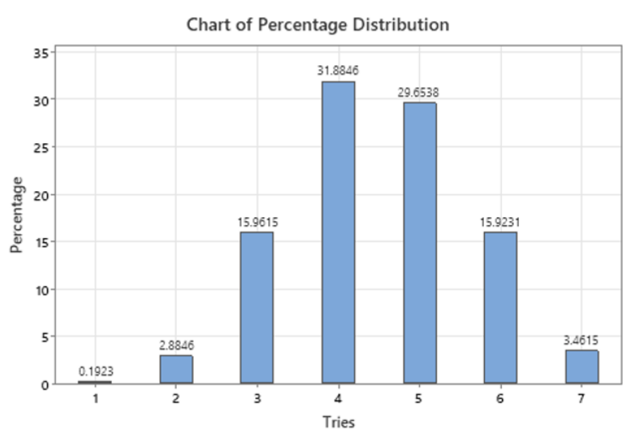

Figure 5: Predicted Score Distribution of "eerie"

4 Player Prediction Model

4.1 Hard Mode Players

One of the main questions that we set out to answer in this paper, is ”what, if any, factors effect the amount of players reporting their scores in hard mode?” and in the end we were able to conclude that the hard mode players as a whole act independently of the many factors that contribute to the difficulty of words.

Below are graphs showing as much,

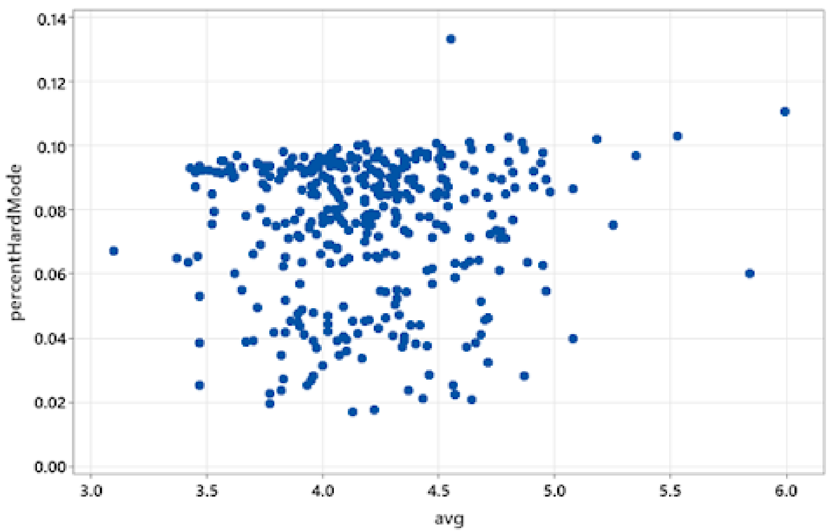

Figure 6: Hard Mode Players vs. Average Reported Score

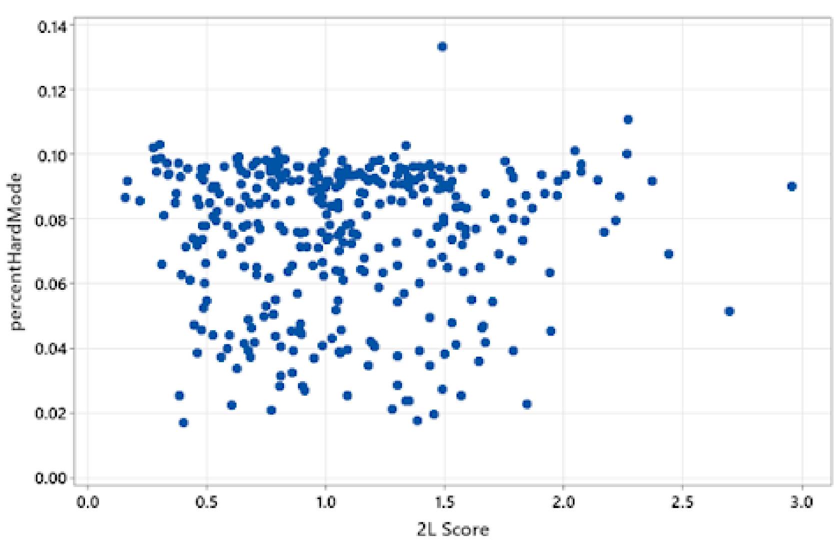

Figure 7: Hard Mode Players vs. 2 letter Chunk Score

4.2 Overview

Our Reporting Player Count Population Model takes advantage of both deterministic and predictive modeling in order to provide both a micro and macroscopic view of the population over time, before combining them to give a full picture of the reporting player count over time. Our predictive model provides insight on the macroscopic view of the population over time using concepts like exponential decay, while our predictive model uses manufactured stochastic fluctuations to simulate the daily variations in player counts. Below we dive into how we made these models and how they communicate to provide a clear understanding of the Wordle score reporting population over time.

4.3 Loyal Players

One of the first challenges one must tackle when creating this type of model is determining the size of the loyal player-base. This is because in order to effectively model our population of reporting players in the long term we must take into account that once only the loyal players remain, the decrease of player reporting size will slow down drastically. In order to determine the size of this group of players, we simply took the amount of players who reported their score on Christmas Day, as only dedicated players would be expected to play then. Once this number was established we plugged this into our established equation for modeling the population of a loyal player-base to get,

$$P_L(t) = \left(\frac{15554}{\log_{10}(t)}\right) \cdot \frac{1}{e^{t^4}}. \tag{4}$$

This model has little relevance in the short term of our population model, however it will dominate the long term behavior of our model. At a glance this may seem counterproductive as our goal with the model only pertains to the short term. While it is true that this model does not predict that data in the slightest, it has a major bearing on our selection of model for which to represent the greater population of players reporting their Wordle score to Twitter.

4.4 Total Player Population Model

We must now create our model for the total reporting population of players. To do so we first smoothed the data provided using a Gaussian Transform with $\alpha = 5$ and window = 45. After that, we isolated longest list of points terminating on December 31st, 2022, that resulted in the lowest overall variation. After obtaining this list of points we optimized for our population curve using Scipy and polynomial regression. Once this was done, we had acquired several functions which we then set equal to $P_L(t)$ as defined above. This allowed us to weed out any population curves with non-realistic results and settle on the following,

$$P_{LT}(t) = 15t^{5/4} - 109 \cdot \left(\frac{t + 174}{1.019}\right) 8t + 39346. \tag{5}$$

One can then see that we have $P_L(t) = P_{LT}(t)$ at $t = 699$ days, a little before two years after December 31st 2022. This indicates that according to our model, assuming no intervention, Wordle will be left with just it's loyal player-base after 699 days.

4.5 Long Term Population Model

To make our long term population model, we created a piecewise function as follows,

$$\begin{cases} P_{LT}(t) & 0 \leq t \leq 699, \\ P_L(t) & t > 699. \end{cases} \tag{6}$$

This model effectively reproduces past trends over the last 60 days of data collection up to the current day as can be seen on google trends.

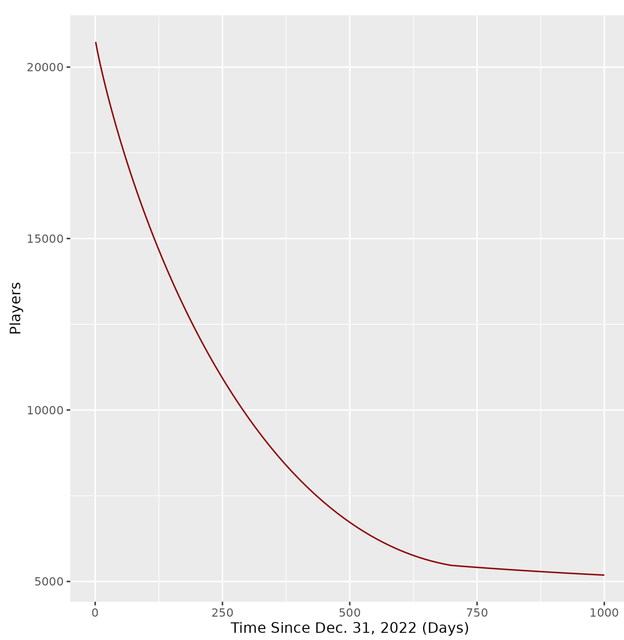

Figure 8: Players vs. Time, Long-term

4.6 Stochastic Fluctuations

After achieving models for both the loyal players and all players over the long term, we next moved onto accurately modeling the reporting population on the microscopic level. To do this we used a random walk with dynamic probabilities. To start we initialized the probability of going up as 40%, and the probability of going down as 60%. Then using a list of the last three walks, or coin-flips, each was assigned a value of 5% which was then subtracted from its corresponding probability. This served two purposes: one, to not allow the probabilities to stabilize, and two, providing movement towards our determined stable equilibrium.

Next, we added in some more forceful methods of moving our random walk towards our equilibrium. Whenever the random walk strayed from our Long Term Population Model by more than 300 people our up and down probabilities would change. If our random walk was 300 above, the probabilities would be shifted by 36%, if it was below they would be shifted by 40%. Further checks are made to ensure that neither of the probabilities are ever greater than or equal to 1.

Then for each time step, after deciding movement direction we would move a random amount of steps between 80 and 150 in said direction, as dictated by our Long Term Total Reporting Population Model, eq.(5) [7], [5], [6].

4.6.1 Justifications

There are several justifications that must be made if we are to accept that this model accurately reflects what is happening in the Wordle reporting population:

(1) Initial Probabilities:

These probabilities were chosen to optimize variance in fluctuations for large t while providing some form of chaos in the short term.

(2) Forceful Shifts:

If we look at the change in players who report their score on a day to day basis and count the consecutive movements in either direction, we rarely see any counts greater than three.

(3) Walk length:

Our walk length is based off of the last 70 days of data provided to us, as this is when the amount of players reporting has begun to stabilize and approach linearity. We saw that when optimizing functions to fit this curve, all of them with low variance had slopes between -80 and -150.

4.7 Complete Population Model

After establishing all the components of our player reporting model, we are able to combine them in order to have a visual metric upon which to evaluate our Stochastic Model. We ran $10,000$ simulations of our Stochastic Model, in order to arrive at an average value of $17,646$ players on March 1st, 2023. This same simulation yielded a minimum of $16,291$ players and a maximum of $18,248$ players.

Below is a graph of our Stochastic Model (Red) and our Long Term Population Model (Blue) over 125 days,

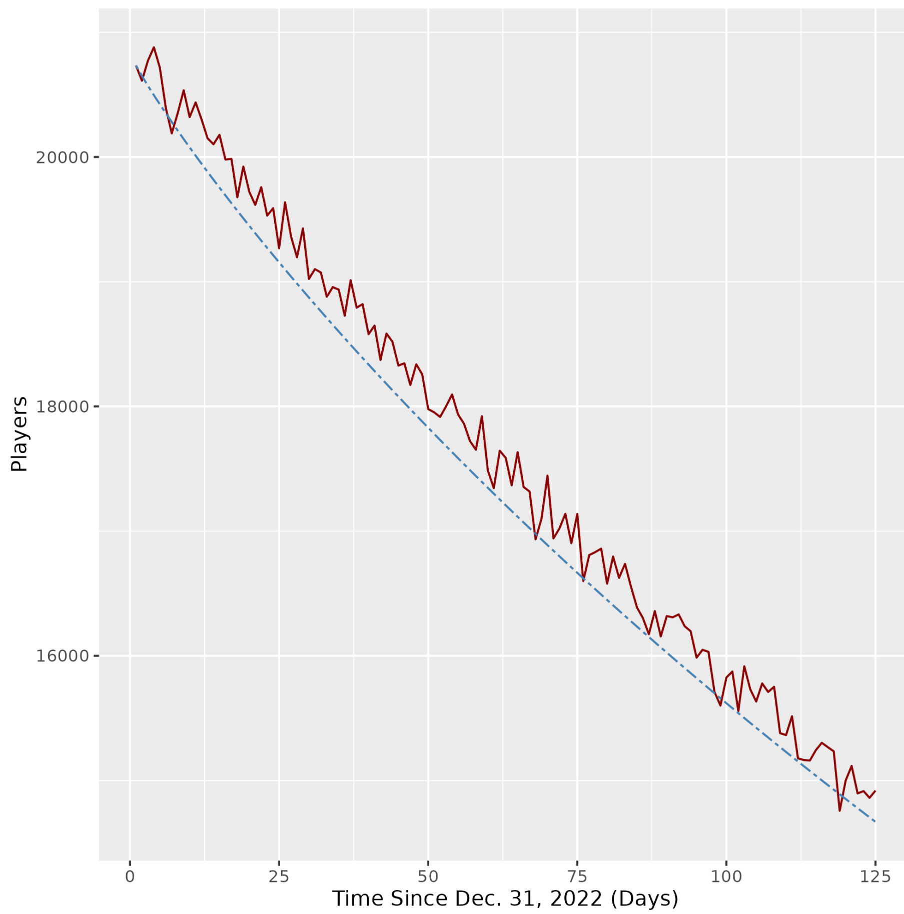

Figure 9: Player vs. Time, Combined Models

5 Limitations

5.1 Strengths

5.1.1 Population Model

(1) **The model does not break down for any time in the future, no matter how far away**. When constructing our Complete Population Model, we did so paying great attention to long-term behavior of our model in order to ensure that the long-term behavior replicated that of games in the past. We did so by looking at games bought from the online retailer, Steam [1], [11]. This allowed us to see how game's player-bases diminish after an initial spike of popularity and provided a way for us to verify our model's accuracy in the long-term.

5.1.2 Score Prediction Model

(1) **The model is easily scalable**. Nothing in this model depends upon words having five characters, so by expanding the data set, the difficulty of words with other numbers of characters could be evaluated. While Wordle currently only has five character words, variants exist that use alternate word lengths.

(2) **The model is efficient**. With only simple computations involved, this model is rapidly computable and can be optimized for speed. This allows it to determine the difficulty of new words quickly and allows for rapid retraining, as the speed of the model directly influences the speed of minimization.

5.2 Weaknesses

5.2.1 Population Model

(1) **The size of the score reporting population is not immune to outside influence**. Our model is unable to take into account outside factors that could contribute to a spike in reporting population in either direction. For instance, this competition has the ability to increase the reporting population by a significant amount, however our model would have no way of knowing this and is only based on the information provided and knowledge of past games and their player-base sizes over time.

(2) **Short term behavior is often chaotic and hard to accurately predict**. While we have great confidence in our model's ability to predict the reporting player population size hundreds of days for now, creating a reliable metric for the short term simply has too many confounding variables to ensure accuracy to a comfortable degree.

5.2.2 Score Prediction Model

(1) **Outliers are poorly estimated**. Both the top end and the bottom end of the model consistently underestimate scores. However, the model preserves order reasonably well, keeping low scoring words low and high scoring words high.

(2) **Word frequency may be disconnected from word knowledge**. The model presumes that the higher the frequency score, the more likely someone is to recall that word and guess it. However, some words are well known, despite being infrequent. ”poise” fell into the lowest frequency bracket, despite being a relatively well known word and resulted in a lower score (3.7) than predicted (4.0). Plural words experience the opposite problem, where ”s” can be added to common four letter words, even if the five letter equivalent is uncommon.

(3) **Over fitting may have occurred**. The model was trained on the entire data set and tested on the entire data set, meaning that memorization could have occurred and success was artificially inflated. Given the small size of the data set, this was hard to avoid, but a portion of the training data (20%) could have been reserved for testing and the rest used to train the model. To ensure that the lack of the reserved data did not affect the final result, multiple models could have been trained using different sets of reserved data.

6 Conclusion

In this paper we predicted the number of players and their success for any given word on any given day. This was done using a hybrid model of exponential decay and linear regression with stochastic fluctuation to predict player population and variation in addition to a linear regression model that predicts the difficulty of a word. We used these models to predict the number of users, range of users, average score, and score distribution for the word ”eerie” on March 1st, 2023. Using the hybrid model, we determined that between $16,291$ and $18,248$ users will post on the date of March 1st, 2023. Using the linear regression model, an average score of 4.49 was predicted for the word ”eerie”, making it one of the more difficult words given to players. Based on a model for predicting score distribution, the percentage of players for each guess was calculated as 1:0.19, 2:2.88, 3:15.96, 4:31.80, 5:29.6, 6:15.90, and 7:7.00. Thus, predictive modeling can be used to successfully predict player success in guessing games.

7 Letter To The Editor

Date:	February 20, 2023
To:	William F. Shortz
From:	Team # 2322072
Subject:	Wordle Difficulty and Player Size Analysis

Mr. William Shortz,

We would like to share with you our models for predicting the number of reported results for a Wordle puzzle on a given day in the future and the difficulty of a word based on the attributes of said word. We created our model using reported scores and hard mode data from Twitter in combination with the list of possible words. Following a peak in February, player population has been steadily declining with significant variation in daily player population. Interestingly, the word of the day had no effect on reported players for that day. Luckily, there is a consistent reporting player-base of about $15,500$ players, which we believe will continue playing. Finally, we developed a model to predict the average score and distribution of guesses for a given word. Using the characteristics of the word and it's frequency in the English language, we can predict the average score. Our average error was only ± 0.13. Using this metric, you can create a difficulty rating for Wordle, allowing players to optimize their game experience and potentially draw more players in. More importantly, our model is easily upgradable and expandable, allowing you to shift its use to other games and enhance player's experience. We hope you will consider our results in the design of your future puzzles.

Sincerely,

Team 2322072

References

[1] Christian Bauckhage, Kristian Kersting, and Fabian Hadiji. "Mathematical Models of Fads Explain the Temporal Dynamics of Internet Memes". In: June 2013.

[2] Tab Atkins Jr. *wordle-list*. `https://github.com/tabatkins/wordle-list`. 2022.

[3] Minitab, LLC. *Minitab*. Version 20.2. Apr. 16, 2021. URL: `https://www.minitab.com`.

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2022. URL: `https://www.R-project.org/`.

[5] Sheldon M. Ross. "10 - Brownian Motion and Stationary Processes". In: *Introduction to Probability Models (Twelfth Edition)*. Ed. by Sheldon M. Ross. Twelfth Edition. Academic Press, 2019, pp. 639–677. ISBN: 978-0-12-814346-9. DOI: `https://doi.org/10.1016/B978-0-12-814346-9.00015-9`. URL: `https://www.sciencedirect.com/science/article/pii/B9780128143469000159`.

[6] Sheldon M. Ross. "11 - Simulation". In: *Introduction to Probability Models (Twelfth Edition)*. Ed. by Sheldon M. Ross. Twelfth Edition. Academic Press, 2019, pp. 679–742. ISBN: 978-0-12-814346-9. DOI: `https://doi.org/10.1016/B978-0-12-814346-9.00016-0`. URL: `https://www.sciencedirect.com/science/article/pii/B9780128143469000160`.

[7] Sheldon M. Ross. "6 - Continuous-Time Markov Chains". In: *Introduction to Probability Models (Twelfth Edition)*. Ed. by Sheldon M. Ross. Twelfth Edition. Academic Press, 2019, pp. 375–429. ISBN: 978-0-12-814346-9. DOI: `https://doi.org/10.1016/B978-0-12-814346-9.00011-1`. URL: `https://www.sciencedirect.com/science/article/pii/B9780128143469000111`.

[8] Robyn Speer. *rspeer/wordfreq: v3.0*. Version v3.0.2. Sept. 2022. DOI: `10.5281/zenodo.7199437`. URL: `https://doi.org/10.5281/zenodo.7199437`.

[9] The New York Times. *Wordle*. Accessed: 2023-02-19. URL: `https://www.nytimes.com/games/wordle/index.html`.

[10] trends.google.com. *Google Trends*. 2023.

[11] Valve. *Steam Store*. https://steampowered.com. Accessed: 2023-02-19. 2005.

[12] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: `https://ggplot2.tidyverse.org`.

[13] Hadley Wickham et al. *dplyr: A Grammar of Data Manipulation*. R package version 1.1.0. 2023. URL: `https://CRAN.R-project.org/package=dplyr`.

[14] Hadley Wickham et al. "Welcome to the tidyverse". In: *Journal of Open Source Software* 4.43 (2019), p. 1686. DOI: `10.21105/joss.01686`.