Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2317059

Wordle – the science backing the puzzle that took the world by storm

Summary

Wordle has been one of the most popular online puzzles in the last couple of years, with millions of players testing their wits every day and sharing their results on Twitter and other social media platforms. In hope of finding answers to some of their questions, the New York Times and MCM, have provided us with numerous data about the players' Twitter reports over the year 2022.

Our first task was solved by developing a SIR model, commonly used to analyze disease transmission, a very useful tool for the pandemic we've suffered. We have reinterpreted this versatile discrete mathematical model as the SPQ model to analyze the variation of player who report their Wordle games on Twitter. In this model, the population is divided into Susceptibles, Players, and Quitters. The number of daily active English-speaking Twitter users is estimated, and less than 10% of them are assumed to be susceptible to getting to play Wordle. The number of players and quitters is calculated using difference equations, and the constants are adjusted to match the reported data. The model can be improved by eliminating some of the simplifying assumptions, and by using a machine learning approach to find the values of our model's parameters.

With regard to the second task, we have developed a model the allows us to classify the difficulty of a word. Analyzing the historical distribution of the reported results for words of the same difficulty, we can estimate the distribution of the reported results of any given word.

For the third task we've developed a model designed to objectively quantify the difficulty of a word in the popular game Wordle, and classify them into easy and hard categories. To achieve this, we needed to prepare the dataset and extract attributes of the words that don't depend on people's interactions. We've also designed a decision tree model that uses a binary tree-like structure to make decisions based on input data. Decision trees were a suitable choice for binary word classification due to their transparency, non-parametric nature, flexibility, and ability to handle missing data. The model can be further improved by moving from a binary tree structure to a multiclass structure, or by even finding a continuously adapting model that assigns a difficulty ranking to each work.

Together with our initial data analysis and pre-processing, we have tried to emphasize some interesting features of the original data set. We've corrected assumed typos and tested our models to confirm our intuitions. For a better understanding of the data, we have provided numerous graphs and charts that clearly outline our findings.

Last but not least, we have summarized our findings in a letter to Will Shortz, the New York Times puzzle editor.

Contents

1 Introduction

1.1 Problem background

The news outlet New York Times has been hosting a daily puzzle game called Wordle, attracting millions of users in the past year. The game received a lot of media coverage and made many news headlines, growing quickly in popularity. Its rules are simple: players have six guesses at most to find the word of the day, and after each try they receive information about the letters of their guess (green means that the letter is contained in the secret word and it's in the right position, yellow means that the secret word contains the letter but in a different place, and lastly grey means that the letter is not in the solution). In addition, the game has a Hard Mode that forces players to use a letter in their guesses, if that letter was orange or green in a previous guess. For both playstyles, users can share their solutions on Twitter, including the number of attempts they needed to find the secret word.

1.2 Restatement of the problem

After almost a year of recorded data (7th of Jan, 2022 - 31st of Dec, 2022), we are required to do the following:

1. Develop a mathematical model to forecast an interval of reported results on the 1st of March 2023 for both Normal and Hard mode.
2. Build a model that predicts the distribution of reports over the 7 possible guess counts 1, . . . , 6, X, and discuss its accuracy.
3. Create a model that assigns a difficulty rating to each word and assess its limitations.
4. Test the two previous models on the word "EERIE"
5. Showcase other interesting attributes of the dataset.

1.3 Our Approach

Being tasked to analyze data provided by the New York Times provided the dataset, we've divided and tackled all problems with different strategies optimal for each model. We solve problem 1 by using an adaptation of the SIR Model for infectious diseases. Tweaking the parameters, we have obtained a good approximation of the interval of submitted results submitted after the inflection point of the trendline. After that, we elaborated a model to estimate the percentage of Hard mode submissions.
To predict the distribution of reports, we have classified the words based on difficulty. For the last model, we create a model, based on supervised Machine Learning, to classify solution words according to their difficulty level, assigning each word a label in a binary matter. In the process, we identified certain characteristics for every word, that would train a Decision Tree. Then we applied the model to determine the label for the word "EERIE," and, finally, we evaluated the accuracy of your classification model, by diving the preexisting dataset into training and testing data.

2 Assumptions

In order to be able to approach the problem effectively we considered a number of assumptions that reduce the complexity and facilitate our models. They are as follows:

Assumption 1: The population remains constant throughout the interval in which we collect and predict the data. *We parametrize the population in our model, as we consider that the relevant sample (potential player base) is aged between 18–65. To simplify computations, we will assume that deaths in this age group are negligible.*

Assumption 2: Once people quit the game, they do not play it again. *Similar to the classic SIR Model for Infectious Diseases, we'll consider that the Recovered (quitters in our case) are not going to return as potential players. For a more accurate prediction of the future number of reports, we've renounced this assumption in Section 5.5.*

Assumption 3: Only some Twitter users are susceptible to Wordle. *We assume that among the Twitter users, some of them have no interest in any kind of game. Thus, those people won't be affected by the spread of Wordle's popularity.*

Assumption 4: Regular users tend to start with the same word sequence. *Some words provide more information than others and thus constitute the foundation of some players' strategies. It is reasonable to assume that most players will start with the same word, or word pairs (not the case for Hard Mode players).*

Assumption 5: Players who play the game in Hard Mode do not quit. *People who switch to Hard Mode, and choose to share their results are prone to get "addicted" to the game, or at least be a lot more devoted to it than casual players. Thus, it's safe to assume that the rate of the players who quit the game from hard mode is negligible*

3 Notations

The notations used throughout the paper are listed in **Table 1**.

Symbol	Definition
$S(t)$	number of susceptibles at time t
$P(t)$	number of daily active players at time t
$Q(t)$	number of people who stopped playing Wordle at time t
α	game spreading factor
γ	rate at which people start playing in hard mode
q	percentage of people who quit the game every day
i	speed of the spread of the game
g	number of letters that turn green
o	number of letters that turn orange
S_G	percentage of short games
L_G	percentage of long games
$DiffLevel$	observed difficulty level scale based on people's performance
W	a word
$W(i)$	letter at position i in W
$L(W)$	length of W
$F_L(ch,\ pos)$	frequency of a letter CH at position POS
$FreqSum$	sum from I to $L(W)$ of $F_L(W(1), i)$
$F_W(W)$	frequency of W (logarithmic scale from 1 to 8)
$DLet$	weight of words with 2 identical letters at position $pos1$ and $pos2$
$TLet$	weight of words with 3 identical letters at position $pos1$, $pos2$ and $pos3$

Table 1: Notations

4 Data Restructuring and Analysis

4.1 Data Cleaning and Corrections

Parsing through the official data in the table "Problem_C_Data_Wordle.xlsx" we found different inconsistencies and possible errors. For example, on day 545 in the original dataset, it reported that the word of the day was "rprobe", which is not only nonsensical but also not the right length. It is safe to assume that the first "r" is a typo, and the solution that day was "probe". Nevertheless, we will analyze with our Report Distribution Model (Section 6) if the word "probe" fits within the data collected that day. Additionally, on day 540, the word of the day is spelled "naïve", yet in our processed dataset, as well as in our models, we used the spelling "naive". Lastly, on day 529, there's been an abnormally low number of reported results. Since we strongly believe that this is not the actual aggregate, we will use the SPQ Model to derive an approximation of the actual number of players that reported their scores.

4.2 Data Analysis and Additional Observations

One of the criteria we will consider when assigning a difficulty rating in Section 7 is the frequency of a particular character in the previously reported words. We summarized the findings in the letter cloud below, becoming immediately apparent that vowels dominate the list, while letters like "z", "j", and "q" have next to no occurrences.

Figure 1: Letter cloud of the most used letters in the secret words Source: https://www.wordclouds.com/

Besides the corrections made in Section 4.4.1, we have identified several additional pieces of information about the dataset. There's a clear correlation between some days of the week and the number of reported results on Twitter. After numerous Sundays, there appears to be a spike in the number of reported results on Monday, graphed below in Figure 2 for the 1st, 10th, 16th, and 17th week.

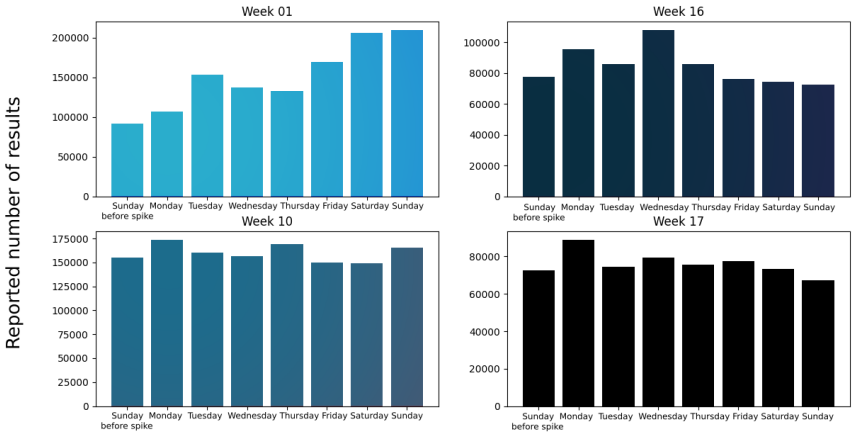

Figure 2: Monday spikes after Sunday troughs

Another connection was found going from one week to another, which seems to coincide with the general usage of Twitter on different days (resource). According to the research, Twitter users are most active on Wednesdays, and Fridays, but least active on Sundays, which is represented in Figure 3.

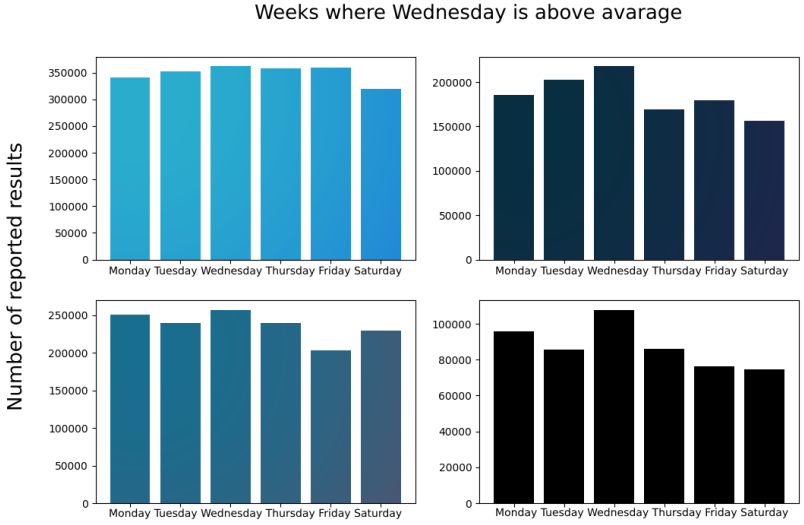

Figure 3: Wednesday spikes

One last observation was made after developing the decision tree for the binary classification of words. There appears to be a spike of reported results the day after a ”hard” word represented in Figure 4, which could tell us that more people are intrigued after a challenging word.

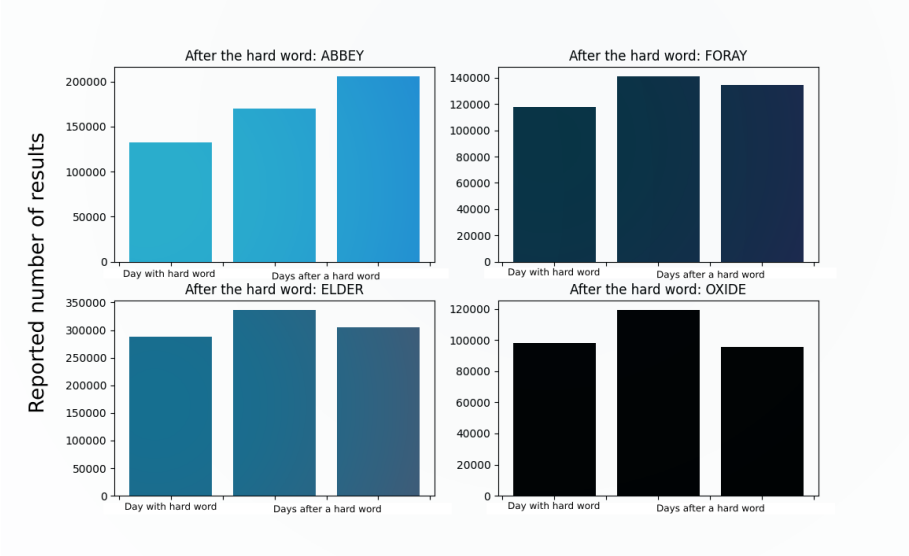

Figure 4: Stylised SPQ Model

5 The SPQ Model of Wordle Reports

Since Wordle was popularized by social media websites (Twitter, Instagram, YouTube), its growth and subsequent drop in popularity can be compared with that of a viral infection. People share their results on Twitter, and a certain fraction of their network interacts with the post, increases exposure, plays the game, shares on their own timelines, and so on.

5.1 Primer on the SIR Model

The SIR Model is a popular compartmental model of disease transmission. It has been especially useful to analyze the Covid-19 pandemic, but as we will point out below, its use is not limited to describing the spread of diseases.
The model works as follows; the population is split into three categories: Susceptibles (S), Infected (I), and Recovered (R). There are many variations of the model, but in the most basic one, people subsequently move from being susceptible to being infected to finally getting recovered. In the classical model, the time-window in which the pandemic takes place allows for the assumption of a constant population. Similar to *Assumption 2*, in the simplified model, people who get recovered from the infection develop immunity and cannot go back to the S compartment. Furthermore, infected people that die will be counted as recovered people, as they cannot fall into the susceptible category. This reinforces the constant population assumption.
With these assumptions in place, we have the following fundamental equations:

$$N = S(t) + I(t) + R(t)$$

Taking the derivative of this equation we obtain:

$$\frac{\Delta N}{\Delta t} = \frac{\Delta S}{\Delta t} + \frac{\Delta I}{\Delta t} + \frac{\Delta R}{\Delta t}$$

But from our assumptions, the population remains constant, thus:

$$0 = \frac{\Delta S}{\Delta t} + \frac{\Delta I}{\Delta t} + \frac{\Delta R}{\Delta t}$$

Now, we can create system of three difference equations:

$$\frac{\Delta S}{\Delta t} = -rS(t)\,I(t)$$

$$\frac{\Delta I}{\Delta t} = rS(t)\,I(t) - \sigma I(t)$$

$$\frac{\Delta R}{\Delta t} = \sigma I(t)$$

5.2 The SPQ Model

The S-P-Q model, which stands for Susceptible, Players, and Quitters, is a reinterpretation of the SIR model.

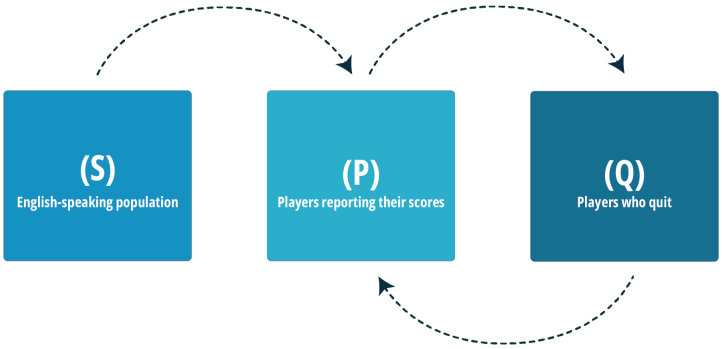

Figure 5: Stylised SPQ Model

S stands for **Susceptibles**; In order to estimate the number of people that could start playing the game at any point in time, it is essential to estimate the number of twitter users. We assume that the total daily active Twitter users are 225 million. To estimate the number of users that can speak English, we multiply the number of non-American Twitter users (150 million) by the percentage of English speaking population outside of USA (10%). Thus, the number of English-speaking Twitter daily users is 90 million. It is unlikely, however, that everyone of them could potentially start playing Wordle. In fact, according to assumption 3, some people have no interest in playing the game. We assume that less that 10% of English-speaking users is susceptible to Wordle, and the total number of susceptible people when the game was lunched is 7.5 million. From now on, we will refer to this data as the **initial population**.
P stands for **Players** who reported their score on Twitter. As of now, we won't make any distinction between players who report games in Normal Mode, or Hard Mode.
Lastly, **Q** is the number of people who **quit** the game. Differently from the SIR model is that we assume that the number of total susceptible people can vary overtime. This is reasonable, since the timeframe considered is larger than one year, and some people will have started/stopped using Twitter in this period.

In order to match the predictions of our model with the data, we have set $S_0 = 6,977,400$. It is important to note that in contrast to the classical SIR model, S_0 in our case does not coincide with the maximum value of susceptibles. Since our data collection begins 202 days after the game was published, many players will have already started playing in this period. For the same reason, we have set $P_0 = 184,720$ and $Q_0 = 436,590$.
Although P_0 is much higher than the number of reports reported in the dataset (80, 630), this difference does not affect the overall accuracy of our model.

The first differential equation is:

$$\frac{\Delta S}{\Delta t} = -\alpha \frac{\Delta S}{\Delta t} \frac{\Delta P}{\Delta t}$$

where α represents the rate at which the number of players spread. It follows that, as more people start playing the game, the number number of susceptible people decreases.

The second differential equation used is:

$$\frac{\Delta P}{\Delta t} = \alpha \frac{(1500 - t)}{1400} S(t) P(t) - qP(t) + \gamma [S(t) + Q(t)] - it$$

The first part of the equation shows the relation between the number of new players every day with the number of susceptible people. As the number of susceptible people decreases, so will the daily number of new players.
Differently from the SIR model, due to the boom of Wordle on social media and due to the effect of sponsored articles written on this game, its popularity drastically increased in the first period. Thus, we have made the number of new players very quickly during the spike in popularity. On the other side, as the popularity of the game decreases, variation in additional players decreases too.
The second element of this equation takes into account q, the quitting rate, that is the percentage of people that stop playing the game every day. The greater this percentage is, the more players will quit the game every day.
The third element of the equation describes a phenomenon that we have not introduced yet, and that we will explain in the following section.
The final element of this equation describes an important social phenomenon. As time passes and the popularity of the game decreases, some people will get tired of publishing their Wordle scores on Twitter. They may even continue playing the game, but this will not appear in our data.
The last differential equation is:

$$\frac{\Delta Q}{\Delta t} = qQ(t) - \gamma Q(t)$$

This equation is almost self-explanatory. It takes into account the players who quit every day as well as the people that re-start playing the game.

It should be noted that the sum of susceptible people, active players, and quitters is not necessarily equal to the **initial population** that we have estimated. This is because, as we have explained above, we assume that during the lifetime of Wordle, some people will start using Twitter, while others will stop using it.

In order for our model to closely match the actual data, we have set:

$$\alpha = \frac{2^{-7}}{10.7}$$

$$\gamma = \frac{1450}{7500000}$$

Note that the denominator of γ is equal to the **initial population**.

$$q = 0.113$$

$$i = 0.9$$

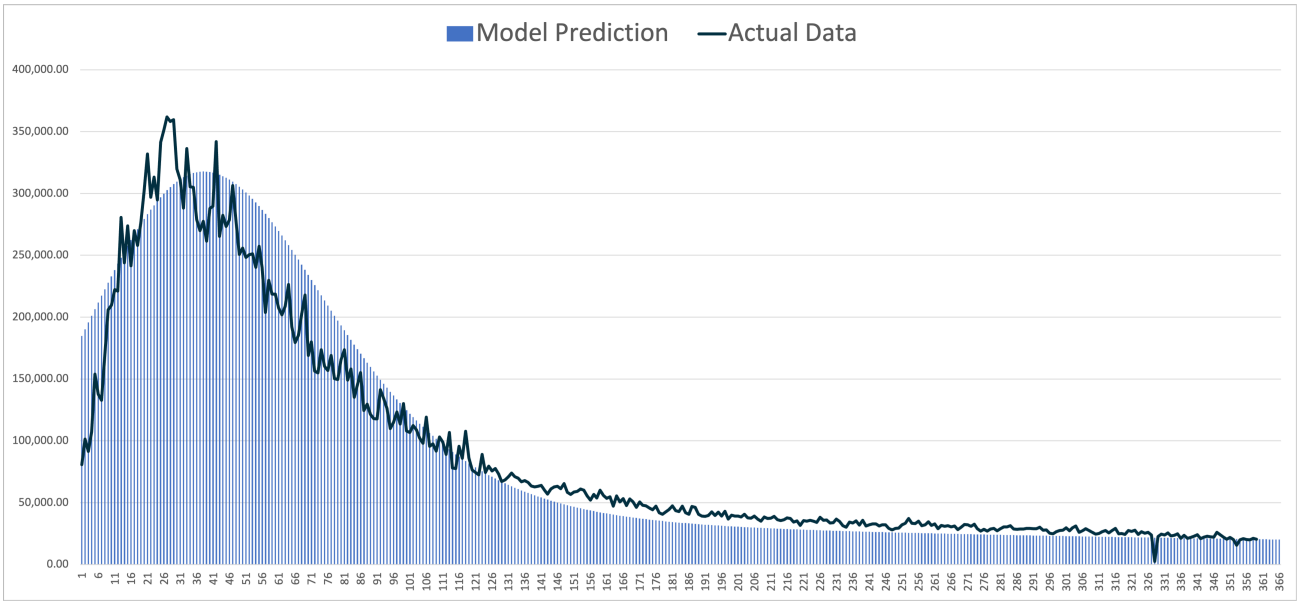

Figure 6: Model Prediction vs Actual Data

Comparing our model to the actual data, we can appreciate how a variation of the SIR Model can accurately describe a phenomenon that at first impact is very different from the spread of a virus.

5.3 Hard-Mode Reports Model

As of now, we have always assumed that all of the active players are using the same versions of the game. However, we know that some of them are actually playing the hard mode. As anticipated above, this idea is the base of the second differential equation of our model

$$\frac{\Delta P}{\Delta t} = \alpha \frac{(1500 - t)}{1400} S\,(t)\,P\,(t) - qP\,(t) + \gamma\,[S(t) + Q\,(t)] - it$$

The third element of this equation, $\gamma[S(t) + Q(t)]$ is strictly connected with one of our assumptions: people who play in hard mode, will not quit the game. This is because people playing in hard mode are very passionate about the game, and are usually not casual players. In addition to this, many of them will find it exciting to publish their scores every day and compare them with the average reported scores. We have assumed that some players start playing the normal mode, and then - after some time - start playing in hard mode. Some other people will start playing in a hard mode right away. We imagine that the people that fall under this category are already playing similar games, and thus are familiar with the mechanics of Wordle (and won't stop playing the game anytime soon).
In order to estimate the percentage of people playing in hard mode in day t_0, we have considered the following equation:

$$\frac{-\frac{1}{5}t_0 \ln P(t_0) + 1.14}{13} + 0.02$$

This equation shows both the flow of people that from "normal" players start playing the hard version, and the number of people that start playing the hard version right away.

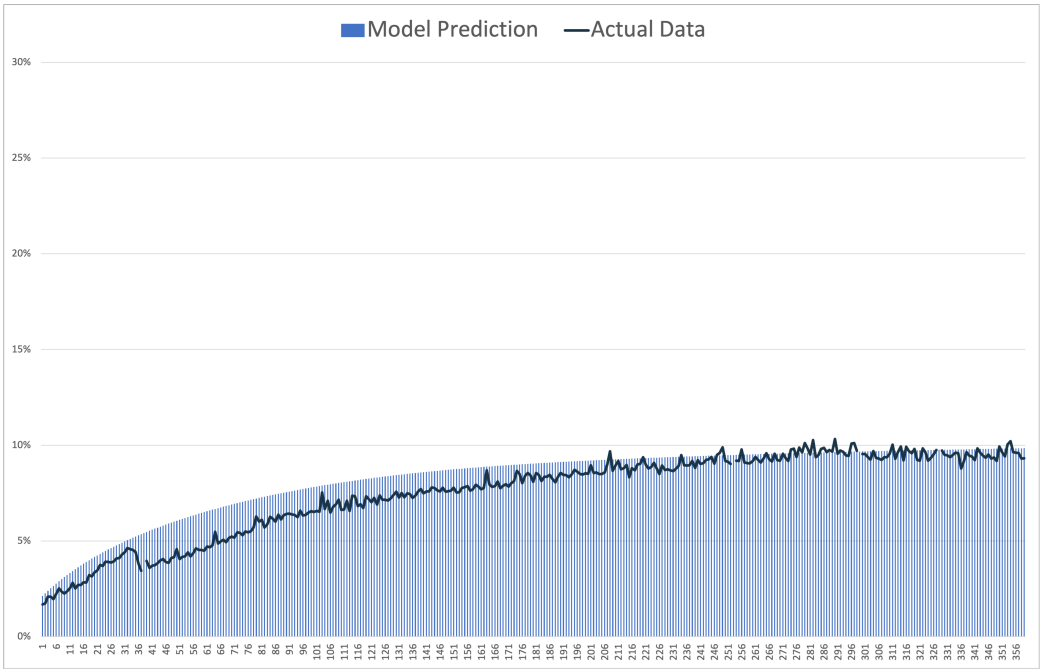

Figure 7: Model Prediction vs Actual Data

The above graph shows the comparison between our model and the actual data (that has been properly cleaned).

5.4 Predictions and Accuracy

In order to predict the future number of players, it is essential to know the accuracy of the predictions of our model.
In order to assess it, we have taken into account our $P(t)$ function. We have computed the percentage difference between our estimates and the actual data. Making an average of those percentage variations, we get -3.3%. If, instead, we use the absolute value of the percentage variation, the result is 20.0%. According to our model, the number of reported results on March 1^{st} will be $18,325$. We expect the actual data to be largely between $14,660$ and $21,990$. Note that in order to compute this range of value, we are having a very pessimistic approach.

5.5 Further Developments

The model that we have provided above, despite being able to accurately resemble the actual data provided, is far from perfect.
The graph below shows one way in which our model could be improved.

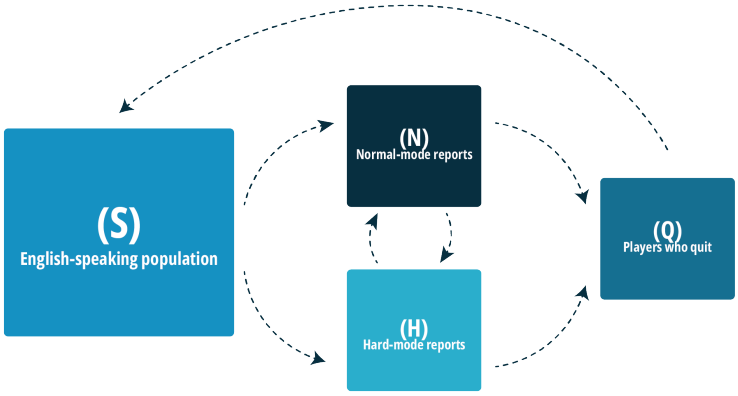

Figure 8: Development of the Model

In fact, we could build our model such that both players that play the normal and hard mode are taken into account.
In this more elaborate model, we assume that the players who quit the game can become susceptible again, and could eventually start playing again.
We also take into account that also players that use the hard mode can stop playing or posting their results on Twitter.

6 The Report Distribution Model

6.1 Basis of the CRANE Model

In this section, we will provide a model that allows us to associate to each word a level of difficulty. In particular, we have assumed that people always start the game using the same word. We have taken "**CRANE**" as a reference word.
Assuming that everyone starts the game typing "CRANE", we can predict - for a given word - what will happen in the game. Indeed it is easy to compute how many letters of the word "CRANE" perfectly match the ones of the mysterious word (i.e. which letters will become green). Through the same method, we can compute the number of letter of the word "crane" that will turn orange.
We will use this mechanic to estimate and classify the difficulty of each word.
The function

$$h(W) = g + \frac{1}{2}o$$

has this purpose. After typing "CRANE", we will compute the difficulty of the word by summing the number of letters that turn green with the number of letters that turn orange.
So, if the mysterious word is "**PAUSE**", the letter "E" will turn green, while the letter "A" will turn orange after typing "CRANE". It follows that the difficulty level of "pause" is equal to

$$h(pause) = 1 + \frac{1}{2} = 1.5$$

It should be noted that the value that the function h associates to a word, is disproportional with respect to the difficulty of the word. So, an extremely difficult word has a score of 0, while a relatively easy word has a score of 2.
Following this method, we have divided the words in six classes, according to the value of $h(W)$:

$$extremely\ hard \iff h(W) = 0$$

$$very\ hard \iff h(W) = 0.5$$

$$hard \iff h(W) = 1$$

$$medium \iff 1.5 \leq h(W) \leq 2$$

$$easy \iff 2.5 \leq h(W) \leq 3$$

$$very\ easy \iff 3.5 \leq h(W) \leq 4$$

$$extremely\ easy \iff 4.5 \leq h(W) \leq 5$$

The maximum value that $h(W)$ can assume in theory is 5. In reality, only 2 of the words included in the data-set have a score above 3.5 (and none has a score of 5).

6.2 Predicting the Distribution of EERIE

We can now use our model to predict the difficulty of the word "**EERIE**". Although - at a first sight - this word could look easy to guess, it turns out that this is far from true.
In fact, "EERIE" contains the letter "E" three times. Typing the word "CRANE", would leave the player with only one green letter, and no other indications. Thus, the value that our function assigns to "EERIE" is 1 - making it an **hard** word.
Whith those notions in mind, we can now find a set of words that are similar to "EERIE" (i.e. have the same difficulty level). We will assume that the distribution of the number of tries to guess the word "EERIE" is similar to the one of any word with the same difficulty level.
Thus, we have graphed the distribution of the number of tries of the words.

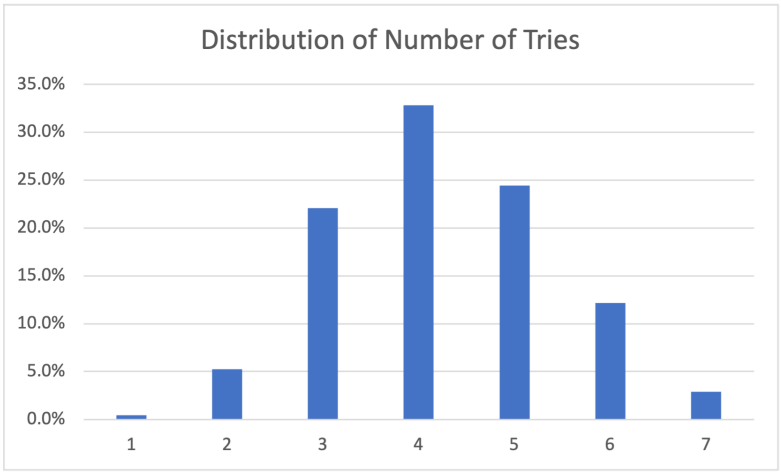

Figure 9: Model Prediction vs Actual Data

According to this model, 0.4% of players will get the word right at the first attempt. Similarly, 5.2% of players will win with 2 tries, 22.0% with 3 tries, 32.8% with 4 tries, 24.4% with 5 tries, 12.2% with 6 tries, and 2.9% will loose the match.

7 The Difficulty Rating Model

Being asked to find a way to classify current and future solutions by difficulty, many challenges arise concerning the subjectivity of the topic such as some common day-to-day words performing worse than other obscure and uncommon words. While analyzing data and possible ways of computing a way to catalog words, some steps needed to be tackled first.

7.1 Cleaning up and preparing the dataset

As pointed out in Problem Updates and Notes for Problem C, some of the data had entry errors, such as misspellings, character bugs etc., but this is normal and expected when working with real-world data and can be easily fixed by python scripts designed to catch errors. After the clean-up, we needed a way to rank past solutions by their difficulty. The way to do that is by analyzing the performance across users when they got that specific secret word. We will define "short games" as games solved in 3 or fewer tries and "long games" as games solved in 6 guesses. Thus, we needed to have a quick look at the percentage distribution of how many tries were needed for each word. After, it's reasonable to assume that "hard" words will have more "long games". This leads to the difficulty rating based on the distribution of "short" and "long" games as

$$S = (\%1t + \%2t + \%3t)$$

$$L = (\%6t)$$

$$\longrightarrow \text{DiffLevel} = \frac{S}{L} - 1$$

This gives us a way to objectively quantify a word's difficulty and to assign labels based on DiffLevel > 0 is a HARD word (label 0) and ≤ 0 is an EASY word (label 1). Some attributes of the word itself that don't depend on people interacting with the Wordle game, are also needed at this stage. Such attributes are letter frequency in the English dictionary, letter distribution among positions in a word, word frequency in the language, weights for words that have repeating letters (double letters, triple letters), and length of words.

7.2 Decision Tree Modelling

7.2.1 Definition

A decision tree is a machine-learning model that uses a tree-like structure to make decisions based on input data. The tree is composed of internal nodes, representing a decision based on a feature or attribute, and leaf nodes, representing an outcome or class. To build a decision tree, the algorithm starts with the entire dataset and recursively partitions it into smaller subsets based on the feature that provides the most information gain. The process continues until a stopping criterion is met, such as reaching a maximum tree depth or a minimum number of samples in a node. Once the decision tree is built, new instances can be classified by traversing the tree from the root node to a leaf node, following the decision path based on the values of the input features. The final classification is the label associated with the reached leaf node. Decision trees can be used for both regression and classification tasks, and they have the advantage of being easy to interpret and visualize. However, they can suffer from over fitting and instability if the tree is too complex, or the training data is noisy/biased.

7.2.2 Reasoning behind our choice

Decision trees are a suitable choice for binary word classification such as our model for several reasons: Transparency and interpretability: Decision trees provide a transparent and easy-to-interpret model that allows researchers to understand how the classification is being performed. This can be important for our model because it's easy to visualize its choices in real time (Fig 10) Non-parametric: Decision trees do not make assumptions about the underlying distribution or structure of the data, which can be useful in exploratory studies where the data characteristics are not well understood. Additionally, decision trees can capture nonlinear relationships between the features and the target variable. Flexibility: Decision trees can handle both categorical and continuous features, which is useful for scientific studies that may involve a mix of different types of features. Ability to handle missing data: Decision trees can handle missing data without requiring imputation or modification of the original data. This can be beneficial in our case, because the available dataset is small and for future guesses (i.e., the word EERIE) where some missing values are expected. In summary, decision trees were a perfect match for our model and give us a transparent and efficient method for the word binary classification.

7.3 Accuracy

No model is perfect and neither is ours. Using an "entropy" based algorithm (fig...), with a maximum depth of 3 levels, we get an approximated 86.1% accuracy in our data set. Of course, with adjustment to some parameters such as maximum depth, entropy weight, etc., the model will become more accurate, but we also wanted it to be easy to use and understandable for us.

Date	Contest number	Word	Number of reported results	Number in hard mode	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 or more tries (X)	<3 tries	DifLevel	FreqSum	DLET	TLET	LEN	FreqLang	F1	F2	F3	F4	F5	LABEL
04.10.2022	472	bough	32014	3060	0	3	17	35	28	13	3	20	-0,35	0.37326	1.0	1.0	5	1.73	0.05571	0.14206	0.07799	0.04457	0.05292	1
05.10.2022	473	manxh	30935	2885	0	9	30	35	19	6	1	39	-0,8461538	0.31198	1.0	1.0	5	-1.0	0.05571	0.13649	0.0585	0.00836	0.05292	1
06.10.2022	474	sloth	32522	2987	1	10	38	34	13	3	0	49	-0,9387755	0.52368	1.0	1.0	5	2.1	0.14206	0.11699	0.13649	0.07521	0.05292	1
07.10.2022	475	dandy	29026	2840	0	2	11	23	29	24	11	13	0,84615385	0.39276	3.88932	1.0	5	2.25	0.03343	0.13649	0.06407	0.02228	0.13649	0
08.10.2022	476	vigor	26905	2642	0	2	15	35	31	14	2	17	-0,1764706	0.25348	1.0	1.0	5	2.03	0.01671	0.06407	0.02507	0.05014	0.09749	1
09.10.2022	477	howdy	28408	2668	0	2	13	32	32	17	4	15	0,13333333	0.3454	1.0	1.0	5	2.08	0.03064	0.14206	0.01393	0.02228	0.13649	0
10.10.2022	478	enjoy	26878	2654	0	3	12	29	33	20	3	15	0,33333333	0.26462	1.0	1.0	5	4.09	0.02786	0.04735	0.00279	0.05014	0.13649	0
11.10.2022	479	valid	28575	2752	0	4	28	38	21	8	1	32	-0,75	0.29805	1.0	1.0	5	3.38	0.01671	0.13649	0.02507	0.06407	0.05571	1
12.10.2022	480	ionic	29151	2947	0	2	13	25	28	21	11	15	0,4	0.31198	3.81794	1.0	5	2.21	0.02786	0.14206	0.06407	0.06407	0.01393	0
13.10.2022	481	equal	27197	2677	0	5	23	35	25	11	2	28	-0,6071429	0.26184	1.0	1.0	5	3.65	0.02786	0.00557	0.07799	0.08357	0.06685	1
14.10.2022	482	floor	28906	2752	0	3	23	44	24	6	0	26	-0,7692308	0.4624	3.63358	1.0	5	3.94	0.06128	0.11699	0.13649	0.05014	0.09749	1
15.10.2022	483	catch	30403	3123	0	7	18	20	15	16	23	25	-0,36	0.37047	3.71517	1.0	5	3.87	0.09192	0.13649	0.03621	0.05292	0.05292	1
16.10.2022	484	spade	30459	2854	1	8	29	36	19	6	1	38	-0,8421053	0.51532	1.0	1.0	5	2.33	0.14206	0.03343	0.11699	0.02228	0.20056	1
17.10.2022	485	stein	31269	2965	1	12	34	32	16	5	1	47	-0,893617	0.37047	1.0	1.0	5	2.65	0.14206	0.039	0.08357	0.06407	0.04178	1
18.10.2022	486	exist	28612	2805	0	5	24	38	23	8	1	29	-0,7241379	0.3454	1.0	1.0	5	3.73	0.02786	0.01393	0.11978	0.05571	0.12813	1
19.10.2022	487	quirk	28322	2794	0	3	23	39	24	9	2	26	-0,6538462	0.30084	1.0	1.0	5	2.13	0.00836	0.04735	0.11978	0.08078	0.04457	1
20.10.2022	488	denim	28741	2769	0	5	29	40	20	5	0	34	-0,8529412	0.25905	1.0	1.0	5	2.52	0.03343	0.07799	0.06407	0.06407	0.0195	1
21.10.2022	489	grove	28637	2794	0	4	18	30	28	17	3	22	-0,2272727	0.50975	1.0	1.0	5	2.94	0.04735	0.09749	0.13649	0.02786	0.20056	1
22.10.2022	490	spiel	29084	2810	0	7	32	36	19	6	1	39	-0,8461538	0.48189	1.0	1.0	5	1.66	0.14206	0.03343	0.11978	0.11978	0.06685	1
23.10.2022	491	mummy	29279	3021	0	1	4	14	27	37	18	5	6,4	0.30084	3.85832	7.54045	5	2.57	0.05571	0.04735	0.0195	0.04178	0.13649	0
24.10.2022	492	fault	28947	2768	0	7	27	35	22	8	1	34	-0,7647059	0.47075	1.0	1.0	5	3.64	0.06128	0.13649	0.07799	0.06685	0.12813	1
25.10.2022	493	foggy	28953	2817	0	2	13	35	32	15	3	15	0	0.40947	3.86184	1.0	5	2.19	0.06128	0.14206	0.02507	0.04457	0.13649	1
26.10.2022	494	flout	30063	2904	0	6	28	37	21	7	1	34	-0,7941176	0.47075	1.0	1.0	5	1.25	0.06128	0.11699	0.13649	0.02786	0.12813	1
27.10.2022	495	carry	27609	2615	0	4	22	35	24	12	3	26	-0,5384615	0.50418	3.72617	1.0	5	3.89	0.09192	0.13649	0.0585	0.08078	0.13649	1
28.10.2022	496	sneak	27905	2636	0	7	28	36	21	7	1	35	-0,8	0.40111	1.0	1.0	5	2.99	0.14206	0.04735	0.08357	0.08357	0.04457	1
29.10.2022	497	libel	25156	2536	0	3	15	32	32	16	2	18	-0,1111111	0.30919	3.79629	1.0	5	2.34	0.03621	0.06407	0.02228	0.11978	0.06685	1
30.10.2022	498	waltz	24672	2496	0	2	11	29	35	19	3	13	0,46153846	0.27019	1.0	1.0	5	2.38	0.03064	0.13649	0.02507	0.07521	0.00279	0
31.10.2022	499	aptly	26498	2572	0	3	26	41	23	7	1	29	-0,7586207	0.35097	1.0	1.0	5	2.12	0.07799	0.03343	0.03621	0.06685	0.13649	1
01.11.2022	500	piney	27502	3667	0	1	14	37	33	14	2	15	-0,0666667	0.44568	1.0	1.0	5	1.48	0.06128	0.06407	0.06407	0.11978	0.13649	1
02.11.2022	501	inept	27670	2640	0	6	30	39	20	6	1	36	-0,8333333	0.30919	1.0	1.0	5	2.13	0.02786	0.04735	0.08357	0.02228	0.12813	1
03.11.2022	502	aloud	29554	2819	1	18	31	30	15	4	1	50	-0,92	0.41504	1.0	1.0	5	2.51	0.07799	0.11699	0.13649	0.02786	0.05571	1
04.11.2022	503	photo	27330	2565	0	5	34	43	15	3	0	39	-0,9230769	0.39554	3.66992	1.0	5	3.97	0.06128	0.09192	0.13649	0.07521	0.03064	1
05.11.2022	504	dream	29743	2751	5	14	31	29	15	4	1	50	-0,92	0.31755	1.0	1.0	5	3.91	0.03343	0.09749	0.08357	0.08357	0.0195	1
06.11.2022	505	stale	31068	3013	2	19	30	27	15	6	2	51	-0,8823529	0.56546	1.0	1.0	5	2.43	0.14206	0.039	0.11699	0.06685	0.20056	1
07.11.2022	506	begin	26096	2439	0	6	26	36	23	7	1	32	-0,78125	0.26462	1.0	1.0	5	3.84	0.05571	0.07799	0.02507	0.06407	0.04178	1
08.11.2022	507	spell	27213	2531	0	4	24	37	24	9	1	28	-0,6785714	0.39276	3.73706	1.0	5	3.37	0.14206	0.03343	0.08357	0.06685	0.06685	1
09.11.2022	508	rainy	28984	2678	1	16	38	31	11	3	1	55	-0,9454545	0.50696	1.0	1.0	5	2.74	0.039	0.13649	0.11978	0.07521	0.13649	1
10.11.2022	509	unite	27467	2575	1	11	31	33	18	5	1	43	-0,8837209	0.46518	1.0	1.0	5	2.9	0.02228	0.04735	0.11978	0.07521	0.20056	1
11.11.2022	510	medal	25993	2438	0	5	25	38	23	8	1	30	-0,7333333	0.31755	1.0	1.0	5	3.39	0.05571	0.07799	0.03343	0.08357	0.06685	1
12.11.2022	511	valet	24660	2356	0	4	22	38	25	9	1	26	-0,6538462	0.42618	1.0	1.0	5	2.17	0.01671	0.13649	0.02507	0.11978	0.12813	1

Figure 10: All data attributes (yellow highlight for "hard words")

7.4 Prediction for word "EERIE"

Pre-calculating for the word "EERIE" of LAN (L), DLet, TLet, FreqSum, and FreqLang(f_w), the data attributes are as follows:

FreqSum = 0.42897

$$F\_W = 2.33$$

$$DLet = 6.37858$$

$$TLet = 6.81018$$

$$L = 5$$

Thus, after using the prediction made by the DT, we can conclude that the word ”EERIE” is HARD (label 0).

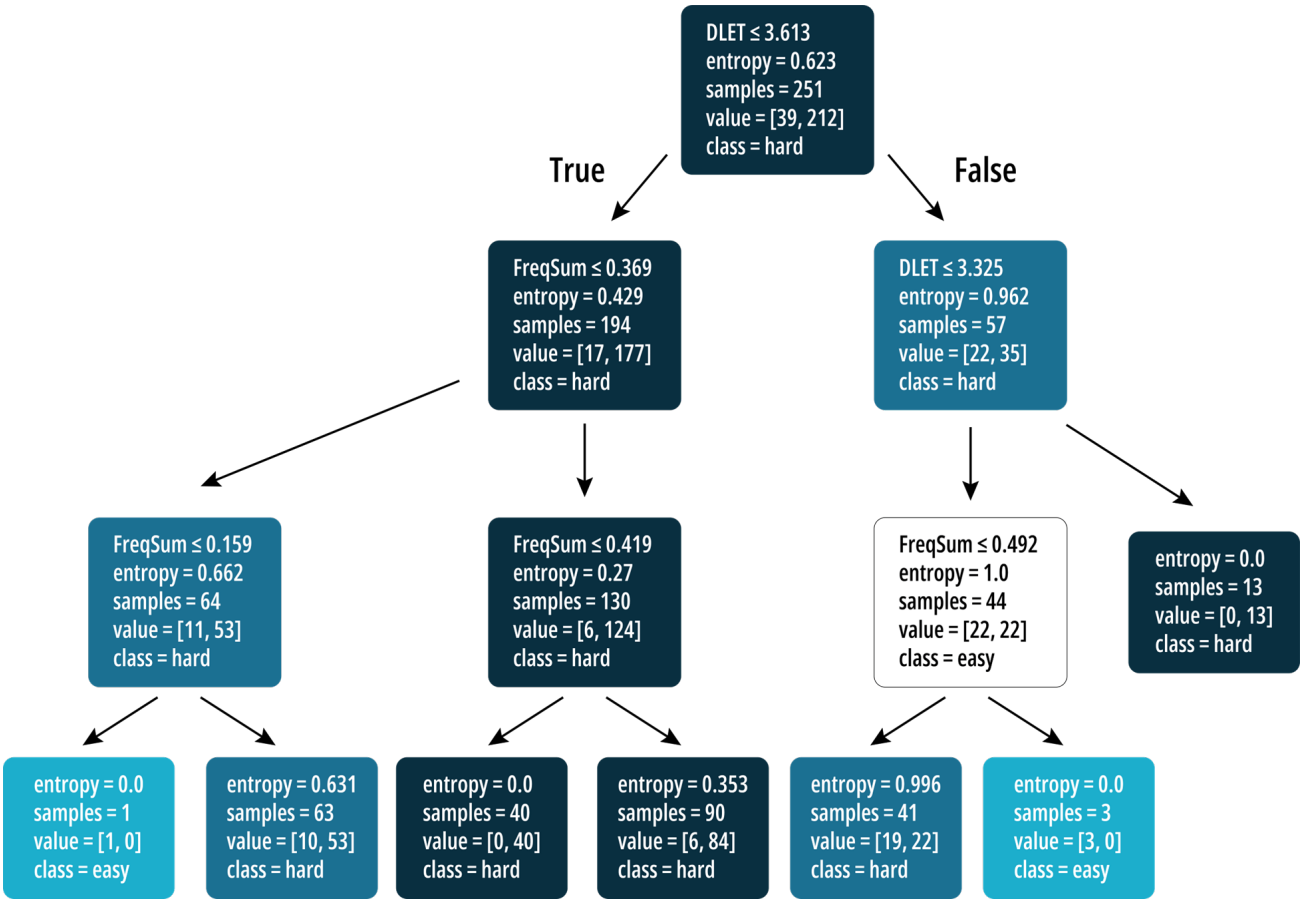

Figure 11: Decision Tree Visualisation

8 References

YouTube. (2022, February 6). Solving wordle using information theory. YouTube. Retrieved February 21, 2023, from https://www.youtube.com/watch?v=v68zYyaEmEA

1.10. decision trees. scikit. (n.d.). Retrieved February 21, 2023, from https://scikit-learn.org/stable/modules/tree.html

Frizler, I. (2022, January 12). The science behind Wordle. Medium. Retrieved February 21, 2023, from https://ido-frizler.medium.com/the-science-behind-wordle-67c8112ed0d1

Myers, L. (2023, January 12). What's the best time to post on Twitter? (2023): Louisem. Louise Myers Visual Social Media. Retrieved February 21, 2023, from https://louisem.com/6624/best-time-to-post-twitter

Navlani, A. (2018, December 28). Python decision tree classification tutorial: Scikit-Learn Decisiontreeclassifier. DataCamp. Retrieved February 21, 2023, from https://www.datacamp.com/tutorial/decision-tree-classification-python

Published by S. Dixon, amp; 11, N. (2022, November 11). Twitter global mdau 2022. Statista. Retrieved February 21, 2023, from https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/

Published by S. Dixon, amp; 22, N. (2022, November 22). Countries with most Twitter users 2022. Statista. Retrieved February 21, 2023, from https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

Published by Statista Research Department, amp; 20, O. (2022, October 20). World population by age and region 2022. Statista. Retrieved February 21, 2023, from https://www.statista.com/statistics/265759/world-population-by-age-and-region/

9 Appendices

```
def findletterdistribuition(words):
    for letter in string.ascii_lowercase:
        outp[letter] = [0]*6

    for letter in string.ascii_lowercase:
        for pos in range(5):
            for word in words:
                try:
                    if word[pos] == letter:
                        outp[letter][pos] += 1
                except: IndexError
    for letter in string.ascii_lowercase:
        for pos in range(5):
            outp[letter][pos] = outp[letter][pos] / len(words)

var = pd.read_excel("data_set_C.xlsx", skiprows = 1)
```

The code above shows the function used for calculating the letter frequency in the english dictionary.

```
def freqsum(word):
    return round(sum(outp[word[pos]][pos] if word[pos] in string.ascii_lowercase else 0 for pos in range(len(word))),5)

with open("freqsum.txt", "w") as f:...

def freqpos(word, pos):...

for i in range(1,6):...

def double_letter_score(word):
    res = 1.0
    for i in range(len(word)):
        for j in range(i + 1, len(word)):
            if word[i] == word[j]:
                res *= (2.0 - outp[word[i]][i]) * (2.0 - outp[word[j]][j])
    return round(res,5)

with open("DLET.txt", "w") as f:...
```

This is a code snippet of the scrip used fir preprocessing, data clean-up and attributes assignment.

```
# Load libraries
import ...
col_names = ["F1","F2","F3","F4","F5","DifLevel", "FreqSum","FreqLang", "DLET", "TLET", "LEN", "LABEL"]
feature_cols = ["FreqSum","FreqLang", "DLET", "TLET", "LEN"]
var = pd.read_excel("data_set_C.xlsx", skiprows=1)
#Data Collection
X_array = []
for i in range(len(var)):...
X_ndarray = numpy.array(X_array, dtype=numpy.float64)
y = list(var.LABEL)

# Split dataset into training and test 70-30
X_train, X_test, y_train, y_test = train_test_split(X_ndarray, y, test_size=0.3, random_state=1)

# Create DT
clf = DecisionTreeClassifier(criterion= "entropy", max_depth= 3)

# Train DT
clf = clf.fit(X_train,y_train)

#Predict for test dataset
y_pred = clf.predict(X_test)
```

This is the script of the machine learning library and functions used to develop the decision tree.

10 Letter

Mr. Will Shortz
The New York Times Company
620 Eighth Avenue
New York, NY 10018

February 20^{th}, 2023

Dear Mr. Shortz,

We hope that this message finds you well! We are writing to you regarding our favorite word puzzle – Wordle. Being part of this study was a very fun experience and an even more instructive opportunity to get behind the science of the game and its players. We hope our work will provide the NYT with useful insights, all the while laying a solid foundation for further research.

Using the data from almost a year-worth of game reports on Twitter, we've been able to come to some very interesting findings. Below, we will outline our results, the strategies we've employed to reach these conclusions, and some closing remarks.

Firstly, it is rather obvious that the game reached the peak of its popularity last year and has been declining ever since. Disregarding day-to-day variations that root from numerous causes such as day of the week, or word difficulty, we have been able to create a model for the macro trend of the user reports. At its core, our model is similar to that of the SIR model for infectious diseases. After numerous tweaks, we were able to predict the interval of tweets with a very high degree of accuracy. With this framework in place, we focused on the number of reports in Hard mode. In contrast to the total number of reports, the ones of Hard mode games follow an increasing trend (logarithmic but increasing nevertheless). This can be explained by the fact that people who stick to playing the game and reporting their results are die-hard fans that have a significantly lower rate of quitting. Therefore, the rate at which people convert to playing Hard-mode from normal mode outweighs the quitters, and hence the continuously growing player base.

Below, we have attached a chart of our model's prediction, including the interval of reported results for March 1st, 2023, namely 18,325 for total reports, and roughly 1821 for Hard mode reports.

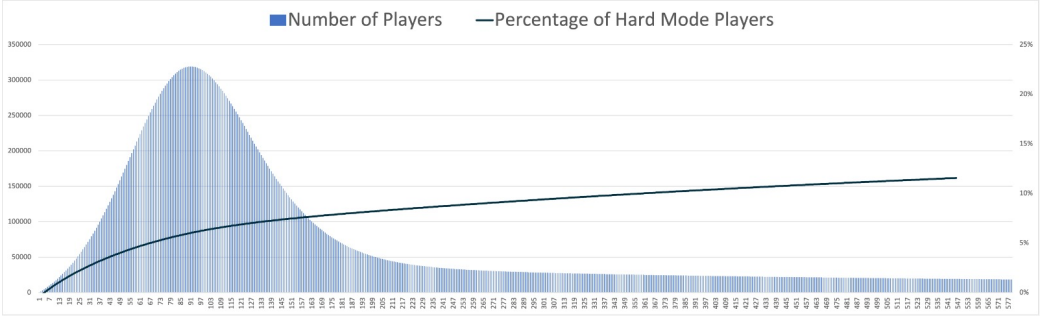

Figure 12: Model Predictions over the entire interval

As for the distribution of reports across the seven compartments (1 try, 2 tries, . . . , 6 tries, failed), we

went for a much simpler model. Taking inspiration from our favorite YouTube channel, 3Blue1Brown, we went for a (very) stylized Information Theory approach, a machine learning approach. Assuming people have developed a strategy that they abide by, we tried to compare how much information can be obtained by that initial guess ("CRANE") and correlate it to the report distribution. We have been able to get a rough estimate, and below we've got the two aforementioned models when analyzing the word "EERIE spa

The difficulty of each word was the last challenge we tackled. Our labeling system is somewhat black and white (we're categorizing words as either challenging/hard words or easy-to-guess words). There were many challenges that arose when attempting to classify solutions, given the subjectivity of the topic, but in the end, we managed to find some word-nonbiased attributes that will stand the test of time. With these attributes assigned, the model was processed using a decision tree, which was chosen for a multitude of reasons some of which are: its transparency and the fact that it can capture nonlinear relationships between the features and the target variable. With a graphical interpretation of the algorithm, some code snippets, and a bit of flair, we can to some degree predict the difficulty of future possible solutions.

Lastly, you might find it interesting that there are some reoccurring events that can tell us something about the users. We've been able to relate daily Twitter usage to the number of reported results, which was not unexpected, but the trend showing more reported results the day after a challenging hard was surprising, nonetheless. Our theory is that more people are intrigued by the game when they are faced with difficult-to-guess words.

We look forward to hearing some of your thoughts on our work, and thank you very much for giving us this opportunity!

Yours sincerely,
Team 2317059