Problem Chosen	2023 MCM/ICM Summary Sheet	Team Control Number
C		2316611

Cracking the Wordle Code: A Machine Learning Approach

In 2022, the game **Wordle** took the world by storm, and became a household name. As the game gained popularity, trends in solving the daily puzzle game have surfaced. Knowing these trends can assist the New York Times, the owner of the game, to improve the experience of users playing the game

The paper aims to investigate the patterns and trends in the Wordle game by analyzing data collected by @WordleStats from Twitter. We used **Facebook Prophet** for **time-series forecasting** to analyze the daily changes in **reported results and forecast the number of hard mode players**. We also used **Multiple Linear Regression** and **Random Forest** to investigate how different word attributes affect the **distribution of the number of guesses** required to guess a word correctly. Lastly, **K-Means Clustering** and **Decision Trees** were used to develop a summarized model that **ranks the difficulty of a word** based on its features. The model was **successful in forecasting future values** based on past trends and this paper provides insight into the Wordle game and its players and demonstrates how the collected data can be analyzed to optimize user experience.

The success of the model in predicting values can be attributed to the various machine learning algorithms used in the study. **Facebook Prophet** was used for **time-series forecasting** that captured seasonal and trend changes in the data, making it an effective choice for predicting the number of players and hard mode players in the game. **Multiple Linear Regression** and **Random Forest**, on the other hand, were used to investigate how different word attributes affected the distribution of the number of guesses required to guess a word correctly. These models can identify which attributes have the most significant impact on the game's difficulty and provide insights into how to improve the game's design.

K-Means Clustering and **Decision Trees** were used to develop a summarized model that ranks the difficulty of a word based on its features. **K-Means Clustering** helped group the words based on their similarities, and **Decision Trees** were used to extract rules that determine the word's difficulty. By combining these models, we created a more accurate and comprehensive assessment of a word's difficulty level.

Overall, the successful prediction of values in this study demonstrates the power of machine learning algorithms and data analysis in optimizing user experience in a game. By identifying trends and patterns in the data, we were able to provide valuable insights that can assist the game's owner in improving the Wordle game's design and enhancing its users' experience.

1 Table of Contents

1. Introduction and Assumptions

1.1 Problem Background

Wordle is a word game that was released in 2021 and gained significant popularity in 2022. The game requires players to guess a five-letter word in six attempts, and it provides feedback through colored tiles. If the letter is correct and in the correct position, the tile turns green, if the letter is in the word but not in the correct position, the tile turns yellow, and if the letter is not in the word, the tile stays gray. Additionally, there is a "Hard Mode" that requires the player to use the same correct guesses in the next attempt. There is only one Wordle challenge every day, and players around the world attempt to guess it.

Many users worldwide report the number of guesses it takes them to solve the daily Wordle problem on Twitter, and the bot @WordleStats collects this data daily. The following information is recorded: the date of the problem, the number of reported scores for the day, the number of reported hard mode players, and the percentage of users who guessed the word in one attempt, two attempts, three attempts, four attempts, five attempts, six attempts, or could not solve the puzzle. This data provides an opportunity to explore patterns and trends in user performance, and understanding these trends can help improve the game's design and optimize user experience.

1.2 Clarifications and Assumptions

In this problem, the summarized Wordle data collected by @WordleStats from Twitter and condensed by the MCM is subjected to potential inaccuracies, given that the data is reliant on self-reported scores from Twitter users. Consequently, false information may be included in the data set due to the origin of the data. Note that the data collection by the MCM may also contain errors. We will account for some of these inaccuracies in our analysis.

1.3 Our Work

For problem 1, our approach involves using Facebook Prophet to conduct time series forecasting to analyze the daily change and variation of reported results. This includes examining past trends in reported user numbers and using them to predict probable future behavior. We also apply Facebook Prophet to forecast the reported number of hard mode players, while exploring potential relationships with word attributes.

Moving on to problem 2, we aim to investigate how different word attributes influence the distribution of percentages for the number of guesses required to correctly guess a word. We start by calculating unique features of the word, such as the sum of the relative occurrence of each letter in the word according to English texts, the relative occurrence of the word in English texts, the number of vowels in the word, and the presence of repeating characters. Using Multiple Linear Regression, we model the difficulty of the word by predicting the mean number of guesses required to guess it correctly. We then use Random Forest to model the distribution of guesses based on the mean number of guesses.

Lastly, in problem 3, we utilize K-Means Clustering to develop a summarized model that ranks the difficulty of a word based on the features calculated in problem 2. This model categorizes the difficulty of the word as either easy, medium, or hard.

2 Dataset

2.1 Given Data File

The provided data file for this problem contains Wordle data from January 7, 2022, through December 31st, 2022. Each entry contains the date, contest number, word, number of reported results from Twitter, number of results recorded in hard mode, percentages of players solving the puzzle in 1, 2, 3, 4, 5, or 6 guesses, and percent of players who did not solve the puzzle in six or fewer guesses.

2.2 External Data

One of the rules in Wordle is that each guess that a player makes must be valid. In our solution, we use a list of valid guesses that is taken from the source code of the game (wordle-list, n.d.). We also use word occurrence data from the Google Books Ngram Corpus using the phrasefinder.io API (Trenkmann, n.d.) (Lin, 2012).

2.3 Data Preparation

2.3.1 Data Cleaning

We were easily able to find words that were either formatted incorrectly or misspelled in the dataset by filtering out the words that did not appear in the list of allowed words. We were able to obtain the actual Wordle answer from a webpage containing a list of past Wordle solutions (List of Wordle Answers, n.d.).We found six words that needed to be corrected:

Contest #207 'favor' was incorrectly formatted as 'favor '
Contest #314 'trash' was incorrectly spelled as 'tash'
Contest #473 'marsh' was incorrectly spelled as 'marxh'
Contest #525 'clean' was incorrectly spelled as 'clen'
Contest #540 'naïve' reformatted as 'naive' since Wordle only accepts English characters
Contest #545 'probe' was incorrectly spelled as 'rprobe'

Additionally, there were a few incorrectly entered dates, which we were able to correct using the WordleStats Twitter page. We corrected three entries:

Contest #239 incorrectly had 2,725 reported results in hard mode, corrected to 9,249 (@WordleBot, #Wordle 239 2022-02-13, 2022)
Contest # 500 incorrectly had 3,667 reported results in hard mode, corrected to 2,667 (@WordleBot, #Wordle 500 2022-11-01, 2022)
Contest # 529 incorrectly had 2,569 total reported results, corrected to 25,569 (@WordleStats, 2022)

Lastly, we found that the sum of the distribution of hard mode guesses in the entry for contest #281 added up to 126. Again, we used the @WordleStats Twitter to correct the data; the values for percentage of players who solved the puzzle in six guesses and percentage of players who did not solve the puzzle were incorrectly entered as 26 and 9, respectively, and were corrected to 9 and 1.

2.3.2 Dataset Features

New features are generated from both the provided dataset and external data (see below for a list of all features included in our working dataset). We started by creating a dataset of allowed Wordle guess words (wordle-list, n.d.). We then queried the phrasefinder.io API to get the number of matches in the Google Books US English corpus for each word in the list of allowed guesses and then calculated their relative frequencies. We also used the list of allowed words to calculate the frequency of each letter.

The provided features in our dataset are:

```
date: date of the contest
contest_num: Wordle puzzle index
word: solution word
num_results: number of reported results
num_hardmode: number of results reported in hard mode
in1: percentage of players solving the Wordle in one guess
in2: percentage of players solving the Wordle in two guesses
in3: percentage of players solving the Wordle in three guesses
in4: percentage of players solving the Wordle in four guesses
in5: percentage of players solving the Wordle in five guesses
in6: percentage of players solving the Wordle in six guesses
over6: percentage of players who did not solve the Wordle in six or fewer guesses
```

The generated features in our dataset are:

```
letter1: first letter of the solution word
letter2: second letter of the solution word
letter3: third letter of the solution word
letter4: fourth letter of the solution word
letter5: fifth letter of the solution word
letter1_int: integer encoded first letter of the solution word
letter2_int: integer encoded second letter of the solution word
letter3_int: integer encoded third letter of the solution word
letter4_int: integer encoded fourth letter of the solution word
letter5_int: integer encoded fifth letter of the solution word
avg_num_guesses: average number of guesses to solve the Wordle in hard mode
day_of_week: day in week
word_score: sum of letter frequencies
word_occurrence: occurrence score of word with respect to other allowed guess words
vowels: number of vowels in a word
repeats: number of repeating letters in a word
```

Each word in the problem dataset is split up by letter. However, most machine learning algorithms only can process numerical input, which is why each letter is encoded with an integer value. The most frequent letter being assigned 0 and the least common letter being assigned 25.

3 Exploratory Data Analysis

Before modelling, it is important to do exploratory data analysis so to get a better understanding of the data that we are working with.

First, we have a correlation plot. It is helpful to see which features are correlated so that we can make informed decisions about what to use in our models.

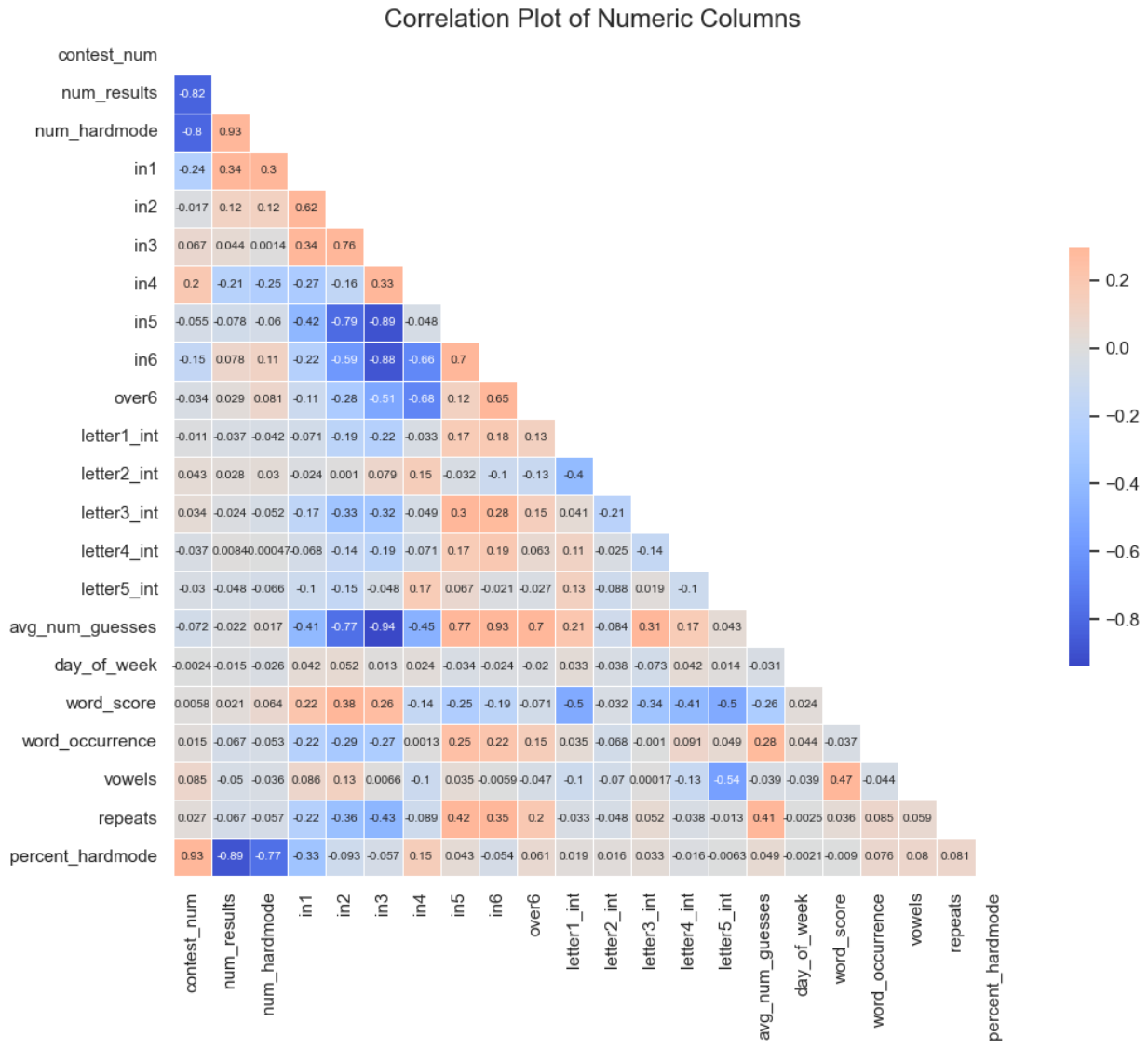

Figure 1 Correlation Plot

Figure 2 shows boxplots for the number of results by the day of week. From these plots, we can see that the day of the week has almost no effect on the number of results.

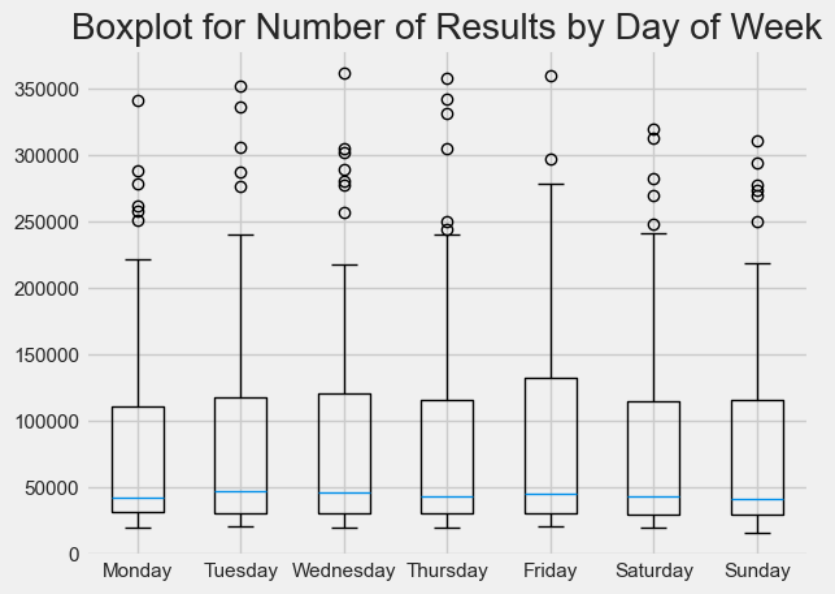

Figure 2 Day of Week Boxplot

Figure 3 shows boxplots of the distribution of the number of guesses it takes players to solve the Wordle. This gives us a good idea of the shape of the distribution of how many guesses it takes for players to solve the Wordle.

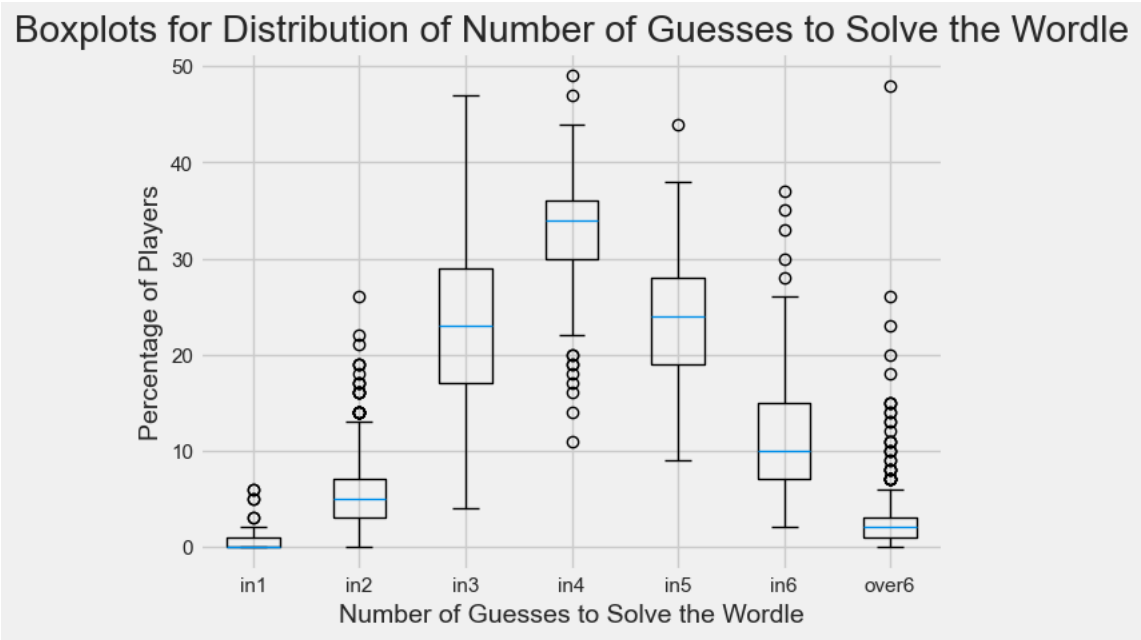

Figure 3 Guess Distribution Boxplot

Figure 4 is a more detailed visualization of what the distribution of number of guesses to solve the Wordle is as it shows the data for each day.

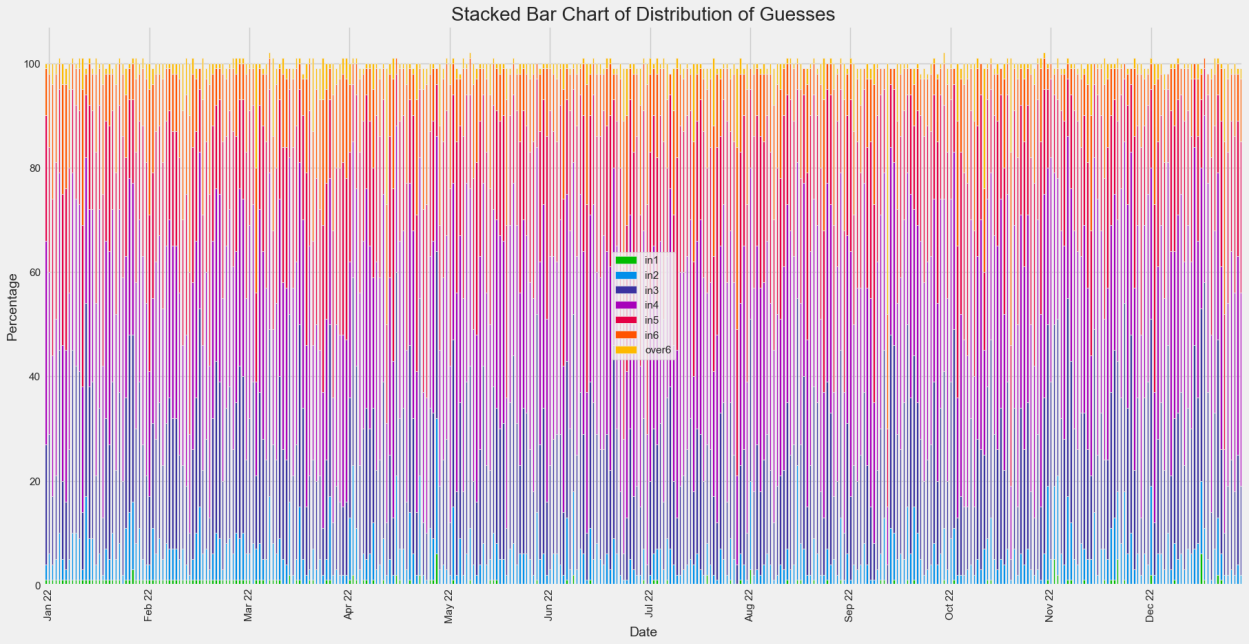

Figure 4 Stacked Bar Chart of Guesses

Figure 5 shows the average number of guesses it takes players to solve the Wordle over time. We can see that this does not have a strong relationship with time, this indicates that the average number of guesses depends on features in the word itself.

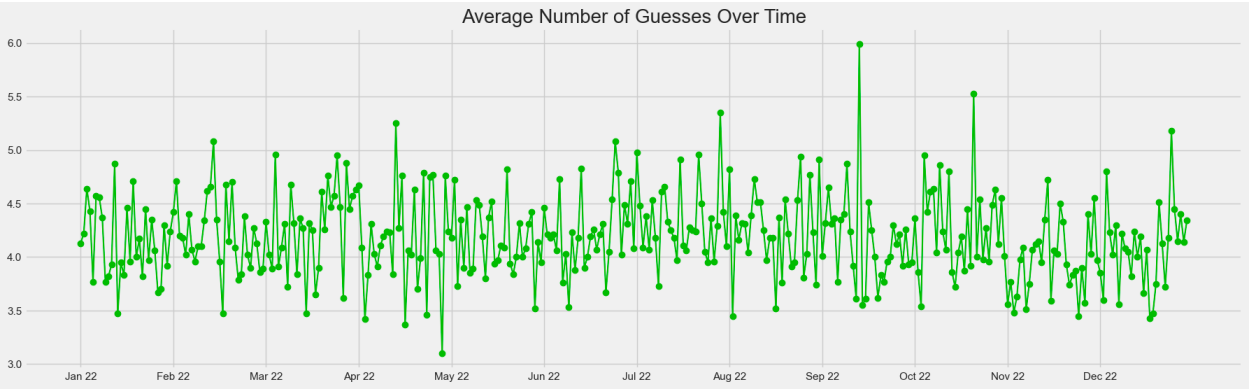

Figure 5 Average Number of Guesses Over Time

4 Time Series Forecasting to Determine Variations in Daily Reported Users and Hard Mode Players

4.1 Method Overview

In this problem, we use Facebook Prophet as our modelling tool to predict future values of the number of reported players. Facebook Prophet's strength comes from its decomposable time series model which is composed of three main components - trend, seasonality, and holidays (Letham B, 2017). This is represented in the following equation:

$$y(t) = g(t) + s(t) + h(t) + e_t$$ (Taylor and Letham 2017)

The trend function *g* captures non-periodic changes in the time series, while the seasonality function *s* accounts for periodic changes in the data such as weekly and monthly patterns, and the holidays function *h* considers the possible impact of irregular schedules on the model (Letham B, 2017). The error term e_t represents any changes that are not accounted for by the model (Letham B, 2017).

The Facebook Prophet model is particularly well-suited to predict future values of the number of reported players, as it can factor in additional features not present in the data set, such as seasonality and holidays. Furthermore, this modeling approach is also applicable for forecasting the ratio of reported hard mode players relative to the total number of reported players.

4.2 Model Construction

We began the construction of the model by visually inspecting the generated graphs to determine if there were any discernible trends in the number of reported players. In Figure 6 , we observe a clear pattern of a rapid increase, followed by a peak, then a sharp decline, and finally, a gradual tapering off.

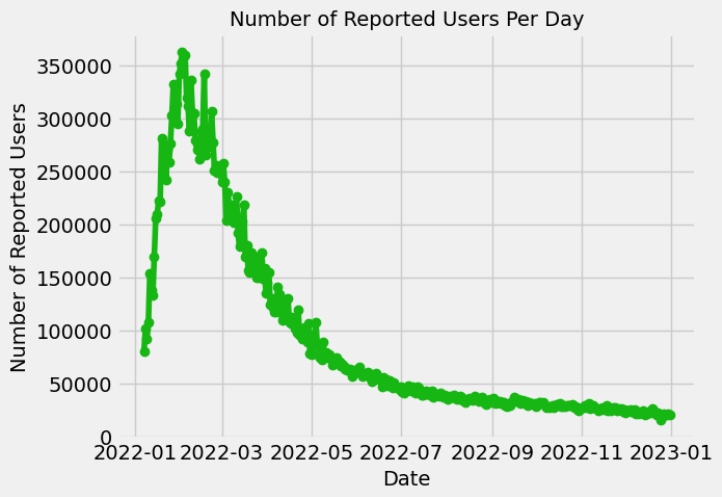

Figure 6 Number of Reported Users Per Day

Knowing that there is a possible predictable trend, we used Facebook Prophet to model the data. To prepare the data for modelling, we partitioned it into a train set, representing the first 80% of the days, and a test set, representing the remaining 20% of the dataset. We then used hyperparameter tuning to optimize the model. Namely, tuning the seasonality mode and the changepoint prior scale

parameters. After testing different hyperparameters, the model that used a seasonality mode of multiplicative and a changepoint prior scale of 10 with the best Mean Absolute Percent Error (MAPE = 0.0512) was chosen.

We then repeated the process by visually inspecting the generated graphs that show the reported number of hard mode players over time. Figure 7 reveals that the graph's shape is similar to that of Figure 6 . Thus, we plotted the ratio of the reported number of hard mode players to the total number of reported players, as shown below in Figure 8 .

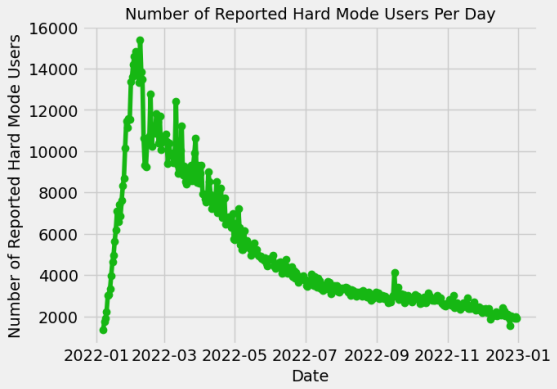

Figure 7 Number of Hard Mode Users per Day

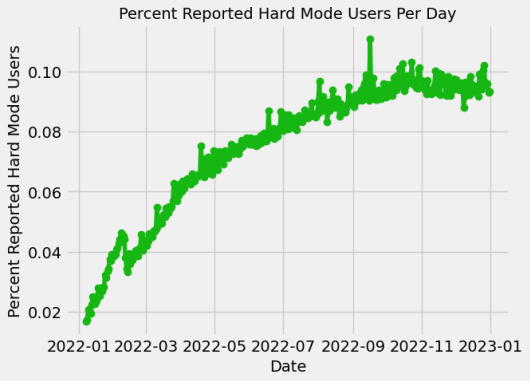

Figure 8 Percentage of Hard Mode Users per Day

As with the number of reported players, we split the data into a training and testing set and used Facebook Prophet to model the reported number of hard mode players. After conducting hyperparameter tuning, we selected the model that had the lowest Mean Absolute Percent Error (MAPE = 0.0719). The selected model used a seasonality mode of additive and a changepoint prior scale of 0.10.

4.3 Forecasting and Results

Based on the models created, we can visualize the trends of the number of reported users, and the percentage of reported hard mode players relative to the number of reported users. Based on these models, we can then make a prediction on the number of reported users and the percentage of those users that are playing in hard mode.

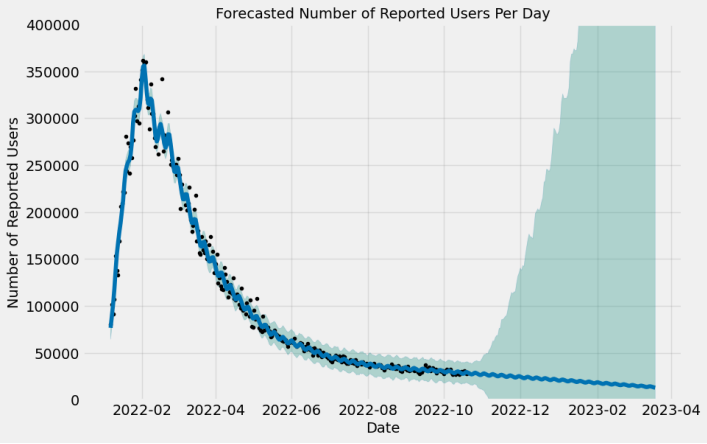

Figure 9 Forecasted Number of Reported Users per Day

Figure 9 indicates a downward trend in the number of players. However, there is a high degree of uncertainty in the forecasted values, shown by the maximum and minimum predicted values shown in the graph. For March 1, 2023, the model predicts the number of reported players to be 15,955, but the actual value could range anywhere between 0 and 689,551.

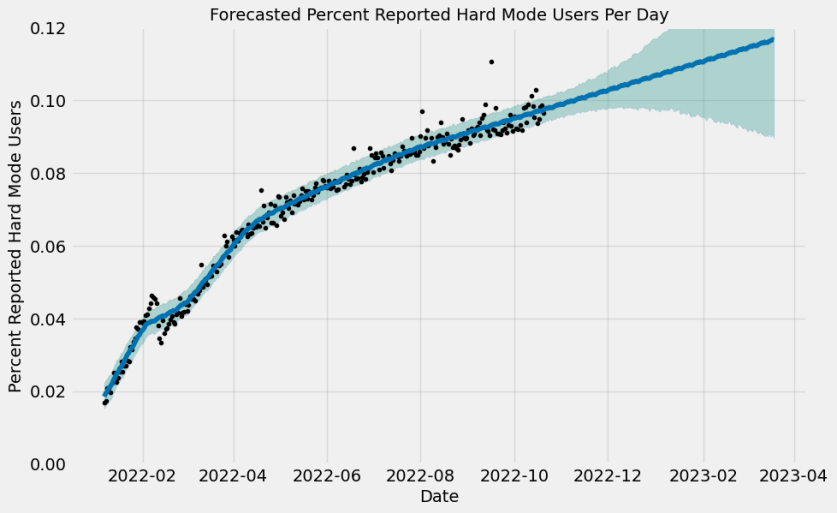

Figure 10 Forecasted Percentage of Reported Hard Mode Users per Day

Figure 10 indicates an upward trend in the ratio of players playing in hard mode. And there is not much uncertainty in the forecasted values, shown by the maximum and minimum predicted values shown in the graph. For March 1, 2023, the model predicts the percentage of reported hard mode players relative to the total number of reported players to be 11.40%, with the actual value ranging from 9.21% to 13.71%.

In the Exploratory Data Analysis, it was determined that there is a weak correlation between the attributes of the word and the percentage of reported players that play in hard mode. Due to the way the game is played, the decision for the player to play in hard mode is decided prior to beginning the game, and subsequently prior to knowing the word. Thus, it is very improbable for attributes of the word to influence the number of people that play the game in hard mode. Therefore, modelling the number of players in hard mode as a relative ratio to the total number of reported players that varies over time is the most precise approach.

5 Predicting Distribution of The Number of Guesses

5.1 Method Overview

In this problem, we will use Multiple Linear Regression and Random Forest to determine the distribution of percentage of the number of guesses based on the word of the day. Multiple Linear Regression was used as it takes advantage of multiple features to predict another given feature. We use this to predict the average number of guesses for a given word using features that will be discussed later.

Random Forest is a supervised learning algorithm that can do regression and classification by using multiple decision trees. Using this model, we can find trends in relationships between the average number of guesses and the distribution of percentage for each given number of guesses.

5.2 Model Construction

The first part of this problem was to identify a way to measure the word's perceived difficulty. The perceived difficulty of the word was measured by the mean number of guesses for the given word. This was chosen as the measure, as a word that is more difficult would theoretically take users more tries to guess, and an easier word would take less guesses. The average number of guesses was calculated as `avg_num_guesses` in the calculated features in Part 3.3.2.

The next part of this problem was identifying features that could affect the average number of guesses. We also created the following calculated features: letter1_int, letter2_int, letter3_int, letter4_int, letter.5_int, word_score, word_occurrence, vowels, repeats. These features are described in Part 3.3.2.

Using these variables, we created a Multiple Linear Regression model that was trained with 85% of the data that used the variables above to predict the average number of guesses. In the graph below, we can see the model versus the actual values. The model had a mean squared error of 0.078 and a mean absolute error of 0.217.

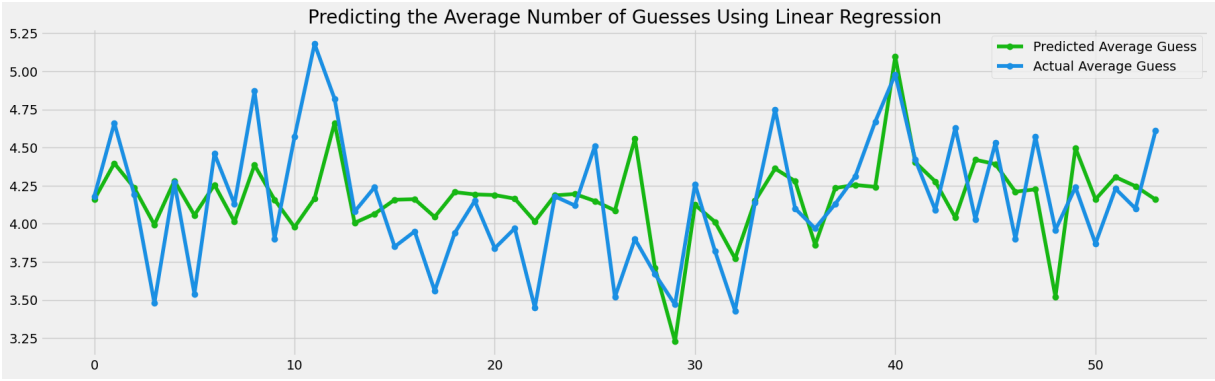

Figure 6 Multiple Linear Regression Predictions for Average Number of Guesses

The final part of the problem was to identify the distribution of the number of guesses made by players based on the average number of guesses. To achieve this, a Random Forest model was employed, using the average number of guesses as an indicator of the distribution trend. The model was trained with 80% of the available data, and the remaining 20% was used for testing. The model's performance was evaluated by calculating the mean absolute error, and the resulting errors are

presented in the table below. Based on the mean absolute error values, we can conclude that the model is fairly accurate at predicting the distribution with a maximum percentage error of 2.75%.

Variable	MAE
in1	0.548130790374884
in2	1.8457377570418037
in3	1.910193362159053
in4	2.7547981883100974
in5	2.0451765139574967
in6	1.2949160857515873
over6	1.135007083561933

By looking at the coefficients, we found that the most important features in determining the distribution is `word_occurrence`, then the `word_score`, `repeats`, and `vowels`.

Additionally the coefficients showed that letter 3 has the highest significance on the average number of guesses, letter 4 was the next most significant. Letters 1, 2, and 5 have roughly the same significance.

5.3 Forecasting and Results

We can see from the mean absolute errors generated by the model that the error for the average is 0.217, which means we can calculate the average difficulty of the word fairly accurately. Once we have the average, we can calculate the distribution accurately with a maximum mean absolute value of 2.75.

Using this model, we predict the average number of guesses for the word "EERIE" to be 4.247 guesses. Using this forecasted average, we predict that the distribution of guesses in the table below. Based on the accuracy of the model measured by the mean absolute errors listed above, we can say with confidence that the prediction is accurate with minimal margin of error.

Variable	Percentage of Guesses
in1	0
in2	4
in3	19
in4	35
in5	27
in6	12
over6	2

6 Classifying Words by Difficulty

6.1 Method Overview

Words are classified by difficulty using the K-means algorithm from the scikit-learn package in Python. K-means is an unsupervised machine learning algorithm that clusters data into K groups based on similarity.

The clustering model simply uses `avg_num_guesses` to cluster the data. For this problem, we want to create three clusters since we are classifying words into easy, medium, and hard.

We then train a decision tree with the difficulty classification ratings so that we can classify new words into difficulty levels. The decision tree uses `word_score`, `word_occurrence`, `vowels`, `repeats`, and `avg_num_guesses` to classify words.

6.2 Model Construction

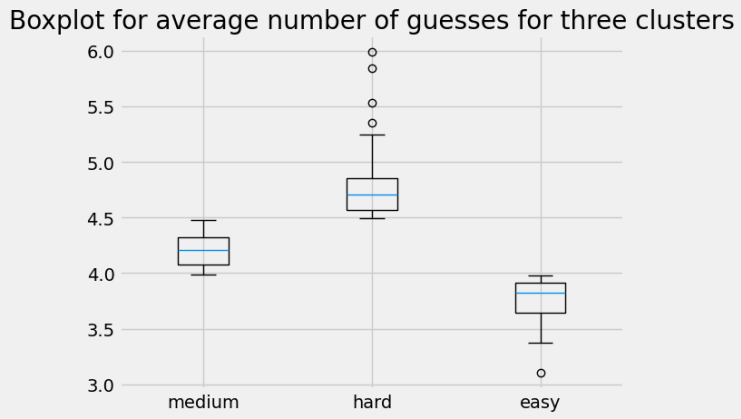

Figure 12 7 Boxplot clusters

In Figure 12 7 we can see that the clusters segment the average number of guesses.

According to this model:

- easy words range from 3.1 to 3.98 average number of guesses
- medium words range from 3.99 to 4.48 average number of guesses
- hard words range from 4.49 to 5.99 average number of guesses

To classify a new word, we must first generate the new features for the word and predict the `avg_num_guesses` as described in section 6.

6.3 Predictions for EERIE

Thus, according to this model EERIE is classified as hard using the predicted `avg_num_guesses` from both the RandomForestRegressor models and the LinearRegression model.

7 Letter to the New York Times

Dear Puzzle Maker at The New York Times,

I am writing to share with you the results of a comprehensive study we conducted on the difficulty of words used in puzzles and the number of guesses that users take to solve them. Our study aimed to identify the factors that contribute to the level of difficulty of a word, and to make recommendations on how to improve the puzzle experience for your users.

Our analysis focused on a range of linguistic features and their impact on the difficulty of guessing a word correctly. We analyzed a large corpus of English texts to identify the occurrence rate of each word in the language and found that words that occur less frequently in the English language tend to be more difficult to guess, resulting in a higher average number of guesses per word.

Additionally, we studied the occurrence rate of each letter in the English language and found that words that use less common letters, such as "z" and "q", are also more challenging to guess. This finding suggests that selecting words with less common letters may increase the difficulty of your puzzles and provide a greater challenge to your users.

We also considered the impact of other linguistic features, including the number of repeating letters and the number of vowels in a word. While these factors also influenced the difficulty of a word, their impact was less pronounced than that of the frequency of occurrence and the use of less common letters.

Based on our findings, we recommend that you consider selecting words that are less common in the English language and that use less common letters in your puzzles, in order to challenge your users more and improve their overall puzzle experience.

Thank you for considering our recommendations, and we hope that our study will provide useful insights to enhance the quality of your puzzles.

8 References

@WordleBot. (2022, February 14). *#Wordle 239 2022-02-13*. (Twitter) Retrieved February 2023, from https://twitter.com/WordleStats/status/1493268878998179840?cxt=HHwWgMCo7ceIlLkpAAAA

@WordleBot. (2022, November 2). *#Wordle 500 2022-11-01*. Retrieved February 2023, from Twitter: https://twitter.com/WordleStats/status/1587852240412254208?cxt=HHwWgICImeXMl4ksAAAA

@WordleStats. (2022, December 1). *#Wordle 529 2022-11-30*. (Twitter) Retrieved February 2023, from https://twitter.com/WordleStats/status/1598361495703670784?cxt=HHwWgMCi4frTwq4sAAAA

Letham B, T. S. (2017). *Forecasting at scale. PeerJ Preprints 5:e3190v2*. Retrieved February 2023, from https://doi.org/10.7287/peerj.preprints.3190v2

Lin, Y. e. (2012). Syntactic Annotations for the Google Books Ngram Corpus. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 169-174.

List of Wordle Answers. (n.d.). Retrieved February 2023, from WordFinder: https://wordfinder.yourdictionary.com/wordle/answers/.

Trenkmann, M. (n.d.). *API*. Retrieved February 2023, from PhraseFinder: https://phrasefinder.io/api

wordle-list. (n.d.). Retrieved February 2023, from GitHub: https://github.com/tabatkins/wordle-list