Wordle Difficulty Analysis

• • •

Vince Gregoric
7/3/23

Background

- Wordle is a word puzzle game where the player has six tries to guess the five-letter answer word
- Each letter in the guessed word will be assigned one of three colors:
  - Gray: not in answer
  - Yellow: in answer, incorrect position
  - Green: in answer, correct position

- Wordle was originally created by Josh Wardle, and is now owned by the New York Times: https://www.nytimes.com/games/wordle/index.html

Project Overview

- The goal of this project was to use machine learning to predict the difficulty of solving a Wordle puzzle given the answer word
- We will use the average player score (i.e., the average number of required guesses) as a measure of difficulty
  - Higher average score → greater difficulty
  - If a player failed to guess the word after 6 guesses, they will be assigned a score of 7

Data

- Player score data was obtained from https://twitter.com/WordleStats, which is a Twitter account that aggregates scores from player tweets
- For this project, we will use data from Wordle 202-696 (except for Wordle 591 and 608, which are missing from the Twitter account)
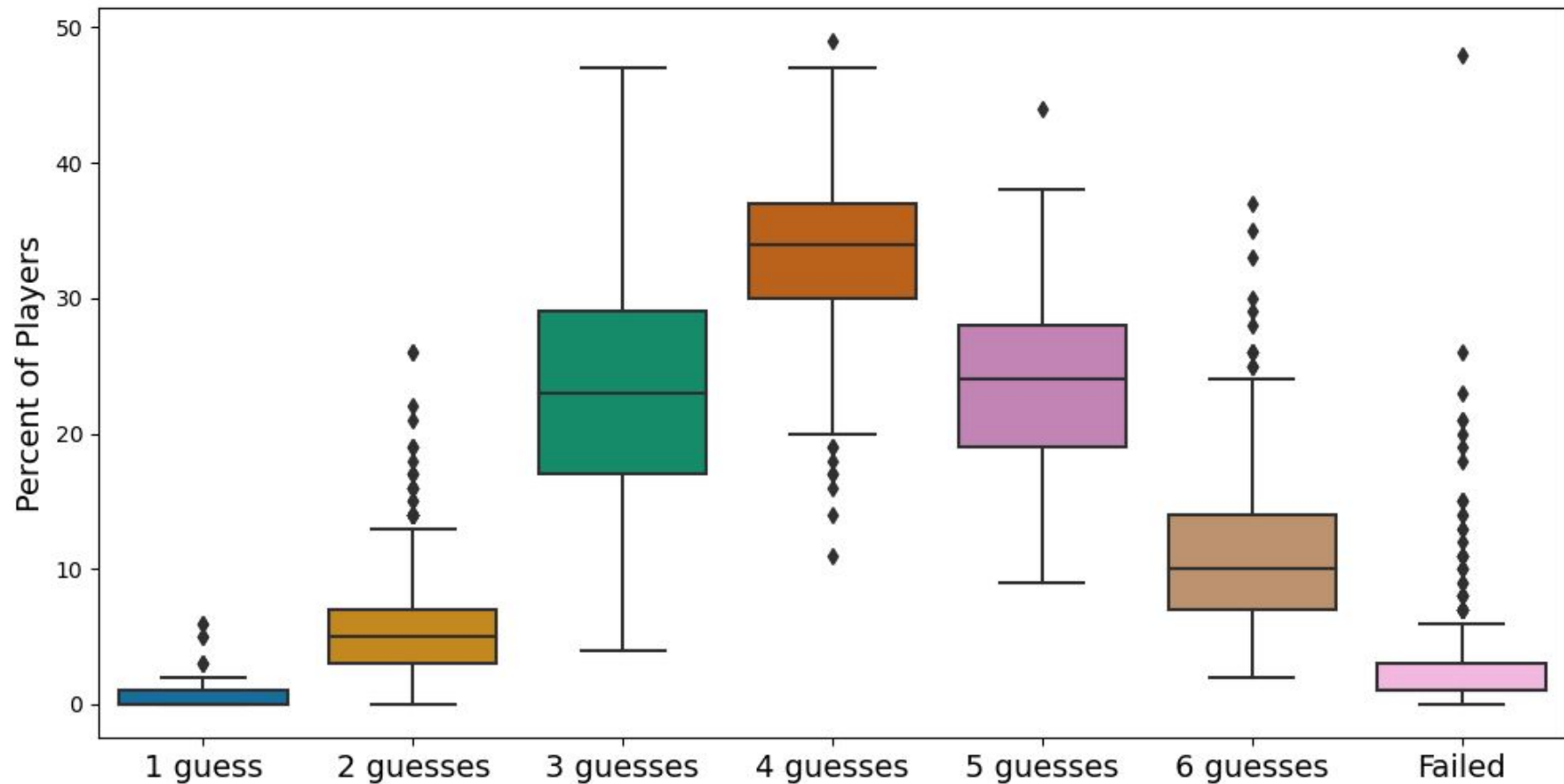
Example tweet:


```
#Wordle 696 2023-05-16
17,831 results found on Twitter.
1,771 hard mode players.

1:    0%
2:   7%
3:          25%
4:            34%
5:         23%
6:    9%
X:    1%

#Wordle696
```
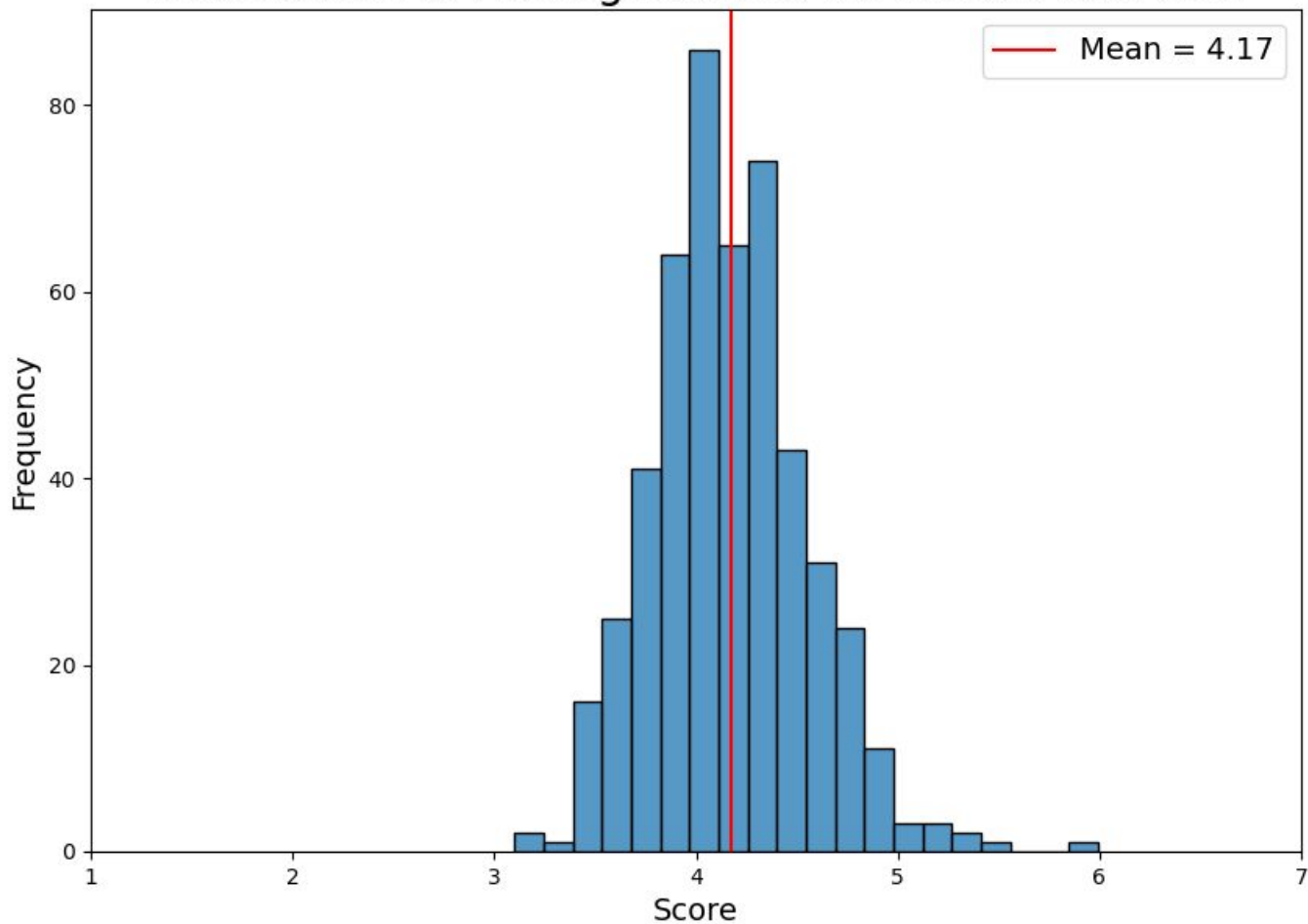
Distribution of Guesses for Wordle 202-696

Distribution of Average Scores for Wordle 202-696

Other Data

- The answer words for each Wordle puzzle were obtained from
  https://wordfinder.yourdictionary.com/wordle/answers/
- Data on word frequency in the English language was obtained from
  https://github.com/IlyaSemenov/wikipedia-word-frequency/tree/master,
  which is a project maintained by Ilya Semenov to count word frequencies
  on Wikipedia articles
- Data on letter frequency in the dictionary was obtained from
  https://en.wikipedia.org/wiki/Letter_frequency

Comparison of Letter Frequency in Data and Dictionary
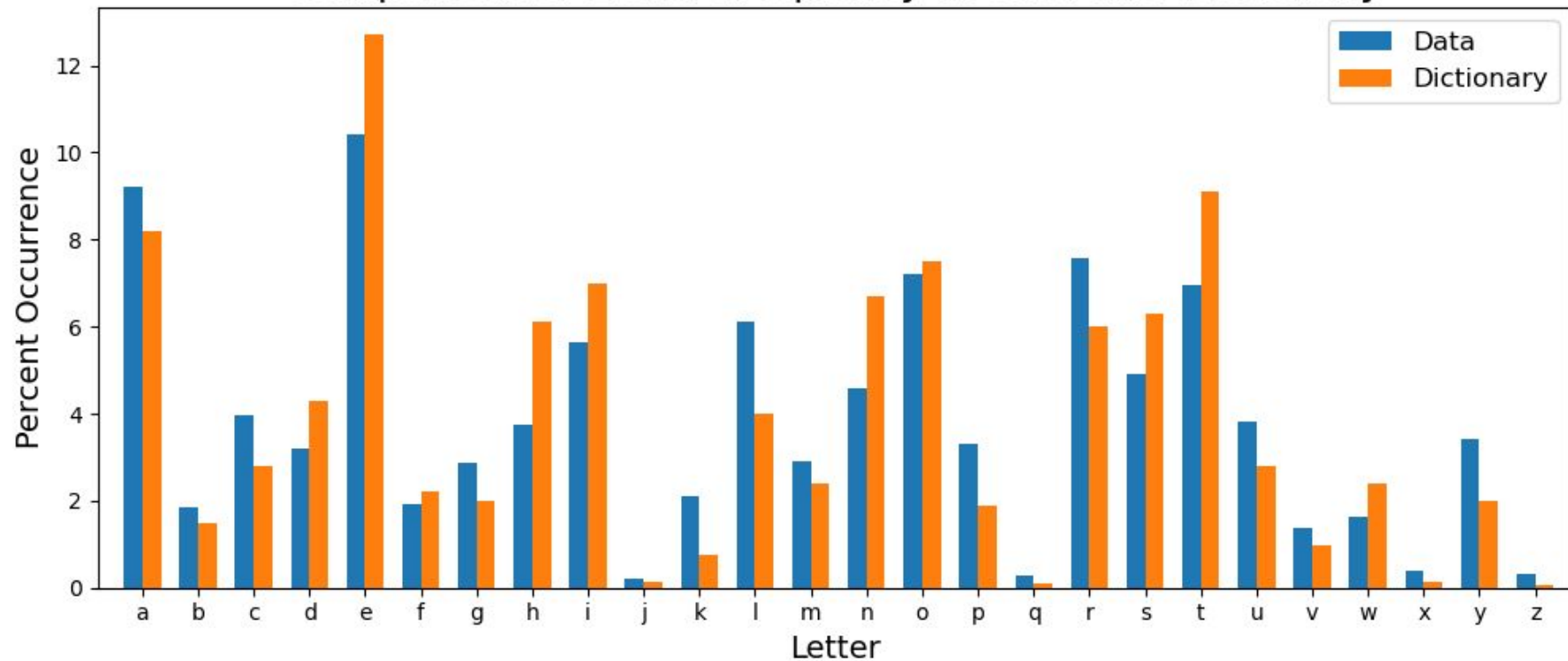
Building a Model

- Target variable: average score (possible values between 1 and 7)
- Features:
  - English language frequency ranking for the answer word
  - For each letter of the alphabet, a boolean for whether the answer word contains the letter
  - A boolean for whether the answer word contains any letter multiple times
  - A boolean for whether the answer word starts with a vowel
- Features were standardized (transformed so that the mean is 0 and the standard deviation is 1) using StandardScaler prior to model fitting

Initial Model: Linear Regression

- The data was fit to a linear regression model using ordinary least squares
  - For the training data, R^2 was 0.506
  - For the test data, R^2 was 0.373
- Since the R^2 value is significantly lower for the test data, this model is overfitting the data
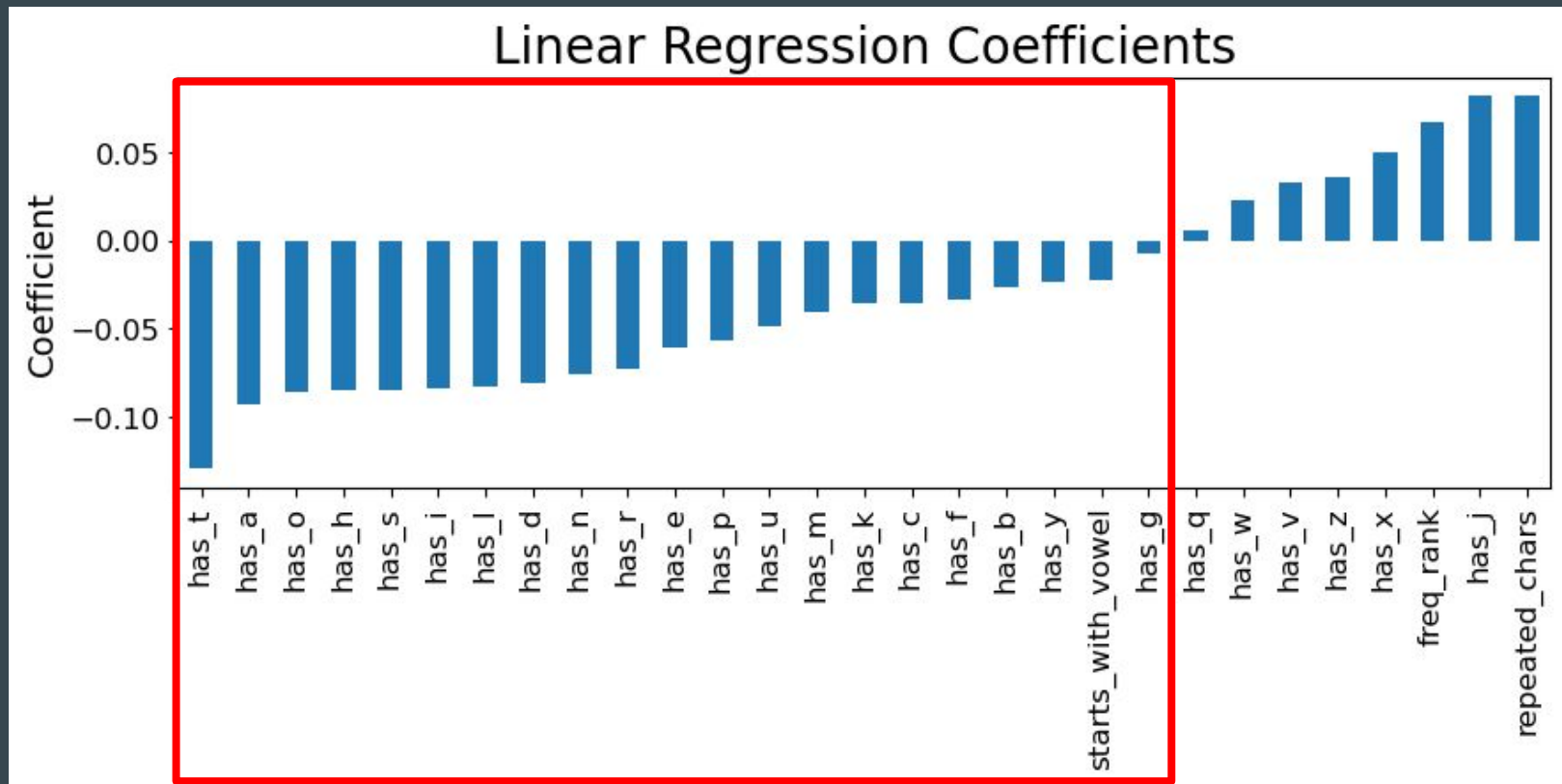- We can still gain insight by looking at the regression coefficients

Linear Regression Coefficients
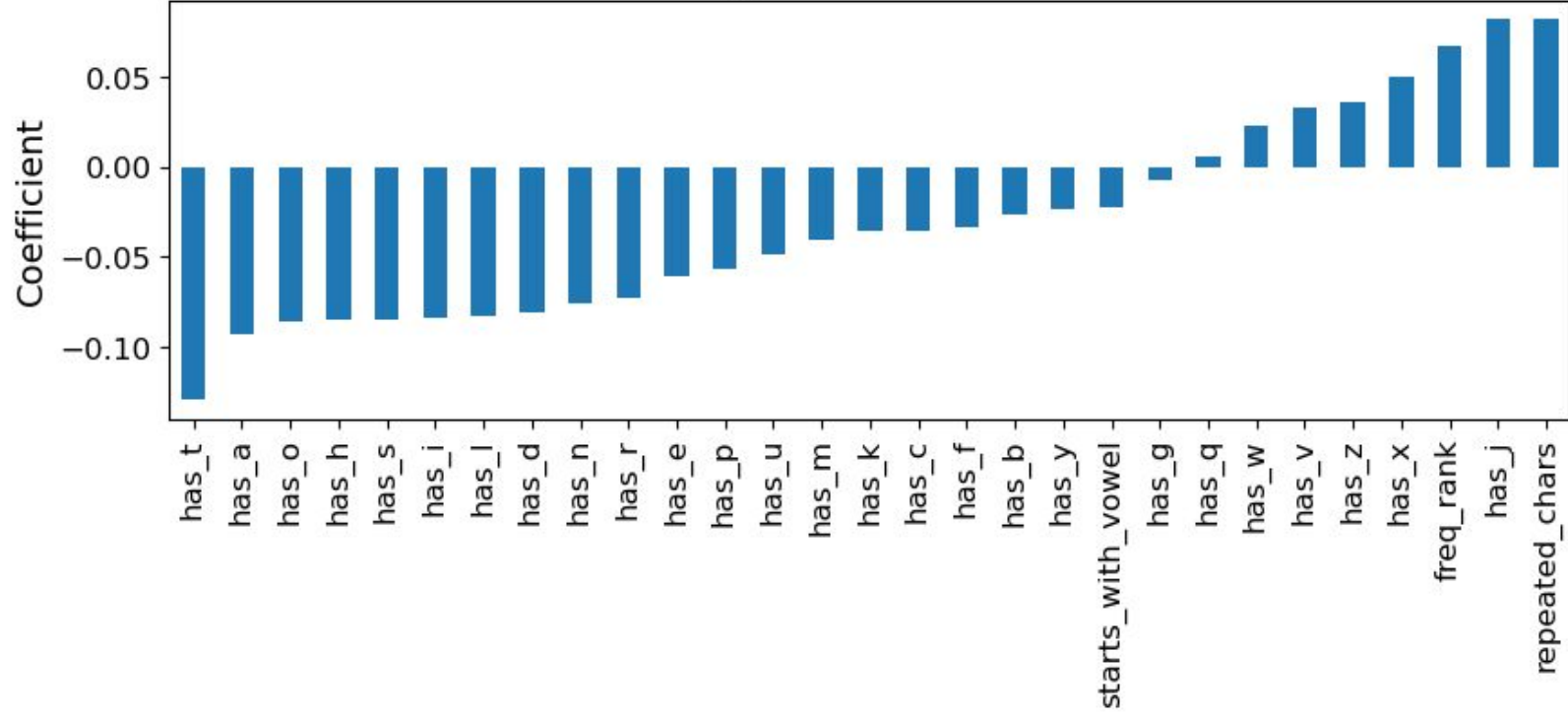
Linear Regression Coefficients
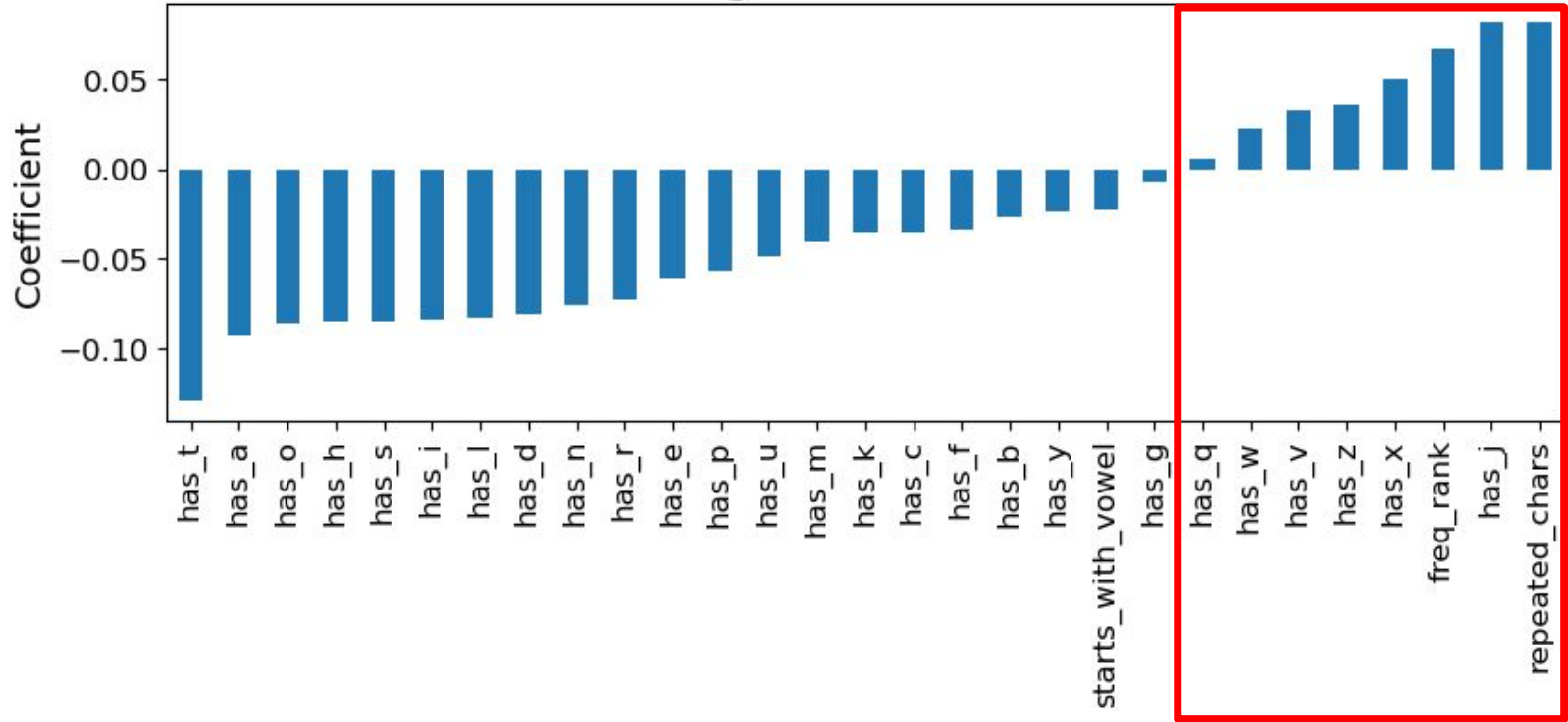
- If a feature has a *positive* coefficient, the puzzle is more difficult when the feature is true and less difficult when the feature is false

Linear Regression Coefficients

- If a feature has a *negative* coefficient, the puzzle is less difficult when the feature is true and more difficult when the feature is false

Linear Regression Coefficients
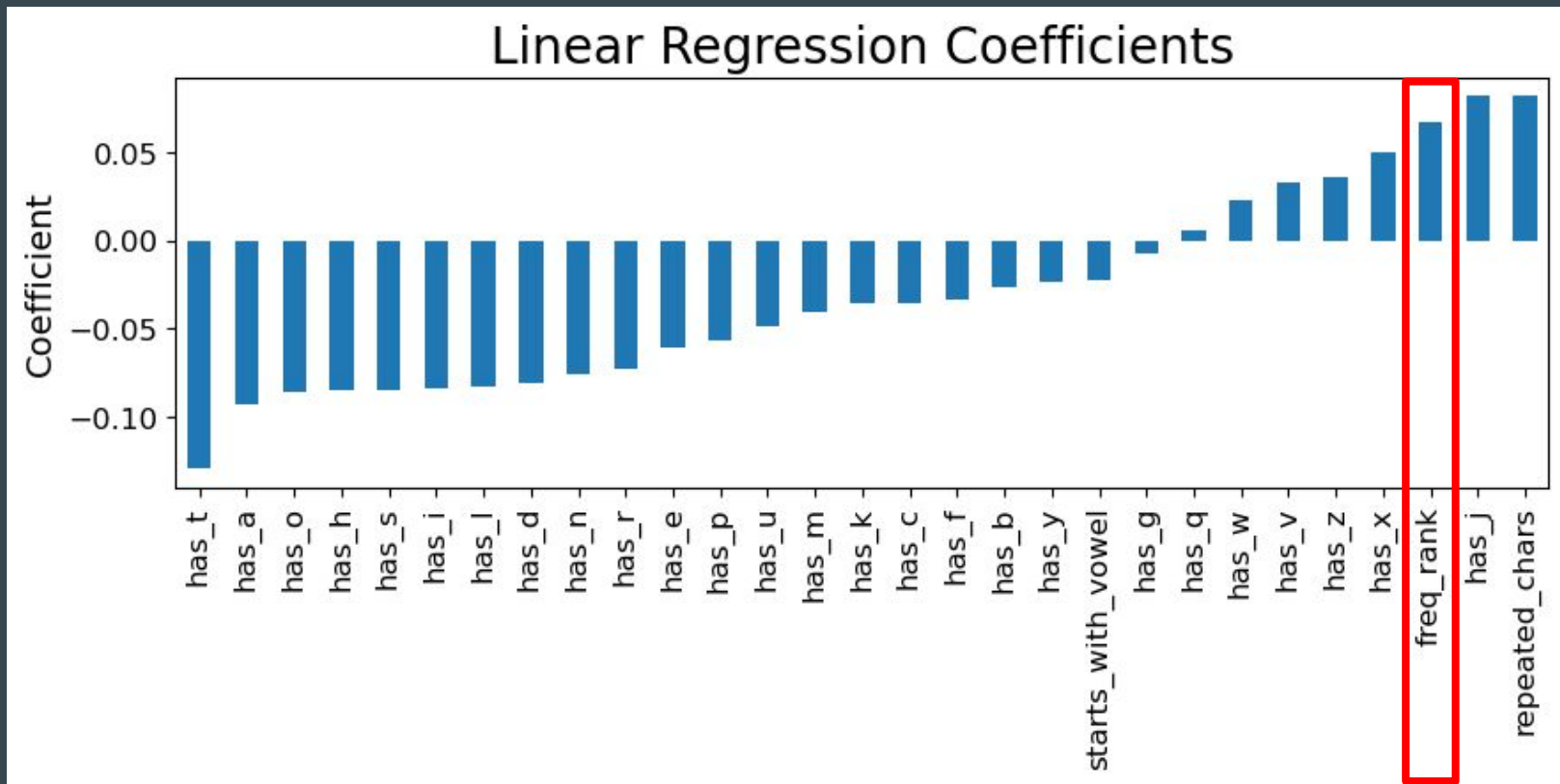
- For the frequency ranking, a higher ranking (less common word) results in a more difficult puzzle

More common letters tend to have more negative coefficients (easier to guess):

Regression Coefficient vs Letter Frequency in Dataset
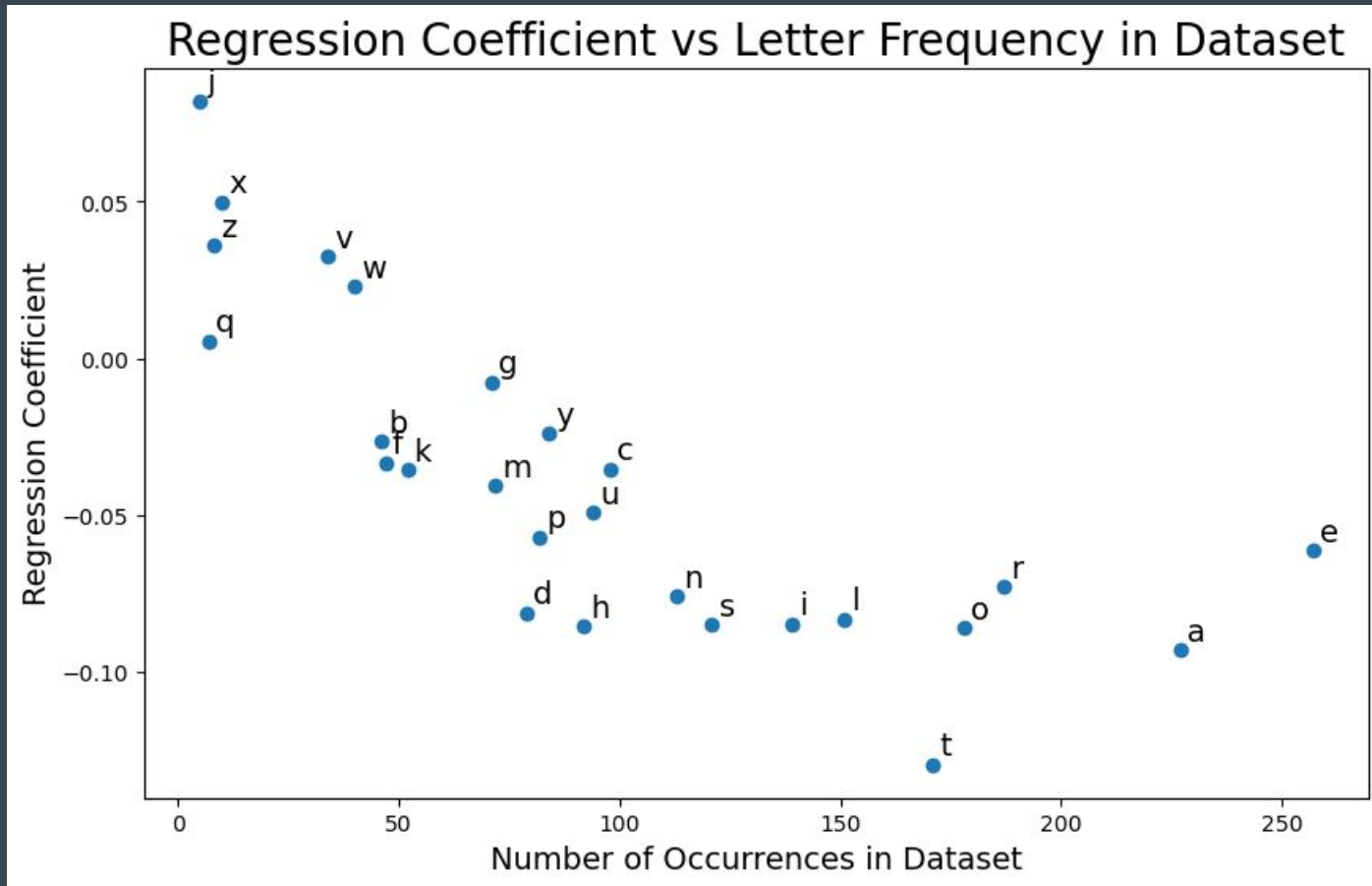
Improving the Model

- To correct for overfitting, the following improvements were tested:
  - L1 (Lasso) regularization
  - L2 (Ridge) regularization
  - Bagging ensemble model
- In each case, the model was cross-validated using GridSearchCV to find the optimal hyperparameter values
- Both L1 and L2 regularization did not significantly improve the model
- The bagging ensemble model was slightly less overfit compared to the original linear regression model (training $R^2 = 0.491$, test $R^2 = 0.380$)

Conclusions

- We have developed a machine learning model to predict the difficulty of a Wordle puzzle given the answer word
- The original linear regression model was overfit
- Regularization did not significantly improve the overfitting
- Using a bagging ensemble model resulted in a slight improvement to the overfitting

Future Work

- Try using different features, such as:
  - The part of speech of the answer word
  - Bigrams (does having certain *pairs* of letters affect the difficulty?)
  - The *number of occurrences* of a given letter in the answer word rather than a boolean for whether the letter is present
- Use automated feature selection methods to find a more predictive combination of features
- Use a categorical target variable (e.g., is the average score greater than 4?)
- Try other machine learning models (e.g., decision tree, k-nearest neighbors)