Introduction

Music is an artform that heavily influences culture. Since there is such a high demand for music and money on the line, artists would like to know if there is anything they can do to have a competitive advantage. For this purpose, an artist would specifically like to know how to have their song rank as high as it can on the Billboard charts. To address this, we asked the following question: How well can we predict the top rank of a song on the Billboard charts, using its lyrics?

Exploratory Data Analysis

1. Explanation of all variables in the dataset
    a. Song Title
        i. This is essentially the ID for each entry. It is the title of the song as a string
    b. Rank
        i. This is the highest rank on the Billboard chart the song achieved
    c. Lyrics
        i. This is a string of the lyrics of the song.
    d. Year
        i. This is the year in which the rank was achieved
    e. songLength
        i. This is the length of the song measured by characters in the lyrics
    f. numExclamation
        i. This is the number of exclamation points in the lyrics
    g. numBigWords
        i. The number of words in the lyrics that are 6 characters or longer
    h. numSmallWords
        i. The number of words in the lyrics that are 4 characters or shorter
    i. numWords
        i. The number of words in the lyrics
    j. percentSmall
        i. The percentage of words that are small in the lyrics (as defined above)
    k. percentBig
        i. The percentage of words that are big in the lyrics (as defined above)
2. Summary Statistics for all variables
    a. Song Title
        i. Summary statistics dont make sense for this variable
    b. Rank
        i. All ranks are an integer between 1-100
    c. Lyrics
        i. No summary statistics for this variable, however most of the variables in the dataset describe this variable
    d. Year
        i. This variable is integers ranging from 1959 to 2019
    e. songLength
        i. Min: 0
        ii. Q1: 1039

   iii. Median: 1473
   iv. Q3: 2121.25
   v. Max: 825001
   vi. Mean: 3230.48
   vii. Standard Deviation: 26852.73

f. numExclamation
   i. Min: 0
   ii. Q1: 0
   iii. Median: 0
   iv. Q3: 0
   v. Max: 670
   vi. Mean: 2.52
   vii. Standard Deviation: 20.09

g. numBigWords
   i. Min: 0
   ii. Q1: 31
   iii. Median: 47
   iv. Q3: 73
   v. Max: 42079
   vi. Mean: 133.78
   vii. Standard Deviation: 1254.87

h. numSmallWords
   i. Min: 0
   ii. Q1: 143
   iii. Median: 205
   iv. Q3: 296
   v. Max: 96589
   vi. Mean: 401.05
   vii. Standard Deviation: 3169.06

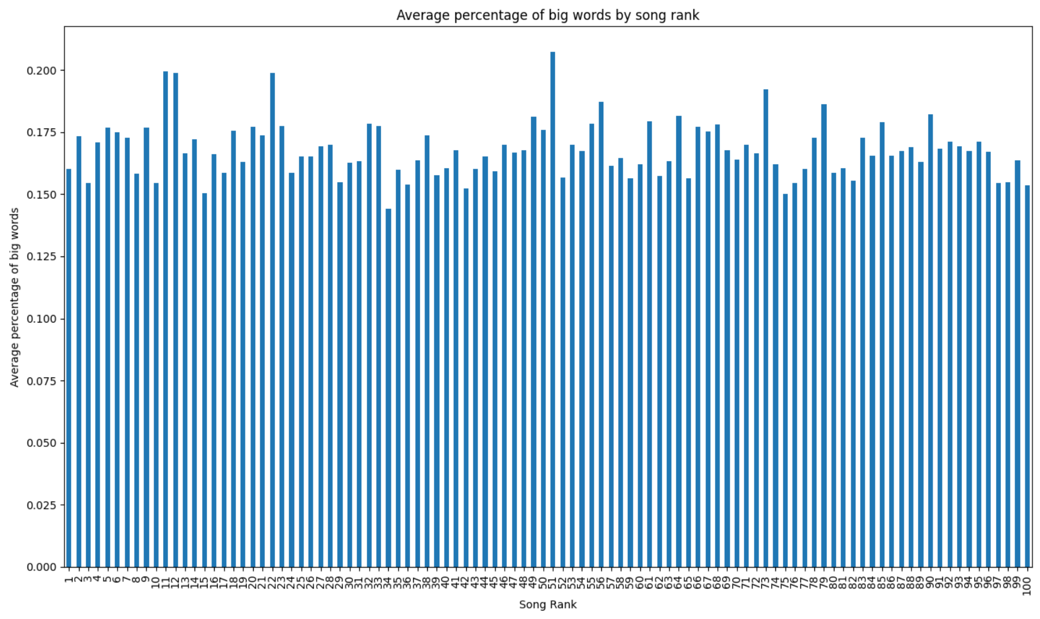
i. numWords
   i. Min: 0
   ii. Q1: 209
   iii. Median: 296
   iv. Q3: 426
   v. Max: 149170
   vi. Mean: 619.59
   vii. Standard Deviation: 5043.89

j. percentSmall
   i. Min: 0
   ii. Q1: 0.6451
   iii. Median: 0.6923
   iv. Q3: 0.7386
   v. Max: 1
   vi. Mean: 0.6912

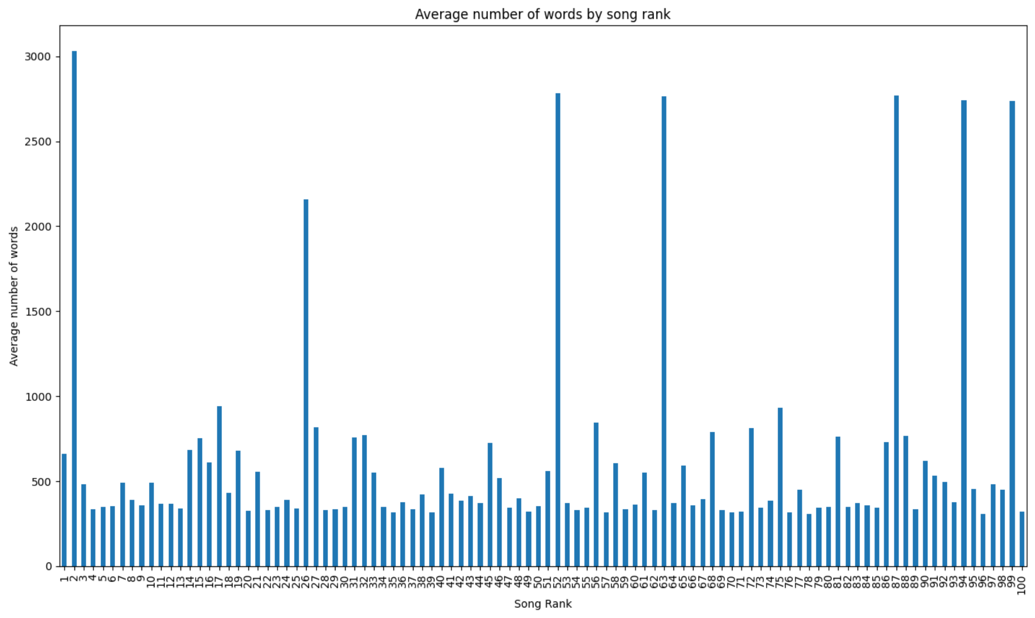
        vii. Standard Deviation: 0.09312

    k. percentBig
        i. Min: 0
        ii. Q1: 0.1233
        iii. Median: 0.1607
        iv. Q3: 0.2015
        v. Max: 1
        vi. Mean: 0.1680
        vii. Standard Deviation: 0.0805

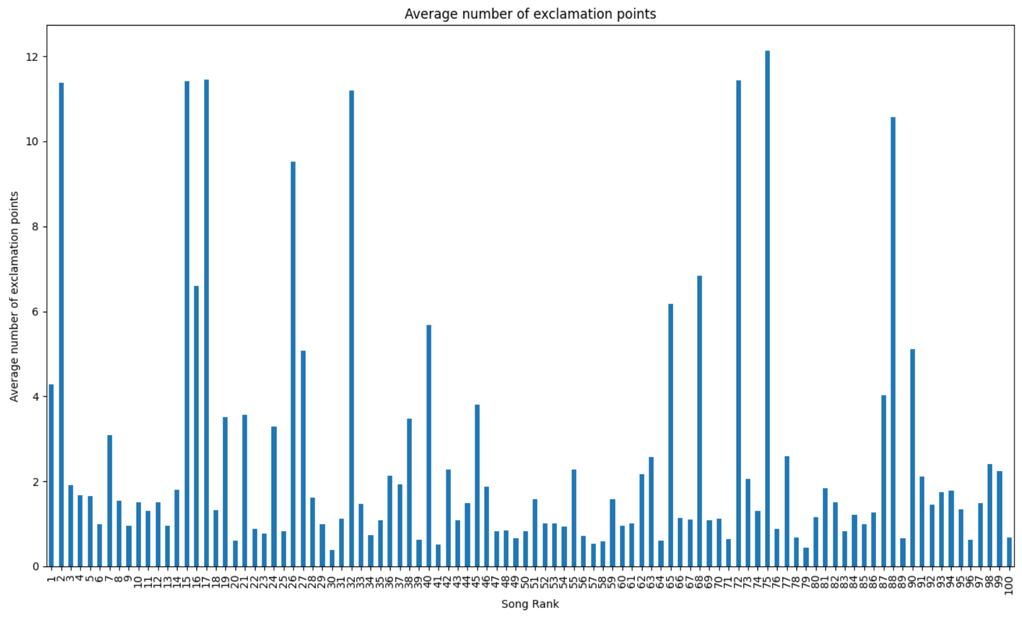
3. Three interesting graphs



    a.



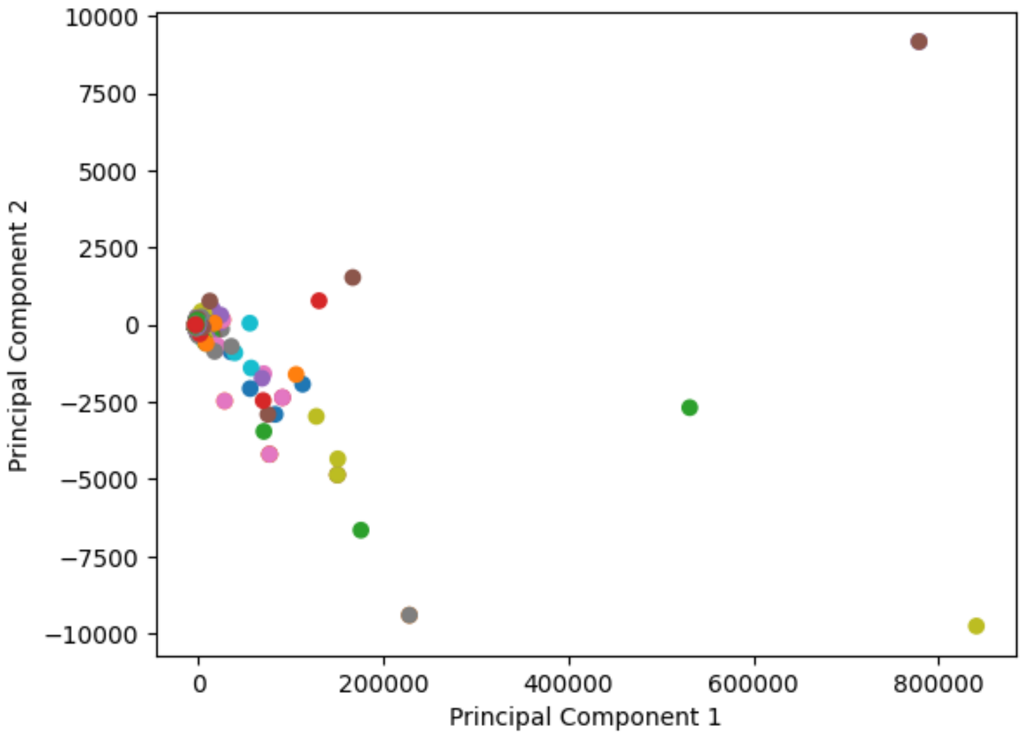
b.

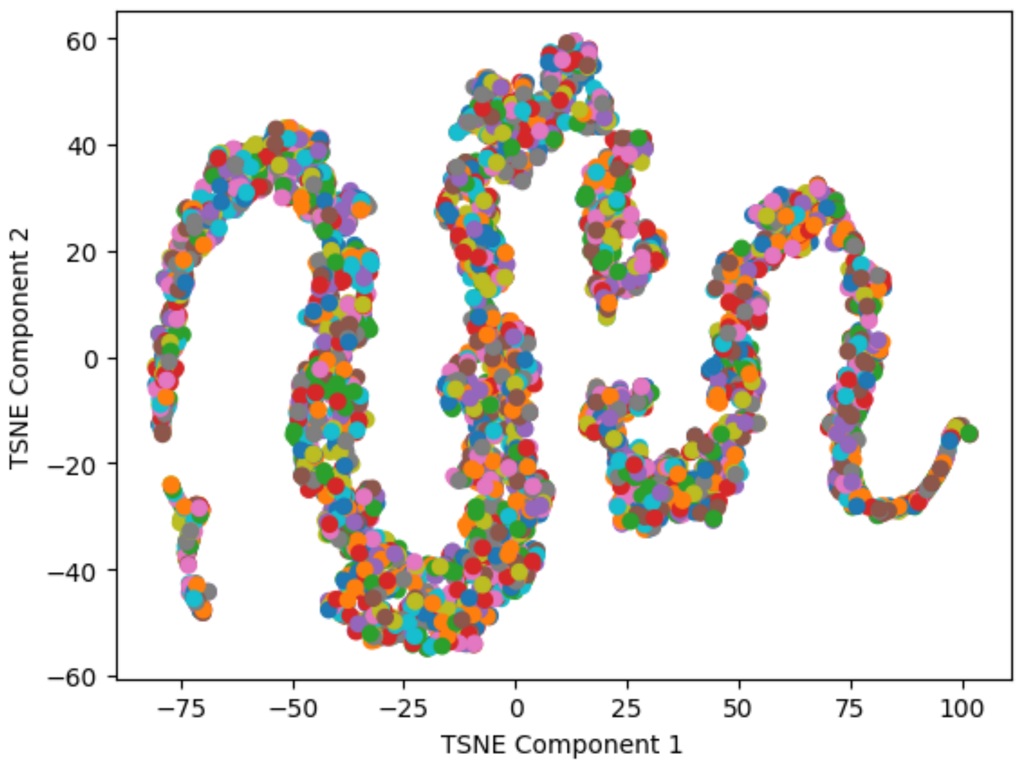


c.

There are a few notable findings from this analysis. The most obvious comes from the graphs. These graphs show that when comparing with song rank, that each type of variable is spread sporadically across the ranks. This suggests that there is likely not a relationship between the song's rank and its lyrics without regard to the lyrics' meaning. Something that is also very apparent is that there is at least one extreme outlier in the data. This song has 825000 words. This would seem to suggest that there may just have been excessive onomatopoeia, however since the song has 42000 words over 5 letters in length, that theory is debunked. This suggests that there is likely some benefit to using outlier resistant models.

These patterns are further enforced when examining the PCA and t-SNE graphs. (The points are colored according to whether the song was in the top 10, 20, 30, etc.)





These graphs both seem to reinforce the observation that there may be no relationship between the rank of a song and its lyrics.

Methods

Feature Engineering

Since this is non-tabular data, I wanted to engineer some numerical variables and analyze those in comparison with the meaning of the lyrics. The features I engineered are detailed above in the EDA, but they all have to do with counting words, letters, or the appearance of certain characters.

As discussed in the EDA, there were some outliers among the songs. As such, I chose to engineer a couple of ratio variables. These were to give a metric that would make comparing certain aspects of the music more logical. Specifically, the concentration of small words vs. big words.

Models Used

Knn - K-nearest neighbors is a supervised machine learning method. It uses the n nearest points to a datapoint to either assign it to the majority class or averages them to predict that point's value

Linear Regression - Linear Regression is a supervised machine learning method that minimizes the distance from a line to each point in the dataset. This line is written as a linear equation, hence the name.

Elastic Net - Elastic Net is a Linear Regression method with L1 regularization.

Random Forest - Random Forest is a supervised machine learning method that fits many decision trees to bootstrap samples of the data. A point is run through these trees and each tree's output is a vote towards the value predicted for that point.

Adaboost - Adaboost is a supervised machine learning method that fits many shallow decision trees (trees that make a single decision). These trees are then tested on datapoints and points that they fail to predict correctly, are given a heavier weight during training. When completed, the decision trees each vote towards the value predicted for the point.

ANN - Artificial Neural Networks are a machine learning method. The network mimics how the brain's neural network works using many 'neurons' and 'Activation functions'. The neurons are adjusted using gradient descent and the last layer of the neural network makes the prediction for the point.

Ransac Regression - Ransac Regression is a supervised machine learning method that is robust to outliers. This method selects two points and creates a line between them. The absolute value of the distance of all other points to this line is recorded. The median absolute value of the distance is used to decide whether a point is an inlier or an outlier. This process is repeated until the line through two points with the most inliers is found. Linear Regression is then used to fit a line through the two points and the inliers associated with the line. This line is the model used.

Huber Regression - Huber Regression is a supervised machine learning method. It works very similarly to linear regression. Just like linear regression, coefficients are initialized and residuals are calculated. However Huber Regression then find the median residual, calculates the scale by dividing it by 0.6745 (the values within one standard deviation), then divides all the residuals by this scale. The epsilon hyperparameter than sets the boundary for whether a scaled residual is considered extreme or not. If it is extreme, its error is calculated using the mean absolute variable, if it isn’t extreme, it uses the mean squared error. This ensures that the values that are outliers have less weight on the final model. Once the weights are calculated, the coefficients are updated until the error has been minimized.

Theil-Sen Regression - Theil-Sen Regression is a supervised machine learning method that is robust to outliers. First the slope between every possible pair of points is calculated. The median slope and median intercept are then calculated and make up the line that is used as the model.

Lyrical Analysis Models

The following table details out the information for the models that were fit to the vectorized lyrics.

Model Name	Hyperparameters Explored	Hyperparameters Chosen	Model Performance

KNN	Leaf_size
	N_neighbors
	P
	Weights
		Leaf_size = 10
		N_neighbors = 100
		P = 2
		Weights = Uniform
			R^2 score: -0.001

Linear Regression	Default Variables used	Fit_intercept = True
		Positive = False
			R^2 score: -11.699

Random Forest	N_estimators
	Max_features
	Bootstrap
	Min_samples_split
	Min_samples_leaf
		N_estimators = 250
		Max_features = log2
		Bootstrap = True
		Min_samples_split = 10
		Min_samples_leaf = 3
			R^2 score: -0.007

Adaboost	N_estimators
	Learning_rate
	Loss
		N_estimators = 100
		Learning_rate = 1
		Loss = Square
			R^2 score: 0.0003
			MSE: 860.869

Neural Network	Number of neurons (per layer) Activation Function (per layer)	Layer 1 : 17, sigmoid Layer 2 : 374, sigmoid Layer 3 : 191, tanh Layer 4 : 318, relu Layer 5: 36, sigmoid	R^2 score: -0.7741 MSE: 859.166
Ransac Regression	Estimator	Estimator = LinearRegression()	R^2 score: -109.766

Huber Regression	Epsilon
	Alpha
	Max_iter
	Fit_intercept
		Epsilon = 1.5
		Alpha = 0.001
		Max_iter = 50
		Fit_intercept = True
			R^2 score: -0.5673

Theil-Sen Regression	Max_subpopulation
	N_samples
	Fit_intercept
		Max_subpopulation = 200,
		N_subsamples = 300,
		Fit_intercept = True
			R^2 score: -0.194

Numerically Engineered Variables

The following tables includes the models that were fit to the numerical variables that were engineered.

Model Name	Hyperparameters	Hyperparameters	Model Performance

	Explored	Chosen	

KNN	N_neighbors
	P
	Weights
		N_neighbors = 50
		P = 2
		Weights = Uniform
			R^2 score: -0.015

Elastic Net	Tol
	Selection
	Max_iter
	L1_ratio
	Alpha
		Tol = 1e-05
		Selection = Cyclic
		Max_iter = 100
		L1_ratio = 0.25
		Alpha = 100
			R^2 score: -0.001

Random Forest	N_estimators
	Max_features
	Bootstrap
	Min_samples_split
	Min_samples_leaf
		N_estimators = 100
		Max_features = log2
		Bootstrap = True
		Min_samples_split = 2
		Min_samples_leaf = 5
			R^2 score: -0.044

Adaboost	N_estimators
	Learning_rate
	Loss
		N_estimators = 75
		Learning_rate = 0.1
		Loss = linear
			R^2 score: 0.0008
			MSE: 866.486

Neural Network	Number of neurons (per layer) Activation Function (per layer)	Layer 1 : 60, relu Layer 2 : 163, relu Layer 3 : 383, sigmoid Layer 4 : 148, tanh Layer 5: 38, tanh	R^2 score: -0.0593 MSE: 859.145
Ransac Regression	Estimator	Estimator = LinearRegression()	R^2 score: -156.166

Huber Regression	Epsilon
	Alpha
	Max_iter
	Fit_intercept
		Epsilon = 10
		Alpha = 0.1
		Max_iter = 200
		Fit_intercept = False
			R^2 score: -0.003

Theil-Sen Regression	Max_subpopulation
	N_samples
	Fit_intercept
		Max_subpopulation = 50
		N_samples = 100
		Fit_intercept = True
			R^2 score:-0.367

Discussion on Model Selection

Patterns across Models

One pattern seen across models is that the engineered numerical features tend to perform better than the vectorized lyrics. Regression models tended to perform worse than the other methods, however one notable selection was the elastic net model that was fitted. It ended up performing as the third best model. Ensemble methods tended to perform slightly better than the other methods. Adaboost performed the best and random forests performed slightly above average.

Major Pitfalls

The most obvious pitfall is that *none* of the models are effective in predicting the rank of a song based on its lyrics. In fact, all, with exception to the Adaboost models, performed so bad, that you would be better off guessing that a song would feature at 50 every single time. I think it is because of this that models like KNN chose a hyperparameter of n so big, that it basically was doing just that. The Adaboost models did work a smidgeon better than guessing, but its predictive edge is so small, that practically, it would be better to guess a rank of 50 every time.

One pitfall is that a number of models such as Adaboost Theil-Sen regression took a very long time to run. Luckily, the Adaboost models did fit fast enough that they were able to be used, however for the lyrical analysis, the number of words examined had to be reduced and a sample of songs had to be used in order to fit the model.

Hyperparameter selection was fairly successful. Randomized grid search and cross validation was used for each model to determine what the best hyperparameters would be. For the neural network, the Optuna library was used to run trials to pick the number of neurons and activation function used at each layer.

Detailed Discussion on Best Model

The best models that were used were the artificial neural networks in regards to the Mean Squared Error. The Adaboost models were the best models by the R-squared score. For this study, we chose the adaboost models as they were the only models to have a positive R-squared value.

To tune the adaboost models, the following hyperparameter grid was used:

```
boostGrid = {
    'n_estimators' : [25, 75, 100, 500],
    'learning_rate' : [0.01, 0.1, 1, 1.5],
    'loss' : ['linear', 'square', 'exponential']
}
```

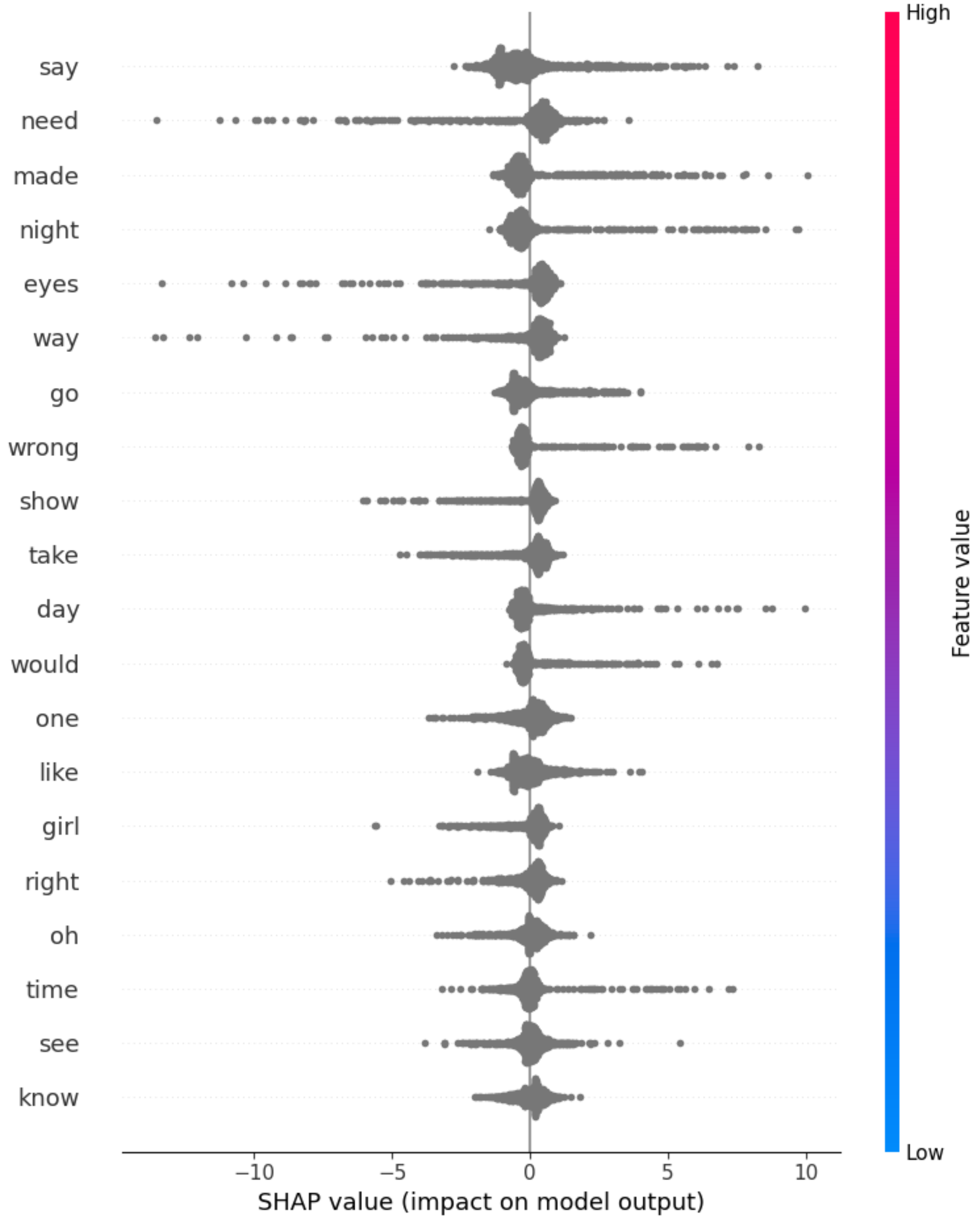
The number of estimators indicates how many rounds of boosting are to be undergone. The learning rate is a measure of how quickly the weights are adjusted with each round and the loss function indicates the measure of how much the deviations from the truth should affect the boosting.

The two models were used:

Lyrical Adaboost

The Lyrical Adaboost model used the following hyperparameters, n_estimators = 100, learning_rate = 1, and loss = Square. The performance metrics measured was mean square error and the R squared score. These were the results R^2 score: 0.0003 MSE: 860.869.

The following graph shows the Shap values for the model. The values indicate that the words “say”, “made”, and “night” were most associated with songs that were predicted with higher ranks (songs ranked worse). Whereas the words “need”, “eyes”, and “way” were most associated with songs that were predicted with lower ranks (songs ranked better).



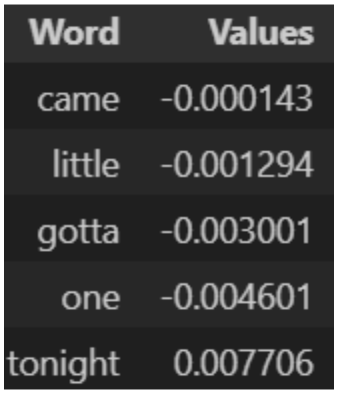
say
need
made
night
eyes
way
go
wrong
show
take
day
would
one
like
girl
right
oh
time
see
know
−10
−5
0
5
10
SHAP value (impact on model output)
High
Low
Feature value

Notable Shap Examples

*Note I tried for several hours to get pretty graphs for these specific examples, however it appears that the documentation for Shap is outdated. I eventually was able to make these dataframes with the shap values.

Hey Jude

Hey Jude is a classic song from the 1960s that reached number one on the charts. It was predicted by the model to be ranked 41 on the charts. The following words were most influential on this prediction according to their shap values

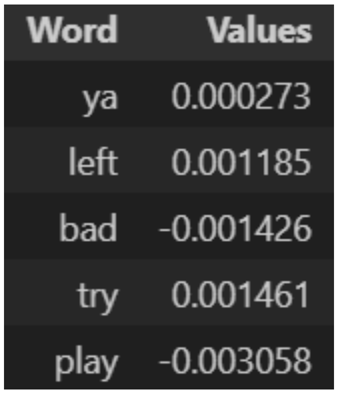


Word	Values
came	-0.000143
little	-0.001294
gotta	-0.003001
one	-0.004601
tonight	0.007706

It is interesting that the word ‘tonight’ was the only one of the top 5 words to have a positive impact on this prediction, especially since this word doesn’t appear in the lyrics. In fact, the only one of these words that does appear is the word ‘little’, which negatively influenced the prediction. It may be a bit confusing, but since songs with a higher rank are ones with smaller numbers, this means that not having the word tonight gave the song a lower rank where as having the word little increased its rank.

Surfin USA

Surfin’ USA is another classic song from the 1960s. Despite its continued popularity to today, it only reached number 2 on the Billboard Charts at its peak. Our model predicted that this song would be ranked 27 on the charts. The following words were most influential on this prediction (according to their Shap value).



Word	Values
ya	0.000273
left	0.001185
bad	-0.001426
try	0.001461
play	-0.003058

These values are all interesting as none of the words appear in the song. Because of that, it is interesting that not having the word bad influenced this song to be predicted with a higher rank.

Numerical Adaboost

The numerical Adaboost model used the following hyperparameters, n_estimators = 75, learning_rate = 0.1, and loss = Linear. The performance metrics measured was mean square error and the R squared score. These were the results R^2 score: 0.0008
MSE: 866.486

Conclusion and Next Steps

A best model for this project cannot be selected in good faith. This extensive analysis leads us to the conclusion that the lyrics of a song are not a good way to try and predict how well a song will perform on the billboard charts as various machine learning models were fitted. In fact, our calculations indicate that over 5000 models were attempted to be fit to this dataset.

In the future, it would be wise to either estimate a different metric using lyrics. One interesting one could be seeing if a song's release decade can be predicted using the lyrics. On the other side, we could try to estimate rank using a different variable. However, in order to do this, more data would have to be collected.