Predicting the Genre of Stories for Optimal Marketing

Ben McPeek

Problem Statement

A children's book company has a new series of funny ghost stories for the upcoming Halloween season. Before they advertise these books they wanted to find a way to get public insight on which genre sticks out the most for each story. Using the subreddits, r/Ghostories and r/funnystories, we will create a model that can interpret a post to which subreddit it belongs to. These subreddits will be a great way to determine a sample of public opinion on what constitutes a funny and ghost story without the costs of generating our own samples.

Success = Because the amount of our subreddits are fairly balanced we will use Accuracy

Null baseline = 52.1% accuracy

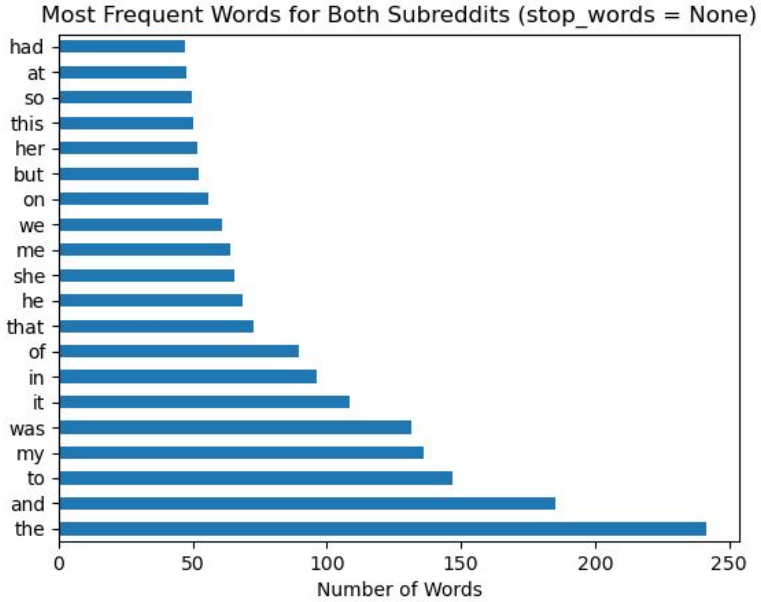
Background

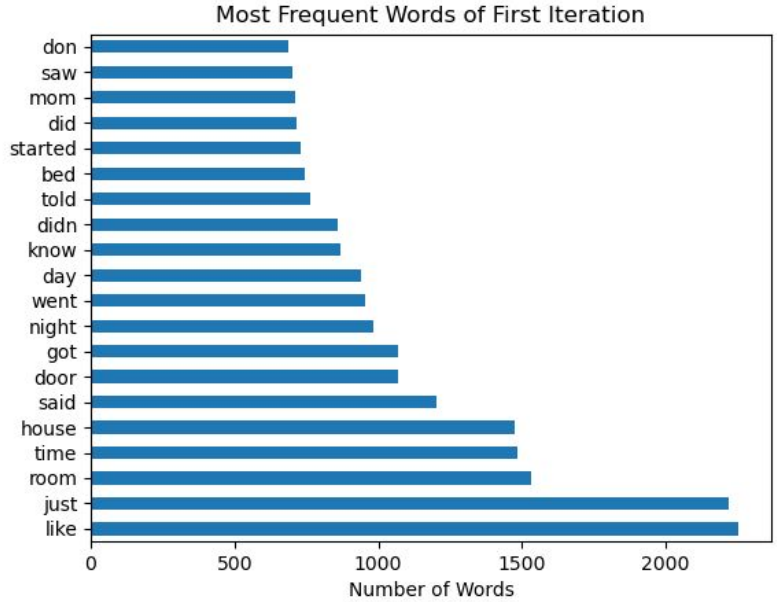
Because of Reddit's recent API update we were limited in the amount of posts we could pull from each subreddit. Filtering through the newest and most popular posts for each subreddit I was able to pull on average 5 posts per day. In future adaptations of this project I will aim to pull more data for these models.

Subreddits = r/Ghoststories, r/funnystories

Total Unique Posts = 1,977

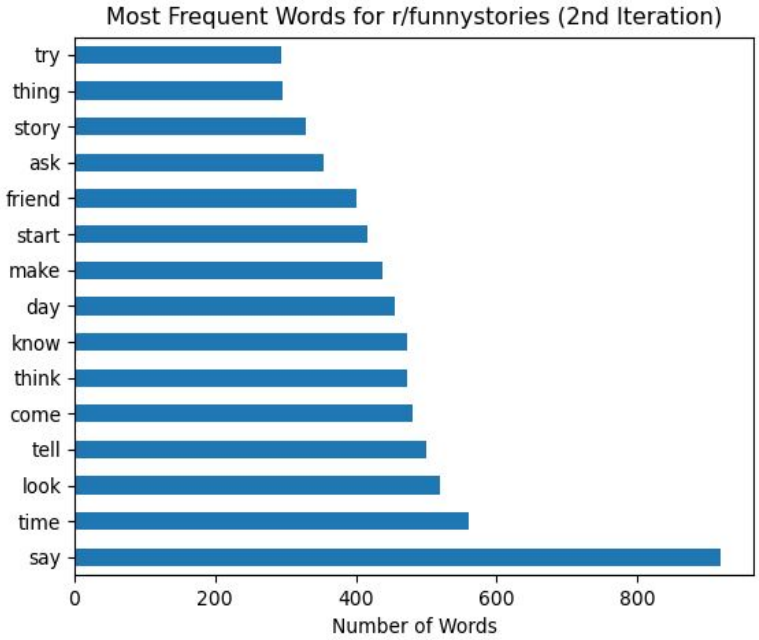
EDA and Preprocessing Pt. 2

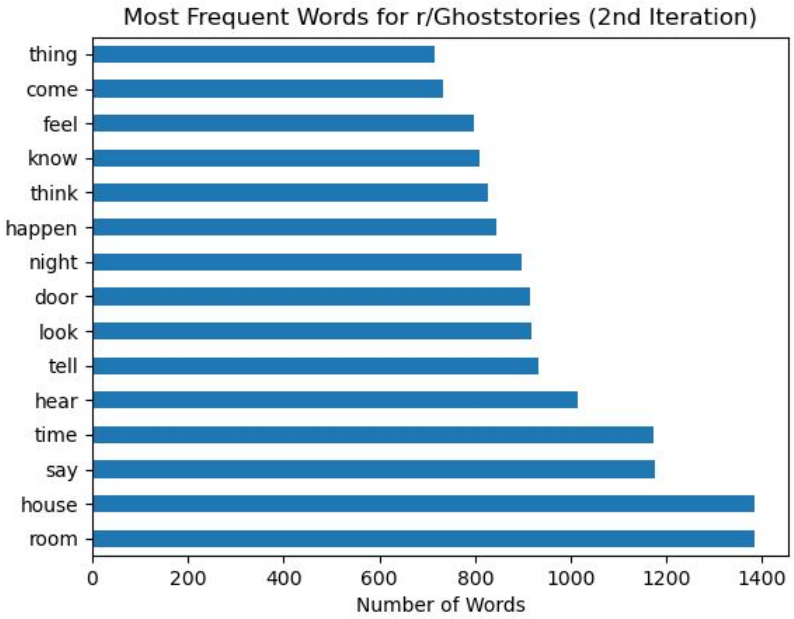




Before iterating our posts I checked to see if there was any significance in the most frequent words. After seeing they were all noise words I used RegExpTokenizer for my first iteration. RegexpTokenizer has split words based on apostrophes. This will give us half words such as 'don', 've', and 'didn' that make it difficult to interpret.

EDA and Preprocessing Pt. 2





We can still see a fair amount of overlap however the overlapped words differ in frequency. We can also find unique words that seem to resemble the characteristics of their subreddits.
- Ghoststories: 'room', 'house', 'hear', 'door', and 'night'
- funnystories: 'friend', 'day', 'story'

Test model: Logistic Regression

- We will use a Logistic Regression model to determine our best features. We set up two different iterations of our features that progressively filter more. We will run two different Logistic Reg models and see which performs the best under which iteration. Based on which model performs best will determine which iteration we will use for further complex models.

RegExpTokenizer Iteration: {Train: 0.975, Test: 0.923, Cross-Val-Score: 0.920} (Overfit: 0.052)

Spacy Iteration: {Train: 0.969, Test: 0.925, Cross-Val-Score: 0.922}

Random Forest and Naive Bayes Models

I ran our spacy iterated features through these two models and found the Random Forest model underperformed in comparison to our baseline LogReg.

Our Naive Bayes model is performing better than our baseline model with an increase in accuracy of our Train set of 0.7% and our Test set of 1.2%

Baseline Model: {Train: 0.969, Test: 0.925, Cross-Val-Score: 0.922}

Random Forest: {Train: 0.901, Test: 0.875, Cross_Val_Score: 0.856}

Naive Bayes: {Train: 0.976, Test: 0.937, Cross_Val_Score: 0.934}

Conclusion

There are two ways to generate insightful predictions to best market Children's Book Company's new Halloween series. We can use our Naive Bayes model to best accurately predict the genre of each story and market them solely based on the predicted genre. This form of marketing neglects the amount of presence of the lesser prominent genre and will put all of its marketing effort to the predicted genre.

We can use our second best performing model, Random Forest, to find the presence of the genres to each story. This will allow the marketing of each story to be more nuanced and specific.

Conclusion

Story Sample:
It starts with a child meeting a lady with red lips and long fingers. She keeps asking him whether he knows what she can do with these red lips and these long fingers. The child is scared but cannot escape this lady with her red lips and long fingers. This repeats itself a couple of times, getting scarier and scarier. At last, he will ask her what she can do with those red lips and long fingers. And she will move her long fingers between her lips, making a funny sound.

r/funnystories = 0.535
r/Ghostories = 0.465