Predicting Popularity: A Social Media Experiment

Hannah Kanjian and **Bryce Anthony**
{hakanjian,branthony}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

Social Media is a constant influence on our lives and can be very powerful when it comes to marketing. Current marketing tools focus on analyzing data after a post is posted, but we wanted to predict how a post will do before it is sent out. For this project we built several machine learning models that use the text content of a post to predict the popularity of a post, With the popularity of a post being measured by the total number of up-votes a post will receive or the total engagement a post receives (a sum of the up-votes, down-votes, and number of comments). Our best model was able to accurately classify posts into 4 levels of popularity with an accuracy of around 80%.

1 Introduction

Starting with a Kaggle dataset containing data from "Reddit Top 1000 Posts," we analyze how the title of a Reddit post and time the post was created affect the total number of likes and overall engagement a post will receive.

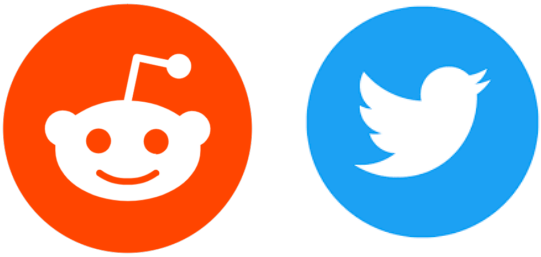

Figure 1: Reddit and Twitter Logos

Other groups have worked on similar problems to ours:

- Yu et al. (Yu, Chen, and Kwok 2011) used content on Facebook, to examine the relationship between the number of likes a post received and the content and media type of the post.
- Dai and Wang (Dai and Wang 2021) built a model that utilizes a psychological approach to understand social media marketing.
- Daga (Daga et al. 2020) experimented with a number of models using a similar approach to us, but focusing solely on Twitter data.

However, our project builds upon existing research by looking to see if there are specific trends in what the public likes to see and if these trends share objective features that can be used to predict the popularity of a post

When examining the data (figure 2), on first glance we can see that the number of up and down votes are correlated. In this graph, as the number of up votes increases, the number of down votes also increases. Thus we can see that the more popular a post is the more diverse 'engagement' it receives.

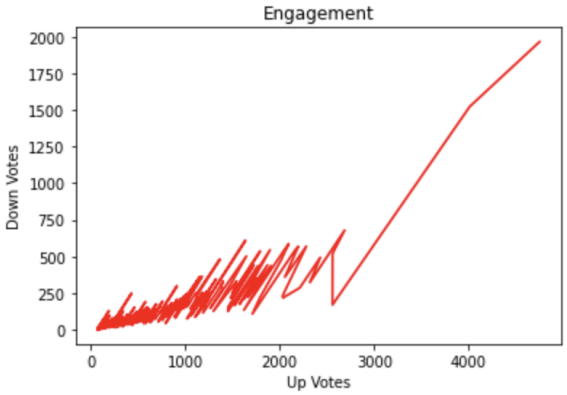

Figure 2: Visualizing Engagement through Up and Down Votes

To train a model with this data, we needed to do a great deal of preprocessing. In order to get the data ready we used a classic approach to Natural Language Processing (NLP) of word embeddings in addition to analyzing the sentiment of each post. This additional data derived from each post was then used to create several models to predict the popularity of a post. The rest of this paper outlines the prepossessing steps performed on the data, our experimental procedure, a discussion of our results, and the broader implications of our work.

2 Preprocessing

While machine learning has come a long way in the past couple years, it continues to have limitations with text data and specifically social media. What makes a post funny or emotional requires a distinctly human context that the machine cannot easily pick up. We can combat this lack of context by utilizing an NLP technique of word embeddings.

To create the word embeddings, we first removed all of the "stop words" or words not important to the overall meaning of the text such as "and", "or", "a", and other words of this nature. Once our text was cleaned of the "stop words" we wanted to embed them by transforming them into their vector representation. We used a tensorflow Tokenizer to strip the text of any punctuation to then tokenize and assign numbers to the words based on tf-idf (figure 3).

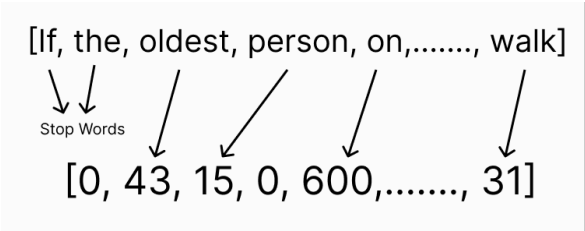

Figure 3: Basic Structure of Word Embedding

To change our data into a classifier format we analyzed the skew of the up-votes and total engagement to create four categories: low, medium, high, and very high engagement. These categories, represented by numbers 1-4, became our targets. We made sure to make these labels as balanced as possible by using the quartiles of our target as the cutoffs for each level.

We wanted to see if a model could accurately predict the total engagement of a post (i.e the summation of the up-votes, down-votes, and number of comments) to get a more well rounded view of engagement. We also wanted to test the relationships between up votes (positive engagement) and the rest of the data so we created models to predict the number of up-votes a post would receive.

After reformatting the data, we performed a test-train split, and separated our targets (the number of up votes, or total engagement) from our features (the text data and time stamps).

3 Experiments

The data we used for our experiments was the "Reddit Top 1,000 Posts" data set from Kaggle. We used the r/Shower Thoughts data set for most of our testing and expanded our best performing models to a much larger data set consisting of the top 1,000 posts from the r/World News, r/Today I Learned, r/Movies, and r/Jokes in addition to the r/Shower Thoughts data set. We reserved 20% of each data set for testing our final models, while the remaining 80% was divided into 80% for training our models and 20% for validating our model results.

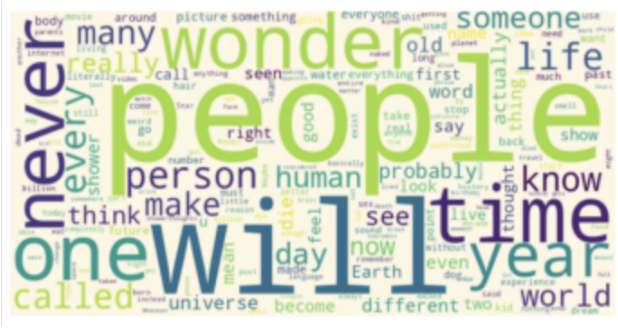

Figure 4: A Word Cloud of the the most frequent words in the dataset

For both the Shower Thoughts data set and more general data set We experimented with the three models listed below:

- Logistic Regression
- Random Forest Classifier
- SVC (Support Vector Classifier)

We optimized the three models to classify the level of likes on a post by using the word embeddings from the post and creation time as inputs. Then in addition to classifying the level of likes on a post by using the word embeddings, we used the optimized models to classify the level of likes on a post by using the word embeddings along with the positive, neutral, and negative sentiments from the post's text. These models were also used to classify the level of total engagement on a post using just the word embeddings, and again using the word embeddings along with the positive, neutral, and negative sentiments from the post's text. The final models were also tested on the larger dataset to see how well the models would generalize to other data.

For the logistic regression model we settled on a model using L2 norm and a inverse of regularization strength of 1.0 as it yielded the best results. We also tested the L1 norm and a combination of L1 and L2 norms, using inverse of regularization strength values of .001, .10, 5, and 20. For the Random Forest Classifier model we tried three variants that to measure the quality of a split differently. Although we found the Gini impurity to be the best, log loss, and entropy were also tested. For the Support Vector Machine model, we tested values of .5, 1, and 2 for the regularization strength and both a radial basis function kernel and polynomial kernel. The results of our experiments are discussed in the next section.

4 Results

While we experimented with a broader range of topics, it turns out that our best model was significantly more accurate on a specific topic, in other words with only one subreddit. Therefore, if the model was being used to predict the popularity of a specific post it would be helpful to train it with other posts of that topic. By narrowing the scope, we are able to give the machine more "context".

For the dataset using posts from the r/Shower Toughts subreddit, our best model was able to correctly categorize the number of up votes a post would receive with an accuracy of nearly 80% (77.5%). In figures 5 and 6, we display the accuracy of the models predicting Up-Votes and Total engagement for the r/ Shower Thoughts dataset. In addition to the models used, the table has columns for each category of features that were used by the model in addition to the time the post was made. Each model was trained using just Word Embeddings (WE) as inputs, with Word Embeddings and Sentiment Analysis (WE + SA) as inputs, and in the final column are the results of testing the best version of each model.

Model	WE	WE + SA	Test
Logistic Regression	20.5%	23.5%	25.5%
Random Forest Classifier	66.5%	71.9%	77.5%
Support Vector Classifier	20.5%	23.5%	25.5%

Figure 5: Accuracy for **Up-Vote Predictions**

Because the splits of our target variables (likes and engagement) are based on the quartiles of the dataset, the accuracy of our baseline model was 25% for both the model predicting likes and the model predicting overall engagement. For the task of predicting up votes (or likes) The Logistic Regression models had accuracies between 20% and 25% in our training but performed about the same as the dummy model in testing with a test accuracy of 25%. Similarly the SVC model performed worse than the dummy model during training and performed about the same as the dummy model during testing with an accuracy of 25.5%. Both the Logistic Regression and SVC models consistently outputted only one label, namely the medium engagement category.

Model	WE	WE + SA	Test
Logistic Regression	25.6%	21.8%	-
Random Forest Classifier	70%	69.4%	69%
Support Vector Classifier	25.6%	21.9%	-

Figure 6: Accuracy for **Total Engagement Predictions**

For the task of predicting total engagement on the r/shower thoughts dataset our best model was the Random Forest Classifier only using word embeddings to make predictions. This model had an accuracy of 70% during training and 69% during testing. Although the train accuracy's of the Logistic regression model and SVC model performed the same as the dummy model during training, when sentiments were used in addition to the word embeddings the accuracy of these models also decreased, making them worse than the dummy model.

For the r/Shower Thoughts dataset, our best models were the random forest classifier using word embeddings with sentiment analysis to predict likes and the random forest classifier using word embeddings to predict the total engagement. Since the test accuracy of these two models was so close 77.5% and 75%, both over 2.5x the accuracy of

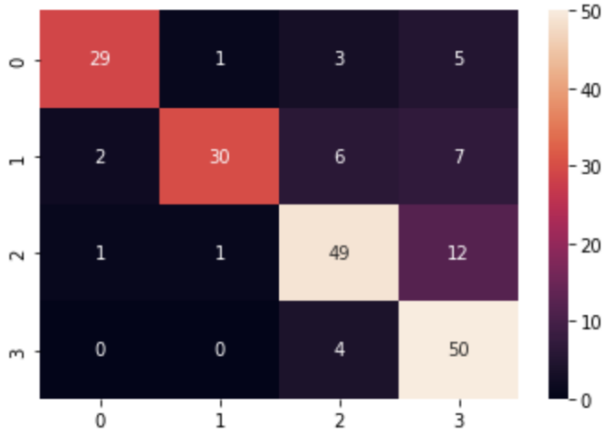

Figure 7: Heat Map of Total Engagement Predictions from the Random Forest Classifier using word embeddings

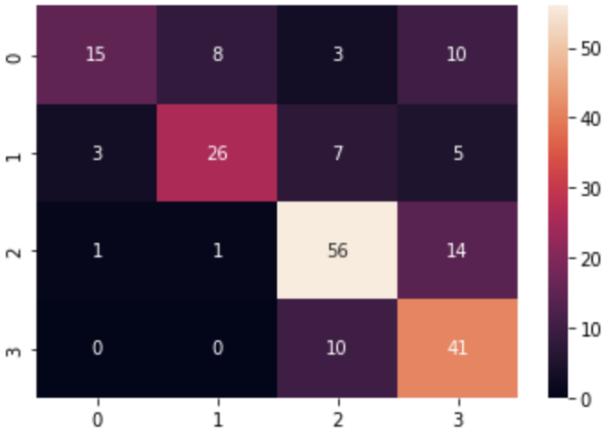

Figure 8: Heat Map of Up-Vote Predictions from the Random Forest Classifier using word embeddings and Sentiment Data

the dummy model, we wanted to compare the results obtained by each model using the heat maps in figures (7 and 8). When looking at their respective heat maps cc, you will notice that, while not perfect, the "heat" is concentrated on the diagonal, meaning the model correctly predicts the label. It also shows us that these model are more likely to under-predict popularity than over-predict popularity. Making them well suited for marketing endeavors.

When Applying these models to the larger dataset consisting of posts from the 5 subreddits mentioned earlier, our best model was less accurate than the model used for an individual subreddit. When applied to the broader dataset, The model's accuracy was still over 2x more accurate than the dummy model with an accuracy of 65%. Although it performed worse than the more specialized model, this model also had a tendency to underpredict the popularity of a post rather than overpredict popularity of a post.

5 Broader Impacts

Since social media is such a large part of our day to day lives, marketing has a huge impact on how we think and what we buy.

This model has the capacity to do great good by influencing people to buy ideas or things that will help them, but on the flip side, bad actors can use this to influence people to do things that will have a negative impact on society.

In addition, the widespread use of this technology could lead to the creation of very similar content - ultimately decreasing the diversity of content we see on the web. Additionally unless the model is continuously trained, predictions from this model would highly rate content that would have previously been popular, potentially changing the target audience.

6 Conclusions

In this paper we explored social media as an influential force. We set out to see if we could build a machine learning model to predict the popularity of a Reddit post before its been released. We used NLP techniques and sentiment analysis to give our models more meaningful information and used this data to make predictions about the popularity of a post. This was best accomplished for predicting Upvotes/likes using a Random Forest Classifier that used word embeddings of the title in combination the sentiments of the title from the post (77.5% accuracy).

For predicting total engagement this was best accomplished using a random forest classifier using only word embeddings (69% accuracy). Both models were more than 2x more accurate than the dummy model and highlight the idea that there may be objective features that can be used to predict the popularity of a post. While our model performs well compared to the dummy model, the inability of the model to more accurately predict the popularity of a post emphasizes the complexity of the human mind when it comes to interpreting information, although we found it is possible to predict a post's performance to a certain degree. In terms of further work, we believe there is a lot to be learned from a deep learning perspective. We would continue by running the word embedding through a deep learning model.

7 Contributions

Typically working with one computer open to the code, and the other open to the document, we were able to offer input on every aspect of the assignment and complete it during our in-person meetings. We both proof-read the entire document as well.

References

Daga, I.; Gupta, A.; Vardhan, R.; and Mukherjee, P. 2020. Prediction of likes and retweets using text information retrieval. *Procedia Computer Science* 168:123–128. "Complex Adaptive Systems"Malvern, PennsylvaniaNovember 13-15, 2019.

Dai, Y., and Wang, T. 2021. Prediction of customer engagement behaviour response to marketing posts based on machine learning. *Connection Science* 33(4):891–910.

Yu, B.; Chen, M.; and Kwok, L. 2011. Toward predicting popularity of social marketing messages. 317–324.