Classifying Subreddits throughout Reddit: Popular or Not?

Nicole Cruz

Santa Clara University

ncruz@scu.edu

CSCI 183: Data Science

I. Introduction

For my project, I decided to use linear classification through Logistic Regression and SVM to predict the class of every single subreddit as either: popular (1) or not popular (0). Popularity as described in the context of this project refers to how active the subreddit is at the moment.

II. Experimental Design

A. The Dataset

The original dataset used contains a list of subreddits that were directly copied and pasted from Reddit's own list of subreddits from r/ListofSubreddits that had originally been compiled in October 2018 [6]. Each of the subreddits were listed within one of the five categories listed:

- General Content: Subreddits that contain "Ask" reddits, gifs, images, and other miscellaneous content not pertaining to the other four.
- Gaming: Subreddits that contain topics regarding video games, consoles, specific game
- NSFW: Subreddits that contained "Not Safe For Work" content including gore, porn, or heavily inappropriate content.
- Politics: Subreddits that contain content related to politicians, or political issues.
- Memes: Subreddits that contained

The original dataset was initially collected on February 24, 2022 at around 9PM. The dataset is contained within "original.csv", which holds the initial 2591 subreddits before data cleaning occurred. The following files contained in the same folder as the code are described below:

- "original.csv": The names of the 2591 subreddits that have been on r/ListOfSubreddits since October 14, 2018.
- "subreddits.csv": The cleaned dataset that contains the names of the 2524 subreddits, which excludes all banned or inaccessible subreddits.
- "reddit_data.csv": The dataset after collecting all the features necessary to perform Logistic Regression and SVM.
- "reddit_cleaned.csv": The dataset includes the target variable and is the dataset that is performed on by Logistic Regression and SVM.

B. Data Cleaning

To begin cleaning the dataset, I chose to use PRAW (Python Reddit API Wrapper) to access each individual subreddit contained within "original.csv". By doing so, I can individually check if the subreddit still exists without having to open up a new browser window that has the name of the subreddit I need. Instead, if I attempt to access the subreddit needed and I get an error return of HTTP 403 or HTTP 404, then I know that the particular subreddit is inaccessible. The HTTP 403 response occurs when the server understands that the program is attempting to access the subreddit, but is unable to because it's forbidden [1]. Likewise, for HTTP 404, the error implies the subreddit could not be found [2].

The inaccessibility of a subreddit can occur for many reasons, some of which include: no moderators, unmoderated, copyright infringement, breaking of Reddit's Terms of Service, or the subreddit's original name being renamed or redirected to a new site that was not originally available in the list.

Data cleaning occurred from 3 PM PST to 2 AM PST starting from March 7, 2022 to March 8, 2022 to get the variety of the number of active users in different timezones.

C. *Features and Target*

The following features were considered as part of the dataset: the ratio_top, ratio_hot, ratio_cont, ratio_rise, and user_count. The category for each of the subreddits (e.g. General, Gaming, NSFW, Politics, and Memes) was not considered as a feature to only give more focus to numerical features in the dataset.

To get the dataset's original "target" variable, each of the numerical columns were compared to their respective average across all subreddits. "reddit_cleaned.csv" contains the full, cleaned dataset with the features and target variables. The target variable was determined by arbitrarily assuming that if the subreddit had 3 or more of the numerical features greater than their respective averages, then the subreddit would be considered "popular". The reason for picking the average as the threshold was because the average of a dataset is generally considered to be a representative sample of the data that they're the average of. [5]

Generally, popularity is defined as being based only on the number of subscribers to that particular subreddit [3]. However, I also wanted to have the other numerical values be accounted for and this is why the other numerical values were compared to their averages. In addition, the reason for active user count as opposed to subscriber count allows for what subreddits are being mass-accessed by users during a day-to-day basis gives better insight into what subreddit is more popular in the current time. Popular subreddits counted only by subscriber count are not as indicative of what interests users at the moment.

The features used in the dataset are the following:

- ratio_top: The average of the upvote-ratio of the top 100 posts on the subreddit
- ratio_hot: The average of the upvote-ratio of the most "hot" 100 posts on the subreddit
- ratio_cont: The average of the upvote-ratio of the most controversial 100 posts on the subreddit
- ratio_rise: The average of the upvote-ratio of the top rising 100 posts on the subreddit
- user_count: The number of active users currently on the subreddit at the time of acquiring the data

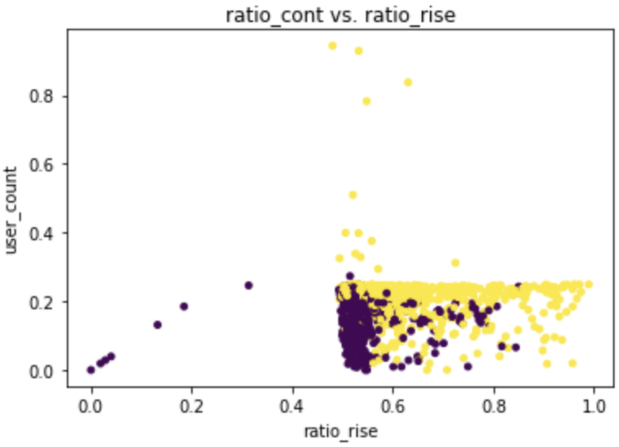
III. Implementation

A. *Graphs*

Graphs of each of the features were plotted against each other as a 2D scatter plot with the third dimension coloring each of the data points based on the target variable. A sample graph is shown in Figure 1, which plots ratio_cont & ratio_rise against one another, where values of 1 were colored yellow and values of 0 were colored purple. Similar graphs were plotted for the corresponding pairings:

- ratio_top & ratio_hot
- ratio_top & ratio_cont
- ratio_top & ratio_rise
- ratio_top & user_count

Figure I
RATIO_CONT VS. RATIO_RISE



B. *Data Split & Library*

Both libraries for Logistic Regression and SVM were from the sci-kitlearn library.

The dataset was split with an 80-20 ratio, where 80% of the data (2019 subreddits) is used for training while the 20% (505 subreddits) were used for testing on all 2524 subreddits.

SVM was performed with a linear kernel so it is easy to compare to Logistic Regression, which is restricted by a linear decision boundary only. [4]

IV. Results

C. Model Evaluation

Evaluating the model of each algorithm in classification utilized four performance metrics: Accuracy, Precision, Recall, and F-1 Score. These four metrics are commonly used for evaluating classification models. [7]

Table 1 contains the performance metrics for Logistic Regression.

Table 2 contains the performance metrics for SVM.

TABLE I

Performance metric for Logistic Regression

Performance Metrics				
Feature-Pairing	**Accuracy**	**Precision**	**Recall**	**F-1 Score**
ratio_top & ratio_hot	0.8336633663366336	0.7830882352941176	0.8949579831932774	0.8352941176470589
ratio_top & ratio_cont	0.7207920792079208	0.8384615384615385	0.4759825327510917	0.6072423398328691
ratio_top & ratio_rise	0.7861386138613862	0.7378277153558053	0.8382978723404255	0.7848605577689244
ratio_top & user_count	0.7524752475247525	0.7341269841269841	0.7613168724279835	0.7474747474747474
ratio_hot & ratio_cont	0.7900990099009901	0.7803921568627451	0.7991967871485943	0.7896825396825397
ratio_hot & ratio_rise	0.8534653465346534	0.8148148148148148	0.9016393442622951	0.8560311284046693
ratio_hot & user_count	0.7683168316831683	0.7518248175182481	0.807843137254902	0.77882797731569
ratio_cont & ratio_rise	0.801980198019802	0.8163265306122449	0.7142857142857143	0.7619047619047619
ratio_cont & user_count	0.6673267326732674	0.7938144329896907	0.3422222222222222	0.4782608695652173
ratio_rise & user_count	0.7287128712871287	0.68	0.7923728813559322	0.7318982387475538

TABLE 2

Performance metric for Support Vector Machine

Performance Metrics				
Feature-Pairing	**Accuracy**	**Precision**	**Recall**	**F-1 Score**
ratio_top & ratio_hot	0.8376237623762376	0.8055555555555556	0.8601694915254238	0.8319672131147542
ratio_top & ratio_cont	0.6554455445544555	0.8571428571428571	0.3076923076923077	0.45283018867924535
ratio_top & ratio_rise	0.7465346534653465	0.6936026936026936	0.8477366255144033	0.7629629629629628
ratio_top & user_count	0.7425742574257426	0.6833333333333333	0.8541666666666666	0.7592592592592592
ratio_hot & ratio_cont	0.801980198019802	0.7218045112781954	0.8807339449541285	0.7933884297520661
ratio_hot & ratio_rise	0.8336633663366336	0.7827715355805244	0.8893617021276595	0.8326693227091634
ratio_hot & user_count	0.80990099009901	0.7569721115537849	0.8444444444444444	0.7983193277310925
ratio_cont & ratio_rise	0.7524752475247525	0.7817258883248731	0.652542372881356	0.7113163972286375
ratio_cont & user_count	0.6633663366336634	0.8155339805825242	0.3574468085106383	0.49704142011834324
ratio_rise & user_count	0.7326732673267327	0.6631205673758865	0.8237885462555066	0.7347740667976425

D. Discussion

Based on F-1 scores seen in Table 1 and Table 2, the two main pairings that indicate popularity are ratio_top & ratio_hot and ratio_hot & ratio_rise. The F-1 Scores were chosen as an indicator for which pairings were more effective in deciding the predicted target because the F-1 Score allows for a balance between Precision and Recall measures. [7] Though not entirely at 1.0 which is the upper limit of F-1 Scoring, both pairings were significantly higher than those of the other pairings with their respective F-1 Scores.

Though accuracy across the different pairings ranged from the 60's to the 80's, I chose not to focus on the accuracy of the model when determining which pair of features would be indicative of popularity. This is because of the fact that accuracy itself might not be a good measure in datasets that are unbalanced.

In addition to this, accuracy itself may also have been affected by the fact that I chose to use SVM with a linear kernel, since SVM generally has more flexibility with the ability to choose between different kernels to get better accuracy. [4] However, by sticking to only the linear kernel, accuracy was bound to reduce because the data might not be as linearly separable, as seen from Figure 1, above.

V. Conclusions

From the results given and the discussion above, I determined that the one of the main features that affects popularity of a subreddit is ratio_hot, which has the average upvote-ratio of the top "hot" 100 subreddits that have been getting more upvotes and attention recently. The next two features to affect popularity in this instance are from the respective pairings: ratio_top and ratio_rise, with ratio_rise having a slightly greater effect. This may be due to the fact that ratio_rise itself focuses on what particular subreddit posts are getting activity at the moment, which is what I based my own definition of popularity for this project on.

For the different models, SVM would definitely be stronger if I went with a kernel other than linear, however, in comparing the two, SVM is determined to be more efficient due to its flexibility if I were to run the model again with a different kernel.

E. References

[1] *403 forbidden - http: MDN*. HTTP | MDN. (2021, November 25).from *https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/403*

[2] *404 not found - http: MDN*. HTTP | MDN. (2021, November 30). from *https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404*

[3] Aaron, J. (2022, January 14). *What is a subreddit on reddit ... and why should you care?* Neal Schaffer. from *https://nealschaffer.com/subreddit/#:~:text=Normally%2C%20the%20more%20%E2%80%9Cniche%E2%80%9D,a%20generally%20higher%20content%20quality*

[4] Bassey, P. (2019, September 19). *Logistic regression vs support vector machines (SVM)*. Medium.from *https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16*

[5] *How to analyze data using the average*. BetterExplained.(n.d.).from *https://betterexplained.com/articles/how-to-analyze-data-using-the-average/*

[6] r/ListOfSubreddits. "List Of Subreddits" October 14, 2018. from *reddit.com/r/ListOfSubreddits/wiki/listofsubreddits/*

[7] Shung, K. P. (2020, April 10). "*Accuracy, precision, recall or F1?"* Medium.from *https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9*