CSE 255 Assignment 2 : Upvotes Prediction for Reddit Submissions

Rakshit Wadhwa
University of California, San Diego
A53100056
rawadhwa@eng.ucsd.edu

Varun Garg
University of California, San Diego
A53103006
vgarg@eng.ucsd.edu

Kshitiz Gupta
University of California, San Diego
A53104364
ksg005@eng.ucsd.edu

ABSTRACT

In this paper we consider models for predicting the number of upvotes on a reddit submission. We examine features such as the number of votes, number of comments, time of submission, upvote history of users, images, and subreddits of the submission. We compare Support Vector Regression, Linear Regression, and Gradient Boosting Regression models for predicting the number of upvotes.

Keywords : Reddit, Submissions, Upvotes

1. INTRODUCTION

Reddit is a social networking website where users can submit and view content such as text posts, images etc. This content is organized into various categories, called subreddits. Each submission has votes, upvotes and downvotes associated with them. The number of votes is an indicator how trending the submission is. The number of upvotes is an indicator how many people liked the submission, and number of downvotes is an indicator of how many people disliked the submission. The difference between number of upvotes and downvotes is termed as score of that submission. Furthermore, user can post comments about the submission, and one can downvote or upvote a comment too. Reddit has a vast user-base, with users vote over 24 million times on the website each day [1]. This makes it interesting to study and predict the pattern for upvotes associated with a submission. The primary motivation of this paper is to gain better insights about factors that determine the number of upvotes on a submission, and hence, to develop a model which can predict the number of expected upvotes on a submission. In section 2, we discuss about the dataset and its important characteristics and relationships between different properties of the dataset. Section 3 defines the predictive task that this paper is focused upon. Section 5 and 6 dives deep into the features used to achieve the predictive task and the models used for the same. Finally, we conclude our findings and insights in section 7.

2. DATASET AND EXPLORATORY ANALYSIS

We use the already collected reddit data for our paper. [2][3].

The data consists of submissions which are identified by their image ids and titles.

Apart from image id and title, each submission has:

- number of upvotes, number of total votes
- number of downvotes
- number of comments
- raw time
- unix time
- subreddits
- user id of the user who submitted the submission

Overall, there are 132,307 submissions, with 16,736 unique images, and 63,328 unique users, and 868 unique subreddits.

The average votes received by any submission is 1883 votes. The ratio of upvote to downvotes on an average is 1.27.

A. Relationship between number of upvotes, downvotes and total number of votes.

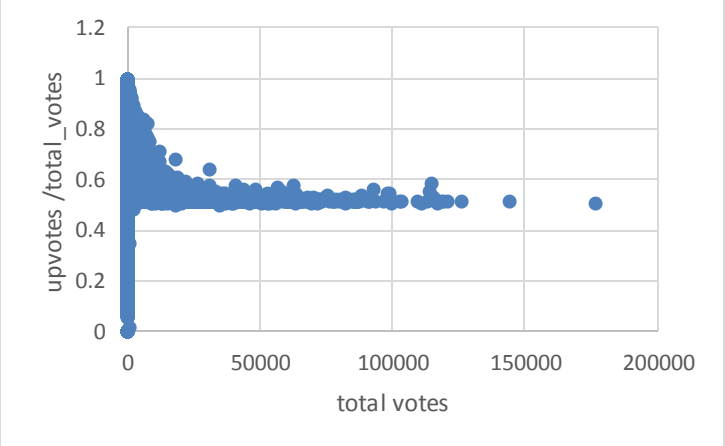

Fig 1. Ratio of upvotes to total votes vs total number of votes

Fig 1, provides insight about the trend around how the percentage of upvotes change, with the increase in number of total votes. We figure out different trends for submissions with total votes.

The ranges of interest are:

- 0 – 2700 (94,686 submissions)
- 2700 – 10000 (8,353 submissions)
- 10,000 and beyond (6,921 submissions)

B. Trends over hour of the day:

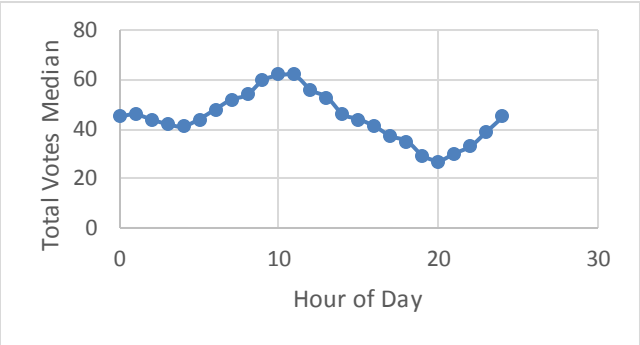

Fig 2. Median of total votes vs Hour of the day

Fig 2, describes how the number of total votes are affected by the hour of the day when a submission was submitted. We can see, a clear trend that submissions submitted between 8 to 12 UTC generally receives more number of total votes as compared to other submissions.

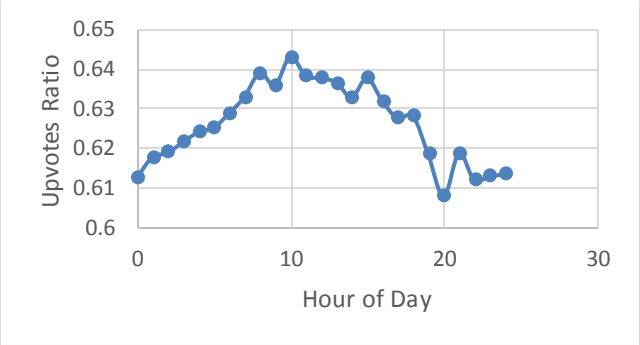

Fig 3. Upvotes Ratio vs Hour of the day

A similar trend as described in Fig 2, can be observed in Fig 3 too. The number of upvotes increase more than the number of downvotes during 8 to 13:00 UTC, hence we see a peak over the ratio of upvotes to total votes during these hours.

C. *Relationship between upvotes and downvotes*

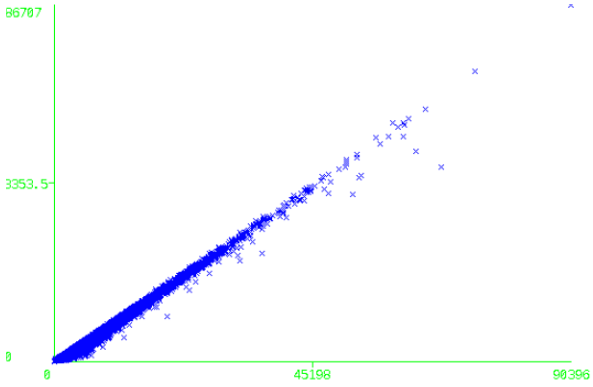

Fig 4. X - axis: upvotes, Y - axis: downvotes

Fig 4, describes the relationship between upvotes and downvotes on a submission. We notice that the slope of the graph is 0.78, hence a submission on an average receives 0.78 times downvotes over the number of upvotes it receives.

Another interesting parameter is score of a submission, which is defined as the difference between the number of upvotes and downvotes received by a submission.

The graph below depicts the relationship between the score of submission and the number of total votes received by the submission.

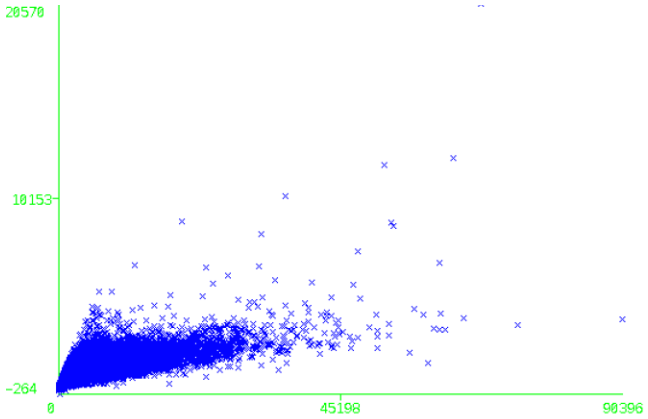

Fig 5. X axis: Total Number of Votes, Y axis: Score

We can observe from the above graph that the score saturates over the number of total votes, and is not affected much after a particular threshold.

D. *Relationship between number of comments and number of total votes*

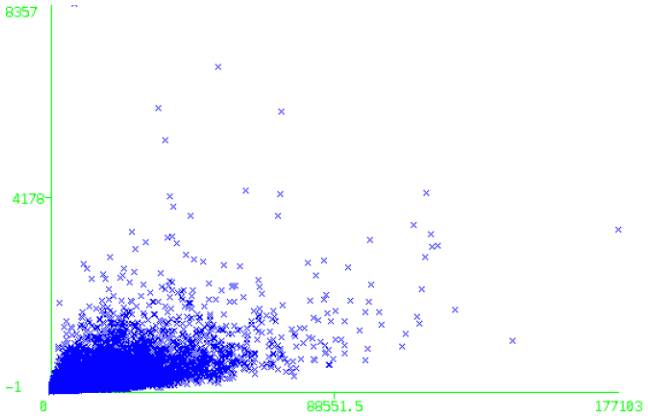

Fig 6: X axis: Total Number of Votes, Y axis: Number of Comments

Fig 6, describes how the number of total votes are affected by the number of comments. Here also we see that beyond a certain threshold, total numbers of votes on a submission are not affected by the number of comments on that submission.

E. Relationship between word count and ratio of the upvotes to total numbers of votes

Fig 7 below describes how number of words affects the ratio of upvotes to total votes associated with a submission. We observe that the proportion of upvotes can be categorized into less than 40 words (category 1), 40 to 52 words (category 2) and above 52 words (category 3). The ratio of upvotes to total votes is constant for category 1, increase for category 2 and then decreases for category 3.

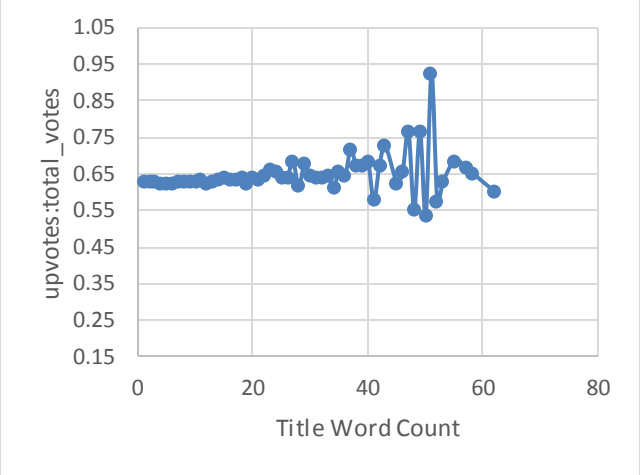

Fig 7. Upvotes to Total Votes ratio vs Title Word Count

F. Relationship between subreddits and ratio of upvotes to total number of votes

In order to understand the effect of subreddit of a submission on the ratio of upvotes to total number of votes on that submission, we find the average ratio of upvotes to total number of votes with in each subreddit.

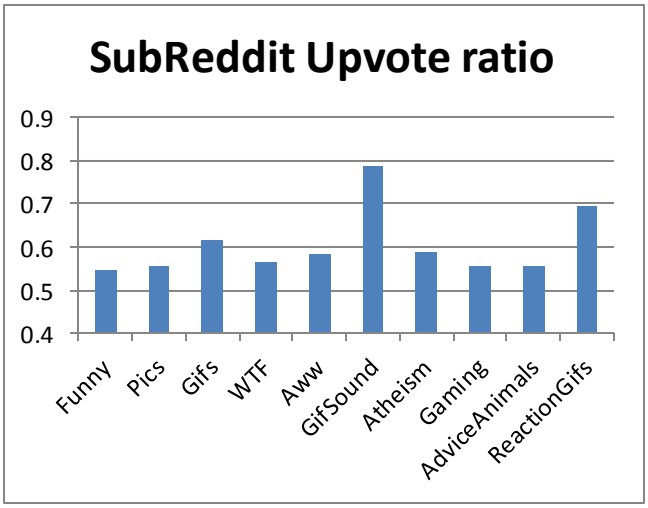

Fig 8. Ratio of upvotes to total number of upvotes vs subreddit

From Fig we can infer that the type of subreddit has a considerable affect upon the ratio of upvotes to total votes of a submission. We can observe that ReactionGifs and SoundGifs subreddits have considerable higher proportion of upvotes per submission. We consider the top 10 subreddits for the analysis.

G. Relationship between number of resubmissions with upvote to total vote ratio

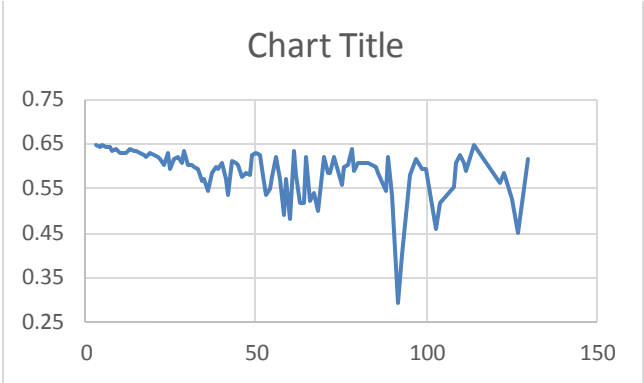

Fig 9. Number of image resubmissions vs upvote to total vote ratio

From the graph above, it can be observed that upvote to total vote ratio for a submission decreases with the number of resubmissions. It is an interesting observation that how the number of resubmission affect the upvote to total vote ratio.

3. PREDICTIVE TASK

The predictive task we will consider in this paper is to predict the upvotes count of a reddit submission, which is a regression task. Fitting the upvotes directly may not make sense, since its scale depends on the total number of votes received. Therefore we will first fit the ratio of upvotes to total number of votes. The ratio is then multiplied by total number of votes to get the predicted number of upvotes. For this task we will use mean absolute error (MAE) for evaluation. For baseline comparisons we use two naive regressors. First one always predicts the mean upvote ratio of the training set and the second one always predicts the mean upvote ratio of corresponding user which made the submission.

4. RELATED LITERATURE

The dataset being used is the Reddit's dataset. It came from Lakkaraju, McAuley, and Leskovec [2]. This data set was used for understanding the interplay between titles, content, and communities in social media. The paper mentions using upvotes as a measure of positive feedback.

Popularity prediction on social media is an interesting problem. Though various studies have been done to predict the popularity of content on social media, none of them mentions the prediction of number of upvotes or likes of a posted content. As far as we know, this particular data set has not yet been used to predict upvote count of a submission.

5. FEATURES USED

As per the exploratory analysis performed above, we finally use following features for the prediction task –

5.1 Time of the day

As per the exploratory analysis of variation of total votes and upvote ratio with time in fig 2, and fig 3, it can be seen that the upvote ratio has high correlation with the hour of the day. So the hour of the day is used as a categorical feature, where the hours are divided in 8 categories of window size of 3 each.

So for example if the post time for a certain post is 4 am, then it would have a feature vector of

[0 1 0 0 0 0 0 0]

↘

Window of 4-6

5.2 Word Count in Title

As per the exploratory analysis of variation of upvote ratio with the word count as per figure 7, it can be seen that the word count has high correlation with the upvote ratio when the word count are higher than a certain threshold. For ex. for the word count in the initial range 0 to 40 there is almost no effect on the upvote ratio. For word counts with length 40 to 52 it has positive effect on upvote ratio, i.e. slightly longer titles seems to attract users to provide more upvotes to the submissions, whereas if the word counts are greater than 52, i.e. for very large titles, users don't generally like the post and most of the votes are downvotes.

5.3 Total Votes

As per figure 1, total votes have high correlation with the upvote to total vote ratio of the post. This analysis is important to the overall prediction results of our model. As per analysis, there are three main categories for total votes i.e.

- 0 – 2700 (94,686 submissions)
- 2700 – 10000 (8,353 submissions)
- 10,000 and beyond (6,921 submissions)

We use the above analysis to improve the results by using the combination of binning and gradient boosting regression. We, train three different gradient boosting regression models for the above ranges, which help us to get a 23.79% improvement over the prediction made by single gradient boosting model. The comparison chart for the different models is shown in figure 11.

5.4 Average Category Upvote Ratio

As per fig 8, where we have shown the variation of upvote to total vote ratio with the top 10 popular subreddits. It can be seen that some subreddits seems to have higher upvote to total vote ratio as compared to the other. We utilized this analysis and have taken average subreddit upvote rate as a feature too which improves the performance of our final model by around 3.254%.

5.5 Number of resubmissions

As per fig 9, it is an interesting observation that how the number of image resubmission affect the upvote to total ratio. Hence it was taken as a feature in our final model. Similarly, the number of user resubmission is also related to the predictive task at hand. Hence, both the features are taken in our final model, which improves the performance by around 4.56%.

5.6 Avg Upvote Ratio of User and Image

The average upvote to total vote ratio of a user and image defines the number of upvotes a user gives on an average or the number of upvotes an image gets on an average. For the aforementioned predictive task, where we are predicting the upvote ratio for a submission, it makes intuitive sense to use these features which indeed provides better performance by improving MAE by around 8.11%.

6. Model Used

The model used is in the following format with relative importance as mentioned below:

Feature	Relative Importance
Total votes	0.25295631
User Avg. Upvote Ratio	0.18887366
Image Avg. Upvote Ratio	0.14539110
Subreddit Rate	0.04702012
Word count in the title	0.06739718
Number of user submissions	0.13912918
Number of Item submissions	0.0771705
Time	0.08206195

The time mentioned here is a feature vector in itself of length 8 as mentioned in section 5.1. The importance mentioned is the average importance of each feature over the 3 models used for total vote ranges as mentioned in section 5.3. For the aforementioned predictive task, training set of 100,000 examples was taken, with validation set being 15,000 and test set being 17307.

A) Models Tried

1) Baseline 1: As part of baseline 1, average upvote to total vote ratio is taken for any user and any image.

2) Baseline 2: As part of baseline2, average upvote to total vote ratio of a user is predicted for the data if the user is known else average upvote ratio is predicted.

3) Linear Regression: Linear Ridge library in python is used to predict upvote ratio using the linear regression model using the aforementioned features. We used the polynomial form of many features such as total votes to train the regressor. For ex. to use the n^3 formula for total votes, we multiplied the total votes by itself 3 times. We also normalized the data before using it for training and regularized the data using regularization parameter of 3.6.

4) Support Vector Regression(SVR): SVR makes use of a non-probabilistic linear regressor to predict data[4]. It is a modification of SVM where loss function is modified to include a distance measure. We used C i.e. regularization parameter as 0.5 with cache size = 2000 to tune the model.

5) Gradient Boosting Regressor(GB): GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fitted on

the negative gradient of the given loss function[5]. We use sklearn gradient boosting library for our predictive task.

6) Binned Gradient Boosting Regressor(GB): Our final model is to GB with binning of total votes wherein we divide total votes into 3 ranges as mentioned in section 5.3) and train 3 different GB regressor for dataset falling in each of these ranges.

B) Hyper tuning of Binned Gradient Boosting Regressor

Following parameters were tuned through validation set for binned GB regressor –

1) Number of Estimators: This is an indicator of number of boosting stages to perform. We hyper tuned this parameter for best results. The best results were achieved with this parameter being equal to 300.

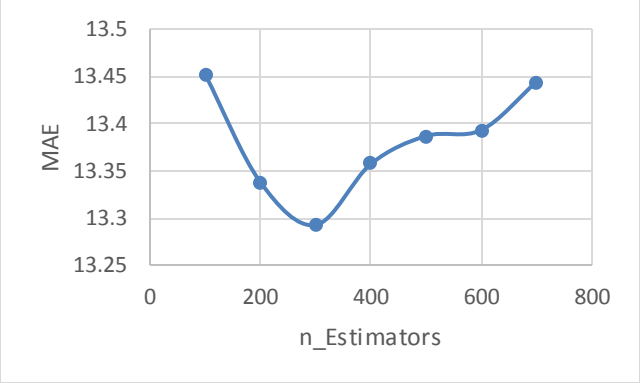

Fig 9. Variation of MAE with number of estimators

2) Max Depth: It is a measure of maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree. We hyper tuned this parameter to get best results on max depth of 6.

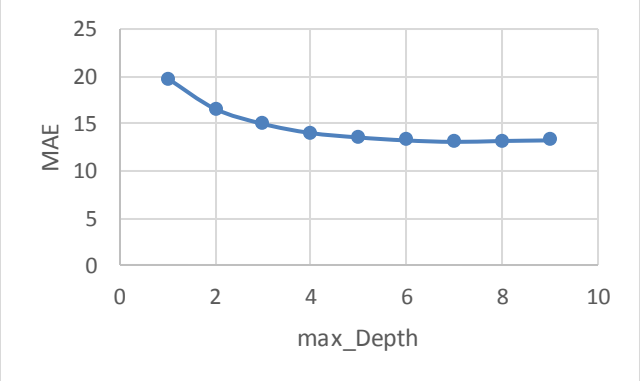

Fig 10. Variation of MAE with max depth

7. RESULTS AND CONCLUSION

Here's the comparison histogram for MAE of various models tried. As can be seen, we get the best results with the binned gradient boosting regression model.

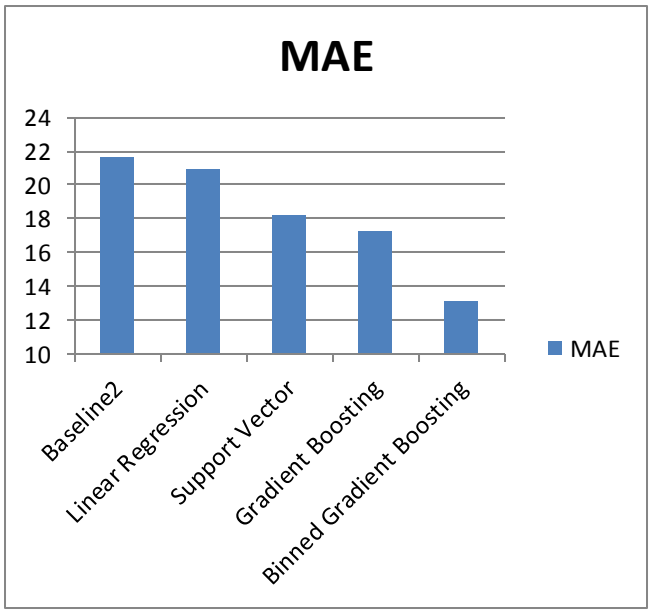

Fig 11. Comparison of various predictive models

The brief discussion below describes the results of all the models:

1) Baseline 1: We get an MAE of 71.1125525114 on test set. This is because this model simply predicts the same outcome for each dataset i.e. the average upvote to total vote ratio.

2) Baseline 2: We get an MAE of 21.6263 on test set. This is certainly a better model than baseline1 but still faces issue of being rigid. As it always predicts the same upvote to total vote ratio for each user.

3) Linear Regression: We get an MAE of 20.92 on test set using this model. With this model, we faced the inherent issues with all linear boundary models i.e. they try to formulate a linear relationship between features and predictive task, which is generally not the case.

4) Support Vector Regression(SVR): This model performs slightly better than the other model. We got an MAE of 18.22 using this model.

5) Gradient Boosting Regressor(GB): We got an MAE of 17.27 using this model.

6) Binned Gradient Boosting Regressor(GB): We finally got an MAE of 13.16 using this model.

As can be seen from fig. 11, the binned gradient boosting regressor provides the best results for our predictive task.

8. ACKNOWLEDGEMENT

Our thanks to Prof. Julian McAuley and all the TAs of CSE 255 for their guidance and suggestions.

9. REFERENCES

.

[1] Reddit. https://www.reddit.com/about (accessed Nov 2015)

[2] H. Lakkaraju, J. McAuley, and J. Leskovec. Whats in a name? understanding the interplay between titles,

[3] http://snap.stanford.edu/data/web-Reddit.htmlMedia, 2013.

[4] Basak, Debasish, Srimanta Pal, and Dipak Chandra Patranabis. "Support vector regression." *Neural Information Processing-Letters and Reviews* 11.10 (2007): 203-224.

[5] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html