Influence

DJ Rose, Sam Kavkewitz and John Duggan

Introduction

Our project was to discover and predict the influence of users on project management sites. Specifically, we decided to focus on GitHub and Bitbucket users; however, we decided to focus on Bitbucket due to time restrictions. To further study the influence of users, we desired to find a way to separate influence into two categories: technical influence and popular influence. Then, for each user, we wanted to calculate and study their technical and popular influence on Bitbucket. Finally, we wished to build a predictive model for both of these influences. This model, when given some basic information about how a user acted on Bitbucket, would then predict how influential they were to the community. While we originally wished to do this prediction for both technical and popular influence, due to time restrictions we focused on only predicting users popular influence.

Data

The data for our project were watcher, commit, fork, and pull request information for each repository on Bitbucket. Additionally, we used the followers of each user on Bitbucket. This data had already been collected and stored in a MongoDB, which we then used to calculate our influence measures and predictors of influence. An interesting note about the data was that the commit log information was incomplete. This was due to the fact that during project 2 of the course, where all of these various data were collected, there wasn't enough time to finish collecting the commit logs for each repository due to the sheer number of them. Thus, we only had roughly 15 million commit logs with which to work. We dealt with this by only studying users who had contributed to the repositories for which we had commit logs.

Methodology

To study user influence we have done the following. First, we decided upon and calculated the influence of each user on Bitbucket. While doing this, we split influence into two separate measures for both technical and popular influence. Then, we calculated six predictors of influence for each user. Next, we built a scatterplot to visualize the measures and predictors with the hope of finding correlations. Finally, we tried several techniques to build a predictive model for the popular influence of users using the calculated predictors.

Influence Measures

We needed two measures of influence: one for popular and one for technical influence. To calculate the popular influence we chose the number of followers for each user. There are perhaps other ways to quantify the popular influence, but we felt that there is not a better measurement of one's popularity than the number of followers. For the technical influence, we chose to implement our own

measure of technical influence which we have called the watchScore. The watchScore for a user *u* can be calculated as follows:

$$watchScore = \sum_{r \in Repos} rWatchers * ruserCommits/rtotalCommits$$

As can be seen, a user's watchScore is calculated by iterating through each repository. For each repo, the number of watchers of that repo is multiplied by the number of commits the user made to that repo, and divided by the total number of commits that were made to that repo; the resulting score for the one repo is added to the users total score. Basically, this measure assigns a user the percentage of the number of watchers for each repo he or she has contributed to.

Predictors

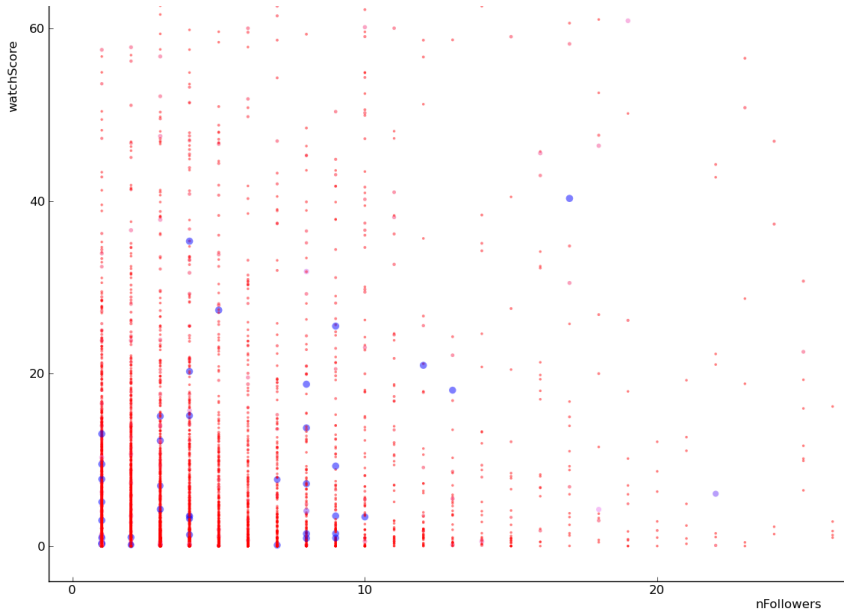
We chose to study six predictors that we thought would be good candidates to have correlation with the influence of a user. These six predictors were calculated for each user, and are listed below:

1. Number of commits
2. Frequency of commits
3. Total number of repos committed to
4. Time span of contribution (in days, the last day minus the first day they have made commits)
5. Number of forks made
6. Number of pull requests made

We felt that these predictors had a good chance of showing correlation to the selected influence measures.

Visualization

Before we built a model around these measures, we wanted to visualize both the measures and predictors to see if any correlations could be identified easily. This visualization needed to show three dimensions simultaneously. These three dimensions were the user's watchScore, number of followers, and a chosen predictor. To put these three dimensions on the same visualization, we created a scatterplot which placed the watchScore on the y-axis, the number of followers on the x-axis, and the predictor was shown by using it to control the color of each node in the plot. The color control was simple, a low value for the predictor colored the node red, and a high value for the predictor colored the node blue and made it slightly larger. This allows the person using the visualization to choose a predictor and see how it affects the measures. Below is an example of one of these scatterplots, where the predictor selected is the total number of repos a user has contributed to.

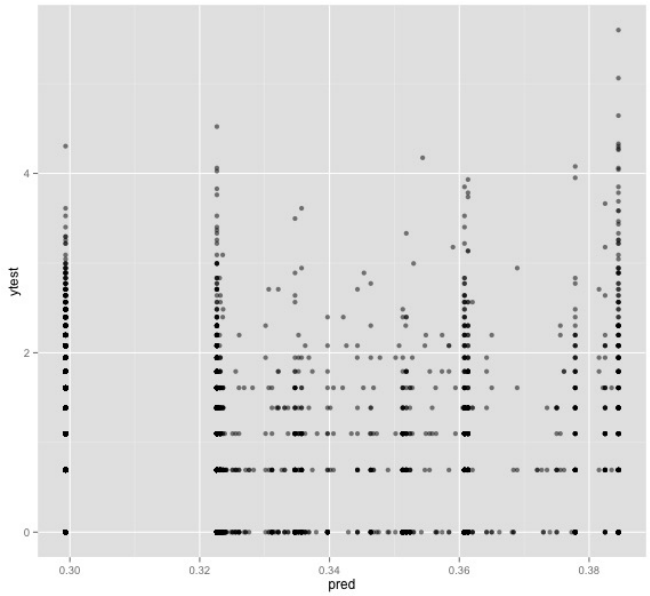


The full size image for this is available here. There are two interesting notes about this. First, there doesn't seem to be any pattern between the number of followers and the watchScore, which indicates that these are not correlated. Second, the colors of the nodes are seemingly random as well. This indicates that the predictor had no correlation between either measure of influence. The images for every predictor can be found on the link from above, but our observations were that there were no easily identifiable correlations between any of the predictors and measures.

Gradient Boosting Machine

We chose to use a gradient boosting machine to attempt to predict the number of followers a user might obtain based on variables like how long they had been contributing, how many repos they had, how many pull requests their repos had, and others.

After training the model, we tested it and compared our predicted number of follower to the actual number of followers for each user in the test set. If our model has predictive power, we would expect to see a strong correlation between the predicted and actual number of followers. However, as shown in the graph below, the data don't show any significant correlation. This means that our gradient boosting machine has little predictive power over followers based on the factors we used.



Further Analysis

Due to the poor performance of the Gradient Boosting Machine further analysis was done to attempt to gain insight from the data. This involved removing missing values and then doing some non-parametric test on the data. After all missing values were removed the number of observations dropped from 207202 to 1363 observations.

Principle component analysis was done on the data and number of commits appeared to be the only variable that explained variation in the data, even after scaling the data. After further observation, it was found that number of commits had an outlier that was profoundly larger than any other outlier that occurred in the other variables.

```
   nFollowers        tqCommits          nCommits            nForks          nPulls         timeContrb
Min.   :0.0000   Min.   :   0.0019   Min.   :     1.0   Min.   :  1.000   Min.   : 1.000   Min.   :   0.0
1st Qu.:0.0000   1st Qu.:   0.1145   1st Qu.:     6.0   1st Qu.:  1.000   1st Qu.: 2.000   1st Qu.:   3.5
Median :0.0000   Median :   0.6000   Median :    33.0   Median :  2.000   Median : 2.000   Median : 296.0
Mean   :0.4315   Mean   :   7.4208   Mean   :  3643.4   Mean   :  4.497   Mean   : 3.159   Mean   : 614.2
3rd Qu.:0.6931   3rd Qu.:   2.0000   3rd Qu.:   376.5   3rd Qu.:  4.000   3rd Qu.: 3.000   3rd Qu.:1031.5
Max.   :5.7869   Max.   :1084.0000   Max.   :638372.0   Max.   :735.000   Max.   :30.000   Max.   :7274.0
  totalRepos
```

To alleviate this extreme skewness, all observations with more than 2000 commits were removed from the data. Principle components were done again and the results were considerably different. After the extreme outliers were removed, time contributed was found to be the most influential variable and number of commits was still a significant part of the first component.

A scatter plot was made to identify if any variables had a strong correlation with any of the number of followers. Due to extreme skewness very few variables appeared to have any correlation with the

number of followers. However, there did appear to be a visible positive correlation between time contributed to and number of followers.

Conclusions

Our work has lead us to make two conclusions regarding the influence of users on Bitbucket. First, there is no correlation between the number of followers of a user and the number of watchers on the repos that user contributes to. Second, the number of followers a user has cannot be predicted with the methods we attempted. While we cannot claim that there exists no way to do this prediction, it is our opinion that it would require a nontrivial technique to do this prediction. An idea for future attempts to do this prediction would be to attempt to predict whether or not a user has at least one follower. Then, if a user is predicted to have a follower, try to predict how many they have. The reason this might have success is that a large number of users have zero followers, likely because the account was created but never seriously used. This segmentation has the potential to better predict the number of followers that a user has, but we have not been able to attempt such a technique. However, we were still able to learn much about influence of users in the Bitbucket community and would be interested in seeing future work done in this area.