Social Media Analytics Assignment 1

MSBA Spring 2018

Ryan Conklin, Reed Dalton, Gihani Dissanayake, Ali Prasla, Jake Schmidt

Part 1

Feature Engineering

Given our original feature set of tweeter A and tweeter B characteristics, we combined each of these characteristics into "difference" features. This means that Follower Count for A and Follower Count for B were combined into "dif_follower_count" calculated by A – B. This was completed for all features.

Best Model

After a few iterations, our best model was the Random Forest Model. This resulted in an accuracy of 74.7%. It was surprising to us that this non-parametric on a relatively small data set performed so well.

Best Predictors

All of the models we used (Random Forest, LASSO Logistic Regression, and Support Vector Machines) unanimously returned the most important predictor: `diff_listed_count`. This predictor can be interpreted as the difference between Twitter User A and Twitter User B's number of apperances in public lists. This is reasonable because if User A is in a public list, it implies that some user took the time to classify User A's content as worth while. Both SVM and Random Forest models also pointed to `diff_retweets_recieved` as being important. This is also reasonable for obvious reasons. To see the relative feature importances, you can look at Exhibits A,B and C in the appendix.

Surprises Business

What surprised us was that `diff_follower_count` was relatively unimportant for all the models. This is counter intuitive because, to the lay person, popularity is measured by followers. However, followers do not necessarily correlate to influence. We suspect this is because even if a user has a high follower count, they may have a low follower engagement. An influencer necessarily needs to have an engaged and active follower community.

Business Uses

The main value of this model is from a cost control angle. By focusing on key distributed influencers rather than high profile celebrities (those with high followers), a company can have a more targeted marketing strategy. This would allow for a greater penetration into certain social networks niches rather than a one size fits all approach.

Financial Value of Model

We can calculate the financial value of the model by finding the expected cost per conversion.

The cost per conversion is calculated by the following formula:
Expenditure per unit (in this case Tweeters) / probability of conversion

Without the model:
Expenditure per unit = $10 (for both tweets)
Probability of Conversion = .05 (probability that if an influencer tweets, a conversion will happen)

Cost per Conversion = $200

With the model:
Expenditure per unit = $10 (predicted influencer tweets twice)
Probability of Conversion = .741 (probability that you select influencer) * .075 (probability of conversion with two tweets)

Cost per Conversion = $178.50

Using those assumptions, this model would reduce the cost of conversion by $21.50.

Part 2

Scraping Twitter

We chose to scrape tweets containing “tensorflow” and created a pickle file called tensorflow.p containing 5,000 rows. Using that, we were able to extract the user_name, target, and the category of the tweet: whether it was an original tweet, a mention, or a retweet. This information is in the tweets.csv file with 6,395 rows of data. This expansion reflects single tweets with multiple targets, for example, a retweet and two mentions.

Insight from Part 1

The analysis from part 1 yielded that follower_count, listed_count, mentions_count, and network_feature_1 were the most significant, in that order. Since we found our random forest model to be the most accurate, these feature importances were derived from there. However, these exact features are not the easiest to extract for several reasons. For example, network_feature_1 seems to be some output of a principle component analysis, the details of which are withheld. There is no way for us to compute this variable, and something like follower_count of the tweet targets cannot be extracted if that twitter user was not in the 5000 tweet sample size.

Scoring Influencers

With that in mind, our score equation = (.35)*listed_count + (.30)*sum(scaled degree, centrality, betweenness) + (.20)*mentions + (.15)*retweets. We gave listed_count the highest weight because it was found to be important in the first part of the assignment and it reflects the power of a target to spread throughout a network. Though it was determined to be more vital than mentions_count in part 1, we chose to give that a weight as well, though not too much because we knew part of this value is reflected in the listed_count. Since the sum of scaled degree, centrality, and betweenness reflect the value of a node in the network and are an aggregated set of variables calculated using networkx, we weighted that to be our second most important predictor. Finally, we thought of reteweets to be important, because that's often how word travels around a twitter network to nodes that are not directly connected to the initial target.

Validating Scores

Using intuition and our knowledge about the subject material, we expected TensorFlow to be the top influencer, since that was the word we scraped for, and we were not disappointed. With a score of 299, it was by far the most influential account. In second to fifth place was rstudio with a score of 92, java with 72, TheNextWeb with 64, and kdnuggets with 49. These are consistent with our domain knowledge that there are 3 general tiers to influence in this network: very influential, moderately influential, and not very influential. The vast majority fall into the last category, with their input on tensorflow not having much of an impact throughout the network.

Appendix:

Exhibit A:

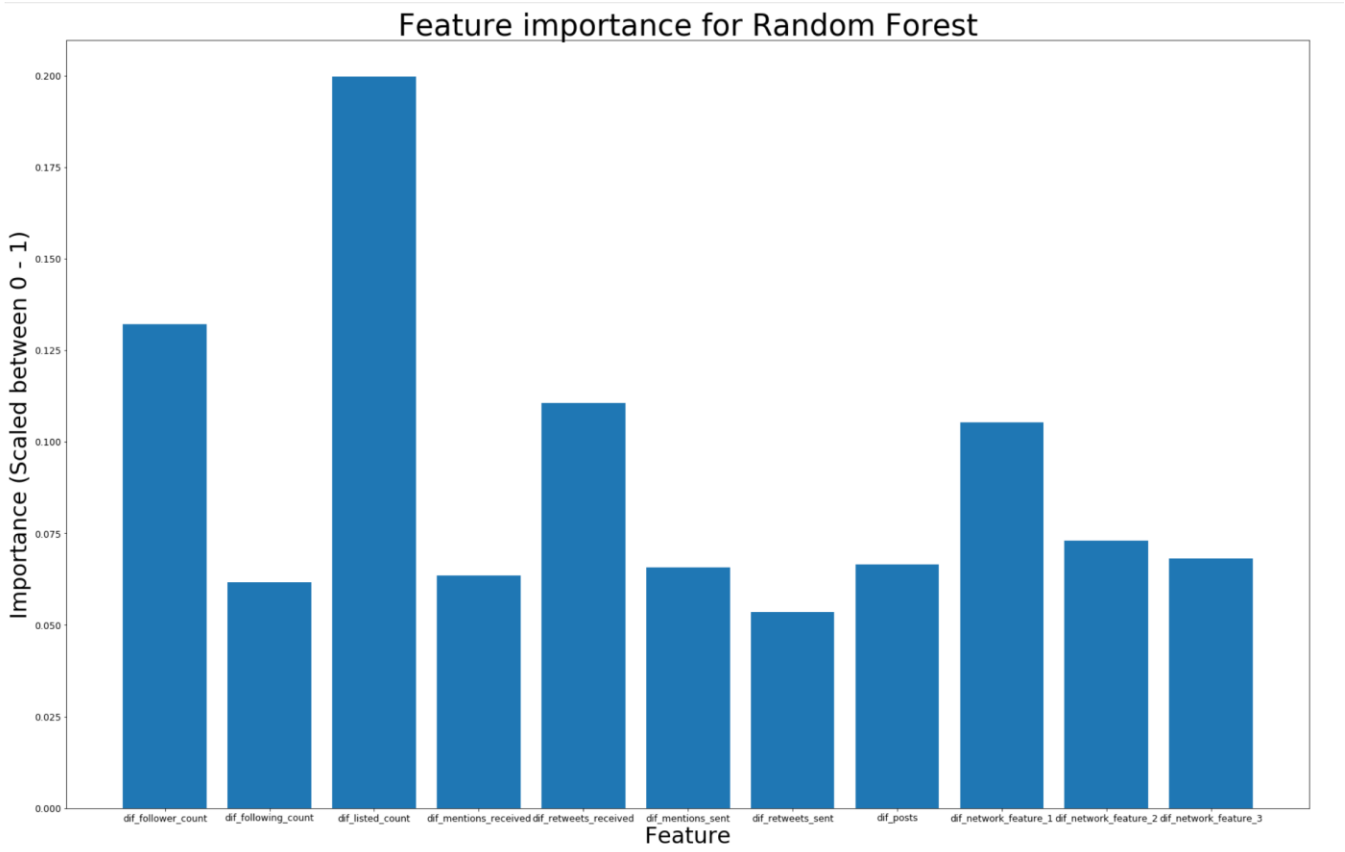


Exhibit B:

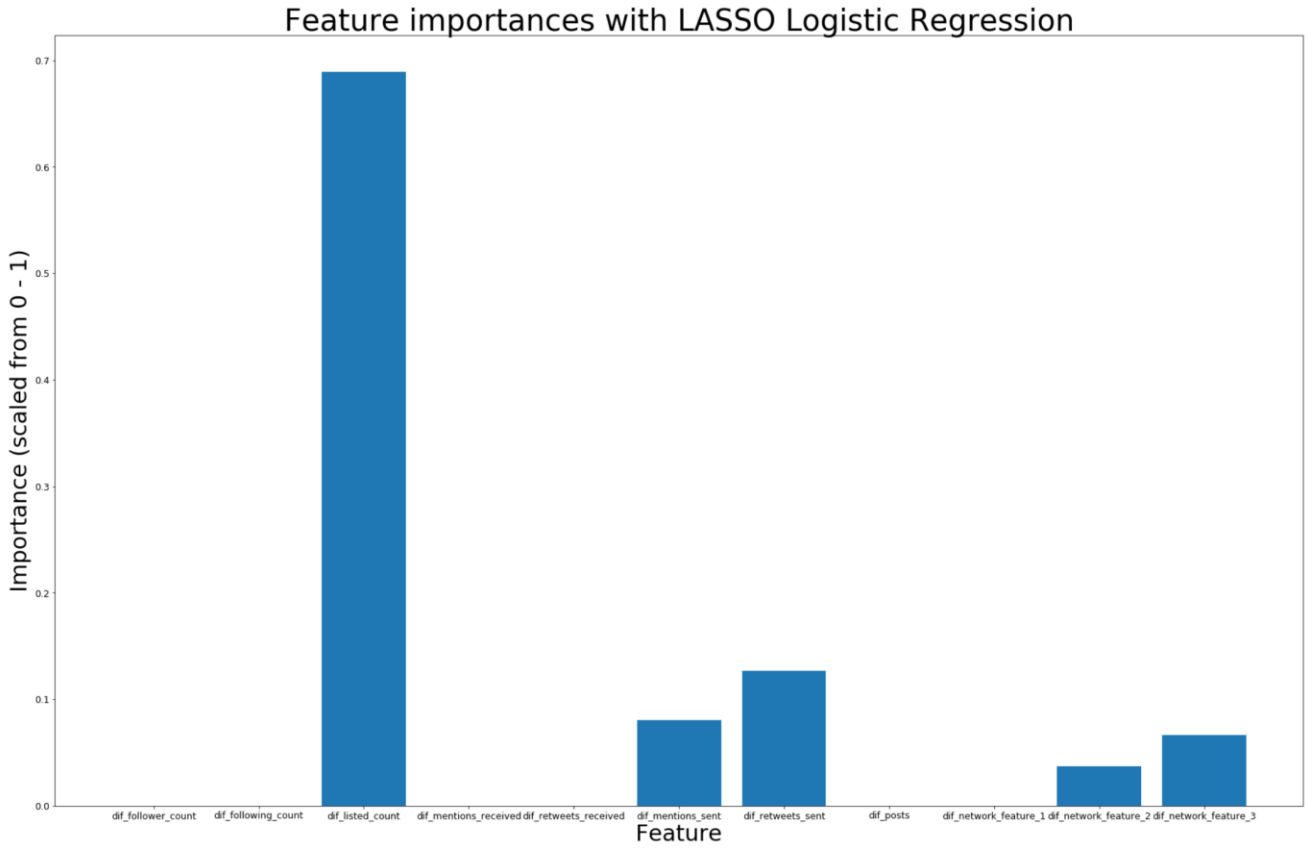
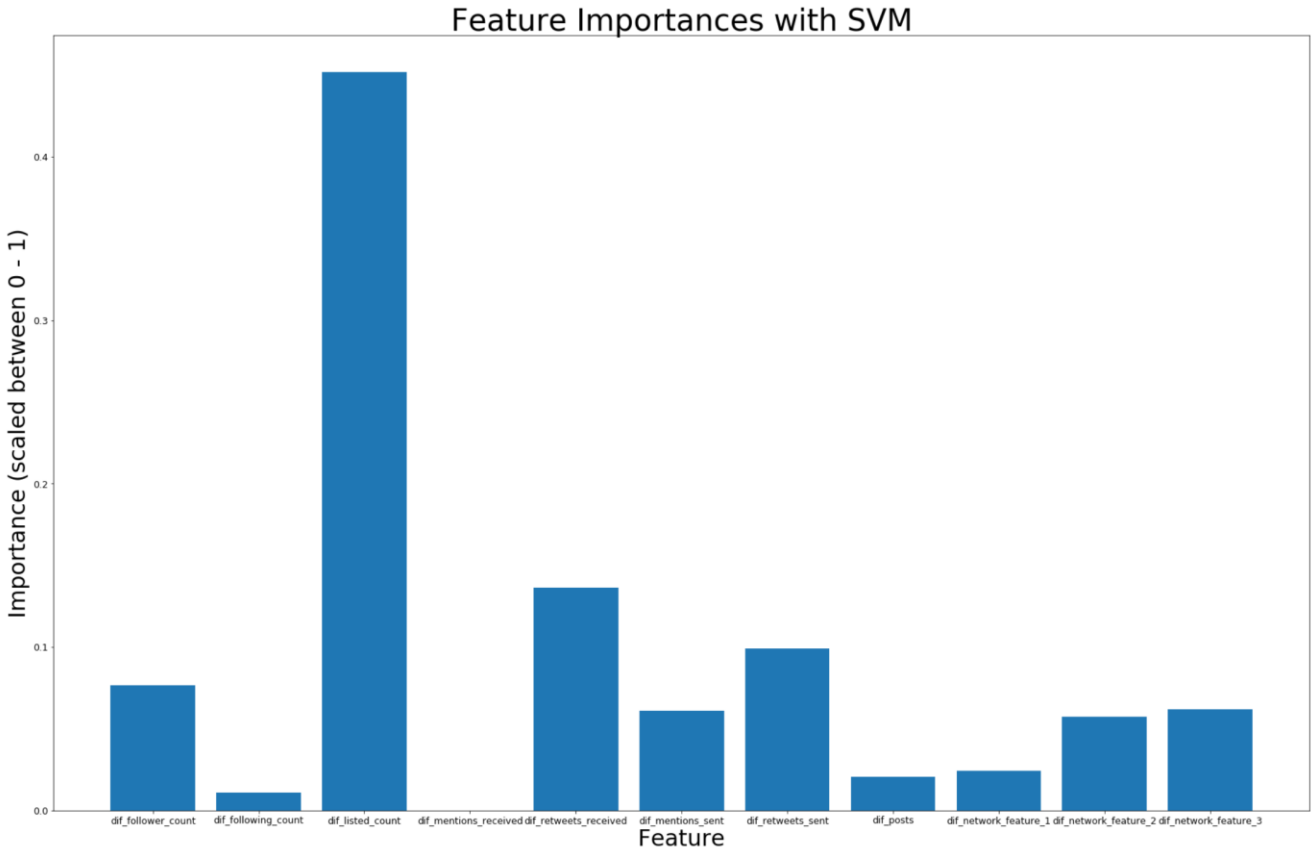

Feature importances with LASSO Logistic Regression
Importance (scaled from 0 - 1)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
dif_follower_count
dif_following_count
dif_listed_count
dif_mentions_received
dif_retweets_received
dif_mentions_sent
dif_retweets_sent
dif_posts
dif_network_feature_1
dif_network_feature_2
dif_network_feature_3
Feature

Exhibit C:


Feature Importances with SVM
Importance (scaled between 0 - 1)
0.0
0.1
0.2
0.3
0.4
dif_follower_count
dif_following_count
dif_listed_count
dif_mentions_received
dif_retweets_received
dif_mentions_sent
dif_retweets_sent
dif_posts
dif_network_feature_1
dif_network_feature_2
dif_network_feature_3
Feature