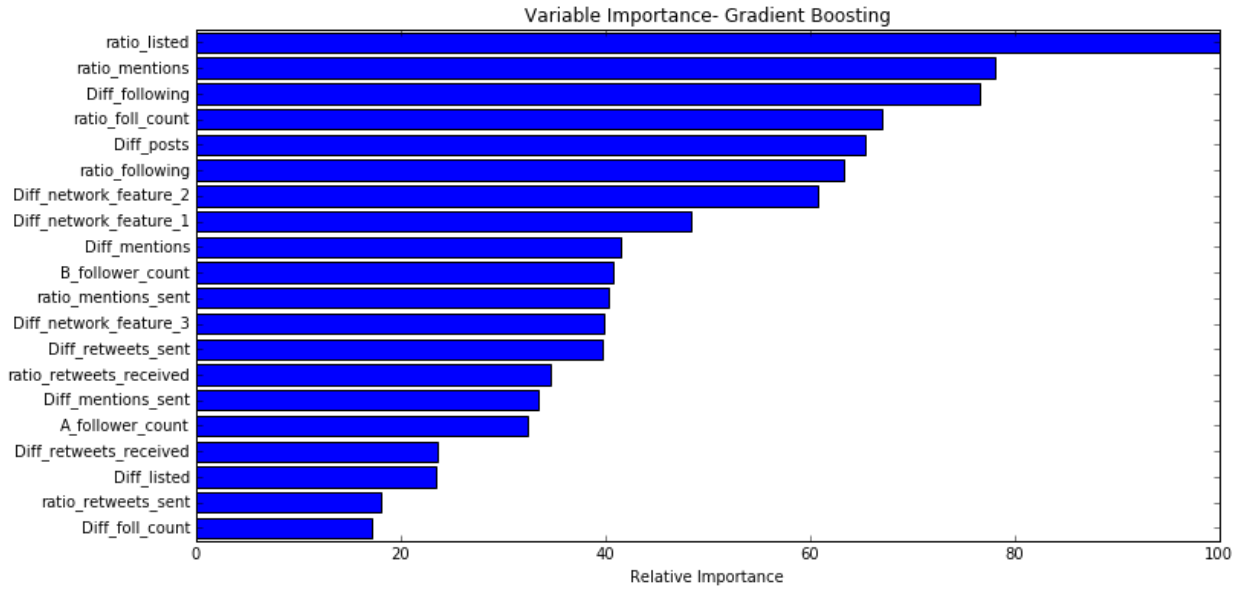
Social Media Analytics - Assignment 2

Vishwa Bhuta, Emily Graves, Ryan Maas, Caroline Nelson, and Jon Zeller

February 15, 2017

Predictors of Influence

Using the dataset from Kaggle, we trained a number of binary classification models to determine the best way to predict influence. Our best model was a gradient boosting classifier, and the confusion matrix as well as variable importance bar graph are shown below. The additional variables we decided to include were differences and quotients of the various figures - mentions, followers, etc. Some ratios were excluded where there were a large number of NAs (resulting from division by zero).



The three most important variables are the ratio of lists one user is on to the other user, the ratio of mentions, and the difference in following count. The first two represent how frequently the two users are mentioned and viewed by others, which captures the attention coming into them. The third measures how active these users are in terms of following other Twitter users, which is a reflection of how active these users are in sending attention to others. Both dimensions - attention in and out - are likely necessary to garner influence, rather than simply creating a lot of content. This shows that two-way relationships are important, as is also demonstrated by companies' social media accounts regularly responding

```
confusion_matrix(new_y_test, pred5)
#predicted columns, actual rows
#514 predicted negative (choice = 0 = not an influencer), actually negative
#188 predicted positive, actually negative
#145 predicted negative, actually positive
#528 predicted positive, actually positive, influencer
#0.1% chance sum(followers*50) from 1 to len(new_X_train) - 5*len(new_X_train)
#0.15% chance sum(followers*50) from 1 to 510 - 10*(510+171)
```

```
array([[514, 188],
       [145, 528]])
```

Financial Value of Our Model

In [17]:
```
# Using results from gradient boosting - highest accuracy score

# Add influencer predictions and actual influencers to the data frame to run profit analysis on
new_X_test['Predictions'] = pred5
new_X_test['Actual'] = new_y_test
```

In [18]:
```
# Calculate the number of A and B influencers
A_inf = new_X_test['Actual'].sum()
B_inf = len(new_X_test) - A_inf

# Calculate the number of followers the influencers have
B_inf_followers = new_X_test[new_X_test['Actual'] == 0]['B_follower_count'].sum()
A_inf_followers = new_X_test[new_X_test['Actual'] == 1]['A_follower_count'].sum()
inf_followers = B_inf_followers + A_inf_followers

total_followers = new_X_test['A_follower_count'].sum() + new_X_test['B_follower_count'].sum()

#Calculate the number of followers non-influencers have
non_inf_followers = total_followers - B_inf_followers - A_inf_followers

# The profit is the number of followers the influencers have times the probability they will respond times the profit, minus the c
before_profit = 0.001*inf_followers*50 - 5*len(new_X_test)*2
```

In [19]:
```
# Obtain the number of followers that will be influenced by the correctly predicted influencers
mask_1 = new_X_test['Predictions'] == new_X_test['Actual']
mask_2 = new_X_test['Predictions'] == 0
mask_3 = mask_1 & mask_2

B_true_followers = new_X_test[mask_3]['B_follower_count'].sum()

mask_4 = new_X_test['Predictions'] == 1
mask_5 = mask_1 & mask_4

A_true_followers = new_X_test[mask_5]['A_follower_count'].sum()

# The profit is the number of followers the correctly predicted influencers have times the probability times the profit, minus the
after_profit = 0.0015*(A_true_followers + B_true_followers)*50 - 10*len(new_X_test)
```

The above image shows calculations used to calculate profit figures (shown below). It is important to note that these calculations come from our test data, which represents approximately 25% of the total data available. Therefore, if we assume that our test data is representative of the raw data and the population, we can assume that the profit from the raw numbers would be approximately 4x higher and the lift values would be similar for the population.

In the first cell, we calculate predictions based on our best model from above. Then, we find the total number of followers of influencers across both user columns (A and B). Initial profit is calculated as the number of influencer followers multiplied by .001 (conversion rate) multiplied by $50 (revenue),

minus the cost of paying each user (A and B) $5. We can see below this number is around **$71M for the basic approach**.

Profit from the analytics-based approach requires calculating the "true followers" of both A and B users - the followers when that user is correctly identified as an influencer (since if they are not an influencer, or are not targeted, they do not reach their followers). Then, of these followers, conversion rate is .0015, with the same revenue and cost. This profit figure is around **$95M for the analytics-based model**.

The perfect model involves us making no mistakes in identifying the influencers, so we would be reaching the followers of every influencer with two tweets. Therefore we can simply multiply the same conversion rate, revenue, and cost figures by this larger number (total influencer followers). This figure is approximately **$106M for the perfect model.**

```
In [22]: # calculated above
         before_profit

Out[22]: 70880002.55

In [21]: # calculated above
         after_profit

Out[21]: 94929123.725

In [30]: ## Perfect model - every influencer is correct

         # same cost - $10 for each person
         cost = 10*len(new_X_test)

         # new revenue - total number of influencer followers is our population
         revenue = .0015*inf_followers*50
         perfect_profit = revenue - cost

         perfect_profit

Out[30]: 106326878.825
```

Lift from the **analytics-based model** is equal to ($94,929,123/$70,880,002) = 1.339, or about an **extra 33%**. The **perfect model** results in a lift of ($106,326,878/$70,880,002) = 1.500 or a **50% gain**.

Finding Influencers from Twitter

We extracted 5,000 tweets about Betsy DeVos, who was recently confirmed as the secretary of education and whose confirmation drew a lot of public attention. In creating a score for each tweet author, we looked at the important features we found in part 1 and matched them to their equivalents in our data; there were five total matches. Since we used a gradient boosting classifier, we

did not have coefficients for each feature, so we used their relative importance scores as a base. The weight for each feature was the weighted relative importance score: for example, the relative feature importance of the lists a user is on was 100 (it was the most important feature), so we divided that by the total possible "importance points" available in the five selected features, getting the weight 0.303.

```
score_coefs = pd.DataFrame(feature_importance[sorted_idx], feature_names,['Feature Importance'])

#taking only features present in twitter data
score_coefs = score_coefs.ix[['ratio_listed','Diff_posts','ratio_foll_count','ratio_following','ratio_retweets_received']]

#assigning weighted coefficient to each variable (adding up to 1)
score_coefs['Weighted Score'] = score_coefs['Feature Importance']/score_coefs['Feature Importance'].sum()
score_coefs
```

	Feature Importance	Weighted Score
ratio_listed	100.000000	0.302778
Diff_posts	63.371702	0.191876
ratio_foll_count	62.646595	0.189680
ratio_following	60.624530	0.183558
ratio_retweets_received	43.631759	0.132108

We next standardized all the relevant variables in the Twitter data so that they were 0 mean with a variance of 1, and applied the weights to them to get an aggregate score for each author. A positive score suggests that the author was above average on most/all the above factors, whereas a negative score suggests the opposite. Sorting the authors by their scores, we were able to identify the top 100 influencers, who were the most above average in all these categories. Below are the usernames and scores for the top 15 influencers, who are a mix of just regular people and people involved in the media industry:

username	score
Dysonpodcast	27.363748
richardwhitmir	18.846654
1cheezymonkey	11.743709
MasonTomoko	7.995315
AlishaRose226	7.381159
hxrleyquiinn	7.120803
patseey501	5.727174
MaidenSammaiden	5.590235
MOMMASOPH	5.388800
pjstevens100	5.364497
cat_next_door	4.487421
ConnorCollett	4.387421
KateMPorter	4.093289
BethEvs	4.046403
LindseyReinache	3.893486