Social Media Analytics: Homework 1

Date: Feb 6, 2018

Professor Anitesh Barua

Clarissa Franklin **Kyle Katzen** **Paige McKenzie**
Meyappan Subbaiah

Problem 1

Find Preditors of Influence:
Note to find accompanying python code for this problem, please navigate to the predictors jupyter notebook on the groups' github.

A From your model, which factors are best predictors of influence? (Provide screenshots). Are there any surprises here? How can a business use your model/results?

B If the model classifies an individual as a non-influencer, s/he is not selected/paid by the retailer to tweet. What is the lift in expected net profit from using your analytic model (versus not using analytics)?

C What is the lift in net profit from using a perfect analytic model (versus not using analytics)?

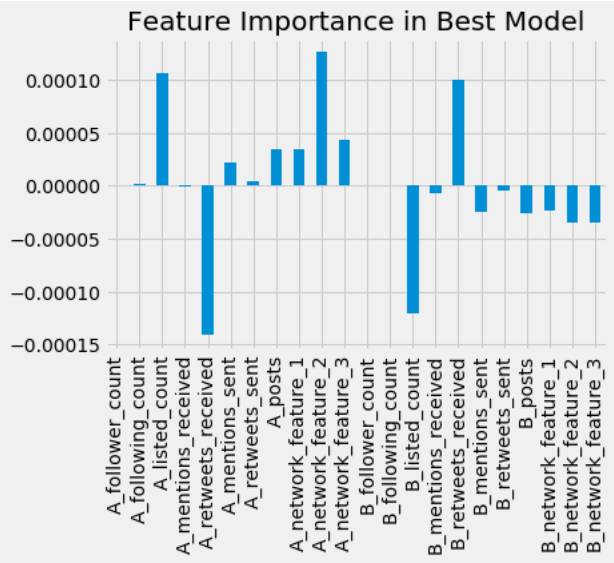
D Show all calculations.

Part A

We ran three primary models, as summarized below:

1. Untransformed variables: accuracy score=0.752
2. A-B variables: accuracy score =0.755
3. A/B variables: accuracy score=0.546

The A/B variable model did not perform well. The A-B model and untransformed model both performed well. We chose the model with the highest accuracy score, the untransformed model. The (normalized) feature importance values are shown in the graph below.



The most important features were A listed count, A retweets received, A network feature 1, B listed count and B listed count. Interestingly, the follower count was not an important feature, nor was the number of mentions received. As expected, listed count and retweets received were prominent features.

In an attempt to improve our model, we ran a subsequent model where the less important variables were omitted. In this model, our accuracy score increased slightly to 0.759. Since the performance metric improved, we selected this model as our final model.

Part B-D

Note: Code embeeded here to highlight calculations.

```
profit_comp = pd.DataFrame([y_test, prediction4, X_test[:,0], X_test[:,11]],
                           index=['true', 'pred', 'A_followers', 'B_followers']).T

profit_comp['correct'] = profit_comp['true']==profit_comp['pred']
#odds purchase * profit if purchase
profit_comp['Strategy_A'] = (profit_comp['true']*profit_comp['A_followers']*.0005*10) +
  (abs(profit_comp['true']-1)*profit_comp['B_followers']*.0005*10)-10


#prediction was incorrect
profit_comp.loc[~profit_comp['correct'], 'Strategy_B'] = -10


profit_comp.loc[(profit_comp['correct']) & (profit_comp['pred']==1.), 'Strategy_B'] =
  (profit_comp['A_followers']*.00075*10)-10
profit_comp.loc[(profit_comp['correct']) & (profit_comp['pred']==0.), 'Strategy_B'] =
  (profit_comp['B_followers']*.00075*10)-10

profit_comp.loc[(profit_comp['true']==1.), 'Strategy_C'] =
  (profit_comp['A_followers']*.00075*10)-10
profit_comp.loc[(profit_comp['true']==0.), 'Strategy_C'] =
  (profit_comp['B_followers']*.00075*10)-10

profit_comp.head(5)
```

	true	pred	A_followers	B_followers	correct	Strategy_A	Strategy_B	Strategy_C
0	0.0	0.0	267.0	102116.0	True	500.580	755.8700	755.8700
1	1.0	1.0	9679.0	209.0	True	38.395	62.5925	62.5925
2	1.0	1.0	404679.0	598.0	True	2013.395	3025.0925	3025.0925
3	0.0	0.0	10270.0	405315.0	True	2016.575	3029.8625	3029.8625
4	1.0	1.0	889300.0	1271.0	True	4436.500	6659.7500	6659.7500

Using this table, we will calculate the financial value of this model.

```
print "Profit of Current Strategy: ", '${:,.2f}'.format(sum(profit_comp['Strategy_A']))
print "Profit with prediction model: ", '${:,.2f}'.format(sum(profit_comp['Strategy_B']))
print "Profit with perfect prediction: ", '${:,.2f}'.format(sum(profit_comp['Strategy_C'])), '\n'

print "Lift of prediction: {}".format(sum(profit_comp['Strategy_B'])/
                                      sum(profit_comp['Strategy_A']))
```

```
print "Lift of perfect model: {}".format(sum(profit_comp['Strategy_C'])/
                                         sum(profit_comp['Strategy_A']))
```

Profit of Current Strategy: $8,512,737.90
Profit with prediction model: $11,898,965.42
Profit with perfect prediction: $12,777,356.84

Lift of prediction: 1.39778359933
Lift of perfect model: 1.50096913591

Please refer to python code and comments above for calculations. As shown, the profit of the current strategy (pay A and B each $5 to tweet once) is $8.513MM, while the profit for our prediction model (pick A OR B to pay $10 to tweet twice) is $11.899 million. A perfect prediction model would yield a profit of $12.777.

The lift of our prediction model is 1.398 and the lift of the perfect model is 1.501. Our model significantly improves expected profit.

Problem 2

Part II: Finding influencers from Twitter

1. Collect about 5000 tweets on any topic (e.g., politics, sports, current events, etc.). In addition to the tweet itself, the Twitter API provides a large quantity of information about the tweet as well as the author. Fetch all of this additional information along with the tweets.
2. Clean the tweets data and display in the format specified. This format highlights the original poster, the tagged individual, and whether the tweet was a retweet or an original tweet.
3. Calculate the degree, betweenness and closeness of each node in the above network.
4. Using the results from Part I, create a list of top 50 influencers from the tweets.

Part 1: Raw Twitter data
We selected the *Pyeongchang 2018 Winter Olympics* as our topic to retrieve tweets from Twitter.

```
## # A tibble: 6 x 4
##    screen_name
##          <chr>
## 1       soompi
## 2    joshrogin
## 3     SBNation
## 4 BishopIkedi1
## 5    RoDCelaya
## 6    AllOnFire
## # ... with 3 more variables: text <chr>, name <chr>, listed_count <int>
```

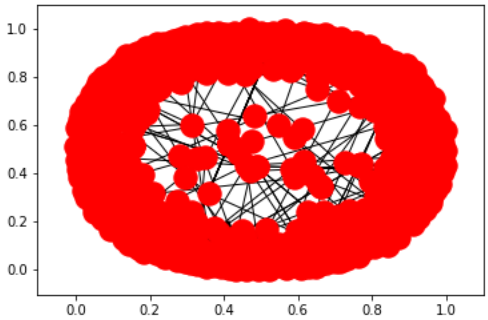
The tweets are quite long to look at, here is a sample of one tweet.

```
## [1] "The Fantastic Four! Nigerias Bobsled &amp;amp; Skeleton reps in South Korea for the Winter Olympics https://t.co/hfWvcfVF3
```

Part 2: Clean Twitter data

```
## # A tibble: 6 x 4
##   post_screen_name text_screen_name content clean_text_screen_name
##              <chr>            <chr>   <chr>                  <chr>
## 1           soompi           soompi   Tweet                 soompi
## 2        joshrogin        joshrogin   Tweet              joshrogin
## 3         SBNation         SBNation   Tweet               SBNation
## 4     BishopIkedi1     BishopIkedi1   Tweet           BishopIkedi1
## 5        RoDCelaya      @nzaccardi:      RT              nzaccardi
## 6        AllOnFire        AllOnFire   Tweet              AllOnFire
```

Part 3: Calculations
Prior to calculations a directed network graph was created, as seen below.



As you can see above, there are only a few people in the middle that connect otherwise wildy disconnected people. However, these are all terminal connections. As you'll see in the dataframe shortly, this will cause the betweeness of every node to be 0.

```
## # A tibble: 6 x 8
##          name betweenness   closeness degree retweets scaled_degree
##         <chr>       <dbl>       <dbl>  <int>    <dbl>         <dbl>
## 1 1037ChuckFM           0 0.000000000      2        0   0.002040816
## 2     1127AM_           0 0.001021450      1        0   0.001020408
## 3 11Yumitsu11           0 0.001021450      1        0   0.001020408
## 4      13OYSL           0 0.000000000      2        0   0.002040816
## 5 14sakura214           0 0.001021450      1        0   0.001020408
## 6  17mamabts4           0 0.002042901      2        0   0.002040816
## # ... with 2 more variables: listed_count <dbl>, score <dbl>
```

All network calculations and graph depictions can be found in the following network jupyter notebook, on the group's github.

Top 50 Influencers

We used the importance ranking from part 1 to guide our decision for how to weight the important variables. Since listed count and retweets were about .0001 and the mystery network feature was about .00014, we gave our variables similar proportions. Because we didn't know what was in the network feature, we just used both of our useful network features and gave them equal weight. We standardized listed count, closeness, retweets, and scaled degree so that they would all be in the same units, and then gave a score to each twitter user based on a linear combination of the weights and the standardized units.

```
## # A tibble: 980 x 2
##                          name      score
##                         <chr>      <dbl>
##  1                     nytimes 16.1568424
##  2                   minmeraki  9.3162696
##  3                     SBNation  4.9265906
##  4            pyeongchang2018  3.8928679
##  5                       NFLRT  3.2979258
```

```
##  6               KoreanUpdates  2.3765472
##  7                  barry2tone  2.0207240
##  8              ARMYIndonesiaa  1.9213391
##  9                 hewitt_riri  1.7534389
## 10               Fenella_Wicks  1.7530173
## 11                   baobobaek  1.5771925
## 12              NBCNightlyNews  1.4537750
## 13                     Essence  1.3680087
## 14             3axel_2toe_tano  1.3326346
## 15                    pamdon18  1.3326346
## 16                         PBS  1.1639157
## 17                    newsmax  1.1458859
## 18                       TIME  1.1264773
## 19                FabienMalbet  0.9973969
## 20                    CBGworld  0.9915881
## 21                  94_degrees  0.8844062
## 22                 weareoneEXO  0.8844062
## 23                 ncsulilwolf  0.8757092
## 24                    Olympics  0.8495422
## 25              Taecyeon_Today  0.7843072
## 26                   joshrogin  0.7601251
## 27              helenenothelen  0.7232450
## 28 httpstcodhTcvMsbNbMiguelnbc  0.6861849
## 29                  SachaAzcona  0.6490893
## 30                  JennVirskus  0.6481524
## 31                  SportByFort  0.6481524
## 32                  snocro4444  0.6467471
## 33                 Achilles_PR  0.6458570
## 34                 CathliaWard  0.6452012
## 35                    Sheasy64  0.6446390
## 36             JenniferDykes12  0.6445453
## 37                    NewsHour  0.6131299
## 38                     YouTube  0.5641610
## 39                Kantar_Media  0.5616537
## 40             EmbassyofRussia  0.5337900
## 41              rebeccafmiller  0.5237176
## 42                  Koreandogs  0.5079119
## 43                diodio0013tw  0.4879636
## 44                 Revolvermag  0.4330525
## 45             whiskynsunshine  0.4160870
## 46                 curtiszupke  0.4130087
## 47                     TeamUSA  0.4117661
## 48              HaraldDoornbos  0.4102390
## 49                   tasalinas  0.4029040
## 50                  nbc25fox66  0.3923142
## # ... with 930 more rows
```

Note: The required csvs are attached in this homework submission.