Connie Chang
4/19/15
CS 591

Homework 6
Final Project Topic:

For the final project topic, I decided to explore and look into two of my favorite words: free and pizza. Fortunately, I was able to find a dataset that contains just that through the Kaggle competition. The dataset includes 5671 requests collected from the Reddit community Random Acts of Pizza between December 8, 2010 and September 29, 2013. All requests ask for the same thing, which is a free pizza. With further analysis using some of the datamining tools shown in class, I'm interested in how, if any, of these attributes predict altruism through free pizza. Other attributes in the data set include the following:

Attributes/Description: Nature of the attributes

1) "giver_username_if_known": Reddit username of giver if known, i.e. the person satisfying the request ("N/A" otherwise).
2) "number_of_downvotes_of_request_at_retrieval": Number of downvotes at the time the request was collected.
3) "**number_of_upvotes_of_request_at_retrieval**": Number of upvotes at the time the request was collected.
4) "**post_was_edited**": Boolean indicating whether this post was edited (from Reddit).
5) "request_id": Identifier of the post on Reddit, e.g. "t3_w5491".
6) "**request_number_of_comments_at_retrieval**": Number of comments for the request at time of retrieval.
7) "request_text": Full text of the request.
8) "**request_text_edit_aware**": Edit aware version of "request_text". We use a set of rules to strip edited comments indicating the success of the request such as "EDIT: Thanks /u/foo, the pizza was delicous".
9) "**request_title**": Title of the request.
10) "**requester_account_age_in_days_at_request**": Account age of requester in days at time of request.
11) "requester_account_age_in_days_at_retrieval": Account age of requester in days at time of retrieval.
12) "**requester_days_since_first_post_on_raop_at_request**": Number of days between requesters first post on RAOP and this request (zero if requester has never posted before on RAOP).

13) "requester_days_since_first_post_on_raop_at_retrieval": Number of days between requesters first post on RAOP and time of retrieval.
14) "**requester_number_of_comments_at_request**": Total number of comments on Reddit by requester at time of request.
15) "requester_number_of_comments_at_retrieval": Total number of comments on Reddit by requester at time of retrieval.
16) "**requester_number_of_comments_in_raop_at_request**": Total number of comments in RAOP by requester at time of request.
17) "requester_number_of_comments_in_raop_at_retrieval": Total number of comments in RAOP by requester at time of retrieval.
18) "**requester_number_of_posts_at_request**": Total number of posts on Reddit by requester at time of request.
19) "requester_number_of_posts_at_retrieval": Total number of posts on Reddit by requester at time of retrieval.
20) "**requester_number_of_posts_on_raop_at_request**": Total number of posts in RAOP by requester at time of request.
21) "requester_number_of_posts_on_raop_at_retrieval": Total number of posts in RAOP by requester at time of retrieval.
22) "**requester_number_of_subreddits_at_request**": The number of subreddits in which the author had already posted in at the time of request.
23) "**requester_received_pizza**": Boolean indicating the success of the request, i.e., whether the requester received pizza.
24) "requester_subreddits_at_request": The list of subreddits in which the author had already posted in at the time of request.
25) "**requester_upvotes_minus_downvotes_at_request**": Difference of total upvotes and total downvotes of requester at time of request.
26) "requester_upvotes_minus_downvotes_at_retrieval": Difference of total upvotes and total downvotes of requester at time of retrieval.
27) "**requester_upvotes_plus_downvotes_at_request**": Sum of total upvotes and total downvotes of requester at time of request.
28) "requester_upvotes_plus_downvotes_at_retrieval": Sum of total upvotes and total downvotes of requester at time of retrieval.
29) "requester_user_flair": Users on RAOP receive badges (Reddit calls them flairs) which is a small picture next to their username. In our data set the user flair is either None (neither given nor received pizza, N=4282), "shroom" (received pizza, but not given, N=1306), or "PIF" (pizza given after having received, N=83).
30) "requester_username": Reddit username of requester.
31) "unix_timestamp_of_request": Unix timestamp of request (supposedly in timezone of user, but in most cases it is equal to the UTC timestamp -- which is incorrect since most RAOP users are from the USA).
32) "unix_timestamp_of_request_utc": Unit timestamp of request in UTC.

Properties/Basic Statistics:

All the columns selected in the file are numerical or Boolean values. Preprocessing the data is not completely necessary before using them in a model. The following gives the type for each attribute or column with a total of 4040 entries.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4040 entries, 0 to 4039
Data columns (total 14 columns):
post_was_edited                                       4040 non-null int64
request_id                                            4040 non-null object
requester_account_age_in_days_at_request              4040 non-null float64
requester_days_since_first_post_on_raop_at_request    4040 non-null float64
requester_number_of_comments_at_request               4040 non-null int64
requester_number_of_comments_in_raop_at_request       4040 non-null int64
requester_number_of_posts_at_request                  4040 non-null int64
requester_number_of_posts_on_raop_at_request          4040 non-null int64
requester_number_of_subreddits_at_request             4040 non-null int64
requester_received_pizza                              4040 non-null bool
requester_upvotes_minus_downvotes_at_request          4040 non-null int64
requester_upvotes_plus_downvotes_at_request           4040 non-null int64
request_title                                         4040 non-null object
request_text_edit_aware                               4040 non-null object
dtypes: bool(1), float64(2), int64(8), object(3)

Mean Values of each Column of Dataframe
post_was_edited                                       1.005868e+08
requester_account_age_in_days_at_request              2.545866e+02
requester_days_since_first_post_on_raop_at_request    1.641703e+01
requester_number_of_comments_at_request               1.150983e+02
requester_number_of_comments_in_raop_at_request       6.450495e-01
requester_number_of_posts_at_request                  2.160149e+01
requester_number_of_posts_on_raop_at_request          6.361386e-02
requester_number_of_subreddits_at_request             1.807673e+01
requester_received_pizza                              2.460396e-01
requester_upvotes_minus_downvotes_at_request          1.160080e+03
requester_upvotes_plus_downvotes_at_request           3.743236e+03
dtype: float64
```

Hypotheses:

1) Do any of the bolded attributes listed above affect the chances that a user, who posted on the Reddit website, will receive a free pizza? How highly are they correlated and how accurate can these attributes predict this outcome?
2) How significant do any specific words affect whether a free pizza is given in user's text request posts?