Assignment II – CS 6474/CS 4803 Social Computing*

Grade	Max 150 points; 15% of overall grade (late policy applies)
Due	Nov 13, 2019, 11:59pm Eastern Time
What to hand in	A report (as a PDF file) with answers to the different questions; reports should be no shorter than 3 pages or no longer than 8 pages in a single-spaced, single column format with at least 1 inch margin. All of the code as a zipped folder
Where to submit	Canvas

The goal of this option of the assignment is to develop different supervised learning models to identify success or failure of altruistic requests on social media. The questions derive from social computing research that aims to understand linguistic markers of altruism as described on social media [1]. The questions in the assignment will test your understanding of theoretical notions of language and help seeking (narratives, moral foundations) and to what extent they can provide insights into the social construct of altruistic requests.

Part 1: Please refer to the enclosed zipped folder that contains dataset and associated information[1]. The dataset, named the file pizza_request_dataset.json, contains a collection of 5671 textual requests for pizza from the Reddit community "Random Acts of Pizza"[2] (henceforth referred to as ROAP) together with their outcome (successful/unsuccessful) and meta-data. All requests ask for the same altruistic request: a free pizza, and span the timeframe December 8, 2010 to September 29, 2013. The outcome of each request – whether its author received a pizza (successful) or not (unsuccessful) – is known. In the questions below, the ground truth data for all of the classification models will be this outcome, specifically in the file pizza_request_dataset.json, the field `requester_received_pizza`. Please refer to Appendix I of this assignment document for an elaborate listing and description of all of the fields in the dataset file.

The features to be used in the classification models are described in the questions below. Please develop one classifier, specifically a Support Vector Machine model with a linear kernel and default parameters corresponding to each question below. For all of the classifiers, use a randomly sampled 10% of the dataset as test set (567 posts), and the remaining 90% as the training dataset (5104 posts) – the training and test sets need to be consistent across all classifiers below, i.e., the same 567 posts should be used for testing and the same 5104 for training for a), b), c) and d).

a) *Model 1 – n-grams (20 points)*: This model will extract the top 500 unigrams and top 500 bigrams[3] as features to classify posts that would be successful or those that will be unsuccessful in their pizza requests. Here "top" means most frequently occurring unigrams and bigrams in the posts belonging to the training set. Using these *n*-gram features, train and test an SVM classifier as described above. Report a table containing the accuracy of your classifier, precision, recall, F1, specificity, and AUC.

b) *Model 2 – Activity and Reputation (20 points)*: This model will utilize a variety of the activity and reputation data included in the dataset file (pizza_request_dataset.json) as features to distinguish between successful and unsuccessful requests. The specific activity features will use the values included in the following fields corresponding to each post:

```
post_was_edited
requester_account_age_in_days_at_request
requester_account_age_in_days_at_retrieval
```

[1] Downloaded from the SNAP Stanford website: http://snap.stanford.edu/data/web-RedditPizzaRequests.html

[2] https://www.reddit.com/r/Random_Acts_Of_Pizza/ Excerpt from the subreddit description: "Feel like giving a random redditor a free pizza, but don't know how or who? Well this is the right place for you! Random giving is why we are here!"

[3] Post content is given in the field "`request_text`" in the dataset file pizza_request_dataset.json.

```
requester_days_since_first_post_on_raop_at_request
requester_days_since_first_post_on_raop_at_retrieval
requester_number_of_comments_at_request
requester_number_of_comments_at_retrieval
requester_number_of_comments_in_raop_at_request
requester_number_of_comments_in_raop_at_retrieval
requester_number_of_posts_at_request
requester_number_of_posts_at_retrieval
requester_number_of_posts_on_raop_at_request
requester_number_of_posts_on_raop_at_retrieval
requester_number_of_subreddits_at_request
requester_subreddits_at_request
```

And the specific reputation features will use the values included in the following fields for each post:

```
number_of_downvotes_of_request_at_retrieval
number_of_upvotes_of_request_at_retrieval
requester_upvotes_minus_downvotes_at_request
requester_upvotes_minus_downvotes_at_retrieval
requester_upvotes_plus_downvotes_at_request
requester_upvotes_plus_downvotes_at_retrieval
requester_user_flair
```

Using these values for activity and reputation as features, train and test an SVM classifier as described above. Report a table containing the accuracy of your classifier, precision, recall, F1, specificity, and AUC.

c) *Model 3 – Narratives (30 points)*: This third model will extract features corresponding to the narrative dimensions identified in [1]. Refer to the enclosed files within "/resources/narratives". There are five narratives – *desire, family, job, money,* and *student*. Each narrative file has a set of words associated with it. To extract post features corresponding to a narrative, perform regular expression match between all words corresponding to the narrative and those corresponding to a post (in the training and test sets)[3]. The narrative features for a post will be the ratio of the number of matches for each narrative to the total number of white spaced words in the post. Using these five narrative features, train and test an SVM classifier as described above. Report a table containing the accuracy of your classifier, precision, recall, F1, specificity, and AUC.

d) *Model 4 – Moral foundations (30 points)*: This third model will use the dimensions of "moral foundations" as features for classifying successful and unsuccessful requests. These dimensions are based on the moral foundations theory[4] that seeks to understand why morality varies so much across cultures yet still shows so many similarities and recurrent themes. In brief, the theory proposes that several innate and universally available psychological systems are the foundations of "intuitive ethics." The dimensions of the moral foundations include: *care/harm, loyalty/betrayal, authority/subversion,* and *sanctity/degradation*. Their descriptions can be found in Appendix II. To extract features corresponding to the different dimensions, first refer to the enclosed file "MoralFoundations.dic" under "/resources" – the file opens with any simple plain text editor program. The dictionary contains terms indexed by integers, where the integers are mapped to the moral foundations dimensions. Then, for a given post in your training or test data[3], obtain one feature corresponding to each dimension, by matching (with regular expressions) each word in the dictionary for that dimension to each word in the post. This way, you will obtain a count variable of the occurrence of the dimension in the post. By dividing this count by the total number of white spaced words in the post, you will obtain a normalized feature value for the same dimension. Using these dimensions as features, train and test an SVM classifier as described above. Report a table containing the accuracy of your classifier, precision, recall, F1, specificity, and AUC.

Part 2: Present a discussion of the performance of the above four models:

a) *(4 points)* Which of the four classifiers performed the best; which one performed the worst?

b) *(6 points)* Describe your anticipated reasoning driving these differences in performance of the classifiers.

[4] http://moralfoundations.org/

c) *(10 points)* For models 3 and 4 in particular, describe their performance compared to models 1 and 2. Why do you think they perform better or worse than models 1 and 2? Between models 3 and 4, which one is better? What could be the reason behind this observation?

d) *(10 points)* Present your reasoning if your models indicate that language is able to predict success of altruistic requests – other than model 2, all of the other models rely on language.

Part 3: Presentation a comparative discussion of the performance of all of your classification models and the performance metrics (AUC) reported in Table 4 of [1]:

a) *(10 points)* In what ways are your models similar or different from those in Table 4 of [1]?

b) *(10 points)* Where and why do they perform better or worse compared to [1]?

Reference:

[1] Althoff, T., Danescu-Niculescu-Mizil, C., & Jurafsky, D. (2014). How to ask for a favor: A case study on the success of altruistic requests. In Proc. ICWSM 2014. Link: https://cs.stanford.edu/~althoff/raop-dataset/altruistic_requests_icwsm.pdf

Appendix I

Format of the file pizza_request_dataset.json:

Field	Description
giver_username_if_known	Reddit username of giver if known, i.e. the person satisfying the request ("N/A" otherwise).
in_test_set	Boolean indicating whether this request was part of our test set.
number_of_downvotes_of_request_at_retrieval	Number of downvotes at the time the request was collected.
number_of_upvotes_of_request_at_retrieval	Number of upvotes at the time the request was collected.
post_was_edited	Boolean indicating whether this post was edited (from Reddit).
request_id	Identifier of the post on Reddit, e.g. "t3_w5491".
request_number_of_comments_at_retrieval	Number of comments for the request at time of retrieval.
request_text	Full text of the request.
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".
request_title	Title of the request.
requester_account_age_in_days_at_request	Account age of requester in days at time of request.
requester_account_age_in_days_at_retrieval	Account age of requester in days at time of retrieval.
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).
requester_days_since_first_post_on_raop_at_retrieval	Number of days between requesters first post on RAOP and time of retrieval.
requester_number_of_comments_at_request	Total number of comments on Reddit by requester at time of request.
requester_number_of_comments_at_retrieval	Total number of comments on Reddit by requester at time of retrieval.
requester_number_of_comments_in_raop_at_request	Total number of comments in RAOP by requester at time of request.
requester_number_of_comments_in_raop_at_retrieval	Total number of comments in RAOP by requester at time of retrieval.
requester_number_of_posts_at_request	Total number of posts on Reddit by requester at time of request.
requester_number_of_posts_at_retrieval	Total number of posts on Reddit by requester at time of retrieval.
requester_number_of_posts_on_raop_at_request	Total number of posts in RAOP by requester at time of request.
requester_number_of_posts_on_raop_at_retrieval	Total number of posts in RAOP by requester at time of retrieval.
requester_number_of_subreddits_at_request	The number of subreddits in which the author

	had already posted in at the time of request.
requester_received_pizza	Boolean indicating the success of the request, i.e., whether the requester received pizza.
requester_subreddits_at_request	The list of subreddits in which the author had already posted in at the time of request.
requester_upvotes_minus_downvotes_at_request	Difference of total upvotes and total downvotes of requester at time of request.
requester_upvotes_minus_downvotes_at_retrieval	Difference of total upvotes and total downvotes of requester at time of retrieval.
requester_upvotes_plus_downvotes_at_request	Sum of total upvotes and total downvotes of requester at time of request.
requester_upvotes_plus_downvotes_at_retrieval	Sum of total upvotes and total downvotes of requester at time of retrieval.
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" (given after received, N=83).
requester_username	Reddit username of requester.
unix_timestamp_of_request	Unix timestamp of request (supposedly in timezone of user but in most cases equal to the UTC timestamp which is incorrect since most RAOP users are from the USA).
unix_timestamp_of_request_utc	Unit timestamp of request in UTC.

Appendix II

Descriptions of the different moral foundations dimensions:

Care/harm: This foundation is related to our long evolution as mammals with attachment systems and an ability to feel (and dislike) the pain of others. It underlies virtues of kindness, gentleness, and nurturance.
Fairness/cheating: This foundation is related to the evolutionary process of reciprocal altruism. It generates ideas of justice, rights, and autonomy.
Loyalty/betrayal: This foundation is related to our long history as tribal creatures able to form shifting coalitions. It underlies virtues of patriotism and self-sacrifice for the group. It is active anytime people feel that it's "one for all, and all for one."
Authority/subversion: This foundation was shaped by our long primate history of hierarchical social interactions. It underlies virtues of leadership and followership, including deference to legitimate authority and respect for traditions.
Sanctity/degradation: This foundation was shaped by the psychology of disgust and contamination. It underlies religious notions of striving to live in an elevated, less carnal, more noble way.