# ECS 171: Pizza Precog

Zach Bair, Andrew Brandes, Anthony Chen,
Taronish Daruwalla, Matthew Morikawa, Alex Rumbaugh,
Yosi Shturm, Blake Tacklind, and Shyam Venkataraman

December 14, 2014

**Abstract**

**Objective:** To predict whether a user posting on the Random Acts of Pizza (RAOP) subreddit is given pizza and to understand why members of the community choose to give pizza to some posters and not to others. Using data about the user's reddit account and RAOP post, we use PCA, KNN, ANN, Logistic Regression, Naïve Bayes, and Random Forest methods to complete our objective.

# 1 Introduction

The problem we attempt to solve addresses aspects of human nature such as why people choose to give things to others without the expectation of something in return. Extracting meaningful information from the user's post is critical to creating good classifiers. The dataset contains 5671 samples and the features are listed in Table 1. Some of the features are collected both at the request time and the retrieval time.

| Feature Name | Description |
|---|---|
| requester received pizza | Boolean indicating the success of the request. |
| giver username | The reddit username of the giver(if known) |
| downvotes of request at retrieval | Number of downvotes at the time the request was collected. |
| 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. |
| request comments at retrieval | Number of comments for the request at time of retrieval. |
| request text | Title of the request. |
| requester account age | Account age of requester in days. |
| days since first post | Number of days between requesters first post on RAOP. |
| requester number of posts | Total number of posts on Reddit by requester. |
| requester number of posts on RAOP | Total number of posts in RAOP by requester. |
| requester number of subreddits | The number of subreddits in which the author had already posted in. |
| requester upvotes minus downvotes | Difference of total upvotes and total downvotes of requester. |
| requester upvotes plus downvotes | Sum of total upvotes and total downvotes of requester. |
| requester user flair | Name of icon next to requester name. |
| requester username | Reddit username of requester |
| unix timestamp of request utc | Unit timestamp of request in UTC. |

Table 1: Dataset feature names and description

There are many reasons that a giver would give pizza to a poster, which makes separating the signal from the noise a difficult task. We want to determine what aspects of the post increase the chance that the poster will get their request fulfilled.

# 2 Background

## 2.1 What is Reddit?

Reddit is a social media website where users can post and comment in communities called subreddits dedicated to specialized topics. Users can upvote or downvote posts and comments. Posts with the most net upvotes will make their way to the front page of a subreddit. In the Random Acts of Pizza subreddit, users post requests for pizza and members of the subreddit can message the poster with an offer to send them a pizza. There are approximately 35,000 registered users on the RAOP subreddit.

## 2.2 Topic Classification

Our data, and much of our initial research came from a paper written by Tim Althoff, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky titled: *How to Ask for a Favor: A Case Study on the Success of Altruistic Requests*[5]. The authors supplied the data

set, along with a method to classify the text into different narratives. They used non-negative matrix factorization (NMF) and term frequency-inverse document frequency (TF-IDF) to break the data into sparse representations.

NMF is a method which takes a matrix of weighted word frequency data, and factors it into term-feature and feature-document matrices. This provides topics and the weights of said documents in the input documents [3].

TF-IDF classifies the importance of a word in a document. It does this by multiplying the term frequency, the number of occurrences of a word over total words in a document, by the inverse document frequency, the log of the total number of documents over number of documents the word appears in. This gives a numerical value for all of the different words [4].

Combining these two methods, the authors were able to extract a number of topics and extract values from them. Their topics were money, job, friend, student, time and family, time, gratitude, pizza, and general. There were overlaps between these topics, and many words appeared more than once. However, they showed that certain topics led to a much larger chance of success while others performed much lower than the average. For instance, posts containing the money keywords typically had a 32% success rate, while those containing the friend keywords only had a 17% success rate. The words occuring in each narrative are listed in Table 2.

| Narrative | Words |
|---|---|
| Desire | friend, party, birthday, boyfriend, girlfriend, date, drinks, drunk, wasted, invite, invited, celebrate,celebrating, game, games, movie, beer, crave, craving |
| Family | husband, wife, family, parent, parents, mother, father, mom, mum, son, dad, daughter |
| Job | job, unemployment, employment, hire, hired, fired, interview, work, paycheck |
| Money | money, bill, bills, rent, bank, account, paycheck, due, broke, bills, deposit, cash, dollar, dollars, bucks, paid, payed, buy, check, spent, financial, poor, loan, credit, budget, day, now, time, week, until, last, month, tonight, today, next, night, when, tomorrow, first, after, while, before, long, hour, Friday, ago, still, due, past, soon, current, years,never, till, yesterday, morning, evening |
| Student | college, student, university, finals, study, studying, class, semester, school, roommate, project, tuition, dorm |

Table 2: List of Narratives with their words

# 3 Feature Preprocessing

Table 1 contains all of the features provided by the dataset. We first remove features that are unknown at time of posting to avoid using information from the future in predicting the response of a sample in the testing set.

As indicated in the introduction, five narratives were extracted from all the requests. Each narrative has keywords that signal that narrative and that influence the likelihood of success.

We create five features for each narrative, each one being the number of times words from a narrative appear in a request's post. In addition to narratives, we extract as a

binary feature the property of reciprocity. Reciprocity is a person's willingness to, in the future, give a pizza to someone on RAOP. For this feature, we search a post for phrases such as *return the favor*, *pay it back*, and *give back*. If one such phrase occurs, the feature is a 1, otherwise, the feature is a 0.

Additionally, we extracted textual features such as the number of words in a post, the number of words in a title, and whether a post has a link to another website that shows proof of the requestor's need.

We then removed some features from the 31 features provided to come up with a final list of 17 features:

> family narrative, requester number of comments, requester days since first post on raop , requester account age in days, reciprocity, number of words in request, requester karma, requester up votes plus down votes, request has link, job narrative, student narrative, number of words in title, desire narrative, requester number of comments in raop, requester number of posts on raop, requester number of subreddits followed, requester number of posts, and money narrative

We found feature weights from a logistic regression on to see the importance and correlation of each feature. Table 5 shows the weights.

# 4 Methods

We used several methods: PCA, KNN, Naïve Bayes, Logistic Regression, Artifical Neural Networks, and Random Forest. This section will explain how we implemented each method with references to packages we used. ROC plots and a summary table are provided in the results section.

## 4.1 Principal Component Analysis

Using Principal Component Analysis, we are able to reduce the number of features to components which are linear combinations of the original features. By selecting the components which capture most of the variance we can potentially save a lot of computation time. Figure 1 shows the variance of the different components.

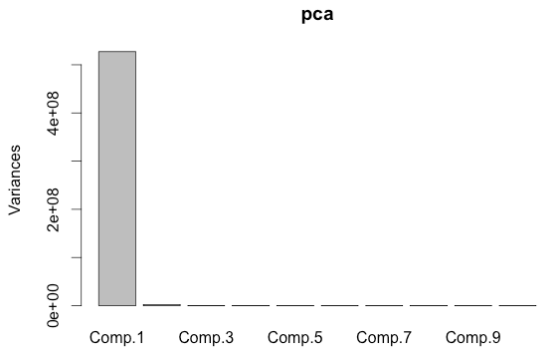


Figure 1: Variance of PCA components

We used the PCA results in our Random Forest model to see if using the top few components made our computation faster. The results can be seen in Table 4.

## 4.2 K-Nearest Neighbors

K nearest neighbors is an algorithm that predicts the response of a sample by taking the average of the k nearest samples from the training set, where near-ness is calculated as the Euclidian distance. KNN assumes each feature has the same weight when calculating the distance between samples. Features that are not very correlated with the response will still be used when calculating nearest neighbors, The 10-fold cross validation ROC plot for KNN are shown in Figure 2.

## 4.3 Naïve Bayes

Naïve Bayes was implemented using the E1071 package in R [11]. The 10-fold cross validation ROC plot for Naïve Bayes are shown in Figure 4.

## 4.4 Logistic Regression

We used several Logistic Regression methods. The first method was R's implementation using the built in glm() function [12]. It uses an iteratively reweighted least squares method. The 10-fold cross validation ROC plot for Logistic Regression are shown in Figure **??**.

We also used three other Logistic Regression methods implemented in Matlab. They are stochastic gradient ascent (SGA), with and without regularization, and Newton's Method.

The weight update equation used for SGA is:

$$w_j \leftarrow w_j + \alpha \left( y^{(i)} - g_w(x^{(i)}) \right) x_j^{(i)} \tag{1}$$

L2 regularization involves putting a penalty on the likelihood maximization with the sum of the weights squared. This encourages smaller weights and therefore combats over fitting [1]. The weight update equation for L2 regularization is:

$$w_j \leftarrow w_j + \alpha \left( -\lambda w_j + y^{(i)} - g_w(x^{(i)}) \right) x_j^{(i)} \tag{2}$$

Newton's Method utilizes the second derivative of the likelihood with respect to the weights, dividing our derivative times our learning rate. This will decrease our step towards maximizing the likelihood, if the second derivative is high (slope is increasing rapidly). If already at a maximum, the second derivative will be 0 and the weight will stay at that maximum. [2] The weight update equation for Newton's method is:

$$w \leftarrow w - H^{-1} \nabla J \tag{3}$$

H is the Hessian matrix:

$$H = \frac{1}{m} \sum_{i=1}^{m} \left[ g_w(x^{(i)})(1 - g_w(x^{(i)}) x^{(i)} (x^{(i)})^T \right] \tag{4}$$

$\nabla J$ is the gradient:

$$\nabla J = \frac{1}{m} \sum_{i=1}^{m} \left( g_w(x^{(i)}) - y^{(i)} \right) x^{(i)} \tag{5}$$

The 10-fold cross validation ROC plot for Newton's method is in Figure 6.

## 4.5 Artificial Neural Network

We implemented an Artificial Neural Network in matlab. We used a multilayer feed-forward network with backpropagation as the weight update equation[14]. The activation function used in the hidden and output layers is the sigmoid function:

$$g_w(x^{(i)}) = \frac{1}{1 + e^{-w^T x}} \tag{6}$$

Our weight update function is:

$$\delta_i^{outputlayer} = -(a_i^{outputlayer} - y_i) * g'_w(x^{(i)}) \tag{7}$$

$$\delta_i^j = -g'_w(x^{(i)}) * \sum_{n=1}^{k} w_{in}^j * \delta_n^{j+1} \tag{8}$$

$$w_{ij}^k = w_{ij}^k - \alpha * (a_i^k * \delta_j^{k+1}) \tag{9}$$

The 10-fold cross validation ROC plot for ANN is in Figure 7

## 4.6 Random Forest

The Random Forest learning method is a classifier that can carry out both classification and regression using a large forest or decision trees. At training time, the method creates a collection of decision trees that randomly select both features and samples, and then train on them. Then it gathers the output of each individual tree, and takes the mode of these outputs in order to find the most outputted values.

The method combines two key concepts, bagging and random selection, in order to achieve its output. Bagging, or *bootstrap aggregating*, is an ensemble algorithm that was created in order to minimize variance and avoid overfitting. It does this by creating a large number of data sets and samples with replacement. The random feature selection results in many different permutations of the data set being trained upon, and essentially circumvents manual feature selection [7].

Because our dataset is used for binary classification, the trees in the forest are decision trees, (whether the user gets pizza or not), rather than regression trees that output a value. This means that the final output value is decided by a vote: the case that receives the majority vote of the trees is selected as the output.

We implemented Random Forest with R's randomForest package [9]. The 10-fold cross validation ROC plot for Random Forest are shown in Figure 5.

# 5 Results

We used pROC to produce our ROC plots [10].

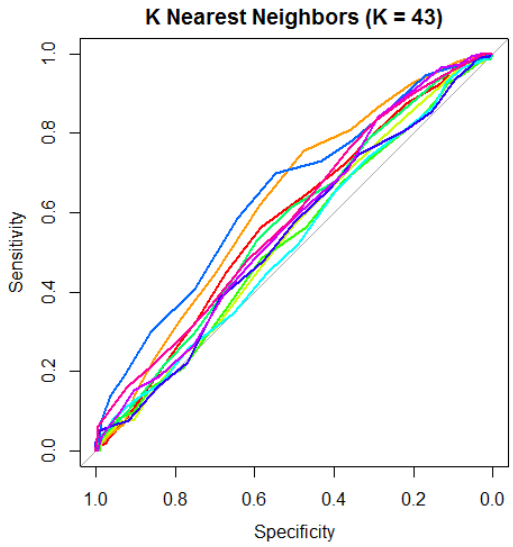


Figure 2: AUC = 0.5714

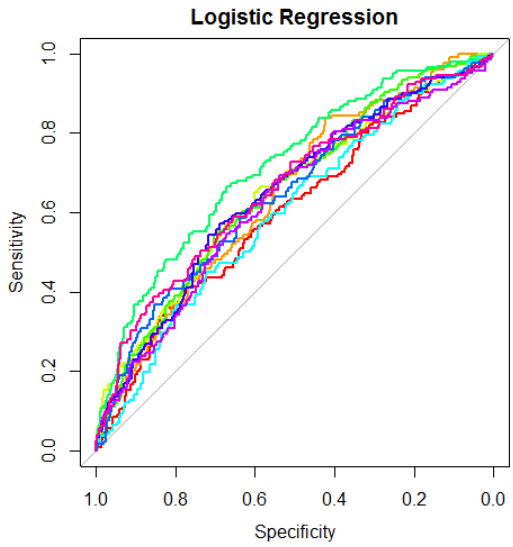


Figure 3: AUC = 0.6451

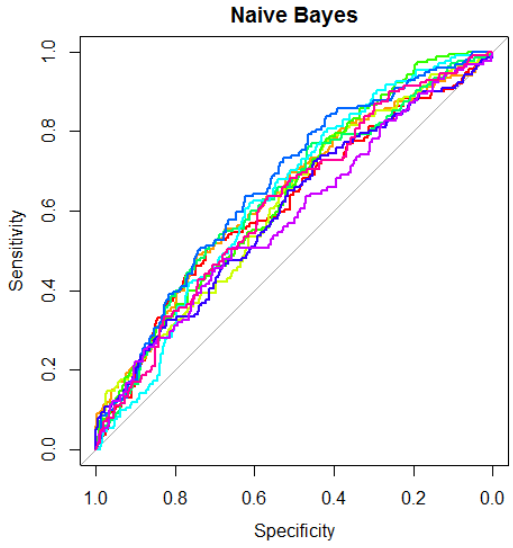


Figure 4: AUC = 0.6231

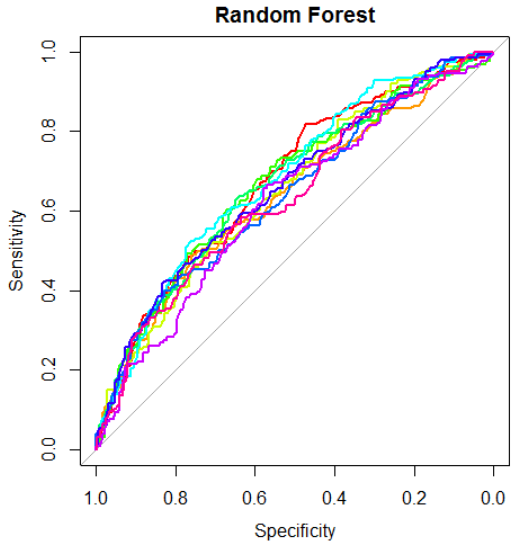


Figure 5: AUC = 0.6547

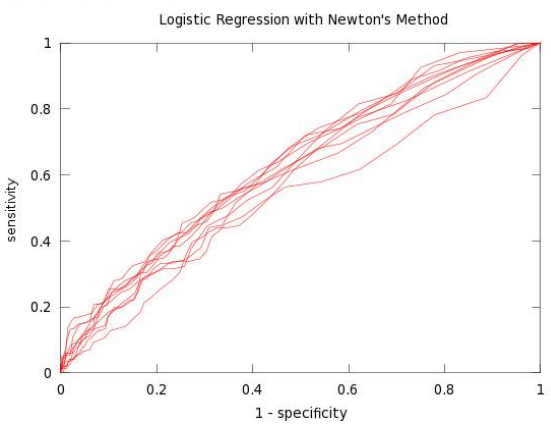


Figure 6: AUC = 0.6121

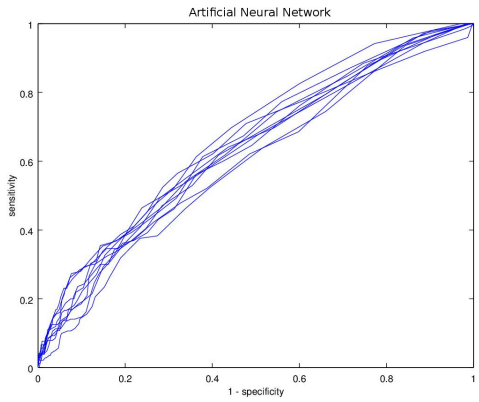


Figure 7: AUC = 0.6372

| Method | AUC |
|---|---|
| KNN | 0.5714 |
| Logistic Regression (R) | 0.6451 |
| Logistic Regression (SGA) | 0.6103 |
| Logistic Regression (L2) | 0.6282 |
| Logistic Regression (Newton's Method) | 0.6121 |
| Naïve Bayes | 0.6231 |
| ANN | 0.6372 |
| Random Forest | 0.6547 |

Table 3: Evaluation Summaries

| Dataset | Time(s) | AUC |
|---|---|---|
| Original | 70.80 | 0.6547 |
| PCA with components | 36.63 | 0.6466 |

Table 4: PCA Timing Comparison (using Random Forest)

| Feature | Weight |
|---|---|
| family narrative | 0.977298 |
| request has link | 0.761888 |
| money narrative | 0.800161 |
| job narrative | 0.766329 |
| number of comments | 0.258961 |
| student narrative | 0.479794 |
| days since first post on raop | 0.899458 |
| desire narrative | -0.053526 |
| account age in days | 0.136098 |
| number of words in title | 0.144902 |
| reciprocity | 0.646266 |
| number of posts | 0.586122 |
| number of words in post | 0.659996 |
| number of posts on raop | 0.485567 |
| karma | 0.636033 |
| number of comments in raop | 0.604201 |
| upvotes + downvotes | 0.094418 |
| number of subreddits followed | 0.713419 |

Table 5: Feature Weights from Logistic Regression

# 6 Discussion

## 6.1 Challenges We Faced

The most difficult part of feature selection was representing the requester's post in numerical form. While the narratives were helpful in understanding and categorizing a request, the depth and ambiguity of natural language led to some misclassification. We will look at two examples of RAOP posts that give the reader the general idea of what they look like and that demonstrate the difficulty of interpreting the text.

Here is the first example of a request that received pizza:

> The government screwed up and now we have to wait over a month for them to refile and reestablish my claim. There is no way to expedite this at all, in spite of the fact that it is their mistake. We have 2 girls (9 and 7) and 3 boys (5,3, 2months).
>
> My wife had to be taken by ambulance to the hospital last week for emergency gall bladder removal surgery and we are feeling a bit beat on at the moment. This would be a humungous pick-us-up. I am happy to provide any verification you need. Thanks in advance.

This post is a good example of a person going through a rough time that is looking for help. He clearly mentions his family and the narrative keyword wife is mentioned. A person reading the first sentence would argue the money narrative is present, but none of the narrative keywords are present. There is also a more subtle reference: because

the insurance claim was misfiled, his wife needed surgery, and they have 5 children, his family will likely be in some sort of financial trouble. An algorithm making this connection is beyond our capabilities. This example shows that there are narratives clearly present to people that won't be detected using the keywords of that narrative.

Here is another example of a request that received pizza:

> Times are really tough, I am a high-school chemistry teacher and dont know how much longer I will live. I just need to throw a pizza on my roof in frustration one more time before I die. Please help, also my son has a disability. Junior loves breakfast, but pizza will be fine. God bless.

This post is a reference to the popular TV Show *Breaking Bad*. A person familiar with the show will understand the reference, but this is a difficult problem for a computer [13]. Users commenting mentioned how they *"loved the reference"* and thought the post *"was one of the most original posts on RAOP"*. There is a keyword for the family narrative present, but the narratives don't apply in this case because people found this post funny or clever rather than appealing to emotion.

Also our dataset had a large proportion of negative samples (around 75%), so this made it difficult for some models to build a strong predictor. Also, because of the relatively small amount of overall samples, maintaining an even ratio of positive to negative samples in the testing set resulted in a small amount of bias.

## 6.2 Comparing Methods

KNN assumes each feature has the same weight when calculating the Euclidian distance between samples. Removing features that have a low correlation with the response (poster number of posts and comments, the number of words in the title of the post, account age, and number of subreddits) increases the 10-fold cross validation auc to 0.6026.

Table 4 shows the speed increase when using the PCA components instead of the original dataset. Random Forest ran twice as fast, at the cost of a lower AUC. If our dataset was much larger (around 100,000 samples) saving computation time would be important. However since we only have 5671 samples it is not necessary.

Random Forest performed the best although logistic regression and ANN were close. Of the three logistic regression in R runs the fastest. If we had a much larger dataset we would have to consider computation time more than we do now.

## 6.3 Narratives

Our results show that the desire narrative had a negative correlation with getting a pizza while the other narratives had a positive correlation. These results can be interpreted that potential givers would rather help someone that has genuine need rather than satisfying someone's craving for pizza. However, without a more sophisticated model we can't have complete confidence that this is the case.

Our results also show that the feature *Requester Number of Comments in RAOP* has a positive correlation with getting a pizza. This indicates that more active users in the community are more likely to write better posts. However it is not clear if givers

are more inclined to give pizza because they recognize a user, or if active users learn how to make better posts.

# 7 Conclusion

Our work can be applied to some of the other Random Acts subreddits such as Random Acts of Kindness. We can see if using the same narratives produces similar results or if we need to develop a new set. We then could attempt to make a broader statement about human nature if results are similar.

In summary, while our learning algorithms were were much better predictors than randomly guessing, there remains a lot of work to be done. We believe that by applying better NLP algorithms, we can extract more information from text to create a better predictor

# References


[1] Guestrin, Carlos. *L2 Regularization for Logistic Regression.* University of Washington. `http://courses.cs.washington.edu/courses/cse599c1/13wi/slides/l2-regularization-online-perceptron.pdf`

[2] Ng, Andrew. *Logistic Regression and Newtons Method.* Stanford University. `http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex4/ex4.html`

[3] *Non-Negative Matrix Factorization.* Wikipedia. Wikimedia Foundation, 25 Oct. 2014. `http://en.wikipedia.org/wiki/Non-negative_matrix_factorization`

[4] *Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining.* `http://www.tfidf.com/`

[5] Althoff, Danescu-Niculescu-Mizil, and Jurafsky. *A Case Study on the Success of Altruistic Requests.* Stanford University. `http://cs.stanford.edu/~althoff/raop-dataset/altruistic_requests_icwsm.pdf`

[6] Yee Whye Teh. *Random Forests.* Oxford University. Lecture Slides. `http://www.stats.ox.ac.uk/~teh/teaching/sdmHT2013/l15a-RF.pdf`

[7] Liaw, Andrew and Weiner, Matthew. *Classification and Regression by randomForest.* University of North Carolina. December 2002. `http://www.bios.unc.edu/~dzeng/BIOS740/randomforest.pdf`.

[8] Altman, N. S. *An introduction to kernel and nearest-neighbor nonparametric regression.* The American Statistician 46 (3): 175185, 2002.

[9] Breiman and Cutler. *Package randomForest.* R package. `http://cran.r-project.org/web/packages/randomForest/randomForest.pdf`

[10] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frdrique Lisacek, Jean-Charles Sanchez and Markus Mller (2011). *pROC: an open-source package for R and S+ to analyze and compare ROC curves.* BMC Bioinformatics, 12, p. 77. DOI: 10.1186/1471-2105-12-77.

[11] TU Wien. *Misc Functions of the Department of Statistics (e1071)*. R Package. http://cran.r-project.org/web/packages/e1071/e1071.pdf

[12] Rodriguez, German. *Generalized Linear Models*. http://data.princeton.edu/R/glms.html

[13] Gonzlez-Ibez, Muresan, and Wacholder. *Identifying Sarcasm in Twitter: A Closer Look*. School of Communication and Information, Rutgers. http://aclweb.org/anthology//P/P11/P11-2102.pdf.

[14] Rojas, Ral *The Backpropagation Algorithm*. Free University of Berlin http://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf

# 8 Author Contributions

- Zach worked on PCA, and helped prepare the presentation.
- Andrew helped with research for some of the methods and helped prepare the presentation.
- Anthony worked on the ANN, helped prepare the presentation, and wrote several sections of the report.
- Taronish worked on the Random Forest, ran tests with the PCA, Naïve Bayes, helped prepare the presentation, and wrote sections of the report.
- Matthew worked on plotting the ROC curves, and helped prepare the presentation, and wrote sections of the report.
- Alex worked on Logistic Regression in R, did LaTeXformatting and wrote several sections of the report.
- Yosi worked on KNN, feature preprocessing, helped prepare the presentation, and wrote several sections of the report.
- Blake helped with research.
- Shyam worked on the Logistic Regression using L2 regularization and Newton's method, helped prepare the presentation, and wrote sections of the report.