Classifying and Clustering Sorority Recruitment

Eric Fichtel[1], Katharine Welch[2]

Northeastern University[1,2]
fichtel.e@northeastern.edu[1], welch.kath@northeastern.edu[2]

Abstract

Sorority recruitment is a process built on conversations and connection for women to find an organization with similar values and interests. In this paper, we did a case study on one campus' Panhellenic Conference and their formal recruitment. Using self-reported qualities of potential new members, we used a variety of classification and clustering algorithms to determine if there was a pattern to which potential new members joined which sorority and if there were similarities between new members in the same sorority. This paper outlines our process of converting our dataset into features and implementing each algorithm, and discusses the issues that arose. While we were able to find a couple of features that were able to categorize the data best, due to a variety of reasons, we were not able to conclude a meaningful classification of potential new members.

Introduction

While sorority recruitment is defined by the numerous interactions of those seeking to join a sorority and the members of a sorority, we narrowed our analysis of this process to analyzing how the attributes of a prospective member shaped which chapter they joined. The attributes of a prospective member were captured through a registration form, which contained several questions about an individual's experiences and information relevant to joining a sorority. The first stage of our project consisted of transforming these text responses to meaningful data that could be consumed by a machine learning algorithm. We then used classification algorithms to determine if the features we extracted from the data could be used to accurately predict what chapter a prospective member would join. We determined that this was not possible with the given data due to not having enough samples and that the attributes of a prospective member alone are not sufficient to encapsulate the recruitment process. After gaining this insight, we shifted our analysis to finding commonalities between prospective members, independent of the chapter they joined.

Background

A sorority (also referred to as chapter) is a Greek-lettered social organization for college-aged women. For campuses that have more than one chapter, Panhellenic is the organization that oversees all inter-chapter relations, including formal recruitment. On most campuses, Panhellenic will hold formal recruitment once per school year to allow potential new members (PNMs) to join one of the chapters within the Panhellenic community. The dataset that we used for this research is from a Panhellenic Conference that contains ten sororities and, for the year we examined (2019), had about 500 PNMs register to go through the process.

Recruitment is a mutual selection process between chapters and PNMs that uses a variation of the Gale-Shapley Matching algorithm. The recruitment process consists of four rounds which take place over four days, with each sorority holding a specified number of events during each round. During the first round, each PNM will visit all sororities. At the end of the day, the PNMs will rank the sororities, with the lowest numbers being the chapters they would like to visit again. The chapters will also make a list of PNMs that they would like to invite back. This data is put into the matching algorithm that will create invite lists for each chapter and PNM for the next round. This process continues with PNMs visiting ten chapters on the first day, followed by eight, five, and two chapters respectively for each of the following rounds. After the final night, each PNM will be given a bid to join one of the two chapters that she visited during the final round.

During each event, PNMs will talk with a few members of the chapter so both the members and PNMs can learn about each other. Chapters attempt to match PNMs with members that have similar interests. For example, if a PNM is a computer science major, she will likely talk with a chapter member who also majors in computer science with the hopes they will connect and have a more meaningful conversation.

The dataset we used for this research comes from a completed recruitment process with ten sororities and about 500 PNMs. PNMs register through a form that asks them questions about their club involvement, abroad experience, hometowns, etc. This registration form is the main piece of data that we used for our research.

Related Work

There has not been any related work into classifying or clustering PNMs into chapters within sorority recruitment based on interests, majors, or any of the features that we looked at during our research. A marginally related research project looked at how race factored into recruitment and if it played a role in which chapter a PNM joined (Schmitz and Forbes 1994). This paper is outdated and was focused on a culturally different region of the United States. Formal recruitment is done in person so chapter members would know the race of a PNM and it could factor into the chapter's decision, but the race of each PNM was not contained in our dataset, so it is not a feature we looked into.

The algorithm used to create invitation lists for each round is a modified version of the Gale-Shapley algorithm. Other researchers have looked into using clustering models in combination with the stable matching algorithm. One paper looked into clustering a grid using k-nearest neighbors and k-means to group points based on distance to the center of a cluster (Eppstein, Goodrich, and Mamano 2017). Their research focused on how algorithms performed based on the setup of each experiment. Experiments differed based on location and number of clusters, while features remained constant. While we looked into their research and methods, based on the properties of our data, our experiments differed based on the number and definitions of features while our clusters remained constant.

Project Description

In this section, we will go into detail about our methodologies for each stage of our project. The first subsection will detail how we transformed the PNM survey data into features for our various models. The second subsection will detail our methodology in predicting the chapters that a PNM will join. We used three different classification models and optimized the accuracy of each model by implementing various feature selection techniques. The third subsection will detail our methodology in clustering the feature attributes of PNMs, independent of chapter.

CSV Processing

The PNM response CSV contains all of our data for this project and can be found in our repository under CSVProcessing/all_pnm_data.csv. The responses had no numerical columns that could be used as features. Thus, we had to use a combination of one-hot encoding and Term Frequency Inverse Document Frequency (TF-IDF) techniques in order to transform the PNM text responses into numerical input for our models. For short text responses, in which we were able to use our knowledge of the domain to form meaningful groupings, we used one hot encoding. For longer text responses, in which we were not able to categorize a priori, we defaulted to using TF-IDF scores. The implementations for all CSV processing described in this section can be found in CSVProcessing/PNM_data_processing.ipynb

We determined that several of the text responses to the PNM responses could be separated into finite, enumerable groups based upon their relevance to the recruitment process. For instance, the major of a PNM, could be categorized by the general domain of study it falls upon. We defined these major domains to be: Computer Science, Nursing, Culture, Pre-Med, Social Sciences, Business, Humanities, Physical Therapy, Math/Science, Engineering, Arts/Design, and Undeclared. These domains were the feature names for all of the one-hot features we assigned to the PNM major data. We then built a dictionary in which an individual major was the key and the major domain was the value. PNMs with multiple majors or minors would receive a 1 in multiple columns. We applied a similar methodology to encoding the *Abroad experiences*, *Legacies*, and *Hometown* data. The features chosen are as follows:

Abroad experiences were divided into NUin, International Student, Other (meaning any other form of study abroad) and None.

A legacy is a PNM whose family member was in a specific sorority (i.e. PNM 10 is a legacy to Chapter 2 because her mother was in Chapter 2 while in college). Chapters often put more emphasis on legacies during the recruitment process. For our *Legacies* features, we had one feature for each chapter (i.e. Chapter 1 Legacy). PNMs can be a legacy to more than one chapter. Any PNM who listed herself as a legacy to a chapter not contained within this Panhellenic Conference would have a "1" in the Other Legacy feature.

We divided each US state into regions, we believed to be relevant. Some states are listed as their own region if there were a lot of PNMs from that state. The *Hometown* features are West, South, Midwest, Northeast, Mass, NJ/NY, CA, and Outside US (international students).

For our major, abroad experience, and hometown features, the categories we chose were subjective and could affect the accuracy of the results. In future iterations of this project, these categories should be reevaluated.

The *High School Involvement* and *Leadership* data was much longer and more complex than the proceeding columns. Moreover, as the nature of the responses were open ended and outside the domain of our knowledge of the problem, it did not inherently make sense to attempt to use one-hot encoding to vectorize these columns. We opted to use TF-IDF scores instead of a bag of words approach as we recognized that many of the responses would use similar language to describe different activities. However, we were only concerned with the activities a PNM was involved in and not the tone nor description of the activities. Therefore, it made more sense to use TF-IDF scores instead of word frequencies. To calculate TF-IDF scores and translate the responses to features, we used the sklearn.feature_extraction.text library. We opted to use bi-grams and trigrams as feature names as unigrams did not encapsulate enough information.

varsity tennis	varsity track	varsity volleyball	vice president	volleyball team	youth group
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0
...	...	...	...	...	...
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.238156	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0
0.0	0.0	0.0	0.000000	0.0	0.0

Figure I: High School Involvement features and TF-IDF scores as values

Chapter Classification

In an attempt to quantify the relationship between a PNM's attributes and the chapter they join, we used the PNMs dataset to train and test several classification algorithms using the sklearn library. We used three different models for classification: a Support Vector Classifier (SVC), a Random Forest model and a Neural Network.

A SVC is an application of a Support Vector Machine, which is an algorithm that creates a hyperplane to separate points into two separate classifications. While originally a binary classification algorithm, we use the sklearn SVM library to classify the PNM data into eleven classes. The scikit SVM library's SVC uses a one to one method to construct k(k-1)/2 classifiers, "where each one is trained on data from two classes" seen in Equation I (Hsu and Lin). Below is the one to one method, which is a binary classification algorithm for a classes *i* and *j*:

$$\begin{aligned} \min_{w^{ij},b^{ij},\xi^{ij}} \quad & \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij} \\ & (w^{ij})^T\phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } y_t = i, \\ & (w^{ij})^T\phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{ if } y_t = j, \\ & \xi_t^{ij} \geq 0. \end{aligned}$$

Equation I: An example of a binary classification algorithm for SVC.

A random forest model is a group of decision trees. A decision tree is a series of nodes at which there are different potential outcomes. The trees work together as an ensemble where each produces a classification guess and the class with the majority from all decision trees is the output the RF model produces. The key to this algorithm is that the trees are not correlated to each other. If the trees are not correlated that implies that any errors will not be correlated. That way if one tree produces the wrong class, we hope the error would be diminished by a majority of the trees producing the correct class.

Neural Networks is an algorithm that is modeled after the human brain. It consists of many nodes that apply certain weights to the data running through the network. Nodes are placed in layers. There is one input layer with a node for each feature in the dataset and one output layer with a node for each target class. The network also contains a variable number of hidden layers between the input and the output. Nodes are given a weight. Each individual piece of data is sent through the network and the weights are reevaluated based on if the classification was correct or not.

In order to optimize the accuracies of these models, we experimented with a variety of feature selection methodologies. To get a baseline, we first ran each model on the entire dataset. In order to reduce the dimensionality of our feature vector, we implemented a variety of feature selection strategies. Our first strategy was to only run the model on an individual feature set (such as only using the Major features or only using the Involvement features) to reduce the dimensionality of the training set. From there we then used the Recursive Feature Elimination and SelectFromModel sklearn modules to algorithmically determine what features to eliminate for the Linear SVC and the Random Forest model. We will detail the accuracy of each model and what we learned from them in the results in the next section. As a result of sub optimal

accuracy after optimizing our classification models, we then used clustering models to further analyze our PNM data.

Clustering

In order to further analyze the PNM data, we created several clustering models to gain additional insight on the relationship between PNM attributes and the chapter they joined. The models we used were: K-Means, DBSCAN, Agglomerative Clustering. In order to evaluate the performance of each model, we compared their inertia and silhouette scores. Inertia is the sum of all of the squared errors per cluster. To find the correct number of clusters within our data, we use the elbow method. The elbow method is a heuristic that finds the point in a plot of inertia as a function of the number of clusters that minimizes both inertia and the total amount of clusters (Gupta 2019). Silhouette score is a measure of a point's similarity to its own cluster and dissimilarity to other clusters. The sklearn silhouette score is that score over all samples. A score of 1 is the best whereas-1 is the worst; a negative value implies that a point is in the wrong cluster (Sklearn 2009). Each of the clustering models' performances were compared by their silhouette score. We built each of these models using the sklearn.cluster library. In order to visualize each of the clustering results for each model, we used Principal Component Analysis (PCA) to reduce the feature set to two principal components. PCA is a feature selection process that groups together features by statistical significance.

Empirical Results

Our first area of inquiry was to try to predict the chapter a PNM would enter based upon their attributes. We used several classification algorithms to maximize the accuracy of these predictions to varying degrees of success. As a result of obtaining relatively low accuracies from attempting to predict the chapter a PNM will join, we then ran several clustering algorithms to gather additional insights into the relationship of attributes between PNMs and the chapters they join. The first subsection contains the results of the classification algorithms and an explanation as to why we were unable to obtain accurate predictions. The second subsection contains the results of the results of the clustering algorithms and what additional insights we drew from those results.

Chapter Classification Results

In using the SVC, we experimented with three parameters: the kernel, the C value and the gamma value. In order to find the optimal parameters for the SVC, we used the GridSearchCV module, which performs a "cross-validated grid-search over a parameter grid" (Sklearn 2017). The figure below contains the best accuracy we achieved with each kernel. We found that the C and Gamma values did not change per kernel and that the RBF kernel provided the best accuracy. We believe that the optimal C and Gamma values were the same across each kernel as they had a large effect on the accuracy level of the models, due to the sparsity of the dataset.

Kernel	Accuracy
Linear	0.138
RBF	0.262
Poly	0.138

Figure II: The rests of parameter tuning the SVC model

In writing the algorithms for Random Forest and Neural Networks, we followed a walkthrough that created both algorithms simultaneously (Ray 2018).
For the Random Forest algorithm, we used the RandomForestRegressor from sklearn.ensemble. There were a few parameters that we looked at, including the number of estimators and the size of the training data. While we tested out different options for parameters, our testing was manual. In future iterations of this project, we would like to automate this testing in order to optimize the parameters. For estimators, which represents the number of sequential trees in our forest, we picked 10. We did not want to increase that number too much because of concerns of overfitting. For our train test split, we did a 70-30 split in order to have enough data in the train set to get meaningful results.

When creating our neural network model, our input and output layers were already set for us. For the output layer, there were eleven targets, one for each chapter, and one for not joining a chapter. The input layer depended on the data we were using for that run, but would end up being the number of features in the dataset. The number of hidden layers was variable. Similar to the Random Forest model, the testing for parameter tuning was done manually, but we would like to automate it in the future. Through our basic tests on our initial dataset, we determined using ten hidden layers gave us the best results.

After running the models on the entire dataset, we then ran our models on a subset of features in order to reduce the dimensionality of the feature set and obtain better

results. We first chose the features based upon the different attributes the features represented (such as Legacy or Major). We then optimized our feature selection by using the Recursive Feature Selection module from Scikit, which chooses the best features from a model based upon feature importance, in the case of the Random Forest, and feature coefficient, in the case of the SVC with a Linear Kernel. Figure II depicts the relationship between the number of features selected from a model and the accuracy of the SVC with a Linear Kernel using the selected features. The SVC was most accurate with six features selected: Pre-Med, Business, Arts/Design, CA, NUin, Other Abroad Experience, with an accuracy of 27.7% (shown in Figure III). These features are relatively encompassing of the relevant attributes of a PNM.

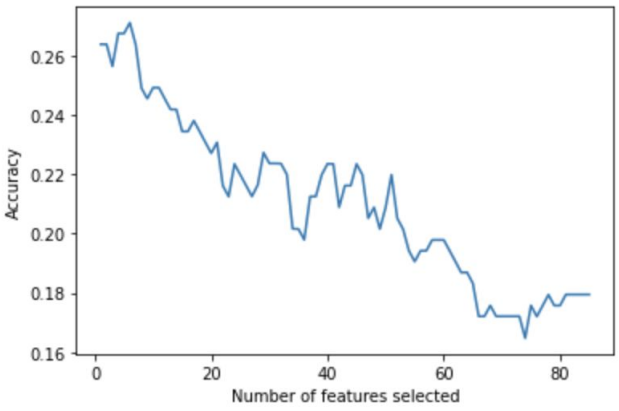


Figure II: The accuracy of the SVC with a linear kernel as a function of number of features selected via Recursive Feature Selection

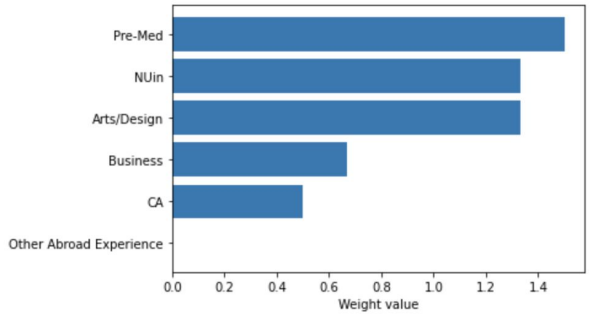


Figure III: Weights for RFE selected features for an example label

graph of feature selection for each model optimizing per model clustering

For the Random Forest Model, we used Recursive Feature Elimination with an emphasis on feature importance to trim our dataset. We determined the model was most accurate with two features: National Honors and Honors Society. National Honors had an importance rating of 0.55 while Honors Society had 0.44. After running the Random Forest model with only these two features, we found the accuracy to be 16.5%, shown in Figure IV below.

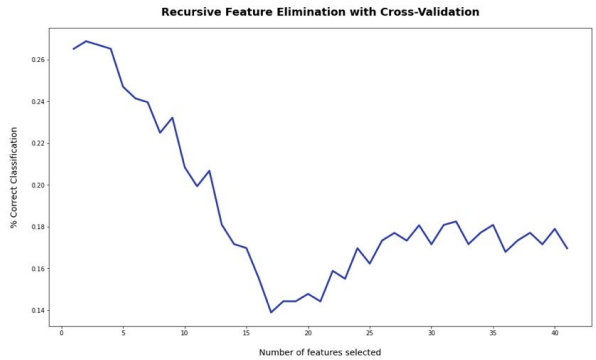


Figure IV: The accuracy of the RF model as a function of number of features selected via Recursive Feature Selection

After running through different iterations of each classification model, we compared the results of each model at each stage of our experiment: initial tests, per feature, RFE. The results are shown in Figure V.

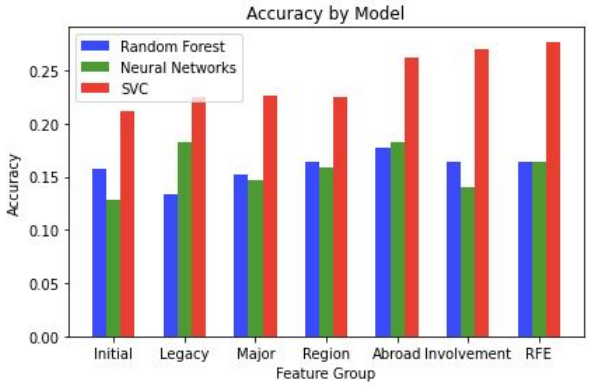


Figure V: The accuracy of each classification model with varying number of features.

It is clear that across all experiments SVC performed better. We believe this has to do with the complexity of the dataset. Since the data was noisy due to human input and lack of clear guidelines, it was difficult to pull out the relevant information. We believe that this affects the accuracy of the Random Forest and Neural Network models. With these complex models, we would like to fine tune our parameters to hopefully produce more useful results.

Additionally, models running with only the Abroad features consistently produced better results. With a

majority of students in the Panhellenic Conference participating in some form of study abroad, as well as a third of PNMs participating in NUin, it makes sense that study abroad would be a talking point during events and would affect which chapter a PNM joins.

Chapter Clustering Results

To determine how to cluster our data, we used the K-Means clustering algorithm and experimented with the number of clusters used in assigning labels. The graphs below contain the inertia and silhouette scores of K-Means as a function of clusters assigned.

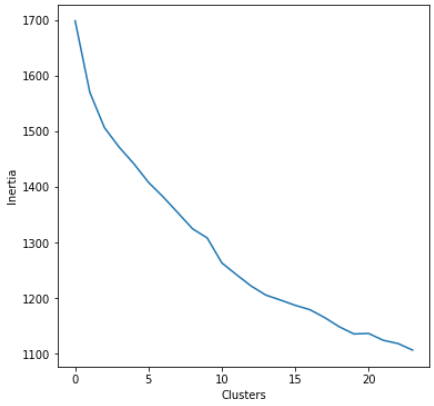


Figure VI: Interia of K-Means vs clusters assigned

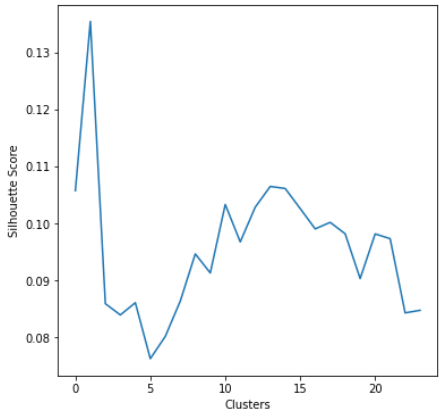
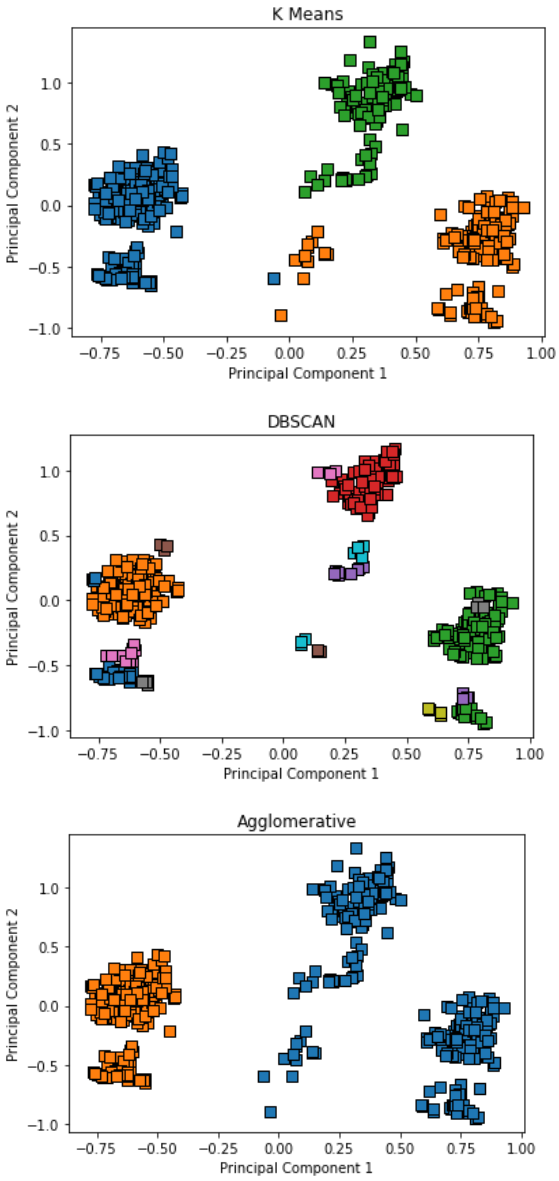


Figure VII: Silhouette Score of K-Means vs clusters assigned

The silhouette score and interia of K-Means reveal that the optimal number of clusters for our data is three clusters. Figure VIII below contains the silhouette scores and distinct clusters for other clustering algorithms we experimented with. The graphs below contain a visual representation of the clusters computed by the clustering algorithms after reducing the feature set to two components using Principal Component Analysis.

Algorithm	Distinct Clusters	Silhouette Score
K-Means	3	.136
DBSCAN	5	-.225
Agglomerative	2	.094

Figure VIII: Distinct Clusters and Silhouette Score of various clustering algorithms.



Figures IX: The above graphs depict a visual representation of the clustering of the PNM attributes after performing PCA

We found that the majority of the components' explained variance was derived from the Abroad Experiences features. Although they only explained 18% of the

variance, we found this to be a significant finding as no other feature set contributed nearly the same to the explained variance. Therefore, it can be concluded that amidst our noisy data, abroad experiences have statistical significance in distinguishing PNMs.

Conclusion & Future Directions

While each model we used gave us different pieces of our result, we were unable to converge to a classification or clustering of PNMs with strong accuracy. We believe this is due to the complexity of the problem and the dataset. While recruitment theoretically can be simplified to conversations between two groups of people, it tends to be a very complex process with several moving parts. For example, each round of recruitment is a long continuous day. Chapter members and PNMs will likely have varying degrees of energy throughout the day which could affect the quality of conversation. This, as well as many other aspects of the process, cannot be quantified in the data that we used. Additionally, our dataset was not consistent between PNMs. While all the same questions were answered, non-descriptive and short answer questions produced a variety of response types. The varying data likely put an inaccurate amount of emphasis on certain features for some PNMs.

Despite the issues we faced, we were able to draw some conclusions. With the three classification models we used, we determined that SVC produced the highest accuracy, classifying over a fourth of the PNMs correctly. Overall, all three models (RF, NN, and SVC), all found high accuracy using only the abroad features. SVC produced it's most accurate results with only the involvement features, while Neural Network found it's best results with the legacy features.

For the classification models, we also added Recursive Feature Elimination to our models. SVC found greater accuracy with this model than previously tested feature sets. The model had an accuracy of 27.7% with the following six features selected: Pre-Med, Business, Arts/Design, CA, NUin, and Other Abroad Experience. Random Forest found similar accuracy results with the RFE model as compared to other feature selection experiments. RF produced an accuracy of 16.4% with Honor Society and National Honors being the features selected.

For the clustering models, we determined that K-Means clustering was the most effective model, which clustered the PNM data into three clusters. After performing PCA, we determined that these two clusters were defined by the abroad experiences of a PNM. Future analysis could focus on how to best represent abroad experiences as features and see if abroad experiences correlate with any other PNM attributes.

The dataset that we used was a registration form with broad questions and little structure. A large portion of the time spent on this project was converting the data into something that could be an input to our models. To simplify and speed up the process due to the time constraints, we had to manually convert some of the data. For example, the major features were grouped based on the college the major resided in, apparent similarities between majors, and what we believe to be important to the process. Consequently, this process was very subjective. With more time, we would like to reevaluate how we group features together and determine an algorithmic way to arrange the feature set.

Additionally, we only used one recruitment period's data for our research. We likely overfit our algorithm to one year's process. Not only do we think more data could give us better results, but it could also help us determine which features are important to a chapter over several years rather than an anomaly. We would also like data from multiple years so we could begin to determine the features that define a chapter. For example, if a substantial number of computer science majors join a chapter one year, we can determine how that will influence the next new member class.

Due to time constraints, there was a limit to how much we were able to analyze and edit our algorithms. For our Random Forest and Neural Network algorithms, there were a number of parameters required. While we were able to manually test with a variety of inputs, the tests were not exhaustive and likely did not produce optimal values. With more time, we would like to automate this process so we can produce more significant results, especially when the inputs are changing.

One of the biggest issues with our project was our dataset. Because of the human element of the data, it was difficult to turn the set into something that was usable and meaningful for our models. For a beginner project with a time constraint, we suggest using a groomed dataset, so students can focus more on optimizing models rather than converting and modifying the dataset

For future CS4100 students, we recommend just trying new things. Since this project was so open-ended, there was little structure to which algorithms we used, how we implemented them, etc. We used this to our advantage by just trying different things. From reading machine learning articles online, we found innovative ways to attack different problems. Most articles also have walkthroughs on how to implement each algorithm, so it is easy to get a baseline for each model and then work on optimizing it. We have both learned so much by using different tools and models that we found online.

References

Eppstein, D.; Goodrich, M. T.; and Mamano, N. 2017. Algorithms for Stable Matching and Clustering in a Grid. *Lecture Notes in Computer Science Combinatorial Image Analysis*

Gupta, Alin D. 2019. Elbow Method for Optimal Value of k in KMeans. *GeeksforGeeks*

Hsu, Chih-Wei.; Lin, Chih-Jen. A Comparison of Methods for Multi-class Support Vector Machines. *Department of Computer Science and Information Engineering - National Taiwan University.*

Ray, Sunil. 2019. Building Your First Neural Network on a Structured Dataset (Using Keras). *Medium Analytics*

Schmitz, S.; and Forbes, S. A. 1994. Choices in a no-choice system: Motives and biases in sorority segregation. *Journal of College Student Development 35(2), 103 - 108*

*S*klearn. 2009. Sklearn.metrics.silhouette_score.*Scikit, scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.*

Sklearn. 2017. Sklearn.mode_selection.GridSearchCV. *Scikit,*

scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.