

Professional Members' Perceptions of Proposed Rule Changes in All Star Cheerleading

by

Aricka Gates

Submitted in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

in the

Mathematics and Statistics Program

YOUNGSTOWN STATE UNIVERSITY

May, 2017

Professional Members' Perceptions of Proposed Rule Changes in All Star Cheerleading

Aricka Gates

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

Aricka Gates, Student Date

Approvals:

Dr. Lucy Kerns, Thesis Advisor Date

Dr. G. J. Kerns, Committee Member Date

Dr. Andy Chang, Committee Member Date

Dr. Salvatore A. Sanders, Dean of Graduate Studies Date

Abstract

The U.S. All Star Federation is the governing body of All Star Cheer in the United States. They surveyed their professional members to better understand their opinions on proposed rule changes. The object of this paper is to predict how the members voted based on their demographic variables. Classification models including multinomial logistic regression, support vector machines, neural networks, and decision trees were used to predict the member's opinion on the proposed rule changes. The predicted response was compared with the actual response to determine how well the classifiers were performing. All the classifiers used in this study had about the same accuracy when predicting unseen observations.

Acknowledgements

I would like to thank Dr. Lucy Kerns, Dr. G. J. Kerns, and Dr. Chang for their help with this thesis. I would also like to thank Amy Clark for providing me with the data necessary to conduct this study.

Contents

1	Introduction	1
2	Literature Review	3
3	Methods	5
3.1	Classification Methods	5
3.1.1	Multinomial Logistic Regression	5
3.1.2	Neural Networks	7
3.1.3	Support Vector Machines	10
3.1.4	Decision Trees	12
3.1.5	rpart	13
3.1.6	ctree	13
3.2	Performance	13
4	Results	17
4.1	Data set	17
4.2	Preliminary Data Analysis	19
4.3	Multinomial Logistic Regression	23
4.4	Neural Networks	29
4.5	Support Vector Machine	31
4.6	Decision Trees	32
4.6.1	rpart	32
4.6.2	ctree	34
5	Analysis	36
6	Conclusion	40
	References	42
7	Appendix: R Code	46

1 Introduction

The All Star Cheerleading industry is comprised of many different roles and businesses and can be difficult for a person to understand unless they are directly involved within the industry. At the micro level, the industry is made up of athletes who have a desire to participate in All Star Cheerleading. There are thousands of athletes involved in this sport, competing in various levels and age brackets. Every athlete belongs to a program. Some programs have multiple gym locations, while others maintain a single gym location. Athletes pay to go to these gyms to practice cheerleading and to belong to competitive cheerleading teams.

Every gym has an owner and a coaching staff. Gyms are typically structured so that they have multiple teams at different age levels and different skill levels. Many gym owners fulfill multiple administrative roles and are also coaches. It is common for a coach to be involved with multiple teams within a program. Some specialty coaches, such as tumbling coaches, stunting coaches, and choreographers travel between several gyms to provide expertise.

Competitions are an important part of All Star Cheerleading. Teams go to events and compete against other teams in the same age group and level. The people or companies who run competitions are called event producers. At every competition there is a panel of judges. The judges score each performance and determine the winners for each age division and skill level.

At the macro level, the governing body of All Star Cheerleading in the United States is the U.S. All Star Federation (USASF), a nonprofit organization founded in 2003 [35]. The USASF sets the industry's standards by making and enforcing rules for competitive cheerleading. There are about 150,000 athletes and 15,000 coaches that are currently registered members of the USASF [1].

The USASF's mission statement is, "To support and enrich the lives of our All Star athletes and members. We provide consistent rules, strive for a safe environment for our athletes, drive competitive excellence, and promote a positive image for the sport." [35] To achieve their mission, the U.S. All Star Federation institutes rules, guidelines, and policies that their members must abide by. USASF coaches are required to pass a background check and must receive appropriate training and credentials according to the levels that they are

coaching. Athletes are required to provide proof of their birthdate to ensure that they are competing in their appropriate age group. The U.S. All Star Federation also certifies safety judges and sanctions events. Sanctioned events are competitions that meet the safety criteria set by the USASF. Event producers must go through an application process for their competition to become a sanctioned event. USASF sanctioned events are required to have at least one safety judge on their panel of judges. These safety judges are experts on the tumbling and stunting rules that are specific to each age division and each level. The USASF holds regional conventions every summer to ensure that the coaches are up to date on new research, policies, and guidelines. These meetings also provide training for coaches covering the stunts and tumbling allowed at each level and the correct way to teach these skills.

One of the USASF's main goals is to keep their athletes safe. In order to do this, they constantly update their rules and guidelines. New studies that provide information on how to achieve proper development and protect the safety of young athletes are being performed every year. The USASF has a Rules Committee that is tasked with developing, maintaining, and enforcing rules in the cheerleading industry [35]. Every two years the Rules Committee polls the USASF members using an online survey to determine their opinions on proposed rule changes and policy updates [1].

The survey examined in this study was sent out to the professional members of the U.S. All Star Federation to gather their opinions on the most recent proposed rule changes. In the past, the results from these polls were simply looked at as votes. In this study, the votes were examined to see if they were related to the demographic variables given by each member. It was hypothesized that there would be a relationship between a member's demographic variables and the way the member voted. For example, it could be argued that coaches and owners may vote differently because their roles are very different. Owners tend to view All Star Cheerleading through a business lens, and they have to be concerned with the amount of money the business is making. Coaches spend a lot of time with the athletes and because of their experience; they may have different opinions than those of the owners regarding certain topics. A second example of a variable that may influence a member's voting trend is the size of the gym they are from. There are small programs and large programs but they all compete at the same competitions. A member from a small gym may have a much different experience than a member from a larger gym and may vote

differently because of that experience.

The goal of this study is to predict the USASF member's opinions using their demographic information. Many different classifiers will be used and compared to determine which classifier is most accurate in predicting how each member may vote.

2 Literature Review

Today's technological advances in analytics and information collection allow companies to capitalize by using machine learning. Facebook tracks what their users view and then uses that information to display suggested advertisements and content for each user. Netflix tracks the shows their users watch and uses that information to predict other shows their users might enjoy. Amazon uses the recent purchases and searches of their customers to suggest other related products to their consumers. The more information these companies gather, the better they are able to predict the preferences and opinions of their users and customers. Similarly, the goal of this study is to use USASF member's demographic information to predict the member's opinions on proposed rule changes.

Yang et al.[38] performed a study that attempted to predict a person's brand preference using their personal traits as well as data collected from social media. The personal traits, which were collected via survey, examined included demographic variables, personality, personal values, and information about individual needs. They determined values for personality, personal values, and individual needs by using a model developed by psychologists. The demographic variables that the researchers used in this study were gender, age, marital status, education, and income. In trying to predict brand preference using these variables, the prediction variable was split into 3 classes: positive, negative, or neutral. The classifiers that were used in this study were AdaBoost, Decision Tree (C4.5), Logistic Regression, Naive Bayes, and Random Forest. They found that Naive Bayes performed the best in their study. They did mention that none of their models were optimized. They used the default parameters that were set by Weka.

Currently there are few studies that have been performed that examine classifying survey results for proposed cheerleading rules. While the subject of Yang et al. research is not directly related to the subject of the present study, their study provides a good foundation for studies dealing with predictive surveys. Further review of research that is closely related

to the subject area of the present study will follow.

Thomas et al. [7] performed a study that analyzed mini rugby union coaches' perceptions of competitive activities. Their data set came from an online survey consisting of 31 questions. The survey contained questions that polled beliefs on how mini rugby should be played at a young age and also collected demographic information. Thomas et al. used Hierarchical cluster analysis and were able to cluster their data set into three groups. There was a significant difference between the groups. They studied the groups and found the significant differences between the groups were age and the highest level of rugby played.

Greco [9] performed a study analyzing coaches' opinions on performance enhancing drugs. The demographic variables included in his study were gender, number of years coaching at the high school level, coaching certificate, the coaches' level of compensation, grade level coached, and the sports coached. The dependent variables examined the study were drugs that the coaches believed were performance-enhancing drugs, if the coaches believed the athletes should be tested for drugs, and what coaches believed the punishment should be for athletes who tested positive for performance-enhancing drugs. Greco ran an ANOVA and a Chi-Squared test to analyze the differences of opinions based on each demographic variable. The goal of his study was to determine if each independent variable was significant in determining the coaches' opinion on performance enhancing drugs.

Moston et al. [24] performed a study examining athletes' and coaches' perceptions of deterrents for performance enhancing drugs. The demographic variables included in their study were role, age, gender, main sport, highest level of competition, and history of anti-doping testing. The athletes and coaches were asked to answer four questions about their opinion of performance enhancing drugs. The goal of their study was to see if a significant difference existed between the coaches and athletes regarding their opinions on performance enhancing drugs. The researchers used a t-test to determine if the means of the two groups differed. They reported that the demographic variables they examined did not account for a significant difference between the categories and opinions of the coaches and athletes. The only independent variable used was role.

The goal of the present study is to use the demographic variables of USASF coaches to predict their opinions on proposed rule changes in All Star Cheerleading. Classification was not used in any of the studies that analyzed sport related surveys.

3 Methods

3.1 Classification Methods

There are many different methods that can be used to classify observations into predefined classes. In this study, the variable that is being predicted is program definition. The variable, program definition, represents the member’s opinions on if the program definition should be altered. The goal is to predict if the professional members chose keep, change, or abstain for program definition using demographic variables. The classification methods used in the study were multinomial logistic regression, neural networks, support vector machines, and decision trees.

The data was split into a training data set and a test data set. The training data set was used to build the model. The models were used to classify the test data set into the three predefined classes. The test data set had no influence on creating the models, because the test data contains all unseen observations [32]. The results predicted from the model were compared with the true values of the test data set. This comparison was used to see how well the model was working. The same training and test data sets were used for all four models.

80% of the data set was used for the training data. The training data set has 864 observations. The remaining 20% of the data set was used for the test data set. The test data set has 215 observations.

3.1.1 Multinomial Logistic Regression

Multinomial logistic regression is a generalization of binary logistic regression and can be used when the dependent variable has more than two categories [5]. Multinomial logistic regression should be used when the categories have no natural order. In multinomial logistic regression, one category is used as the base category and the other categories are compared to it. If the dependent variable has c categories, the model requires $c - 1$ equations. There will be a $c - 1$ binary logistic models. Each model will represent one category relative to the reference category [25]. Each equation describes the relationship between the dependent variable and the independent variable. All the equations will have an intercept term and coefficients for the predictor variables.

In this study, the response variable has 3 categories: keep, change, and abstain. The

category keep was chosen as the base category.

There are certain assumptions that need to be met in order to run a multinomial logistic regression. One assumption, which was already mentioned, is the dependent variable needs to be nominal. The categories of the dependent variable need to satisfy the Independence of Irrelevant Alternatives (IIA) assumption [21]. This assumption means that the odds of each category should stay the same if we add or subtract categories. If the odds change, this assumption will fail. The multinomial logistic regression model works for independent variables that are categorical and/or continuous. There should be no multicollinearity in the model [37]. Multicollinearity would occur when independent variables have a strong correlation.

Dummy variables need to be used when the data set contains categorical variables. Every categorical variable compares each of its categories to a base category [25]. The first category in each variable was the default for the base category.

Equation 1 and Equation 2 can be used to represent the general multinomial logistic regression models for this study. Equation 1 shows the effect of the predictors on the probability of the member choosing change versus the probability of the member choosing keep.

$$\log \frac{P_{change}}{P_{keep}} = \beta_{0c} + \beta_{1c}x_{1c} + \beta_{2c}x_{2c} + \dots + \beta_{20c}x_{20c} \quad (1)$$

$(x_{1c}, x_{2c}, \dots, x_{20c})$ represent the predictor variables used in this model.

$(\beta_{1c}, \beta_{2c}, \dots, \beta_{20c})$ represent the regression coefficients.

β_{0c} represents the intercept term.

The second equation shows the effect of the predictors on the probability of the member choosing abstain versus the probability of the member choosing keep.

$$\log \frac{P_{abstain}}{P_{keep}} = \beta_{0a} + \beta_{1a}x_{1a} + \beta_{2a}x_{2a} + \dots + \beta_{20a}x_{20a} \quad (2)$$

$(x_{1a}, x_{2a}, \dots, x_{20a})$ represent the predictor variables used in this model.

$(\beta_{1a}, \beta_{2a}, \dots, \beta_{20a})$ represent the regression coefficients.

β_{0a} represents the intercept term.

The multinomial logistic regression model can be used to predict the probability for all the categories of the dependent variable [5]. Equation 3 can be used to find the probability of the reference category keep. Equation 4 can be used to find the probability of change. Equation 5 can be used to find the probability of the member choosing the option abstain. The three probabilities will always add up to equal one.

The main goal of this multinomial logistic regression model is to classify each observation as keep, change, or abstain. The instance will be classified as the category that has the highest probability.

$$P(K) = \frac{1}{1 + \exp(\beta_{0c} + \beta_{1c}x_{1c} + \dots + \beta_{20c}x_{20c}) + \exp(\beta_{0a} + \beta_{1a}x_{1a} + \dots + \beta_{20a}x_{20a})} \quad (3)$$

$$P(C) = \frac{\exp(\beta_{0c} + \beta_{1c}x_{1c} + \dots + \beta_{20c}x_{20c})}{1 + \exp(\beta_{0c} + \beta_{1c}x_{1c} + \dots + \beta_{20c}x_{20c}) + \exp(\beta_{0a} + \beta_{1a}x_{1a} + \dots + \beta_{20a}x_{20a})} \quad (4)$$

$$P(A) = \frac{\exp(\beta_{0a} + \beta_{1a}x_{1a} + \dots + \beta_{20a}x_{20a})}{1 + \exp(\beta_{0c} + \beta_{1c}x_{1c} + \dots + \beta_{20c}x_{20c}) + \exp(\beta_{0a} + \beta_{1a}x_{1a} + \dots + \beta_{20a}x_{20a})} \quad (5)$$

3.1.2 Neural Networks

Neural networks are a type of machine learning that can be used to classify observations. This model is very similar to the multinomial logistic regression model [4]. The neural network classification model finds weights for each independent variable. It uses these weights to determine the class that the observation should be classified as. The independent variables are sometimes referred to as the input layer [10]. Neural networks have hidden layers that influence the classification of the data. The hidden levels can make decision on a more abstract level than the first layer [31]. The output layer is how the instance is classified [10].

Neural networks are modeled after the brain and its system of processing knowledge. The human nervous system is composed of a network of cells called neurons that communicate with one another to transport information [3]. Neurons are often referred to as the building blocks of the nervous system. A neuron contains axon terminals, which receive signals from

other neurons, and dendrites, which relay signals to other neurons [3]. If a signal received by a neuron is strong enough, it is sent through the cell and will be transported to the next cell. This process will start over with the new neuron.

A simple type of artificial neural networks is perceptrons. Perceptrons take all binary inputs and produce a single binary output. A perceptron can be modeled using the equation below.

$$\text{output} = \begin{cases} 0 & \text{if } \sum w_i x_i \leq \text{threshold} \\ 1 & \text{if } \sum w_i x_i > \text{threshold} \end{cases} \quad (6)$$

x_1, \dots, x_i represent the variables used in the model and w_1, \dots, w_i represent the weights of each variable.

It is common to see this equation using the term bias instead of threshold. The perceptron's bias will be represented by b . It will be easy for the perceptron model to output a 1 if the bias is a large positive number. It will be hard for the perceptron model to output a 1 if the bias is a very negative number.

This idea can be compared to the anatomy of a neuron. If the signal received by the dendrites from the axon terminal was strong, the signal will be relayed to the next neuron. If the signal received by the dendrites was very weak, it would not be relayed to the next neuron. The perceptron can be modeled as follows:

$$\text{output} = \begin{cases} 0 & \text{if } \sum w_i x_i + b \leq 0 \\ 1 & \text{if } \sum w_i x_i + b > 0 \end{cases} \quad (7)$$

A step function can be used to represent the output from perceptrons since the output is either a 0 or a 1.

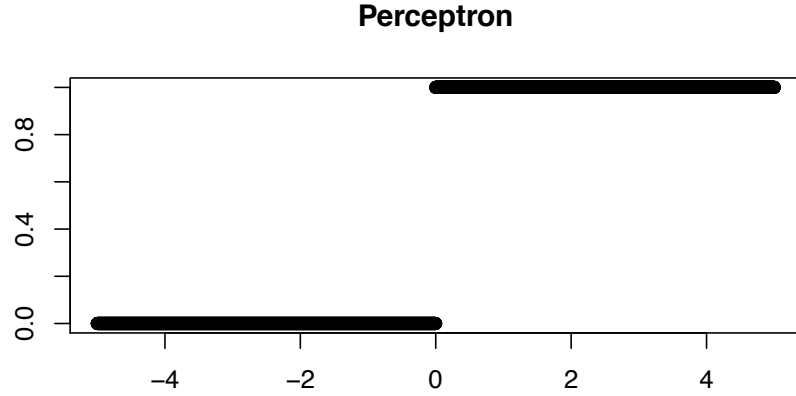


Figure 1: Perceptron Model

In this study, the sigmoid neuron is used. This type of artificial neuron is very similar to the perceptron neurons. The output variables for sigmoid neurons can be on the interval from 0 to 1 [10]. The sigmoid function, which is also known as the logistic function, can be represented by the following formula:

$$\frac{1}{1 + \exp(-(\sum w_i x_i + b))} \quad (8)$$

Let $z = \sum w_i x_i + b$. The sigmoid function can now be defined by the following formula:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (9)$$

As z gets really positive, the function will go to one. As z gets really negative, the function will go to 0. This is exactly what happened in the perceptron. The difference in the models occurs when z is not an extremely positive, or an extremely negative number. The sigmoid function will output values on the interval from 0 to 1. This model will become more complex if hidden layers are included in the model. The model can be represented by the following graph:

The main goal of this study is to predict if the member chose keep, change, or abstain for the response variable. The dependent variable has three possible classes, so the output

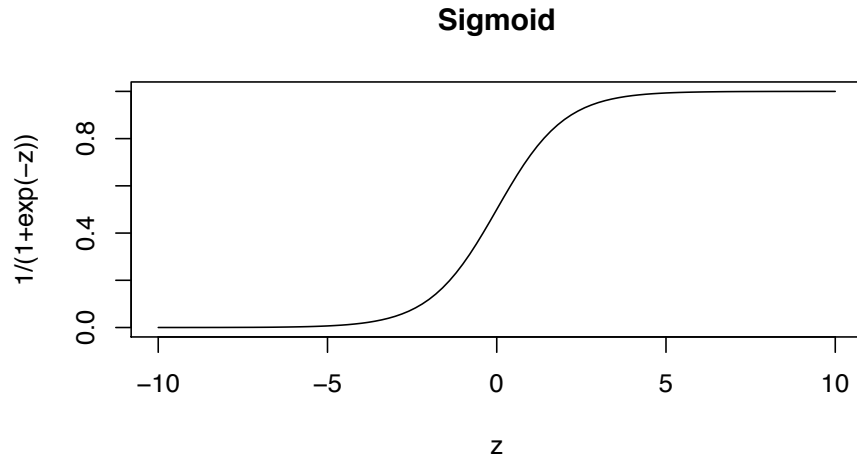


Figure 2: Sigmoid Model

layer for this study will have three categories. The category with the highest output value will be how the instance is classified [26].

The function neuralnet was used for this study. This function comes from neuralnet package in R. All the categorical variables must be transformed using dummy variables. The continuous variables should be scaled. Scaling the continuous variables will help the model perform better [29]. This package uses resilient back-propagation to determine the minimum error function [8]. The number of hidden nodes can be set [8].

3.1.3 Support Vector Machines

Support vector machines (SVM) are a type of classifier that uses a hyperplane to classify the data. Support vector machines were originally created for binary classification [14]. There are now multiple approaches used to classify multiclass data sets [22]. The goal is split the data so that most of the observations are classified correctly and the margin is as big as possible. When the margin is maximized, the model will perform better with out of sample data. Large margins give more room for error.

Support vector machines are a very common type of machine learning. They are often used with data set containing many dimensions. Support vector machines normally have a high accuracy [23]. The theory behind support vector machines is based in linear algebra [36]. Support vector machines sometimes overfit or underfit the data [16].

The general idea of support vector machines are relatively easy to understand when looking at a two-dimensional space with linearly separable data. The figure below shows a data set with two classes that can be correctly classified using a straight line.

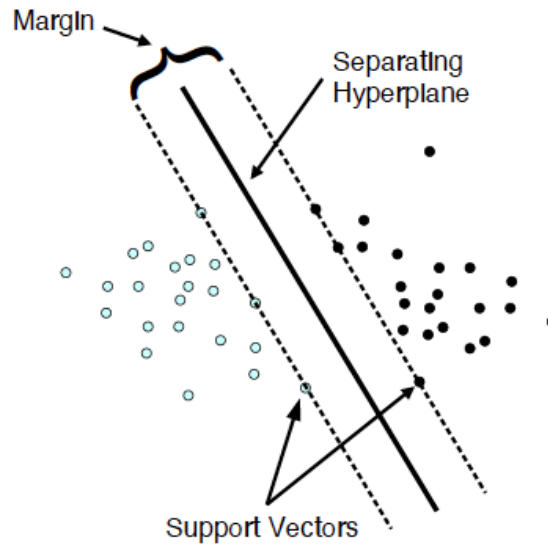


Figure 3: Support Vector Machines
Source: Reprinted from "Support Vector Machines" by D. Meyer

From the figure above, it is obvious that the line correctly classifies all the observations. All the data points above the line belong to one class. All the data points below the line belong to another class. There are many ways that the line could be inserted such that the data is classified correctly. The goal is to insert the line so the marginal distance is maximized. The closest data point to the line is known as the support vector [23].

When the data cannot be linearly separated, it is mapped to a space with higher dimension [36]. The data becomes linearly separable in the larger dimensional space [23]. The kernel can be adjusted to account for non-linear data. There are four common options for the kernel: linear, polynomial, radial, and sigmoidal [23].

The model requires that all categorical variables are transformed using dummy variables [12]. Numeric variables should be scaled on the interval from $[0,1]$ or $[-1,1]$ [12]. The SVM function in the R package `e1071` was used for this study. This function transforms all the variables to meet the model requirements. The default kernel in this package is radial. The function `tune.svm` was used. This function runs a loop to determine the best values

for the parameters gamma and cost [12]. Once the best values were determined from the tune.svm function, they were used to fit the model. This package uses the "one-against-one" approach to model multiclass data sets [22]. This method makes a model for each pair of classes [22].

3.1.4 Decision Trees

Decision trees are a type of machine learning that follow the general structure of a tree. The model starts at the root, the root is then broken into different branches, and the branches eventually lead to leaves [19]. The branches are also known as the decision nodes. Each branch represents different categories of predictor variables. Branches may lead to another branch or to a leaf. A leaf represents the class that the instance will be classified as [32].

Once the decision tree is created, it is relatively easy to classify new attributes. Decision trees use the variables that are important when predicting the dependent variable. Many independent variables may be excluded from the model. Decision trees can be looked as a flow chart [19]. No computations need to be done to determine how attributes should be classified.

A popular technique when forming decision trees is the divide and conquer approach. This approach is more accurate and more complete than other methods [19]. The divide and conquer approach uses recursive partitioning. The classifier looks at the entire data set and all the predictor variables. The model chooses the variable that is the most predictive for the outcome. The most popular approach to find the most predictive value is by calculating entropy [32]. The variable that contains the most information will be used for the first split. The data will then be split up into different categories based on that variable. The process will start all over and will continue until almost all of the attributes are in the same class. The decision tree could also stop if it became too large or has no information left to use [19].

Decision trees can run into problems when the data set has lots on nominal variables with many levels. It can also run into problems when there are many numeric variables. If either of these occurs, the model may overfit or underfit the data.

There are many different types of decision trees that follow the divide and conquer technique. In this study, two different decision tree classifiers will be examined: rpart (also

known as CART) and ctree. It is expected that the outcomes from this data set should be very similar.

3.1.5 rpart

The classifier rpart is from the rpart library in R. Rpart stands for recursive partitioning and regression trees [33]. Rpart is also known as CART (classification and regression trees) [6]. The default option for splitting data is using the gini index [6]. The gini index is an alternative approach to calculate the most predictive variable.

3.1.6 ctree

The classifier ctree is from the PARTY library in R. Ctree uses significance tests in order to select variables to split the data [11]. Ctree uses the following outline to determine how to split data sets:

Null Hypothesis: Independence between input variables and response variable

If the null hypothesis cannot be rejected, stop here.

If the null hypothesis can be rejected, chose a variable that has the strongest relationship to the response variable. Split the data using this variable.

Repeat the above steps [12].

3.2 Performance

A confusion matrix will be examined to determine how each classifier is performing. A general confusion matrix is represented in Table 1.

Table 1: General Confusion Matrix

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	A	B	C	$A + B + C$
	Change	D	E	F	$D + E + F$
	Abstain	G	H	I	$G + H + I$
Total		$A + D + G$	$B + E + H$	$C + F + I$	N

The numbers on the main diagonal represent the number of observations that were classified correctly. A represents the observations that correctly predicted the response variable as keep. E represents the observations that correctly predicted the response variable

as change. I represents the observations that correctly predicted the response variable as abstain.

The accuracy is found by adding the correctly placed observations together and dividing by the total number of observations [30]. Accuracy is one of the easiest measures to interpret when decided how well the model performed. Accuracy can sometimes be misleading. Accuracy does not do a good job of explaining how the model is performing for all classes. If one class is overbearing, the model may predict all observations to that class. The accuracy for this model would be very high, but the model is not doing a good job because it is only predicting the class with the highest frequency.

$$\text{Accuracy} = \frac{A+E+I}{N} \quad (10)$$

$A + E + I$ represents the number of correctly classified observations and N represents the total number of observations in the data set.

Cohen's kappa statistic can also be used to determine how well the classifier is doing. Kappa statistics take into account the possibility of objects being classified correctly by chance [28]. Kappa statistics are more complicated to calculate than accuracy. The following equations show how to calculate the kappa statistic.

$$\text{Kappa} = \frac{(A + E + I) - \sum_{i=1}^3 EF_i}{N - \sum_{i=1}^3 EF_i} \quad (11)$$

$\sum_{i=1}^3 EF_i$ represents the sum of the expected frequencies.

The following formulas can be used to find the expected frequencies.

$$\text{Class1 : (Keep)} = EF_1 = \frac{(A + B + C)(A + D + G)}{N} \quad (12)$$

$$\text{Class2 : (Change)} = EF_2 = \frac{(D + E + F)(B + E + H)}{N} \quad (13)$$

$$\text{Class3 : (Abstain)} = EF_3 = \frac{(G + H + I)(C + F + I)}{N} \quad (14)$$

There is a general table to represent how well the classifier is doing by using the kappa statistic. The table below shows how the kappa statistic should be interpreted [34].

Table 2: Kappa Statistic

<0	No Agreement
0.0-0.2	Very Low Agreement
0.21-0.40	Low Agreement
0.41-0.60	Moderate Agreement
0.61-0.80	Full Agreement
0.81-1.00	Almost Perfect Agreement

The accuracy and kappa statistic are the two main measures that will be used to see how well the classifier is performing overall. It is also important to determine how well the classifier is performing for each response class [30]. Sensitivity, specificity, positive predictive value, negative predictive value, balanced accuracy, detection rate, prevalence, and detection prevalence are measures that show how well the model is predicting each class of the dependent variable [2]. In order to calculate these measures the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) must be used [30]. These values can be found using the confusion matrix. These values are intuitive when there are only two classes, but they become more confusing as more classes are introduced. The following table will provide a reference for finding the true positive, true negative, false positive, and false negative values when the response variable has three classes.

Table 3: TP, TN, FP, FN

<i>Class</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
True Positive(TP)	<i>A</i>	<i>E</i>	<i>I</i>
True Negative(TN)	$(E + F + H + I)$	$(A + C + G + I)$	$(A + B + D + E)$
False Positive (FP)	$(B + C)$	$(D + F)$	$(G + H)$
False Negative (FN)	$(D + G)$	$(B + H)$	$(C + F)$

Note that the table above is using the variables from the General Confusion Matrix (Table 1).

True positive value refers to the number of observations that were correctly placed. True negative represents the number of observations that were correctly not classified as the specified class. False positives represent the number of observations that were predicted

incorrectly for a specified class. False negatives represent the observations that were actually the specified class but they were incorrectly placed.

Sensitivity is a measure that looks at the proportion of correctly placed observations of a specific class to the total number of actual observations from the specific class [27]. The equation below shows how to calculate sensitivity using the true positive and false negative values.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (15)$$

Specificity measures the proportion of observations that were correctly not classified as the specified class to the total number of actual observations for the other classes (not the specified class) [27].

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}} \quad (16)$$

The positive predictive value looks at proportion of correctly placed instances for a certain class to the total number of observations that were predicted for that class [18]. The positive predictive value can be represented using the true positive and false positive values.

$$\text{Positive Predictive Value} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (17)$$

The negative predictive value looks at the number of observation that were correctly not classified as the specified class and is divided by the total number of predicted observation for the other classes (not the specified class)[27]. The negative predictive value can be found using the true negative and false negative value.

$$\text{Negative Predictive Value} = \frac{\text{TN}}{\text{TN}+\text{FN}} \quad (18)$$

There are four other equations that can be used to analyze how well the classifier is doing with each class. The equations for these for measures are listed below.

$$\text{Balanced Accuracy} = \frac{\text{sensitivity}+\text{specificity}}{2} \quad (19)$$

$$\text{Detection Rate} = \frac{\text{total number of correctly placed observation for specified class}}{N} \quad (20)$$

$$\text{Detection Prevalence} = \frac{\text{total number of observation predicted for specified class}}{N} \quad (21)$$

$$\text{Prevalence} = \frac{\text{total number of actual observation for specified class}}{N} \quad (22)$$

The caret package from R was used see how well the model was performing [17]. The confusionMatrix function was used to determine the confusion matrix, accuracy, kappa statistic, sensitivity, specificity, positive predictive value, negative predictive value, balanced accuracy, detection rate, prevalence, and detection prevalence for each classifier [17].

4 Results

4.1 Data set

The data set used in this study came directly from raw data collected using an online survey distributed by the USASF via Survey Monkey. The survey was emailed to USASF professional members who passed their background checks and were up to date on their membership fees. The members had one week to fill out the survey. The survey included questions on demographic information as well as rule changes. Eight demographic variables were examined in this study: region, role, experience, size, division, location, Level 5/6, and gender.

The first demographic variable examined was region. The USASF is broken down into five regions: West, Northeast, Southeast, Southwest, or Midwest.

The role variable examined the primary job of the member completing the survey and was broken down into four options: owner, coach, program director, or other.

The demographic variable of experience was a discrete variable that looked at how many years the member has been involved in All Star Cheerleading.

The variables of size and division were closely related. The size variable looked at the number of cheer athletes the program thought they would have for the 2016-2017 season. There were nine options the member could pick from: less than 50, 51-75, 76-100, 101-125,

126-150, 151-175, 175-200, 200-299, and 300 or more. The U.S. All Star Federation contains two divisions. Division II contains programs with less than 126 athletes, while Division I is made up of programs with more than 126 athletes.

Location was a demographic variable that was concerned with the number of different locations a program had. There were three options the member could choose: single location, multiple locations sharing athletes, multiple locations not sharing athletes.

Levels represent the difficulty of the routines performed and the skills of the athlete. Level 5 and Level 6 are the highest levels. Teams that compete in these levels are performing the most advanced routines. The demographic variable Level 5/6 was a binary variable used in this study. It was coded as 1 if the program has a Level 5 and/or a Level 6 team. The variable was coded as 0 if the program did not have a Level 5 or Level 6 team.

The last demographic variable used in this study was the gender of the member completing the survey. The members could chose male or female when filling out the survey.

The survey contained several different questions about rule change policies. For the purpose of this study, the only question that was examined was Program Definition. The survey first gave background information on the current definition. "Branded programs" are programs that have more than one location. This becomes a problem when trying to determine the size and division of a program. Should these programs be considered separately or together? Another issue that comes into play with branded program is the issue of crossover athletes. Crossover athletes are athletes that compete for more than one gym location.

Currently the branded programs are broken into two groups that are based on whether or not they share athletes.

If the program shares athletes:

The total athlete count is used to determine eligibility for division status.

When the total athlete count is over 125, the programs is no longer eligible for Division II status.

If the program does not shares athletes:

Each location is viewed as their own program. The athlete counts are separate for each location.

The following are the new options proposed for Program Definition by the U.S. All Star

Federation:

Option 1: Keep as is.

Option 2: Branded programs will have only one profile.

Option 3: Branded programs will not be eligible for Division II regardless of size.

Each member was asked to pick the option that best matched their beliefs on how to define a program. They were able to choose from the following list of options:

1. I am in favor of Option 1 "Keep As Is"
2. I am in favor of change (I prefer Option 2 or Option 3)
3. Abstain

For the rest of this study, the three categories for program definition will be referred to as keep, change, and abstain.

4.2 Preliminary Data Analysis

The total number of professional members that responded to the survey was 1196. 117 observations had missing values and were not used in this study. 101 of those observations were due to the member stopping the survey part way through. The other 16 observations were deleted because one of the variables used in this study was missing. 1079 observations were used in this study.

Table 4: Region

Program Definition	Southwest	Southeast	Northeast	Midwest	West	Sum
Keep	59	118	99	59	41	376
Change	83	113	177	123	107	603
Abstain	12	30	23	16	19	100
Sum	154	261	299	198	167	1079

From the table above, it is clear that Northeast region had the most members fill out the survey. The Southwest region had the least amount of people fill out the survey. All the regions, except the southeast region, chose the category change most frequently. The southeast region had the highest number of votes for keeping the program definition as it currently is.

The role with the highest frequency in this survey was coach. The role with the lowest

Table 5: Role

Program Definition	Owner	Coach	Program Director	Other	Sum
Keep	104	195	66	11	376
Change	217	272	99	15	603
Abstain	27	63	8	2	100
Sum	348	530	173	28	1079

frequency in this survey was other. Owners, coaches, program directors, and others all chose the category change the most, then the category keep, and finally the category abstain.

Table 6: Experience

Minimum	0 years
Mean	10 years
Maximum	40 years

When it comes to experience there is a wide variety of members who filled out this survey. The smallest amount of experience was 0 years and the greatest amount of experience was 40 years. There was a mean experience of 10 years.

Table 7: Division

Program Definition	Division I	Division II	Sum
Keep	167	209	376
Change	142	461	603
Abstain	36	64	100
Sum	345	734	1079

Most of the members who filled out this survey are affiliated with a division II gym. Division II programs chose the category change most frequently. Division I programs chose the category keep most frequently.

Table 8: Size

ProDef	S1	S2	S3	S4	S5	S6	S7	S8	S9	Sum
Keep	62	64	43	40	34	14	19	30	70	376
Change	167	132	87	75	45	21	28	22	26	603
Abstain	27	20	8	9	7	4	3	5	17	100
Sum	256	216	138	124	86	39	50	57	113	1079

Where S1= less than 50, S2=51-75, S3=76-100, S4=101-125, S5=126-150, S6=151-175, S7=175-200, S8=200-299, and S9= greater than 300

The size with the highest frequency in this survey programs with less than 50 athletes.

The role with the lowest frequency in this survey was programs with 151-175 athletes. Change is the most popular vote for the programs with less than 200 athletes. Keep is the most popular vote for programs with more than 200 athletes.

Table 9: Location

Program Definition	Single	Multiple-Sharing	Multiple-NOT Sharing	Sum
Keep	197	53	126	376
Change	513	31	59	603
Abstain	66	14	20	100
Sum	776	98	205	1079

Most of the members who filled out this survey were from a single location program. The smallest number of members was affiliated with multiple location gym that shared athletes. Single location gyms had the highest number of votes for the category change. Multiple sharing and multiple not sharing had the highest number of votes for the category keep.

Table 10: Level 5/6

Program Definition	No Level 5/6	Level 5/6	Sum
Keep	157	219	376
Change	364	239	603
Abstain	58	42	100
Sum	579	500	1079

The number of members who were affiliated with a gym that had level 5 or level 6 athletes was very close to the number of members who were not affiliated with a gym that had level 5 or 6 athletes. The most popular vote in both categories was change, followed by keep, then abstain.

Table 11: Gender

Program Definition	Male	Female	Sum
Keep	116	260	376
Change	140	463	603
Abstain	23	77	100
Sum	279	800	1079

There were 800 females who filled out this survey and only 279 males. Change was the most popular vote between both genders.

Figure 4 is a mosaic plot that examines the relationships between program definition,

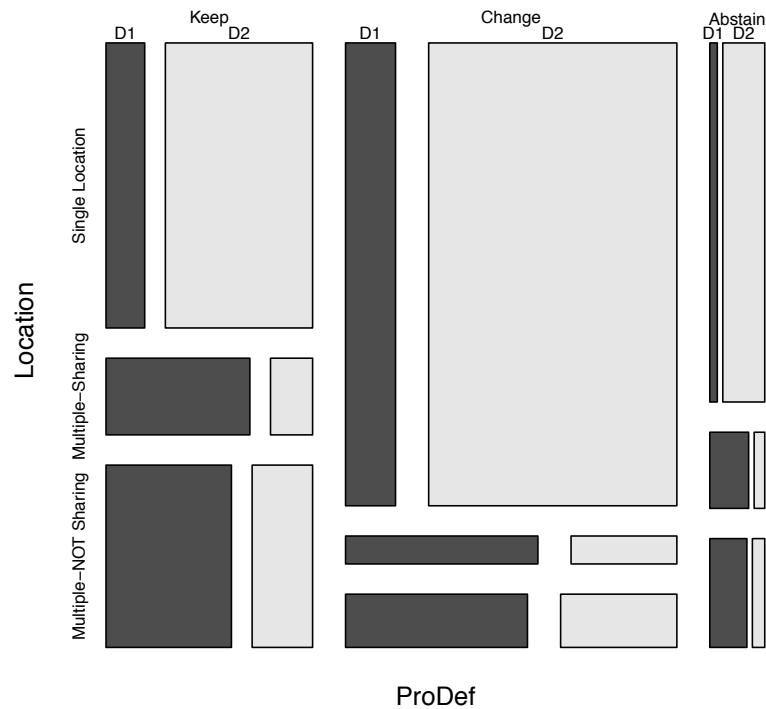


Figure 4: Mosaic Plot: Program Definition, Division, Location

division, and location. From the plot it is clear that change is the program definition that was chosen the most frequently. Single location programs chose change most frequently. Multiple locations chose keep most frequently. Member affiliated with division II programs chose change twice as frequently as they chose keep. Members affiliated with division I programs had a pretty even split between keep and change

80% of the data set was used for the training data. The training data set has 864 observations. The remaining 20% of the data set was used for the test data set. The test data set has 215 observations. The same training and testing data sets were used for each classifier.

The table above shows the frequency for program definition for the entire data set, the training data set, and the testing data set.

Table 12: Program Definition Votes

	Keep		Change		Abstain	
Entire Data Set	376	35%	603	56%	100	9%
Training Data Set	313	36%	467	54%	84	10%
Testing Data Set	63	29%	136	63%	16	7%

4.3 Multinomial Logistic Regression

Multinomial logistic regression was run on the training data. The data set contains eight predictor variables. The variable size and the variable division are directly related. In order to avoid multicollinearity problems, the variable division was excluded from this classification model. The multinomial logistic regression model used the remaining seven variables.

Table 13 shows the results from the regression. This table is looking at the response class Change and comparing it to the response class Keep. The regression coefficients and the intercept were used to make the to regression equation for this model. The formula for this model is shown by equation 24.

Table 13: Multinomial Logistic Regression: Change with respect to Keep

<i>Variable</i>	<i>Estimate</i>	<i>Z-value</i>	<i>P-value</i>	<i>Significance</i>
Intercept	1.178	3.419	0.001	***
Size51-75	-0.210	-0.863	0.388	
Size76-100	-0.251	-0.905	0.365	
Size101-125	-0.236	-0.752	0.452	
Size126-150	-0.632	-1.665	0.096	
Size151-175	-0.162	-0.327	0.743	
Size175-200	-0.145	-0.322	0.748	
Size200-299	-0.634	-1.407	0.160	
Size300 or more	-0.985	-2.418	0.016	*
LocationMult-Sharing	-1.135	-3.471	0.001	***
LocationMult-NOT Sharing	-1.418	-5.854	0.000	***
RoleCoach	-0.167	-0.884	0.377	
RoleProgramDirector	0.177	0.712	0.477	
RoleOther	-0.873	-1.738	0.082	
RegionSoutheast	-0.206	-0.777	0.437	
RegionNortheast	0.320	1.209	0.227	
RegionMidwest	0.242	0.858	0.391	
RegionWest	0.390	1.309	0.191	
Experience	0.010	-0.747	0.455	
GenderFemale	-0.169	-0.887	0.375	
L56Level	0.105	0.455	0.648	

From Table 13, the only variables that are significant, at $\alpha = 0.05$ level, are the intercept term, Size, and Location. The model can be represented using the formula below.

$$\begin{aligned}
 \log \frac{P_{change}}{P_{keep}} = & 1.178 - 0.210x_1 - 0.251x_2 - 0.236x_3 - 0.632x_4 \\
 & - 0.162x_5 - 0.145x_6 - 0.634x_7 - 0.985x_8 - 1.135x_9 - 1.418x_{10} \\
 & - 0.167x_{11} + 0.177x_{12} - 0.873x_{13} - 0.206x_{14} + 0.320x_{15} \\
 & + 0.242x_{16} + 0.390x_{17} + 0.010x_{18} - 0.169x_{19} + 0.105x_{20} \quad (23)
 \end{aligned}$$

where $x_1 = \text{Size51-75}$, $x_2 = \text{Size76-100}$, $x_3 = \text{Size101-125}$, $x_4 = \text{Size126-150}$,
 $x_5 = \text{Size151-175}$, $x_6 = \text{Size175-200}$, $x_7 = \text{Size200-299}$, $x_8 = \text{Size300 or more}$, $x_9 =$
 $\text{Location:Mult-Sharing}$, $x_{10} = \text{Location:Mult-NOT Sharing}$, $x_{11} = \text{Role: Coach}$, $x_{12} =$
 $\text{Role: ProgramDirector}$, $x_{13} = \text{Role: Other}$, $x_{14} = \text{Region: Southeast}$, $x_{15} = \text{Region: Northeast}$,
 $x_{16} = \text{Region: Midwest}$, $x_{17} = \text{Region: West}$, $x_{18} = \text{Experience}$, $x_{19} = \text{Gender: Female}$,
and $x_{20} = \text{Level:5/6}$

Table 14 shows the results comparing the response class abstain with the reference class keep. The regression coefficients and the intercept were used to make the to regression equation for this model. Equation 24 shows the formula for this model.

Table 14: Multinomial Logistic Regression: Abstain with respect to Keep

<i>Variable</i>	<i>Estimate</i>	<i>Z-value</i>	<i>P-value</i>	<i>Significance</i>
Intercept	-1.185	-2.020	0.043	*
Size51-75	-0.096	-0.248	0.804	
Size76-100	-0.532	-1.096	0.273	
Size101-125	0.030	0.059	0.953	
Size126-150	0.280	0.467	0.640	
Size151-175	-0.677	-0.590	0.555	
Size175-200	-0.302	-0.343	0.731	
Size200-299	0.812	0.111	0.912	
Size300 or more	0.546	0.912	0.362	
LocationMult-Sharing	0.032	0.066	0.947	
LocationMult-NOT Sharing	-0.362	-0.931	0.352	
RoleCoach	0.222	0.730	0.465	
RoleProgramDirector	-0.690	-1.375	0.169	
RoleOther	-0.372	-0.456	0.649	
RegionSoutheast	0.333	0.719	0.472	
RegionNortheast	0.357	0.755	0.450	
RegionMidwest	0.306	0.610	0.542	
RegionWest	0.951	1.935	0.053	
Experience	-0.014	-0.648	0.517	
GenderFemale	0.0119	0.039	0.969	
L56Level	-0.742	-0.767	0.077	

Table 14 shows that the only significant variable, when predicting abstain, is the intercept term. This shows that predicting abstain may be hard. There does not appear to be a relationship between the predictor variables and the response class abstain.

$$\begin{aligned}
\log \frac{P_{change}}{P_{keep}} = & -1.185 - 0.096x_1 - 0.532x_2 + 0.030x_3 + 0.280x_4 \\
& - 0.677x_5 - 0.302x_6 + 0.812x_7 + 0.546x_8 + 0.032x_9 - 0.362x_{10} \\
& + 0.222x_{11} - 0.690x_{12} - 0.372x_{13} + 0.333x_{14} + 0.357x_{15} + 0.306x_{16} \\
& + 0.951x_{17} - 0.014x_{18} + 0.0119x_{19} - 0.742x_{20}
\end{aligned} \tag{24}$$

$x_1 = \text{Size51-75}$, $x_2 = \text{Size76-100}$, $x_3 = \text{Size101-125}$, $x_4 = \text{Size126-150}$,

$x_5 = \text{Size151-175}$, $x_6 = \text{Size175-200}$, $x_7 = \text{Size200-299}$, $x_8 = \text{Size300 or more}$, $x_9 = \text{Location:Mult-Sharing}$, $x_{10} = \text{Location:Mult-NOT Sharing}$, $x_{11} = \text{Role: Coach}$, $x_{12} = \text{Role: ProgramDirector}$, $x_{13} = \text{Role: Other}$, $x_{14} = \text{Region: Southeast}$, $x_{15} = \text{Region: Northeast}$, $x_{16} = \text{Region: Midwest}$, $x_{17} = \text{Region: West}$, $x_{18} = \text{Experience}$, $x_{19} = \text{Gender: Female}$, and $x_{20} = \text{Level:5/6}$

The multinomial logistic regression model was used to predict what option the member would chose for program definition. The test data set was applied to this model. Equations 3, 4, and 5 were used to determine the probability of each class for each observation. The highest probability was chosen and that is how the data was classified. The confusion matrix shows the results. The test data set had 215 observations. 145 of the test observations were classified correctly.

Table 15: Confusion Matrix: Multinomial Logistic Regression

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	37	28	6	71
	Change	26	108	10	144
	Abstain	0	0	0	0
Total		63	136	16	

The multinomial logistic regression model was rerun with only the significant factors. The only predictive variables in the new model are size and location. Table 16 shows the regression results.

Table 16: Multinomial Logistic Regression: Change with Respect to Keep (significant factors only)

<i>Variable</i>	<i>Estimate</i>	<i>Z-value</i>	<i>P-value</i>	<i>Significance</i>
Intercept	1.001	6.110	0.000	***
Size51-75	-0.160	-0.671	0.502	
Size76-100	-0.200	-0.757	0.449	
Size101-125	-0.168	-0.598	0.550	
Size126-150	-0.588	-1.779	0.075	
Size151-175	-0.064	-0.144	0.885	
Size175-200	-0.068	-0.169	0.866	
Size200-299	-0.494	-1.220	0.223	
Size300 or more	-0.889	-2.478	0.013	*
LocationMult-Sharing	-1.126	-3.564	0.000	***
LocationMult-NOT Sharing	-1.427	-6.218	0.000	***

Location is highly significant in the new multinomial logistic regression model that is used to predict the response class Change with the response class keep as the reference level. Size 300 or more is also significant. The reference level for size is less than 51 athletes.

$$\log \frac{P_{change}}{P_{keep}} = 1.001 - 0.160x_1 - 0.588x_2 - 0.168x_3 - 0.588x_4 - 0.064x_5 - 0.068x_6 - 0.494x_7 - 0.889x_8 - 1.126x_9 - 1.427x_{10} \quad (25)$$

where $x_1 = \text{Size: 51-75}$, $x_2 = \text{Size: 76-100}$, $x_3 = \text{Size: 101-125}$, $x_4 = \text{Size: 126-150}$, $x_5 = \text{Size: 151-175}$, $x_6 = \text{Size: 175-200}$, $x_7 = \text{Size: 200-299}$, $x_8 = \text{Size: 300 or more}$, $x_9 = \text{Location:Mult-Sharing}$, and $x_{10} = \text{Location:Mult-NOT Sharing}$

The results from the new regression model that is trying to predict abstain, with keep as the reference level, is similar to the model that included all the variables. The only significant factor is the intercept term. This shows that it is going to be hard to predict when the response variable is abstain.

$$\log \frac{P_{abstain}}{P_{keep}} = -0.920 - 0.084x_1 - 0.571x_2 - 0.259x_3 - 0.264x_4 - 1.233x_5 - 0.775x_6 - 0.437x_7 - 0.029x_8 + 0.240x_9 - 0.614x_{10} \quad (26)$$

Table 17: Multinomial Logistic Regression: Abstain with respect to Keep (significant factors only)

<i>Variable</i>	<i>Estimate</i>	<i>Z-value</i>	<i>P-value</i>	<i>Significance</i>
Intercept	-0.920	-3.484	0.000	***
Size51-75	-0.084	-0.222	0.824	
Size76-100	-0.571	-1.206	0.228	
Size101-125	-0.259	-0.557	0.557	
Size126-150	-0.264	-0.514	0.607	
Size151-175	-1.233	-1.134	0.257	
Size175-200	-0.775	-0.952	0.341	
Size200-299	-0.437	-0.675	0.450	
Size300 or more	0.029	0.058	0.954	
LocationMult-Sharing	0.240	-0.532	0.595	
LocationMult-NOT Sharing	-0.614	-1.692	0.091	

where $x_1 = \text{Size: 51-75}$, $x_2 = \text{Size: 76-100}$, $x_3 = \text{Size: 101-125}$, $x_4 = \text{Size: 126-150}$, $x_5 = \text{Size: 151-175}$, $x_6 = \text{Size: 175-200}$, $x_7 = \text{Size: 200-299}$, $x_8 = \text{Size: 300 or more}$, $x_9 = \text{Location:Mult-Sharing}$, and $x_{10} = \text{Location:Mult-NOT Sharing}$

The test data set was applied to the new regression model. Equations 3, 4, and 5 were used to determine the probability of each class for each observation. The class with the highest probability was chosen for the output. The confusion matrix shows the results. The test data set had 215 observations. 149 of the test observations were classified correctly. Notice that multinomial logistic regression model did better with fewer variables. The model using only significant factors will be used to compare with the other classifiers.

Table 18: Confusion Matrix: Multinomial Logistic Regression (Sign. Factors)

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	38	25	6	69
	Change	25	111	10	146
	Abstain	0	0	0	0
Total		63	136	16	

The accuracy for the multinomial logistic regression model was 69.40%. When looking at accuracy, it appears like the model is doing a decent job. The kappa statistic is 0.3556. This means that the model had low agreement.

Table 19: Overall Statistics: Multinomial Logistic Regression

Accuracy	0.6930
95 Percent CI	(0.6267,0.7540)
No Information Rate	0.6326
P-Value [Acc>NIR]	0.0372
Kappa	0.3556
Mnmemars Test P-Value	0.0011

Table 20 can be used to see how the model did for each class. As predicted, the model did very poorly with predicting the response variable abstain.

Table 20: Statistics by Class: Multinomial Logistic Regression (Sign. Factors)

<i>Class</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Sensitivity	0.6032	0.8162	0.0000
Specificity	0.7961	0.5570	1.0000
Pos Pred Value	0.5507	0.7603	NaN
Neg Pred Value	0.8288	0.6377	0.9256
Prevalence	0.2930	0.6326	0.0744
Detection Rate	0.1767	0.5163	0.0000
Detection Prevalence	0.3209	0.6791	0.0000
Balanced Accuracy	0.6996	0.6866	0.5000

4.4 Neural Networks

The neural network was run on the training data set. One hidden layer with three nodes was chosen for the model. After the model was created, the test data was used to see how the model was performing. There were 152 observations that were correctly classified. This model had an accuracy of 70.70%.

Table 21: Confusion Matrix: Neural Networks

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	31	15	5	51
	Change	32	121	11	164
	Abstain	0	0	0	0
Total		63	136	16	

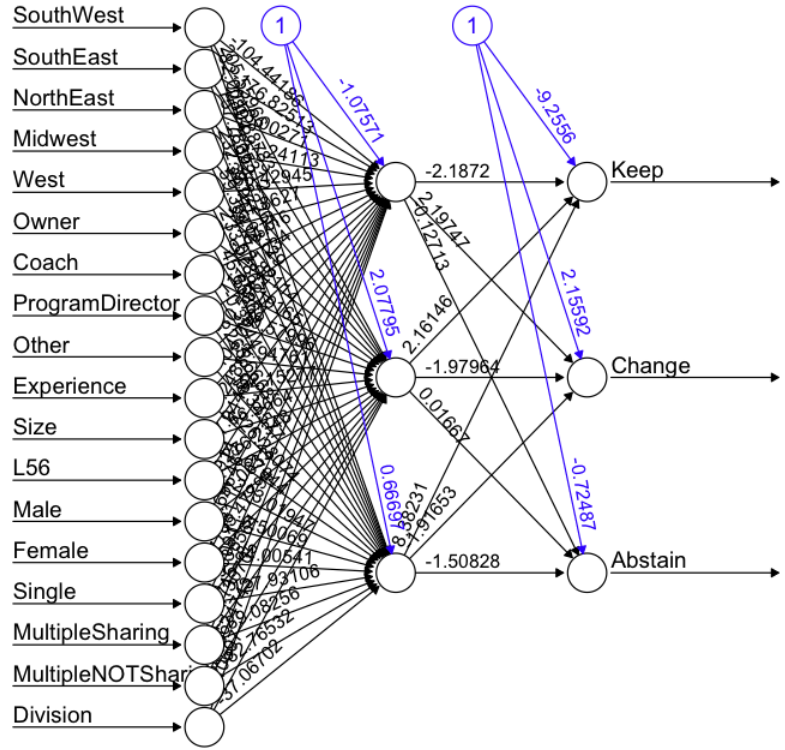


Figure 5: Neural Network

This is a visual representation of the neural network model. Each line contains the weight of that variable. The blue lines are the bias.

The accuracy for the neural network classifier was 70.70%. When looking at accuracy, it appears like the model is doing a decent job. The kappa statistic is 0.3459. This means that the model had low agreement. Table 22 can be used to examine the performance of the model for each class of the dependent variable.

Table 22: Overall Statistics: Neural Network

Accuracy	0.7070
95 Percent CI	(0.6421,0.7669)
No Information Rate	0.6326
P-Value [Acc>NIR]	0.0132
Kappa	0.3459
Mnnemars Test P-Value	0.0001

Table 23: Statistics by Class: Neural Network

<i>Class</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Sensitivity	0.4921	0.8897	0.0000
Specificity	0.8684	0.4557	1.0000
Pos Pred Value	0.6078	0.7378	NaN
Neg Pred Value	0.8049	0.7059	0.9256
Prevalence	0.2930	0.6326	0.0744
Detection Rate	0.1442	0.5628	0.000
Detection Prevalence	0.2372	0.7628	0.0000
Balanced Accuracy	0.6802	0.6727	0.5000

4.5 Support Vector Machine

The support vector machine was ran on the training data set. The kernel was set to "radial". The tune.svm function was ran to determine the best value to set the gamma and cost parameters. The gamma parameter was set to 0.0001. The cost parameter was set to 301. The support vector model was used on the test data set. The support vector machine was able to predict 149 instances correctly out of the 215 test data set.

Table 24: Confusion Matrix: SVM

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	38	25	6	69
	Change	25	111	10	146
	Abstain	0	0	0	0
Total		63	136	16	

This model has an accuracy of 69.30 percent. The kappa statistic for the support vector machine mode was 0.3556. Table 25 examines how well the support vector machine did for each class.

Table 25: Overall Statistics: SVM

Accuracy	0.6930
95 Percent CI	(0.6267,0.7540)
No Information Rate	0.6326
P-Value [Acc>NIR]	0.0372
Kappa	0.3556
Mnnemars Test P-Value	0.0011

Table 26: Statistics by Class: SVM

<i>Class</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Sensitivity	0.6032	0.8162	0.0000
Specificity	0.7961	0.5570	1.0000
Pos Pred Value	0.5507	0.7603	NaN
Neg Pred Value	0.8288	0.6377	0.9256
Prevalence	0.2930	0.6326	0.0744
Detection Rate	0.1767	0.5163	0.0000
Detection Prevalence	0.3209	0.6791	0.0000
Balanced Accuracy	0.6996	0.6866	0.5000

4.6 Decision Trees

4.6.1 rpart

The rpart decision tree used the predictor variables location, region, and size. The model was able to correctly predict 153 out of the 215 test observations. This model correctly predicted the correct class category 71.16% of the time.

Figure 6 is a visual representation of the decision tree that was made using the rpart function. Based on this plot, it is very easy to classify new observations.

The first split in the decision tree depends on the variable location. If the location variable is single then the instance will be classified as change. This means that the model predicts that the member will want to change the program definition if they belong to a program with only one location. If the location is multiple-sharing or multiple-not-sharing more information is needed to classify the data.

The next attribute that the decision tree uses to classify the data is region. If the variable region is Southeast, Southwest, Northeast, or Midwest the data is classified as keep. This means if a member's location is multiple-sharing or multiple-not-sharing and their region is Southeast, Southwest, Northeast, or Midwest then the model predicts that the member will choose to keep the program definition as it currently is.

If the member's location is multiple-sharing or multiple-not-sharing and the member belongs to the West region, then size is used to classify the data. If the size is the largest category possible (over 300 athletes), then the data will be classified as abstain. If the size is any other category, the data will be classified as change.

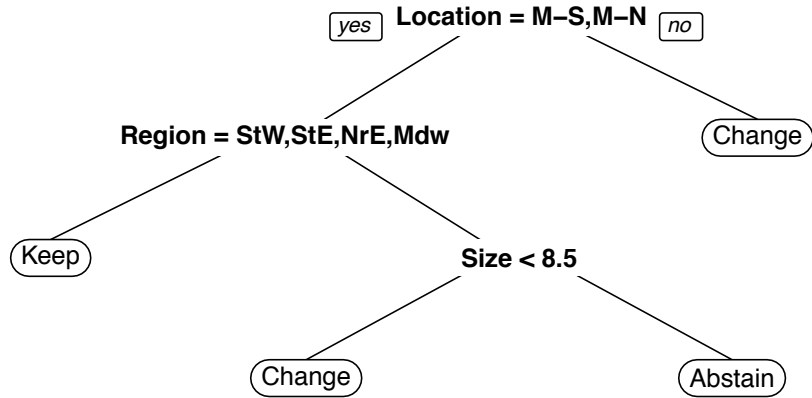


Figure 6: Decision Tree: rpart

Table 27: Confusion Matrix: rpart

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	38	20	5	63
	Change	25	115	11	151
	Abstain	0	1	0	1
Total		63	136	16	

From the confusion matrix, 153 observations were correctly classified. This model incorrectly predicted one observation as abstain. This is the only model that predicted any observations as abstain.

The accuracy for the rpart classification model was 71.16%. The kappa statistic was 0.3858. The accuracy seems relatively high but the kappa statistic show that this model has low agreement. Table 29 shows how the rpart classifier is doing for each response class.

Table 28: Overall Statistics: rpart

Accuracy	0.7116
95 Percent CI	(0.6461,0.7712)
No Information Rate	0.6326
P-Value [Acc>NIR]	0.0090
Kappa	0.3858
Mnmemars Test P-Value	0.0031

Table 29: Statistics by Class: rpart

<i>Class</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Sensitivity	0.6032	0.8456	0.0000
Specificity	0.8355	0.5443	0.9950
Pos Pred Value	0.6032	0.7616	0.0000
Neg Pred Value	0.8356	0.6719	0.9252
Prevalence	0.2930	0.6326	0.0744
Detection Rate	0.1767	0.5349	0.000
Detection Prevalence	0.2930	0.7023	0.0047
Balanced Accuracy	0.7194	0.6949	0.4975

4.6.2 ctree

The decision tree created using the `ctree` function is very similar to the decision tree created by `rpart`. The decision tree created by the `ctree` function is simpler and outperforms the tree created by `rpart`. The first split in the decision tree is made by location. Single location goes directly to being classified as change. When the location is multiple (multiple-sharing and multiple-not-sharing) then the variable region is taken into account. If the location is multiple and the region is Southeast, Southwest, Northeast, or Midwest, then the data is classified as keep. If the location is multiple and the region is West, then the data is classified as change.

The output below shows the decision tree in word form. Figure 7 is a visual plot of the decision tree created by the `ctree` function.

Conditional inference tree with 3 terminal nodes

Response: ProDef

Inputs: Region, Role, Experience, Size, Division, Location, L56, Gender

Number of observations: 864

- 1) Location == {Single}; criterion = 1, statistic = 95.115
- 2)* weights = 630
- 1) Location == {Mult-Share, Mult-NOTShare}
- 3) Region == {West}; criterion = 0.975, statistic = 23.149
- 4)* weights = 23
- 3) Region == {SouthWest, SouthEast, NorthEast, Midwest}
- 5)* weights = 211

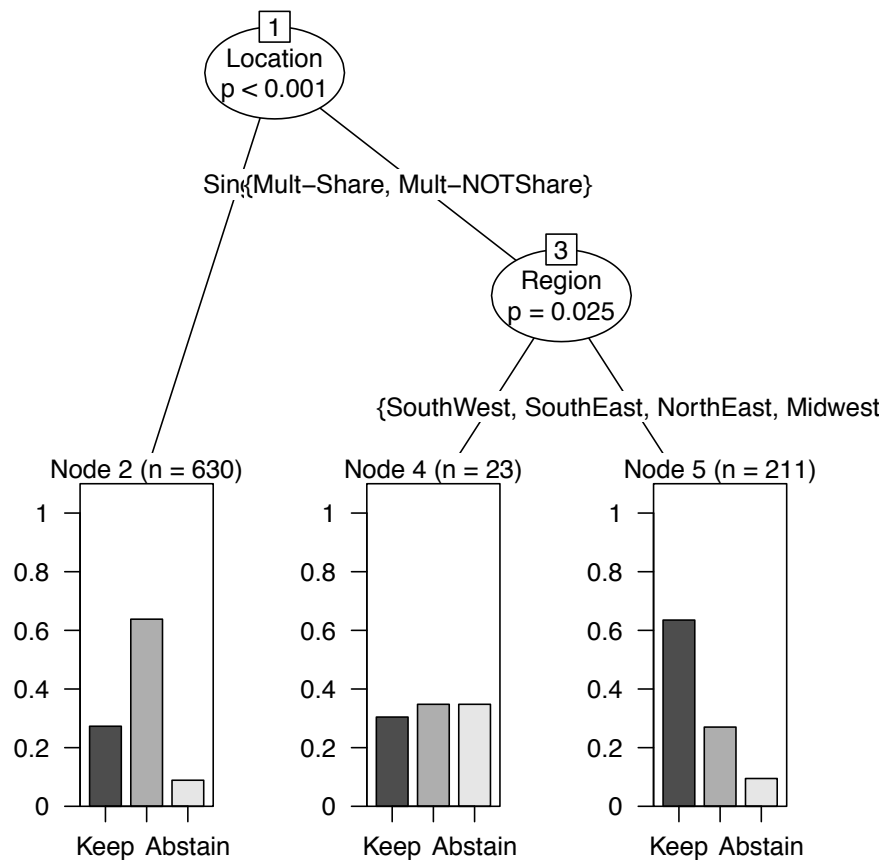


Figure 7: Decision Tree: ctree

It is hard to understand how each observation is classified from Figure 7. The data was analyzed to better understand this plot. The leaf on the left means the observation was classified as change. The leaf in the middle also represents the observations being classified as change. The leaf on the right classifies the observations as keep.

Table 30: Confusion Matrix: ctree

		Actual			Total
		Keep	Change	Abstain	
Predicted	Keep	38	20	5	63
	Change	25	116	11	152
	Abstain	0	0	0	0
Total		63	136	16	

The accuracy for this decision tree was 71.63 percent. 154 observations were correctly classified. No observations were classified as abstain. The kappa statistic for this model was 0.3924. Table 32 can be used to determine how well the model was doing for each class.

Table 31: Overall Statistics: ctree

Accuracy	0.7163
95 Percent CI	(0.6510,0.7755)
No Information Rate	0.6326
P-Value [Acc>NIR]	0.0060
Kappa	0.3924
Mnnemars Test P-Value	0.0001

Table 32: Statistics by Class:party

<i>Class</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Sensitivity	0.6032	0.8529	0.0000
Specificity	0.8355	0.5443	1.0000
Pos Pred Value	0.6032	0.7632	NaN
Neg Pred Value	0.8356	0.6825	0.9256
Prevalence	0.2930	0.6326	0.0744
Detection Rate	0.1767	0.5395	0.000
Detection Prevalence	0.2930	0.7070	0.0000
Balanced Accuracy	0.7194	0.6986	0.5000

5 Analysis

There are no major differences between any of the classification models. The results from all the models are extremely close. According to accuracy, the best model was created by the decision tree using the ctree package. The accuracies are extremely close. The data was split randomly into the training data set and the test data set. If this study was repeated with a different seed the results would vary. It is not a good idea to conclude that

ctree does the very best at predicting this data because the differences between the model's accuracies are so small.

Table 33: Accuracy

Multinomial Logistic Regression	69.30%
Neural Network	70.70%
Support Vector Machine	69.30%
rpart	71.16%
ctree	71.63%

The best model when looking at the kappa statistic is also the decision tree created by using the ctree function. The kappa statistic shows that there is low agreement in all classifiers. Once again, these results are so similar that we should not conclude that one model is outperforming the others.

Table 34: Kappa Statistic

Multinomial Logistic Regression	0.3556
Neural Network	0.3459
Support Vector Machine	0.3556
rpart	0.3858
ctree	0.3924

Multinomial logistic regression, support vector machines, rpart, and ctree all tie with highest sensitivity for the class Keep. This means that if an observation has a response variable of keep, there is a 60.32% chance that it will be classified correctly. Neural Network performs the best when looking at sensitivity for the response class change. If the observation has the response variable of change, there is an 88.97% chance that the observation will be classified correctly. All the classifiers had 0 for sensitivity for the response class abstain.

Table 35: Sensitivity

<i>Classifier</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Multinomial Logistic Regression	0.6032	0.8162	0.0000
Neural Network	0.4921	0.8897	0.0000
Support Vector Machine	0.6032	0.8162	0.0000
rpart	0.6032	0.8456	0.0000
ctree	0.6032	0.8529	0.0000

The neural network model has the highest value for specificity for the class keep. If an observation does not have the response variable keep, there is a 86.84% chance that it will not be classified as keep. The support vector machine model and the multinomial logistic

regression model performs the best when looking at specificity for the class change. If the observation does not have the response variable of change, there is a 55.70% chance that the observation will not be classified as change. The multinomial logistic regression classifier, neural network classifier, support vector machine classifier, and the ctree classifier all have a value of 1 for specificity for the class abstain. If the observation is not the class abstain, there is a 100% chance that it will not be classified as abstain. This occurs because these classifiers did not classify any observations as abstain.

Table 36: Specificity

<i>Classifier</i>	<i>Keep</i>	<i>Change</i>	<i>Abstain</i>
Multinomial Logistic Regression	0.7961	0.5570	1.000
Neural Network	0.8684	0.4557	1.000
Support Vector Machine	0.7961	0.5570	1.000
rpart	0.8355	0.5443	0.9950
ctree	0.8355	0.5443	1.000

It is interesting to note that the support vector machine classifier and the multinomial logistic regression classifier (with only the significant factors) have the same number of predicted observations for each class. It seems like there is only so much useable information in this data set and all the classifiers are using it.

All the observations that were classified differently by the classification models were examined. The test data set contain 215 observations. Out of those 215 observations, 188 observations were classified the same by all the classifiers. The classification models classified 27 observations differently. This means that the classifiers agreed on the predicted class for 87.44% of the observations.

The table below lists all the observations that were classified differently by multinomial logistic regression, neural network, support vector machine, ctree decision tree, or rpart decision tree. This table does not explore if the observation was classified correctly or incorrectly.

Table 37: Observations Classified Differently

<i>Obs</i>	<i>Region</i>	<i>Role</i>	<i>Exper</i>	<i>Size</i>	<i>Div</i>	<i>Loc</i>	<i>L56</i>	<i>Gender</i>	<i>ProDef</i>
2	NE	Coach	9	151-175	1	MS	1	F	Keep
140	W	ProD	13	101-125	2	MS	1	F	Change
258	NE	Coach	8	101-125	2	MNS	1	F	Keep
286	NE	ProD	8	151-175	1	MNS	1	F	Keep
318	W	Coach	14	151-175	1	MS	1	F	Abstain
385	M	Coach	7	101-125	2	MS	1	F	Change
425	NE	Owner	18	51-75	2	MS	1	M	Keep
480	NE	ProD	10	51-75	2	MS	1	M	Keep
597	NE	ProD	12	126-150	1	MNS	1	M	Keep
608	M	ProD	18	>300	1	MS	1	M	Keep
609	M	ProD	14	151-175	1	MNS	1	M	Change
626	NE	ProD	21	<50	2	MNS	0	F	Change
675	NE	Coach	3	126-150	1	MNS	1	F	Keep
690	SE	Coach	20	126-150	1	S	1	F	Keep
727	W	Coach	8	101-125	2	MNS	1	F	Change
865	NE	Owner	15	51-75	2	MS	0	F	Keep
874	NE	Owner	17	101-125	2	MNS	1	F	Change
907	SW	Owner	20	51-75	2	S	0	F	Keep
922	SW	Coach	16	200-299	1	S	0	F	Keep
975	NE	ProD	10	>300	1	MS	1	F	Change
1022	W	Coach	16	126-150	1	MNS	1	F	Change
1092	M	Other	15	126-150	1	S	0	F	Change
1102	NE	Coach	14	51-75	2	MNS	1	F	Change
1106	SW	Coach	14	51-75	2	MNS	0	F	Change
1137	W	ProD	5	200-299	1	MS	1	F	Change
1139	W	ProD	16	>300	1	MS	1	M	Change
1182	M	Owner	6	101-125	2	MNS	0	M	Keep

Table 38 shows how these observations were actually classified. From the table below it is clear that support vector machines and multinomial logistic regression predicted classified all observations exactly the same. The neural network model classified almost all these observations the opposite of how multinomial logistic regression and support vector machine classified the observations. The two decision trees predicted all the observations the same except observation 1139. The rpart decision tree predicted this value to be abstain. The ctree decision tree predicted the value to be change.

Table 38: How Observations were Classified

<i>Observation</i>	<i>MLR</i>	<i>SVM</i>	<i>NN</i>	<i>rpart</i>	<i>ctree</i>
2	Keep	Keep	Change	Keep	Keep
140	Keep	Keep	Change	Change	Change
258	Keep	Keep	Change	Keep	Keep
286	Keep	Keep	Change	Keep	Keep
318	Keep	Keep	Change	Change	Change
385	Keep	Keep	Change	Keep	Keep
425	Keep	Keep	Change	Keep	Keep
480	Keep	Keep	Change	Keep	Keep
597	Keep	Keep	Change	Keep	Keep
608	Keep	Keep	Change	Keep	Keep
609	Keep	Keep	Change	Keep	Keep
626	Keep	Keep	Change	Keep	Keep
675	Keep	Keep	Change	Keep	Keep
690	Change	Change	Keep	Change	Change
727	Keep	Keep	Change	Change	Change
865	Keep	Keep	Change	Keep	Keep
874	Keep	Keep	Change	Keep	Keep
907	Change	Change	Keep	Change	Change
922	Change	Change	Keep	Change	Change
975	Keep	Keep	Change	Keep	Keep
1022	Keep	Keep	Keep	Change	Change
1092	Change	Change	Keep	Change	Change
1102	Keep	Keep	Change	Keep	Keep
1106	Keep	Keep	Change	Keep	Keep
1137	Keep	Keep	Change	Change	Change
1139	Keep	Keep	Change	Abstain	Change
1182	Keep	Keep	Change	Keep	Keep

6 Conclusion

Data was collected from USASF members and analyzed to see if their opinions on proposed rule changes could be predicted using demographic variables. The proposed rule change that was examined in this study was program definition. Program definition had three classes: Keep, Change, and Abstain. Multinomial logistic regression, support vector machines, neural network and decision trees were used to classify the data. Based on eight demographic variables the response variable was correctly classified about 70% of the time.

Multinomial logistic regression, support vector machine, neural network, and decision trees all classified the data very similarly. No classifier is outperforming the other classifiers. The ctree decision tree had the best accuracy and kappa statistic, but only by a very small

margin.

The multinomial logistic regression model (with only significant variables) and the support vector machine classified all the observations exactly the same. It seems like this data set only had a limited amount of predictive information and it was used by all the classifiers. This study should be rerun multiple times using a different split of the data to see if it yields the same results.

The most important variables used to predict the member's opinions on the program definition was location, followed by region and size. In the multinomial logistic regression model and the decision trees models, these were the only variables that were even used to predict the response class.

References

- [1] Clark, A. (2016, August 30) Personal interview.
- [2] Brenner, H., & Gefeller, O. (1997). Variation of sensitivity, specificity, likelihood ratios and predictive values with disease prevalence. *Statistics in medicine*, 16(9), 981-991.
- [3] Demuth, H. B., Beale, M. H., De Jess, O., & Hagan, M. T. (2014). *Neural network design*. Martin Hagan.
- [4] Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5), 352-359.
- [5] Fagerland, M. W., Hosmer, D. W. & Bofin, A. M. (2008). Multinomial-goodness-of-fit tests for logistic regression models. *Statistics in medicine*, 27(21), 4238-4253.
- [6] Fu, C. Y. (2004). Combining loglinear model with classification and regression tree (CART): an application to birth data. *Computational Statistics & Data Analysis*, 45(4), 865-874.
- [7] Gethin L. Thomas, T. Coles and Mark R. Wilson (2016): Exploring mini rugby union coaches' perceptions of competitive activities, *Sports Coaching Review*, DOI:[10.1080/21640629.2016.1244425](https://doi.org/10.1080/21640629.2016.1244425)
- [8] Günther, F., & Fritsch, S. (2010). neuralnet: Training of neural networks. *The R journal*, 2(1), 30-38.
- [9] Greco, J. (2015). High School Athletics: Coaches' Opinions on Performance Enhancing Drugs. *College of Saint Mary*, 0-181.
- [10] Höffgen, K. U. (1993). Computational limitations on training sigmoid neural networks. *Information Processing Letters*, 46(6), 269-274.
- [11] Hothorn, T., Hornik, K., Strobl, C., & Zeileis, A. (2010). Party: A laboratory for recursive partytioning.

- [12] Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3), 651-674.
- [13] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.
- [14] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2), 415-425.
- [15] Karatzoglou, Alexandros and Meyer, David and Hornik, Kurt (2005) Support Vector Machines in R. *Research Report Series / Department of Statistics and Mathematics*, 21. Department of Statistics and Mathematics, WU Vienna University of Economics and Business, Vienna.
- [16] Kecman, V. (2005). Support vector machines—an introduction. *Support vector machines: theory and applications*, 605-605.
- [17] Kuhn, M. (2008, November). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5).
- [18] Lalkhen, A. G., & McCluskey, A. (2008). Clinical tests: sensitivity and specificity. *Continuing education in anaesthesia, critical care & pain*, 8(6), 221-223.
- [19] Lantz, B. (2015, August). Divide and Conquer- Classification Using Decision Trees and Rules. *Packtpub*, 1-9.
- [20] Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1), 14-23.
- [21] McFadden, D., Tye, W. B., & Train, K. (1977). An application of diagnostic tests for the independence from irrelevant alternatives property of the multinomial logit model. Institute of Transportation Studies, University of California.
- [22] Milgram, J., Cheriet, M., & Sabourin, R. (2006, October). “One against one” or “one against all”: Which one is better for handwriting recognition with SVMs?. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

- [23] Meyer, D., & Wien, F. T. (2015). Support vector machines. *The Interface to libsvm in package e1071*.
- [24] Moston, S., Terry, E., and Skinner, J. (2014). Athletes' and coaches' perceptions of deterrents to performance enhancing drug use. *International Journal of Sport Policy and politics*.
- [25] Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W. (1996). *Applied linear statistical models* (Vol. 4, p. 318). Chicago: Irwin.
- [26] Odom, M. D., & Sharda, R. (1990, June). A neural network model for bankruptcy prediction. *Neural Networks, 1990., 1990 IJCNN International Joint Conference* 163-168.
- [27] Parikh, R., Mathai, A., Parikh, S., Chandra Sekhar, G., & Thomas, R. (2008). Understanding and using sensitivity, specificity and predictive values. *Indian Journal of Ophthalmology*, 56(1), 45-50.
- [28] Rossiter, D. G. (2004). Technical Note: Statistical methods for accuracy assesment of classified thematic maps. Enschede (NL): *International Institute for Geo-information Science & Earth Observation*, 25(92), 107.
- [29] Shanker, M., Hu, M. Y., & Hung, M. S. (1996). Effect of data standardization on neural network training. *Omega*, 24(4), 385-397.
- [30] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.
- [31] Strickland, J. (2015, May 13). Neural Networks using R. *Business Intelligence Analytics*.
- [32] Suh, S.C. (2012) *Practical Applications of Data Mining*. Sundbury, MA: Jones and Bartlett Learning Canada.
- [33] Therneau, T. M., & Atkinson, E. J. (1997). *An introduction to recursive partitioning using the RPART routines* (Vol. 61, p. 452). Mayo Foundation: Technical report.
- [34] University of Nebraska-Lincoln. Cohen's Kappa [Table]. In *Bivariate Statistics Hand-Computation Cache*. Lincoln, NE.

- [35] U.S. All Star Federation. (2017). Retrieved from the U.S.All Star Federation website: usasf.net
- [36] Vanschoenwinkel, B., & Manderick, B. (2005). Appropriate kernel functions for support vector machine learning with sequences of symbolic data. *In Deterministic and statistical methods in machine learning* 256-280 Springer Berlin Heidelberg.
- [37] Wooldridge, J. M. (2015). *Introductory econometrics: A modern approach*. Nelson Education.
- [38] Yang, C., Pan, S., Mahmud, J., Yang, H., & Srinivasan, P. (2015). Using Personal Traits For Brand Preference Prediction. *Association for Computational Linguistics*, 86-96.

7 Appendix: R Code

```
#####  
# PART 1: R code for MULTINOMIAL LOGISTIC REGRESSION in Section 4.2      #  
#                                                                           #  
#####  
#clear workspace  
rm(list=ls())  
  
# read in data  
mydata<- read.csv("~/Downloads/usasfProDefNN.csv",header=TRUE)  
  
# clean data  
# delete missing values  
# change categorical variables from integer to factor  
mydata <- na.omit(mydata)  
mydata=mydata[!mydata$Size =="10",]  
mydata=mydata[!mydata$Division =="3",]  
mydata=mydata[!mydata$Location =="4",]  
mydata=mydata[!mydata$Region =="6",]  
mydata=mydata[!mydata$Gender =="3",]  
mydata$Size=factor(mydata$Size,  
                    levels=c(1:9),  
                    labels=c("<50", "51-75", "76-100", "101-125", "126-150",  
                              "151-175", "175-200", "200-299", "300 or more"))  
mydata=mydata[!mydata$Division =="3",]  
mydata$Division=factor(mydata$Division,  
                        levels=c(1:2),  
                        labels=c("D1", "D2"))  
mydata=mydata[!mydata$Location =="4",]  
mydata$Location=factor(mydata$Location,  
                        levels=c(1:3),
```

```

        labels=c("Single Location","Multiple-Sharing",
                 "Multiple-NOT Sharing"))
mydata$Role=factor(mydata$Role,
                   levels=c(1:4),
                   labels=c("Owner","Coach","Program Director","Other"))
mydata$Gender=factor(mydata$Gender,
                     levels=c(0:1),
                     labels=c("Male","Female"))
mydata$L56=factor(mydata$L56,
                  levels=c(0:1),
                  labels=c("No Level 5/6","Level 5/6"))
mydata$Region=factor(mydata$Region,
                     levels=c(1:5),
                     labels=c("Southwest","Southeast","Northeast","Midwest",
                               "West"))

set.seed(34123)
mydata=mydata[!mydata$ProDef == 4,]
mydata$ProDef=factor(mydata$ProDef,
                     levels=c(1:3),
                     labels=c("Keep","Change", "Abstain"))

#Load R Packages
library("VGAM")
library(nnet)

# divide data set into testing and training data sets
# set seed so they are the same for each classifier
set.seed(34123)
indexes = sample(1:nrow(mydata), size=0.2*nrow(mydata))
test = mydata[indexes,] #dim(test) # 215 9
train = mydata[-indexes,] #dim(train) # 864 9
nnttrain<-train

```

```

#fit regression model with all variables (not division)
fitProDef=vglm(ProDef~ Region + Role + Experience + Size
              + Location + Gender + L56, data=train,
              family=multinomial(refLevel="Keep"))
summary(fitProDef)
fitProDef=multinom(ProDef~ Region + Role + Experience + Size
                  + Location + Gender + L56, data=train, family=multinomial(refLevel="Keep"))
summary(fitProDef)

#predict test data using regression model
prediction <- predict(fitProDef, test, type="prob")
prediction <- max.col(prediction)
cm<- table(prediction,test$ProDef)

#fit regression model with only significant factors
fitProDef=vglm(ProDef~Size+Location, data=train,
              family=multinomial(refLevel="Keep"))
summary(fitProDef)
fitProDef=multinom(ProDef~Size+Location,
                  data=train, family=multinomial(refLevel="Keep"))
summary(fitProDef)

#predict outcome for test data set using regression model
predictionMLR <- predict(fitProDef, test, type="prob")
predictionMLR <- max.col(predictionMLR)
cm <- table(predictionMLR, test$ProDef)
print(cm)

#calculated probability of first observation of test data
#change
a=1.178501-0.1665722+0.3199648+0-0.16956748

```

```

#abstain
b=-1.184458+0.2224649+0.3568687+0+0.01197002
#keep
k=1/(1+exp(a)+exp(b))
k
#change
c=exp(a)/(1+exp(a)+exp(b))
c
#abstain
ab=exp(b)/(1+exp(a)+exp(b))
ab

#####
#           PART 2: R code for NEURAL NETWORK in Section 4.3           #
#                                                                                   #
#####
#clear Data
rm(list=ls())

#Load R package
library(neuralnet)
library(nnet)

#read in data
mydata<- read.csv("~/Downloads/usasfProDefEX.csv",header=TRUE)

#clean data
#delete missing values
mydata <- na.omit(mydata)
mydata=mydata[!mydata$Size == "10",]
mydata=mydata[!mydata$Division == "3",]

```

```

mydata=mydata[!mydata$Location == "4",]
mydata=mydata[!mydata$Region == "6",]
mydata=mydata[!mydata$Gender == "3",]
mydata2=mydata[!mydata$ProDef == 4,]
mydata2$ExperienceD<-NULL

#split data into training and testing data set
set.seed(34123)

indexes = sample(1:nrow(mydata2), size=0.2*nrow(mydata2))
test = mydata2[indexes,] #dim(test) # 215 9
nntest<-test

train = mydata2[-indexes,] #dim(train) # 864 9
nntrain<-train

#transform variables from train data so they are in correct format
#change other variables from integers to nominal
nnet_train_new<- cbind(nntrain, class.ind(train$ProDef))
names(nnet_train_new)[10] <- 'Keep'
names(nnet_train_new)[11] <- 'Change'
names(nnet_train_new)[12] <- 'Abstain'
nnet_train_new<- cbind(nnet_train_new, class.ind(train$Gender))
nnet_train_new$"3"<-NULL
names(nnet_train_new)[13] <- 'Male'
names(nnet_train_new)[14] <- 'Female'
nnet_train_new<- cbind(nnet_train_new, class.ind(train$Region))
nnet_train_new$"6"<-NULL
names(nnet_train_new)[15] <- 'SouthWest'
names(nnet_train_new)[16] <- 'SouthEast'
names(nnet_train_new)[17] <- 'NorthEast'
names(nnet_train_new)[18] <- 'Midwest'
names(nnet_train_new)[19] <- 'West'
nnet_train_new<- cbind(nnet_train_new, class.ind(train$Role))

```

```

nnet_train_new$"5"<-NULL
names(nnet_train_new)[20] <- 'Owner'
names(nnet_train_new)[21] <- 'Coach'
names(nnet_train_new)[22] <- 'ProgramDirector'
names(nnet_train_new)[23] <- 'Other'
nnet_train_new<- cbind(nnet_train_new, class.ind(train$Location))
nnet_train_new$"4"<-NULL
names(nnet_train_new)[24] <- 'Single'
names(nnet_train_new)[25] <- 'MultipleSharing'
names(nnet_train_new)[26] <- 'MultipleNOTSharing'
nnet_test_new<- cbind(nntest, class.ind(test$ProDef))
names(nnet_test_new)[10] <- 'Keep'
names(nnet_test_new)[11] <- 'Change'
names(nnet_test_new)[12] <- 'Abstain'

#transform variables from test so they are in correct format
#change other variables from integers to nominal
nnet_test_new<- cbind(nnet_test_new, class.ind(test$Gender))
nnet_test_new$"3"<-NULL
names(nnet_test_new)[13] <- 'Male'
names(nnet_test_new)[14] <- 'Female'
nnet_test_new<- cbind(nnet_test_new, class.ind(test$Region))
nnet_test_new$"6"<-NULL
names(nnet_test_new)[15] <- 'SouthWest'
names(nnet_test_new)[16] <- 'SouthEast'
names(nnet_test_new)[17] <- 'NorthEast'
names(nnet_test_new)[18] <- 'Midwest'
names(nnet_test_new)[19] <- 'West'
nnet_test_new<- cbind(nnet_test_new, class.ind(test$Role))
nnet_test_new$"5"<-NULL
names(nnet_test_new)[20] <- 'Owner'
names(nnet_test_new)[21] <- 'Coach'

```



```

names(nnet_test_new)[22] <- 'ProgramDirector'
names(nnet_test_new)[23] <- 'Other'
nnet_test_new<- cbind(nnet_test_new, class.ind(test$Location))
nnet_test_new$"4"<-NULL
names(nnet_test_new)[24] <- 'Single'
names(nnet_test_new)[25] <- 'MultipleSharing'
names(nnet_test_new)[26] <- 'MultipleNOTSharing'

#delete old variable names that are no longer being used
nnet_test_new$"ProDef"<-NULL
nnet_test_new$"Role"<-NULL
nnet_test_new$"Region"<-NULL
nnet_test_new$"Location"<-NULL
nnet_test_new$"Gender"<-NULL
nnet_train_new$"ProDef"<-NULL
nnet_train_new$"Role"<-NULL
nnet_train_new$"Region"<-NULL
nnet_train_new$"Location"<-NULL
nnet_train_new$"Gender"<-NULL

#set seed
set.seed(34123)

#fit model
nn <- neuralnet(Keep + Change + Abstain ~ SouthWest + SouthEast +
                NorthEast + Midwest + West + Owner + Coach +
                ProgramDirector + Other + Experience + Size
                + L56 + Male + Female + Single + MultipleSharing
                + MultipleNOTSharing + Division,
                data=nnet_train_new, err.fct="ce",
                linear.output = FALSE,hidden=c(3))

```

```

#use the model nn to predict outcome of test data
set.seed(34123)
predict <- compute(
  nn, nnet_test_new[,c("SouthWest",
    "SouthEast", "NorthEast", "Midwest",
    "West", "Owner", "Coach", "ProgramDirector",
    "Other", "Experience", "Size", "L56",
    "Male", "Female", "Single", "MultipleSharing",
    "MultipleNOTSharing", "Division")]]$net.result
set.seed(34123)
predictionN<-max.col(predict)

#confusion matrix
library(caret)
confusionMatrix(predictionN, test$ProDef)

#####
# PART 3: R code for SUPPORT VECTOR MACHINE in Section 4.4 #
# #
#####
#clear workspace
rm(list=ls())

#read in data
mydata <- read.csv("~/Downloads/usasfProDefNN.csv",header=TRUE)

#clean data
#delete missing values

```

```

mydata <- na.omit(mydata)
mydata=mydata[!mydata$Size == "10",]
mydata=mydata[!mydata$Division == "3",]
mydata=mydata[!mydata$Location == "4",]
mydata=mydata[!mydata$Region == "6",]
mydata=mydata[!mydata$Gender == "3",]
mydata2=mydata[!mydata$ProDef== 4,]

#change categorical variables from integer to factor
mydata2$Region <- as.factor(mydata2$Region)
mydata2$Role <- as.factor(mydata2$Role)
mydata2$Location <- as.factor(mydata2$Location)
mydata2$Gender <- as.factor(mydata2$Gender)
mydata2$ProDef <- as.factor(mydata2$ProDef)

#set seed
set.seed(34123)

#Load R packages
library(e1071)
library(MASS)

#split data into training and testing
indexes = sample(1:nrow(mydata2), size=0.2*nrow(mydata2))
test = mydata2[indexes,] #dim(test) # 215 9
train = mydata2[-indexes,] #dim(train) # 864 9
nntrain<-train

#run SVM function
model<- svm(ProDef~ Region + Role + Experience + Size
            + Division + Location + Gender + L56,
            data=train, kernel="radial",

```

```

        type="C-classification", gamma=.0001, cost=301)
print(model)
summary(model)

#predict outcome using model and test data
predictionSV<- predict(model, test[1:8])
tab<- table(pred=predictionSV, true=test[,9])

#confusion matrix
library(caret)
confusionMatrix(prediction, test$ProDef)

#try to set parameters so model does the best
tune.svm(ProDef~.,data=train, gamma= seq(.0001,.1, by= .01),
         cost=seq(1,1000,by=100))

#####
#   PART 4: R code for DECISION TREE in Section 4.5   #
#                                                     #
#####

#####RPART
#clear data set
rm(list=ls())

#load data
mydata<- read.csv("~/Downloads/usasfProDefNN.csv",header=TRUE)

#clear data

```

```

mydata <- na.omit(mydata)
mydata=mydata[!mydata$Size == "10",]
mydata=mydata[!mydata$Division == "3",]
mydata=mydata[!mydata$Location == "4",]
mydata=mydata[!mydata$Region == "6",]
mydata=mydata[!mydata$Gender == "3",]
mydata2=mydata[!mydata$ProDef== 4,]

#change variables to categorical
mydata2$Region <- as.factor(mydata2$Region)
mydata2$Role <- as.factor(mydata2$Role)
mydata2$Location <- as.factor(mydata2$Location)
mydata2$Gender <- as.factor(mydata2$Gender)
mydata2$ProDef <- as.factor(mydata2$ProDef)
#label variables that are used in tree
levels(mydata2$Region) <- c('SouthWest', 'SouthEast',
                           'NorthEast', 'Midwest', 'West')
levels(mydata2$Location) <- c('Single', 'Mult-Share',
                              'Mult-NOTShare')
levels(mydata2$ProDef)<-c("Keep", "Change", "Abstain")

#set seed
set.seed(34123)

#load library
library(rpart)

#split data into train and test
indexes = sample(1:nrow(mydata2), size=0.2*nrow(mydata2))
test = mydata2[indexes,] #dim(test) # 215 9
train = mydata2[-indexes,] #dim(train) # 864 9
nntrain<-train

```

```

#fit model
fit<- rpart(ProDef~., data=train, method="class")

# predict response class of test data using rpart model
predictionsDT1<-predict(fit, test[,1:8], type="class")
table(predictionsDT1, test$ProDef)

#summary information on decision tree
printcp(fit) #this shows us that the only variables used in
              #tree construction was Location, Region, and Size

#confusion matrix
library(caret)
confusionMatrix(predictions, test$ProDef)

#plot decision tree
library(rpart.plot)
prp(fit)

#####CTREE
#load library
library(party)

#plot(train$ProDef)
fitP<- ctree(ProDef~., data=train)
fitP

#plot decision tree
plot(fitP)

# predict test data using ctree model

```

```
predictions2<-predict(fitP, test[,1:8])
table(predictions2, test$ProDef)
predictions

#confusion matrix
library(caret)
confusionMatrix(predictions2, test$ProDef)
```

April 27, 2017

Dr. Lucy Kerns , Principal Investigator
Ms. Aricka Gates, Co-investigator
Department of Mathematics & Statistics
UNIVERSITY

RE: HSRC PROTOCOL NUMBER: 184-2017
TITLE: Professional Members' Perceptions of Proposed Rule Changes in All Star
Cheerleading

Dear Dr. Kerns and Ms. Gates:

The Institutional Review Board has reviewed the abovementioned protocol and determined that it is exempt from full committee review based on a DHHS Category 3 exemption.

Any changes in your research activity should be promptly reported to the Institutional Review Board and may not be initiated without IRB approval except where necessary to eliminate hazard to human subjects. Any unanticipated problems involving risks to subjects should also be promptly reported to the IRB.

The IRB would like to extend its best wishes to you in the conduct of this study.

Sincerely,

Mr. Michael Hripko
Associate Vice President for Research
Authorized Institutional Official

MAH:cc

c: Dr. Angela Spalsbury, Chair
Department of Mathematics & Statistics

