Analysis of College Student Diets

Supah Hot Fiya Sizzle Sizzle Yum
Clayton Coulter, Katie Bullock, and Jake LaMotte

I. Introduction

“Food is the ingredient that binds us together.” Throughout our lives we have been brought together through food in some fashion, whether that is at a birthday party, a christmas dinner, or even thanksgiving. Food is something that we all have a connection to in some way. Each individual has a different relationship with food and is able to use food in a variety of ways. However, college students are known for having a distinct diet unlike any others. College students' diets are not necessarily determined on what “tastes good” or fuels their bodies but by the amount of money they have in their bank account. Within our dataset we were able to collect a variety of results on college students' relationship with food and how they viewed their own health.

Though we initially believed that our datasets would indicate a student’s weight, we learned that we saw better results when trying to determine which attributes were most predictive for determining whether an individual was male or female. In determining what attributes were most predictive we used several different models and techniques. We began with classification learning, stringToWordVector, and meta learning. Through each model we developed different percentages of accuracy in determining whether the student was male or female. In addition, we found many interesting results of how each gender views food and the differences between the two gender’s diets while in college.

I. Dataset

Our dataset came from Kaggle (https://www.kaggle.com/borapajo/food-choices) and required tedious cleaning in order for data to be ready for analysis within Weka. The dataset originally consisted of 60 attributes. The attributes data types were strings and numerical values. The string attributes have been coded into specific numerical values in order to categorize keywords from the free response questions.

Attributes:	Values:
1. GPA	Numerical, actual Grade Point Average.

(numerical)	
2. Gender	1. Female 2. Male
3. Breaky (Participants were shown a picture of cereal and donut and associated which food is breakfast food.)	1. Cereal 2. Donut
4. Calories_Chicken (guessing calories in chicken piadina)	1. 265 2. 430 3. 610 4. 720
5. Calories_Day (Importance of consuming calories per day)	1. I don't know how many calories I should consume 2. It is not at all important 3. It is moderately important.
6. Calories_Scone (Guessing calories in a scone)	1. 107 2. 315 3. 420 4. 980
7. Java (which two pictures are associated with coffee)	1. Creamy Frapuccino 2. Espresso
8. Comfort_Food (listing 3-5 foods that come to mind)	Free response
9. Comfort_Food_Reasons (Reasons that make you eat comfort food)	Free response
10. Comfort_Food_Reasons_Coded (reasons comfort food is eating in numerical values)	1. Stress 2. Boredom 3. depression/sadness 4. Hunger 5. Laziness

	6. Cold weather 7. Happiness 8. Watching tv 9. none
11. Ratatouille (How often do you cook?)	1. Every day 2. A couple of times a week 3. Whenever I can, but that is not very often 4. I only help a little during holidays 5. Never, Do not know way in kitchen
12. Cuisine (What type of cuisine did you eat growing up?)	1. American 2. Mexican, Spanish 3. Korean/Asian 4. Indian 5. American inspires international dishes 6. other
13. Diet_Current (describe you current diet)	Free response
14. Diet_Current_Coded (Words that describe current diet numerically)	1. Healthy/balanced/moderated 2. unhealthy/cheap/too much random 3. Same thing over and over 4. Unclear
15. Slurp (Picture do you associate with the word “drink”)	1. Orange 2. Soda
16. Eating_changes (describe eating changes since the moment you got into college?)	Free response
17. Eating_changes_coded (eating changes since the moment you got into college numerically)	1. Worse 2. Better 3. The same 4. unclear
18. Eating_changes_coded1	1. Eat faster 2. Bigger quantity

(eating changes since the moment you got into college numerically)	3. Worse quality 4. Same food 5. Healthier 6. Unclear 7. Drink coffee 8. Less food 9. More sweets 10. Timing 11. More carbs or snacking 12. Drink more water 13. More variety
19. Eating_out (frequency of eating out in a typical week)	1. Never 2. 1-2 3. 2-3 4. 3-5 5. Every day
20. Employment (do you work?)	1. Yes full time 2. Yes part time 3. No 4. other

21. Ethnic_food (How likely to eat ethnic food)	1. Very unlikely 2. Unlikely 3. Neutral 4. Likely 5. Very likely
22. Exercise (How often do you exercise in a regular week?)	1. Everyday 2. Twice or three times per week 3. Once a week 4. Sometimes 5. Never
23. Father_education	1. Less than high school 2. High school degree 3. Some college degree 4. College degree 5. Graduate degree
24. Father_profession (What is your father's profession?)	Free response

25. Fav_cuisine (What is your favorite cuisine?)	Free response
26. Fav_cuisine_coded (favorite cuisine numerically)	1. Italian/French/greek 2. Spanish/Mexican 3. Arabic/Turkish 4. Asian/Chinese/Thai/Nepal 5. American 6. African 7. Jamaican 8. Indian
27. Fav_food (Was your favorite food cooked at home or store bought?)	1. Cooked at home 2. Store bought 3. Both bought at store and cooked at home
28. Food_Childhood (What was your favorite childhood food?)	Free response
29. Potato sticks (Which pictures you associate with word fries?)	1. Mcdonald's fried 2. Home fries
30. Fruit_day (How likely to eat fruit in a regular day?)	1. Very unlikely 2. Unlikely 3. Neutral 4. Likely 5. Very likely
31. Grade_level	1. Freshman 2. Sophomore 3. Junior 4. Senior
32. Greek_food (How likely to eat Greek food when available?)	1. Very unlikely 2. Unlikely 3. Neutral 4. Likey 5. Very likely
33. Healthy_feel	Scale 1 - 10. 1 is strongly agree and 10 is strongly disagree

(How likely are you to agree with the following statement: “ i feel very healthy!”?)	
34. Healthy_meal (What is a healthy meal?)	Free response
35. Ideal_diet (describe your ideal diet?)	Free response
36. Ideal_diet_coded (Describe your ideal diet numerically)	1. Portion control 2. Adding veggies/eating healthier food/ adding fruit 3. Balance 4. Less sugar 5. Home cooked/organic 6. Current diet 7. More protein 8. Unclear
37. Income	1. Less than $15,000 2. 15,001 to $30,000 3. 3 - $30,001 to $50,000 4. 4 - $50,001 to $70,000 5. 5 - $70,001 to $100,000 6. 6 - higher than $100,000
38. Indian_food (How likely are you to eat Indian food when available?)	1. very unlikely 2. unlikely 3. neutral 4. likely 5. very likely
39. Italian_food (How likely are you to eat Italian food when available?)	1. very unlikely 2. unlikely 3. neutral 4. likely 5. very likely
40. Life_rewarding (How likely are you to agree with the	1 to 10 where 1 is strongly agree and 10 is strongly disagree

following statement: “I feel like is very rewarding”)	

41. Marital_status	1. Single 2. In a relationship 3. Cohabiting 4. Married 5. Divorced 6. Widowed
42. Meals_dinner_friend (What would you serve to a friend for dinner?)	Free response
43. Mothers_education	1. less than high school 2. high school degree 3. some college degree 4. college degree 5. graduate degree
44. Mothers_profession	Free response
45. Nutritional_check (Checking nutritional values frequency)	1. never 2. on certain products only 3. very rarely 4. on most products 5. on everything
46. On_off_campus (living situation)	1. On campus 2. Rent out of campus 3. Live with my parents and commute 4. Own my own house
47. Parents_cook (Approximately how many days a week did your parents cook?)	1. Almost everyday 2. 2-3 times a week 3. 1-2 times a week 4. on holidays only 5. never
48. Pay_meal_out (How much would you pay for a meal out?)	1. up to $5.00 2. $5.01 to $10.00 3. $10.01 to $20.00 4. $20.01 to $30.00 5. $30.01 to $40.00

	6. more than $40.01
49. Persian_food (How likely to eat Persian food when available?)	1. very unlikely 2. unlikely 3. neutral 4. likely 5. very likely
50. Self_perception_weight (Self perception of weight)	6. I dont think myself in these terms 5. Overweight 4. Slightly overweight 3. Just right 2. Very fit 1. slim
51. Food_in_a_bath (Which of the two pictures do you associate with the word soup?)	1. Veggie soup 2. Creamy soup
52. Sports (Do you do any sporting activity?)	1. Yes 2. no
53. Thai_food (How likely to eat Thai food when available?)	1. very unlikely 2. unlikely 3. neutral 4. likely 5. very likely
54. Tortilla_callories (guessing calories in a burrito sandwich from Chipotle?)	1. 580 2. 725 3. 940 4. 1165
55. Turkey_calories (Can you guess how many calories are in a Panera Bread Roasted Turkey and Avocado BLT)	1. 345 2. 500 3. 690 4. 850
56. Type_sports (What type of sports are you involved in?)	Free response
57. Veggies_day	1. very unlikely

(How likely to eat veggies in a day?)	2. unlikely 3. neutral 4. likely 5. very likely
58. Vitamins (Do you take any supplements or vitamins?)	1. Yes 2. No
59. Waffles_Calories (Guessing calories in waffle potato sandwiches?)	1. 575 2. 760 3. 900 4. 1315
60. Mass (What is your weight in pounds?)	Free response

III. Data Preparation

1. Data Collection

Our team collected our dataset from a website called Kaggle. We then downloaded the file and transferred it to a text editor to be able to edit it as an ARFF file. We found a few problems with our dataset once we transferred our over to an ARFF file. Our dataset contained various types of data including numerical data, nominal data, and string data. Interestingly, the dataset came with a code book that had the string data already coded into key words for analysis. However, we felt that we should use their code book for analysis while also using our own technique of stringToWordVector to compare our results in determining whose dataset was more predictive. We had to go through and delete any apostrophe's that were in the free response questions in order for weka not to see it as a separate value. We also had to rename numerous attributes due to the fact that some attributes had the same name. In addition, for all the data types that were strings we had to name those attributes as @attribute string. Once we named all the sections and data types within the ARFF file and deleted all the apostrophes, weka was able to upload the dataset.

2. Sparse Data/Discretization

Within our dataset we had a small quantity of missing values called “nan” which stood for, “not a number”. This could have been due to the fact that students were giving invalid answers to specific questions. Therefore, as a group we felt that it was best to delete every value that was listed as “nan”. We then used the ReplaceMissingValues filter within Weka. We kept the default setting for this filter in order to limit bias for replacing the values. By keeping the default setting this filter would take the mean and mode of each instance within the dataset to replace the missing values instead of just focusing on the class in order to build a mode. We felt like using this filter would produce the best results for our data.

In addition we used equal height discretization for every attribute. Depending on the attribute we would divide each attribute into a specific number of bins based on the amount of options a student was given within a question. For example, if a student was given only 4 options on what their favorite food was, we would then discretize this attribute into four bins based off of the class of gender.

3. Unbalanced Data

When going through our dataset we immediately utilized zeroR to see if we had any unbalanced data. The results gave us unbalanced data. Therefore, ZeroR produced 60.8% accuracy. In order to build a more accurate model we needed to even the distribution between male and female. In the beginning, there were 76 females and 49 males. We chose to use five different strategies for pre-processing our data to see which strategy would produce the best results. Each one of these filters was used on data that had no strings, and used the original code book of key words for the free responses that was provided by Kaggle.

a. Resample Filter

The first strategy we used to balance out our data was called resample. To perform this filter it is located in supervised learning underneath the instances within Weka. What this filter does is oversample the minority class and undersample the majority class. Once we apply this filter the females amount would drop bringing the total females to 62 and males at 62 with a total of 124 instances.

b. SpreadSubsample filter

In order to use this filter it was located within supervised learning underneath the instances within Weka. The SpreadSubSample was similar to

the resample filter. Similarly, this filter also undersamples the majority class. However, this filter allows you to choose the maximum spread between the minority and majority class. Within our settings we chose to set the distribution spread to one to get a perfect 1:1 ratio between male and female. Once adding this filter the total amount of Females were 49 and the males were 49.

c. ClassBalancer

This filter is located in the same area as the resample and SpreadSubsample and resample filter. The ClassBalancer filter assigns each instance weight so that each class instance weight will be the exact same and the total sum of the instance weight will remain unchanged. Therefore, out of the 125 instances the class attribute was 62.5 male and 62.5 female. However, in order to use this filter you must locate the FilteredClassifier within the classify tab. This allows you to use this filter for algorithms that deal with rules.

d. SMOTE

When applying the SMOTE we kept the class value as 0 because weka automatically detects the minority class. Within the settings we changed the percentage to 80% which will then multiply the minority class by 80%. Therefore, giving us the results of having 88 males and 76 females to build a model in comparison to the original amount of 76 females and 49 males.

e. CostSensitiveClassifier

This classifier is able to add a penalty for when an attribute is misclassified. This is located in the classify tab. Within this classifier we first adjusted the class matrix to the size of 2 since there class attribute values is male and female. We then went into the cost matrix editor and changed the penalty to 5.0 for males when they are misclassified. Based on our results, we decided to change the cost matrix to 2.0. This lowered the amount of misclassification for males.

4. Text Mining

Our dataset contained several open-ended questions which he had to convert to string attributes within the .arff file in order to make it compatible with Weka. These string attributes are classified using “@attribute attribute_name string”. After loading the

dataset with strings into Weka, we used the stringToWordVector filter to return multiple numeric attributes. This means that Weka sorted through the string attributes and created new numeric attributes from words that were used across the open-ended answers. The use of these words is indicated within each instance by either a 0 if it is not present or a 1 if it is. We wanted to narrow down the amount of words we were given to work with and chose to filter stringToWordVector to only words that appeared five or more times. This function in Weka produces the collection of words as unique attributes for use in classifying.

IV. Data Analysis

1. Classification Learning

Through testing our data we used 10-fold cross validation for all algorithms due to the size of our dataset since we only had 125 instances. 10-fold cross validation holds out 10% of our data to use as test data and builds a model on the rest of the 90% and repeats that process 10 times until it covers all the data.

A. Zero - Rule

The Zero-Rule algorithm that is also known as ZeroR, is an algorithm that is used as a baseline for analysis. This algorithm takes zero input attributes and only predicts based on the majority of the class. This algorithm requires minimal computation and is used to see whether you have unbalanced data or not. The way this algorithm works is that it ignores all other attributes but the class attribute. It then uses the only the class attribute in order to build a model and uses the majority class for prediction. We needed to use this algorithm because having more females than males within the dataset was causing our results to have been more predictive for females since it was the majority. We used this algorithm for each one of the different pre-processing techniques to see if our data was balanced in comparison to our original dataset.

Table 1. Lists the filter and the accuracy of ZeroR as a percentage for the number of instances within the data set. (Without string data)

Filter	ZeroR accuracy	Majority Class	Number of Instances
None / original	60.8%	Female	125

Resample	48.3871%	Equal	124
SpreadSubsample	48.9796%	Equal	98
SMOTE	53.6585%	Male	164
ClassBalancer	48.0129%	Equal	125
CostSensitiveClassifier (2.0 penalty)	39.2%	Female	125
CostSensitiveClassifier (5.0 penalty)	39.2%	Female	125

This algorithm was helpful in determining which filter was providing us with the most balanced data. Through the use of ZeroR we determined that the SpreadSubsample filter was the best for making our data well balanced. This filter undersampled the majority class of females to 49 and males to 49. Giving us a close to even split between male and female at 48.9796%.

B. One-Rule

One-Rule which is also known as oneR is another baseline algorithm similar to ZeroR. The reason this algorithm is called OneR is because it learns a set of rules, and is based on only one input attribute. OneR is more complex than ZeroR because it predicts the majority for each individual attribute. OneR is useful because sometimes the most simple algorithm is the most useful. This algorithm is useful because occasionally there could be one attribute that is most predictive in determining the class. If a more complex algorithm has a lower predictive accuracy than OneR, then it probably is not worth keeping.

Table 2. Lists the filter and the accuracy of OneR as a percentage for the number of instances within the data set. (Without string data)

Filter	**OneR Accuracy**	**Most Predictive Attribute**	**Number of Instances**
None	63.2%	Ratatouille	125
Resample	75.8065%	Ideal_Diet_Coded	124
SpreadSubsample	60.2041%	Ideal_Diet_Coded	98

SMOTE	60.3659%	Mass	164
ClassBalancer	60.0564%	Fruit_Day	125
CostSensitiveClassifier (2.0 penalty)	63.2%	Mass	125
CostSensitiveClassifier (5.0 penalty)	61.6%	Mass	125

Through the use of OneR we found that the Resample filter produced the highest accuracy in determining whether the student was male or female. The most predictive attribute that the Resample filter produced through the OneR algorithm was Ideal Diet Coded. This attribute was from the free response questions the students answered on what they thought was an ideal diet. This was interesting to us because the SpreadSubsample also produced Ideal_Diet_Coded as the most predictive attribute but had 15% less accuracy than the Resample attribute. By increasing the minority class and decreasing the majority class through the Resample filter it produced a higher accuracy. While the SpreadSubsample only decreased the majority class, therefore having a lower accuracy.

C. J48

The J48 algorithm is one of the more complex algorithms we utilized for this project. J48 takes a “divide and conquer” approach from the training data. First, the algorithm uses the most predictive attribute that is created from OneR in order to best divide the attribute into subgroups. From there, the algorithm uses the next best attribute that is able to split off into subgroups until there are no more attributes to use, or the accuracy can no longer be improved. By best dividing the attributes from the most predictive it creates a visualization tree that can be followed down to see the class value.

Table 3. Lists the filter and the accuracy of J48 as a percentage for the number of instances within the data set. (Without string data)

Filter	**J48 Accuracy**	**Most Predictive Attribute (top of tree)**	**Number of Instances**
None	73.6%	Mass	125
Resample	85.4839%	Ideal_Diet_Coded	124

SpreadSubsample	75.5102%	Mass	98
SMOTE	84.1463%	Comfort_Food_Reasons	164
ClassBalancer	70.2873%	Mass	125
CostSensitiveClassifier (2.0 penalty)	68%	Mass	125
CostSensitiveClassifier (5.0 penalty)	63.2%	Mass	125

From applying the filters we found that the Resample filter produced the highest accuracy using J48 at 85.4839%. Although the SMOTE filter was not far behind at 84.1463%. What was interesting about applying these two filters is that J48 used two different attributes between the Resample filter and SpreadSubsample filter for what was most predictive. The Resample filter in conjunction with the J48 algorithm chose Ideal_Diet_Coded as most predictive in determining the class, while SMOTE in conjunction with J48 chose Comfort_Food_Reasons.

2. Comparing Effectiveness of Datasets Using Text Mining

Our dataset contained string attributes from open-ended questions, as well as numeric attributes that were coded by the dataset creators. We wanted to run an experiment that would compare the predictiveness of string attributes versus the pre-determined coded numeric attributes. To do this we ran a side-by-side analysis of the datasets, one with string attributes that had been converted using stringToWordVector and one that had no strings and only the coded attributes. Attributes that were not answered in an open-ended format nor that had been coded were used in both comparisons (ex. GPA, Gender, Income).

V. Results

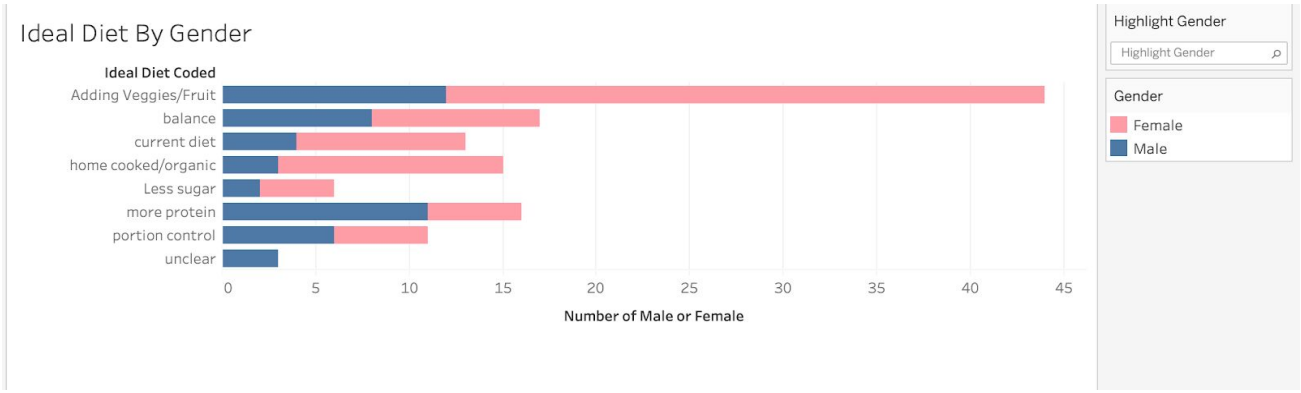

Figure 1. Is the data from the original dataset showing the ideal diet based on gender.

From Figure 1 what we found was interesting is the differences in what gender thought was an ideal healthy diet. Females tend to believe that adding fruits and veggies to their diet is considered more healthy. While the majority who thought adding more protein to their diet is males. We suspect to believe the reason males think adding more protein to their diet is healthier because it will make them more muscular (swole).

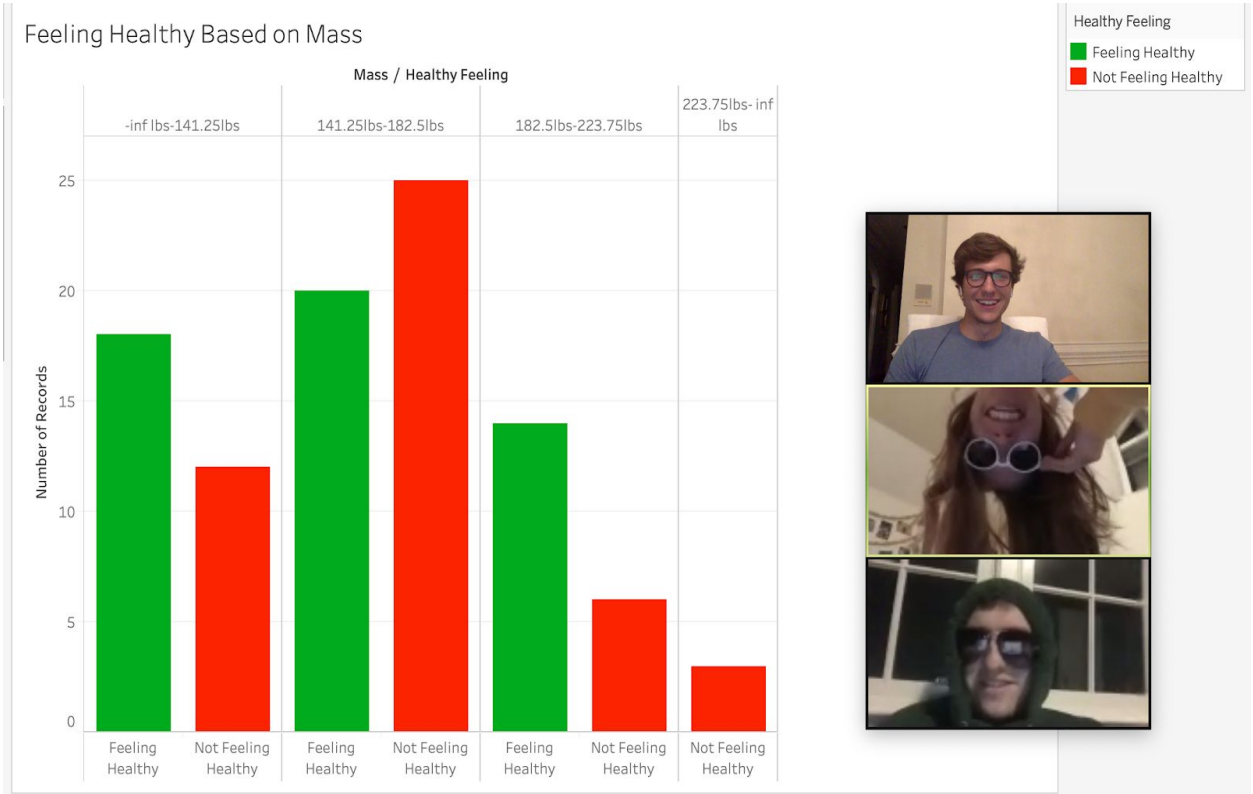

Figure 2. Using the SpreadSubsample filter to determine how healthy a person feels based on their mass.

What was interesting from Figure 2 is that every person who weighed above 223 lbs felt unhealthy. Additionally, there are many people who are at an average weight of 141lbs - 182lbs who also felt unhealthy.

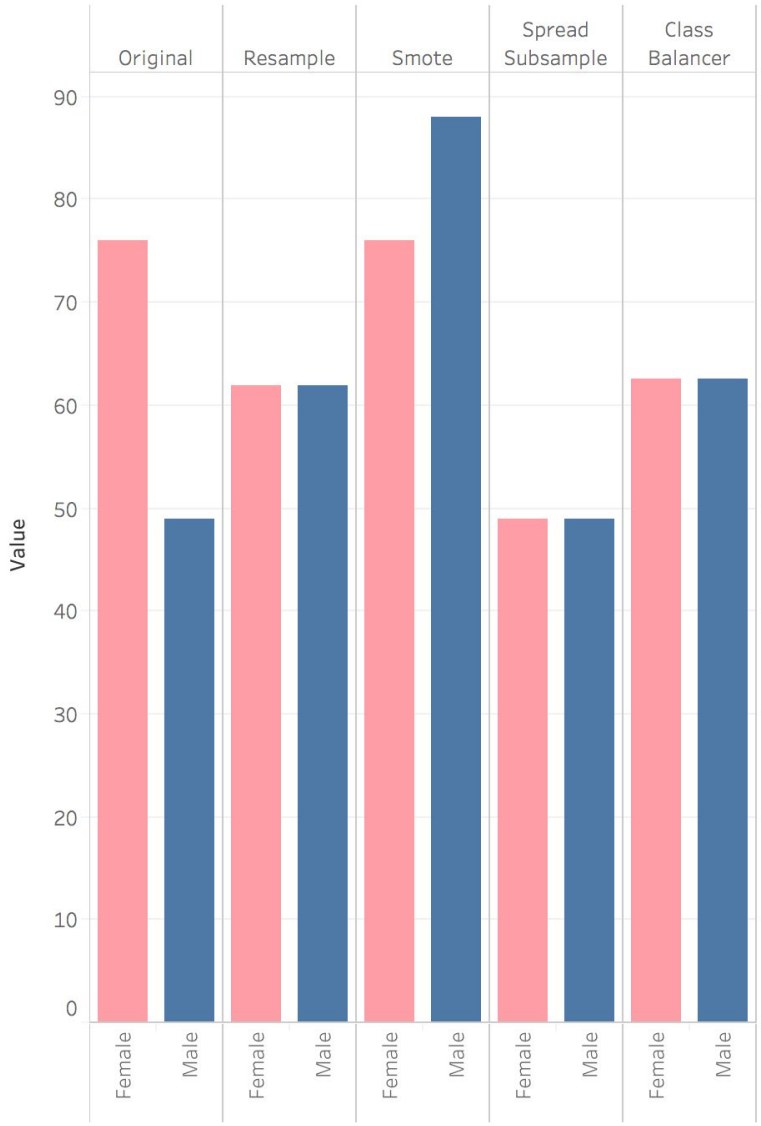

Figure 3. Is the class attribute differences when using the different filters for pre-processing.

The filters used were Resample, Smote, SpreadSubsample, and ClassBalancer. We can see how these filters manipulated the amount of instances used to attempt to create an equal balance between males and females.

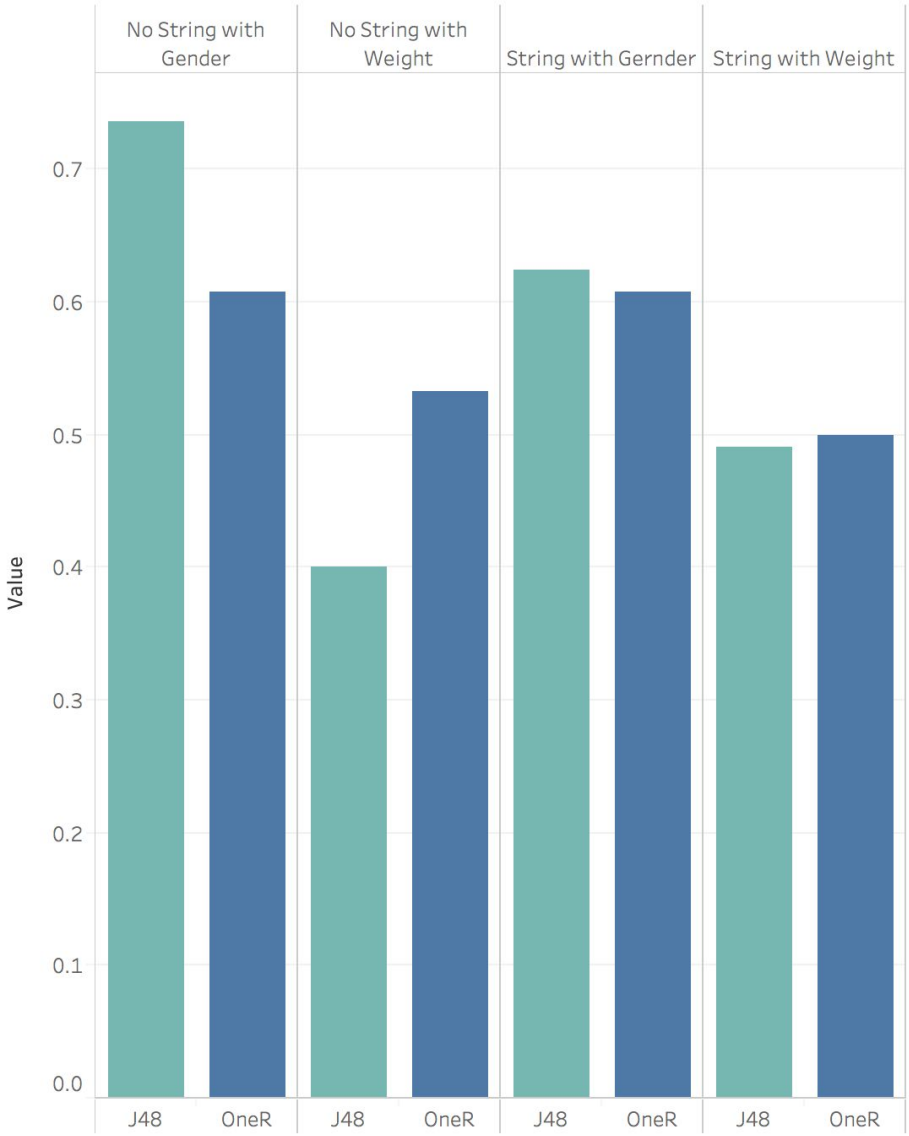

Figure 4. Data collected from the success of datasets containing strings vs. those without.

As Figure 4 suggests, the dataset using no string attributes with Gender as the class attribute is the most predictive. All of the datasets also determine that Gender is more

successfully determined than Weight/Mass when it acts as the class attribute. We see that the datasets including string attributes are consistent in the percentage of instances correctly predicted, regardless of what algorithm is run. From these results, we decided to apply the Resample filter to both string and no string datasets in an attempt to increase the predictability of each.

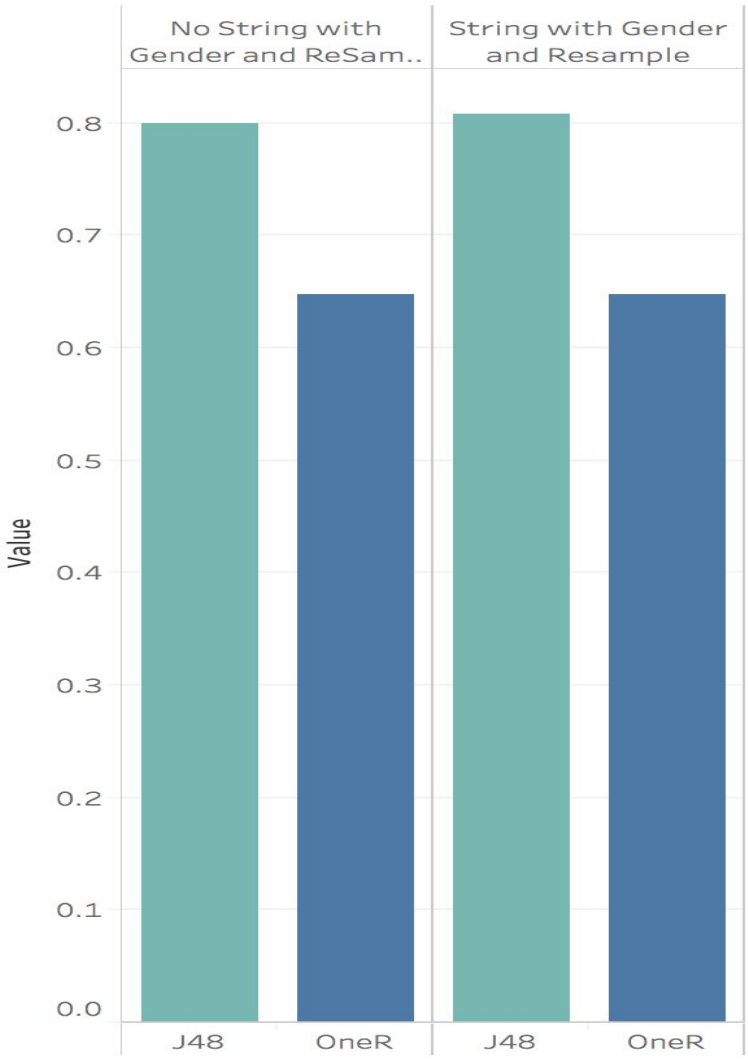

Figure 5. Data collected comparing string dataset vs. without when Resample is applied

We were surprised to find that both datasets, one with strings and one without, were equally successful. This allows us to suggest that the string attributes, which are now being tested as numeric attributes, are not as predictive as the initial numeric attributes in the dataset (Gender, GPA, etc.). We would expect OneR to remain the same due to the nature of the algorithm, which uses only the most predictive attribute to predict the correct class; however the two yielded different results. We found that ideal_diet was

the most predictive value for the dataset with no strings and self_perception was the most predictive attribute for the dataset using strings. We also found that, because the stringToWordVector filter creates many unique attributes, it led to massive overfitting in the J48 tree as seen in Figure 6.

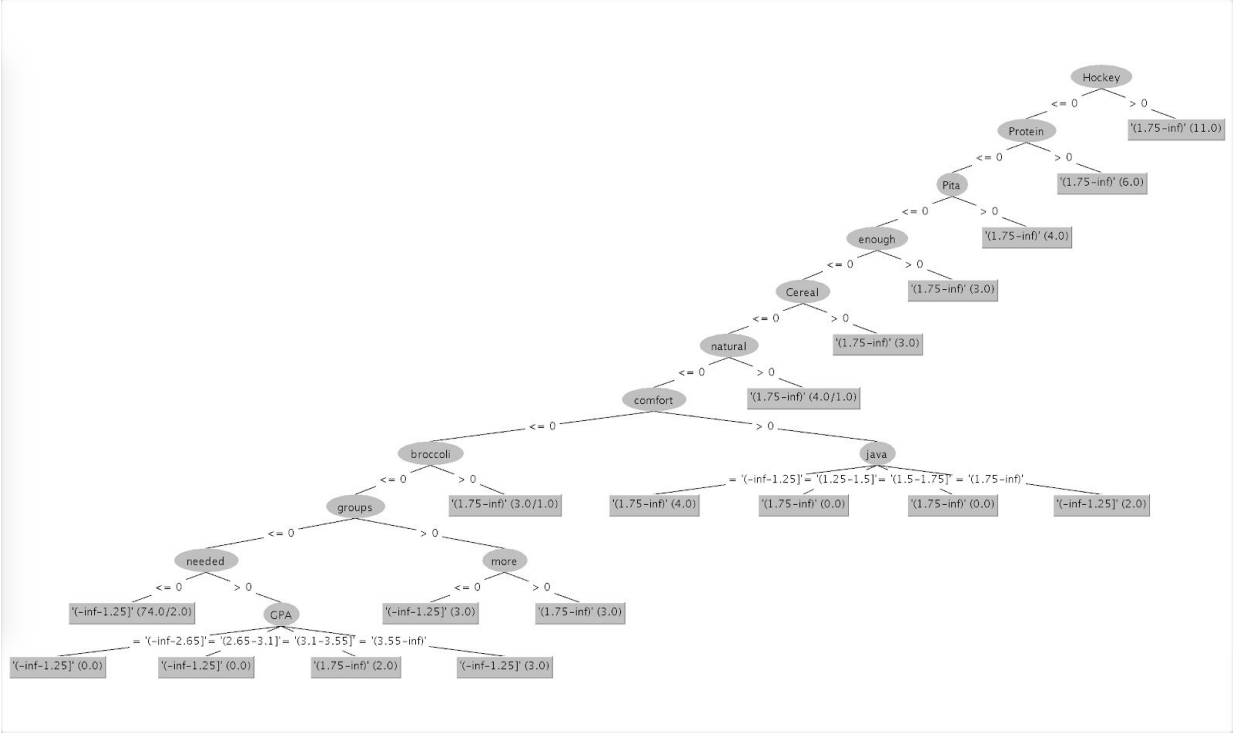

Figure 6. J48 tree from dataset with strings after applying Resample

Confusion	Matrix	
a	b	← classified as
59	17	a = female
16	33	b = male

Figure 7.1 Cost Sensitive Classifier Original Matrices with No penalty

Confusion	Matrix	
a	b	← classified as

33	43	a = female
3	46	b = male

Figure 7.2 Cost Sensitive Classifier matrices with a penalty on males of 5.0

Confusion	Matrix	
a	b	← classified as
49	27	a = female
13	36	b = male

Figure 7.3 Cost Sensitive Classifier Matrices with a penalty on males with 2.0

Applying the Cost Sensitive Classifier produced interesting results. We tested the original data on J48 with the cost sensitive classifier. Originally the data had an accuracy of 73.6% on the J48 algorithm. We found that when we put a penalty of 5.0 misclassifying males it dramatically increased the amount of females that were misclassified and the accuracy was 63.2%. We decided to reduce the penalty of misclassification to 2.0 and found a much more even distribution between the misclassified males and females and produced an accuracy of 68% on J48. This was a good learning experiment to see how dramatically you can alter the data when putting a penalty for misclassifying the data.

VI. Conclusion

While we initially expected the dataset to show high success at predicting the weight of an instance, running various algorithms such as J48, OneR, and ZeroR determined that gender was easier to predict from the given data. As our data was more heavily collected from females than males, we moved to working with various filters in order to balance the data. These filters greatly increased the success of each model. Working with gender as a class attribute allowed us to gain insight into how males and females structured their diets and thought about their health. We were also able perform experimentation on the dataset that we collected in order to determine whether we could make it more predictive through the use of text mining; however, the results proved no different.

Work Cited

Frank, Eibe. “Oversampling and Undersampling.” *WEKA Blog*, 29 Jan. 2019, waikato.github.io/weka-blog/posts/2019-01-30-sampling/.

“ClassBalancer.” *WEKA*, weka.8497.n7.nabble.com/ClassBalancer-td33653.html#a33654.

Google Search, Google, www.google.com/search?q=quotes+about+food&rlz=1C5CHFA_enUS756US756&sxsrf=ALeKk03oihNBws8rnE2z-F-7SolVWb8wig:1588207641219&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjc7KLC9o7pAhXuguAKHZUvDBYQ_AUoAXoECBIQAw&biw=1440&bih=821&dpr=2#imgrc=RF9bRkNEOrNn7M.