**Jonathan Thompson**

**MSBA 645 (Applied Machine Learning): Assignment 3**

**Problem 1:**

The LASSO model is a supervised learning process & model that I felt better attended to the needs of my project. This is different compared to unsupervised & mixed learning such as SOM clustering/regression or Neural Network Deep Learning.

The purpose of analyzing the LASSO model is to find trends of attributes that not only Republican congressmen voted for in 1984 but compare them to what their voters and main audience may adhere to in comparison for similar issues in 2024. We are also attempting to find out if the data set fits overall decent performance within the model itself.

In the data set, I replaced text values with integer values to better fit my model. They are listed as follows:

0=No

1=Yes

2=Absent

**Explanation of coding process:** This multivariate regression model requires partitioning data to improv scalability and optimize performance. Once the last part of this step is performed for the dummy variables, you would then have to create your seeds, index and create the training and testing models to link with the general voting model that you are validating. Then you would input a coefficient function for their outputs to analyze them based on the training set information. Then you would input predicted probability function which produces its own individual output. You would use that function to link it with a new voting prediction for the test set. You would then generate a function to evaluate the voting performance from that prediction and generate TPR & FPR rates if applicable. You would then plot the ROC curve which plots the voting performance & prediction. Finally, you would generate an AUC function to evaluate the overall performance of the LASSO model.

**Findings & additional thoughts:** After analyzation of the data set as a whole, I first want to start off by discussing the coefficients. Based on a potential political consulting job that I would be working in, it is important to know which attributes effect my dataset the most when it comes to how the Republicans voted. So I ranked them based on how much they impacted how republicans voted. The low high magnitude negative absolute values are based how likely the Republicans were at that time to vote no on an issue. The high magnitude positive values are if they are more likely to vote yes.

Here are the coefficients ranked from highest to lowest absolute value:

1. physician.fee.freeze (-7.3462570940)

2. immigration (-4.1488629964)

3. adoption.of.the.budget.resolution (4.2739567808)

4. synfuels.corporation.cutback (4.2380480194)

5. religious.groups.in.schools (2.9371112033)

6. mx.missile (2.8734746520)

7. education.spending (-2.8716710656)

8. el.salvador.aid (-1.9508317926)

9. superfund.right.to.sue (1.2929924170)

10. aid.to.nicaraguan.contras (-1.2306520591)

11. duty.free.exports (1.2031440088)

12. export.administration.act.south.africa (-0.9640994296)

13. crime (-0.6106832590)

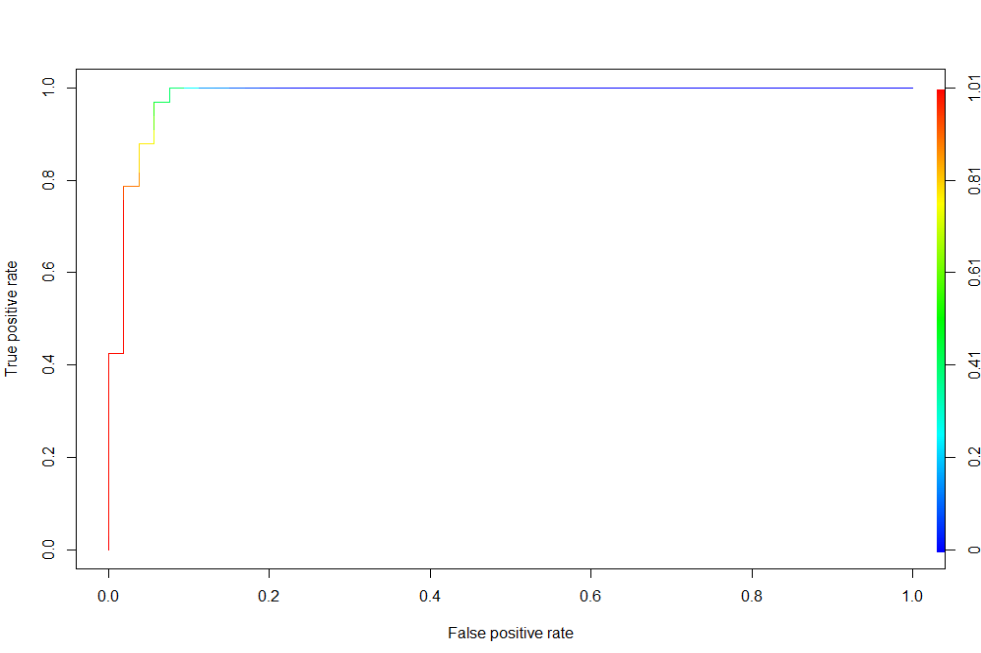
14. handicapped.infants (0.2526083232)

15. water.project.cost.sharing (-0.0033913393)

16. anti.satellite.test.ban (-0.0006578661)

Attributes like physician freeze, immigration, and education spending could potentially be taken out of the data set and not be considered while attributes like Synfuels, adoption of the budget resolution and religious schools in groups can be more heavily utilized in an example of a focus group.

Next, I analyzed the ROC Curve below:



After visualizing this, I noticed for the TPR, it shoots almost straight up with very little movement to the right. This indicates a high accuracy of the test and would indicate better performance for the model. I would say this is a good sign that the model is a good fit for this data.

Finally, I performed an AUC function to output a score for the overall prediction of the test set of 0.9822756. AUC values above 0.9 are considered excellent and suggests that the model can very effectively distinguish between positive/negative classes but outstanding model performance.

**Summary:** In summary, I would use this LASSO model for this dataset as it not only showcases exemplary performance, but also model interpretability, prediction accuracy and handling multicollinearity. We can use this model in order to better interpret general results from it with another more recent data set.

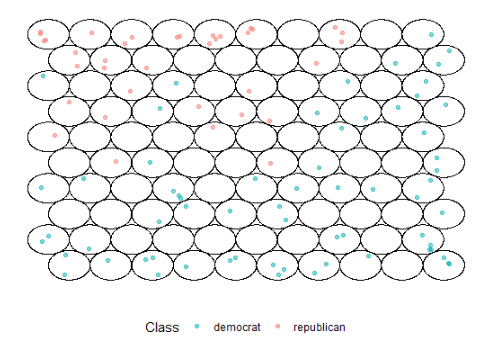
**Problem 2:**

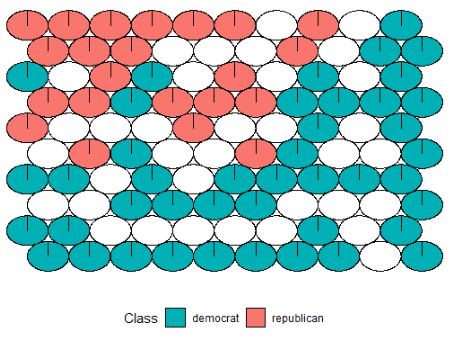
The SOM model is an unsupervised learning method & model that is excellent for arranging categorical variables into dimensional grids and comparing them to predictor variables based on clusters.

In contrast to Assignment 2, I wanted to use an Unsupervised learning to gain an overall perspective of how Republicans vote compared to Democrats in the data set I have. This could be compared to what they vote on, how often they vote and how frequently they may vote on a particular bill/issue. For my fictional consulting job, it could be used to compare target older voters based on using the model to compare the entire data set. It can also be used to compare young and older voters in 2024 who also voted in 1984. We want to use this data to hopefully improve campaign efforts in order to flip voting from Democrat to Republican in swing states.

**Explanation of coding process:** The SOM model requires creating a sample from the main data set and then modeling that sample. I decided to use a 100-observation sample as a base for the solution to my problem. Next, I created a base grid for the plot as well as points using tibble construction. Proceeding this, you would then have to use a mapping function to map as well as color the plot. Once this is performed and executed, you would then write out the coding for the beginning components of how you would like your circle-layer jitter plot to be outputted. Then you would print that plot to make sure the coding is accurate. You would then plot the actual jitter plot you want that would include linking some points data, and the DV (Class) along with its colors in with the formula. The final step of the coding would be to assess the fractions of each Class based on the plot you just outputted by counting, grouping, filtering, & plotting it.

**Findings & additional thoughts:** After analyzing the below circle layer jitter plots, I noticed that based on the random sample of data attained that democrats tend to vote more on a specific bill/variable than a Republican would. While this may have been a good strategy back during the Reagan era in 1984, not voting on a specific bill in today’s current climate could have an adverse impact on the entire Republican class. Historically, Republicans have gained a reputation of voting no or even being casted as “Absent” during a voting call. If the hypothetical 2023 data set shows the same trends as the jitter plots below, this could result in such examples as a lost seat or flipped state in the upcoming 2024 U.S. election. Below, the data points from the plots are useful in visualizing individual voting data points and their distribution. Hence, why my glimpse outputs at times show a point & distribution meter. The clustering in the coding itself really helps me and hopefully others to gain an overall perspective of what this sample data can provide knowledge to.





**Summary:** Overall, for the SOM model specified, it gives a good overview of what 1984 congressional voting echoes based on specific issues from a sample and can be a good model in order to compare to a more recent voting data set. In conclusion, it does this by helping the viewer better understand high dimensional data by reducing the dimensions of voting data to a map.

**Problem 3:**

The Single-Node and Multi-Class Neural Network model is essential for capturing intricate relationships in voting data and interpreting results for binary classifiers. Unlike regular Unsupervised learning, Mixed learning can use more complex models that incorporate supervised and unsupervised techniques and train my model to identify voting patterns of labeled and unlabeled data. For my fictional consulting job, I could use both to predict voting patterns from the 1984 data set to another recent data set to flip House seats or swing states.

In my coding and problem, I decided to use two different parts. Part A coding represents a Single-Node Neural Network model to showcase issues that have the most significant impact on distinguishing between Democrats and Republicans. Part A is especially great for binary classifiers such as my Class variable of Republican & Democrat. Part B presents a Multi-Class Neural Network model to showcase nuanced relationships between voting patterns and party affiliations.

**Explanation of coding process:** Both Neural Network models require creating their own separate, unique models. They also require generating a regular plot to produce its output. Specifically, with the Multi-Class model you would have to write a separate function for converting the DV as a factor, adding a fake class for all factors, and generate outputs for all levels, and re-leveling them as a factor to fit the model. You would then link all dummy variables not including the Class DV. You would then proceed to adding column names for the DV in regards to the plot and then create the actual Multi-Class model. Then you would add a separate function to link and compute the Multi-Class model with the plot you are about to create. In the final step, you would import the Useful library and then plot your model out. For the Multi-Class plot, I generated 10 different pop-up plots just to gain an overall perspective and compare.

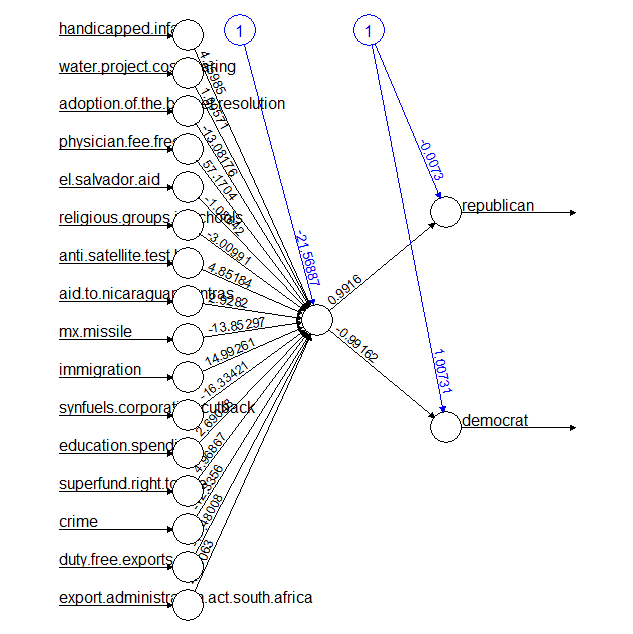
**Findings & additional thoughts:** Viewing the both model’s plots, I want to take examples from the one plot I generated for the Single-Node and one of the 10 I generated for the Multi-Class.

For the **Single-Node plot** below, the 16 IV predictors alone have an output value of a little over –21 which indicates that the model is highly confident that the input data belongs to a specific class, but this class is typically associated with the negative output in the model's design. Based on 2 of the top 3 highest coefficients being extremely negative from our LASSO regression findings in Assignment 2, I would most likely guess that the negative class is the Republican class. Negative output values from the neural network sometimes represent probabilities or confidence scores associated with the negative class. A highly negative value would indicate a very low probability or confidence in predicting the "Republican" class based on the input data. However, the variables themselves with just the Republican voting indicate a high probability of 0.9916. Finally, there is a link connecting Republicans and Democrats where Republicans have an output value of -0.0073 and Democrats have an output value of 1.00731. This indicates a strong preference for Democrats and a weak preference for Republicans in regards to confidence and uncertainty. A possible specific example or reasoning for this could be that when Ronald Reagan was president, there was a Democratic house majority and the Republicans rather felt there was no point in voting for their usual interests or that they indicated an absent vote. Let’s also not forget that this model is good for analyzing binary predictors & classifiers.

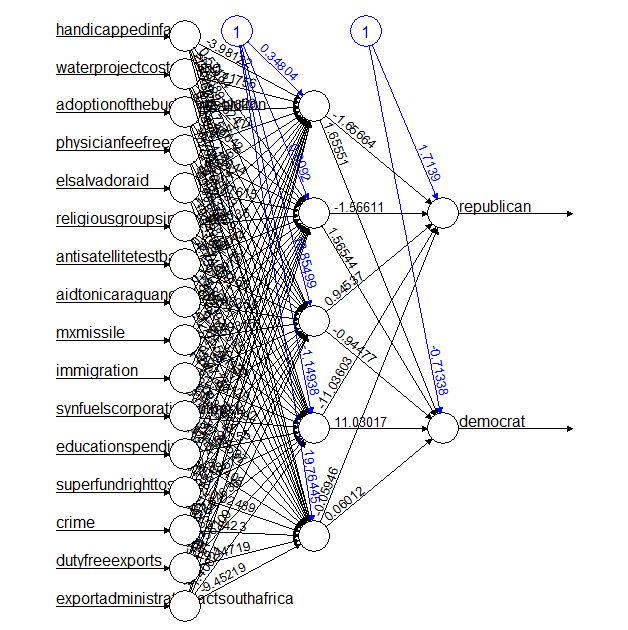
For the **Multi-Class plot** below, this is a more complex handling of classification tasks that could be interpreted as more accurate. There are variance output values in blue that are of strong and weak confidence between variables. However, there is an positive output value for the Republicans for 1.7139 and a negative output value for the Democrats of –0.71338. This means that there is higher confidence within this model when it comes to the Republican class. Because of the richer insights and the real-world relevance that this model offers over the Single-Node one, I would say this is a more accurate interpretation of what the voting for each issue means and how they are linked.

**\*\*\*Please see both plots on each of the next two pages\*\*\***

**\*\*Single-Node Plot\*\***



**\*\*Multi-Class Plot\*\***



**Summary:** After analyzing both models & plots, I would consider the Multi-Class Neural Network model over the Single-Node model. This is because not only does it generate a higher confidence level between the two variables, but it also presents more accurate results in general that I could use with a more recent data set in order to help flip House seats or swing states.

**SOURCES:**

1.) **SPECIFIC VARIOUS CLASSMATE DISCUSSIONS**

2.) **DATA SET SOURCE:** <https://archive.ics.uci.edu/dataset/105/congressional+voting+records>

3.) **LASSO MODEL CREATION FOR PROBLEM #1:** MSBA 635 PREDICTIVE ANALYTICS: LASSO MODEL SESSION FROM PROFESSOR BRITTANY GREEN

4.) **SOM REFERENCE FOR PROBLEM #2:**

<https://www.r-bloggers.com/2018/01/soms-and-ggplot/>

5.) **SINGLE NODE & MULTI-CLASS NEURAL NETWORK REFERENCE FOR PROBLEM # 3:**

<https://www.learnbymarketing.com/tutorials/neural-networks-in-r-tutorial/>