Multidimensional Scaling and Self-Organizing Maps

Serra Uzun, MSDS_411 FALL 2020
10/25/2020

Introduction

Multidimensional Scaling (MDS) and Self-Organizing Maps (SOM) are effective unsupervised learning models in visualizing and understanding the proximities of observations as well as seeing the grouping and clustering of the data within a large complex dataset. In the context of exploring these two models, we will be conducting them separately on different datasets in our analysis. Component 1 of our report will walk through the MDS on Recidivism dataset, and Component 2 will be on the SOM model for College Acceptance dataset. We aim to get an idea of the clusters and groups of observations that have proximity and essentially indicate the same or similar outcome or characteristics through both models.

COMPONENT #1: MULTIDIMENSIONAL SCALING

Exploratory Data Analysis *(Question #1)*

The Recidivism Dataset consists of random sample records on convicts released from prison between 1977 and 1978. The dataset has 18 variables and 1,445 observations. The dataset variables are either binary or continuous, and there are no categorical variables that came with the original raw dataset. Below is the list of variables and their types:

VARIABLE	TYPE	DESCRIPTION
BLACK	binary	1 if black, 0 if not
ALCOHOL	binary	1 if uses alcohol, 0 if not
DRUGS	binary	1 if uses drugs, 0 if not
SUPER	binary	1 if super, 0 if not
MARREID	binary	1 if married, 0 if not
FELON	binary	1 if a felon, 0 if not
WORKPRG	binary	1 if in a work program, 0 if not
PROPERTY	binary	1 if has property, 0 if not
PERSON	binary	1 if person, 0 if not
PRIORS	continuous	Number of priors
EDUC	continuous	Education level
RULES	continuous	Rules
AGE	continuous	Age
TSERVED	continuous	Time Server
FOLLOW	continuous	Follows
DURAT	continuous	Duration
CENS	binary	1 if cens, 0 if not
LDURAT	continuous	Log function of Duration (DURAT)

Table 1: Recidivism Dataset Variables and Descriptions

We plot the histogram of each variable for a quick EDA, which we can do for all variables as they are all numeric, to see the frequencies and overall distribution. Also presented as histogram plots in Figure 1, in our dataset, we have observations for 701 black and 744 non-black persons, 303 alcohol users, and 1142 non-alcohol users, 349 drug users, and 1096 non-drug users. In addition, we observe that the education level of the people referred to in our dataset ranges mainly between 5 and 12, time served mainly below 50, and priors mostly less than 4.

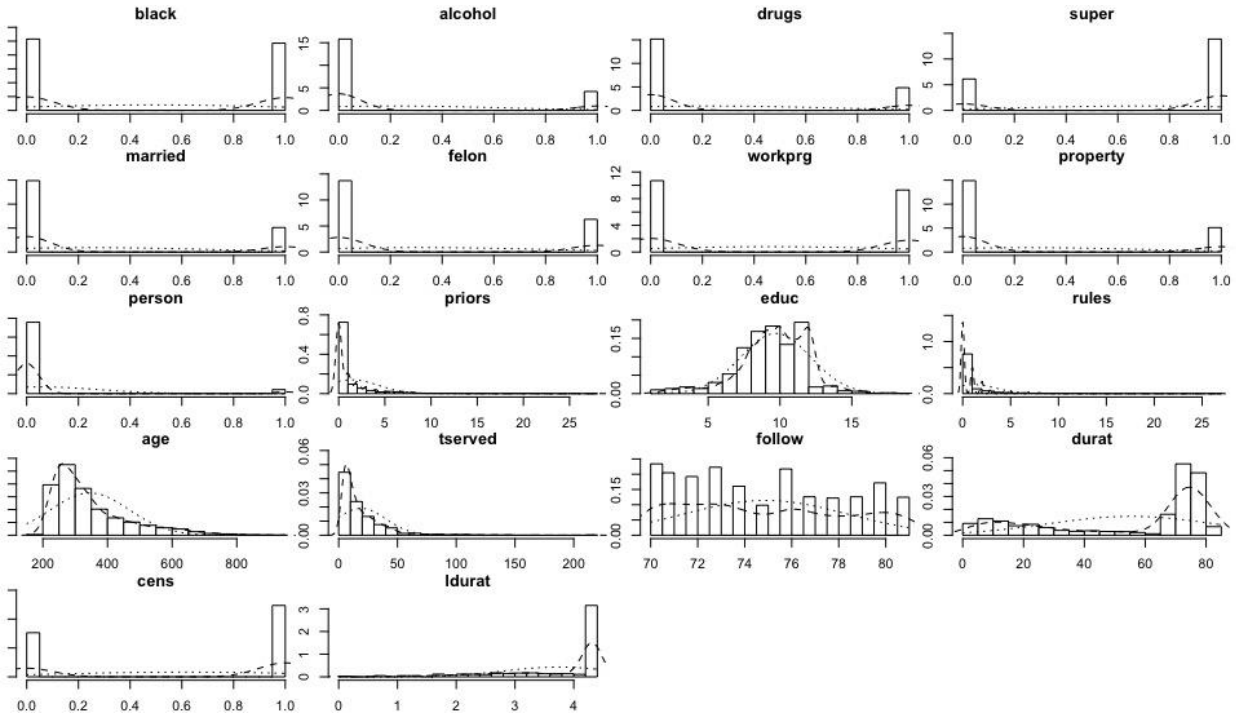

Figure 1: Multi-Histogram of Recidivism Dataset

Following the general outlook and histograms of the dataset, we plot a correlation plot that shows any significant and insignificant relationships between variables. Presented in Figure 2, the correlation plot shows us that the highest correlation is between cens, durat and ldurat. This shows us that we may not need to include all of them in our model moving forward due to significantly high correlations between these variables. In addition to these correlations, we observe a noticeable correlation of 0.7 between felon and property variables as well as a correlation of 0.5 between time served and felon. This suggests that we may see some clustering of variables related to this correlation. We do not observe any significant (correlation greater than -0.5) negative correlation between any variables. Aside from these correlations, the correlations between variables in the Recidivism dataset are below 0.5 and -0.5.

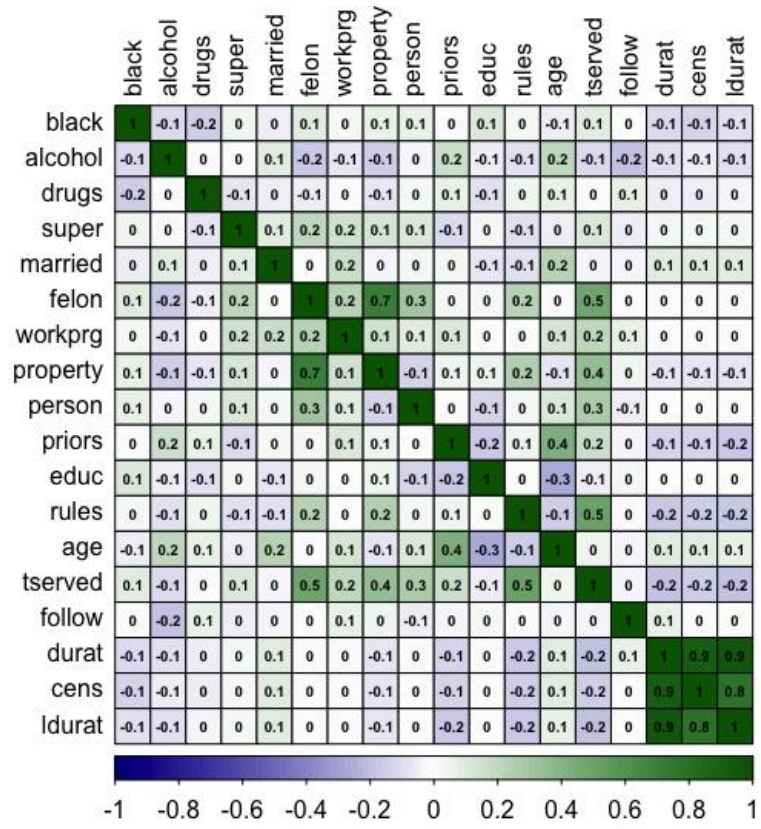

Figure 2: Correlation Plot of Recidivism Dataset

Per our EDA results, we will not include cens and ldurat in our MDS model moving forward. We want to explore few of the key variables through our MDS model: time served (tserved), black, priors, property, felon, follow, and duration (durat). We aim to see a pattern or cluster with these variables as we will plot our MDS in two classes in the black variable that are 1 if black and 0 if not. While other variables are also important, we are looking to analyze variables that are more race-related in this context.

Dissimilarity Matrix *(Question #2)*

The Dissimilarity Matrix we produced with Euclidean Distances is extensively large, with 1,043,290 elements, making it very difficult to identify any patterns in the matrix. Even by looking at the first 100 elements of our matrix, we cannot observe any patterns. In order to better understand and interpret our Euclidean Distances Dissimilarity Matrix we use the Classical Multidimensional Scaling with k=2 and use the two columns of the fit to plot a 2D view of all of the distances, which is presented below in Figure 3.

(Question #3)

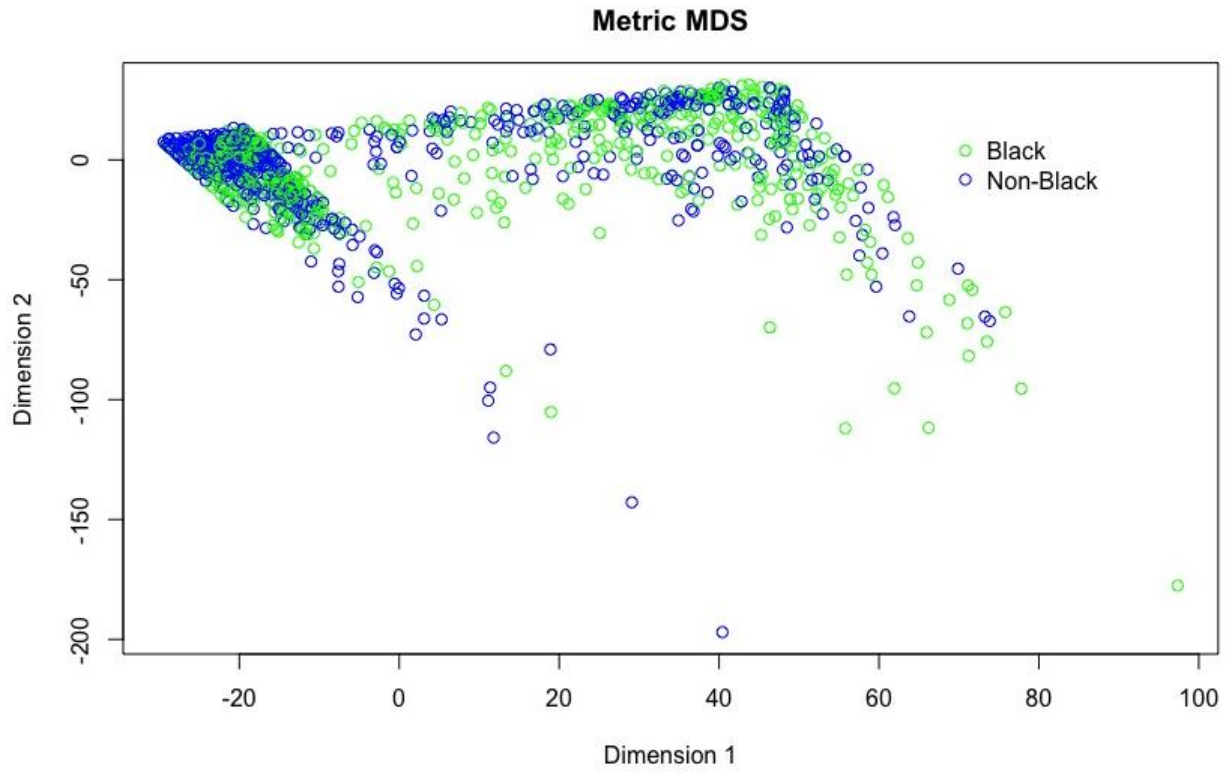

Figure 3: 2D Plot of Metric MDS

As our dataset did not have any categorical variables, we chose to present the plot by coloring the black person's or non-black person's data points. Also seen in Figure 3, most of the points on the plot that represent distance are located near or slightly above along the line of 0 of Dimension 2. As this cluster moves towards above 0 of Dimension 1, we see an increase in the number of points representing black convicts' observations. That said, we can clearly see the pattern along Dimension 1 (x-axis) where the dimension increases the number of 'black' points presented in green increase and the distribution of the points gets less clustered and more spread out. On the other hand, when we look at the distribution of the points along Dimension 2 (y-axis), we see that the points' cluster start fading out significantly pass the -50 dimension. Along Dimension 2, we don't observe noticeable distancing between black and non-black points plotted and even see some similar type of outliers towards -150 and -200.

(Question #4)
As the final step of our MDS analysis, we will run our model using Ramsay's Method while keeping the same variables that we have in our previous model above. The Ramsay Method, aka Non-Metric Multidimensional Scaling (Non-Metric MDS), uses qualitative similarities instead of quantitative and presents the distances on an ordinal scale. In the case of the Recidivism dataset and the variables we have chosen to explore, there are almost non-qualitative data, so we are not expecting to obtain any useful insights from the Non-Metric MDS.

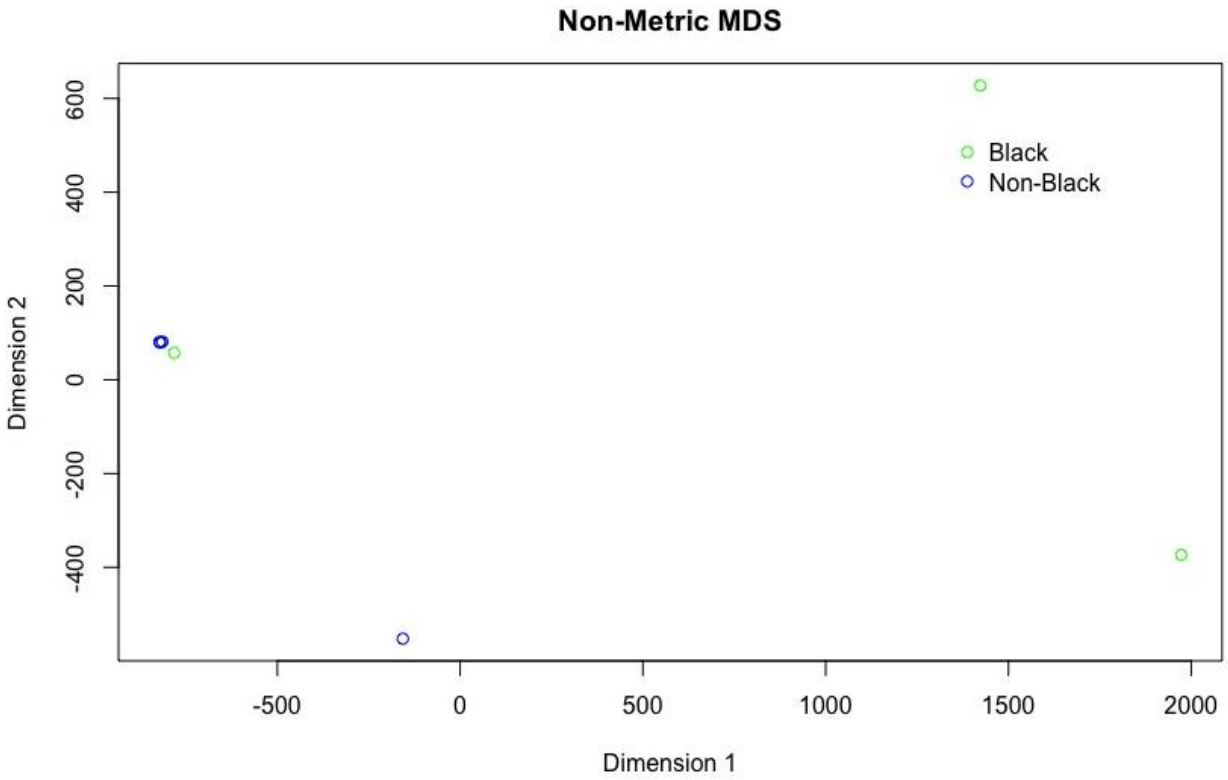

Figure 4: 2D Plot of Non-Metric MDS

As seen in Figure 4, the non-metric MDS that was run with the variables we have chosen for our analysis does not provide useful and interpretable results. The only thing to note is that Black group points are more towards the lower and upper end of the dimensions, whereas Non-Black groups l relatively closer to 0 on either one of the dimensions.

In the conclusion of our MDS analysis, we can state that we observed different patterns for Black and Non-Black groups in a Metric Multidimensional Scaling. While we need extensive further study to understand these patterns and relationships in detail, the MDS analysis with the Recidivism dataset has shown us that there is a difference in the pattern for Black and Non-Black convictions between 1977/78. The dataset should be further analyzed and modeled with time served (tserved), black, priors, property, felon, follow, and duration (durat) variables to unveil any racism underlying the convictions.

COMPONENT #2: SELF-ORGANIZING MAPS

Exploratory Data Analysis *(Question #5)*

The College Acceptance datasets consist of 4 variables and 400 observations. While no missing data was observed in the dataset, multiple duplicate observations were observed, which we decided to keep due to the dataset not having a unique identifier. There is a likelihood that different students coincidently got the

same results. The admit variable is the only binary variable in the dataset, which indicates whether or not the student has been admitted to college. gre and gpa are numeric variables related to academic performance and test scores the student in the observation has obtained. Finally, Rank is the ordinal variable that indicates the school rank. Please see the summary statistics below:

VARIABLE	TYPE	DESCRIPTION
admit	binary	1 if admitted, 0 if not
gre	numeric	GRE score of the student
gpa	numeric	GPA of the student
rank	ordinal	School rank

Table 2: College Acceptance Dataset Variables, Types and Descriptions

There are no missing values within the dataset, and there are 26 unique GRE scores and 132 unique GPAs. This re-iterates the duplicate rows we have found in the dataset, with 400 observations and the number of unique variables much less than that count. The multiple histogram plots for each variable in the college acceptance dataset show the overall distribution of the values under each value, presented in Figure 5 below:

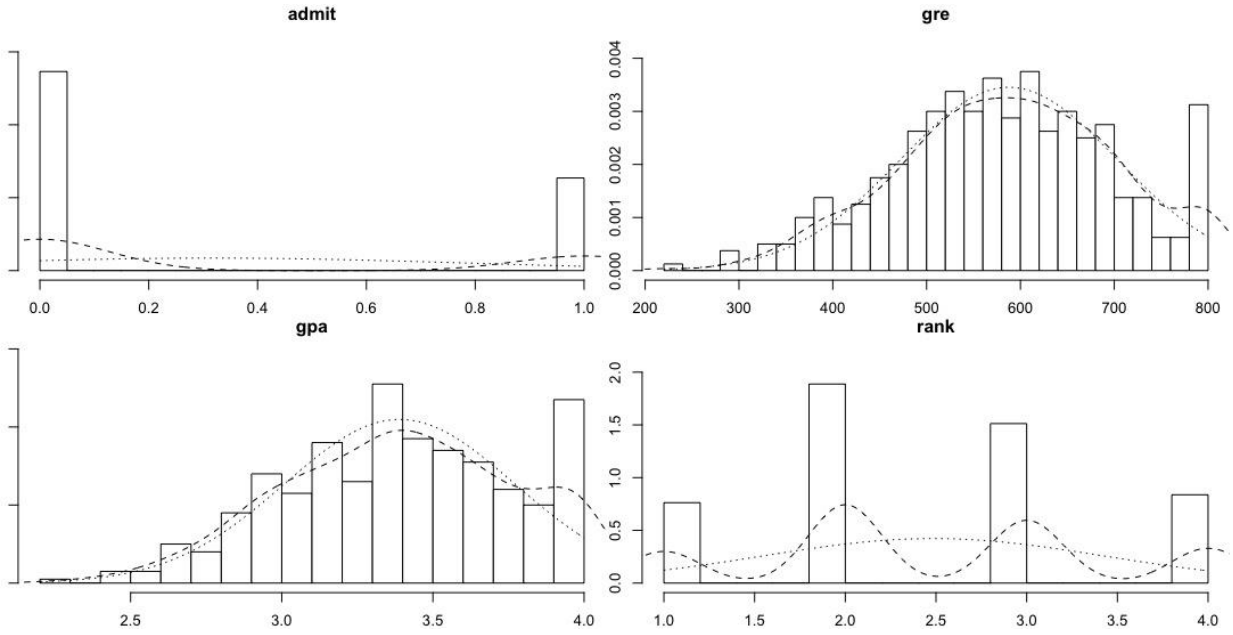

Figure 5: Histogram Plots for All Variables in College Acceptance Dataset

	Min	1st Qu	Median	Mean	3rd Qu	Max
admit	0	0	0	0.3175	1	1
gre	220	520	580	587.7	660	800
gpa	2.26	3.13	3.395	3.39	3.67	4
rank	1	2	2	2.485	3	4

Table 3: Summary Statistics of College Acceptance Dataset

Through the histogram in Figure 5 and Table 3, we have more observations of students who were not admitted in the dataset than admitted. While there are 127 admitted students, there are more twice as much not admitted students of 273. We see that the mean of gpa and gre are both towards the higher end of the spectrum through the summary statistics and the histogram plots, thus slightly right-skewed. While overall, the histogram plots for gre and gpa show a symmetrical distribution, there are large amounts of maximum values, gre score 800 and gpa 4.0, which seem to be the reason for the right skew.

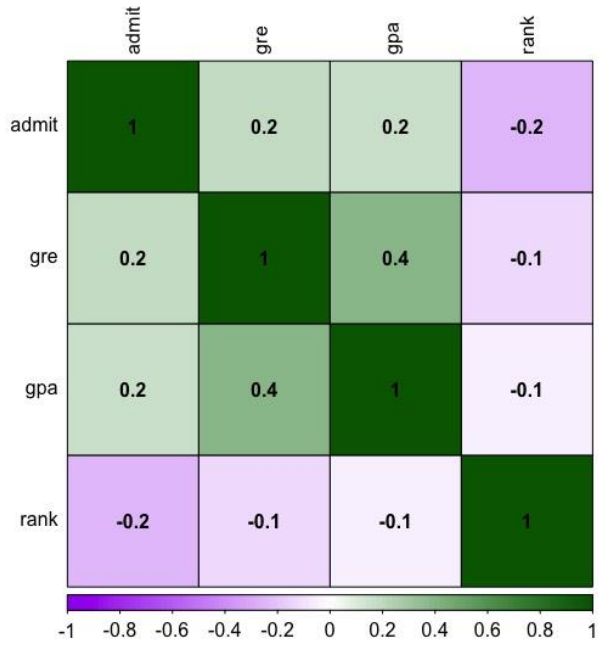

Figure 6: Correlation Plot of College Acceptance Dataset

The correlation plot presented in Figure 6 identifies the positive correlation between admit, gre and gpa, which are essentially all academic and admission success related and expected to see. Rank variable on the other hand is negatively correlated with all variables in the dataset, the most with admit.

In addition to summary statistics and correlation plot, visualizing the distribution of the variables in the context of admitted vs. not admitted is crucial. Presented in Figure 7 below, the density distribution plots give us a glimpse of the possible patterns that we may see throughout our SOM analysis.

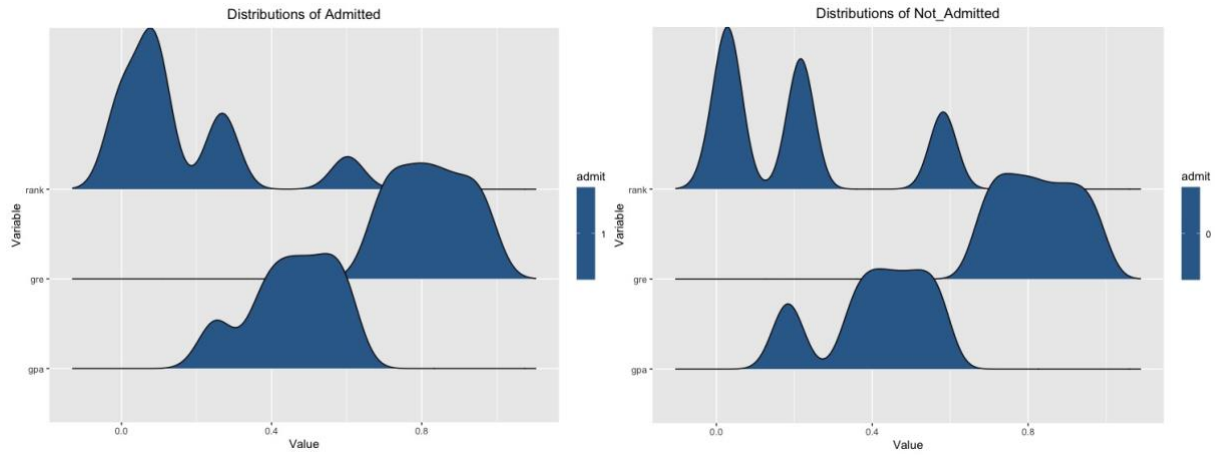

Figure 7: Distribution of Admitted vs. Not Admitted Students

Per the two plots in Figure 7, we can see that the admission outcome's distributions are not substantially different and contradicting. While we see the same layout of distribution pattern in general, we can see that for each variable, the student group who were admitted have slightly higher rank, gpa, and gre. The most noticeable difference between the distributions are with rank, and gpa. We can observe that rank variable is distributed more heavily towards the left side of the spectrum, where rank is closer to 1. On the other hand, gpa is more heavily plotted closer to the 0 on the Not_Admitted Distribution plot. Both these observations are expected due to the nature of the dataset and not concerning.

Finally, following our basic EDA we prepare our dataset for Self-Organizing Map (SOM) model by normalizing all variables in order to have them on a common scale. Our SOM is likely to perform better with all variables in the same range.

Self-Organizing Map Model *(Question #6, #7 & #8)*

The grid for our SOM model was set by multiplying the square root of the number of observations in the dataset with five. For the College Acceptance dataset, this approach gives us a 10 by 10 grid. As for the number of epochs, which indicate the number of passes the model will take on the training data, we are choosing to begin our analysis with 1000 epochs for our one-layer SOM model. Our SOM model with 10 by 10 grid and 1000 epochs produce the following plots in Figure 8, which suggest that we need to explore a smaller grid size to begin with. As we look at our 'Counts Plot' we can see that there quite a few empty nodes that we aim not to have in our model. In order to remove these nodes, we need to bring the grid from 10 by 10 to 7 by 7, multiplying the square root of the number of observations by 3 instead of 5.

In addition to decreasing the size of the grid, increasing the number of epochs is likely to improve the model's performance and give us better, more accurate results.

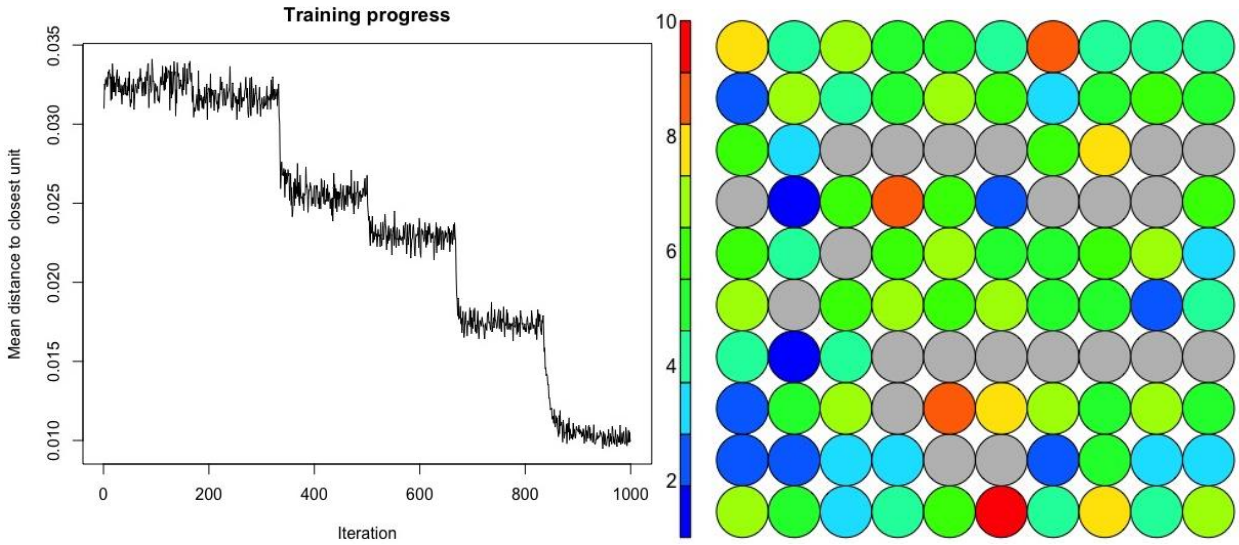

Figure 8: Changes and Count Plots with 10x10 grid and 1000 epochs

Per the counts plot in Figure 8 we see that there are at least 4 nodes that are quite distant from their neighbours and more than a dozen empty nodes. These results suggest that we need to improve and increase the iterations, as well as create a more compact grid for better results. In order to improve the performance of our model, we ran it with 7 by 7 grid and 2000 epochs.

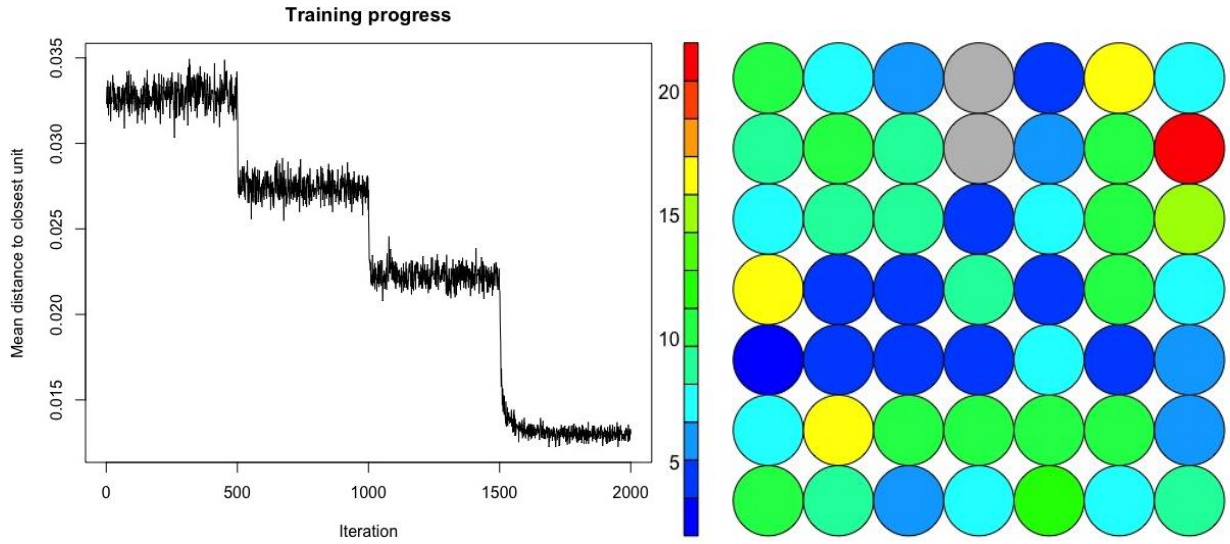

Figure 9: Changes and Count Plots with 7x7 grid and 2000 epochs

As we decrease the grid size and increase the epochs, we observe a significant improvement in our SOM model. Specifically focusing on the counts' plot in Figure 9, we see that the number of distant nodes from their neighbors has decreased significantly. Presented below is the Neighbour Distance Plot of both version of our model, with 10 by 10 grid and 1000 epochs and 7 by 7 grid with 2000 epochs.

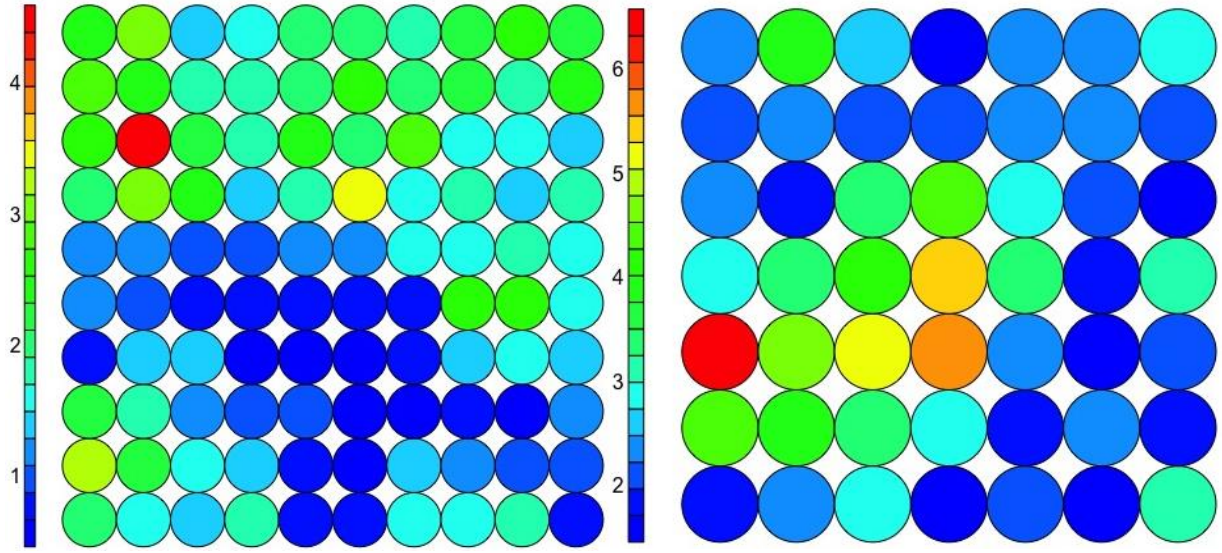

Figure 10: Neighbor Distance Plot 10x10 grid (left) and 7x7 grid (right)

Upon reviewing the Neighbour Distance Plots in Figure 10, we can observe that the distances of nodes from their neighbors have improved as we modified our grid's size and increased the number of epochs from 1000 to 2000. While the improvement is not extensive, the overall cluster of close neighbors is better spread out and distributed with 7x7 grid than with 10x10 grid. To get a more granular look at and compare the first row of the unit.classif of both version of our SOM model.

10x10 grid SOM Model	23	89	96	91	1	77	71	43	92	59	9	63	96	57	96	14	9	21	87
7x7 grid SOM Model	6	38	43	25	19	44	16	4	24	48	47	3	43	8	43	13	47	18	37

Table 4: First row of nodes of both versions of the SOM Model

The improved neighbor distance of our second version of the SOM model can be seen in Table 4 and Figure 10. The fluctuation and large distances between the neighbouring nodes indicate the improvement of our

model performance. The 'codes' plot of each version of our SOM model will give us a more granular look into the variables and how they are in our SOM model. We will be running the codes plot with our improved model with 7x7 grid and 2000 epochs. (See Figure 11 on the next page)

Within each fan within the node in the codes plot we are able to see the magnitude of the variable. The codes plot suggest that the acceptance is generally clustered by the gre score and student gpa. On the other hand, gre and gpa also seem to be clustered by the rank presented in the nodes on our plot's right portion. This result not only gives us a better look at the distributions. Clustering and relationships of variables within the College Acceptance dataset, but also re-iterates our findings through our EDA and previous SOM plots.

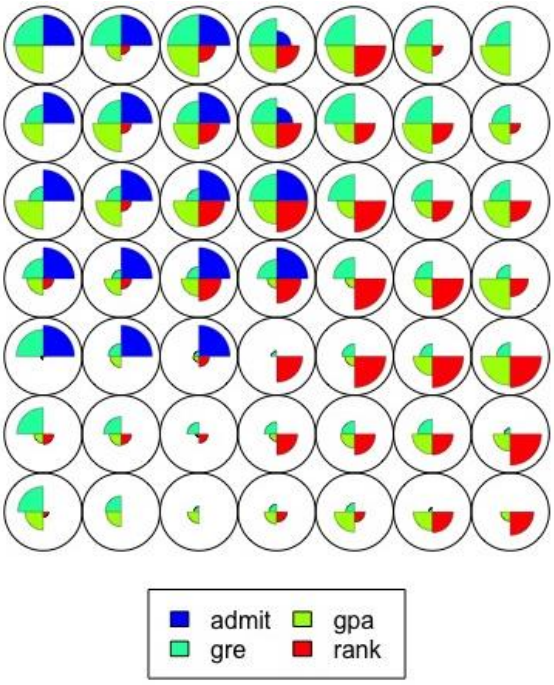

Figure 11: Codes Plot of SOM Model Version 2 with 7x7 grid

Reflection *(Question #9)*

SOM and MDS to me, seemed to be a more advanced and thorough way of assessing and visualizing the clustering and proximity that we have performed with t-SNE. While I found the MDS to be greatly useful in identifying any patterns and significant distributions, I found the granularity SOM offered through the various plots is something I found more useful and easier to interpret. I would personally be interested in using the Recidivism dataset with the type of SOM model we ran for college acceptance in order to determine any clusters or patterns related to underlying racism in the context of convictions.

The R Code for Assignment #3

```
library(plyr)
library(psych)
install.packages("corrplot")
library(corrplot)
install.packages("vegan")
recidivism.df <- data.frame(recidivism)

colnames(recidivism.df)
head(recidivism.df)
str(recidivism.df)
summary(recidivism.df)

unique(recidivism.df$priors)
apply(recidivism.df, 2, function(x) length(unique(x)))

x1 <- recidivism.df$black
x2 <- recidivism.df$alcohol
x3 <- recidivism.df$drugs
x4 <- recidivism.df$super
x5 <- recidivism.df$married
x6 <- recidivism.df$felon
x7 <- recidivism.df$workprg
x8 <- recidivism.df$property
x9 <- recidivism.df$person
x10 <- recidivism.df$priors
x11 <- recidivism.df$educ
x12 <- recidivism.df$rules
x13 <- recidivism.df$age
x14 <- recidivism.df$tserved
x15 <- recidivism.df$follow
x16 <- recidivism.df$durat
x17 <- recidivism.df$cens
x18 <- recidivism.df$ldurat

multi.hist(recidivism.df)
rec_cor <- cor(recidivism.df)
corrplot(rec_cor, method = "color", outline = T, addrect = 4, rect.col = "black", rect.lwd = 3,cl.pos = "b", tl.col =
"black", tl.cex = 1, cl.cex = 1, addCoef.col = "black", number.digits = 1, number.cex = 0.60, col =
colorRampPalette(c("dark blue","white","dark green"))(100))

recidivism.df$ldurat <- NULL

rec.df.mds1 <-cbind.data.frame(x1,x10,x14,x6,x8,x15,x16)


d1 <- dist(rec.df.mds1)
d1[1:200]
d1a <- as.matrix(d1)
d1a
fit1 <- cmdscale(d1a, eig=TRUE, k=2) # k is the number of dim
fit1 # view results
```

```
# plot solution
par(mar = c(5,5,3,1), bg="white")
x <- fit1$points[,1]
y <- fit1$points[,2]
plot(x, y, col = ifelse(recidivism.df$black <= 0,'blue','green'),  xlab="Dimension 1", ylab="Dimension
2",main="Metric MDS")
legend("topright",
       legend = c("Black", "Non-Black"),
       col = c("green",
               "blue"),
       pch = 1,
       bty = "n",
       pt.cex = 1,
       cex = 1,
       text.col = "black",
       horiz = F ,
       inset = c(0.1, 0.1))


d3 <- as.matrix(dist(rec.df.mds1)) # euclidean distances between the rows
d3
fit3d <- cmdscale(d3, eig=TRUE, k=3) # k is the number of dim
fit3d # view results

#Plot in 3d
install.packages("rgl")
library(rgl)
plot3d(fit3d$points, col="dodgerblue2", size=4, pch=19, xlab="Dimension 1", ylab="Dimension 2",
zlab="Dimension 3", main="3d Visualization of Florida Lakes MDS")
text3d(fit3d$points, texts=row.names(rec.df.mds), cex=0.6, col="dodgerblue2")

###
###
library(MASS)
d2 <- dist(rec.df.mds1)
fit2 <- isoMDS(d2, k=2) # k is the number of dim
fit2 # view results

# plot solution
par(mar = c(5,5,3,1))
x <- fit2$points[,1]
y <- fit2$points[,2]
plot(x, y, col = ifelse(recidivism.df$black <= 0,'blue','green'),  xlab="Dimension 1", ylab="Dimension 2",main="Non-
Metric MDS")
legend("topright",
       legend = c("Black", "Non-Black"),
       col = c("green",
               "blue"),
       pch = 1,
       bty = "n",
       pt.cex = 1,
       cex = 1,
       text.col = "black",
       horiz = F ,
       inset = c(0.1, 0.1))

library(focusedMDS)
focusedMDS(d2, ids=row.names(rec.df.mds1))
```

```
install.packages("tidyverse")
install.packages("scales")
install.packages("magrittr")
install.packages("dplyr")
install.packages("corrplot")
library(corrplot)
library(magrittr)
library(dplyr)
library(plyr)
library(psych)
library(tidyverse)
require(tidyverse)
require(kohonen)
require(ggplot2)
require(ggridges)

find_node_by_coordinates <- function(x, y, grid_width) {
  return(((y * grid_width) + x) - grid_width)
}

# Shane Lynn 14-01-2014 used to define the palette
coolBlueHotRed <- function(n, alpha = 1) {
  rainbow(n, end=4/6, alpha=alpha)[n:1]
}
#################################

college.df <- file.choose()
college.df <- read.csv(college.df,header=TRUE)

multi.hist(college.df)
summary(college.df)
dim(college.df)
unique(college.df$gre)
table(college.df$admit)
sapply(college_acceptance, function(x) sum(is.na(x)))
coll_cor <- cor(college.df)
corrplot(coll_cor, method = "color", outline = T, addrect = 4, rect.col = "black", rect.lwd = 3,cl.pos = "b", tl.col =
"black", tl.cex = 1, cl.cex = 1, addCoef.col = "black", number.digits = 1, number.cex = 1, col =
colorRampPalette(c("purple","white","dark green"))(100))

apply(college.df, 2, function(x) length(unique(x)))
levels(college.df$admit) <- c('Not Admitted', 'Admitted')

gre <- college.df$gre
admit <- college.df$admit
rank <- college.df$rank
gpa <- college.df$gpa

gre
admit
rank
gpa

admitted <- college.df[college.df$admit == 1,]
not_admitted <- college.df[college.df$admit == 0,]
```

```
str(college.df)
# review the distributions of normal and fraudulent applications
admitted %>%
  gather(variable, value, -admit) %>%
  ggplot(aes(y = as.factor(variable),
             fill = admit,
             x = percent_rank(value))) +
  geom_density_ridges() +
  ggtitle('Distributions of Admitted') +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab('Value') +
  ylab('Variable')

not_admitted %>%
  gather(variable, value, -admit) %>%
  ggplot(aes(y = as.factor(variable),
             fill = admit,
             x = percent_rank(value))) +
  geom_density_ridges() +
  ggtitle('Distributions of Not_Admitted') +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab('Value') +
  ylab('Variable')


find_grid_size <- function(N) {
  return(floor(sqrt(sqrt(N) * 3)))
}

get_node_counts <- function(x) {
  df <- data.frame(node = x)
  counts <- df %>%
    group_by(node) %>%
    summarize(observations = n())
}

########normalize
college.df$admit <- as.numeric(college.df$admit)
college.df$gre <- as.numeric(college.df$gre)
college.df$gpa <- as.numeric(college.df$gpa)
college.df$rank <- as.numeric(college.df$rank)

college.df_norm <- normalize(college.df)
gre_norm <- college.df_norm$gre
admit_norm <- college.df_norm$admit
rank_norm <- college.df_norm$rank
gpa_norm <- college.df_norm$gpa

str(college.df_norm)

######

epochs = 2000
set.seed(123)
map_dimension = find_grid_size(dim(college.df_norm)[1])
som_grid = somgrid(xdim = map_dimension
                   ,ydim = map_dimension
                   ,topo = "rectangular")
```

```
college.sc = scale(college.df)
college.som <- supersom(college.sc, grid=som_grid, rlen=epochs, alpha=c(0.05,0.01), keep.data = TRUE)
summary(college.som)

college.som$unit.classif
observations_by_node <- get_node_counts(college.som$unit.classif)

college.som$unit.classif

par(mar = c(5,5,3,1)) #no of plots to combine
plot(college.som, type="changes", palette.name = coolBlueHotRed)
plot(college.som, type="count", palette.name = coolBlueHotRed)
plot(college.som, type="codes", palette.name = coolBlueHotRed)
plot(college.som, type = "dist.neighbours", palette.name = coolBlueHotRed)
```