STAT 5630 - Statistical Machine Learning

due tuesday december 3, 2019

HW 3

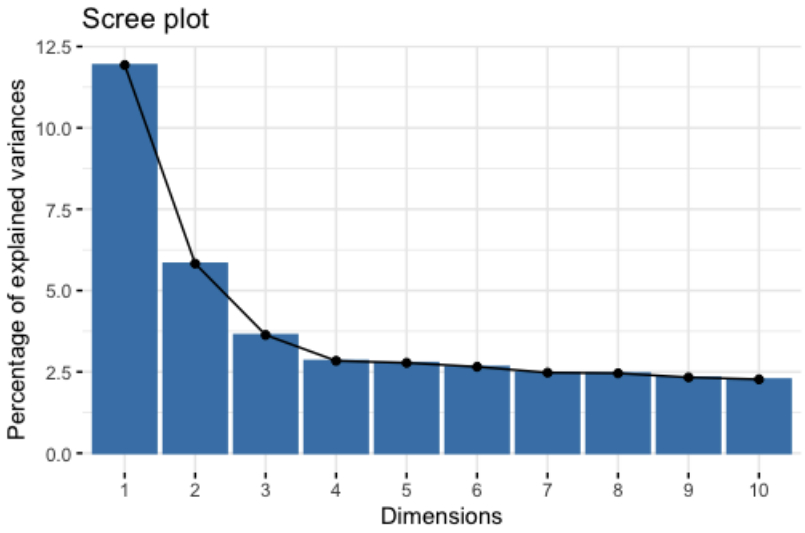
Max Ryoo (hr2ee)

Professor: Xiwei (Denny) Tang

problem 1 *Dimension Reduction and Clustering*

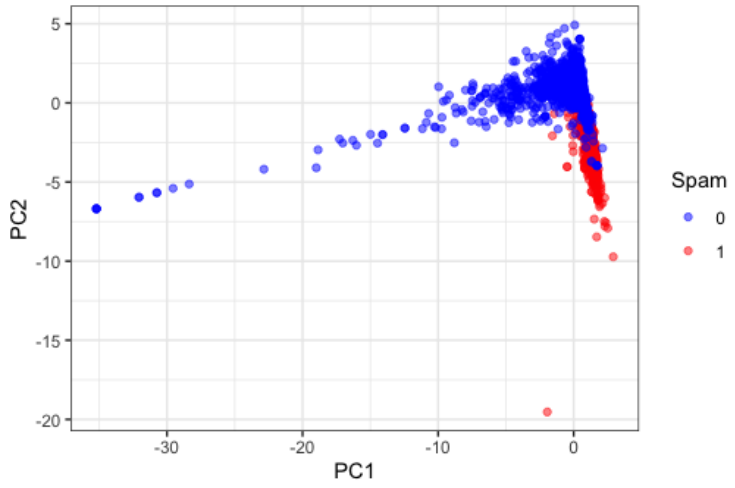
(a) Dimension Reduction

(a) Upon doing dimensionality reduction of the data set of 57 attributes and 1 response variable, the first 43 attributes after running PCA were chosen to use with the data. The percent of explained variances shows as follows in the graphical representation.



The criteria for choosing 43 attributes was because if 43 variables is chosen then 90% of variance is preserved. Although it is inevitable to lose variance using any sort of dimensionality reduction it was aimed to preserve as much as possible, which was why the value 90% was chosen as the cut off for variance preservation

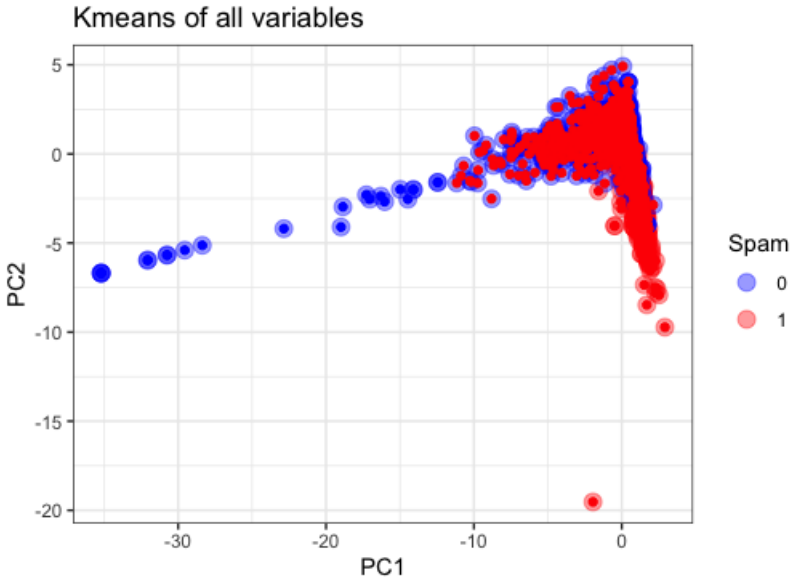
(b) The dataset is shown below with a seperation of spam and non spam.



The graph above shows spam and not spam with the colors of blue for non spam and red for spam emails. The x axis is the first principle component and the y axis is the second principle component. As one can see there are lots of overlaps in the right top hand corner and the distinction of spam to not spam is hard to see in that specific corner. Although many different axis and components were tried this selection was the most appealing to the visual eye. This may lead to the hypothesis that maybe this dataset is not easily separable in determining spam or not spam using a two dimensional plane.

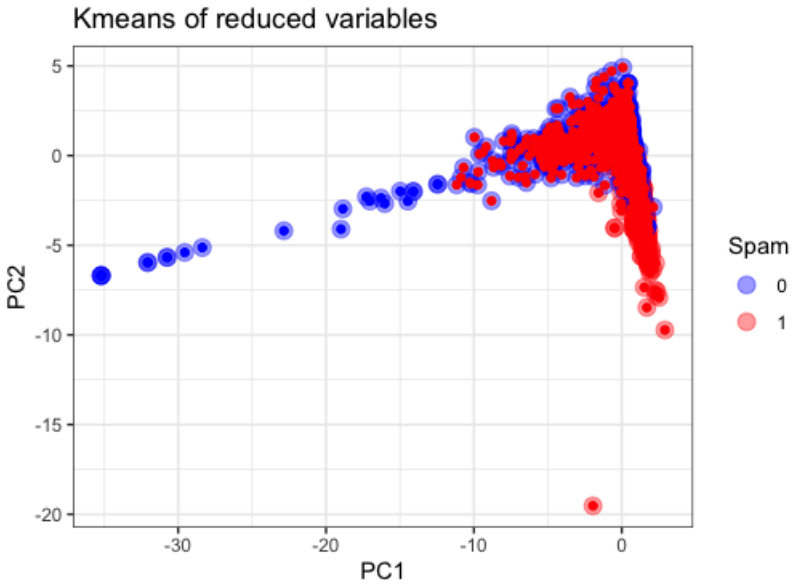
(c) **K-means**

i. All 57 variables of the dataset ...
The settings for running kmeans was that the centeroid numbers for running kmeans is 2 clusters and the the 'nstart' (number of random sets chosen) was set to the default of 20. The following graphical representation is of the kmeans clustering. The points are the clustering results while the shading outside is the true clusters.



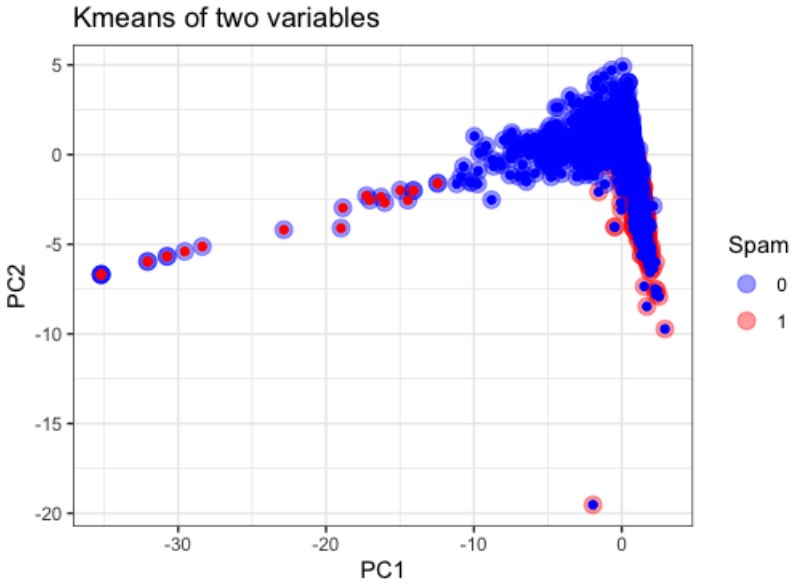
As one can see the kmeans using all attributes of the dataset over classified the spam datasets. Many of the original blue dots have become red (blue outline with red filling). However, since the plotting of the points is so dense in a small area it is hard to clearly see exactly how it is separated, but from a glance it seems to not be clustering well in comparison to the plot we have seen in in part (b).

ii. With reduced variables in part (a), which is in our case 43 variables ...
The settings for running kmeans is exactly the same as before. The only difference is the dataset that we inputted, which for our case is the data set that was reduced to contain 43 components. The following graphical representation is of the kmeans clustering. The points are the clustering results while the shading outside is the true clusters.



The pattern we saw with all the variables is again showing up in the reduced dataset. It could just be that the top level is changing, to blue or red, but from a visual glance it seems to follow the trend of the kmeans algorithm completed on all the variables.

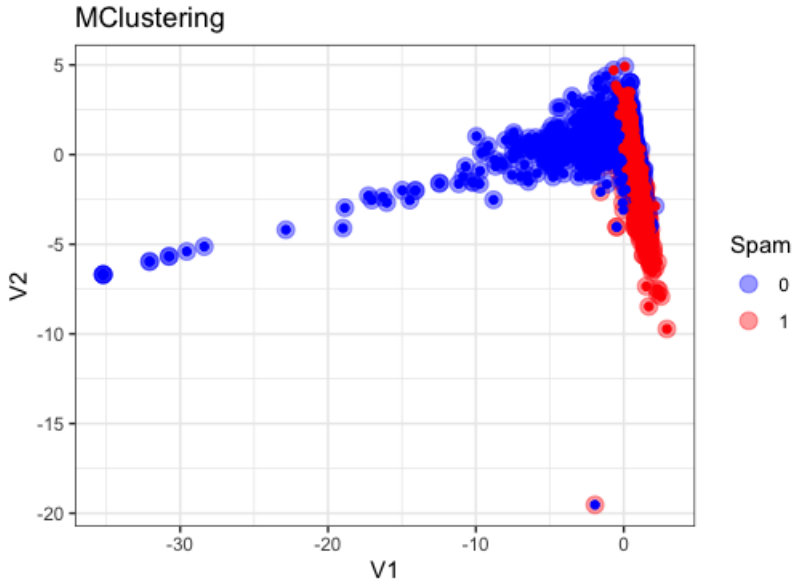
iii. With two dimensions that are used in part (b) ...
The setting for running kmeans is exactly the same as before. The only difference is the dataset that was used, which for this case was the dataset only containing two variables needed to graph in part (b). The following graphical representation is of the kmeans clustering. The points are the clustering results while the shading outside is the true clusters.



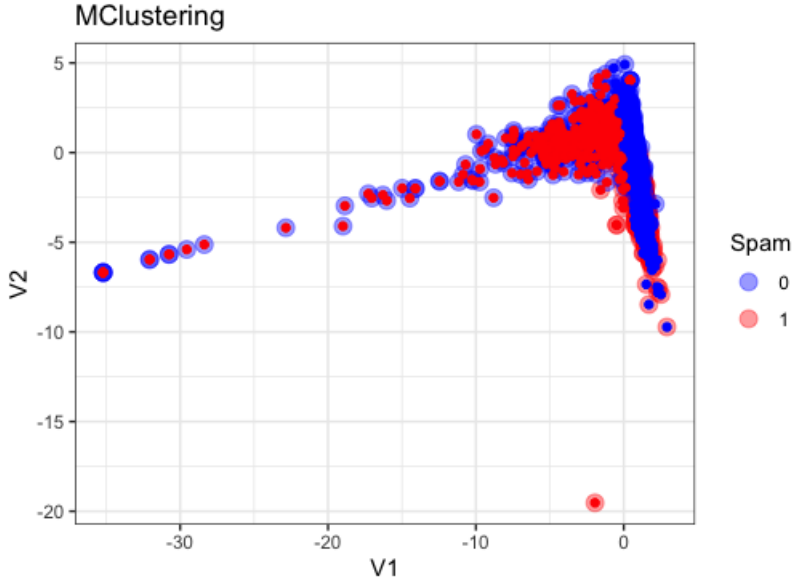
There is a very big difference between the two variables and reduced variables dataset. For the very populated area in the graphical representation it seems to be that the clustering is correctly classifying the class. However, in the contrary the spam data points are being classified as non spam. So interestingly enough for the reduced dataset it seems to be that the algorithm

is over classifying the non spam data points. However, it is not safe to say the reduced dataset is better. Interestingly, if the seed is set for different number the graph will show something similar to the above graph of the full dataset. This is partly due to the randomized selection of centroids at the start of the kmeans clustering algorithm.

(d) **Two - componenet mixture of normal model using two dimensions**
For carrying out a clustering method based on a two-component mixture of normal model using two dimensions of part (b), the function mclust was used. The 'G' variable (number of mixture components) was set to two for this question. Interestingly enough there were two different graphs that were yielded that had drastically different results. The first 'correct' graph is shown below.



In comparison to the other graphs shown it has been the best. However, even after setting a seed of 1 the classification can change every time someone runs the function. The other result that was achieved is as follows.



This graph is as if the classifications are changed. This could be due to the fact that the membership can be heavily influenced based on the random centroids

that were assigned in the very beginning.

(e) **Comparison**
All the graphs shown above have a shading of the true clusters. The outer ring of each point represents the true cluster and the shading inside represents the cluster that the kmeans clustering algorithm clustered the point into. While working with this dataset and the different kmeans algorithm (different input datasets), it became clear to me that the kmeans clustering algorithm has a high reliance on chance for datasets that have no clear distinction of clusters. By looking at the two different graphs for mclust algorithm (two-component mixture of normal model with two dimensions) one can see that although the settings are the same the outputs are very much different if not the opposite. Throughout the procedure the thought was that this dataset is not compatible with clear clustering and non of the methods have a clear insight to the differentiating factor of spam and not spam datasets. This could also be because we have been looking at two dimensional representations. If we calculated and disregarded the two dimensional representation it could yield a higher result. If a choice has to be made from the 4 methods above, I would personally choose the mclustering method. It has been the closest so far to the true clustering as shown in part (b) of the assignment. Also since the classification seems to be flipped sometimes (spam to non spam and non spam to spam) if this happens switching the classification title is also a possibility since both results have the same pattern.

```
Code used for Assignment 3...
library(e1071)
library(ElemStatLearn)
library(kernlab)
library(gbm)
library(MASS)
library(glmnet)
library(kknn)
library(class)
library(deldir)
library(lars)
library(PerformanceAnalytics)
library(leaps)
library(ncvreg)
library(party)
library(rpart)
library(tree)
library(ipred)
library(rpart)
library(randomForest)
library(ggplot2)
library(caret)
library(graphics)
library(factoextra)
library(mclust)


set.seed(1)

setwd("/Users/maxryoo/Documents/Fall 2019/STAT 5630/Code/hw_last")
train = read.table("traindata.txt")
```

```
pca = prcomp(train[1:57], scale = TRUE)
plot(pca)
fviz_eig(pca)
summary(pca)
colnames(train)
cum_prop <- cumsum(pca$sdev^2 / sum(pca$sdev^2))
last_index = min(which(cum_prop > 0.90))
dr_data <- as.data.frame(pca$x[,1:last_index])
plot <- cbind(dr_data[,1],dr_data[,2], train$V58)
colnames(plot) <- c("PC1", "PC2", "Class")
ggplot(as.data.frame(plot)) +
  geom_point(aes(PC1,PC2, color=factor(Class)), alpha=0.5) +
  scale_color_manual(values=c("blue", "red")) + theme_bw() + labs(color="Spam")

## all variables
spam.kmeans.all <- kmeans(pca$x, centers = 2, nstart = 20, trace = TRUE)
spam.kmeans.all$centers
all_variables = cbind(as.data.frame(pca$x), train$V58)
colnames(all_variables) <- c(colnames(pca$x), "Class")
ggplot(all_variables, aes(PC1, PC2, color=factor(Class))) +
  geom_point(alpha = 0.4, size = 3.5) + # true cluster
  geom_point(col = c("blue", "red")[spam.kmeans.all$cluster]) +
  scale_color_manual(values = c("blue", "red")) + theme_bw() +
  ggtitle("Kmeans of all variables") + labs(color="Spam")

spam.kmeans.dr <- kmeans(dr_data, centers = 2, nstart = 20, trace = TRUE)
spam.kmeans.dr$centers
dr_variables = cbind(dr_data, train$V58)
colnames(dr_variables) <- c(colnames(dr_data), "Class")
ggplot(dr_variables, aes(PC1, PC2, color=factor(Class))) +
  geom_point(alpha = 0.4, size = 3.5) + # true cluster
  geom_point(col = c("blue", "red")[spam.kmeans.dr$cluster]) +
  scale_color_manual(values = c("blue", "red")) + theme_bw() +
  ggtitle("Kmeans of reduced variables") + labs(color="Spam")

spam.kmeans.two <- kmeans(plot[,1:2], centers = 2, nstart = 20, trace = TRUE)
spam.kmeans.two$centers
ggplot(as.data.frame(plot), aes(PC1, PC2, color=factor(Class))) +
  geom_point(alpha = 0.4, size = 3.5) + # true cluster
  geom_point(col = c("blue", "red")[spam.kmeans.two$cluster]) +
  scale_color_manual(values = c("blue", "red")) + theme_bw() +
  ggtitle("Kmeans of two variables") + labs(color="Spam")



#
?Mclust
summary(Mclust(as.data.frame( cbind(dr_data[,1],dr_data[,2])), G =2),
        parameter=TRUE)
plot(Mclust(as.data.frame( cbind(dr_data[,1],dr_data[,2])), G =2),
     what="classification")
mclustering = Mclust(as.data.frame( cbind(dr_data[,1],dr_data[,2])), G =2)
data_mclust = mclustering$data
data_mclust_class = mclustering$classification
```

```
plot_mclust <- cbind(data_mclust, data_mclust_class, train$V58)
colnames(plot_mclust) = c("V1", "V2", "Class", "True")
ggplot(as.data.frame(plot_mclust), aes(V1, V2, color=factor(True))) +
  geom_point(alpha = 0.4, size = 3.5) +
  scale_color_manual(values = c("blue", "red")) +
  geom_point(col = c("blue", "red")[as.data.frame(plot_mclust)$Class]) + theme_bw() +
  ggtitle("MClustering") + labs(color="Spam")


ggplot(as.data.frame(plot_mclust), aes(V1, V2, color=factor(Class))) +
  geom_point() +    scale_color_manual(values = c("blue", "red"))


as.data.frame(plot_mclust)$Class
as.data.frame(plot_mclust)$True
#Normal mix
```