Landsat Satellite Image Classification

Rachel Wang & Masuzyo Mwanza

20 April 2019

Auburn University

Abstract:

The dataset is Statlog (Landsat Satellite) Data Set from UCI. The dataset consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image. Each pixel contains four levels of scene in different spectral bands. For each image, a number indicating the classification label of the central pixel. Our goal is to find a model that can recognize the groups. And check if the built model can give convincible predict. The dataset contains 4435 observation (image) in the training set, and 2000 observation in the testing set. In this project, we use multivariate method to classify the groups of each observation and compare the result with several prediction methods.

According to our analysis, random forest reaches the lowest error rate (0.0613) among all classification methods.

Key words: Landsat, Principle Component Analysis, Random Forest, Support Vector Machine, Error Rate, Linear Discriminant Analysis, Scree Plot, Scatter Plot.

Table of Contents:

1. Introduction

The Landsat satellite data is one of the many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions including multispectral and radar data, maps indicating topography, land use etc. is expected to assume significant importance with the onset of an era characterized by integrative approaches to remote sensing (for example, NASA's Earth Observing System commencing this decade).

The dataset consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image. Within each pixel, it consists of four digital images of the same scene in different spectral bands. Two of them on visible region, and other two on(near) infra-red. Therefore, there are 36 (9 pixels * 4) variables in this dataset. There are 7 number in total to indicating classification groups, which are 1: red soil, 2: cotton crop, 3: grey soil, 4: damp gray soil, 5: soil with vegetation stubble, 6: mixture class (all type presents), 7: very damp soil. Among these 7 groups, there is no groups 6 since it has been removed by doubts about the validity of this class. Therefore, there are 6 indication groups, 1,2,3,4,5, and 7.

In this dataset, we name 36 variables as V1-V36, within them, 17, 18, 19, and 20 are indicate the four level of scene of the central pixel.

2. Statistical Approach

Based on means vector, we can use multivariate analysis of variance method to compare mean vectors arranged according to treatment level, which is MANOVA. After that, we tried Linear Discriminate analysis (LDA). In order to use MANOVA and LDA, we first decide to check the assumption.

After that, we decide to apply other method that doesn't need assumptions on the dataset, which are Principle Component Analysis (PCA), Random Forest (RF), and Support Vector Machine (SVM). After the PCA method, we are getting dimension reduction version of variables, then use these "new" variables to run the LDA, RF and SVM again.

In addition to previous test, we found that group 3, 4, and 7 are indicate different levels of grey, which makes the model hard to classify the groups. We create 2 stages for to analysis this case. Stage 1 is to combine these groups together denote as group "3" and run the test. Stage 2 is run another test to classify these three groups.

The following table gives scatter plot with correlation of first 9 variables.

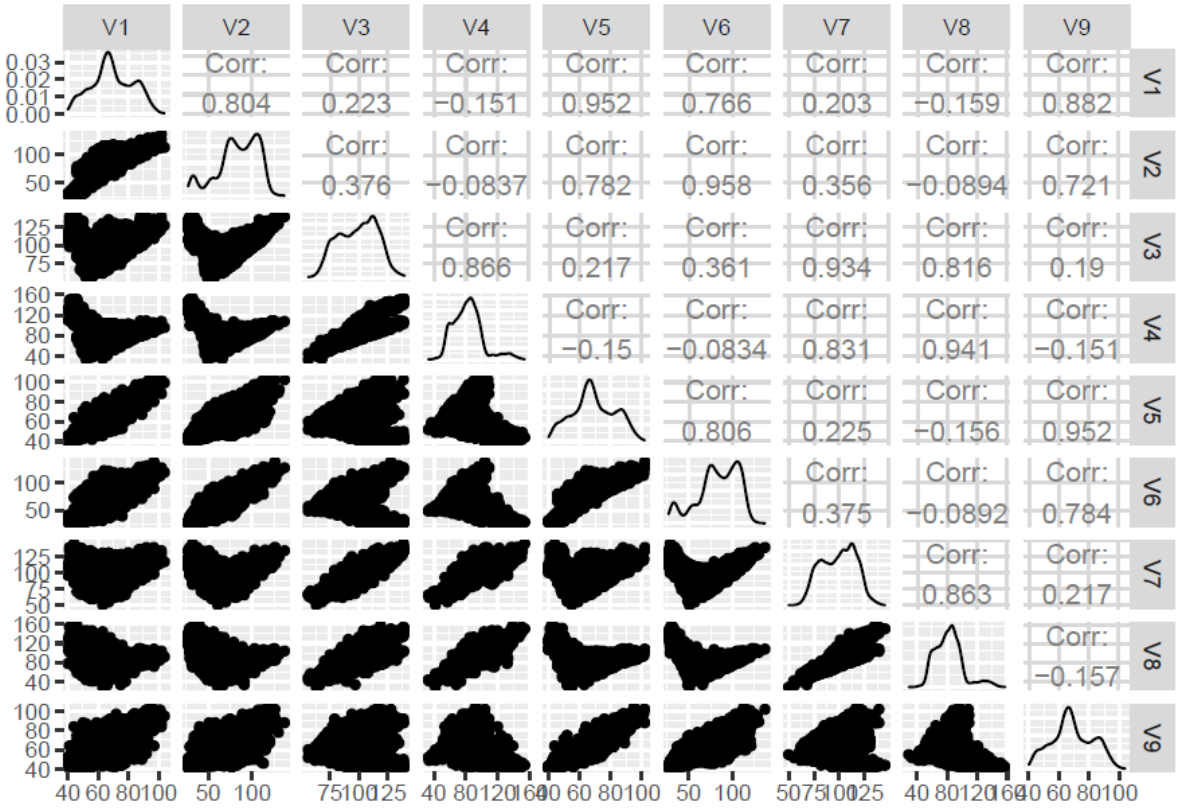

Table 1: Scatter Plot with Correlation Value

3. Result

3.1 MANOVA

```
##             Df Pillai approx F num Df den Df    Pr(>F)
## res          5 2.4502   117.39    180  21990 < 2.2e-16 ***
## Residuals 4429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig.1: MANOVA Result

Based on the MANOVA output, we can see there are significant different between 6 groups. In order to check is this result is reliable, we run a test on normality.

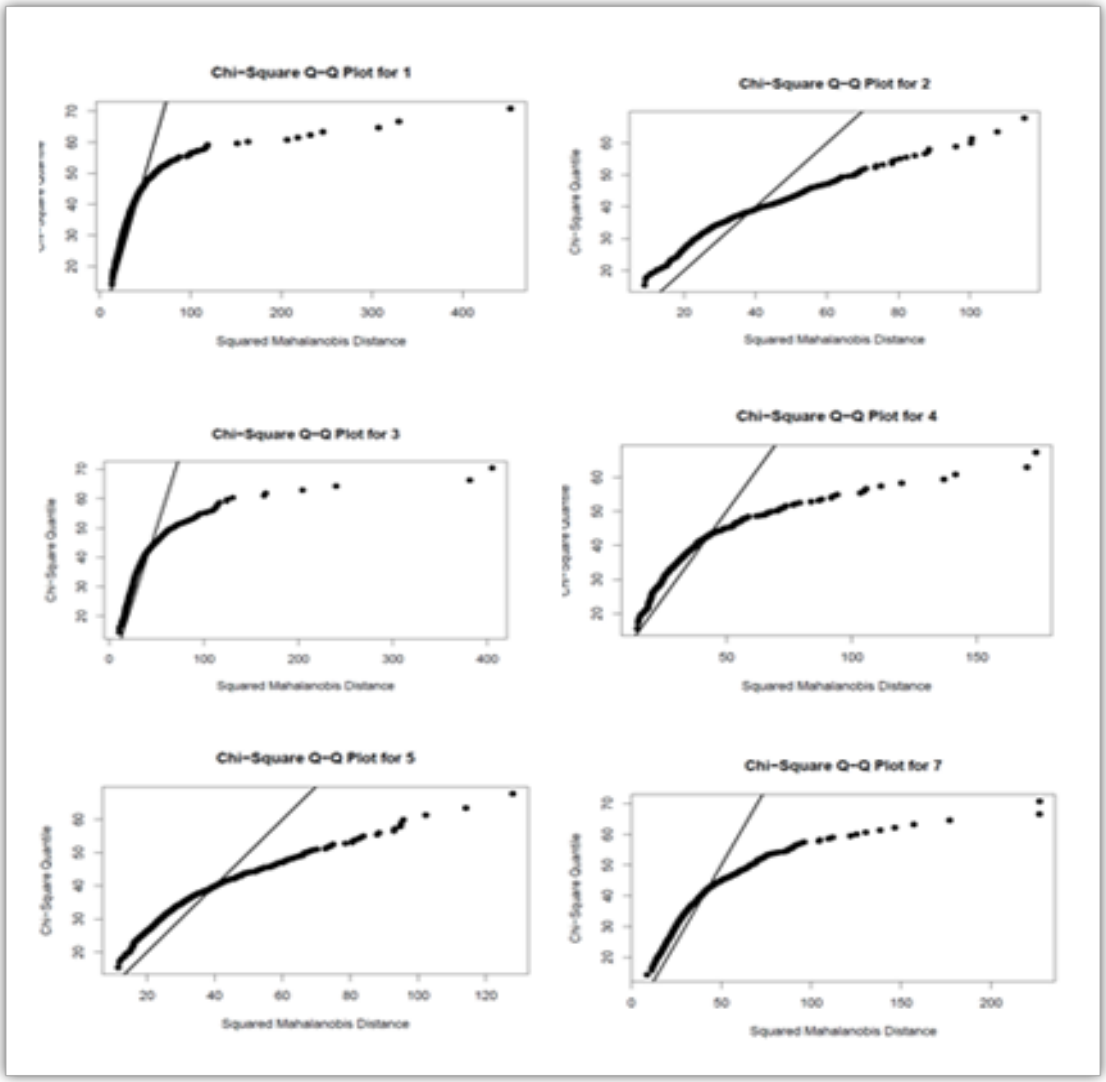

Fig. 2: Chi-Square QQ-Plot for 6 Groups

Based on the Chi-Square QQ-Plot we can clearly see that all the groups are not follow normal distribution. Therefore, we cannot use MANOVA and LDA on this dataset. This might because of not enough number of observations, or there is a hidden distribution that we don't know.

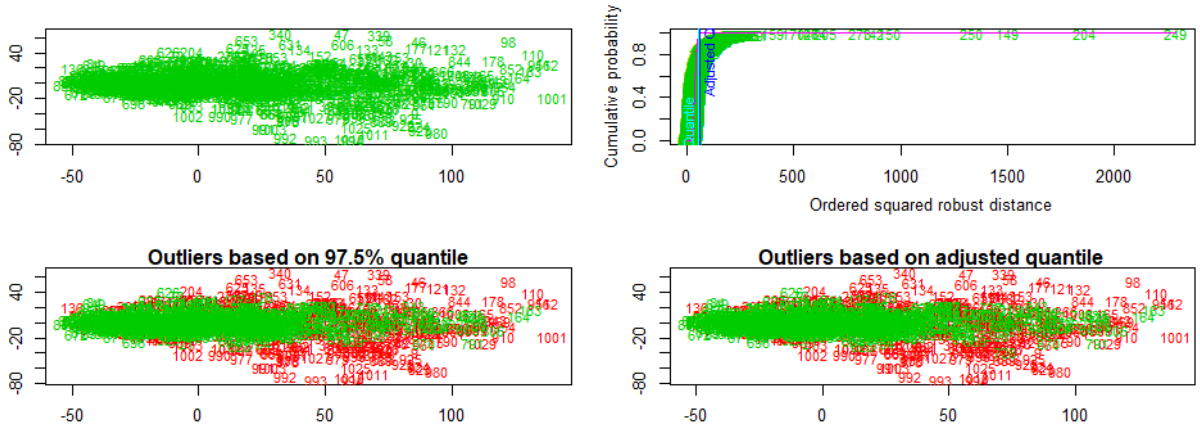

Fig. 3: Outlier for Group 7

For the test of outlier plot, we could see with adjusted MD, there are a lot of outliers hanging around. The rest groups' plots are like group 7.

3.2 Linear Discriminate Analysis

	res1					
lda.class	1	2	3	4	5	7
1	450	1	2	0	6	0
2	0	197	0	0	1	0
3	7	1	372	54	3	24
4	1	1	20	62	9	35
5	1	23	0	3	168	3
7	2	1	3	92	50	408

Table 2: Confusion Matrix of LDA

Table 2 shows the confusion matrix on the test set by using LDA model. The error rate is 0.1715. Since the normality test doesn't satisfy, this result is not reliable.

3.3 Random Forest

```
##         res1
## rf.pred   1   2   3   4   5   7
##       1 459   0   4   0   7   0
##       2   0 219   0   0   3   0
##       3   0   1 375  34   1  11
##       4   0   1  12 134   1  28
##       5   2   1   1   2 210  12
##       7   0   2   5  41  15 419
```

Table 3: Confusion Matrix of RF

Table 3 shows the confusion matrix on the test set by using Random Forest model. The error rate is 0.092.

3.4 Support Vector Machine

```
##         res1
## svm.pred   1   2   3   4   5   7
##        1 459   0   3   0   7   0
##        2   1 217   1   1   4   0
##        3   1   0 383  40   0  16
##        4   0   1   5 123   3  39
##        5   0   4   1   2 204  10
##        7   0   2   4  45  19 405
```

Table 4: Confusion Matrix of SVM

Table 4 shows the confusion matrix on the test set by using RF model. The error rate is 0.1045.

3.5 Principle Component Analysis

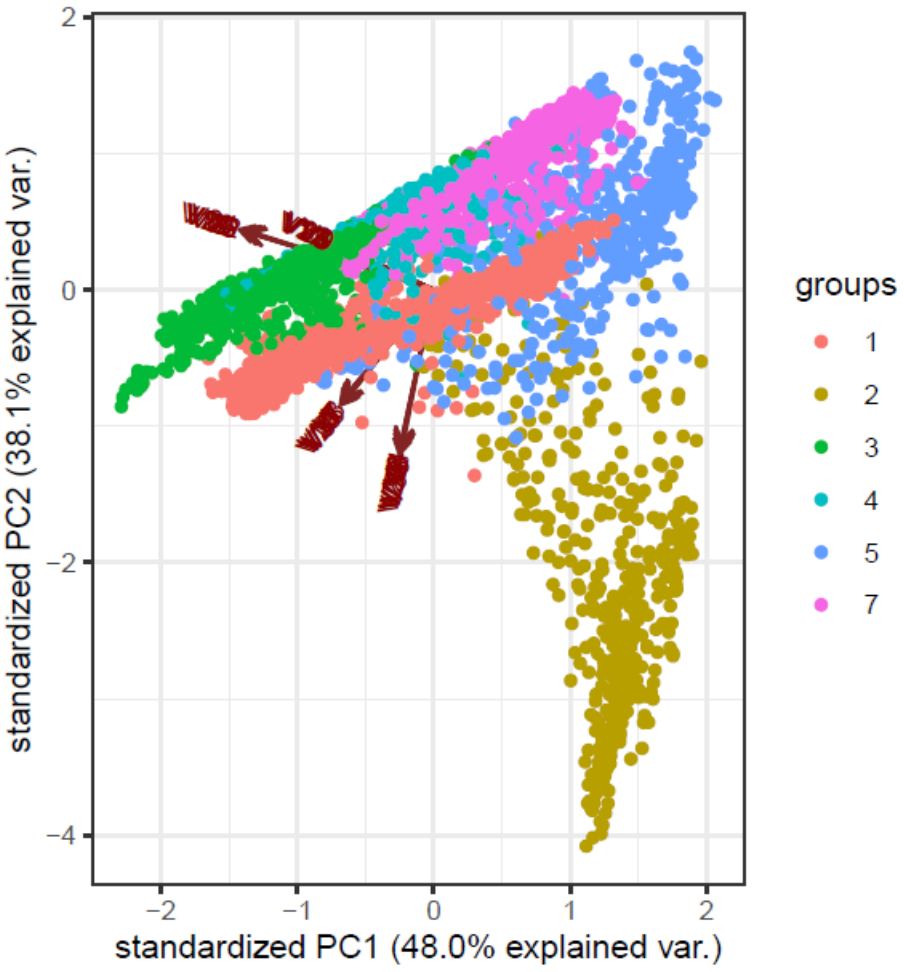

Fig. 4: Principle Component Analysis Plot

Based on the PCA plot, we can see that group 3,4, and 7 are mixed and overlap on top of each other.

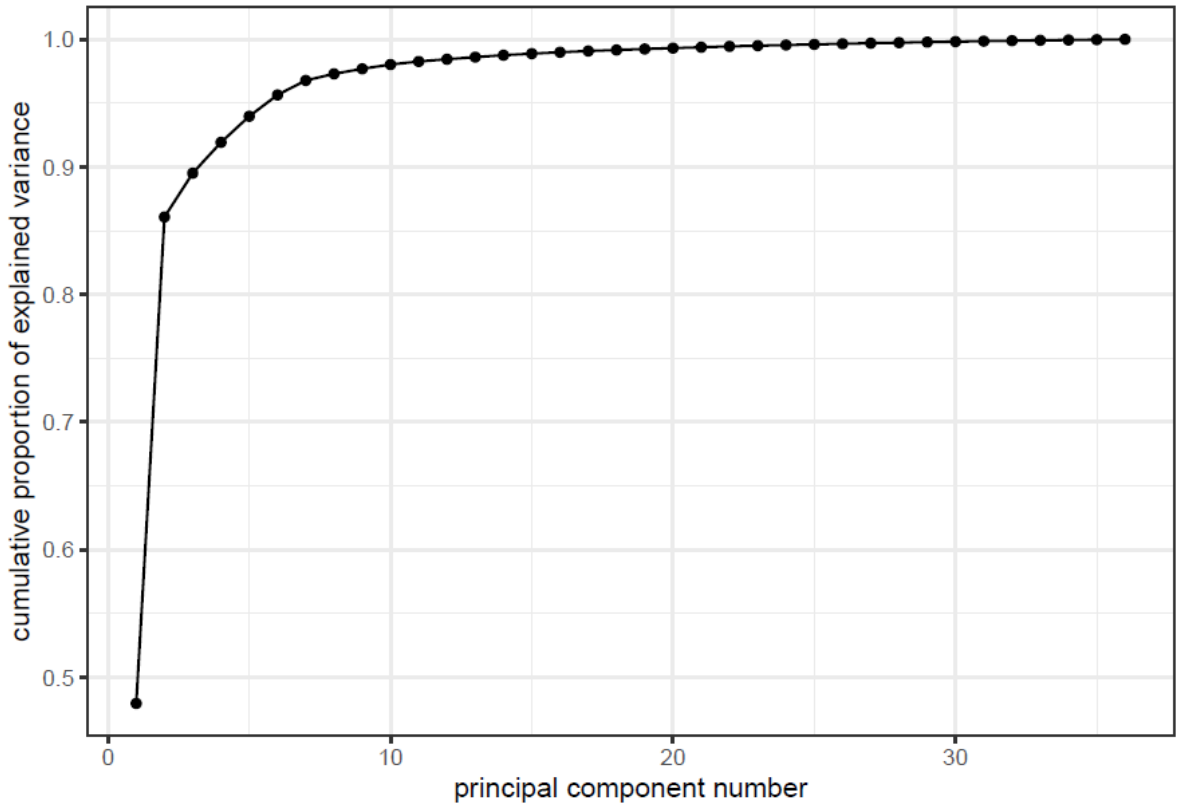

Fig. 5: PCA Scree Plot

From the scree plot, we decide to use 5 PCs to represent the dataset. This gives about 95% of the data.

3.5 LDA - With PCA output

	res1					
lda.class	1	2	3	4	5	7
1	447	0	0	0	10	0
2	0	195	0	0	3	0
3	6	2	374	61	4	23
4	4	3	18	44	12	37
5	2	23	0	4	142	6
7	2	1	5	102	66	404

Table 5: Confusion Matrix of LDA-With PCA output

Table 5 shows the confusion matrix on the test set by using LDA model. The error rate is 0.197.

3.5 RF - With PCA output

##	res1					
## rf1.pred	1	2	3	4	5	7
## 1	454	0	3	0	8	0
## 2	0	215	1	2	9	1
## 3	1	1	363	33	3	16
## 4	2	4	24	134	4	38
## 5	4	4	1	4	192	11
## 7	0	0	5	38	21	404

Table 6: Confusion Matrix of RF-With PCA output

Table 6 shows the confusion matrix on the test set by using RF model. The error rate is 0.119.

3.5 SVM - With PCA output

##	res1					
## svm1.pred	1	2	3	4	5	7
## 1	458	0	2	0	11	0
## 2	1	216	1	3	5	1
## 3	2	1	380	42	2	16
## 4	0	1	6	119	1	33
## 5	0	5	1	4	194	11
## 7	0	1	7	43	24	409

Table 7: Confusion Matrix of SVM-With PCA output

Table 7 shows the confusion matrix on the test set by using SVM model. The error rate is 0.112.

How many PCi do we really need?

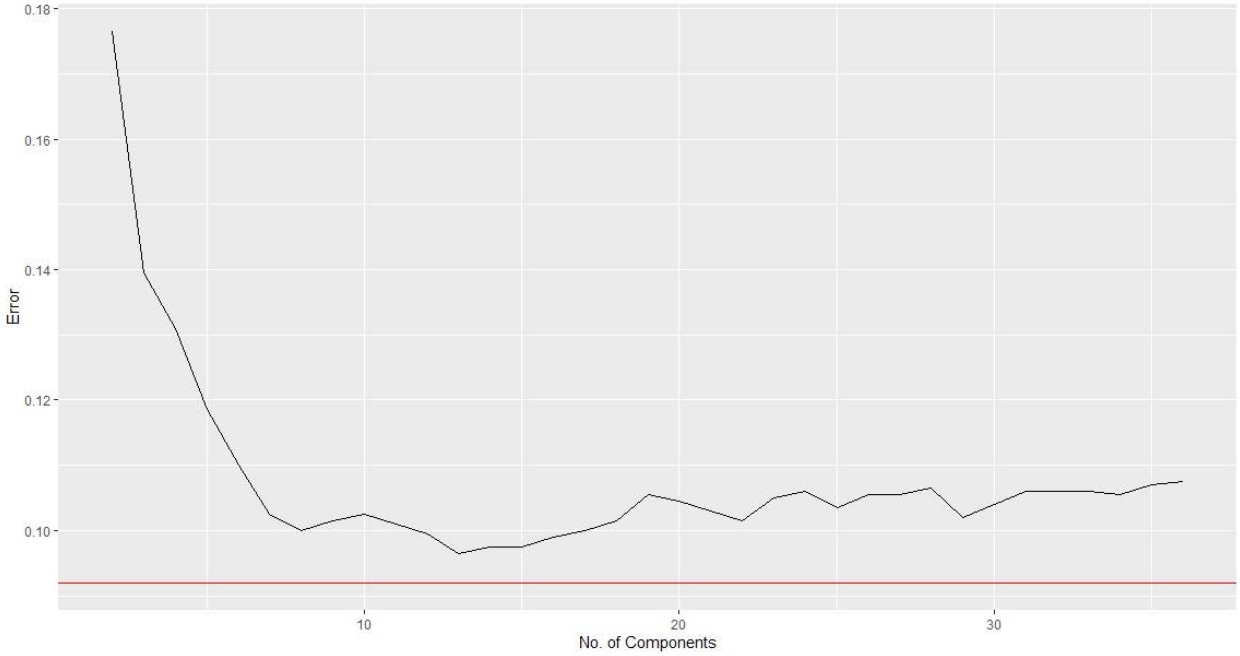

Fig. 6: Number of PC Selection

With help on random forest, we could see that in order to improve the PCA result, we need to include about 10 to 15 numbers of PC.

With 15 PCs output, we run random forest again. The error rate turns out to be 0.097.

3.6 Two Stage Analysis - Stage 1

We run the LDA, RF, and SVM test on the new group that we created.

```
##           res3                          ##         res3                          ##          res3
## lda.class    1    2    3    5          ## rf.pred     1    2    3    5          ## svm.pred     1    2    3    5
##         1  449    1    2    4          ##        1  457    0    4    8          ##         1  459    0    3    7
##         2    0  196    0    1          ##        2    1  220    0    3          ##         2    1  218    1    4
##         3   10    3 1070   60          ##        3    2    3 1065   22          ##         3    1    3 1065   28
##         5    2   24    6  172          ##        5    1    1    9  204          ##         5    0    3    9  198
```

Table 8: Confusion Matrix of Stage 1 on LDA, RF, and SVM Test

From the three test we have testing error for LDA, RF, and SVM are 0.0565, 0.027, and 0.03, respectfully. We can see that the random forest test gives the lowest testing error. So, we use the random forest's stage 1 output for stage two.

3.6 Two Stage Analysis - Stage 2

With the output from Random Forest, we run the LDA, RF, and SVM test again.

```
##                                  ##                                  ##
## lda.class   3   4   7            ## rf.pred   3   4   7            ## svm.pred   3   4   7
##         3 371  47  14            ##       3 375  35  11            ##       3 384  41  15
##         4  20  82  50            ##       4  12 133  28            ##       4   5 130  37
##         7   2  81 398            ##       7   6  42 423            ##       7   4  39 410
```

Table 9: Confusion Matrix of Stage 1 on LDA, RF, and SVM Test

In this case, we have testing error for LDA, RF, and SVM are 0.200939, 0.1258216, and 0.1323944, respectfully.

The overall or weighted testing error on stage 1&2 for LDA, RF, and SVM are 0.087439, 0.061338, 0.063622, respectfully. We can see the Random Forest test provides the lowest error rate.

3.7 Cluster Analysis

After previous test, we try an unsupervised method, which is cluster analysis.

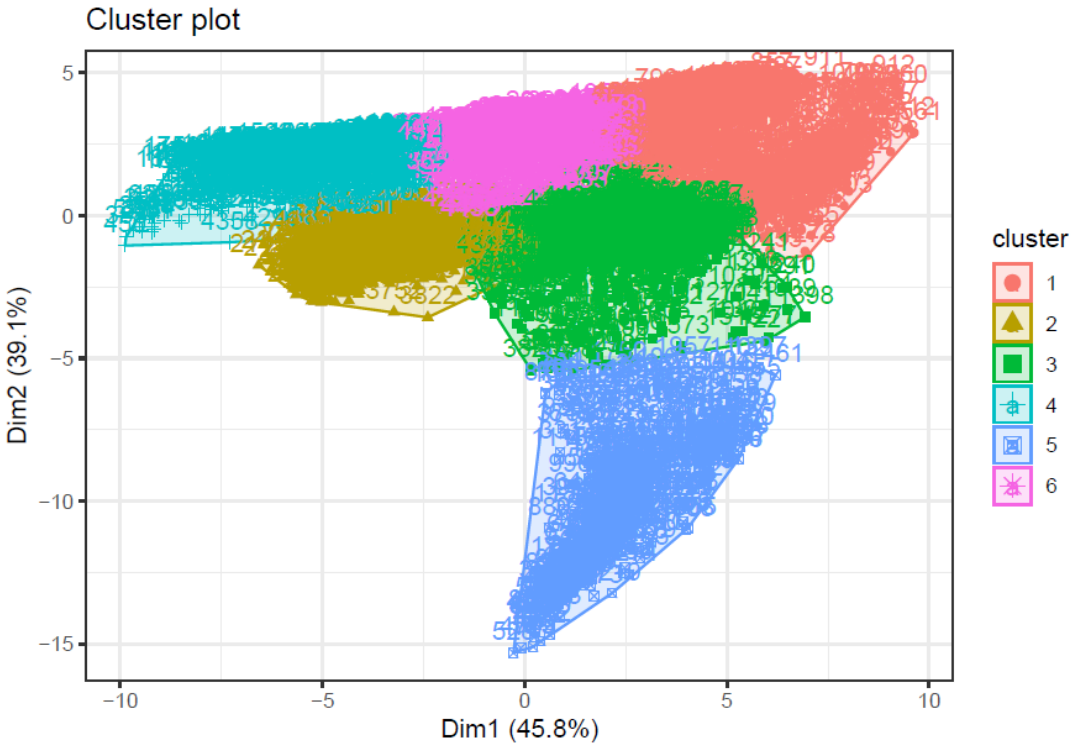

Fig. 7: Cluster Plot

The plot shows that the computer thinks the best way to separate the 6 groups, if we compare it with the PCA plot (Fig. 3), we could see that the computer fails to detect the difference between group 3, 4, and 7.

4. Discussion

For raw data, we have random forest give the lowest error rate which is 0.092.

For PCA version of the data, we have support vector machine give the lowest error rate which is 0.112. In these three results, we can see that the error rates are higher than the raw data. This due to the loss of information when doing PCA. With 15 PCs, we end up have 0.097 error rate on random forest.

For two stages analysis, we can clearly see that the model can predict group 1, 2, 5, and “3” well. The model has lowest testing error is random forest, which is 0.027. All three of them kind of fail to distinguish the difference between group 3, 4, and 7. The model that has the lowest testing error on stage 2 still be random forest, which is 0.1323944. The lowest overall/weighted error rate is 0.061338, which performed by random forest.

With these three trials of testing, we could see that the model that gives lowest prediction error is random forest on the raw data. Random forest also gives the lowest testing error on classify group 1, 2, 5, and “3”.

5. Summary

Base on the result, we can tell that random forest can performs the best result based on the tests that we done. Within all three version of tests on random forest (Raw data, with PCA, and Stage analysis), the stage analysis provides the best result, which is 0.0613 on the error rate.

6. Further Analysis

Convolutional Neural Network could be done.

7.Self-Evaluation

For this project, I mainly working on MANOVA and two stages analysis. After running the normality test and outlier test, we realize that the assumption of MANOVA will not satisfy. But we still decide to do it just to look at it even thought we know the output will not be reliable. After Mas finish the PCA and get the plot, I found out that group 3, 4 and 7 are overlapped together, which increase the hardness to classify these groups. So, I came an idea that let the computer classify another group first and then once it detects it is in new group "3", then detect which specific group it should lies in. This analysis ends up becomes the "best" result in the methods that we choose. Masuzyo is also reliable, since we use R to work on this project, and I have less knowledge in R, he helped me a lot on coding.

8. Reference

Srinivasan, Ashwin. "Statlog (Landsat Satellite) Data Set." *UCI Machine Learning Repository: Statlog (Landsat Satellite) Data Set*, 1993, archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29.

8. Code

```
library(GGally)
source("ggbiplot.r")
source("ggscreeplot.r")
source("CNN.R")
library(tidyverse)
library(factoextra)
library(randomForest)
library(MVN)
library(MASS)
library(e1071)
library(tensorflow)
library(keras)
library(magrittr)
library(mvoutlier)
set.seed(260)
train <- read.table("sat.trn",sep = " ")
test <- read.table("sat.tst",sep=" ")
ggpairs(train[,1:9])
ggpairs(train[,10:18])
ggpairs(train[,19:27])
ggpairs(train[,27:36])
pic.tr <-as.data.frame(train[,1:36])
pic.ts <- as.data.frame(test[,1:36])
res <- as.factor(train[,37])
res1 <- as.factor(test[,37])
train1 <- cbind(pic.tr,res)
mvn(train1,subset="res",multivariatePlot =
"qq")
for(i in c(1:5,7))
{
  out<-train1 %>% filter(res==i)
  aq.plot(out[,-37])
}
m1 <- manova(as.matrix(pic.tr)~res)
summary(m1)
res2 <- res3 <- NA
for(i in 1:length(res))
{
 if (res[i] %in%
c(3,4,7)){res2[i]=3}else{res2[i]=res[i]}
}
for(i in 1:length(res1))
{
 if (res1[i] %in%
c(3,4,7)){res3[i]=3}else{res3[i]=res1[i]}
}
res2<-as.factor(res2)
res3<-as.factor(res3)
pc<-prcomp(pic.tr,scale. = F)
ggbiplot(pc,groups = res)+theme_bw()
ggscreeplot(pc,type = 'cev')+theme_bw()
pca.z<-as.data.frame(pc$x[,1:5])
R<-pc$rotation[,1:5]
pca.ts <-
as.data.frame(as.matrix(scale(pic.ts,scale =
F))%*%R)
la<-MASS::lda(res~.,pic.tr)
la.pred=predict(la, pic.ts)
table(lda.class=la.pred$class ,res1)
    1-   mean(la.pred$class==res1)
    la1<-MASS::lda(res~.,pca.z)
    la1.pred=predict(la1,pca.ts)
    table(lda.class=la1.pred$class ,res1)
```

```
1-   mean(la1.pred$class==res1)

rf <-
randomForest::randomForest(res~.,pic.tr,mtr
y=20,importance=TRUE)

rf.pred=predict(rf,pic.ts)

table(rf.pred ,res1)

1-   mean(rf.pred==res1)

rf1 <-
randomForest::randomForest(res~.,pca.z,mtr
y=20,importance=TRUE)

rf1.pred=predict(rf1,pca.ts)

table(rf1.pred ,res1)

1-   mean(rf1.pred==res1)

svm<-svm(res~.,pic.tr,
kernel="radial",ranges=list(cost=c(0.1,1,10,
100,1000),gamma=c(0.5,1,2,3,4)))

svm.pred=predict(svm,pic.ts)

table(svm.pred ,res1)

1-   mean(svm.pred==res1)

svm1<-svm(res~.,pca.z,
kernel="radial",ranges=list(cost=c(0.1,1,10,
100,1000),gamma=c(0.5,1,2,3,4)))

svm1.pred=predict(svm1,pca.ts)

table(svm1.pred ,res1)

1-   mean(svm1.pred==res1)

km<-kmeans(pic.tr, centers = 6, nstart = 25)

fviz_cluster(km, data = pic.tr)+theme_bw()

err<-NA

pc<-prcomp(pic.tr,scale. = F)

for(i in 2:36)

{


  pca.z<-as.data.frame(pc$x)

  pca.i<-pca.z[,1:i]

  R<-pc$rotation[,1:i]

  pca.ts <-
as.data.frame(as.matrix(scale(pic.ts,scale =
F))%*%R)

  rf1 <-
randomForest::randomForest(res~.,pca.i,mtr
y=i,importance=TRUE)

  rf1.pred=predict(rf1,pca.ts)

  err[i-1]<-1-mean(rf1.pred==res1)

  print(i)

}

err<-NA

pc<-prcomp(pic.tr,scale. = F)

for(i in 2:36)

{


  pca.z<-as.data.frame(pc$x)

  pca.i<-pca.z[,1:i]

  R<-pc$rotation[,1:i]

  pca.ts <-
as.data.frame(as.matrix(scale(pic.ts,scale =
F))%*%R)

  rf1 <-
randomForest::randomForest(res~.,pca.i,mtr
y=i,importance=TRUE)

  rf1.pred=predict(rf1,pca.ts)

  err[i-1]<-1-mean(rf1.pred==res1)

  print(i)

}

set<-as.data.frame(cbind(k=2:36,err))

sp<-
ggplot(set,aes(x=k,y=err))+geom_line()+lab
s(x="No. of
Components",y="Error")+geom_hline(yinter
cept=0.092,linetype="dashed", color =
"red")+theme_bw()
```

```
sp

ggbiplot(pc,groups = res2)+theme_bw()

la<-MASS::lda(res2~.,pic.tr)

la.pred=predict(la, pic.ts)

table(lda.class=la.pred$class ,res3)

1- mean(la.pred$class==res3)

rf <-
randomForest::randomForest(res2~.,pic.tr,m
try=20,importance=TRUE)

rf.pred=predict(rf,pic.ts)

table(rf.pred ,res3)

1- mean(rf.pred==res3)

svm<-svm(res2~.,pic.tr,
kernel="radial",ranges=list(cost=c(0.1,1,10,
100,1000),gamma=c(0.5,1,2,3,4)))

svm.pred=predict(svm,pic.ts)

table(svm.pred ,res3)

1- mean(svm.pred==res3)
```

```
gry.tr<-train%>%filter(V37%in% c(3,4,7))

gry.ts<-
test%>%filter(rf.pred==res3&res3==3)

la<-MASS::lda(V37~.,gry.tr)

la.pred=predict(la, gry.ts[,-37])

table(lda.class=la.pred$class ,gry.ts[,37])

1- mean(la.pred$class==gry.ts[,37])

rf <- randomForest(as.factor(V37)~.,gry.tr)

rf.pred=predict(rf,gry.ts[,-37])

table(rf.pred ,gry.ts[,37])

1- mean(rf.pred==gry.ts[,37])

svm<-svm(as.factor(V37)~.,gry.tr,
kernel="radial",ranges=list(cost=c(0.1,1,10,
100,1000),gamma=c(0.5,1,2,3,4)))

svm.pred=predict(svm,gry.ts[,-37])

table(svm.pred ,gry.ts[,37])

1- mean(svm.pred==gry.ts[,37])
```