<u>Data Exploration for Alzheimer Patients Dataset</u>
<u>Phase -1</u>

Dataset Information:

The Excel spreadsheet Alzheimer.csv contains one sheet named Alzheimer, which is data attempting to explain whether a patient has Alzheimer's Disease. These are data from a sample of 336 employees and consists of 9 variables for each patient. These are:

1) Dementia-Outcome variable-patient diagnosis
2) Gender-Female=0 and Male=1
3) Age-Age of patient (in years)
4) Education-Years of Education
5) SES-Socioeconomic Status 1=Low and 5=High
6) MMSE-Mini mental state examination score
7) CDR-Clinical Dementia Rating
8) eTIV-estimated total intracranial volume
9) nWBV-Normalize whole brain volume
10) ASF-Atlas Scaling Factor

Developed a Linear Discriminant Analysis model to classify the Dementia event from the other variables.

a) Performance of the classifier using cross-validation:

```
> library(MASS)
> library(plyr)
> library(readr)
> setwd("C:/Users/DELL/Desktop/MS Assignments/Sem1/Data_Stats/Assignmnet7")
> #Read in Datasets
> Alz = read.csv("alzheimer.csv")
> View(Alz)
> #Check dimensions of Alz
> dim(Alz)
[1] 1008   10
> str(Alz)
'data.frame':   1008 obs. of  10 variables:
 $ Dementia: chr  "No Alzheimer" "No Alzheimer" "Alzheimer" "Alzheimer" ...
 $ Gender  : int  1 1 1 1 1 0 0 1 1 1 ...
 $ Age     : int  87 88 75 76 80 88 90 80 83 85 ...
 $ EDUC    : int  14 14 12 12 12 18 18 12 12 12 ...
 $ SES     : int  2 2 NA NA NA 3 3 4 4 4 ...
 $ MMSE    : int  27 30 23 28 22 28 27 28 29 30 ...
 $ CDR     : num  0 0 0.5 0.5 0.5 0 0 0 0.5 0 ...
 $ eTIV    : int  1987 2004 1678 1738 1698 1215 1200 1689 1701 1699 ...
 $ nWBV    : num  0.696 0.681 0.736 0.713 0.701 0.71 0.718 0.712 0.711 0.705 ...
 $ ASF     : num  0.883 0.876 1.046 1.01 1.034 ...
> head(Alz)
      Dementia Gender Age EDUC SES MMSE CDR eTIV  nWBV   ASF
1 No Alzheimer      1  87   14   2   27 0.0 1987 0.696 0.883
2 No Alzheimer      1  88   14   2   30 0.0 2004 0.681 0.876
3    Alzheimer      1  75   12  NA   23 0.5 1678 0.736 1.046
4    Alzheimer      1  76   12  NA   28 0.5 1738 0.713 1.010
5    Alzheimer      1  80   12  NA   22 0.5 1698 0.701 1.034
6 No Alzheimer      0  88   18   3   28 0.0 1215 0.710 1.444
> #For All Variables
> sum(is.na(Alz))
[1] 63
> #Listwise Deletion
> Alz_new <- na.omit(Alz)
> #Check new data has no missing data
> sum(is.na(Alz_new))
[1] 0
> View(Alz_new)
> head(Alz_new)
      Dementia Gender Age EDUC SES MMSE CDR eTIV  nWBV   ASF
1 No Alzheimer      1  87   14   2   27 0.0 1987 0.696 0.883
2 No Alzheimer      1  88   14   2   30 0.0 2004 0.681 0.876
6 No Alzheimer      0  88   18   3   28 0.0 1215 0.710 1.444
7 No Alzheimer      0  90   18   3   27 0.0 1200 0.718 1.462
8 No Alzheimer      1  80   12   4   28 0.0 1689 0.712 1.039
9 No Alzheimer      1  83   12   4   29 0.5 1701 0.711 1.032
> Alz_new$Dementia<- revalue(Alz_new$Dementia,c("Alzheimer"=0, "No Alzheimer"=1))
> Alz_new$Dementia<- as.factor(Alz_new$Dementia)
```

```
> View(Alz_new)
> #Graph Data
> library(psych)
> pairs.panels(Alz_new[1:10],
+             gap = 0,
+             bg = c("red", "green", "blue")[Alz_new$Dementia],
+             pch = 21)
> #Q1
> #a) What is the performance of the classifier using cross-validation?
> #With Cross Validation
> # The dependent variable must be categorical
> Alz_LDA <- lda(Dementia ~ ., data=Alz_new, CV=TRUE)
> Alz_LDA
$class
  [1] 1 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1
 [58] 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1
[115] 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1
[172] 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1
[229] 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0
[286] 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0
[343] 1 1 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 1 0 0 0
[400] 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1
[457] 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1
[514] 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1
[571] 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 0
[628] 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1
[685] 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
[742] 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0
[799] 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1
[856] 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1
[913] 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 0 1 1 1
Levels: 0 1
```

```
> #To Plot the Data, you cannot use CV
> Alz_LDA <- lda(Dementia ~ ., data=Alz_new)
> Alz_LDA
Call:
lda(Dementia ~ ., data = Alz_new)

Prior probabilities of groups:
        0         1 
0.4006309 0.5993691 

Group means:
     Gender      Age     EDUC      SES     MMSE         CDR     eTIV      nWBV      ASF
0 0.5984252 76.20472 13.82677 2.771654 24.32283 0.673228346 1490.701 0.7151811 1.192417
1 0.3210526 77.05789 15.14211 2.394737 29.22632 0.005263158 1495.500 0.7409000 1.191063

Coefficients of linear discriminants:
                 LD1
Gender -0.7749657489
Age     0.0170393843
EDUC    0.0525355103
SES    -0.0502323960
MMSE   -0.0265038084
CDR    -5.1051949423
eTIV    0.0001057809
nWBV    4.5687180022
ASF    -1.9933545065
> p <- predict(Alz_LDA, newdata=Alz_new[,1:10])$class
```

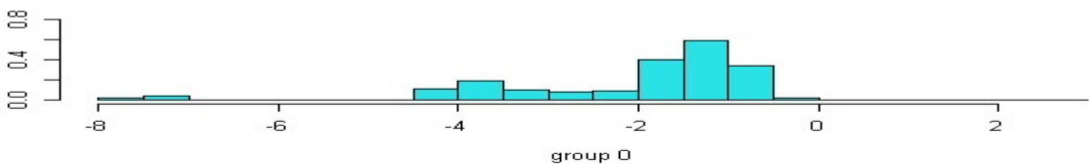
```
> table_1 <- table(p, Alz_new$Dementia)
> table_1

p     0   1
  0 378   6
  1   3 564
> sum(diag(table_1)/sum(table_1))
[1] 0.9905363
> accuracy <- (378+564)/(378+564+6+3)
> accuracy
[1] 0.9905363
```

The accuracy of found from the model is approximately 99% while using corresponding analysis. This shows that it will 99% times it will tell you correctly if a person has dementia or not.

b) Performance of the classifier using training and testing:

```
> require(caTools)  # loading caTools library
> library(caTools)
> set.seed(123)
> sample = sample.split(Alz_new,SplitRatio = 0.70)
> train =subset(Alz_new,sample ==TRUE)
> test=subset(Alz_new, sample==FALSE)
> # The dependent variable must be categorical (Assuming No Cross-Validation)
> Alz_LDA = lda(Dementia ~ ., data=train)
> Alz_LDA
Call:
lda(Dementia ~ ., data = train)

Prior probabilities of groups:
        0         1 
0.4009009 0.5990991 

Group means:
     Gender      Age     EDUC      SES     MMSE         CDR     eTIV      nWBV      ASF
0 0.5692884 76.51685 13.82772 2.749064 24.42697 0.672284644 1485.614 0.7154569 1.196749
1 0.3182957 77.35088 15.11028 2.413534 29.23308 0.006265664 1497.203 0.7399298 1.189356

Coefficients of linear discriminants:
                LD1
Gender -0.720747343
Age     0.022923539
EDUC    0.070927518
SES    -0.008742151
MMSE   -0.011730650
CDR    -4.630819629
eTIV   -0.001341651
nWBV    5.018157340
ASF    -3.899604691
```

```
> plot(Alz_LDA)
> prd<-predict(Alz_LDA, train)$class
> prd
  [1] 1 1 1 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 1 1 1 1
 [67] 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
[133] 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 1 0 0 0
[199] 0 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 1 1 1 0 0 0 1 0 1 0 1 0 0 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0
[265] 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0
[331] 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 1 0 1 0 0 0 0 0 1
[397] 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 1
[463] 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0
[529] 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1
[595] 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 0 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1 1 1
[661] 1 0 0 1 1 1
Levels: 0 1
> Table<- table(prd, train$Dementia)
> Table

prd   0   1
  0 264   5
  1   3 394
> sum(diag(Table)/sum(Table))
[1] 0.987988
> mean(prd== train$Dementia)
[1] 0.987988
> prd <- predict(Alz_LDA, train)
> #Stacked Histogram of LDA Functions
> ldahist(data=prd$x[,1], g = train$Dementia)
> "Problem 2"
[1] "Problem 2"
> 
```



We achieve an accuracy of roughly 98.7%~ 99% by utilizing a Training and Testing method. This is as good as accuracy of previous test by correspondence.

c) Analyzing and finding out, would certain misclassification errors be worse than others? If so, how do we measure it?

The misclassification in this case can be if the model incorrectly judges if a parent has dementia or not or are likely to have it. According to the confusion matrix data, the number of True Positive values is 394, while the number of True Negative values is 264. The number of False Positives and False Negatives is 3 and 5, respectively. The negative anticipated value, which appears in the confusion matrix output, is a good indicator of this. The negative projected value is used to calculate the True negative out of all the negatives. As a result, the objective should be to maximize its worth.