Exploring the Relationship Between Different Attributes of a Dataset in the Diagnosis Of Alzheimer’s Disease

Sandhya Lekshmana Kammath

Registration Number: 2200835

21 June 2023

Abstract

This study explores the relationship between various characteristics and the diagnosis of Alzheimer's disease, specifically differentiating between individuals with Alzheimer's (Demented) and those without (Nondemented). The dataset includes features such as age, education level, brain volume, cognitive test scores, and clinical dementia rating. Statistical analysis techniques, including clustering, logistic regression, and feature selection, were used to investigate this relationship. The study aims to identify potential risk factors and predictive indicators for Alzheimer's disease.

Contents

1 Introduction **3**

2 Preliminary Analysis (Exploratory Data Analysis) **3**

3 Analysis **4**

4 Discussion **5**

5 Conclusion **7**

Word Count 1969

1 Introduction

Alzheimer's disease is a widespread neurodegenerative condition that impacts a large population globally. It is characterized by cognitive decline, memory impairment, and behavioral changes. Early detection and accurate diagnosis are crucial for effective intervention and treatment. Exploring the association between various characteristics, including demographics, neuroimaging measures, cognitive tests, and clinical assessments, with the diagnosis of Alzheimer's disease can offer valuable insights into its underlying mechanisms, potential risk factors, and predictive indicators. The findings of this study hold promise for enhancing our comprehension of Alzheimer's disease.

2 Preliminary Analysis (Exploratory Data Analysis)

A preliminary analysis was conducted to gain initial insights into the dataset and understand the characteristics of the variables involved. This involved handling missing values, transforming features, and implementing other necessary adjustments. Table 1 provides an overview of the dataset's structure. A total of 19 rows containing missing values and rows with the group labeled as Converted were also dropped from the dataset. Using one hot encoding, categorical variable M.F is represented as binary vectors. Table 2 shows the minimum, maximum, mean, median, skewness and kurtosis of each attribute. Skewness value of MMSE indicates that its distribution is negatively skewed. On the other hand, CDR indicates that its distribution is positively skewed. Furthermore, the kurtosis values for MMSE and CDR suggest that these variables may have more extreme values or outliers compared to a normal distribution. Figure 1 provide valuable insights into various aspects of the data and contribute to a comprehensive understanding of the relationships between different variables and the diagnosis of Alzheimer's disease.

Table 1: Dataset Structure

Group	M.F	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
chr	chr	int	int	int	int	num	int	num	num

Table 2: Min, max, mean, median, skewness and kurtosis values of each attribute

	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
Min	6	1	4	0	1106	0.644	0.876
Max	23	5	30	2	2004	0.837	1.587
Mean	14.62	2.546	27.26	0.2729	1494	0.7306	1.192
Median	15	2	29	0	1476	0.732	1.189
Skewness	-0.0913	0.154	-2.305	1.486	0.509	0.185	0.0728
Kurtosis	3.059	1.936	9.808	5.621	2.838	2.514	2.757

In Figure 1A, the analysis of dementia rates based on gender groups (Female = 0, Male = 1) highlights a noticeable disparity between men and women. This analysis reveals a higher likelihood of dementia in men compared to women.

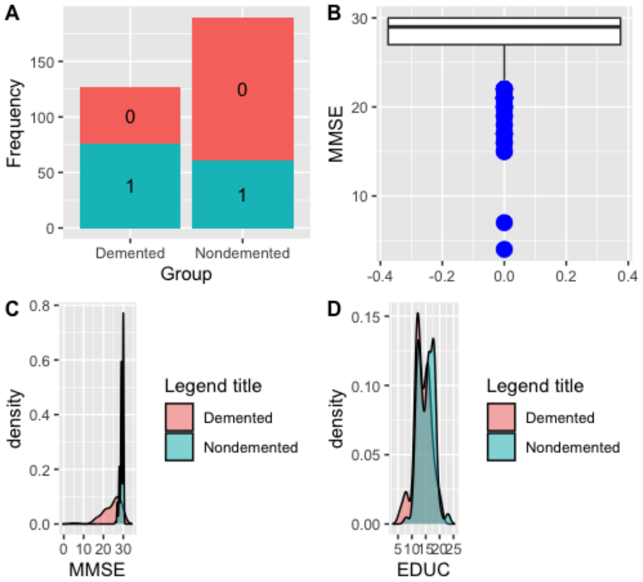

Figure 1: Figure 1A Analysis of the of gender groups (Female = 0, Male = 1). Figure 1B: Boxplot of MMSE. Figure 1C: Analysis of MMSE. Figure 1D: Analysis of Education Year

Figure 1B presents a box plot of MMSE scores, providing insights into the range, spread, and presence of outliers. Figure 1C focuses on the analysis of MMSE scores for demented and non-demented groups of patients. The findings indicate that the non-demented group tends to have higher scores on MMSE compared to the demented group. Figure 1D explores the association between education levels and the risk of developing Alzheimer's disease (AD) or the onset of symptoms. The figure suggests that higher levels of education are associated with a reduced risk of AD or a delayed onset of symptoms.

3 Analysis

Correlation:

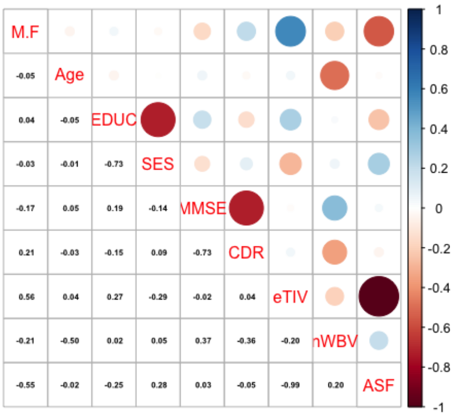

Figure 2: Correlation

Figure 2 presents the correlation matrix depicting the relationships between variables in the dataset. It reveals that EDUC and SES, CDR and MMSE, and ASF and eTIV exhibit strong negative correlations. Additionally, a negative correlation is observed between M.F and ASF. On the other hand, there is a positive correlation between M.F and eTIV.

Feature Selection: The process of feature selection is essential for enhancing model performance, reducing complexity, improving interpretability and reducing overfitting. The wrapper method is a feature selection technique that involves selecting features by assessing their impact on the performance of a specific model.

Principle Component Analysis (PCA): PCA is a powerful technique for dimensionality reduction and data exploration. It helps in understanding the underlying structure of datasets.

Methods - Clustering and Logistic Regression: Clustering is an unsupervised learning technique used to group similar instances together based on the similarity of their features. Kmeans Clustering and Hierarchical Clustering techniques were implemented to analyze the dataset. Logistic regression is a supervised learning algorithm used for predicting categorical outcomes. It models the relationship between independent variables and a binary dependent variable using the logistic function.

Cross Validation: Cross-validation is a resampling technique commonly used in model evaluation to assess the performance and generalization capability of a predictive model. It helps in detecting overfitting or underfitting issues. k-Fold Cross-Validation is a popular technique for model evaluation and selection.

4 Discussion

As the variables have different measure ranges, the numeric variables in the dataset undergo a standardization or rescaling process. This transformation ensures that the variables have a mean of 0 and a standard deviation of 1. The dataset is divided into a train data and a test data using an 80:20 ratio.

Clustering Algorithms: For the k-means and hierachical clustering analysis , all variables in the dataset except for the Group variable, as it represents the pre-defined class labels, are selected as feature. WCSS (Within-Cluster Sum of Squares), is a metric used to evaluate the quality of clustering results. K-means clustering was performed on the train set using different numbers of cluster centers (2, 3, and 4) and 50 random starts. By analyzing the results from NbClust, plot 3 is created to visualize the WCSS values against the number of clusters. The optimal number of clusters in a train set is determined using elbow point. It has been determined that the appropriate number of clusters is 3. A new column was added to the dataset, which contains the values of clusters obtained from the k-means clustering algorithm. A logistic regression model was used to predict the ”cluster” variable based on the remaining variables of train data. The estimated intercept of the logistic regression model is 2.07051, with a standard error of 0.02420. The variables M.F, EDUC, MMSE, CDR, eTIV and ASF were found to be statistically significant predictors with p-values less than 0.05. A LDA (Linear Discriminant Analysis) model was evaluated using 5-fold cross-validation on the train data and then applied to the test data for predicting the cluster variable, achieving a high accuracy of 96%. In hierarchical clustering, the distance matrix is computed using the Euclidean distance method and applied different linkage methods (e.g. single, complete, average, etc.). Additionally, examined the clustering results for different numbers of clusters (k) ranging from 2 to 6. After analyzing the WCSS values obtained from the clustering results, it has been determined that the optimal number of clusters for the train set is 2 and linkage method is complete. Cluster 1 comprises 143 data points, while cluster 2 encompasses 174 data points. Figure **??** shows the optimal number of clusters. Similar to k-means, a logistic regression model was used to predict the ”cluster” variable based on the remaining variables on

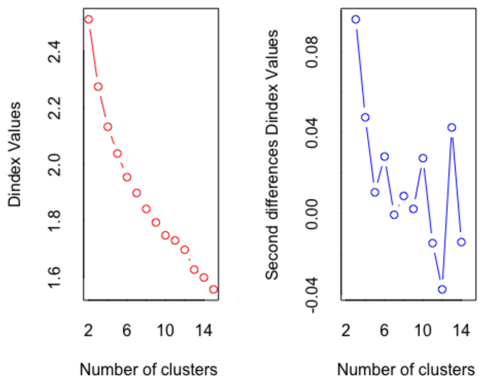

Figure 3: K-means Clusters

the train set. The estimated intercept of the logistic regression model is 1.54703, with a standard error of 0.01621. The variables M.F, EDUC, eTIV, and nWBV were found to be statistically significant predictors with p-values less than 0.05. The LDA model was evaluated using 5-fold cross-validation on the train data and then applied to the test data for predicting the cluster variable, achieving a high accuracy of 94%.

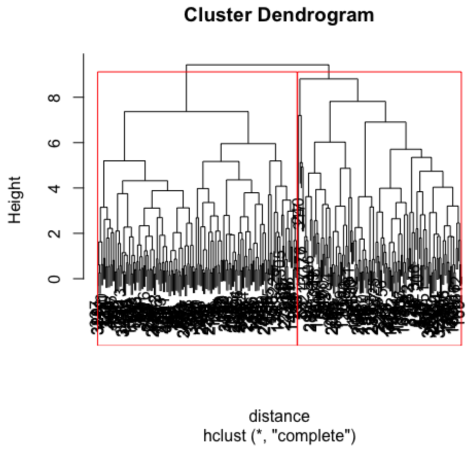

Figure 4: Hierarchical Clusters

Logistic Regression: A logistic regression model was applied to the remaining variables of the dataset to predict the ”Group” variable. The deviance residuals, ranging from -1.041e-04 to 1.211e-04, indicate a good fit of the model as they are close to zero. The coefficient for the ”M.F” variable is -21.281 with a standard error of 31751.668. The z-value of -0.001 suggests that the coefficient is not statistically significant (p-value less than 0.05). This pattern holds for the other variables in the logistic regression model. The residual deviance of 4.0826e-08 on 244 degrees of freedom indicates a good fit of the model to the data. The VIF (Variance Inflation Factor) values for the variables in the both models were high indicating high collinearity between each variable and the other variables in the model. In Figure 5, Normal Q-Q plot of the residuals reveals deviations from the expected line in both tails. The accuracy of the model in correctly predicting the Group variable was found to be 98%. The LDA model was evaluated using 5-fold cross-validation on the train data and then applied to the test data for predicting the Group variable, achieving a accuracy of 98%.

The wrapper method both forward and backward are performed with the goal to find the optimal set of features that maximizes the model’s performance. A models is fitted with variables EDUC, MMSE, CDR, ASF as features selected on forward method with AIC 20 and another with Age, SES, CDR, eTIV, nWBV as features selected on backward method with AIC 10 to predict variable the Group. The results were similar to the complete model with high p values, indicates not good fit of the model to the data. The VIF values for the variables in the both models were high.

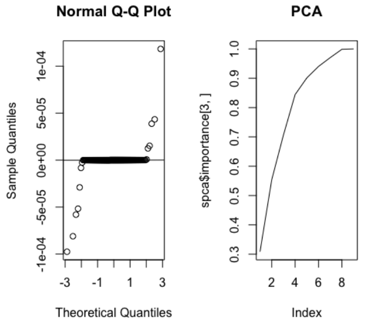

Figure 5: Logistic Regression Model Residual Plot And PCA proportion of variance Plot

PCA is performed for handling high collinearity among variables and extracting meaningful information from a dataset. From the cumulative proportion values obtained from the results of PCA analysis, the first four principal components (PC1 to PC4) explains approximately 84.43% of the total variance. In Figure 5, PCA plot displays the proportions of variance explained by each principal component (PC) in the PCA analysis. Logistic regression model is fitted with PC1 to PC4 to predict variable Group on the train data. The intercept of the logistic regression model is estimated to be 1.59733 and it has a standard error of 0.01854. The t-value is 86.167, indicating a highly significant relationship with the response variable. All coefficients, except for PC3, are significant with p-values less than 0.05. The null deviance is 61.039, and the residual deviance is 21.721, indicating a good fit of the model to the data. The AIC value is 108.22, suggesting reasonable model performance. The accuracy of the model in correctly predicting the

Group variable was found to be 60% and the LDA model was evaluated using 5-fold cross-validation achievied a high accuracy of 90

5 Conclusion

Several analyses and models were performed on the dataset. The analysis of the dataset revealed significant relationships between different attributes. Correlation analysis indicated strong positive as well as negative correlations between the attributes.

Next, k-means clustering was applied with different numbers of clusters and random starts. The within-cluster sum of squares (WCSS) metric was used to evaluate the clustering results. The NbClust and elbow point were utilized to determine the optimal number of clusters, which was found to be 3 based on the analysis.

Further, hierarchical clustering was conducted using different linkage methods and varying the number of clusters from 2 to 6. The WCSS values were examined to assess the effectiveness of the clustering algorithms. Based on the results, the most suitable number of clusters was determined to be 2.

Overall, the combination of PCA, logistic regression, and LDA allowed for effective classification of the Group variable, with the LDA model exhibiting the highest accuracy when compared to logistic regression model.

Reference

References

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3193875/

[2] https://www.sciencedirect.com/science/article/pii/S0169260720314152

[3] https://www.frontiersin.org/articles/10.3389/fpubh.2022.853294/full

Appendix

```
install.package(”caTools”)
install.packages(”car”)

library(car)
library(caTools)
library(tidyverse)
library(moments)
library(corrplot)
library(plotly)
library(factoextra)
library(cowplot)
library(gridExtra)
library (NbClust)
library(caret)


# read the file
projData <- read.csv("/Users/praseedpai/ACADEMIC/Data␣Modelling/Assignment/Final␣project-2
# gets column name of the data frame
names(projData)
# displays number of rows and columns
dim(projData)
# split categorical and numerical columns
# describes the structure of the data
str(projData)
# displays n rows of the data frame
head(projData)
# count NULL values
sum(is.na(projData))
#select rows with NA values in any column
naRows <- projData[!complete.cases(projData), ]
#view results
count(naRows)
count <- projData$Group == 'Converted'
length(which(projData$Group == 'Converted'))
#view results
count(naRows)

nrow( projData[projData$Group == 'Converted',])
####################################################################
## data cleaning
## 373-(37+19)
## 373-19


nrow(projData)
# remove null values
cleanedData <- projData%>%drop_na()
sum(is.na(cleanedData))  # result is 0
```

```
nrow(cleanedData)
xtabs(~cleanedData$Group+cleanedData$M.F)
# remove Group = "Converted"
cleanedData <- subset(cleanedData, Group != "Converted")
# converted M.F values to 1 and 0
cleanedData["M.F"] <- ifelse(cleanedData$M.F == "M", 1, 0)
unique(cleanedData$M.F)
nrow(cleanedData)  # 317

################################################################
## Data Analysis
#

dataDF <- cleanedData
head(dataDF)
# summaries values in a vector
summary(dataDF)

# check for skewness and kurtosis which tells the shape of the distribution
skewness(dataDF[2:10])
kurtosis(dataDF[2:10])

# Correlation rounded to two decimals
roundedData <- round(cor(dataDF[2:10]),
                     digits = 2
)
# plot correlation
corrplot.mixed(roundedData, lower.col = "black", number.cex=.5)

par(mfrow=c(2,2))

# plot Group against frequency of gender
detailsDF<-as.data.frame(table(dataDF$Group, dataDF$M.F))
p1 <- ggplot(detailsDF, aes(x=Var1,y=Freq, fill=Var2,))+
  geom_bar(stat = 'identity')+
  geom_text(aes(label=Var2),position = position_stack(0.5))+
  theme(legend.position = 'none')+labs(x="Group",y="Frequency")
ggplotly(p1)
```

```
# plot MMSE to reveal the outliers
p2 <- ggplot(dataDF, aes( y=MMSE)) +
  geom_boxplot(outlier.colour=”blue”,
               outlier.size=4)
ggplotly(p2)

# plot the analysis of MMSE, ASF, eTIV, nWBV for demented and
# non-demented group of patients
denx <- density(dataDF$MMSE[dataDF$Group == ”Demented”])
deny <- density(dataDF$MMSE[dataDF$Group == ”Nondemented”])

p3 <- ggplot(dataDF, aes(x = MMSE)) +
  geom_density(aes(group = Group, fill = Group), alpha = 0.5) +
  xlim(c(min(denx$x, deny$x), c(max(denx$x, deny$x)))) +
  scale_fill_discrete(name = ”Legend␣title”,
                      labels = c(”Demented”, ”Nondemented”))
ggplotly(p3)
denx <- density(dataDF$EDUC[dataDF$Group == ”Demented”])
deny <- density(dataDF$EDUC[dataDF$Group == ”Nondemented”])
p4 <- ggplot(dataDF, aes(x = EDUC)) +
  geom_density(aes(group = Group, fill = Group), alpha = 0.5) +
  xlim(c(min(denx$x, deny$x), c(max(denx$x, deny$x)))) +
  scale_fill_discrete(name = ”Legend␣title”,
                      labels = c(”Demented”, ”Nondemented”))
ggplotly(p4)
plot_grid(p1, p2, p3, p4, labels = c(’A’, ’B’,’C’,’D’), label_size = 12)

#################################################################
## Clustering algorithms
## Kmeans

clusteringData <- cleanedData[2:10]
clusteringData <- scale(clusteringData)
set.seed(123)
kmeans2 <- kmeans(clusteringData, centers = 2, nstart = 50)
kmeans3 <- kmeans(clusteringData, centers = 3, nstart = 50)
kmeans4 <- kmeans(clusteringData, centers = 4, nstart = 50)

str(kmeans2)
str(kmeans3)
str(kmeans4)

fviz_cluster(kmeans2, data = clusteringData)
fviz_cluster(kmeans3, data = clusteringData)
fviz_cluster(kmeans4, data = clusteringData)

# elbow method
fviz_nbclust(clusteringData, kmeans, method = "wss")+
  geom_vline(xintercept = 3, linetype = 2)

f1 <- fviz_cluster(kmeans2, geom = "point", data = clusteringData) + ggtitle("k␣=␣2")
f2 <- fviz_cluster(kmeans3, geom = "point", data = clusteringData) + ggtitle("k␣=␣3")
f3 <- fviz_cluster(kmeans4, geom = "point", data = clusteringData) + ggtitle("k␣=␣4")
grid.arrange(f1, f2, f3, nrow = 2)

# Nbclust
clusternum <- NbClust(clusteringData, distance="euclidean", method="kmeans")

finalData <- as.data.frame(cbind(clusteringData, cluster = kmeans3$cluster))
splitData <- sample.split(Y = finalData$cluster, SplitRatio = 0.8)
trainData = finalData[splitData,]
testData = finalData[!splitData,]
dim(trainData)
dim(testData)
glm.model <- glm(cluster ~ ., data=trainData)
summary(glm.model)

############### For 5-fold CV
trainData$cluster <- as.factor(trainData$cluster)
trControl <- trainControl(method = "cv", number = 5)
lda.fit <- train(cluster ~ .,
                 method = "lda",
                 trControl = trControl,
                 metric = "Accuracy",
                 data = trainData)
lda.pred <- predict(lda.fit,testData)
table(lda.pred, testData$cluster)
lda.fit
```

```
testData$cluster <- as.factor(testData$cluster)
confusion <- confusionMatrix(lda.pred, testData$cluster)

# Extract accuracy, recall, and precision from the confusion matrix
accuracy <- confusion$overall["Accuracy"]

########################
## Hierarchical Clustering

# calculate the distance matrix
distance <- dist(clusteringData, method = "euclidean")
# list to store the clustering results
cresults <- list()

# the linkage methods to be evaluated
linkageMethods <- c('single', 'complete', 'average', 'centroid')

# Define the range of k values to be evaluated
kvalues <- 2:6

# Perform hierarchical clustering for each linkage method and k value
for (method in linkageMethods) {
  for (k in kvalues) {
    # Perform hierarchical clustering
    fit <- hclust(distance, method = method)

    # cuts the tree into k clusters
    groups <- cutree(fit, k = k)

    #  the clustering results stored in the list
    cresults[[paste(method, k, sep = "_")]] <- groups
  }
}

# the clustering results evaluated using a WCSS
wcss <- numeric()

for (result in cresults) {
  wcssValue <- 0
  for (k in kvalues) {
    wcssValue <- wcssValue + sum((clusteringData -
                                     aggregate(clusteringData,
                                               by = list(result), mean)[, -(ncol(clusteringDa
  }
  wcss <- c(wcss, wcssValue)
}

# print the WCSS values
print(wcss)

# the linkage method and k value that minimize the WCSS
minIndex <- which.min(wcss)
wcssNames <- paste(names(cresults), ”WCSS␣=”, wcss)
bestResult <- wcssNames[minIndex]

# Extracting the linkage method and k value
bestMethod <- strsplit(bestResult, ”_”)[[1]][1]
bestK <- as.integer(str_extract(strsplit(bestResult, ”_”)[[1]][2],”\\d+”))

# plot the clustering result
plot(hclust(distance, method = bestMethod))
fit.best <- hclust(distance, method=bestMethod)
# draw dendrogram with red borders around the 2 clusters
rect.hclust(fit.best, k = bestK, border = ’red’)
groups.fit.best <- cutree(fit.best, k=bestK) # cut tree into k=2 clusters
table(groups.fit.best)



finalData <- as.data.frame(cbind(clusteringData, cluster = groups.fit.best))
splitData <- sample.split(Y = finalData$cluster, SplitRatio = 0.8)
trainData = finalData[splitData,]
testData = finalData[!splitData,]
dim(trainData)
dim(testData)
glm.model <- glm(cluster ~ ., data=trainData)
summary(glm.model)
par(mfrow=c(2,2))
```

```
plot(glm.model)

############### For 5-fold CV
trainData$cluster <- as.factor(trainData$cluster)
trControl <- trainControl(method = ”cv”, number = 5)
lda.fit <- train(cluster ~ .,
                 method = ”lda”,
                 trControl = trControl,
                 metric = ”Accuracy”,
                 data = trainData)
lda.pred <- predict(lda.fit,testData)
table(lda.pred, testData$cluster)
lda.fit

testData$cluster <- as.factor(testData$cluster)
confusion <- confusionMatrix(lda.pred, testData$cluster)

# Extract accuracy, recall, and precision from the confusion matrix
accuracy <- confusion$overall[”Accuracy”]
recall <- confusion$byClass[”Sensitivity”]
precision <- confusion$byClass[”Pos␣Pred␣Value”]

######################################
## Logistic Regression
##
logisticData <- as.data.frame(scale(cleanedData[2:10]))
logisticData$Group <- cleanedData$Group
#logisticData[”Group”] <- ifelse(logisticData$Group == ”Demented”, 1, 0)
logisticData[”Group”] <- as.factor(logisticData$Group)

set.seed(123)

splitData <- sample.split(Y = logisticData$Group, SplitRatio = 0.8)
trainData = logisticData[splitData,]
testData = logisticData[!splitData,]
dim(trainData)
dim(testData)

glm.fits<-glm(Group~.,data=trainData,family=binomial)
```

```
summary(glm.fits)
#get list of residuals
residual <- resid(glm.fits)
#create Q-Q plot for residuals
qqnorm(residual)
#add a straight diagonal line to the plot
qqline(residual)

# predicting the accuracy
glm.probs <- predict(glm.fits,testData,type=”response”) #Pr(Y=1|X)
glm.predicted <- rep(”Demented”,63)
glm.predicted[glm.probs>0.5]=”Nondemented”
mean(glm.predicted==testData$Group)

step1<-step(glm.fits,
            method='backward')

bestModelB <-glm(Group ~ Age + SES + CDR + eTIV + nWBV, data=trainData, family=binomial)
summary(bestModelB)
vif(bestModelB)

glm.fitsForward<-glm(Group~1,data=trainData,family=binomial)
step1<-step(glm.fitsForward,scope=~M.F+ Age + EDUC + SES + MMSE + CDR + eTIV + nWBV + ASF
            method='forward')
bestModelF <-glm(Group ~ CDR + ASF + EDUC + MMSE, data=trainData, family=binomial)
summary(bestModelF)
# a VIF value greater than 5 or 10 is considered indicative of multicollinearity.
vif(bestModelF)

######################################
## Cross validation For 5-fold CV
trainData$Group <- as.factor(trainData$Group)
trControl <- trainControl(method = ”cv”, number = 5)
lda.fit <- train(Group~ .,
                 method = ”lda”,
                 trControl = trControl,
                 metric = ”Accuracy”,
                 data = trainData)
lda.pred <- predict(lda.fit,testData)
```

```
table(lda.pred, testData$Group)
lda.fit

confusion <- confusionMatrix(lda.pred, testData$Group)

# Extract accuracy, recall, and precision from the confusion matrix
accuracy <- confusion$overall[”Accuracy”]
recall <- confusion$byClass[”Sensitivity”]
precision <- confusion$byClass[”Pos␣Pred␣Value”]


####################################
## PCA
##

names <- c(”M.F”,”Age”, ”EDUC”, ”SES”,”MMSE”,”CDR”,”eTIV”,”nWBV”,”ASF”)
data <- logisticData
logisticData <- logisticData[names]
logisticData$Group <- data$Group
x <- logisticData[1:9] # excluded Group
pca <- prcomp(x, center = TRUE, scale. = TRUE)
names(pca)
summary(pca)
print(pca,digit=2) # the loadings
pca.loadings<- pca$rotation
pca.scores <- pca$x
per.var <- pca$sdev^2
prop.var.expl <- per.var/sum(per.var); prop.var.expl
cumsum(prop.var.expl)
cumulative.prop.var <- cumsum(prop.var.expl)

spca = summary(pca)
plot(spca$importance[3,], type=”l”,main=”PCA”)

# Kaiser’s Rule: Retain principal components with eigenvalues greater than 1.
# This rule suggests that components with eigenvalues less than 1 may not
# contain enough information to be useful. and elbow method
components <- cbind(Group = logisticData[, ”Group”], pca$x[, 1:4]) %>%
  as.data.frame()
```

```
#fitModel <- glm(Group ~ ., data = components)
#summary(fitModel)
set.seed(123)
splitData <- sample.split(Y = components$Group, SplitRatio = 0.8)
trainData = components[splitData,]
testData = components[!splitData,]
dim(trainData)
dim(testData)
# fit a logistic model
fitModel <- glm(Group ~ ., data = trainData)
summary(fitModel)

# predicting the accuracy of model
glm.probs <- predict(fitModel,testData,type=”response”) #Pr(Y=1|X)
glm.predicted <- rep(1,63)
glm.predicted[glm.probs>0.5]=2
mean(glm.predicted==testData$Group)


######################################
## Cross validation For 5-fold CV
trainData$Group <- as.factor(trainData$Group)
trControl <- trainControl(method = ”cv”, number = 5)
lda.fit <- train(Group~ .,
                 method = ”lda”,
                 trControl = trControl,
                 metric = ”Accuracy”,
                 data = trainData)
lda.pred <- predict(lda.fit,testData)
table(lda.pred, testData$Group)
lda.fit
```