Analysis of Alzheimer's disease

using machine learning models and feature selection

Author

Venkata Sushma Pagidala

(RegID:2212197)

Date

21-06-2023

Abstract

Alzheimer's disease is a progressive brain disorder that causes memory loss and cognitive decline. Around 48 million people around the world are suffering from the loss of cognitive functioning of brain. Prediction of this disease at an early stage is challenging. Alzheimer's disease accounts for around 60-80% of the dementia cases. It is caused by the build-up of proteins in the brain which damage the brains cell's ability to transmit messages. It is currently ranked as the seventh leading cause of the death in the United States. There is no particular treatment for this disease, but treating this disease at an early stage could aid in reducing damage caused by it to human brains. There are several factors that influence dementia analysis, such as demographic factors, clinical factors and derived anatomic volumes. Analysis of this data and predicting the disease can be done by using machine learning algorithms, both supervised and unsupervised. In this report, using Kmeans unsupervised clustering algorithms data is divided into three clusters. The data set is divided into two parts, training and testing dataset. The logistic regression model is trained using train dataset and predictions are made using test dataset. In order to minimise cost, enhance time, space complexity and performance of the model, feature selection is done on the data. Through these models, using given factors, doctors would be able to estimate whether the person is having disease or not, which in turn helps in early diagnosis of the patient and increases the chance of effective treatment and extended life.

Contents

Word Count:1784

Introduction:

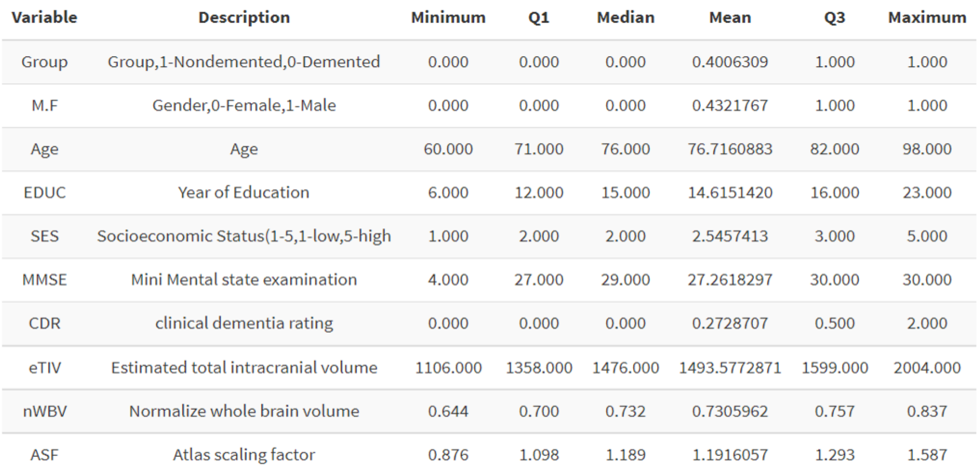
Dementia is the loss of the cognitive functioning-thinking, remembering, and reasoning to such an extinct that it effects the daily human life activities. It is most commonly observed in old age people. People with dementia (damage caused to brain cells) cannot control their emotions which results in the change in their personality. The severity of the disease varies with the age, as the person with the dementia age the severity also increases. There are different forms of dementia such as, Alzheimer's dementia, Vascular dementia are two important dementia.

There are several attributes which are taken into consideration while estimating the AD such as Group (demented, nondemented), Demographical information (Gender, Age, years of education, Socioeconomic status(1-highest status,5-lowest status)), Clinical information (Mini mental state Examination score(0- worst,30-best), Clinical Dementia Rating(0-no dementia, 0.5- very mild AD, 1- mild AD, 2- moderate AD)), Derived anatomic volumes (Estimated total intracranial volume (mm3), Normalized whole-brain volume , Atlas scaling factor). Using these features one can build a model to predict the Alzheimer's disease at an early stage and aid in treatment.

Preliminary Analysis:

Data Cleaning: The dataset consists of 10 features with 373 observations. On removing the converted group and dropping the null values there are only 317 observations. Group and gender are converted into a categorical variable.

From the below data summary table, it can be interpreted that the dataset consists of individuals diagnosed as demented or nondemented. The gender distribution is slightly biased towards females. It can be said that the average age of a person with AD is 76 years, with a range from 60-98 years. The average years of educational attainment is around 14 years, indicating moderate level of education. Moreover, most of the individuals are having the moderate socioeconomic status. The CDR reflects the severity of dementia, with the average value of 0.27, stating the mild dementia. The cognitive function scores, suggest that there is a mild cognitive impairment on average. Other features, such as eTIV, nWBV, ASF exhibit varying ranges and distributions. This dataset summary provides a comprehensive overview of the variables, allowing for deeper understanding of characteristics and potential relationships associated with the diagnosis of dementia in AD.



Variable	Description	Minimum	Q1	Median	Mean	Q3	Maximum
Group	Group,1-Nondemented,0-Demented	0.000	0.000	0.000	0.4006309	1.000	1.000
M.F	Gender,0-Female,1-Male	0.000	0.000	0.000	0.4321767	1.000	1.000
Age	Age	60.000	71.000	76.000	76.7160883	82.000	98.000
EDUC	Year of Education	6.000	12.000	15.000	14.6151420	16.000	23.000
SES	Socioeconomic Status(1-5,1-low,5-high	1.000	2.000	2.000	2.5457413	3.000	5.000
MMSE	Mini Mental state examination	4.000	27.000	29.000	27.2618297	30.000	30.000
CDR	clinical dementia rating	0.000	0.000	0.000	0.2728707	0.500	2.000
eTIV	Estimated total intracranial volume	1106.000	1358.000	1476.000	1493.5772871	1599.000	2004.000
nWBV	Normalize whole brain volume	0.644	0.700	0.732	0.7305962	0.757	0.837
ASF	Atlas scaling factor	0.876	1.098	1.189	1.1916057	1.293	1.587

Table 1: Summary of given project dataset

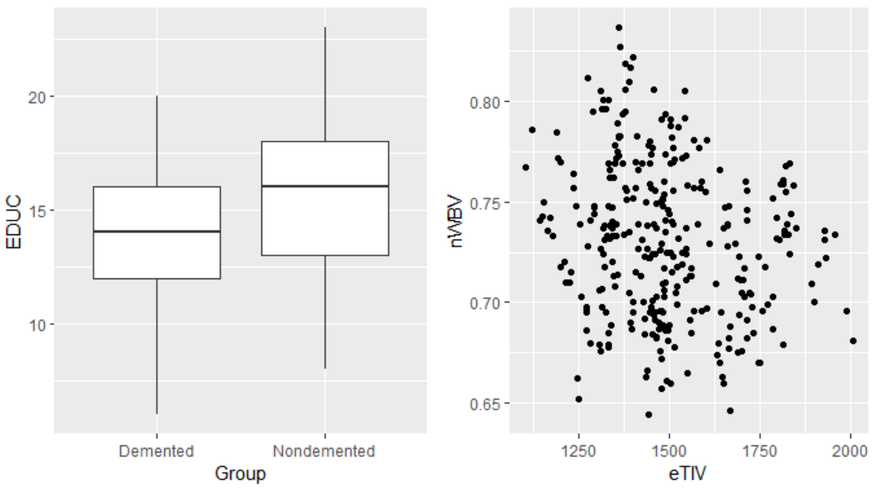


Figure 1: Boxplot of EDUC vs Group, scatter plot of nWBV vs eTIV

From the above boxplot, it can be interpreted that there is a significant difference in educational level between the two groups. The nondemented individuals has the higher educational level than the demented group. The IQR(Inter Quartile Range) for the demented group is wider than the IQR for the demented group, showing the variation of EDUC among individuals. It can be said that people with higher EDUC might engage in activities that keeps brain active, such as reading books, learning new things and staying mentally active.

It can be interpreted from the above scatter plot that the age and the volume of the brain are negatively correlated. This states that, the decrease in brain volume with age is likely due to the fact that the brain undergoes various changes, such as loss of neurons and build of plaques and tau tangles, which can lead to decrease in the size of the brain. People with low nWBV at a younger age may be at a higher risk of developing AD later in life.

Clustering Algorithm:

Unsupervised Learning is a type of machine learning algorithm in which models are trained using the unlabelled data. It identifies the pattern or structures within the data without prior knowledge or guidance. **Clustering** involves in grouping similar data points together based on their characteristics or similarities patterns as a cluster. There are different types of clustering, such as Hierarchical, K-Means, Agglomerative clustering and Dendrogram are some commonly used.

K-means Clustering: It partitions a dataset into a specified number of k clusters by iteratively assigning data points to the nearest centroid and updating centroids based on the mean of the data points in each cluster until convergence. It aims to minimise the within-cluster sum of squared distances, making cluster more compact and data points within each cluster more similar. The ‘means’ in k-means refers to the averaging of the data (centroid).

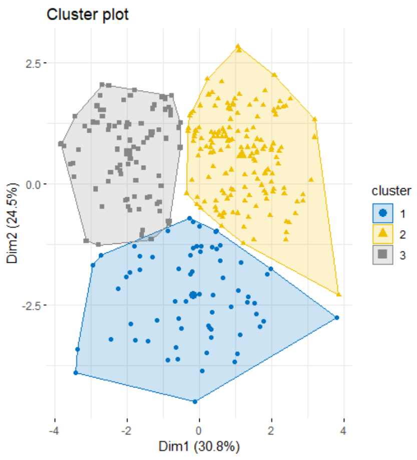
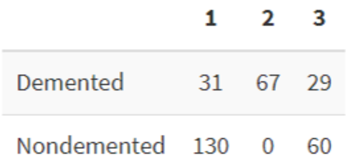


Figure 2: K-means cluster plot of given data



	1	2	3
Demented	31	67	29
Nondemented	130	0	60

Table2: K-means, cluster table

The above plot shows that the data is divided into three clusters using K-means(k=3) algorithm. By the elbow method, it is evident that the optimum number of clusters is 3. Scaled data (data_value-mean/standard_deviation) is used for constructing clustering model. It can be interpreted that in second cluster there are only demented group observations, where as in second and third clusters are the heterogeneous mixture of both groups. The first cluster has around 80% of nondemented group data, while in the third cluster the demented group data is around half of the nondemented data.

Logistic Regression:

Supervised learning is a machine learning approach where algorithm will be trained from labelled data to make predictions or decisions. There are two types of supervised learning algorithms, such as Classification and regression. **Classification** algorithms are used when target variable is categorical or discrete. Following are few types of classification algorithms, Logistic regression, Support Vector Machines, Decision tree. **Regression** algorithms are used when target variable is continuous or numerical. **Logistic Regression**: It estimates of the probability of the event occurring based on given dataset of independent variables by fitting data to a logistic function. The outcome of the model is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds (the probability of success divided by the probability of failure), which is also called as log odds, or the natural logarithm of odds.

For the given Alzheimer's dataset, dataset is divided into training and testing dataset in a 70:30 ratio. Logistic regression model is trained using train dataset and made predictions based on the input values of the test dataset. The output of the model can be interpreted as a probability of the input data varies between 0 and 1. If the probability is greater than 0.5 it is considered as nondemented else demented. Overall, the model gave around 97% of the accuracy. Converted the data values with CDR= 0.5 to CDR=1. If we remove the CDR feature while constructing logistic regression model then the model is having accuracy around 86 %. Thus, it indicates that Clinical Dementia Rating (CDR) plays a significant role in model prediction.

Feature selection:

It is a process of selecting a subset of relevant features or variables from a larger set of available features in the dataset by removing redundant, irrelevant or noisy features. It aims to reduce the dimensionality, enhance model interpretability and prevent overfitting. There are different techniques in selecting features, such as wrapper method, Filter method and Embedded method. Wrapper methods evaluate the performance of a machine learning model with different subsets of features. In this report we used mostly wrapper methods, such as step forward, step backward and recursive feature elimination methods. Forward selection, this method starts with an empty set of features and iteratively adds one feature at a time, evaluating the model's performance at each step, selects gender,CDR,eTIV,EDUC features with AIC values of -901.43 and R2 value of 76.51 . Backward selection, this method begins with the full set of features and iteratively removes one feature at a time, evaluating the model performance after each removal, selects gender, Age, EDUC, CDR, nWBV features with AIC value -901.19 and R2 value 76.89. RFE eliminates less important features, starts with all features and them based on their importance. 10-fold RFE selects CDR, nWBV, ASF, gender features. The Boruta methods states that all the features are confirmed, yet through graphical interpretation it can be said that the CDR is highly influential. Using these three feature selections, three different logistic regression models are built and these are cross validated. It shows that step backward features perform the best among the three models.

Discussion:

The analysis of the Alzheimer’s dataset provides valuable insights into the relationships between various features and diagnosis of disease. The descriptive statistics aids in having a comprehensive overview of the dataset, highlighting key statistical measures using visualizations and summarising the data. With the help of clustering algorithm, the data is classified into three groups which helps in understanding the patterns within the data and disease. The logistic regression helps in understanding the data and the influential features in the dataset, which can highly influence the overall model. Feature selection aids in identifying the important features for diagnosis, which enhances our understanding of the disease underlying factors.

Conclusion:

Predicting Alzheimer’s disease at an early stage is very important to control the damage caused to brain. Through the data exploration it can be concluded mostly dementia is observed in people who lack intellectual background and age between 70-90 years. Moreover, CDR plays a influential role in model building. Using elbow method, the data is clustered into three groups, two clusters having mixed groups. When the complete model is considered, the logistic model gave around 97% of accuracy, while eliminating CDR it gave 86% of accuracy. By feature selection, it can be said that the CDR,ASF,M.F,nWBV,Age,EDUC are significant features for predicting the AD among individuals.

References:

1. C.Kavitha,Vinodhini Mani, “Early-stage Alzheimer’s disease prediction using machine learning models” ,2022(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8927715/)
2. Adithya Kumar Pandey, “A simple explanation of k-means clustering”,2020 (https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/)
3. Daniel Johnson, “Supervised Machine Learning”, 2023 (https://www.guru99.com/supervised-machine-learning.html)
4. Rahul Agarwal, “The 5 feature selection algorithms every data scientist should know”,2019 (https://shorturl.at/moEF8)

Appendix

```
##libraries used
library(tidyverse);library(kableExtra);library(tibble);library(factoextra);library(caret)
library(MASS);library(boot);library(gridExtra)
##loading dataset
setwd("C:/Users/sushm/OneDrive/Documents/ma335/dataset")
project_data<- read.csv("project data.csv") # loading the dataset
logistic_data<-read.csv("project data.csv") # loading the dataset
##Data Exploration
names(project_data)      # printing the column names of the dataset
summary(project_data)    # viewing the summary of the data
head(project_data)       # viewing top 6 records of the data
str(project_data)        #  viewing the structure of the data
##removing na's and converted group
project_data<-na.omit(project_data)  # omiting null values
project_data$M.F<-as.factor(project_data$M.F)  #converting the M.F column to a factor
project_data <- project_data %>%   filter(Group!="Converted")# filtering the data where group is not converted
project_data$Group<-as.factor(project_data$Group)  #converting the Group column to a factor
##converting the character data to the factors
project_data$M.F<-ifelse(project_data$M.F=="M",1,0)  # converting the char to numeric
levels(as.factor(project_data$M.F))     # printing the levels of the gender
project_data_frame<-project_data        # assigning the cleaned data to a new variable
project_data_frame$Group<-ifelse(project_data_frame$Group=="Nondemented",0,1) # converting the char to numeric
str(project_data_frame)    # viewing the structure of the data
##question 1: Descriptive statistics
#generating the table
#considering only the numerical columns
numeric_data_columns <- sapply(project_data_frame, is.numeric)  # assigning the numeric data columns to a new variable
```

```
numeric_data <- project_data_frame[, numeric_data_columns]      # storing the numeric data in a new variable

Description<-c("Group,0-Nondemented,1-Demented ","Gender,0-Female,1-Male","Age","Year of Education"," Socioeconomic Status(1-5,1-low,5-high","Mini Mental state examination","clinical dementia rating","Estimated total intracranial volume","Normalize whole brain volume","Atlas scaling factor")            # vector of description of data columns

# Calculate summary statistics for each variable

summary_project_data <- tibble(

  Variable = names(numeric_data),                        # adding variable names of column

  Description=Description,                               # adding description of each column

  Minimum = sapply(numeric_data, min, na.rm = TRUE),      # computing the minimum value of the each column

  Q1 = sapply(numeric_data, quantile, 0.25, na.rm = TRUE), # computing the first quantile of each column

  Median = sapply(numeric_data, median, na.rm = TRUE),     # computing the median of each column

  Mean = sapply(numeric_data, mean, na.rm = TRUE),        # computing the mean of the each column

  Q3 = sapply(numeric_data, quantile, 0.75, na.rm = TRUE), # computing the third quantile of each column

  Maximum = sapply(numeric_data, max, na.rm = TRUE)       # computing the maximum value of each column

)

# printing the summary table

summary_table <- kable(summary_project_data, format = "html", align = rep("c", 7)) %>%

  kable_styling(bootstrap_options = "striped", full_width = FALSE)

summary_table

##Graphical representation

par(mfrow=c(2,2))

#boxplot of EDUC vs Group

boxplot<-ggplot(project_data,aes(y=EDUC,x=as.factor(Group)))+ geom_boxplot()+

  xlab("Group")+  ylab("EDUC")

##scatter plot of eTIV vs nWBV

scatterplot<-ggplot(project_data, aes(eTIV,nWBV))+geom_point()
```

```
grid.arrange(boxplot,scatterplot,ncol=2)

##Q2 clustering

scaled_data<- scale(project_data[,-1]) # scaling the data

set.seed(123)

fviz_nbclust(project_data[,-1], kmeans, method = "wss")+ geom_vline(xintercept = 3, linetype = 2)

kmeans_cluster_data<-kmeans(scaled_data,centers=3,nstart=25) # constructing the three clusters

kmeans_cluster_table<-table(project_data$Group,kmeans_cluster_data$cluster)  # tabulating the clustered data

kmeans_cluster_table_kable<-kable(kmeans_cluster_table, format = "html", align = rep("c", 7)) %>%

  kable_styling(bootstrap_options = "striped", full_width = FALSE)  # printing the cluster data table

kmeans_cluster_table_kable

# graphical representation of the clustering plot

fviz_cluster(kmeans_cluster_data,data=scaled_data, geom="point",  ellipse.type = "convex",

             ellipse = TRUE, palette="jco", ggtheme = theme_minimal())

## Q3 logistic regression

# cleaning the data

logistic_data<-logistic_data %>%

  filter(Group!='Converted')

logistic_data <- logistic_data %>%  dplyr::mutate(dementia = ifelse (Group == "Nondemented", 1, 0),gender = if_else(M.F == "F", 0, 1))

#converting the gender and group into factors

logistic_data$dementia<-as.factor(logistic_data$dementia)

logistic_data$gender<-as.factor(logistic_data$gender)

logistic_model_with_CDR<-glm(dementia ~.,data=logistic_data[3:12],family=binomial()) # building the logistic regression model on complete data

summary(logistic_model_with_CDR)  # visualizing the summary of the logistic model

logistic_data$CDR<-ifelse(logistic_data$CDR==0.5,1,logistic_data$CDR)  #converting the CDR of 0.5 to 1

##training the data

one_rows_CDR<-logistic_data[which(logistic_data$dementia==1),]     # extracting the rows containing group nondemented
```

```
zero_rows_CDR<-logistic_data[which(logistic_data$dementia==0),]    # extracting the rows containing group demented

set.seed(123)

one_rows_train_CDR<-sample(1:nrow(one_rows_CDR),0.7*nrow(one_rows_CDR))   # splitting the 70% of the nondemented group data as train data

zero_rows_train_CDR<-sample(1:nrow(zero_rows_CDR),0.7*nrow(zero_rows_CDR)) # splitting the 70% of the demented group data as train data

one_rows_train_data_CDR<-one_rows_CDR[one_rows_train_CDR,]              # extracting the data using the split-ted row numbers of nondemnted

zero_rows_train_data_CDR<-zero_rows_CDR[zero_rows_train_CDR,]             # extracting the data using the split-ted row numbers of demented

train_data_CDR<-rbind(one_rows_train_data_CDR,zero_rows_train_data_CDR)    # row binding the data to form a train dataset containing both demented and nondemented

##test data

one_rows_test_data_CDR<-one_rows_CDR[-one_rows_train_CDR,]        # extracting the rows of nondemented which are not used for training

zero_rows_test_data_CDR<-zero_rows_CDR[-zero_rows_train_CDR,]     # extracting the rows of demented which are not used for training

test_data_CDR<-rbind(one_rows_test_data_CDR,zero_rows_test_data_CDR)  # row binding the data for test data

logistic_model_train<-glm(dementia ~.,data=train_data_CDR[3:12],family=binomial(link="logit"))  # training the logistic regression model using training dataset

predicted_test_CDR<-predict(logistic_model_train,test_data_CDR,type='response')  # predicting the test data using the above logistic model

summary(logistic_model_train)                              # summarizing the trainied logistic model

predicted_test_CDR<-as.factor(ifelse(predicted_test_CDR>0.5,1,0))   # factorizing the predicted data as 1 and 0

levels(predicted_test_CDR)            # viewing the levels of predicted data

levels(test_data_CDR$dementia)        # viewing the levels of dementia data

test_data_CDR$dementia<-as.factor(test_data_CDR$dementia)   # converting the dementia data to factor

confusion_matrix_CDR<-confusionMatrix(data=predicted_test_CDR,reference = test_data_CDR$dementia) # building a confusion matrix

confusion_matrix_table_CDR<-confusion_matrix_CDR$table   # extracting the table from the confusion matrix

confusion_matrix_CDR$overall["Accuracy"]              # printing the accuracy of model
```

```
## logistic regression model without CDR

logistic_data_without_CDR<-logistic_data %>%

  dplyr::select(-CDR)

##training the data

one_rows_without_CDR<-logistic_data_without_CDR[ which(logistic_data_without_CDR$dementia==1), ]  #extracting data of nondemented group without CDR

zero_rows_without_CDR<-logistic_data_without_CDR[which(logistic_data_without_CDR$dementia==0),] # extracting data of demented group without CDR

set.seed(123)

one_rows_train_without_CDR<-sample(1:nrow(one_rows_without_CDR),0.7*nrow(one_rows_without_CDR))  # splitting 70% of the nondemented data (without CDR)

zero_rows_train_without_CDR<-sample(1:nrow(zero_rows_without_CDR),0.7*nrow(zero_rows_without_CDR)) # splitting 70% of the demendted data (without CDR)

one_rows_train_data_without_CDR<-one_rows_without_CDR[one_rows_train_without_CDR,]      # extracting 70% of the nondemented data(without CDR)

zero_rows_train_data_without_CDR<-zero_rows_without_CDR[zero_rows_train_without_CDR,]   # extracting 70% of the demented data (without CDR)

train_data_without_CDR<-rbind(one_rows_train_data_without_CDR,zero_rows_train_data_without_CDR)  # rowbinding the training data (without CDR)


##test data without CDR

one_rows_test_data_without_CDR<-one_rows_without_CDR[-one_rows_train_without_CDR,]    # extracting 30% of nondemented data (without CDR)

zero_rows_test_data_without_CDR<-zero_rows_without_CDR[-zero_rows_train_without_CDR,] # extracting 30% of demented data (without CDR)

test_data_without_CDR<-rbind(one_rows_test_data_without_CDR,zero_rows_test_data_without_CDR) # rbinding the above data to build the test data

logistic_model_train_without_CDR<-glm(dementia ~.,data=train_data_without_CDR[3:11],family=binomial(link="logit"))  #  training the logistic model with train dataset (without CDR)
```

```
predicted_test_without_CDR<-predict(logistic_model_train_without_CDR,test_data_without_CDR,type='response') # predicting the data using test data (without CDR)

summary(logistic_model_train_without_CDR) # viewing the summary of logistic model


predicted_test_without_CDR<-as.factor(ifelse(predicted_test_without_CDR>0.5,1,0))  # converting the predicted data into factors of levels 1 and 0

levels(predicted_test_without_CDR)   # visualizing the levels of predicted data

test_data_without_CDR$dementia<-as.factor(test_data_without_CDR$dementia)  # converting the dementia data into factors

levels(test_data_without_CDR$dementia)  # visualizing the levels of dementia data

confusion_matrix_without_CDR<-confusionMatrix(data=predicted_test_without_CDR,reference = test_data_without_CDR$dementia)  # building teh confusion matrix

confusion_matrix_table_without_CDR<-confusion_matrix_without_CDR$table   # constructing the confusion matrix table

confusion_matrix_without_CDR$overall["Accuracy"]  # printing the accuracy of the model (without CDR)

##Q4 feature selection

# step forward

complete_model<-lm(Group ~1,data=project_data_frame)    # building the lm model

step_forward<-step(complete_model,scope=~M.F+Age+EDUC+SES+MMSE+CDR+eTIV+nWBV+ASF,method='forward') # computing the step forward feature selection

summary(step_forward)  # visualizing the summary of the step forward feature selection

#step backward

y1<-project_data_frame[,1]  # extracting the group data

X1<-project_data_frame[,2:10] # extracting the predictors data

model_backward<-lm(y1~.,data=X1)  # constructing the lm model

step_backward<-step(model_backward,method="backward")  # constructing the step backward model

summary(model_backward)       # visualizing the summary of the model

##boruta feature selection

library(Boruta)

boruta_feature_selection <- Boruta(y1 ~.-Group, data=project_data_frame, doTrace=1)  # building the boruta model
```

```
decision<-boruta_feature_selection$finalDecision  # decision of the boruta feature selection

significant_variables<-decision[boruta_feature_selection$finalDecision%in% c("Confirmed")]  #visualizing the significant variables

print(significant_variables)  # printing the significant variables

plot(boruta_feature_selection, xlab="", main="Variable Importance") #plotting the boruta feature selection graph

attStats(boruta_feature_selection)


#caret cross validation

attach(project_data_frame)    # attaching the project data

y<-cbind(Group)              # exctracting the group data

X<-cbind(M.F,Age,EDUC,SES,MMSE,CDR,eTIV,nWBV,ASF)  # extracting the predictors data

set.seed(10)

control <- rfeControl(functions = lmFuncs, method = "repeatedcv", repeats = 10,

                   number = 10)      # building an rfe control on the lm

lmProfile <- rfe(X, y, sizes = c(1:9),  rfeControl = control)   # building the rfe model

lmProfile ; predictors(lmProfile)         #viewing the selected features

lmProfile$fit  # lm model with the extracted features


##cross validation

glm_forward<-glm(Group~ CDR+EDUC+M.F+eTIV, data=project_data_frame,

            family=binomial)    # constructing the logistic regression on step forward feature selection

glm_backward<-glm(Group~ CDR+nWBV+eTIV+EDUC+Age, data=project_data_frame,

             family=binomial)   # constructing the logistic regression on step backward feature selection

glm_rfe<-glm(Group~CDR+M.F+ASF+nWBV, data=project_data_frame,

             family=binomial)  # constructing the logistic regression on rfe feature selection

cv_forward<-cv.glm(project_data_frame,glm_forward,K=10) # constructing 10 fold cross validation on forward features

cv_forward$delta    # finding the deviation of forward step feature selection
```

```
cv_backward<-cv.glm(project_data_frame,glm_backward,K=10) # constructing 10 fold cross validation on backward features

cv_backward$delta  # computing the cross validation deviation

cv_rfe<-cv.glm(project_data_frame,glm_rfe,K=10) # constructing 10 fold cross validation on rfe features

cv_rfe$delta  # computing the cross validation deviation
```