Thema09DementiaPrediction

Ewoud

2023-09-06

## Introduction

Dementia is a pressing global health concern, with a significant impact on individuals, families, and healthcare systems. Timely diagnosis and intervention are crucial for improving the quality of life for those affected by dementia. Advances in machine learning and healthcare technology offer promising opportunities to enhance the accuracy and efficiency of dementia diagnosis.

The question this research is aiming to give an answer to is: *How accurate can a machine learning model be, that predicts if a subject has dementia using different clinical parameters?*

Our approach combines machine learning and dementia research to uncover hidden patterns in clinical data. We will conduct an Exploratory Data Analysis (EDA) to identify correlations with the dementia group, assisting in feature selection and model development.

Dataset: <https://www.kaggle.com/datasets/shashwatwork/dementia-prediction-dataset>

# Load in the data   
Data1 <- read\_excel("oasis\_longitudinal\_demographics.xlsx")  
colnames(Data1) <- c("Subject ID","MRI ID","Group","Visit","MR Delay","M/F","Hand","Age","EDUC","SES","MMSE","CDR","eTIV","nWBV","ASF")  
  
Data2 <- read\_excel("Predictions.xlsx")

## Codebook

codebook <- enframe(get\_label(Data1))  
  
colnames(codebook) <- c("variable\_id", "item\_text")  
describtion = c("Id of subject","Id of MRI","Converted / Demented/ Nondemented","Number of visit ","Delay with MRI","Gender : Male / Female ","Handedness","Age of the subject at time of visit","Years of education","Socioeconomic status","Mini-Mental State Examination score","Clinical Dementia Rating","Estimated total intracranial volume","Normalized whole-brain volume","Atlas scaling factor")  
codebook$item\_text = describtion  
  
#Codebook  
My\_Codebook <- read.table("Codebook.txt", sep ="|",  
 header = TRUE, dec =".")  
My\_Codebook <- data.frame(My\_Codebook)  
pander::pander(My\_Codebook, style = "simple", split.table = Inf)

| Name | description | type | value |
| --- | --- | --- | --- |
| Subject.ID | Id of the patient | character | OAS2\_0001 - OAS2\_0186 |
| MRI ID | Id of MRI | character | OAS2\_0001\_MRI1 - OAS2\_0186\_MRI3 |
| Group | Converted / Demented/ Nondemented | character | Converted-Demented-Nondemented |
| Visit | Number of visit | character | 1-5 |
| MR Delay | Delay with MRI | double | 0-2639 |
| M/F | Gender : Male / Female | character | M-F |
| Hand | Handedness | character | R-L |
| Age | Age of the subject at time of visit | double | 60- 98 |
| EDUC | Years of education | double | 6-23 |
| SES | Socioeconomic status | double | 1-5 |
| MMSE | Mini-Mental State Examination score | double | 4-30 |
| CDR | Clinical Dementia Rating | double | 0.0-2.0 |
| eTIV | Estimated total intracranial volume | double | 1105-2005 |
| nWBV | Normalized whole-brain volume | double | 0.64-0.84 |
| ASF | Atlas scaling factor | double | 0.87-1.59 |

### Description of some of the rows

SES : Socioeconomic status as assessed by the Hollingshead Index of Social Position and classified into categories from 1 (highest status) to 5 (lowest status)

MMSE : Mini–Mental State Examination (MMSE) The Mini–Mental State Examination (MMSE) or Folstein test is a 30-point questionnaire that is used extensively in clinical and research settings to measure cognitive impairment. It is commonly used in medicine and allied health to screen for dementia. It is also used to estimate the severity and progression of cognitive impairment and to follow the course of cognitive changes in an individual over time; thus making it an effective way to document an individual’s response to treatment. The MMSE’s purpose has been not, on its own, to provide a diagnosis for any particular nosological entity.

Interpretations

Any score greater than or equal to 24 points (out of 30) indicates a normal cognition. Below this, scores can indicate severe (≤9 points), moderate (10–18 points) or mild (19–23 points) cognitive impairment. The raw score may also need to be corrected for educational attainment and age. That is, a maximal score of 30 points can never rule out dementia. Low to very low scores correlate closely with the presence of dementia, although other mental disorders can also lead to abnormal findings on MMSE testing. The presence of purely physical problems can also interfere with interpretation if not properly noted; for example, a patient may be physically unable to hear or read instructions properly, or may have a motor deficit that affects writing and drawing skills.

CDR : Clinical Dementia Rating (CDR) The CDR™ in one aspect is a 5-point scale used to characterize six domains of cognitive and functional performance applicable to Alzheimer disease and related dementias: Memory, Orientation, Judgment & Problem Solving, Community Affairs, Home & Hobbies, and Personal Care. The necessary information to make each rating is obtained through a semi-structured interview of the patient and a reliable informant or collateral source (e.g., family member) referred to as the CDR™ Assessment Protocol.

The CDR™ Scoring Table provides descriptive anchors that guide the clinician in making appropriate ratings based on interview data and clinical judgment. In addition to ratings for each domain, an overall CDR™ score may be calculated through the use of an CDR™ Scoring Algorithm. This score is useful for characterizing and tracking a patient’s level of impairment/dementia:

0 = Normal 0.5 = Very Mild Dementia 1 = Mild Dementia 2 = Moderate Dementia 3 = Severe Dementia

eTIV: Estimated total intracranial volume (eTIV) The ICV measure, sometimes referred to as total intracranial volume (TIV), refers to the estimated volume of the cranial cavity as outlined by the supratentorial dura matter or cerebral contour when dura is not clearly detectable. ICV is often used in studies involved with analysis of the cerebral structure under different imaging modalities, such as Magnetic Resonance (MR), MR and Diffusion Tensor Imaging (DTI), MR and Single-photon Emission Computed Tomography (SPECT), Ultrasound and Computed Tomography (CT). ICV consistency during aging makes it a reliable tool for correction of head size variation across subjects in studies that rely on morphological features of the brain. ICV, along with age and gender are reported as covariates to adjust for regression analyses in investigating progressive neurodegenerative brain disorders, such as Alzheimer’s disease, aging and cognitive impairment. ICV has also been utilized as an independent voxel based morphometric feature to evaluate age-related changes in the structure of premorbid brai, determine characterizing atrophy patterns in subjects with mild cognitive impairment (MCI) and Alzheimer’s disease (AD), delineate structural abnormalities in the white matter (WM) in schizophrenia, epilepsy, and gauge cognitive efficacy.

nWBV : Normalized whole-brain volume, expressed as a percent of all voxels in the atlas-masked image that are labeled as gray or white matter by the automated tissue segmentation process

ASF: Atlas scaling factor (unitless). Computed scaling factor that transforms native-space brain and skull to the atlas target (i.e., the determinant of the transform matrix)

Source : <https://www.kaggle.com/code/ruslankl/dementia-prediction-w-tree-based-models>

## Cleaning

First thing to do is to delete useless parameters and change the parameters with character type to numeric values so we can work with them.

The MRI id and delay because these are not parameters that have influence on the outcome if a subject has dementia. Also CDR because it is basically a parameter telling if a patient has dementia or not so it will not be taking in for machine learning but testing if algorithm is correct.

And we change the group, visit, hand and gender parameters.

In the article of the data set, the converted group are people that were identified as demented but a second test confirmed that they were non demented so it is a group that swings in the middle. If i want to transform the nominal group of demented converted and non demented to numerical i can change it to 1 : “demented” 2 : “converted” and 3 : “non demented” because there is a relation.

<https://www.sciencedirect.com/science/article/pii/S2352914819300917?via%3Dihub>

“Explaining the present MRI sessions categorization based on the current CDR (0–2) score and total sessions of non-demented (190), demented (146) and converted (37) were evaluated. In particular, some subjects treated as demented at initial visit later transformed into the non-demented managed by converted type.”

Data1$`MR Delay` <- NULL  
Data1$`MRI ID` <- NULL  
  
CDR <- Data1$CDR   
Data1$CDR <- NULL  
  
Data1$Group <- as.numeric(c("Demented" = "1", "Converted" = "2", "Nondemented" = "3")[Data1$Group])  
Data1$`M/F` <- as.numeric(c("M" = "1", "F" = "2")[Data1$`M/F`])  
Data1$Visit <- as.numeric(c("1" = "1", "2" = "2", "3" = "3", "4" = "4","5" = "5" )[Data1$Visit])  
Data1$Hand <- as.numeric(c("R" = "1", "L" = "2")[Data1$Hand])

second thing to do is to clean the data set of zero values or outliers that can obstruct this research

Lets look at how patients and objects and missing data we have before we delete any missing values.

MissingData <- Data1[rowSums(is.na(Data1)) > 0,]  
  
PatientData = data.frame(  
Name = c("Patients", "Objects", "Dementia Groups"),  
Value = c(length(unique(Data1$`Subject ID`)),length(Data1$`Subject ID`),length(unique(Data1$Group))),  
Missing\_data = c(length(unique(MissingData$`Subject ID`)), length(MissingData$`Subject ID`), unique(MissingData$Group))  
)  
pander(PatientData)

| Name | Value | Missing\_data |
| --- | --- | --- |
| Patients | 150 | 8 |
| Objects | 373 | 19 |
| Dementia Groups | 3 | 1 |

There are 8 patients with missing data, lets view if they are missing only 1 data point or multiple data points.

pander(MissingData, style = "simple", split.table = Inf)

| Subject ID | Group | Visit | M/F | Hand | Age | EDUC | SES | MMSE | eTIV | nWBV | ASF |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| OAS2\_0002 | 1 | 1 | 1 | 1 | 75 | 12 | NA | 23 | 1678 | 0.7363 | 1.046 |
| OAS2\_0002 | 1 | 2 | 1 | 1 | 76 | 12 | NA | 28 | 1738 | 0.7134 | 1.01 |
| OAS2\_0002 | 1 | 3 | 1 | 1 | 80 | 12 | NA | 22 | 1698 | 0.7012 | 1.034 |
| OAS2\_0007 | 1 | 1 | 1 | 1 | 71 | 16 | NA | 28 | 1357 | 0.7481 | 1.293 |
| OAS2\_0007 | 1 | 3 | 1 | 1 | 73 | 16 | NA | 27 | 1364 | 0.727 | 1.286 |
| OAS2\_0007 | 1 | 4 | 1 | 1 | 75 | 16 | NA | 27 | 1372 | 0.71 | 1.279 |
| OAS2\_0063 | 1 | 1 | 2 | 1 | 80 | 12 | NA | 30 | 1430 | 0.737 | 1.228 |
| OAS2\_0063 | 1 | 2 | 2 | 1 | 81 | 12 | NA | 27 | 1453 | 0.721 | 1.208 |
| OAS2\_0099 | 1 | 1 | 2 | 1 | 80 | 12 | NA | 27 | 1475 | 0.7625 | 1.19 |
| OAS2\_0099 | 1 | 2 | 2 | 1 | 83 | 12 | NA | 23 | 1484 | 0.7504 | 1.183 |
| OAS2\_0114 | 1 | 1 | 2 | 1 | 76 | 12 | NA | 27 | 1316 | 0.7268 | 1.333 |
| OAS2\_0114 | 1 | 2 | 2 | 1 | 78 | 12 | NA | 27 | 1309 | 0.7086 | 1.341 |
| OAS2\_0160 | 1 | 1 | 1 | 1 | 76 | 12 | NA | 27 | 1557 | 0.7052 | 1.127 |
| OAS2\_0160 | 1 | 2 | 1 | 1 | 78 | 12 | NA | 29 | 1569 | 0.7042 | 1.119 |
| OAS2\_0181 | 1 | 1 | 2 | 1 | 74 | 12 | NA | 26 | 1171 | 0.7328 | 1.499 |
| OAS2\_0181 | 1 | 2 | 2 | 1 | 75 | 12 | NA | NA | 1169 | 0.7416 | 1.501 |
| OAS2\_0181 | 1 | 3 | 2 | 1 | 77 | 12 | NA | NA | 1159 | 0.7328 | 1.515 |
| OAS2\_0182 | 1 | 1 | 1 | 1 | 73 | 12 | NA | 23 | 1661 | 0.6976 | 1.056 |
| OAS2\_0182 | 1 | 2 | 1 | 1 | 75 | 12 | NA | 20 | 1654 | 0.6961 | 1.061 |
| All the patie | nts are | missing | the SE | S data | point | and pat | ient 1 | 81 is a | lso mis | sing his | MMSE 2 times |

There are 2 possible things to do here, delete or enter the data our self based on the mean of the other data. I don’t really have that big of a data set so i am going to do the latter. 1 data point of the MMSE of patient 181 is measured so i can put this number at the other two NA For the SES data points i am going to take the mean of the other data points of the demented group

DementedSub <- subset(Data1, Group==1)  
DementedSub <- DementedSub %>% drop\_na()  
mean(DementedSub$SES)

## [1] 2.771654

I will round it up to 3 and put this number at the NA’s. And test if there are any NA’s left

Data1$SES[is.na(Data1$SES)] <- 3  
Data1$MMSE[is.na(Data1$MMSE)] <- 26  
  
#Test if it went well  
sum(is.na(Data1))

## [1] 0

There are no longer any NA’s in the dataset, lets continue with the cleaning process

### finding outliers

Now lets take a look at the summary of the date and see if anything stands out

## Subject ID Group Visit M/F Hand   
## Length:373 Min. :1.000 Min. :1.000 Min. :1.000 Min. :1   
## Class :character 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:1.000 1st Qu.:1   
## Mode :character Median :3.000 Median :2.000 Median :2.000 Median :1   
## Mean :2.118 Mean :1.882 Mean :1.571 Mean :1   
## 3rd Qu.:3.000 3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.:1   
## Max. :3.000 Max. :5.000 Max. :2.000 Max. :1   
## Age EDUC SES MMSE eTIV   
## Min. :60.00 Min. : 6.0 Min. :1.000 Min. : 4.00 Min. :1106   
## 1st Qu.:71.00 1st Qu.:12.0 1st Qu.:2.000 1st Qu.:27.00 1st Qu.:1357   
## Median :77.00 Median :15.0 Median :2.000 Median :29.00 Median :1470   
## Mean :77.01 Mean :14.6 Mean :2.488 Mean :27.34 Mean :1488   
## 3rd Qu.:82.00 3rd Qu.:16.0 3rd Qu.:3.000 3rd Qu.:30.00 3rd Qu.:1597   
## Max. :98.00 Max. :23.0 Max. :5.000 Max. :30.00 Max. :2004   
## nWBV ASF   
## Min. :0.6444 Min. :0.8755   
## 1st Qu.:0.7002 1st Qu.:1.0990   
## Median :0.7288 Median :1.1938   
## Mean :0.7296 Mean :1.1955   
## 3rd Qu.:0.7557 3rd Qu.:1.2930   
## Max. :0.8368 Max. :1.5873

When looking at the parameters i dont see anything that stands out, i compare the min and max of every group and see if they are really far apart of the mean/median. The only thing that is really far from the mean is the min of the MMSE group, lets zoom in on that object.

pander(Data1[which.min(Data1$MMSE),], split.table = Inf)

| Subject ID | Group | Visit | M/F | Hand | Age | EDUC | SES | MMSE | eTIV | nWBV | ASF |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| OAS2\_0048 | 1 | 5 | 1 | 1 | 69 | 16 | 1 | 4 | 1701 | 0.6761 | 1.032 |

MMSE is a value between the 0 and 30, if it is lower than 9 that means the patient has severe dementia. This patient is in group 1 which is the demented group so i don’t think it is a outlier, just a patient with severe dementia.

### Remove colums with no meaning

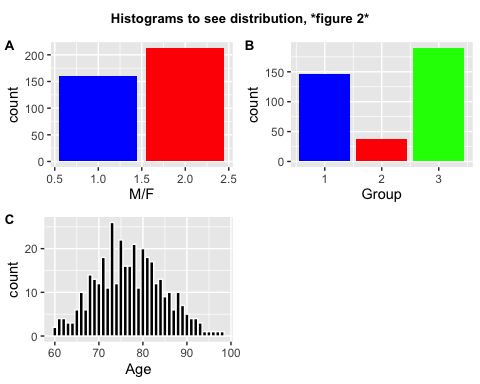
I noticed one more thing when looking at the summary, the min and max of the hand parameter were 1. that means that all patients only are right handed. lets check this one more time with summary

When looking at the Handedness summary we can see that the only variable is 1 for right handed people, because everybody is right handed we can remove the column.

## Testing the dataset

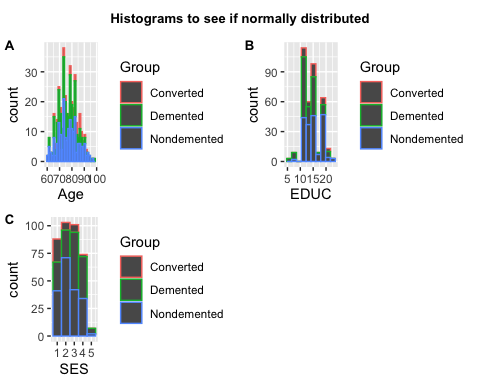
Second thing to do is to look at the underlying distribution and the variation within the dataset

### equal distribution

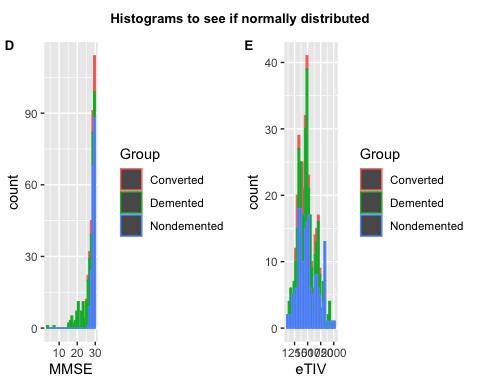
first look at the distribution of the nominal data.  In figure 2 we can see that there is a nice normally distribution of the age and gender parameters. In the dementia status group the converted group has much less patients then the other two, in this case it doesn’t really matter because the coverted group are patients that were first diagnosed with dementia but with a second test were labeled nondemented. so it really is part of the non demented group but is interesting to keep apart to see if it will be some kind of middle group

###Demonstrating normally distributed

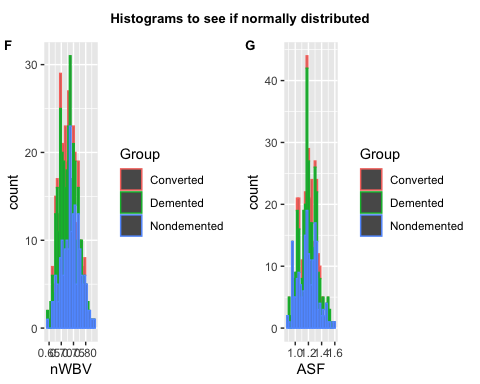
Data1$Group <- as.character(c("1" = "Demented", "2" = "Converted", "3" = "Nondemented")[Data1$Group])  
p1 <- ggplot(Data1, aes(x= Age, color=Group)) +   
 geom\_histogram(bins = 30)  
  
p2 <-ggplot(Data1, aes(x= EDUC, color=Group)) +   
 geom\_histogram(bins = 10)  
  
p3 <-ggplot(Data1, aes(x= SES, color=Group)) +   
 geom\_histogram(bins = 5)  
  
p4 <-ggplot(Data1, aes(x= MMSE, color=Group)) +   
 geom\_histogram(bins = 30)  
  
p5 <-ggplot(Data1, aes(x= eTIV, color=Group)) +   
 geom\_histogram(bins = 30)  
  
p6 <-ggplot(Data1, aes(x= nWBV, color=Group)) +   
 geom\_histogram(bins = 30)  
  
p7 <-ggplot(Data1, aes(x= ASF, color=Group)) +   
 geom\_histogram(bins = 30)  
  
title <- ggdraw() + draw\_label("Histograms to see if normally distributed", fontface='bold', size = 10)  
plot\_1 <- plot\_grid(p1,p2,p3, labels = c('A','B','C'), label\_size = 10)  
plot\_grid(title, plot\_1, ncol=1, rel\_heights=c(0.1, 1))



title <- ggdraw() + draw\_label("Histograms to see if normally distributed", fontface='bold', size = 10)  
plot\_2 <- plot\_grid(p4,p5, labels = c('D','E'), label\_size = 10)  
plot\_grid(title, plot\_2, ncol=1, rel\_heights=c(0.1, 1))



title <- ggdraw() + draw\_label("Histograms to see if normally distributed", fontface='bold', size = 10)  
plot\_3 <- plot\_grid(p6,p7, labels = c('F','G'), label\_size = 10)  
plot\_grid(title, plot\_3, ncol=1, rel\_heights=c(0.1, 1))



Data1$Group <- as.numeric(c("Demented" = "1", "Converted" = "2", "Nondemented" = "3")[Data1$Group])

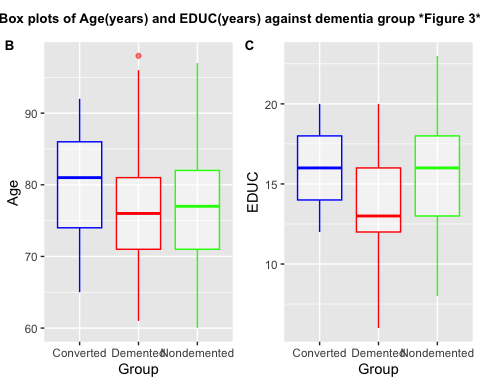
All the parameters are normally distributed exept for MMSE, this score is mostly between the 20 and 30. This is because all the people that are non demented probably have a score of 24 of higher (explained at the codebook).

This will maybe cause some over fitting so we will need to keep it in mind when making the model.

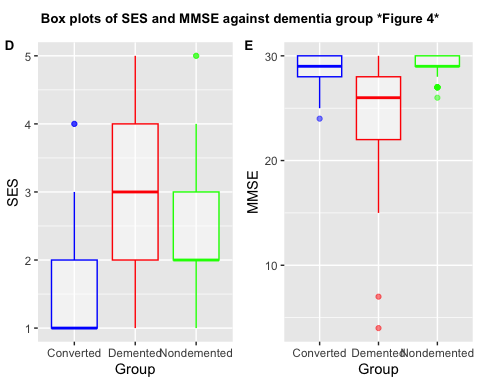
## Finding variation

Now lets take a look at how the parameters are coherent to the dementia status. This will be done via box plots.

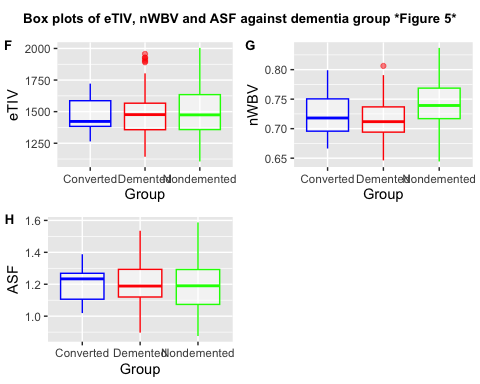
Data1$Group <- as.character(c("1" = "Demented", "2" = "Converted", "3" = "Nondemented")[Data1$Group])  
  
p2 <- ggplot(Data1, aes(x = Group, y= Age)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
   
p3 <- ggplot(Data1, aes(x = Group, y= EDUC)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
   
p4 <- ggplot(Data1, aes(x = Group, y= SES)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
   
p5 <- ggplot(Data1, aes(x = Group, y= MMSE)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
   
p6 <- ggplot(Data1, aes(x = Group, y= eTIV)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
   
p7 <- ggplot(Data1, aes(x = Group, y= nWBV)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
  
p8 <- ggplot(Data1, aes(x = Group, y= ASF)) +  
 geom\_boxplot(alpha = 0.5, color= c('blue', 'red', 'green'))   
   
  
title <- ggdraw() + draw\_label("Box plots of Age(years) and EDUC(years) against dementia group \*Figure 3\*", fontface='bold', size = 10)  
plot\_1 <- plot\_grid(p2,p3, labels = c('B','C'), label\_size = 10)  
plot\_grid(title, plot\_1, ncol=1, rel\_heights=c(0.1, 1))



title <- ggdraw() + draw\_label("Box plots of SES and MMSE against dementia group \*Figure 4\*", fontface='bold', size = 10)  
plot\_2 <- plot\_grid(p4,p5, labels = c('D','E'), label\_size = 10)  
plot\_grid(title, plot\_2, ncol=1, rel\_heights=c(0.1, 1))



title <- ggdraw() + draw\_label("Box plots of eTIV, nWBV and ASF against dementia group \*Figure 5\*", fontface='bold', size = 10)  
plot\_3 <- plot\_grid(p6,p7,p8, labels = c('F','G','H'), label\_size = 10)  
plot\_grid(title, plot\_3, ncol=1, rel\_heights=c(0.1, 1))



Data1$Group <- as.numeric(c("Demented" = "1", "Converted" = "2", "Nondemented" = "3")[Data1$Group])

The things to look for in a boxplot is to see if the values of the dementia groups are far apart of each other with the parameters, Because this means that the influence of the parameters effects the groups different and is therefor maybe a good parameters for correlation and the machine learning model

The parameters with the most difference are: Educatie, SES, MMSE, nWBV These parameters are mostly the ones that are going the be used to make a model.

## Finding correlation

A way to find correlation within the data set is to use a anova test. We will use a anova test to see which parameter is correlated the most with the dementia group. The anova test uses the eta-squared to show the correlation.

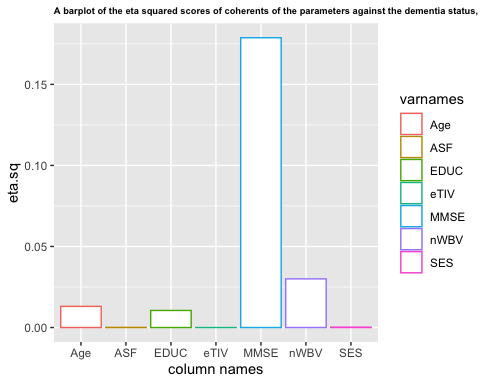
Eta-squared ranges from 0 to 1 and is interpreted as follows:

η² = 0: There is no effect of the independent variable on the dependent variable. η² ≈ 0.01: A small effect. η² ≈ 0.06: A medium effect. η² ≈ 0.14: A large effect.

The data is normally distributed except for MMSE so this test is just a ranking

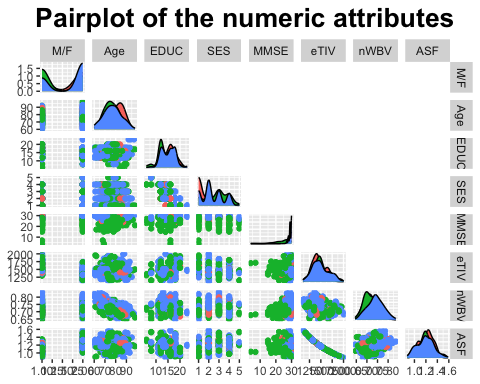
####anova test

Data1$Visit <- NULL  
  
one.way <- aov(Group ~ Age + EDUC +SES + MMSE + eTIV + nWBV + ASF, data = Data1)  
  
anovatest <- data.frame(unclass(etaSquared(one.way)), check.names = FALSE, stringsAsFactors = FALSE)  
anovatest <- as\_tibble(anovatest)  
anovanames <- c("Age","EDUC","SES","MMSE","eTIV","nWBV","ASF")  
anovatest <- anovatest %>% mutate(varnames = anovanames)  
  
ggplot(data=anovatest, aes(x=varnames, y= eta.sq, color = varnames)) +  
 geom\_bar(stat="identity", fill = 'white') +  
 xlab("column names") +  
 ggtitle("A barplot of the eta squared scores of coherents of the parameters against the dementia status, \*figure 6\*") + theme(plot.title = element\_text(size = 7, face = "bold"))

 When looking at the outcome we see that MMSE, nWBV, EDUC and Age have a effect on the dementia status. We will be taking these parameters for further analysis.

####heatmap with pairplot Another way to check for correlation is with a correlation matrix that is shown in a heat map. Before we do that, lets check if the parameters are coherent to each other with a pairplot. In this plot we lay each parameter out against one and another

Data1$Group <- as.character(c("1" = "Demented", "2" = "Converted", "3" = "Nondemented")[Data1$Group])  
  
Nummeric\_columns <- select\_if(Data1, is.numeric)  
  
ggpairs( Nummeric\_columns, ggplot2::aes(colour=Data1$Group),  
progress = F, upper = "blank") +  
labs(title = "Pairplot of the numeric attributes") +  
theme(plot.title = element\_text(hjust = 0.5, face = "bold", size = 20))



Data1$Group <- as.numeric(c("Demented" = "1", "Converted" = "2", "Nondemented" = "3")[Data1$Group])

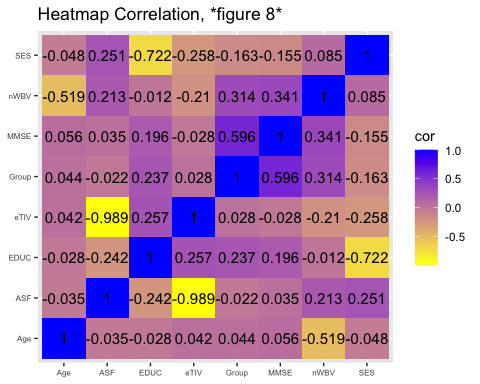
Looking at this pair plot something strange stands out. The eTIV value and the ASF value are almost entirely coherent to each other. This means that the probably have a correlation number of 1. This means that the say the same thing, we can therefor delete one of the parameters. Let look at the heat map to be cetrain.

Using a heat map is a great way to visually represent the correlations between different parameters with colors.

#create a core matrix  
  
colnames(Data1) <- c("Subject-ID","Group","M/F","Age","EDUC","SES","MMSE","eTIV","nWBV","ASF")  
  
Data\_names <- c("Group","Age","EDUC","SES","MMSE","eTIV","nWBV","ASF")  
  
Dementia\_matrix <- cor(Data1[, -c(1,3)])  
Dementia\_matrix <- as\_tibble(Dementia\_matrix)  
Dementia\_matrix <- Dementia\_matrix %>% mutate(varnames = Data\_names)  
  
  
(cor\_matrix\_long <- pivot\_longer(Dementia\_matrix,   
 cols = all\_of(Data\_names),   
 names\_to = "variable",   
 values\_to = "cor"))

## # A tibble: 64 × 3  
## varnames variable cor  
## <chr> <chr> <dbl>  
## 1 Group Group 1   
## 2 Group Age 0.0442  
## 3 Group EDUC 0.237   
## 4 Group SES -0.163   
## 5 Group MMSE 0.596   
## 6 Group eTIV 0.0280  
## 7 Group nWBV 0.314   
## 8 Group ASF -0.0215  
## 9 Age Group 0.0442  
## 10 Age Age 1   
## # ℹ 54 more rows

round\_cor <- cor\_matrix\_long %>% mutate\_at(vars(cor), funs(round(., 3)))  
  
  
ggplot(data = round\_cor, aes(x=varnames, y=variable, fill=cor)) +  
geom\_tile() +  
theme(axis.text = element\_text(size = 6)) +  
labs(x=NULL, y=NULL, title="Heatmap Correlation, \*figure 8\*") +  
scale\_fill\_gradient(high = "blue", low = "yellow" ) +  
 geom\_text(aes(varnames, variable, label = cor), color = "black", size = 4)

 A positive correlation means that both variables increase or decrease together. A negative correlation means that one variable increases while the other variable decreases.

The things we are looking for are the correlation number between eTIV and ASF and the correlation numbers between the parameters and the dementia group.

We can see that the correlation number between ASF and eTIV is almost -1. This means that the pair plot was right and that we can delete one of the other. I will delete the ASF value because the eTIV value has a slightly higher correlation number.

The groups with the highest correlation number with the dementia group are the MMSE, EDUC, SES, and nWBV parameters.

These parameters were also very high in the anova test, except voor the SES parameter.

Because of the high negative correlation with the dementia group and SES, i think it can be of an influence for the machine learning process. But it had a low number with the anova test, so to be certain lets add an extra test to check.

chisq.test(Data1$Group, Data1$SES)

##   
## Pearson's Chi-squared test  
##   
## data: Data1$Group and Data1$SES  
## X-squared = 49.912, df = 8, p-value = 4.248e-08

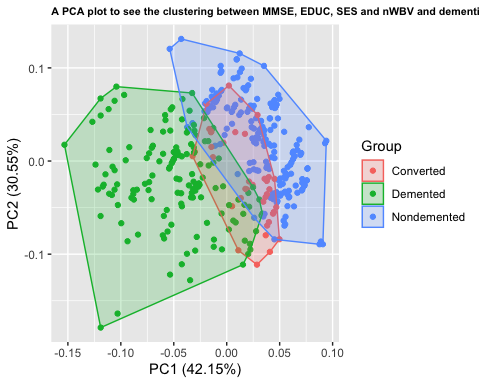
The p value is under 0.005 so that means there is a correlation between the two. So we now have a clear view of the parameters that are highest correlated with the dementia group. Before we are going to the machine learning process we are going to check how good the model is probably going to be

## Principal component analysis

The use of PCA is to represent a multivariate data table as smaller set of variables (summary indices) in order to observe trends, jumps and clusters. This overview may uncover how good of a model we can make.

Looking for clusters with a principal component analysis plot using the 4 most correlated parameters with dementia. MMSE, EDUC, nWBV and SES

#pca plot  
library(ggfortify)  
library(cluster)  
  
Data1$Group <- as.numeric(c("Demented" = "1", "Converted" = "2", "Nondemented" = "3")[Data1$Group])  
  
Pca\_df <- data.frame(Data1$MMSE, Data1$EDUC, Data1$nWBV, Data1$SES, Data1$Group)  
pca\_res <- prcomp(Pca\_df, scale. = TRUE)  
  
  
Data1$Group <- as.character(c("1" = "Demented", "2" = "Converted", "3" = "Nondemented")[Data1$Group])  
  
autoplot(pca\_res, data = Data1, colour = 'Group',frame = TRUE) +  
 ggtitle("A PCA plot to see the clustering between MMSE, EDUC, SES and nWBV and dementia group \*Figure 9\*") +  
 theme(plot.title = element\_text(size = 8, face = "bold"))

 Clearly 3 cluster groups can be seen, demented and non demented lie nicely apart with little overlap and the converted group sits as a clear middle group in between with more similarity to non demented as the article had already indicated. What can be gleaned from this is that 3 groups are with clearly different values so making a machine learning model to predict the 3 groups is probably quite possible. With the PC1 and PC2 component there is 72 % variation which means is that with 2 parameters 72 procent of the data can be correctly identified. This means that the accuracy of our model will probably be around that number. If we have a model that has 99 % accuracy we are probably doing something wrong because the PCA plot is showing us that that won’t be possible.

That concludes the first fase, now lets make a model.

## Dataset for weka model

In the context of addressing my research question, it became clear that the use of the “Converted” group might not be essential. This subgroup comprises patients initially classified as demented but were later reclassified as non-demented following a secondary evaluation. Notably, the PCA plot analysis illustrated that the “Converted” group exhibits a distribution leaning more towards the non-dementia side than the demented group.

In the pursuit of model optimization, we will be taking three distinct datasets to get the highest level of accuracy. These datasets were organized as follows: one in which the “Converted” group is retained, another in which it is reclassified as demented, and a third in which it is reclassified as non-demented.

Data1$CDR = CDR  
  
#converted as a solo group  
write\_csv(Data1, "Filterd\_dementia\_data.csv")  
  
#Converted with demented  
Data\_ConToDem <- Data1  
Data\_ConToDem$Group[Data\_ConToDem$Group == "Converted"] <- "Demented"  
write\_csv(Data\_ConToDem, "Data\_ConToDem.csv")  
  
#Converted with Nondemented   
Data\_ConToNon <- Data1  
Data\_ConToNon$Group[Data\_ConToNon$Group == "Converted"] <- "Nondemented"  
write\_csv(Data\_ConToNon, "Data\_ConToNon.csv")

# Weka analysis

In this section, we are delving into the analysis of various machine learning algorithms using the Weka tool. We consider different algorithms with distinct hyperparameters, datasets with varying characteristics (such as the presence of NAs), and preprocessing techniques like log transformation and standardization/normalization. The goal is to explore the impact of these factors on the performance of the algorithms.

First lets look at the dataset with the converted group

Test <- ResultsFromWekaFilterd$Key\_Scheme[c(1,101,201,301,401,501,601,701,801,901)]  
Accuracy <- colMeans(matrix(ResultsFromWekaFilterd$Percent\_correct,nrow=100))  
Number\_FP <- colMeans(matrix(ResultsFromWekaFilterd$Num\_false\_positives, nrow = 100))  
Number\_TP <- colMeans(matrix(ResultsFromWekaFilterd$Num\_true\_positives, nrow = 100))  
Number\_FN <- colMeans(matrix(ResultsFromWekaFilterd$Num\_false\_negatives, nrow = 100))  
Number\_TN <- colMeans(matrix(ResultsFromWekaFilterd$Num\_true\_negatives, nrow = 100))  
ROC\_area <- colMeans(matrix(ResultsFromWekaFilterd$Area\_under\_ROC, nrow = 100))  
  
ResultsDF <- data.frame(Test, Accuracy, Number\_TP, Number\_FP, Number\_TN, Number\_FN, ROC\_area)  
pander::pander(ResultsDF, style = "simple", split.table = Inf)

| Test | Accuracy | Number\_TP | Number\_FP | Number\_TN | Number\_FN | ROC\_area |
| --- | --- | --- | --- | --- | --- | --- |
| weka.classifiers.rules.ZeroR | 50.95 | 19 | 18.3 | 0 | 0 | 0.5 |
| weka.classifiers.rules.OneR | 76.68 | 18 | 7 | 11.3 | 1 | 0.7825 |
| weka.classifiers.trees.J48 | 76.06 | 17.78 | 7.03 | 11.27 | 1.22 | 0.7876 |
| weka.classifiers.trees.RandomForest | 76.14 | 17.96 | 7.21 | 11.09 | 1.04 | 0.8753 |
| weka.classifiers.lazy.IBk | 75.09 | 17.69 | 6.83 | 11.47 | 1.31 | 0.8804 |
| weka.classifiers.meta.CostSensitiveClassifier | 76.06 | 17.78 | 7.03 | 11.27 | 1.22 | 0.7876 |
| weka.classifiers.functions.SMO | 73.52 | 17.67 | 8.26 | 10.04 | 1.33 | 0.7415 |
| weka.classifiers.functions.SimpleLogistic | 75.36 | 16.9 | 6.29 | 12.01 | 2.1 | 0.8753 |
| weka.classifiers.bayes.NaiveBayes | 74.46 | 17.73 | 7.43 | 10.87 | 1.27 | 0.8503 |
| weka.classifiers.meta.Bagging | 76.89 | 17.47 | 6.36 | 11.94 | 1.53 | 0.8894 |
| The group is so small that it is not considered | in the algo | rithms, whic | h actually l | eads to a de | crease in ac | curacy. |
| The group was interesting for the EDA but not fo | r model cre | ation. I wil | l continue t | he process w | ith the conv | erted group changed to Non-demented and Demented and see which dataset gives me the best results. |

Converted to Non-demented

Test <- ResultsFromWekaNondemented$Key\_Scheme[c(1,101,201,301,401,501,601,701,801,901)]  
Accuracy <- colMeans(matrix(ResultsFromWekaNondemented$Percent\_correct,nrow=100))  
Number\_FP <- colMeans(matrix(ResultsFromWekaNondemented$Num\_false\_positives, nrow = 100))  
Number\_TP <- colMeans(matrix(ResultsFromWekaNondemented$Num\_true\_positives, nrow = 100))  
Number\_FN <- colMeans(matrix(ResultsFromWekaNondemented$Num\_false\_negatives, nrow = 100))  
Number\_TN <- colMeans(matrix(ResultsFromWekaNondemented$Num\_true\_negatives, nrow = 100))  
ROC\_area <- colMeans(matrix(ResultsFromWekaNondemented$Area\_under\_ROC, nrow = 100))  
  
ResultsDF <- data.frame(Test, Accuracy, Number\_TP, Number\_FP, Number\_TN, Number\_FN, ROC\_area)  
pander::pander(ResultsDF, style = "simple", split.table = Inf)

| Test | Accuracy | Number\_TP | Number\_FP | Number\_TN | Number\_FN | ROC\_area |
| --- | --- | --- | --- | --- | --- | --- |
| weka.classifiers.rules.ZeroR | 60.86 | 22.7 | 14.6 | 0 | 0 | 0.5 |
| weka.classifiers.rules.OneR | 84.72 | 21 | 4 | 10.6 | 1.7 | 0.8256 |
| weka.classifiers.trees.J48 | 84.35 | 20.86 | 4 | 10.6 | 1.84 | 0.8275 |
| weka.classifiers.trees.RandomForest | 84.37 | 21.15 | 4.28 | 10.32 | 1.55 | 0.9196 |
| weka.classifiers.lazy.IBk | 83.47 | 21.47 | 4.94 | 9.66 | 1.23 | 0.8983 |
| weka.classifiers.meta.CostSensitiveClassifier | 84.35 | 20.86 | 4 | 10.6 | 1.84 | 0.8275 |
| weka.classifiers.functions.SMO | 80.52 | 21.31 | 5.88 | 8.72 | 1.39 | 0.7684 |
| weka.classifiers.functions.SimpleLogistic | 84.03 | 20.74 | 4 | 10.6 | 1.96 | 0.9263 |
| weka.classifiers.bayes.NaiveBayes | 82.98 | 21.54 | 5.19 | 9.41 | 1.16 | 0.9074 |
| weka.classifiers.meta.Bagging | 83.76 | 20.75 | 4.11 | 10.49 | 1.95 | 0.9121 |

Converted to demented

Test <- ResultsFromWekaDemented$Key\_Scheme[c(1,101,201,301,401,501,601,701,801,901)]  
Accuracy <- colMeans(matrix(ResultsFromWekaDemented$Percent\_correct,nrow=100))  
Number\_FP <- colMeans(matrix(ResultsFromWekaDemented$Num\_false\_positives, nrow = 100))  
Number\_TP <- colMeans(matrix(ResultsFromWekaDemented$Num\_true\_positives, nrow = 100))  
Number\_FN <- colMeans(matrix(ResultsFromWekaDemented$Num\_false\_negatives, nrow = 100))  
Number\_TN <- colMeans(matrix(ResultsFromWekaDemented$Num\_true\_negatives, nrow = 100))  
ROC\_area <- colMeans(matrix(ResultsFromWekaDemented$Area\_under\_ROC, nrow = 100))  
  
ResultsDF <- data.frame(Test, Accuracy, Number\_TP, Number\_FP, Number\_TN, Number\_FN, ROC\_area)  
pander::pander(ResultsDF, style = "simple", split.table = Inf)

| Test | Accuracy | Number\_TP | Number\_FP | Number\_TN | Number\_FN | ROC\_area |
| --- | --- | --- | --- | --- | --- | --- |
| weka.classifiers.rules.ZeroR | 50.95 | 19 | 18.3 | 0 | 0 | 0.5 |
| weka.classifiers.rules.OneR | 78.2 | 17.86 | 6.99 | 11.31 | 1.14 | 0.779 |
| weka.classifiers.trees.J48 | 77.35 | 16.33 | 5.78 | 12.52 | 2.67 | 0.8047 |
| weka.classifiers.trees.RandomForest | 78.58 | 17.56 | 6.55 | 11.75 | 1.44 | 0.8776 |
| weka.classifiers.lazy.IBk | 78.87 | 16.46 | 5.34 | 12.96 | 2.54 | 0.8761 |
| weka.classifiers.meta.CostSensitiveClassifier | 77.35 | 16.33 | 5.78 | 12.52 | 2.67 | 0.8047 |
| weka.classifiers.functions.SMO | 77.79 | 16.79 | 6.08 | 12.22 | 2.21 | 0.7759 |
| weka.classifiers.functions.SimpleLogistic | 78.37 | 16.29 | 5.36 | 12.94 | 2.71 | 0.8744 |
| weka.classifiers.bayes.NaiveBayes | 79.14 | 17.83 | 6.61 | 11.69 | 1.17 | 0.864 |
| weka.classifiers.meta.Bagging | 80.92 | 17 | 5.12 | 13.18 | 2 | 0.8922 |
| When looking at the results that the weka algori | tmes are gi | ving us it i | s clear that | the dataset | with the co | nverted group changed to Non-Demented has better scores at every algoritme then the dataset with converted changed to demented. This means that i will continue working with the converted group changed to Non demented dataset. I will now begin changing the hyper parameters of all the algoritmes and getting the best results with every one of them. |

The first adjustment I make involves the number of attributes used in my model. Upon reviewing the EDA results, it becomes evident that not all attributes exhibit a high correlation with the dementia status. I aim for my final model to have fewer attribute inputs, simplifying its usability since users won’t need to input a multitude of attributes.

To assess this, I employ the ‘Select attributes’ function in Weka to identify which attributes have the most impact when constructing a model. I conduct two tests. The initial test involves using ‘cfsSubsetEval’ with the ‘Best first’ method, resulting in the selection of the attributes: SES, MMSE, and nWBV.

For the second test, I employ ‘WrapperSubsetEval’ with the ‘Best first’ method and the J48 classifier. The selected attributes are MMSE and SES.

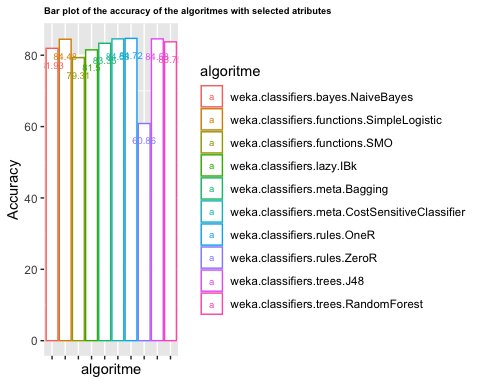
Considering these outcomes and understanding the significance of attributes, I opt to retain only three attributes—SES, MMSE, and nWBV—for constructing the model.

Now, let’s reassess all the algorithms with the updated dataset and determine the most effective algorithm along with the optimal hyperparameters.

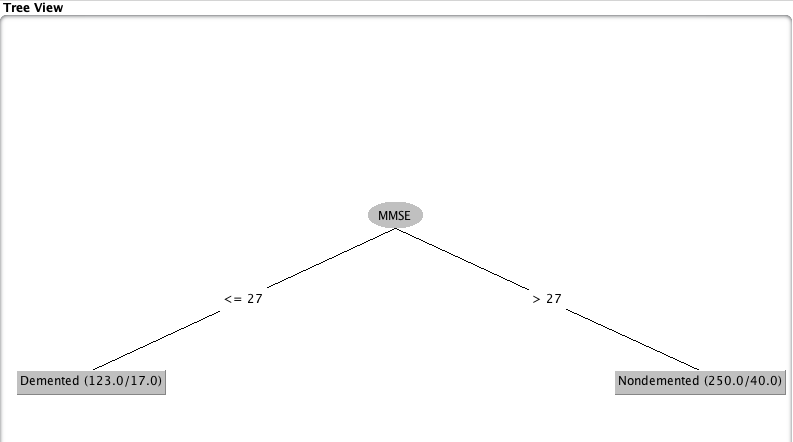
ResultsFromWekaAtributes <- read.arff(file = 'WekaResultsAfterAtribute.arff')  
Test <- ResultsFromWekaAtributes$Key\_Scheme[c(1,101,201,301,401,501,601,701,801,901)]  
Accuracy <- colMeans(matrix(ResultsFromWekaAtributes$Percent\_correct,nrow=100))  
Number\_FP <- colMeans(matrix(ResultsFromWekaAtributes$Num\_false\_positives, nrow = 100))  
Number\_TP <- colMeans(matrix(ResultsFromWekaAtributes$Num\_true\_positives, nrow = 100))  
Number\_FN <- colMeans(matrix(ResultsFromWekaAtributes$Num\_false\_negatives, nrow = 100))  
Number\_TN <- colMeans(matrix(ResultsFromWekaAtributes$Num\_true\_negatives, nrow = 100))  
ROC\_area <- colMeans(matrix(ResultsFromWekaAtributes$Area\_under\_ROC, nrow = 100))  
  
ResultsDF <- data.frame(Test, Accuracy, Number\_TP, Number\_FP, Number\_TN, Number\_FN, ROC\_area)  
pander(ResultsDF, style = "simple", split.table = Inf)

| Test | Accuracy | Number\_TP | Number\_FP | Number\_TN | Number\_FN | ROC\_area |
| --- | --- | --- | --- | --- | --- | --- |
| weka.classifiers.rules.ZeroR | 60.86 | 22.7 | 14.6 | 0 | 0 | 0.5 |
| weka.classifiers.rules.OneR | 84.72 | 21 | 4 | 10.6 | 1.7 | 0.8256 |
| weka.classifiers.trees.J48 | 84.58 | 21.01 | 4.06 | 10.54 | 1.69 | 0.8255 |
| weka.classifiers.trees.RandomForest | 83.75 | 21.05 | 4.41 | 10.19 | 1.65 | 0.8906 |
| weka.classifiers.lazy.IBk | 81.5 | 20.26 | 4.46 | 10.14 | 2.44 | 0.8534 |
| weka.classifiers.meta.CostSensitiveClassifier | 84.58 | 21.01 | 4.06 | 10.54 | 1.69 | 0.8255 |
| weka.classifiers.functions.SMO | 79.31 | 22.58 | 7.6 | 7 | 0.12 | 0.7373 |
| weka.classifiers.functions.SimpleLogistic | 84.48 | 21.01 | 4.1 | 10.5 | 1.69 | 0.8831 |
| weka.classifiers.bayes.NaiveBayes | 81.93 | 21.94 | 5.98 | 8.62 | 0.76 | 0.8839 |
| weka.classifiers.meta.Bagging | 83.35 | 20.7 | 4.21 | 10.39 | 2 | 0.878 |

test\_x <- ResultsFromWekaAtributes$Key\_Scheme[c(1,101,201,301,401,501,601,701,801,901)]  
test\_y <- colMeans(matrix(ResultsFromWekaAtributes$Percent\_correct,nrow=100))  
test\_y <- round(test\_y, digits = 2)  
test\_df <- data.frame(algoritme = test\_x,  
 Accuracy = test\_y)  
p <- ggplot(data=test\_df,  
 aes(x=algoritme,y=Accuracy,color = algoritme))+ geom\_bar(stat="identity", fill = 'white') + xlab("algoritme")+ ggtitle("\*") + theme(plot.title = element\_text(size = 7, face = "bold")) + geom\_text(aes(label = Accuracy), size = 2.5, hjust = 0.5, vjust = 3, position = "stack") +  
 theme(axis.text.x=element\_blank())  
p + ggtitle("Bar plot of the accuracy of the algoritmes with selected atributes")

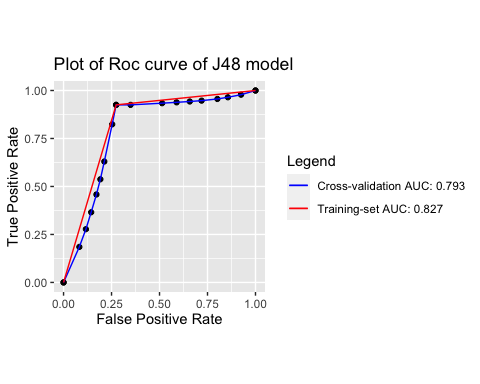
 Looking at the accuracy of the classifiers i can see that the difference is low between all the algorithms see i need to make my chose not based on accuracy of the algorithms. On closer inspection, the classifiers also dont have a significant difference in False negatives. This is a importent factor because i don’t want people with demented wrongly classified.

Another factor is the simplicity of the models, some models take a longer time or need all the 3 attributes for making the model. But one model stands out which only uses 1 attribute and also haves the highest accuracy score and lowest rate of false negatives, the j48 model.

 The J48 model only uses the mmse score to make a decision for the classification of the dementia status of a patient. Because of its simplicity and high accuracy score and low false negative rate i will pick this model

Last thing to do is to look at the ROC curve if the algorithm covers most of the data.

#install.packages("pROC")  
library(pROC)  
  
RocDataNonDem <- read.arff(file = 'RocDataNondem.arff')  
RocDataNonDemTrainSet <- read.arff(file = 'RocDataNonDemTrainSet.arff')  
  
colors <- c("Cross-validation AUC: 0.793" = "blue", "Training-set AUC: 0.827" = "red")  
  
p <- ggplot(RocDataNonDem, aes(x = `False Positive Rate`, y = `True Positive Rate`)) + geom\_point() + geom\_point(data = RocDataNonDemTrainSet) + geom\_line(aes(color = "Cross-validation AUC: 0.793")) + geom\_line(data = RocDataNonDemTrainSet, aes(color = "Training-set AUC: 0.827")) +  
 labs(color = 'Legend') + scale\_color\_manual(values = colors)  
p <- p + coord\_fixed()  
p + ggtitle("Plot of Roc curve of J48 model")

 The above ROC plot illustrates the performance of the J48 algorithm, evaluated through cross-validation and with the training set. A positive aspect is that the ROC scores of the two lines are closely aligned, indicating minimal overfitting. Although the score doesn’t reach perfection (1), the J48 algorithm emerges as the highest-scoring algorithm for this dataset.

#referenties

<https://meetinstrumentenzorg.nl/instrumenten/mini-mental-state-examination-gestandaardiseerde-versie/>