Predict patient's extent of cognitive impairment and identify most important determined measures

Jennifer Ci, Lan Shui, Lishan Wang

2022-12-04

1 Question and background information on the data

The dataset that we will look at merges together several of the key variables from various case report forms and biomarker lab summaries across the Alzheimer Disease Neuroimaging Initiative (ADNI). Specifically, it contains demographic information like age, gender and education level, cognitive test scores like MMSE, Montreal Cognitive Assessment (MoCA) and CDR-SB, and cerebrospinal fluid markers like total tau (t-tau), phospho-tau (p-tau) and Abeta42, as well as the diagnosis group information which is the outcome we are interested. This ordinal outcome variable describes the extent of cognitive impairment which from having least severe cognitive impairment to having most severe cognitive impairment are Cognitive Normal (CN), Significant Memory Concern (SMC), Early Mild cognitive impairment (EMCI), Late Mild cognitive impairment (LMCI) and Dementia (AD). Though it's a longitudinal dataset, we will only focus on the baseline values for the purpose of this analysis which is to find out the predictors significant in predicting the cognitive impairment level.

1.1 Loading all the necessary packages and dataset.

```
##--------------Load data
biomarker_merged <- read.csv("~/Desktop/Biostat/BIOST_544/final_project/ADNIMERGE.csv",
                             comment.char="#",na.strings=c("","NA"))
##--------------Package
library(dplyr)
library(stringr)
library(readr)
library(ordinalForest)
library(missForest)
library(naniar)
library(tidyr)
library(purrr)
library(ggplot2)
library(randomForest)
library(ranger)
```

1.2 Cleaning the data

The original dataset contains 16037 observation with 116 variables. After including only the baseline datapoints and excluding useless variables like subject ID, we left with 2396 observations and 49 variables.

```
#filter baseline
baseline<-filter(biomarker_merged,VISCODE=="bl")
str_detect(colnames(baseline),"bl")
col_index<-which(!str_detect(colnames(baseline),"bl"))
baseline_sub<-baseline[,col_index]

baseline_sub$DX<-baseline$DX_bl
baseline_sub$DX<-as.factor(baseline_sub$DX)
baseline_sub<-filter(baseline_sub,DX !="")


#And exclude the variables containing ID information
#I also exclude variables "PTETHCAT","PTRACCAT","PTMARRY" since I don't think
#they are useful in our prediction model
baseline_sub2 <- baseline_sub[,-which(names(baseline_sub) %in%
                                        c("VISCODE","update_stamp","SITE","RID",
                                          "PTID","ORIGPROT","COLPROT","Month","M",
                                          "FSVERSION","FLDSTRENG","EXAMDATE",
                                          "LDELTOTAL_BL","PTETHCAT","PTRACCAT",
                                          "PTMARRY","IMAGEUID"))]
```

2 Features

2.1 Scientific meaning of important variables in this dataset:

AGE Age at baseline
PTGENDER Sex
PTEDUCAT Education
CDRSB (Clinical Dementia Rating)
MMSE (Mini Mental State Exam)
FAQ (Functional Assessment Questionnaire)
LDELTOTAL (Logical Memory - Delayed Recall)
mPACCtrailsB (ADNI modified Preclinical Alzheimer's Cognitive Composite (PACC) with Trails B)
mPACCdigit (ADNI modified Preclinical Alzheimer's Cognitive Composite (PACC) with Digit Symbol Substitution)
EcogSPTotal (total everyday cognition test score)
EcogSPPlan (everyday cognition test score upon plan function)
EcogSPMem (everyday cognition test score upon memory function)
EcogSPLang (everyday cognition test score upon language function)
EcogSPVisspat (everyday cognition test score upon visual and spatial function)

The full description of the variable information can be found at https://adni.bitbucket.io/reference/adnimerge.html.

2.2 Converting the remaining variables properly into numeric or factor.

```
#Converting the original dataset to numeric or factor
baseline_sub2$PTGENDER=as.factor(baseline_sub2$PTGENDER)
baseline_sub2$APOE4<-as.factor(baseline_sub2$APOE4)
```

```
 turn_to_numeric=function(a){
   a=as.numeric(a)
 }

baseline_sub2[,-which(names(baseline_sub2)%in%c("DX","PTGENDER","APOE4"))]<-
  apply(baseline_sub2[,-which(names(baseline_sub2)%in%
                                c("DX","PTGENDER","APOE4"))],2,turn_to_numeric)
```

2.3 Numerical/Categorical Variables

The current dataset contains 46 numeric variables and 3 categorical variables. The density plots are shown below for the numeric variables. Among the predictors, there are categorical variables like "PTGENDER" (indicator variable of gender), APOE4 (Number of APOEe4 alleles: 0, 1, 2) and continuous variables like age in years, education in years. The outcome variable DX is one of the categorical variable.

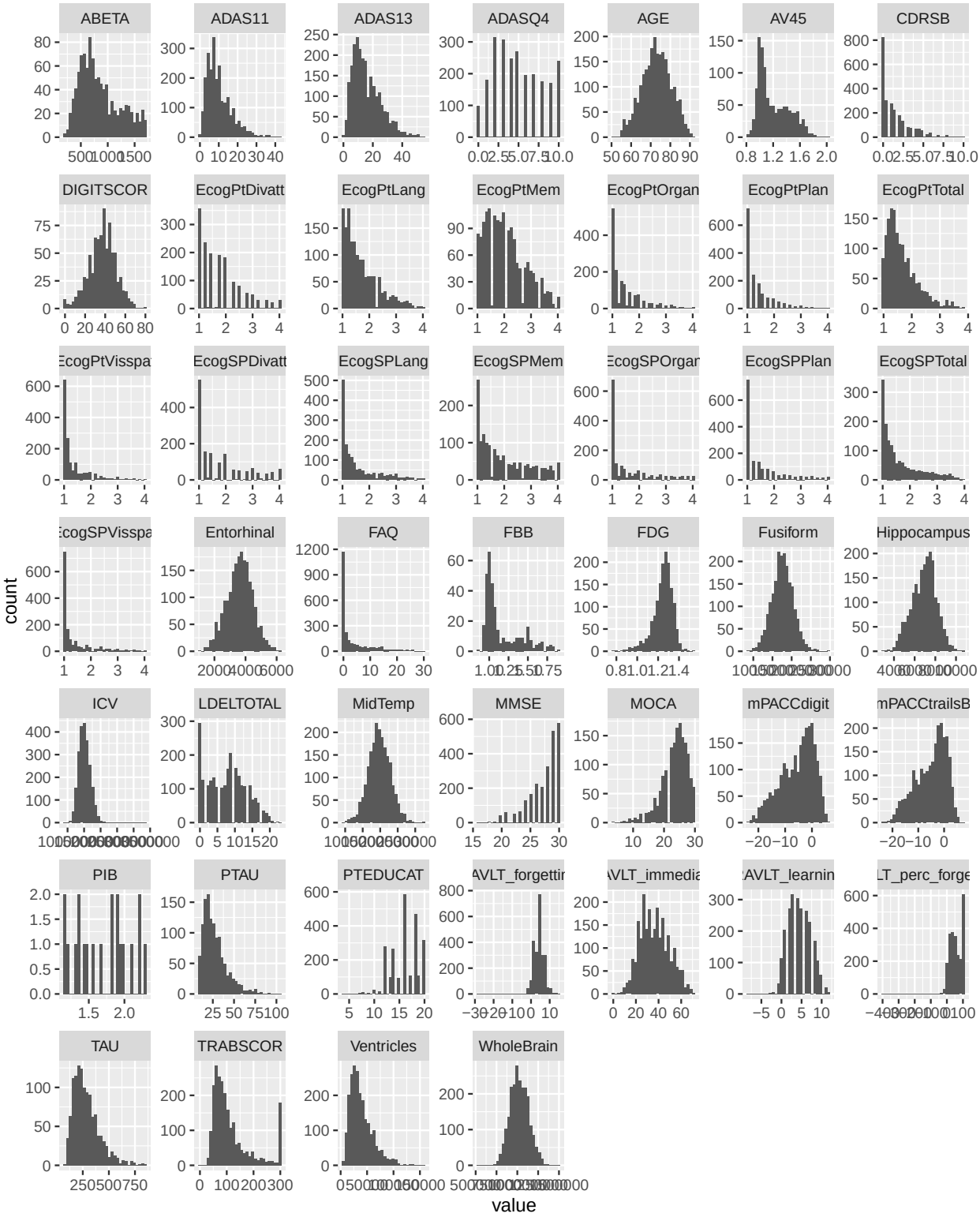
```
sapply(baseline_sub2, is.numeric) %>%
  which() %>%
  names()
```

```
##  [1] "AGE"                "PTEDUCAT"           "FDG"
##  [4] "PIB"                "AV45"               "FBB"
##  [7] "ABETA"              "TAU"                "PTAU"
## [10] "CDRSB"              "ADAS11"             "ADAS13"
## [13] "ADASQ4"             "MMSE"               "RAVLT_immediate"
## [16] "RAVLT_learning"     "RAVLT_forgetting"   "RAVLT_perc_forgetting"
## [19] "LDELTOTAL"          "DIGITSCOR"          "TRABSCOR"
## [22] "FAQ"                "MOCA"               "EcogPtMem"
## [25] "EcogPtLang"         "EcogPtVisspat"      "EcogPtPlan"
## [28] "EcogPtOrgan"        "EcogPtDivatt"       "EcogPtTotal"
## [31] "EcogSPMem"          "EcogSPLang"         "EcogSPVisspat"
## [34] "EcogSPPlan"         "EcogSPOrgan"        "EcogSPDivatt"
## [37] "EcogSPTotal"        "Ventricles"         "Hippocampus"
## [40] "WholeBrain"         "Entorhinal"         "Fusiform"
## [43] "MidTemp"            "ICV"                "mPACCdigit"
## [46] "mPACCtrailsB"
```

```
sapply(baseline_sub2, is.factor) %>%
  which() %>%
  names()
```

```
## [1] "PTGENDER" "APOE4"    "DX"
```

```
baseline_sub2 %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```


ABETA
ADAS11
ADAS13
ADASQ4
AGE
AV45
CDRSB
DIGITSCOR
EcogPtDivatt
EcogPtLang
EcogPtMem
EcogPtOrgan
EcogPtPlan
EcogPtTotal
EcogPtVisspa
EcogSPDivatt
EcogSPLang
EcogSPMem
EcogSPOrgan
EcogSPPlan
EcogSPTotal
EcogSPVisspa
Entorhinal
FAQ
FBB
FDG
Fusiform
Hippocampus
ICV
LDELTOTAL
MidTemp
MMSE
MOCA
mPACCdigit
mPACCtrailsB
PIB
PTAU
PTEDUCAT
AVLT_forgettin
AVLT_immedia
AVLT_learnin
LT_perc_forge
TAU
TRABSCOR
Ventricles
WholeBrain
count
value

3 Missing values

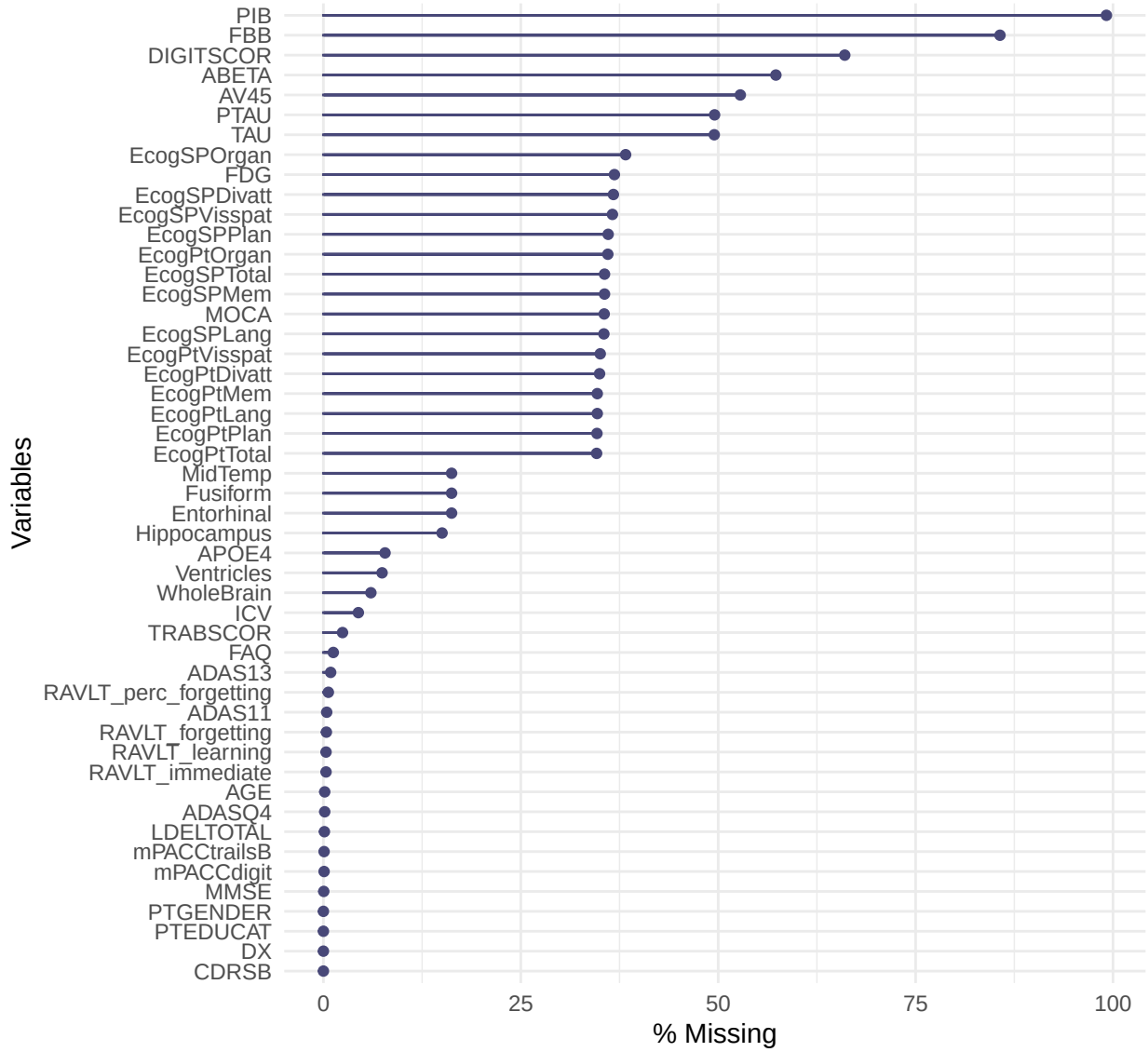
Data columns with too many missing values won't be of much value, therefore we have to check if there is any missing value in our dataset. So there are 2396 rows/observations which has missing values in at least one of the variables recorded, which is the entire population. We cannot delete all the observations with any missingness, by doing that, we will lose all of the data and the information in it.

```
mis_rows=which(complete.cases(baseline_sub2)==FALSE)
length(mis_rows)
```

```
## [1] 2396
```

The following figure shows the percentage of missingness for each of the remaining 49 variables, The final outcome DX doesn't have any missing observations. We will first exclude the three variables "PIB", "FBB" and "DIGITSCOR", which have more than 60% missing values.

```
naniar::gg_miss_var(baseline_sub2, show_pct = TRUE)
```



```
#Exclude variables "PIB""FBB""DIGITSCOR" due to their large missingness
baseline_sub2<-baseline_sub2%>% select (-c("PIB","FBB","DIGITSCOR"))
```

For the remaining 46 variables, we then imputed the missing values in the predictors using the missForest algorithm.

```
#missingness
set.seed(101)
data.imp.covariates <- missForest(xmis=baseline_sub2[,-which(names(baseline_sub2)%in%
                                                               c("DX"))], maxiter = 50)$ximp
# imputes the missing values in the covariates using random forest, COLPROT not important
data <- cbind(data.imp.covariates, DX=baseline_sub2[,which(names(baseline_sub2)%in%
                                                           c("DX"))]) # returns the imputed dataset
```

```
######################### Model building ##############################
data=data[!data$DX=="",]
str(data)
data$DX=as.factor(data$DX); levels(data$DX)=c("4","0","2","3","1")
```

```
table(data$DX)
```

```
##
##   4   0   2   3   1
## 411 541 420 683 341
```

```
data[,c("PTGENDER","PTEDUCAT","APOE4")] <- lapply(data[,c("PTGENDER","PTEDUCAT",
                                                          "APOE4")], factor)
```

4 Data Partitioning

Before applying any methods to chosen variables, dataset should be divided into two subsets: train and test. Training sample is used to train the model, while test sample is used for making the predictions and verify performance of the model. The entire (imputed) dataset will be splitting into a training and a test set with 70% and 30% observations respectively.

```
#split the data
set.seed(1)
index <- sample(2,nrow(data),replace = TRUE,prob=c(0.7,0.3))
datatrain <- data[index==1,]
datatest<- data[index==2,]
```

5 Classification method

We will use both Ordinal Forest method from the r package `ordinalForest` and classical random forest method from the r package `ranger`, comparing the performance of the two methods. The ordinal forest method is a random forest-based prediction method for ordinal response variables. Based on our understanding, classical random forest method is not appropriate for ordinal response data since it ignores the ordering in the levels and implements standard classification trees and will lead to loss information.

5.1 Ordinal forest

More information on the r package 'ordinalForest' can be find here: https://cran.r-project.org/web/packages/ordinalForest/ordinalForest.pdf

Train the data

Based on the Ordinal Forest method, the top 10 significant predictors are CDRSB(Clinical Dementia Rating), mPACCtrailsB(ADNI modified Preclinical Alzheimer's Cognitive Composite (PACC) with Trails B), mPACCdigit(ADNI modified Preclinical Alzheimer's Cognitive Composite (PACC) with Digit Symbol Substitution), MMSE(Mini Mental State Exam), LDELTOTAL(Logical Memory - Delayed Recall),EcogSPTotal(total everyday cognition test score), EcogSPPlan(everyday cognition test score upon plan function), EcogSPMem(everyday cognition test score upon memory function), FAQ(Functional Assessment Questionnaire)), and EcogSPVisspat(everyday cognition test score upon visual and spatial function).

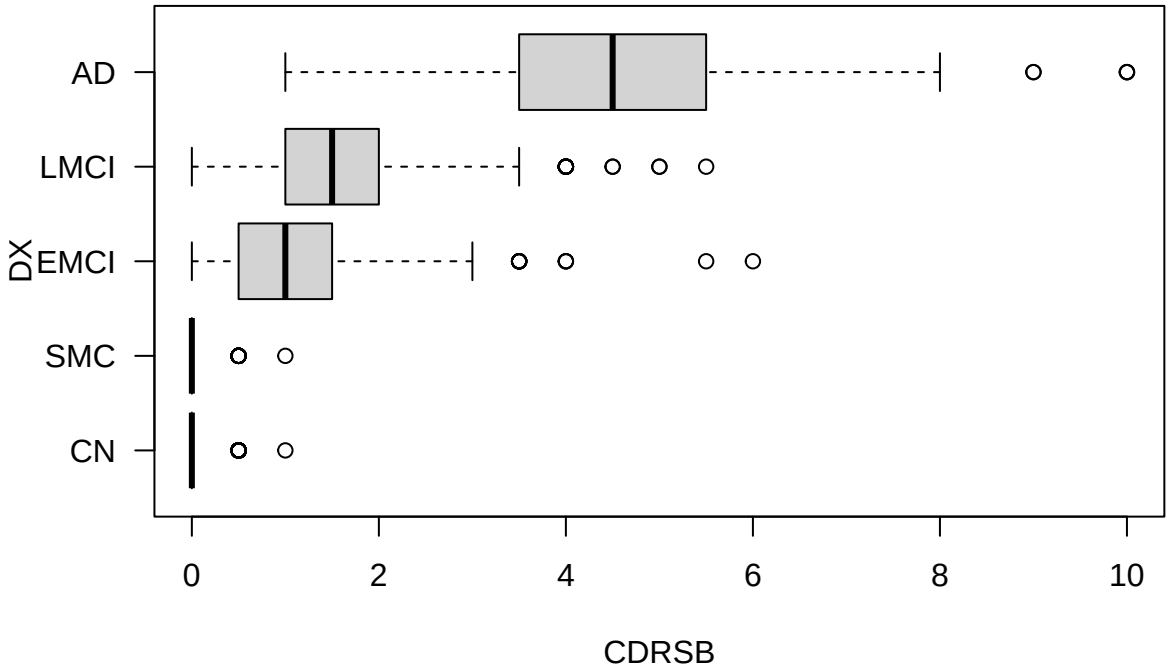
```
# Construct OF prediction rule using the training dataset:
set.seed(123)
ordforres <- ordfor(depvar="DX", data=datatrain, nsets=1000, ntreeperdiv=100,
                    ntreefinal=5000, perffunction = "equal")
set.seed(123)
ordforres1 <- ordfor(depvar="DX", data=datatrain, nsets=1000, ntreeperdiv=100,
                    ntreefinal=5000, perffunction="probability")

# Study variable importance values:
set.seed(123)
sort(ordforres$varimp, decreasing=TRUE)
```

```
##                CDRSB          mPACCtrailsB            mPACCdigit
##         5.600516e-02          2.876918e-02          2.671473e-02
##                MMSE             LDELTOTAL           EcogSPTotal
##         2.243521e-02          2.225760e-02          1.654158e-02
##           EcogSPPlan             EcogSPMem                   FAQ
##         1.627694e-02          1.413720e-02          1.096730e-02
##        EcogSPVisspat           EcogPtTotal            EcogSPLang
##         1.058262e-02          9.871796e-03          8.979732e-03
##          EcogSPOrgan          EcogSPDivatt                  MOCA
##         8.737588e-03          8.244359e-03          8.224939e-03
##             EcogPtMem                ADAS13            EcogPtPlan
##         8.036733e-03          7.742768e-03          7.232711e-03
##               ADAS11          EcogPtDivatt         EcogPtVisspat
##         6.569192e-03          5.934772e-03          4.495954e-03
##            EcogPtLang           EcogPtOrgan                   AGE
##         3.781506e-03          3.528268e-03          3.331169e-03
##                  FDG       RAVLT_immediate                  AV45
##         2.510927e-03          2.445517e-03          2.214542e-03
##               ADASQ4             TRABSCOR                 ABETA
##         2.153230e-03          2.105411e-03          1.731391e-03
##           Entorhinal RAVLT_perc_forgetting                  PTAU
##         1.431356e-03          1.408609e-03          1.192787e-03
##                  TAU           Hippocampus              Fusiform
##         1.112767e-03          9.711743e-04          9.558905e-04
##              MidTemp            WholeBrain        RAVLT_learning
##         7.447335e-04          6.744017e-04          5.483635e-04
```

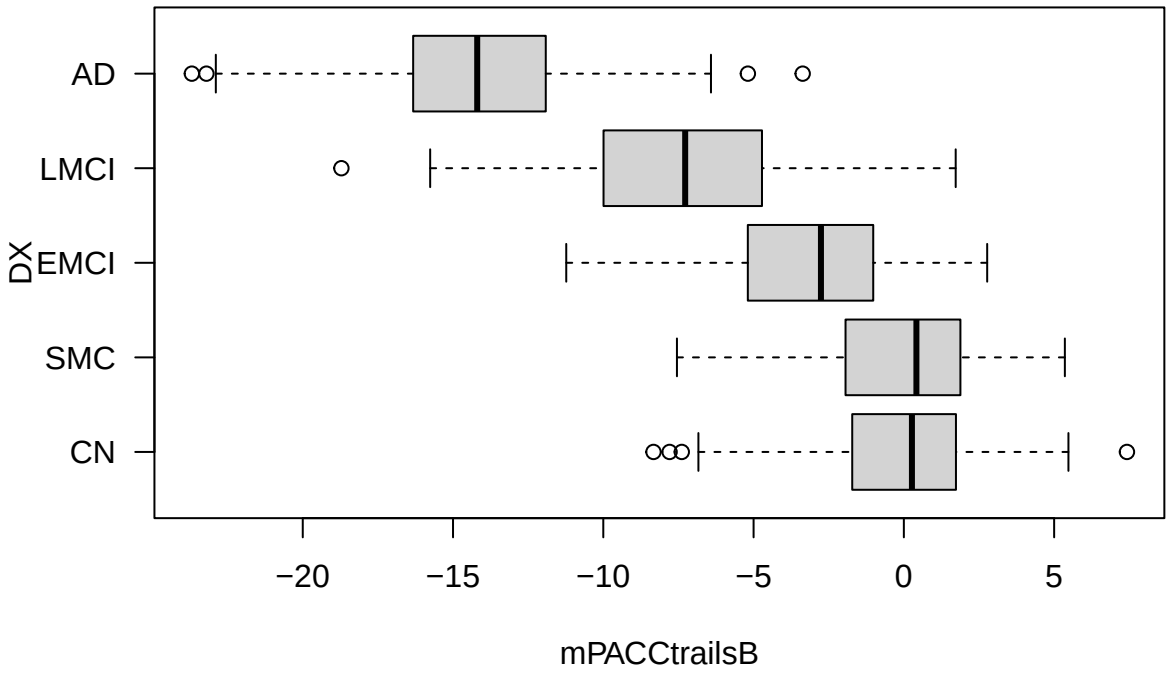
```
##              PTEDUCAT                   ICV           Ventricles
##          5.088253e-04          3.229987e-04          3.032697e-04
##              PTGENDER                  APOE4      RAVLT_forgetting
##          1.482653e-04          5.208829e-06         -4.642462e-05
```

The following three boxplots are the spreading of the top three most important predictors among the five extent of cognitive impairment outcome groups. Participants with Alzheimenr's disease overall has the highest average CDRSB, lowest mPACCtrailsB and lowest mPACCdigit.

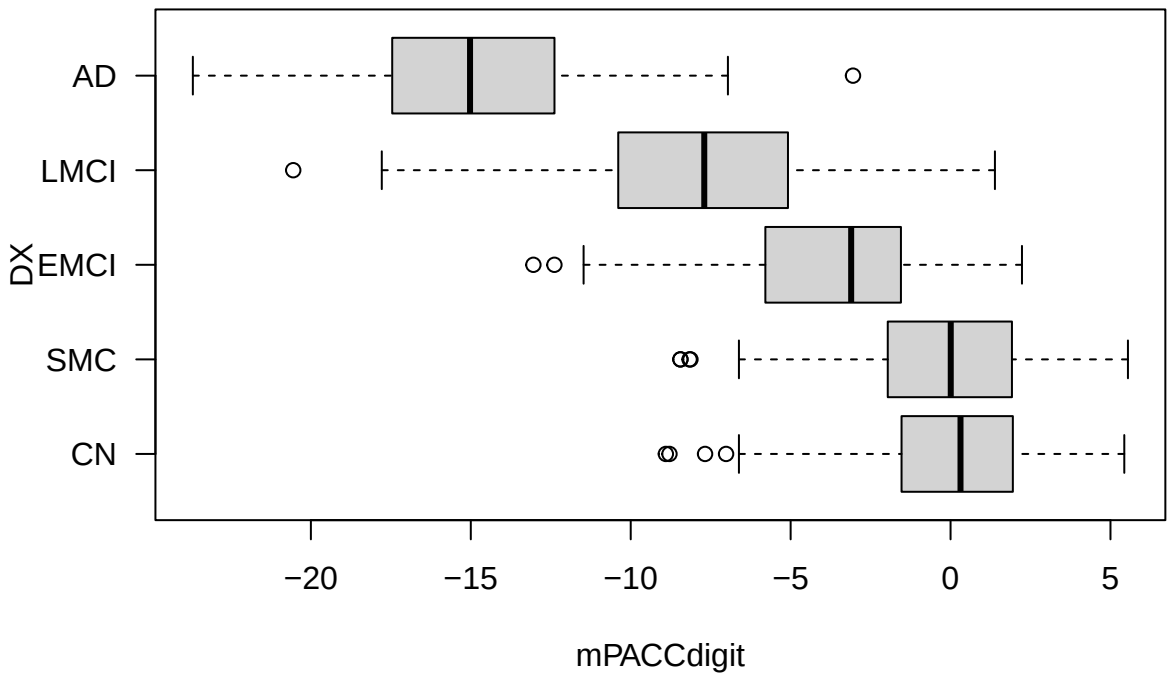
```
# Take a closer look at the top variables:
datatrain$DX <- factor(datatrain$DX, levels=c("0", "1", "2", "3", "4"))

boxplot(datatrain$CDRSB ~ datatrain$DX,ylab="DX",xlab="CDRSB", horizontal=TRUE,
        names=c("CN","SMC","EMCI","LMCI","AD"),las=1)
```




```
boxplot(datatrain$mPACCtrailsB ~ datatrain$DX, ylab="DX",xlab="mPACCtrailsB",
        horizontal=TRUE,names=c("CN","SMC","EMCI","LMCI","AD"),las=1)
```




```
boxplot(datatrain$mPACCdigit ~ datatrain$DX, ylab="DX",xlab="mPACCdigit",horizontal=TRUE,
        names=c("CN","SMC","EMCI","LMCI","AD"),las=1)
```



Test the prediction model

Evaluation of model accuracy

The model accuracy will be assessed through confusion matrix and missclassification rate.

```
# Compare predicted values with true values:

testDX = factor(datatest$DX,levels = c(0,1,2,3,4),labels = c("CN", " SMC",
                                                             "EMCI","LMCI", "AD"))

preds$ypred = factor(preds$ypred,levels = c(0,1,2,3,4),labels = c("CN", " SMC",
                                                             "EMCI","LMCI", "AD"))


tab<-table(data.frame(true_values=testDX, predictions=preds$ypred))
tab
```

```
##             predictions
## true_values  CN  SMC EMCI LMCI  AD
##         CN  104   44    3    1   0
##          SMC  30   55    9    1   0
##         EMCI   1    9  100   14   1
##         LMCI   2    0   35  162  24
##         AD     0    0    1   18 106
```

```
require(caret)
cm<-confusionMatrix(testDX, preds$ypred)
overall <- cm$overall
overall.accuracy <- overall['Accuracy']
1-sum(diag(tab))/sum(tab)
```

```
## [1] 0.2680556
```

```
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CN  SMC EMCI LMCI  AD
##       CN   104   44    3    1   0
##        SMC  30   55    9    1   0
##       EMCI   1    9  100   14   1
##       LMCI   2    0   35  162  24
##       AD     0    0    1   18 106
##
## Overall Statistics
##
##                Accuracy : 0.7319
##                  95% CI : (0.698, 0.764)
##     No Information Rate : 0.2722
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.66
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: CN Class:  SMC Class: EMCI Class: LMCI Class: AD
## Sensitivity             0.7591     0.50926      0.6757      0.8265    0.8092
## Specificity             0.9177     0.93464      0.9563      0.8836    0.9677
## Pos Pred Value          0.6842     0.57895      0.8000      0.7265    0.8480
## Neg Pred Value          0.9419     0.91520      0.9193      0.9316    0.9580
## Prevalence              0.1903     0.15000      0.2056      0.2722    0.1819
## Detection Rate          0.1444     0.07639      0.1389      0.2250    0.1472
## Detection Prevalence    0.2111     0.13194      0.1736      0.3097    0.1736
## Balanced Accuracy       0.8384     0.72195      0.8160      0.8551    0.8885
```

```
round(overall['Accuracy'],3)
```

```
## Accuracy
##    0.732
```

The prediction using test data showed that accuracy for Ordinal Forest is 0.732, and the missclassification rate is 0.268.

5.2 Random Forest - ranger

Train the data

```
set.seed(123)
rfr=ranger(DX ~ ., data = datatrain, importance = "permutation", classification=TRUE)
# Finding which predictors are significant
set.seed(123)
imprfr<-importance_pvalues(rfr, method = "altmann", formula = DX ~ ., data = datatrain)
imprfr
```

```
##                          importance     pvalue
## AGE                    4.914274e-03 0.00990099
## PTGENDER               2.464074e-04 0.09900990
## PTEDUCAT               6.189331e-03 0.00990099
## APOE4                  2.014499e-04 0.18811881
## FDG                    3.084396e-03 0.00990099
## AV45                   4.777441e-03 0.00990099
## ABETA                  3.384986e-03 0.00990099
## TAU                    1.892024e-03 0.00990099
## PTAU                   1.835195e-03 0.00990099
## CDRSB                  1.585409e-01 0.00990099
## ADAS11                 9.787070e-03 0.00990099
## ADAS13                 1.460881e-02 0.00990099
## ADASQ4                 4.336428e-03 0.00990099
## MMSE                   2.339796e-02 0.00990099
## RAVLT_immediate        6.053313e-03 0.00990099
## RAVLT_learning         1.388937e-03 0.00990099
## RAVLT_forgetting       6.570644e-05 0.45544554
## RAVLT_perc_forgetting 2.679575e-03 0.00990099
## LDELTOTAL              1.146057e-01 0.00990099
## TRABSCOR               2.832456e-03 0.00990099
## FAQ                    1.793036e-02 0.00990099
## MOCA                   1.589967e-02 0.00990099
## EcogPtMem              1.679218e-02 0.00990099
## EcogPtLang             9.981911e-03 0.00990099
## EcogPtVisspat          1.108411e-02 0.00990099
## EcogPtPlan             1.101919e-02 0.00990099
## EcogPtOrgan            6.918711e-03 0.00990099
## EcogPtDivatt           1.076496e-02 0.00990099
## EcogPtTotal            1.749832e-02 0.00990099
## EcogSPMem              3.173642e-02 0.00990099
## EcogSPLang             1.664019e-02 0.00990099
## EcogSPVisspat          1.520038e-02 0.00990099
## EcogSPPlan             2.496380e-02 0.00990099
## EcogSPOrgan            1.423962e-02 0.00990099
## EcogSPDivatt           1.239305e-02 0.00990099
## EcogSPTotal            2.751170e-02 0.00990099
## Ventricles             7.300047e-04 0.10891089
## Hippocampus            1.869737e-03 0.00990099
## WholeBrain             9.711135e-04 0.05940594
## Entorhinal             1.980172e-03 0.00990099
## Fusiform               1.937130e-03 0.00990099
```

```
## MidTemp               1.104452e-03 0.02970297
## ICV                   1.127036e-03 0.02970297
## mPACCdigit            4.554919e-02 0.00990099
## mPACCtrailsB          4.734455e-02 0.00990099
```

The top 10 important predictors by using random forest- ranger() are CDRSB(Clinical Dementia Rating), LDELTOTAL(Logical Memory - Delayed Recall), mPACCtrailsB(ADNI modified Preclinical Alzheimer's Cognitive Composite (PACC) with Trails B), mPACCdigit(ADNI modified Preclinical Alzheimer's Cognitive Composite (PACC) with Digit Symbol Substitution), EcogSPMem(everyday cognition test score upon memory function), EcogSPTotal(total everyday cognition test score), EcogSPPlan(everyday cognition test score upon plan function), MMSE(Mini Mental State Exam), FAQ(Functional Assessment Questionnaire), and EcogPtTotal(total everyday cognition test score). Within the 45 predictors, there are 5 predictors have p-value less than 0.05 for the variable importance: PTGENDER (p=0.099) APOE4 (p-value=0.188), RAVLT_forgetting (p-value=0.455), Ventricles (p-value=0.109), WholeBrain (p-value= 0.059), so we conclude that these variables are not significant predictors for determining the patient's extent of cognitive impairment.

Test the prediction model

```
# Predicting the responses in the test set and obtaining the misclassification
# error rate
pred_rfr0=predict(rfr, data=as.data.frame(datatest[,-46]), type="response")

pred_rfr=pred_rfr0$predictions
```

Evaluation of model accuracy

```
pred_rfr = factor(pred_rfr,levels = c(0,1,2,3,4),labels = c("CN", " SMC","EMCI",
                                                            "LMCI", "AD"))

tabr<-table(testDX,pred_rfr)

1-sum(diag(tabr))/sum(tabr)
```

```
## [1] 0.2263889
```

```
cm2<-confusionMatrix(testDX, pred_rfr)
overall2 <- cm2$overall
overall2.accuracy <- overall2['Accuracy']
1-sum(diag(tabr))/sum(tabr)
```

```
## [1] 0.2263889
```

```
cm2
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction  CN  SMC EMCI LMCI  AD
##       CN   114   35    2    1   0
##        SMC  42   44    7    2   0
##       EMCI   0    1  103   20   1
##       LMCI   0    0   10  192  21
##       AD     0    0    0   21 104
##
## Overall Statistics
##
##                Accuracy : 0.7736
##                  95% CI : (0.7413, 0.8037)
##     No Information Rate : 0.3278
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7091
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: CN Class:  SMC Class: EMCI Class: LMCI Class: AD
## Sensitivity             0.7308     0.55000      0.8443      0.8136    0.8254
## Specificity             0.9326     0.92031      0.9632      0.9360    0.9646
## Pos Pred Value          0.7500     0.46316      0.8240      0.8610    0.8320
## Neg Pred Value          0.9261     0.94240      0.9681      0.9115    0.9630
## Prevalence              0.2167     0.11111      0.1694      0.3278    0.1750
## Detection Rate          0.1583     0.06111      0.1431      0.2667    0.1444
## Detection Prevalence    0.2111     0.13194      0.1736      0.3097    0.1736
## Balanced Accuracy       0.8317     0.73516      0.9037      0.8748    0.8950
```

```
round(overall2['Accuracy'],3)
```

```
## Accuracy
##    0.774
```

The prediction using test data showed that accuracy for Random Forest using `ranger()` is 0.774, and the missclassification rate is 0.226.

Conclusions

Both models have CDRSB, mPACCtrailsB, mPACCdigit, MMSE, LDELTOTAL, EcogSPTotal,EcogSPPlan, EcogSPMem, FAQ as their top ten significant predictors. To our surprise, the top 10 significant predictors for both models are all important cognitive test scores. The widely accepted AD biomarkers like ABETA, TAU and PTAU are not among them. In summary, our two models have similar prediction accuracy. However, it was surprised to us that Random Forest using `ranger()` performed better with 0.774 accuracy and 0.226missclassification rate, than Ordinal Forest with0.732 accuracy and 0.268 missclassification rate.