Brexit Indicative Votes Modelling

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.1.1     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.2.1     v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(maptools)
```

```
## Warning: package 'maptools' was built under R version 3.5.3
```

```
## Loading required package: sp
```

```
## Checking rgeos availability: TRUE
```

```
library(rgdal)
```

```
## Warning: package 'rgdal' was built under R version 3.5.3
```

```
## rgdal: version: 1.4-3, (SVN revision 828)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
##  Path to GDAL shared files: C:/Users/jackb/Documents/R/win-library/3.5/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Users/jackb/Documents/R/win-library/3.5/rgdal/proj
##  Linking to sp version: 1.3-1
```

```
library(rgeos)
```

```
## Warning: package 'rgeos' was built under R version 3.5.3
```

```
## rgeos version: 0.4-2, (SVN revision 581)
##  GEOS runtime version: 3.6.1-CAPI-1.10.1
##  Linking to sp version: 1.3-1
##  Polygon checking: TRUE
```

```
library(ggplot2)
library(mapproj)
```

```
## Warning: package 'mapproj' was built under R version 3.5.3
```

```
## Loading required package: maps
```

```
## Warning: package 'maps' was built under R version 3.5.3
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(nnet)
library(kernlab)
```

```
## Warning: package 'kernlab' was built under R version 3.5.2
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.5.3
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.5.3
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 3.5.3
```

```
## Loading required package: parallel
```

```
library(foreach)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 3.5.3
##
## Attaching package: 'xgboost'
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.5.3
##
## Attaching package: 'skimr'
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ada)
```

```
## Warning: package 'ada' was built under R version 3.5.3
## Loading required package: rpart
## Warning: package 'rpart' was built under R version 3.5.3
```

```
library(adabag)
```

```
## Warning: package 'adabag' was built under R version 3.5.3
```

```
library(fastAdaboost)
```

```
## Warning: package 'fastAdaboost' was built under R version 3.5.3
```

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.5.3
## Loaded gbm 2.1.5
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.5.3
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(naivebayes)
```

```
## Warning: package 'naivebayes' was built under R version 3.5.3
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.5.3
```

```
library(klaR)
```

```
## Warning: package 'klaR' was built under R version 3.5.3
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(rpart)
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.5.3
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

```
## The following object is masked from 'package:xgboost':
##
##     xgboost
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(RColorBrewer)
```

```
## Warning: package 'RColorBrewer' was built under R version 3.5.2
```

```
source("final/final_code_functions.R")
```

```
#### Importing data ####
mpdata <- read_csv("data/mpdataFinal.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   BetterLeavePercent = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

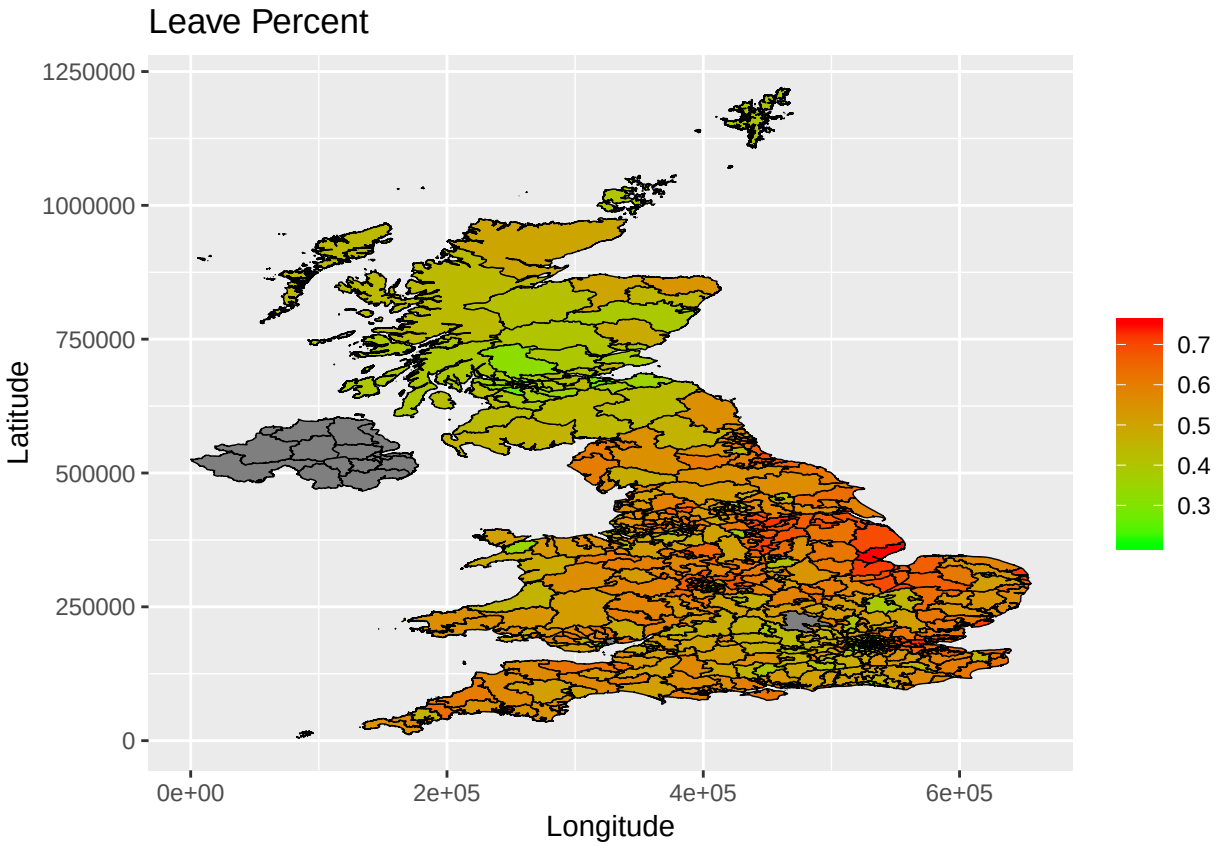
```
# Exploratory ------------------------------------------------------------
#### Choropleths ####
ukMap <- readOGR('Shapefiles/Westminster_Parliamentary_Constituencies_December_2017_Generalised_Clipped
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\jackb\OneDrive - University of St Andrews\Everything\Uni\Data Intensive Analysis\I
## with 650 features
## It has 9 fields
## Integer64 fields read as strings:  bng_e bng_n objectid
```

```
ukMapf <- fortify(ukMap, region = "pcon17nm") %>% rename(Geography = id) %>% arrange(Geography)
mpdata$Geography <- mpdata$Constituency

# Fixing geographies
ukMapf$Geography <- gsub('[[:punct:] ]+',' ',ukMapf$Geography)
mpdata$Geography <- replace(mpdata$Geography, mpdata$Name=="Albert Owen", "Ynys Mon")
geos.map <- sort(unique(ukMapf$Geography))
geos.map <- geos.map[!geos.map == "Newport West"]
geos.map <- geos.map[!geos.map == "Buckingham"]
mpdata <- mpdata[order(mpdata$Geography),]
mpdata$Geography <- geos.map

leavePercentMap <- UKChoroplethFunction(ukMapf, mpdata, 'BetterLeavePercent', "Leave Percent", "Leave P
leavePercentMap <- leavePercentMap + scale_fill_gradient(aes(fill = ""), low = 'green', high = 'red')
```

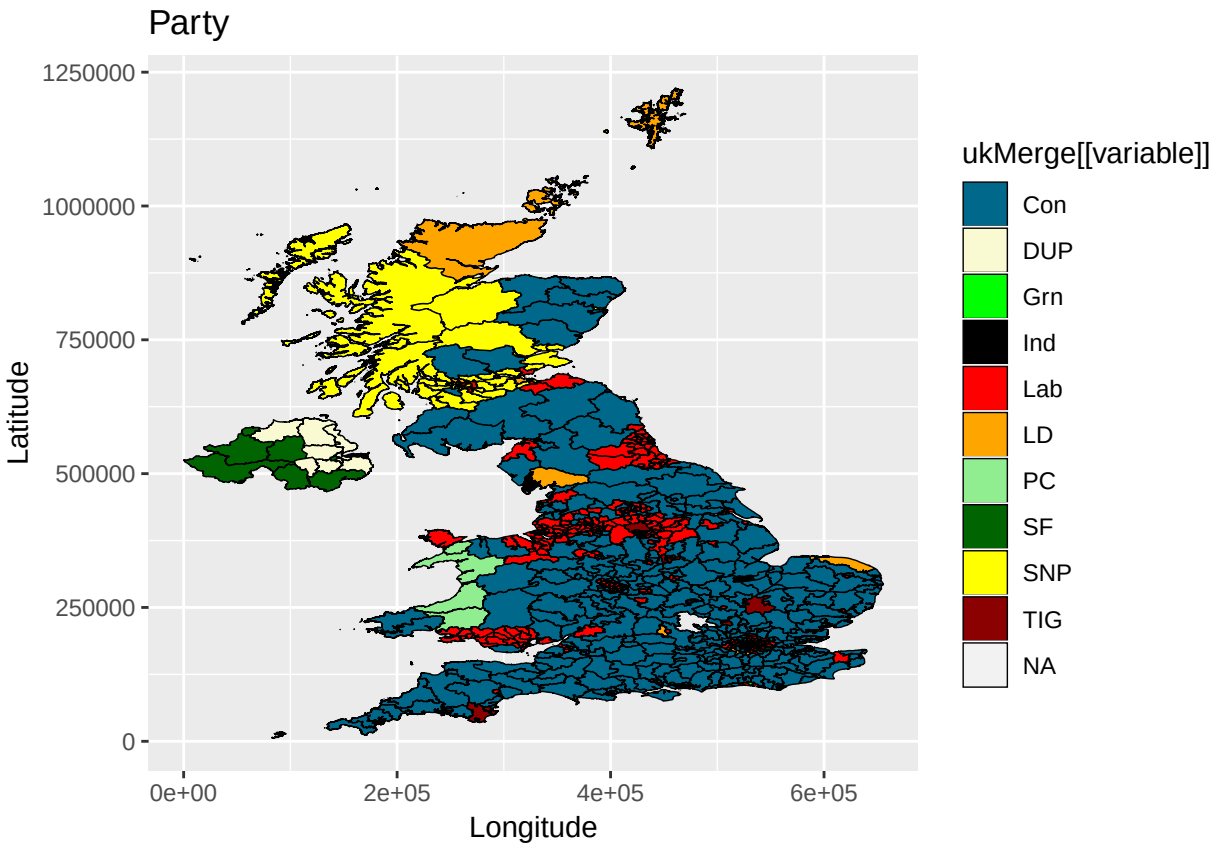
```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

```
leavePercentMap
```




```
partyMap <- UKChoroplethFunction(ukMapf, mpdata, 'Party', "Party", "Party", FALSE)
party.color.scheme <- c("deepskyblue4", "lightgoldenrodyellow", "Green", "Black", "Red", "Orange", "Lig
```

```
partyMap <- partyMap + scale_fill_manual(values=party.color.scheme)
partyMap
```



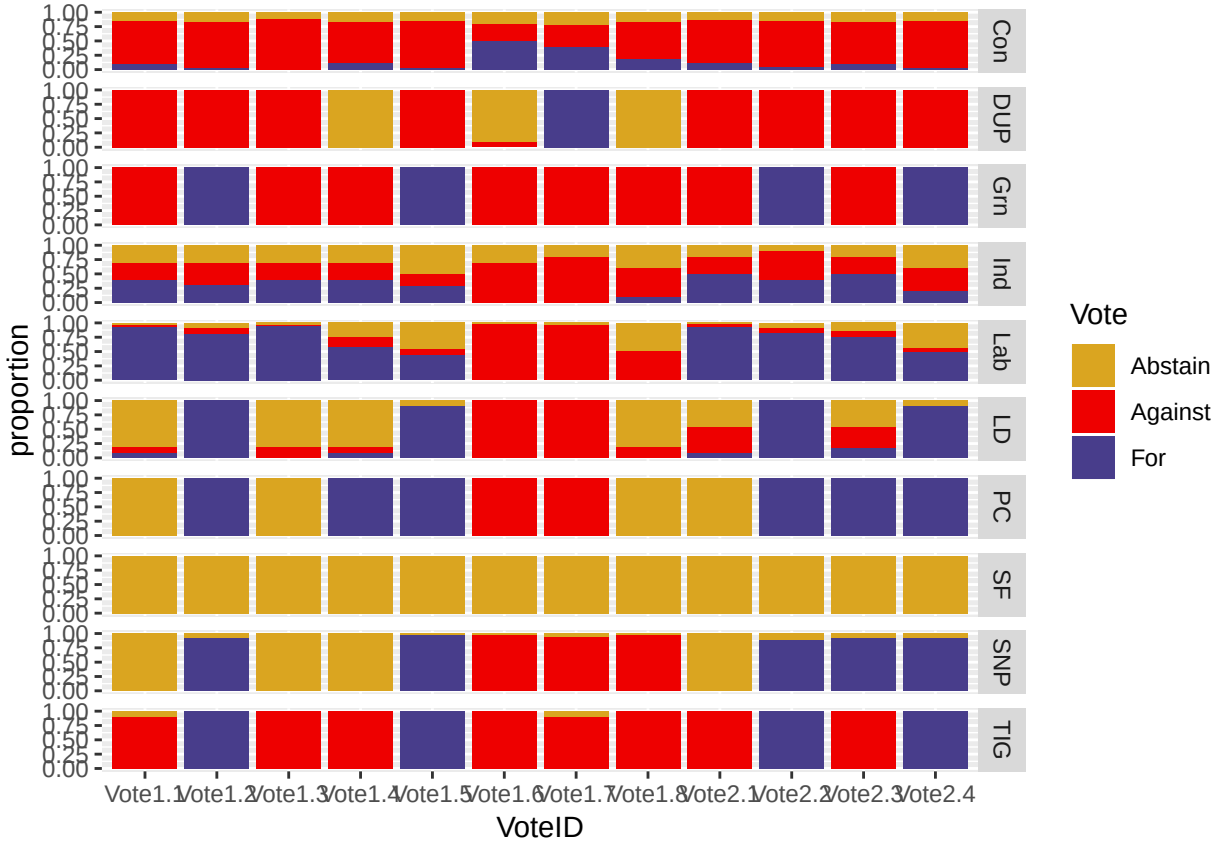

```
#### Graphs ####
# All Parties
mpdata <- read_csv("data/mpdataFinal.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   BetterLeavePercent = col_double()
## )
## See spec(...) for full column specifications.
```

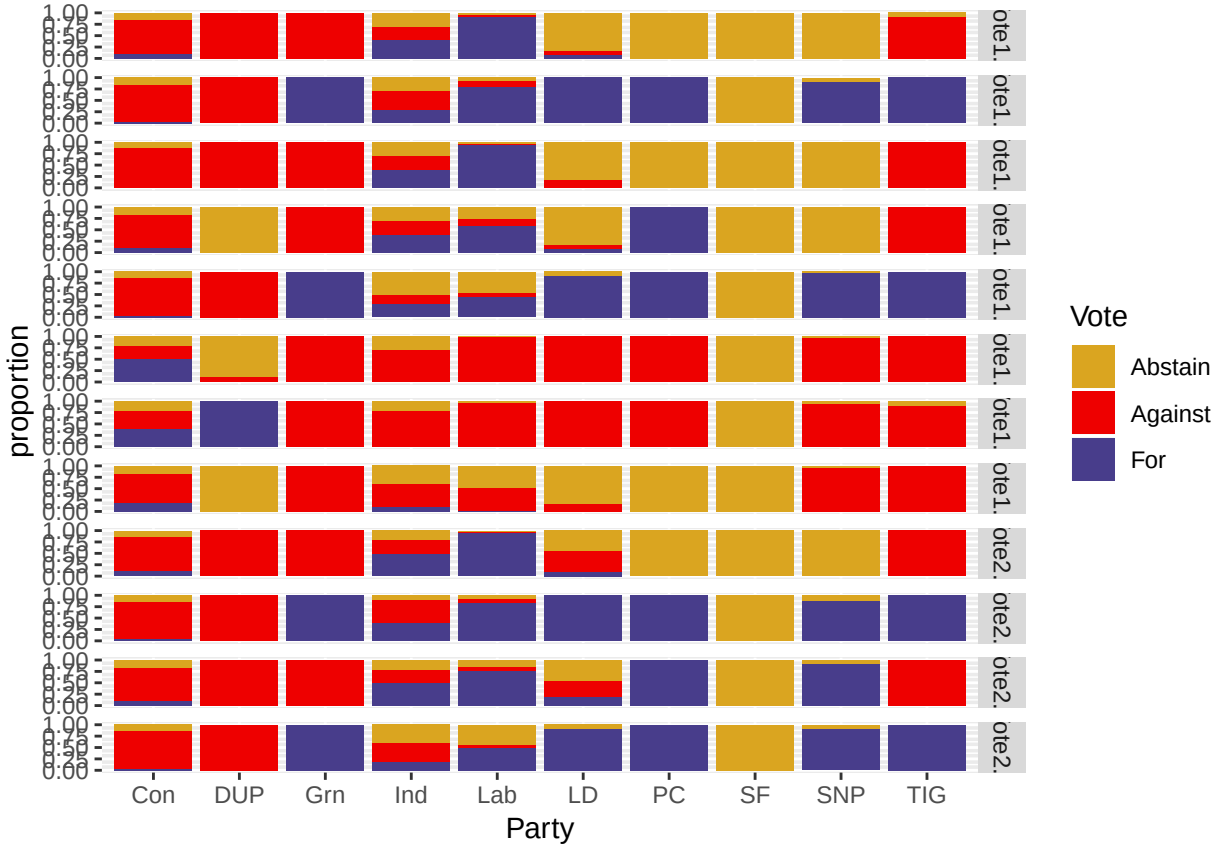
```
mpdata.gathered <- mpdata %>% gather("VoteID", "Vote", -Party, -PartyGroup, -BetterLeavePercent, -Groupe
mpdata.gathered$VoteID <- factor(mpdata.gathered$VoteID, levels=c("Vote1", "Vote2", "Vote3", "Vote4", "V
levels(mpdata.gathered$VoteID) <- c("Vote1.1", "Vote1.2", "Vote1.3", "Vote1.4", "Vote1.5", "Vote1.6", "V

party.nums <- data.frame(mpdata %>% group_by(Party) %>% summarise(mps = n()))
rownames(party.nums) <- party.nums$Party
summ.party <- data.frame(mpdata.gathered %>% group_by(Party, VoteID, Vote) %>%
                           summarise(count = n(), meanleave = mean(BetterLeavePercent))) %>%
                           arrange(VoteID)
summ.party$proportion <- NA
for (i in 1:nrow(summ.party)) {
  summ.party$proportion[i] <- summ.party[i, 'count']/party.nums[summ.party[i,'Party'],2]
}
```

```
# summ.party$proportion <- apply(summ.party, 1, function(x) return(x[['count']]/party.nums[x[['Party']]]
```

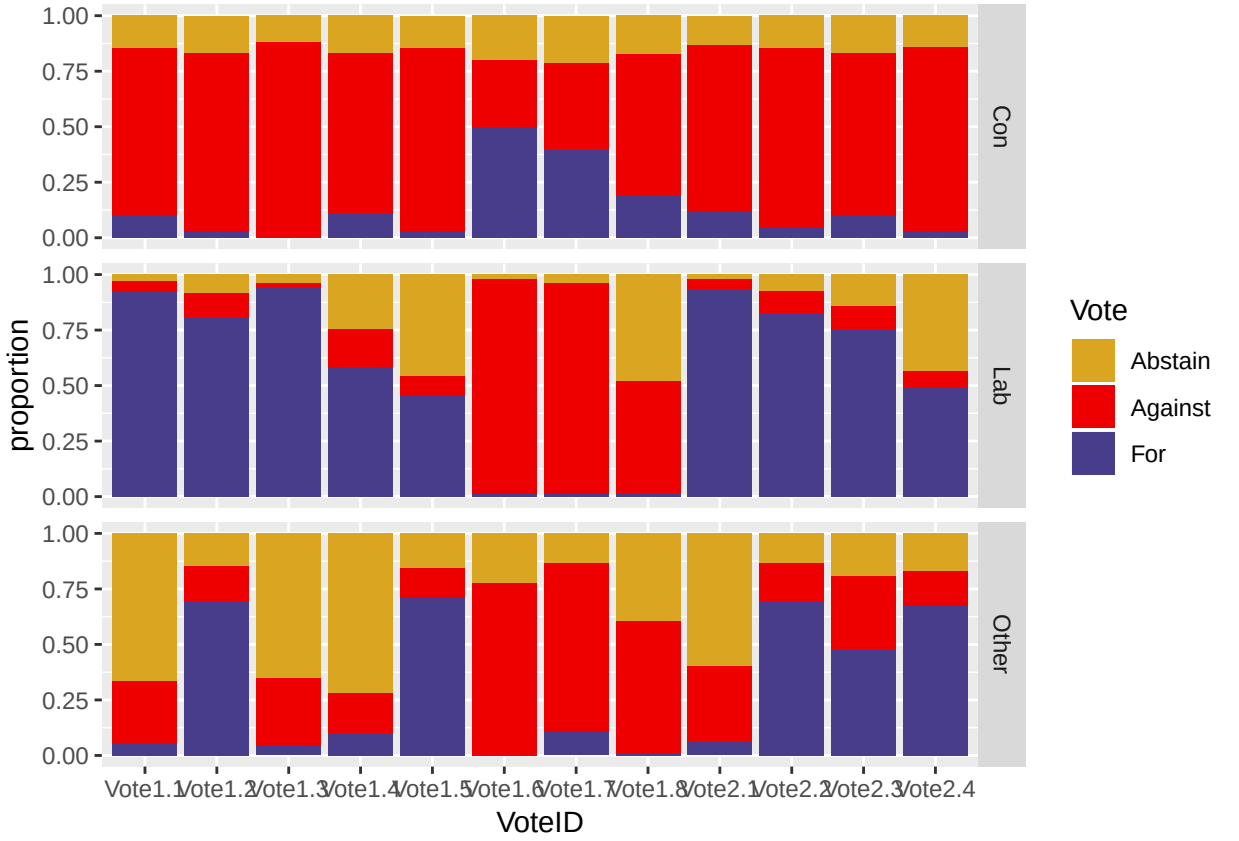
```
all.parties.plot <- ggplot(summ.party, aes(x=VoteID, y=proportion, fill=Vote)) +
  geom_bar(stat='identity', position='stack') +
  scale_fill_manual(values=c("goldenrod", "red2", "darkslateblue")) +
  facet_grid(rows=vars(Party))
all.parties.plot
```



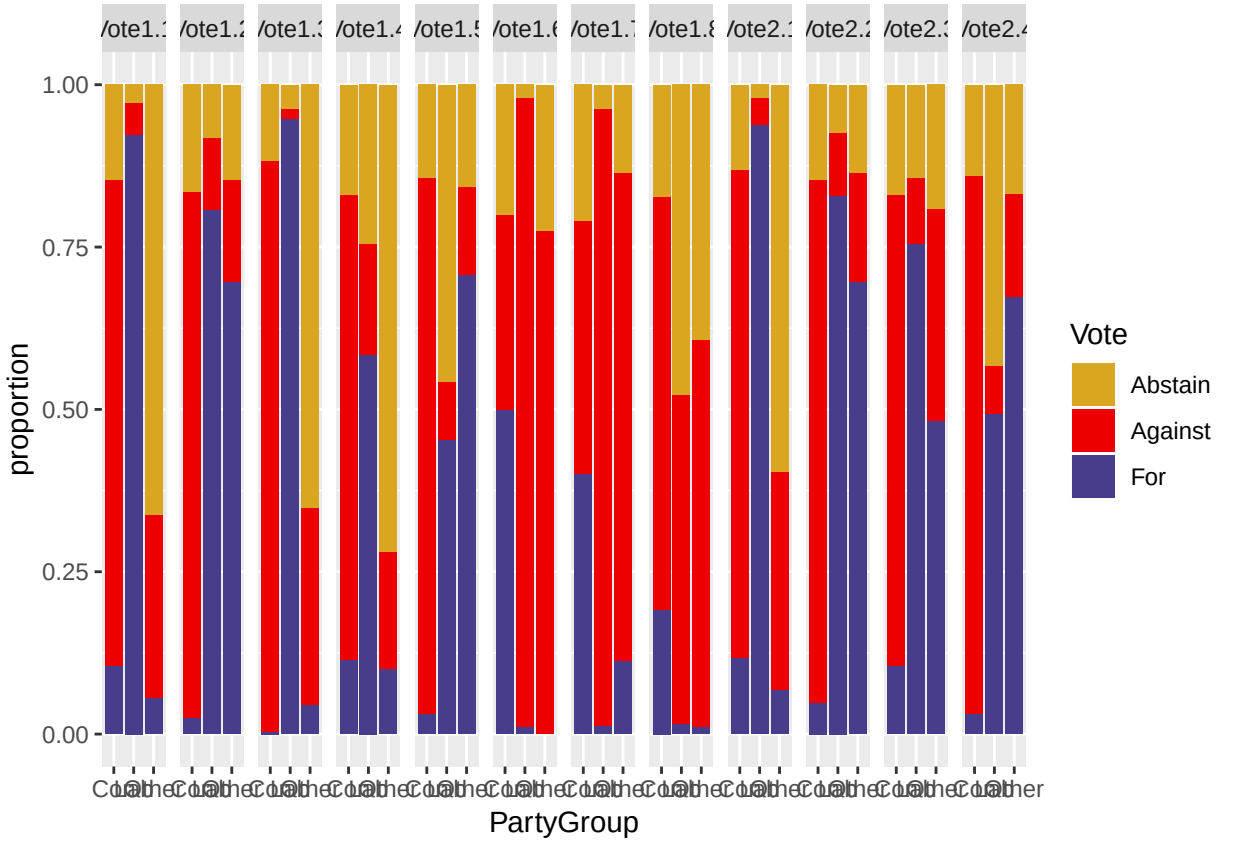

```
all.parties.vote.plot <- ggplot(summ.party, aes(x=Party, y=proportion, fill=Vote)) +
  geom_bar(stat='identity', position='stack') +
  scale_fill_manual(values=c("goldenrod", "red2", "darkslateblue")) +
  facet_grid(rows=vars(VoteID))
all.parties.vote.plot
```




```
# Party Group
party.nums <- data.frame(mpdata.gathered %>% group_by(PartyGroup) %>% summarise(mps = n()/12))
rownames(party.nums) <- party.nums$PartyGroup
summ.party <- data.frame(mpdata.gathered %>% group_by(PartyGroup, VoteID, Vote) %>% summarise(count = n
summ.party$proportion <- NA
for (i in 1:nrow(summ.party)) {
  summ.party$proportion[i] <- summ.party[i, 'count']/party.nums[summ.party[i,'PartyGroup'],2]
}
# summ.party$proportion <- apply(summ.party, 1, function(x) return(x[['count']]/party.nums[x[['Party']]

parties.group.plot <- ggplot(summ.party, aes(x=VoteID, y=proportion, fill=Vote)) +
  geom_bar(stat='identity', position='stack') +
  scale_fill_manual(values=c("goldenrod", "red2", "darkslateblue")) +
  facet_grid(rows=vars(PartyGroup))
parties.group.plot
```



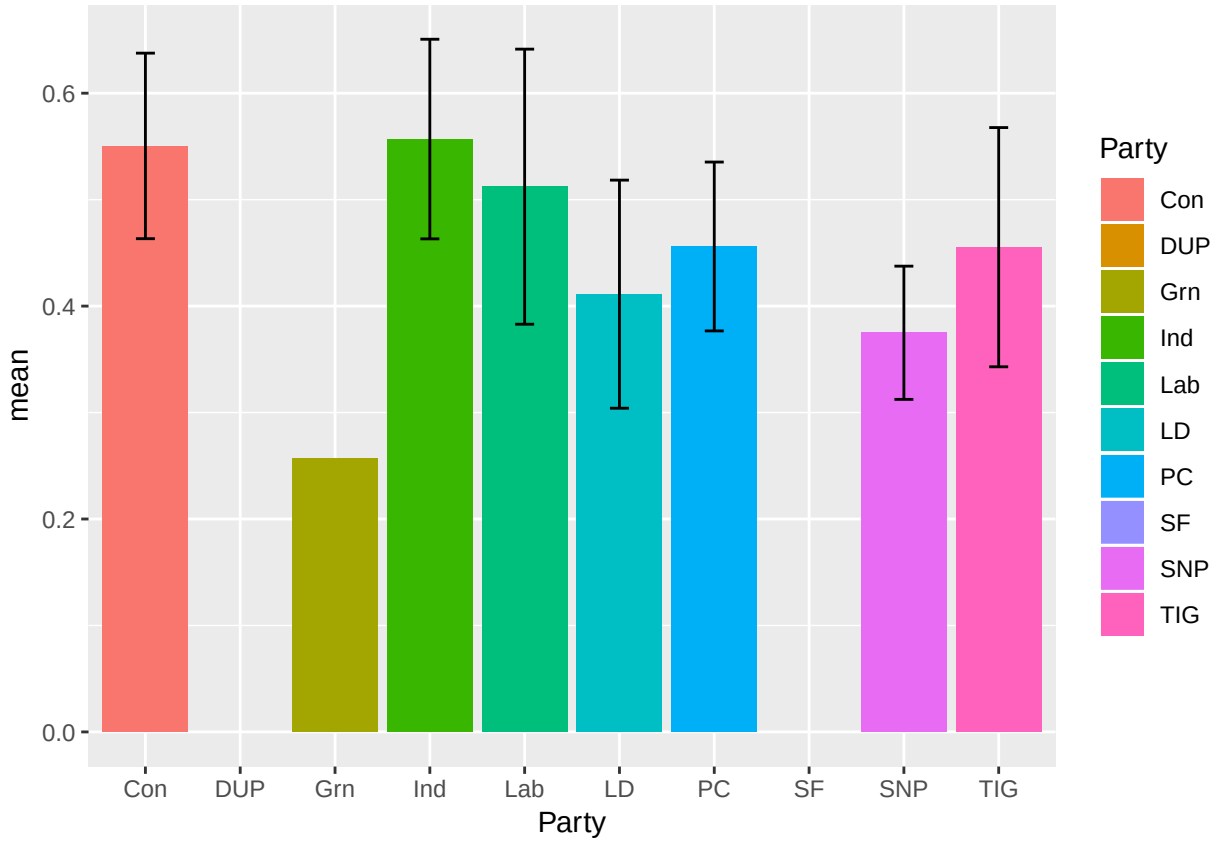

```
all.parties.group.vote.plot <- ggplot(summ.party, aes(x=PartyGroup, y=proportion, fill=Vote)) +
  geom_bar(stat='identity', position='stack') +
  scale_fill_manual(values=c("goldenrod", "red2", "darkslateblue")) +
  facet_grid(cols=vars(VoteID))
all.parties.group.vote.plot
```




```
# Party Leave Percents
parties.leave <- mpdata %>% group_by(Party) %>% summarise(mean = mean(BetterLeavePercent, na.rm = TRUE)
ggplot(parties.leave, aes(x=Party, y=mean, fill=Party)) + geom_col() +
  geom_errorbar(aes(ymin=mean-sd, ymax=mean+sd), width=0.2)
```

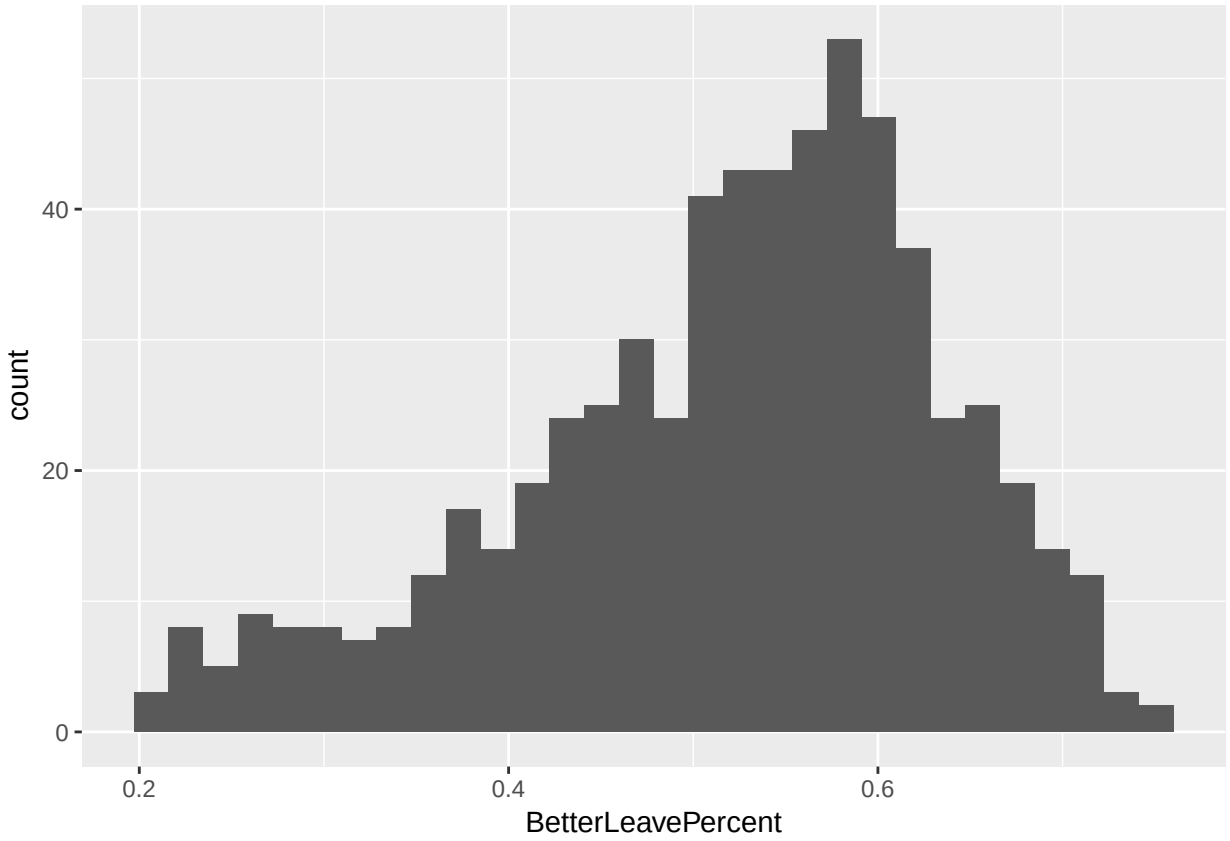
```
## Warning: Removed 2 rows containing missing values (position_stack).
```

```
## Warning: Removed 3 rows containing missing values (geom_errorbar).
```




```
ggplot(mpdata, x=BetterLeavePercent) + geom_histogram(aes(x=BetterLeavePercent))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 18 rows containing non-finite values (stat_bin).
```




```
#### Descriptive stats ####
desc.data <- dplyr::select(mpdata, -Name, -Constituency)
skimmed.data <- skim_to_wide(desc.data)
View(skimmed.data[, c(1:10, 16)])
```

```
#### Setting up data ####
# Import data
data <- mpdata
data <- as.data.frame(unclass(data))
data$GroupedPercent <- as.factor(data$GroupedPercent)

# Generate holdout split
set.seed(5059)
train.indices <- createDataPartition(data$Party, p=0.8, list=FALSE)
```

```
## Warning in createDataPartition(data$Party, p = 0.8, list = FALSE): Some
## classes have a single record ( Grn ) and these will be selected for the
## sample
```

```
train <- data[train.indices,]
holdout <- data[-train.indices,]
train.no.NA <- na.omit(train)
holdout.no.NA <- na.omit(holdout)

# Generate dummy data
matrix <- dummyVars(~ Vote1 + Vote2 + Vote3 + Vote4 + Vote5 + Vote6 + Vote7 + Vote8 +
                      SecondVote1 + SecondVote2 + SecondVote3 + SecondVote4,
```

```
                    data = data)
dummyData <- data.frame(predict(matrix,newdata=data))
dummyData <- cbind(data[c(1,2,3,4,5,6)],dummyData)
covars <- paste(c(colnames(dummyData[-c(1,2,3,4,5,6)])),collapse="+")
train.dummy <- dummyData[train.indices,]
train.dummy.no.NA <- na.omit(train.dummy)
holdout.dummy <- dummyData[-train.indices,]
holdout.dummy.no.NA <- na.omit(holdout.dummy)

# Generate grid for NN and SVMs
nn.grid <- expand.grid(.decay = c(0.5,0.1), .size = c(2,4,8,12))
svm.grid <- expand.grid(C = c(0,0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2,5))

# Setting up parralellisation
cores <- makeCluster(detectCores()-2)
registerDoParallel(cores = cores)

# Covars for binary tree
bin.covars <- colnames(train)[7:18]

# SEE APPENDIX FOR MODELLING CODE
load("data/saved_models.RData")
```

```
# Model Analysis ------------------------------------------------------------
# Load workspace to save time - load("data/saved_models.RData")
# All classification - ROC curves, confusion matrices, varImp
# Cool plots to do with final model:
#   Choropleth based on predictions of best model, suggests a changing environment
#   Most important votes, and for what parties
#     Keep in mind 'importance' only means 'more different that other parties'
#     This could be a good predictor for how Brexit could 'resolve'
#       Whichever votes had the least 'importance' could be more likely to be compromises/have defector
# all.class.models <- list("Party" = party.models,"PartyGroup" = party.grp.models,
                         #  "ConstituencyVote" = ref.vote.models, "GroupedPercent" = leave.quant.models,
                         # "GroupedPercent2" = leave.disc.models)
# all.continuous.models <- list("NN"= leave.percent.models$NN, "lm" = leave.percent.models$lm,
                          #     "GBM"=leave.percent.models$GBM)
accuracy <- matrix(ncol=10, nrow=5)
colnames(accuracy) <- c("NN", "SVM", "Tree", "RF", "XGB", "GBM", "GLM", "NB", "ada", "adaboost")
rownames(accuracy) <- c("Party", "PartyGroup", "VoteResult", "LeaveQuantile", "LeaveDiscrete")

kappa <- matrix(ncol=10, nrow=5)
colnames(kappa) <- c("NN", "SVM", "Tree", "RF", "XGB", "GBM", "GLM", "NB", "ada", "adaboost")
rownames(kappa) <- c("Party", "PartyGroup", "VoteResult", "LeaveQuantile", "LeaveDiscrete")

best.kappa <- matrix(ncol=10, nrow=5)
colnames(best.kappa) <- c("NN", "SVM", "Tree", "RF", "XGB", "GBM", "GLM", "NB", "ada", "adaboost")
rownames(best.kappa) <- c("Party", "PartyGroup", "VoteResult", "LeaveQuantile", "LeaveDiscrete")

auc <- matrix(ncol=10, nrow=5)
colnames(auc) <- c("NN", "SVM", "Tree", "RF", "XGB", "GBM", "GLM", "NB", "ada", "adaboost")
rownames(auc) <- c("Party", "PartyGroup", "VoteResult", "LeaveQuantile", "LeaveDiscrete")

for (i in 1:length(all.class.models)) {
```

```
  for (j in 1:length(all.class.models[[i]])) {
    response.type <- names(all.class.models)[i]
    model.type <- names(all.class.models[[i]])[j]
    model <- all.class.models[[i]][[j]]

    if (model.type == "NN" | model.type == "SVM") {
      if (response.type != "Party" & response.type != "PartyGroup") {
        test.data <- holdout.dummy.no.NA[,7:42]
        response <- holdout.dummy.no.NA[,response.type]
      } else {
        test.data <- holdout.dummy[,7:42]
        response <- holdout.dummy[,response.type]
      }
    } else {
      if (response.type != "Party" & response.type != "PartyGroup") {
        test.data <- holdout[,7:18]
        response <- holdout[,response.type]
      } else {
        test.data <- holdout.no.NA[,7:18]
        response <- holdout.no.NA[,response.type]
      }
    }

    accuracy[i, model.type] <- max(na.omit(model$results[,"Accuracy"]))
    kappa[i, model.type] <-
      na.omit(model$results[,"Kappa"])[which.max(na.omit(model$results[,"Accuracy"]))]

    best.kappa[i, model.type] <- max(na.omit(model$results[,"Kappa"]))


    if (model.type != "SVM") {
      preds.probs <- predict(model, test.data, type="prob")
      auc[i, model.type] <- multiclass.roc(response, preds.probs)$auc[1]
    }

    analyse.model(model, test.data, response, response.type, model.type)

  }
}
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): Grn, PC
```

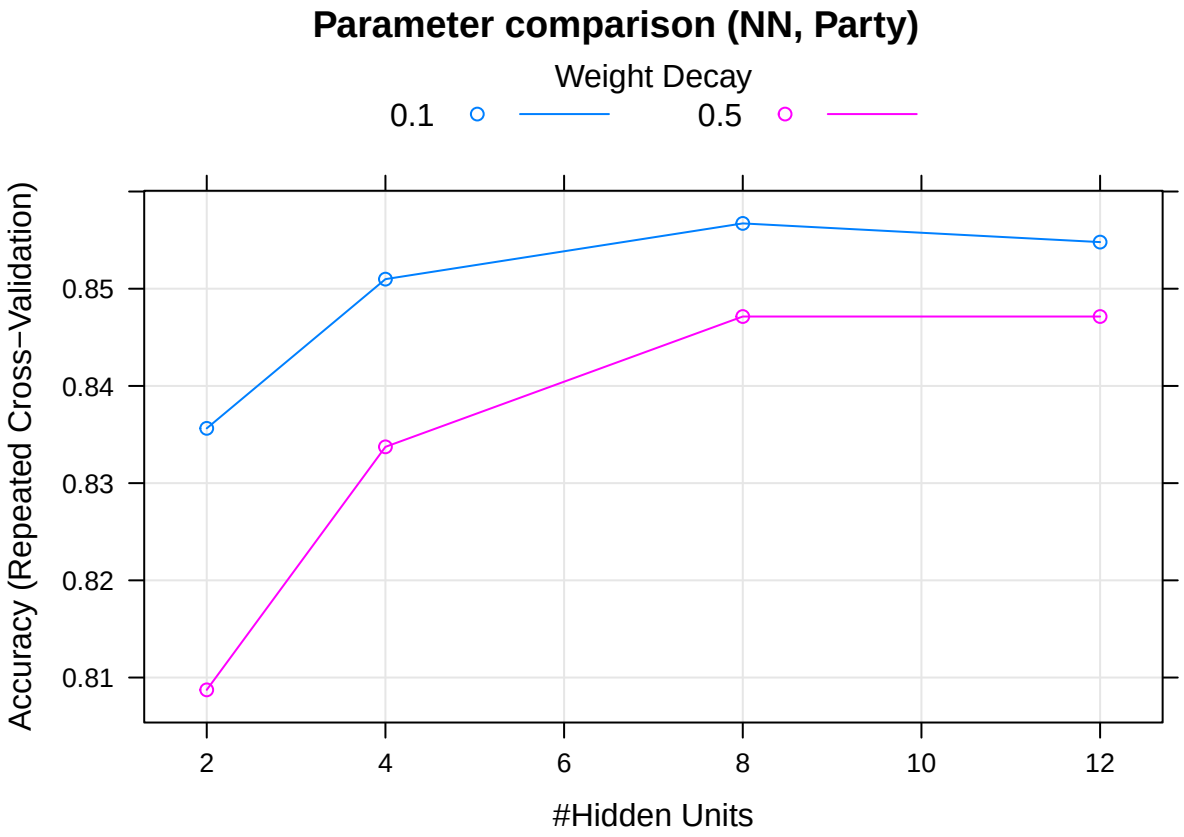
```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response': Grn,PC.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  62   0   0   1   1  0  0  1   0   0
##        DUP   0   2   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  47  1  0  0   0   0
```

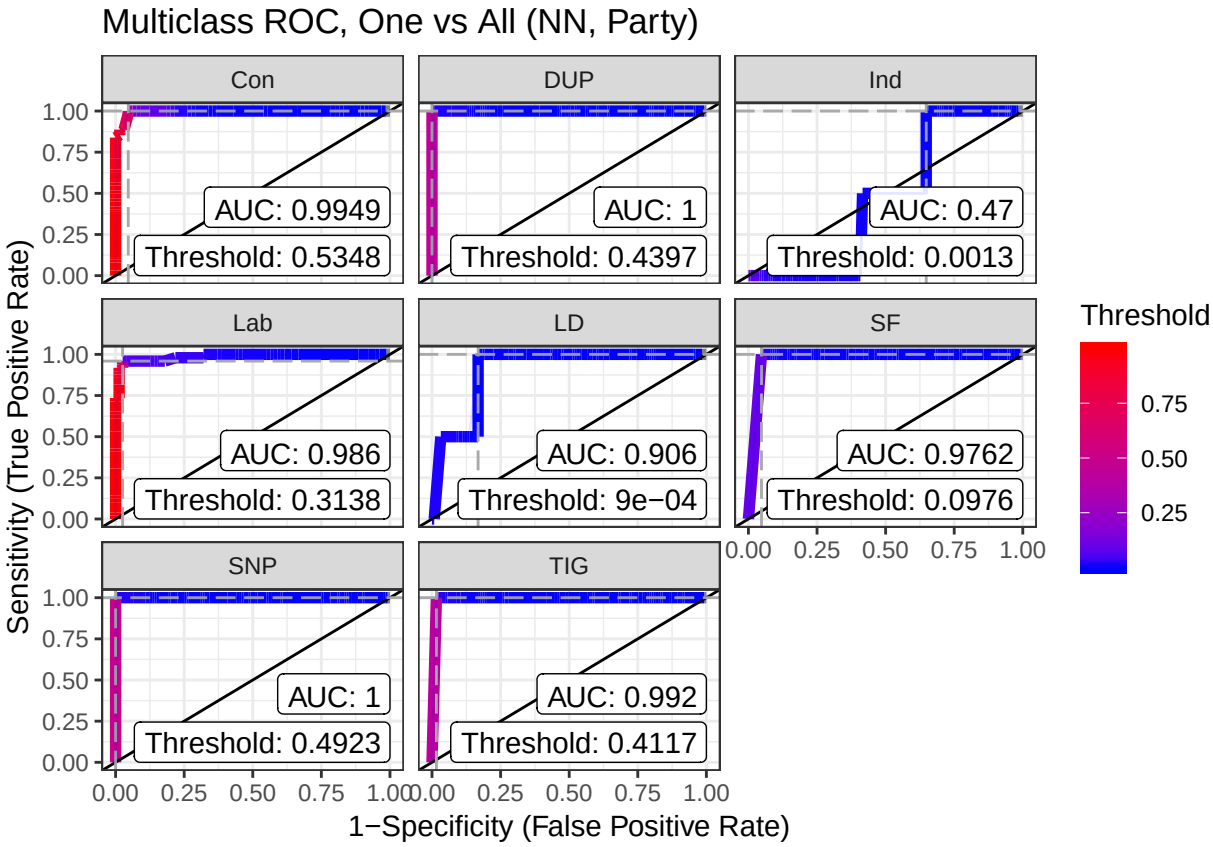
```
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   0   0   0   0   1  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.9449
##                  95% CI : (0.8897, 0.9776)
##     No Information Rate : 0.4882
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9078
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              1.0000    1.00000         NA    0.00000
## Specificity              0.9538    1.00000          1    1.00000
## Pos Pred Value           0.9538    1.00000         NA        NaN
## Neg Pred Value           1.0000    1.00000         NA    0.98425
## Prevalence               0.4882    0.01575          0    0.01575
## Detection Rate           0.4882    0.01575          0    0.00000
## Detection Prevalence     0.5118    0.01575          0    0.00000
## Balanced Accuracy        0.9769    1.00000         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9592   0.00000        NA  0.000000    1.00000
## Specificity              0.9744   1.00000         1  1.000000    1.00000
## Pos Pred Value           0.9592       NaN        NA       NaN    1.00000
## Neg Pred Value           0.9744   0.98425        NA  0.992126    1.00000
## Prevalence               0.3858   0.01575         0  0.007874    0.05512
## Detection Rate           0.3701   0.00000         0  0.000000    0.05512
## Detection Prevalence     0.3858   0.00000         0  0.000000    0.05512
## Balanced Accuracy        0.9668   0.50000        NA  0.500000    1.00000
##                      Class: TIG
## Sensitivity             1.00000
## Specificity             0.98400
## Pos Pred Value          0.50000
## Neg Pred Value          1.00000
## Prevalence              0.01575
## Detection Rate          0.01575
## Detection Prevalence    0.03150
## Balanced Accuracy       0.99200
```

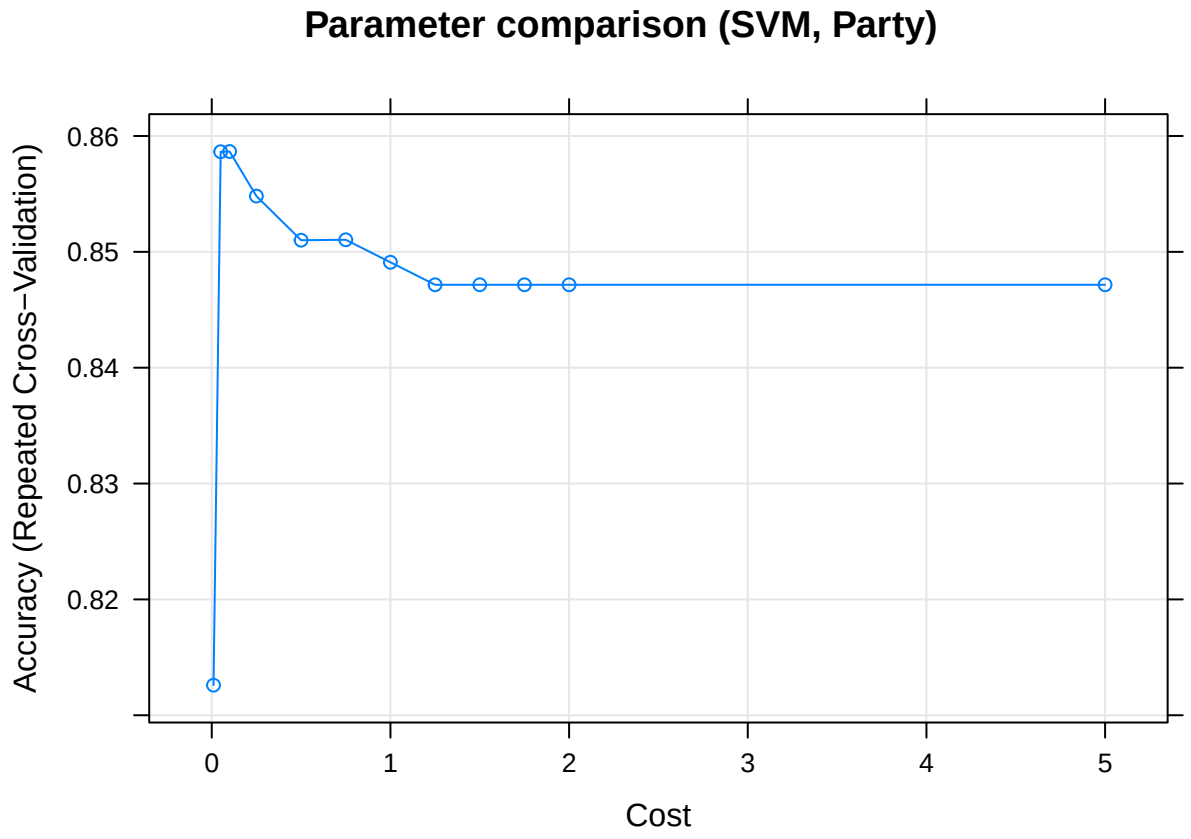
```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): Grn, PC
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response': Grn,PC.
```




```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 8 levels of response: Con, DUP, Ind, Lab, LD, SF, SNP,
## Multi-class area under the curve: 0.847
```


Multiclass ROC, One vs All (NN, Party)
Con
DUP
Ind
Lab
LD
SF
SNP
TIG
AUC: 0.9949
Threshold: 0.5348
AUC: 1
Threshold: 0.4397
AUC: 0.47
Threshold: 0.0013
AUC: 0.986
Threshold: 0.3138
AUC: 0.906
Threshold: 9e−04
AUC: 0.9762
Threshold: 0.0976
AUC: 1
Threshold: 0.4923
AUC: 0.992
Threshold: 0.4117
Sensitivity (True Positive Rate)
1−Specificity (False Positive Rate)
0.00
0.25
0.50
0.75
1.00
Threshold
0.75
0.50
0.25

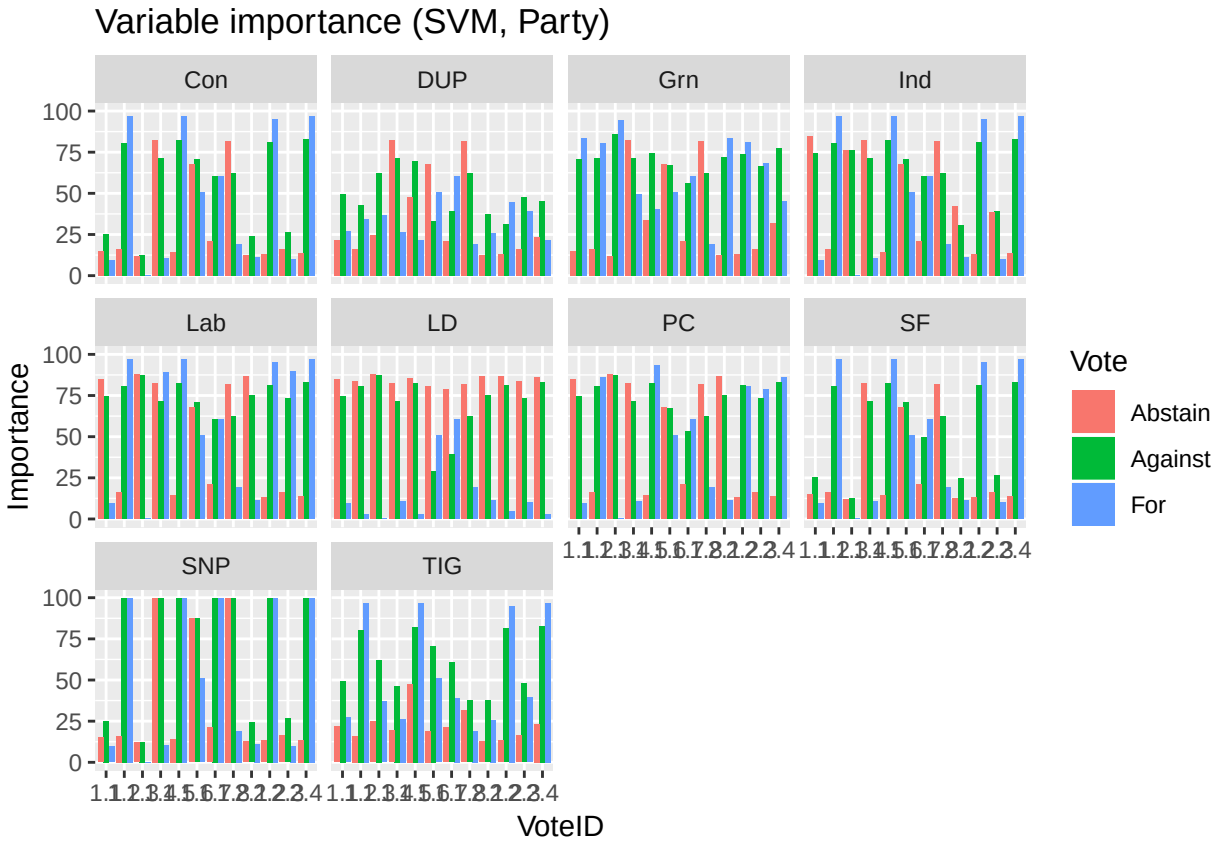


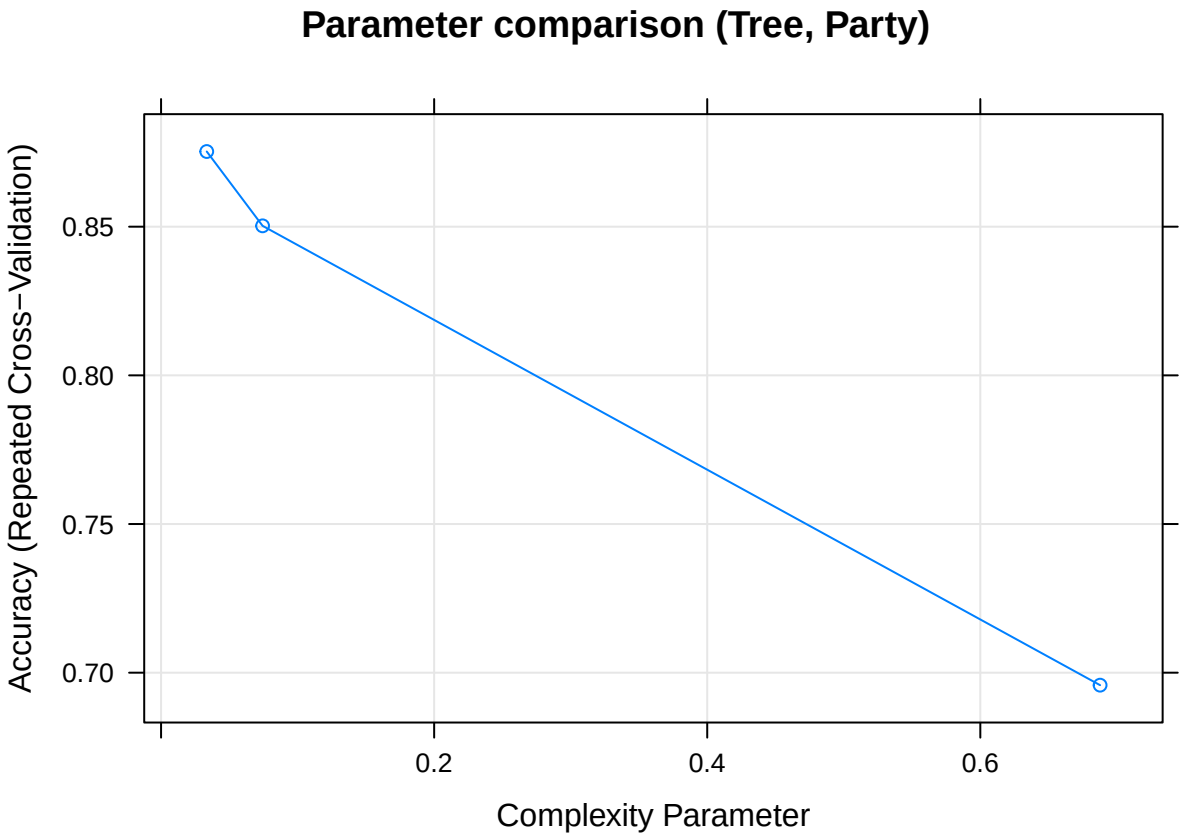

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  60   0   0   1   1  0  0  1   0   0
##        DUP   1   2   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  45  1  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   1   0   0   0   3  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.9134
##                  95% CI : (0.8503, 0.956)
##     No Information Rate : 0.4882
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8582
##
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              0.9677    1.00000         NA    0.00000
## Specificity              0.9538    0.99200          1    1.00000
## Pos Pred Value           0.9524    0.66667         NA        NaN
## Neg Pred Value           0.9688    1.00000         NA    0.98425
## Prevalence               0.4882    0.01575          0    0.01575
## Detection Rate           0.4724    0.01575          0    0.00000
## Detection Prevalence     0.4961    0.02362          0    0.00000
## Balanced Accuracy        0.9608    0.99600         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9184   0.00000        NA  0.000000    1.00000
## Specificity              0.9744   1.00000         1  1.000000    1.00000
## Pos Pred Value           0.9574       NaN        NA       NaN    1.00000
## Neg Pred Value           0.9500   0.98425        NA  0.992126    1.00000
## Prevalence               0.3858   0.01575         0  0.007874    0.05512
## Detection Rate           0.3543   0.00000         0  0.000000    0.05512
## Detection Prevalence     0.3701   0.00000         0  0.000000    0.05512
## Balanced Accuracy        0.9464   0.50000        NA  0.500000    1.00000
##                      Class: TIG
## Sensitivity             1.00000
## Specificity             0.96000
## Pos Pred Value          0.28571
## Neg Pred Value          1.00000
## Prevalence              0.01575
## Detection Rate          0.01575
## Detection Prevalence    0.05512
## Balanced Accuracy       0.98000
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```


Variable importance (SVM, Party)
Con
DUP
Grn
Ind
Lab
LD
PC
SF
SNP
TIG
Importance
VoteID
Vote
Abstain
Against
For

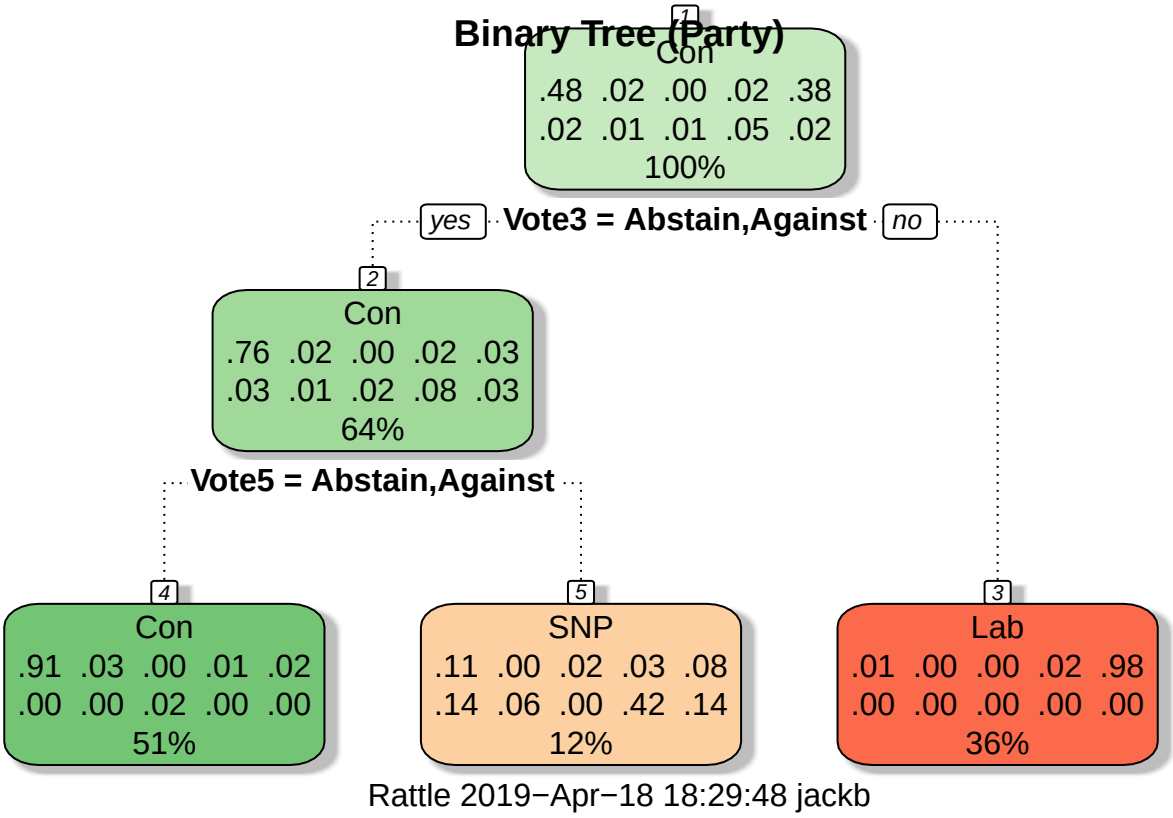



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  59   0   0   1   2  1  0  0   0   0
##        DUP   0   0   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  46  0  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   2   0   0   0   1  1  0  0   7   2
##        TIG   0   0   0   0   0  0  0  0   0   0
##
## Overall Statistics
##
##                Accuracy : 0.9106
##                  95% CI : (0.8456, 0.9545)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8478
##
##  Mcnemar's Test P-Value : NA
##
```

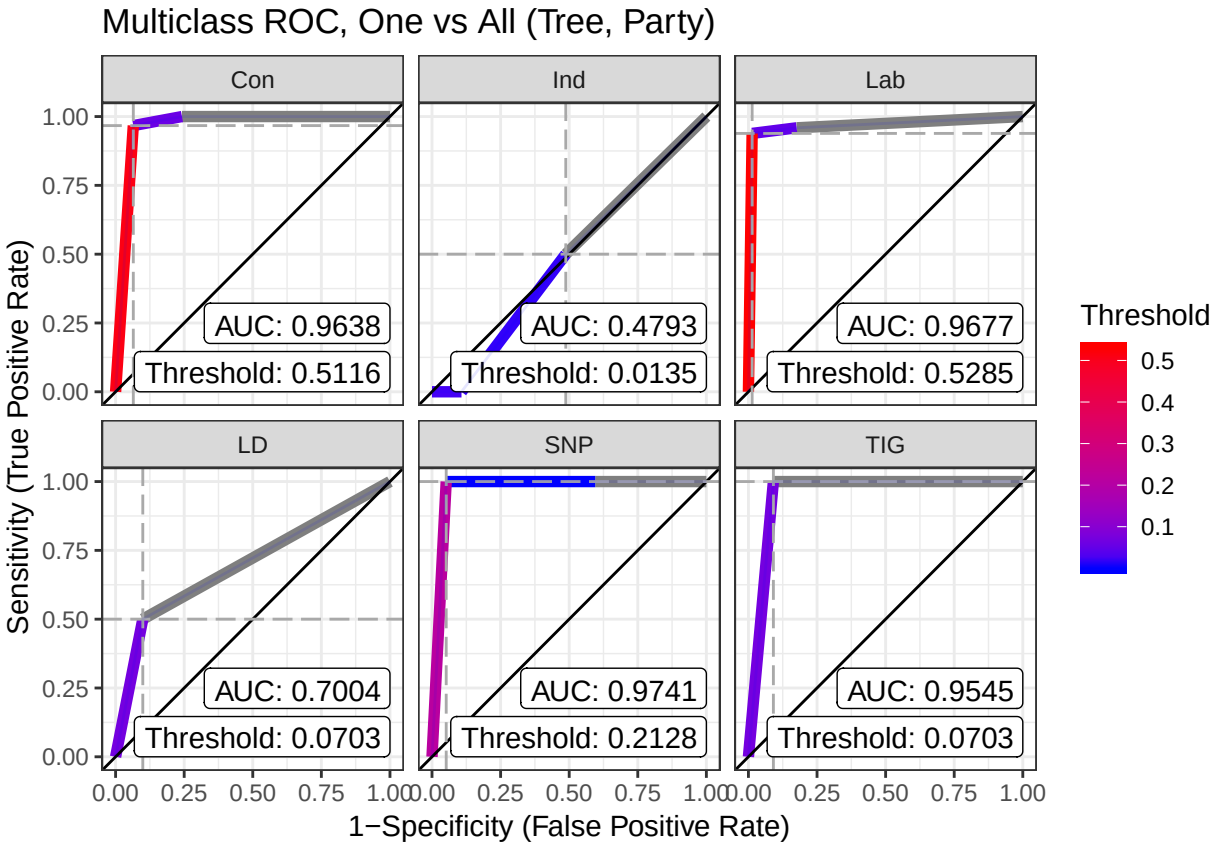
```
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              0.9672         NA         NA    0.00000
## Specificity              0.9355          1          1    1.00000
## Pos Pred Value           0.9365         NA         NA        NaN
## Neg Pred Value           0.9667         NA         NA    0.98374
## Prevalence               0.4959          0          0    0.01626
## Detection Rate           0.4797          0          0    0.00000
## Detection Prevalence     0.5122          0          0    0.00000
## Balanced Accuracy        0.9513         NA         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9388   0.00000        NA        NA    1.00000
## Specificity              0.9865   1.00000         1         1    0.94828
## Pos Pred Value           0.9787       NaN        NA        NA    0.53846
## Neg Pred Value           0.9605   0.98374        NA        NA    1.00000
## Prevalence               0.3984   0.01626         0         0    0.05691
## Detection Rate           0.3740   0.00000         0         0    0.05691
## Detection Prevalence     0.3821   0.00000         0         0    0.10569
## Balanced Accuracy        0.9626   0.50000        NA        NA    0.97414
##                      Class: TIG
## Sensitivity             0.00000
## Specificity             1.00000
## Pos Pred Value              NaN
## Neg Pred Value          0.98374
## Prevalence              0.01626
## Detection Rate          0.00000
## Detection Prevalence    0.00000
## Balanced Accuracy       0.50000
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF
```

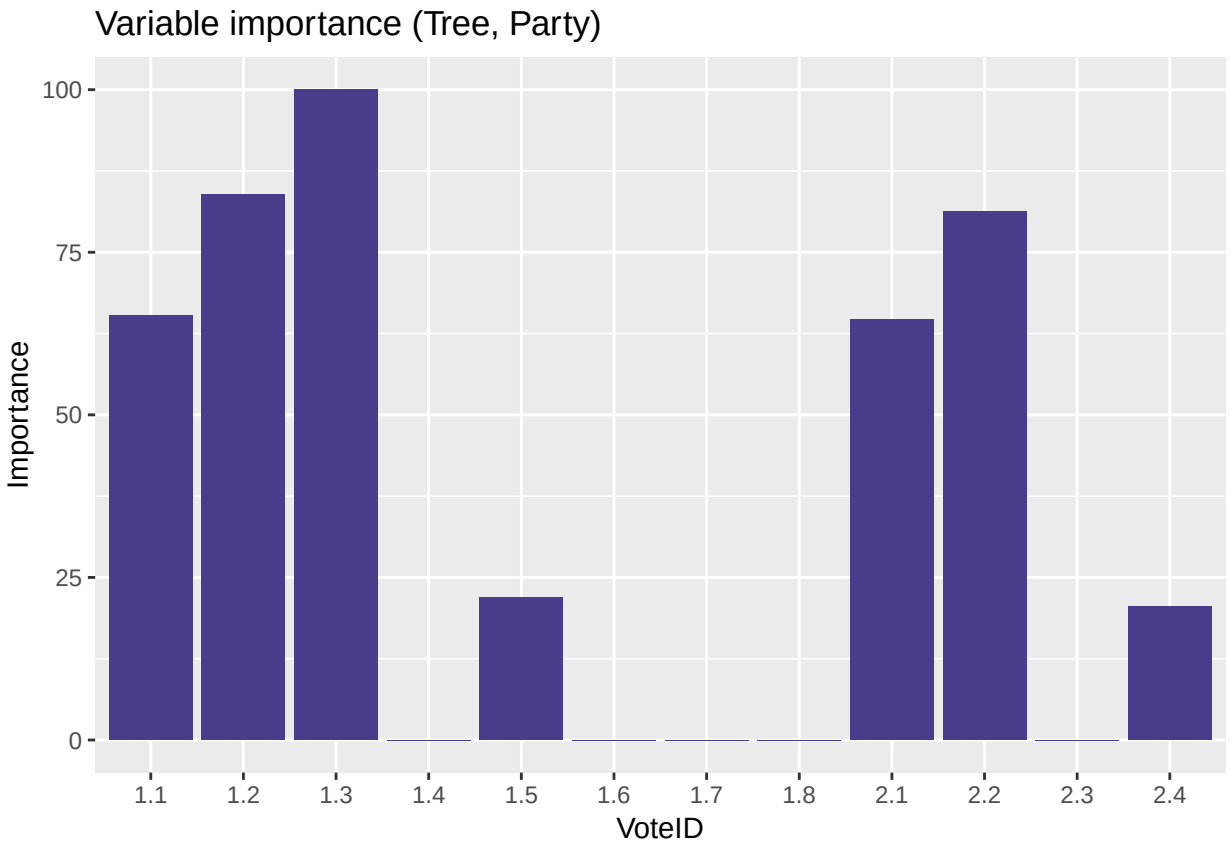
```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```

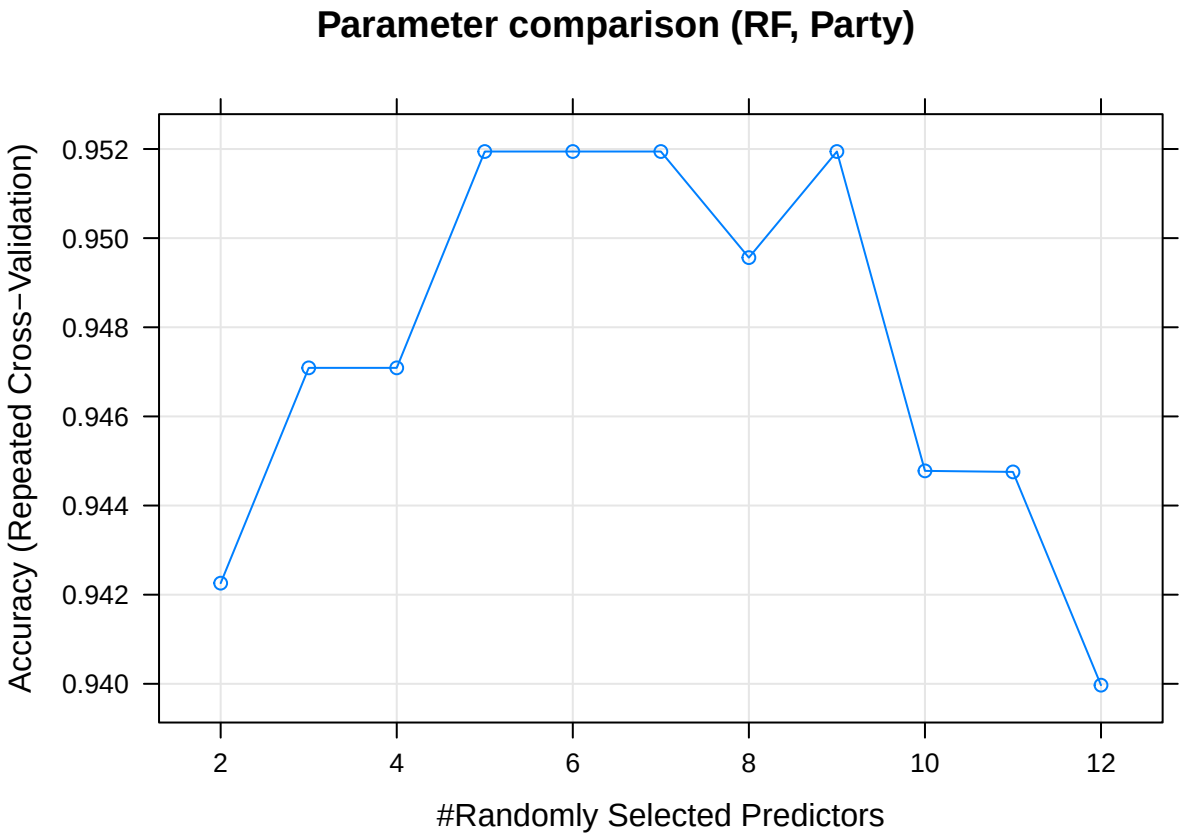



```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 6 levels of response: Con, Ind, Lab, LD, SNP, TIG.
## Multi-class area under the curve: 0.7174
```




```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```


Variable importance (Tree, Party)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

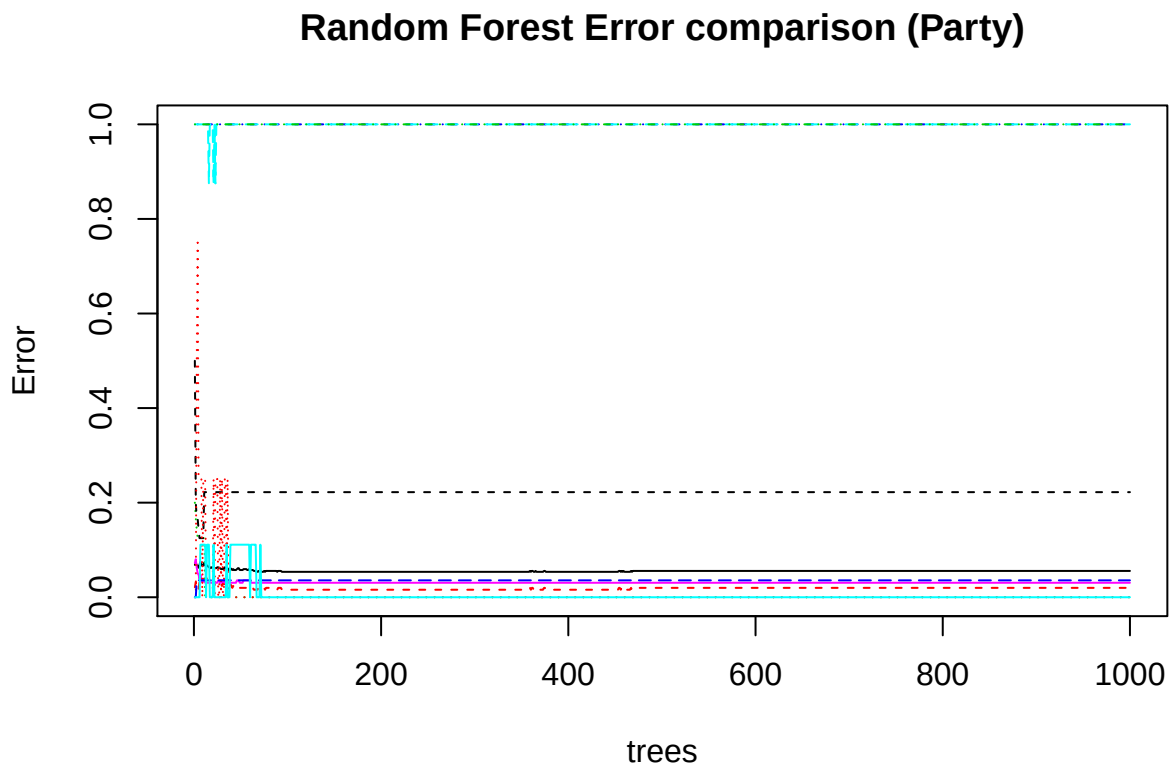



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  61   0   0   1   2  0  0  0   0   0
##        DUP   0   0   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  46  1  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   0   0   0   0   1  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.9431
##                  95% CI : (0.8863, 0.9768)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9023
##
##  Mcnemar's Test P-Value : NA
##
```

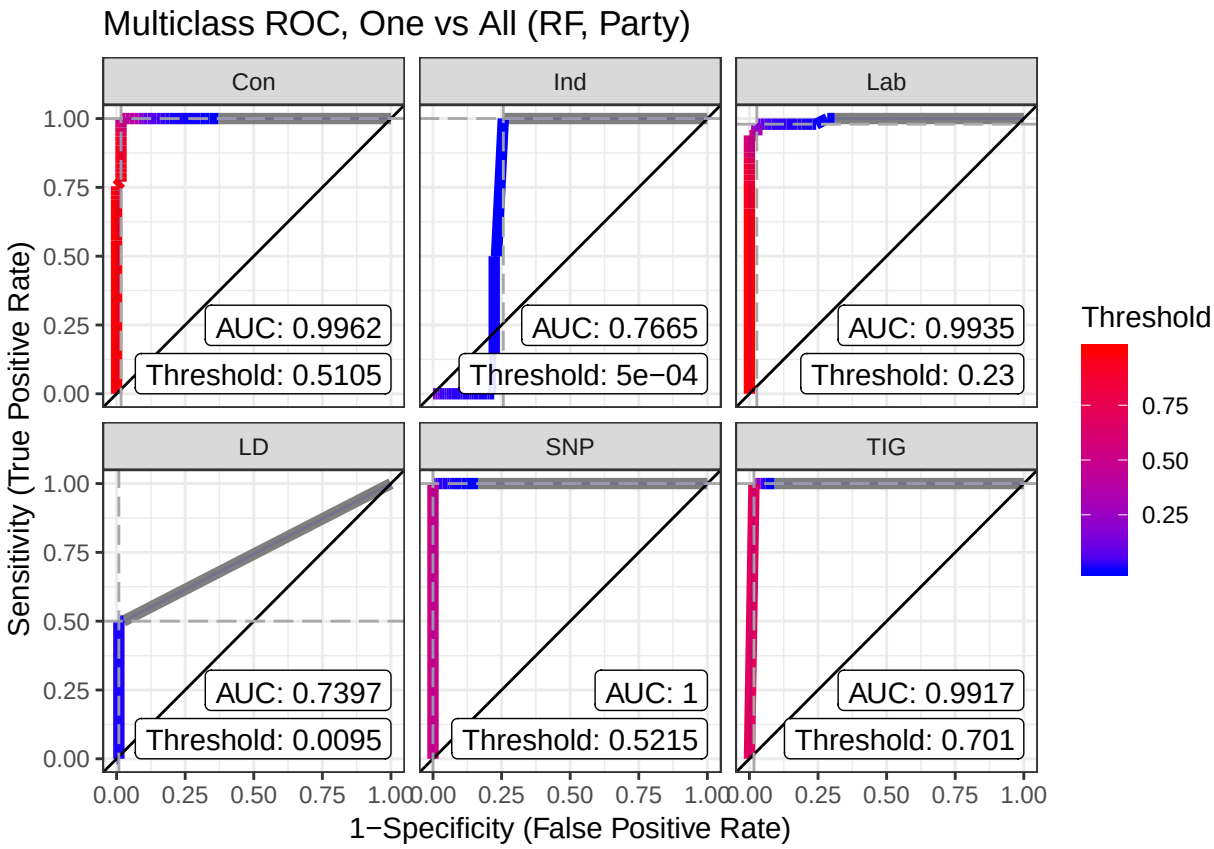
```
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              1.0000         NA         NA    0.00000
## Specificity              0.9516          1          1    1.00000
## Pos Pred Value           0.9531         NA         NA        NaN
## Neg Pred Value           1.0000         NA         NA    0.98374
## Prevalence               0.4959          0          0    0.01626
## Detection Rate           0.4959          0          0    0.00000
## Detection Prevalence     0.5203          0          0    0.00000
## Balanced Accuracy        0.9758         NA         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9388   0.00000        NA        NA    1.00000
## Specificity              0.9730   1.00000         1         1    1.00000
## Pos Pred Value           0.9583       NaN        NA        NA    1.00000
## Neg Pred Value           0.9600   0.98374        NA        NA    1.00000
## Prevalence               0.3984   0.01626         0         0    0.05691
## Detection Rate           0.3740   0.00000         0         0    0.05691
## Detection Prevalence     0.3902   0.00000         0         0    0.05691
## Balanced Accuracy        0.9559   0.50000        NA        NA    1.00000
##                      Class: TIG
## Sensitivity             1.00000
## Specificity             0.98347
## Pos Pred Value          0.50000
## Neg Pred Value          1.00000
## Prevalence              0.01626
## Detection Rate          0.01626
## Detection Prevalence    0.03252
## Balanced Accuracy       0.99174
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF
```

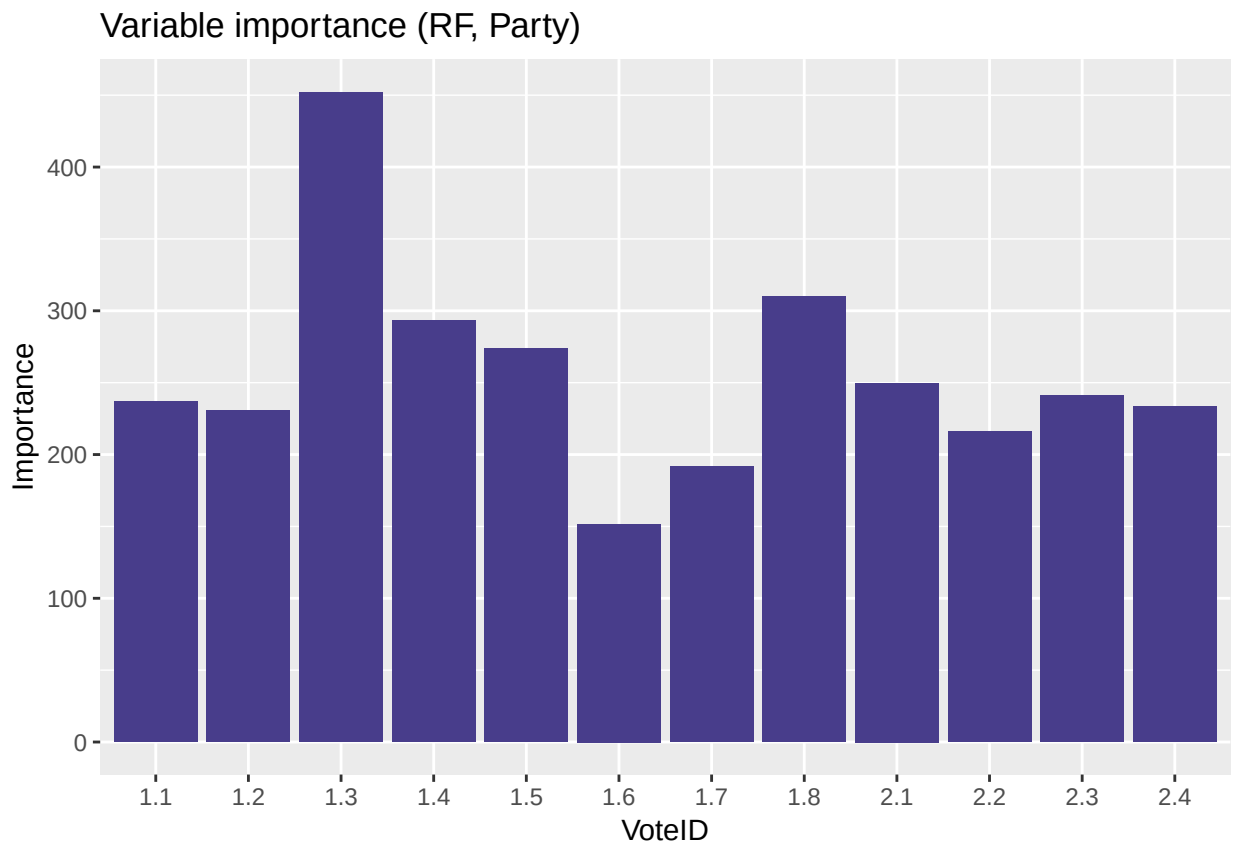
```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```




```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 6 levels of response: Con, Ind, Lab, LD, SNP, TIG.
## Multi-class area under the curve: 0.8598
```




```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```

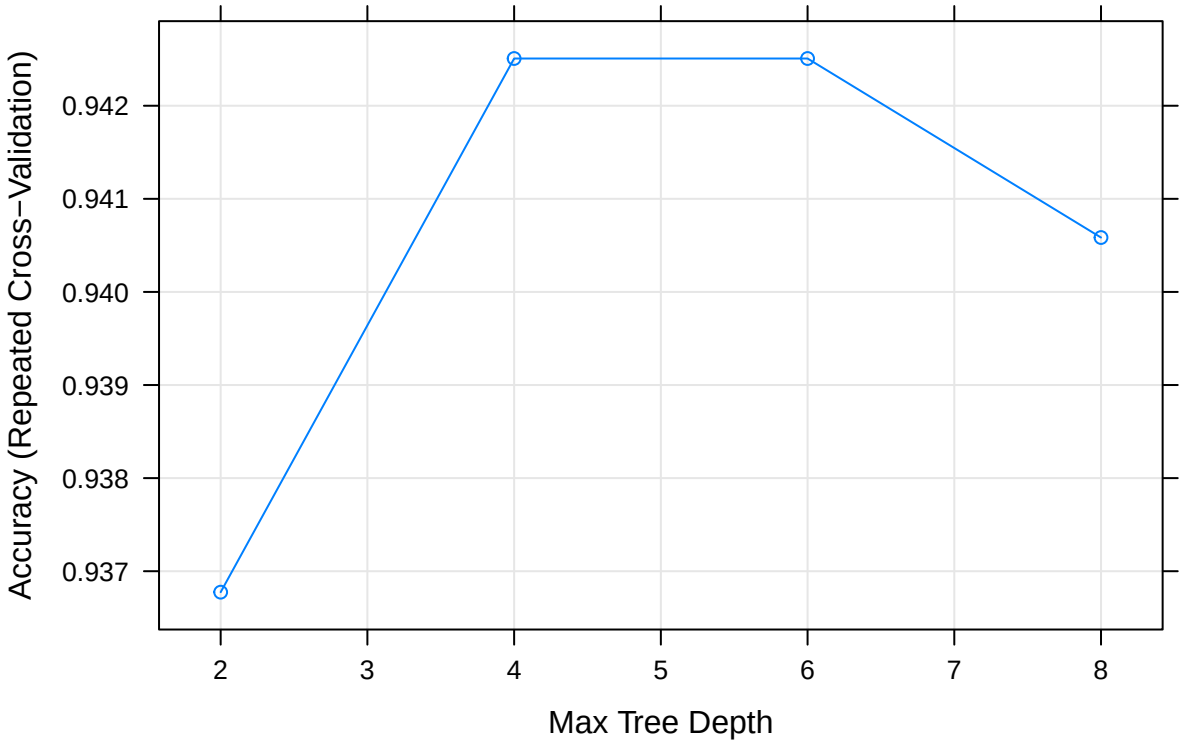



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  61   0   0   1   1  0  0  0   0   0
##        DUP   0   0   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  47  1  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   0   0   0   0   1  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9164
##
##  Mcnemar's Test P-Value : NA
##
```

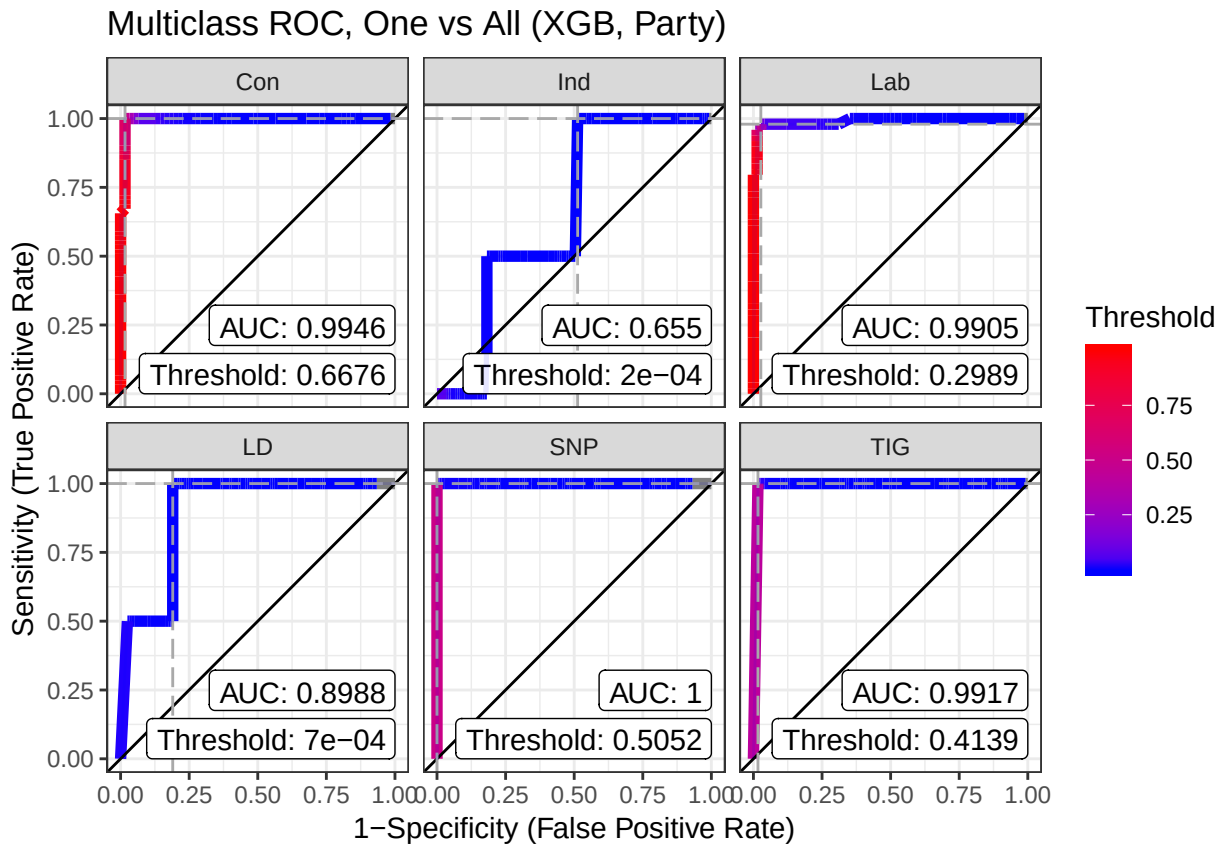
```
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              1.0000         NA         NA    0.00000
## Specificity              0.9677          1          1    1.00000
## Pos Pred Value           0.9683         NA         NA        NaN
## Neg Pred Value           1.0000         NA         NA    0.98374
## Prevalence               0.4959          0          0    0.01626
## Detection Rate           0.4959          0          0    0.00000
## Detection Prevalence     0.5122          0          0    0.00000
## Balanced Accuracy        0.9839         NA         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9592   0.00000        NA        NA    1.00000
## Specificity              0.9730   1.00000         1         1    1.00000
## Pos Pred Value           0.9592       NaN        NA        NA    1.00000
## Neg Pred Value           0.9730   0.98374        NA        NA    1.00000
## Prevalence               0.3984   0.01626         0         0    0.05691
## Detection Rate           0.3821   0.00000         0         0    0.05691
## Detection Prevalence     0.3984   0.00000         0         0    0.05691
## Balanced Accuracy        0.9661   0.50000        NA        NA    1.00000
##                      Class: TIG
## Sensitivity              1.00000
## Specificity              0.98347
## Pos Pred Value           0.50000
## Neg Pred Value           1.00000
## Prevalence               0.01626
## Detection Rate           0.01626
## Detection Prevalence     0.03252
## Balanced Accuracy        0.99174
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```




```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 6 levels of response: Con, Ind, Lab, LD, SNP, TIG.
## Multi-class area under the curve: 0.8229
```



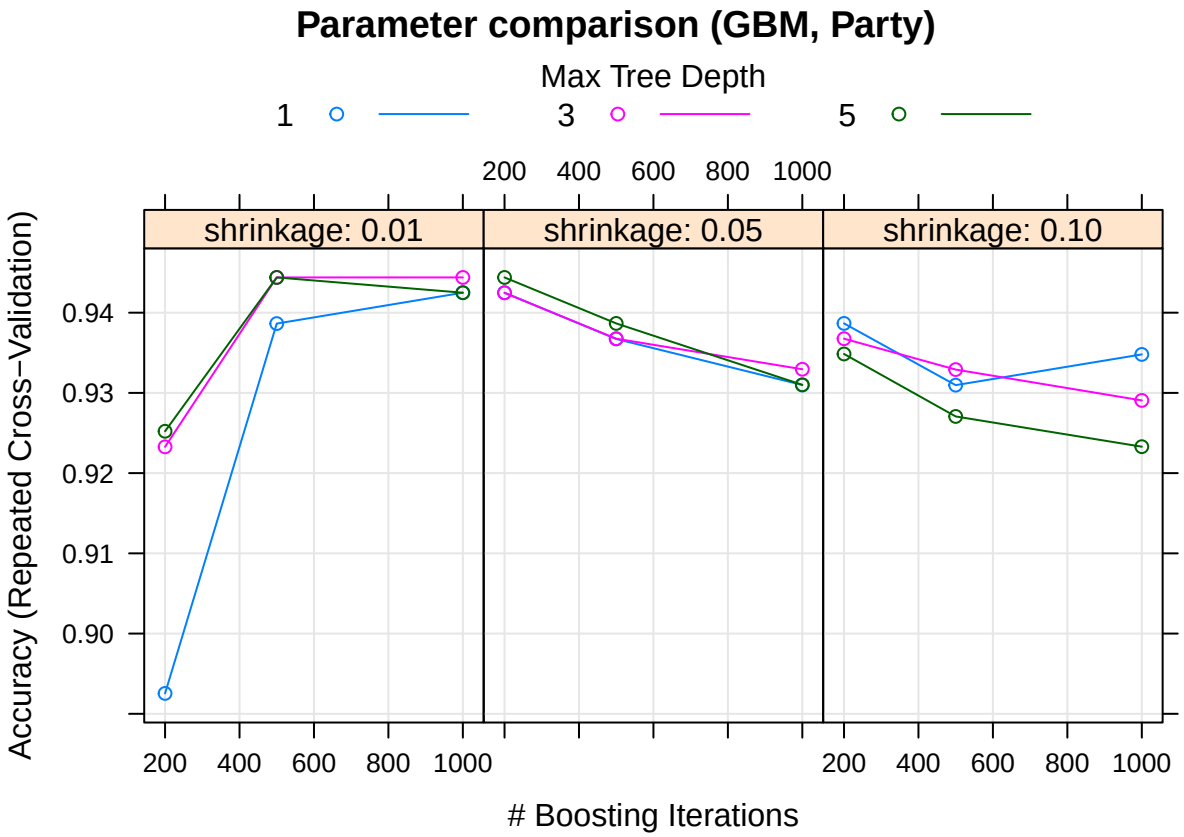

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```


```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  61   0   0   1   1  0  0  0   0   0
##        DUP   0   0   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  47  1  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   0   0   0   0   1  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9164
##
##  Mcnemar's Test P-Value : NA
##
```

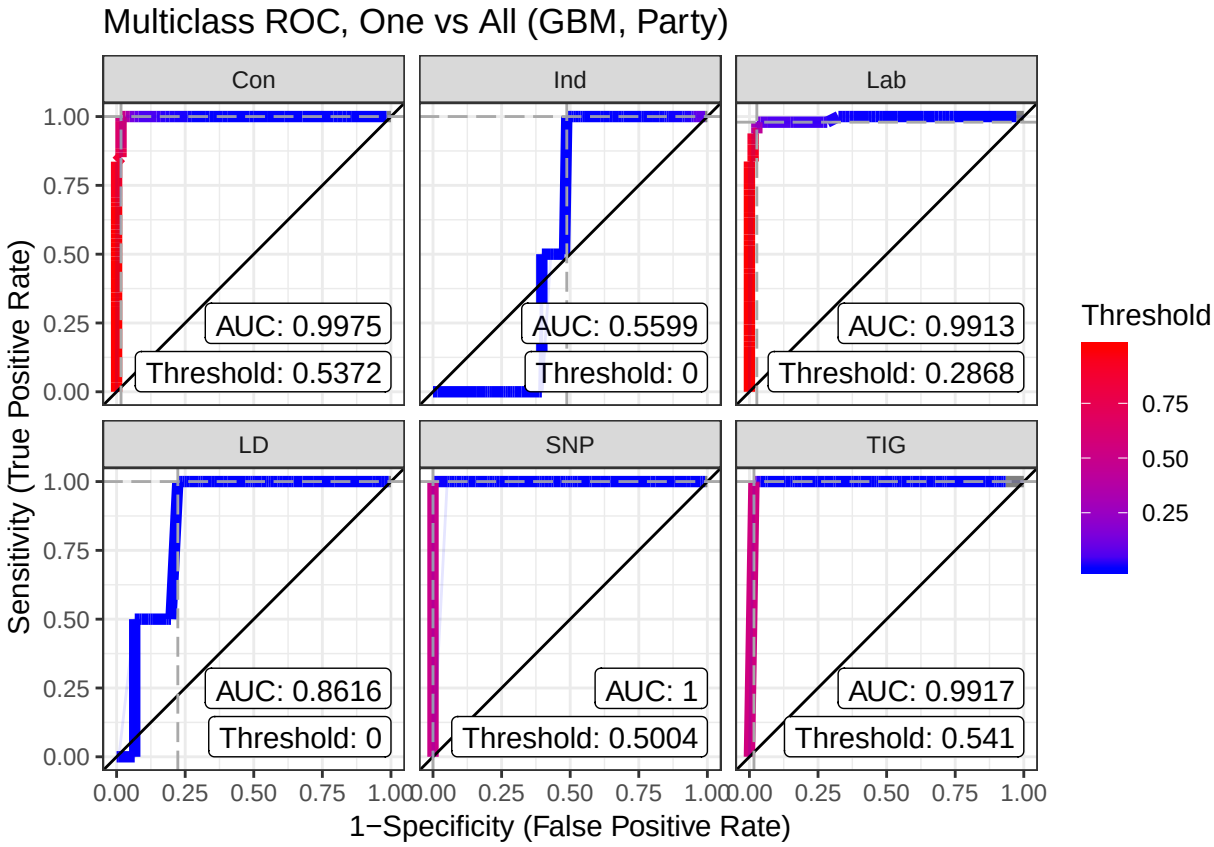
```
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              1.0000         NA         NA    0.00000
## Specificity              0.9677          1          1    1.00000
## Pos Pred Value           0.9683         NA         NA        NaN
## Neg Pred Value           1.0000         NA         NA    0.98374
## Prevalence               0.4959          0          0    0.01626
## Detection Rate           0.4959          0          0    0.00000
## Detection Prevalence     0.5122          0          0    0.00000
## Balanced Accuracy        0.9839         NA         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9592   0.00000        NA        NA    1.00000
## Specificity              0.9730   1.00000         1         1    1.00000
## Pos Pred Value           0.9592       NaN        NA        NA    1.00000
## Neg Pred Value           0.9730   0.98374        NA        NA    1.00000
## Prevalence               0.3984   0.01626         0         0    0.05691
## Detection Rate           0.3821   0.00000         0         0    0.05691
## Detection Prevalence     0.3984   0.00000         0         0    0.05691
## Balanced Accuracy        0.9661   0.50000        NA        NA    1.00000
##                      Class: TIG
## Sensitivity              1.00000
## Specificity              0.98347
## Pos Pred Value           0.50000
## Neg Pred Value           1.00000
## Prevalence               0.01626
## Detection Rate           0.01626
## Detection Prevalence     0.03252
## Balanced Accuracy        0.99174
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF
```

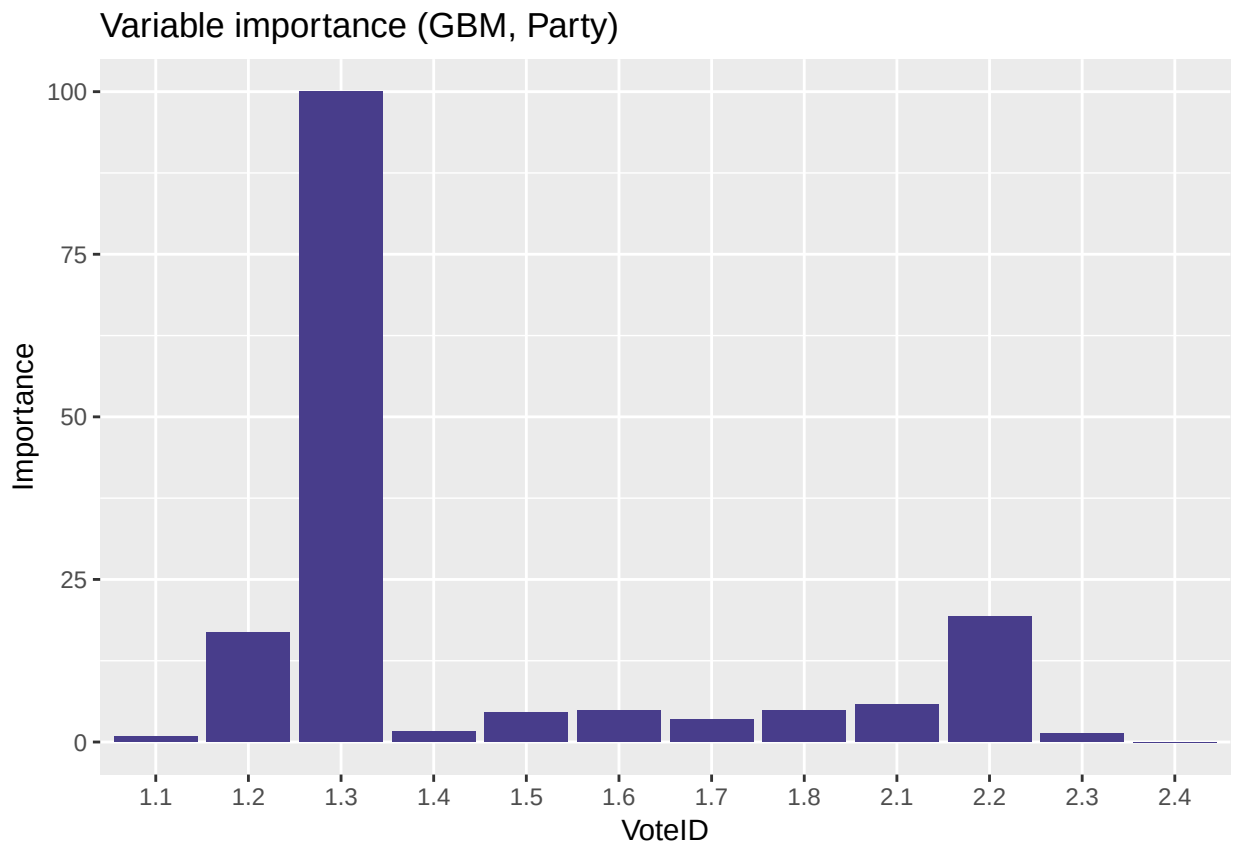
```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```




```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 6 levels of response: Con, Ind, Lab, LD, SNP, TIG.
## Multi-class area under the curve: 0.8024
```




```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```

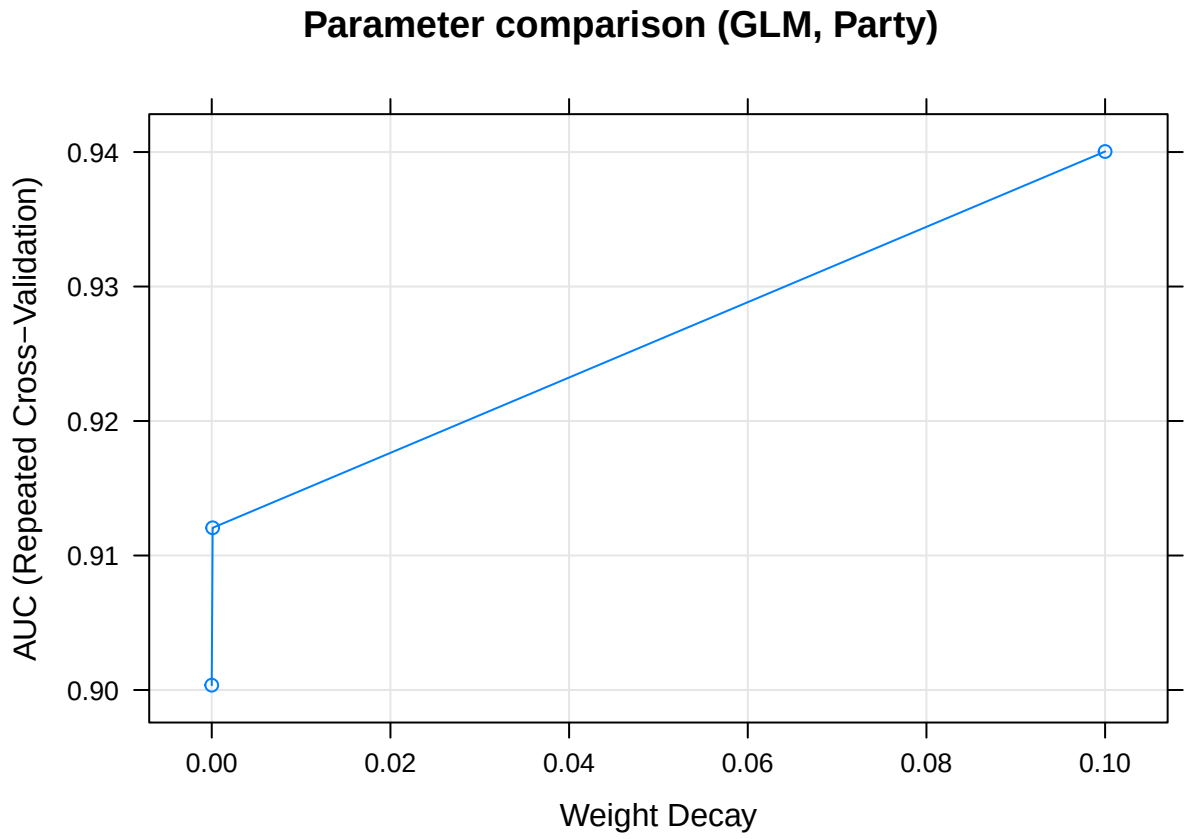



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  61   0   0   1   1  0  0  0   0   0
##        DUP   0   0   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   0  0  0  0   0   0
##        Lab   0   0   0   1  47  1  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    0   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   0   0   0   0   1  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9164
##
##  Mcnemar's Test P-Value : NA
##
```

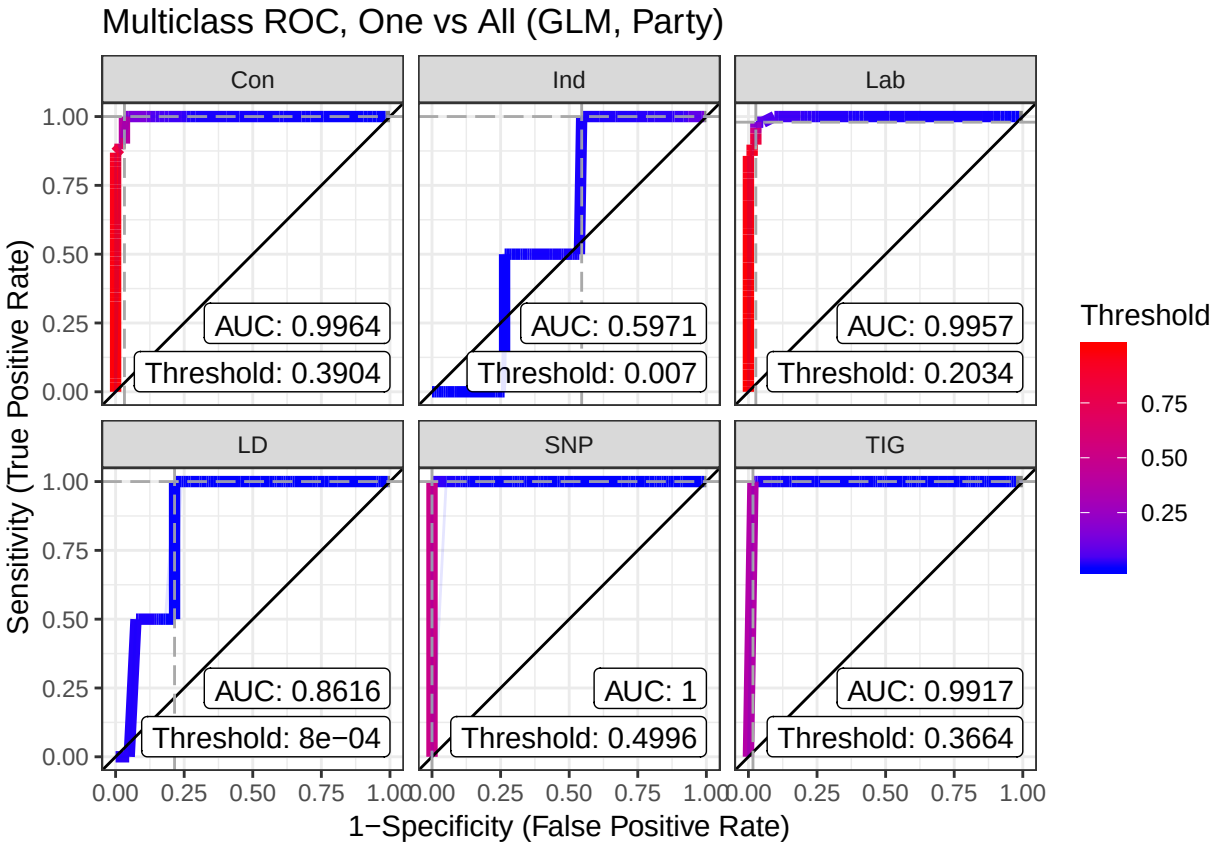
```
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              1.0000         NA         NA    0.00000
## Specificity              0.9677          1          1    1.00000
## Pos Pred Value           0.9683         NA         NA        NaN
## Neg Pred Value           1.0000         NA         NA    0.98374
## Prevalence               0.4959          0          0    0.01626
## Detection Rate           0.4959          0          0    0.00000
## Detection Prevalence     0.5122          0          0    0.00000
## Balanced Accuracy        0.9839         NA         NA    0.50000
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.9592   0.00000        NA        NA    1.00000
## Specificity              0.9730   1.00000         1         1    1.00000
## Pos Pred Value           0.9592       NaN        NA        NA    1.00000
## Neg Pred Value           0.9730   0.98374        NA        NA    1.00000
## Prevalence               0.3984   0.01626         0         0    0.05691
## Detection Rate           0.3821   0.00000         0         0    0.05691
## Detection Prevalence     0.3984   0.00000         0         0    0.05691
## Balanced Accuracy        0.9661   0.50000        NA        NA    1.00000
##                      Class: TIG
## Sensitivity              1.00000
## Specificity              0.98347
## Pos Pred Value           0.50000
## Neg Pred Value           1.00000
## Prevalence               0.01626
## Detection Rate           0.01626
## Detection Prevalence     0.03252
## Balanced Accuracy        0.99174
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF
```

```
## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```

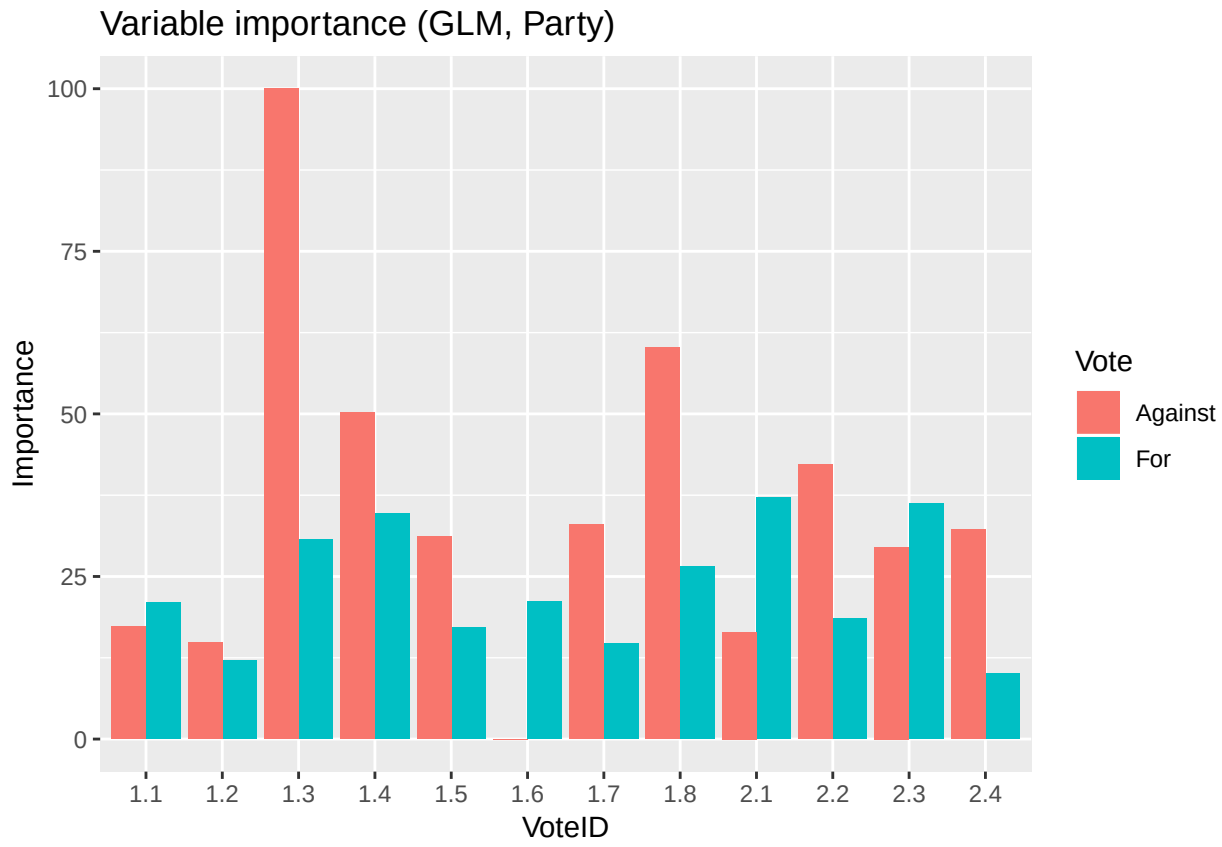



```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 6 levels of response: Con, Ind, Lab, LD, SNP, TIG.
## Multi-class area under the curve: 0.7982
```



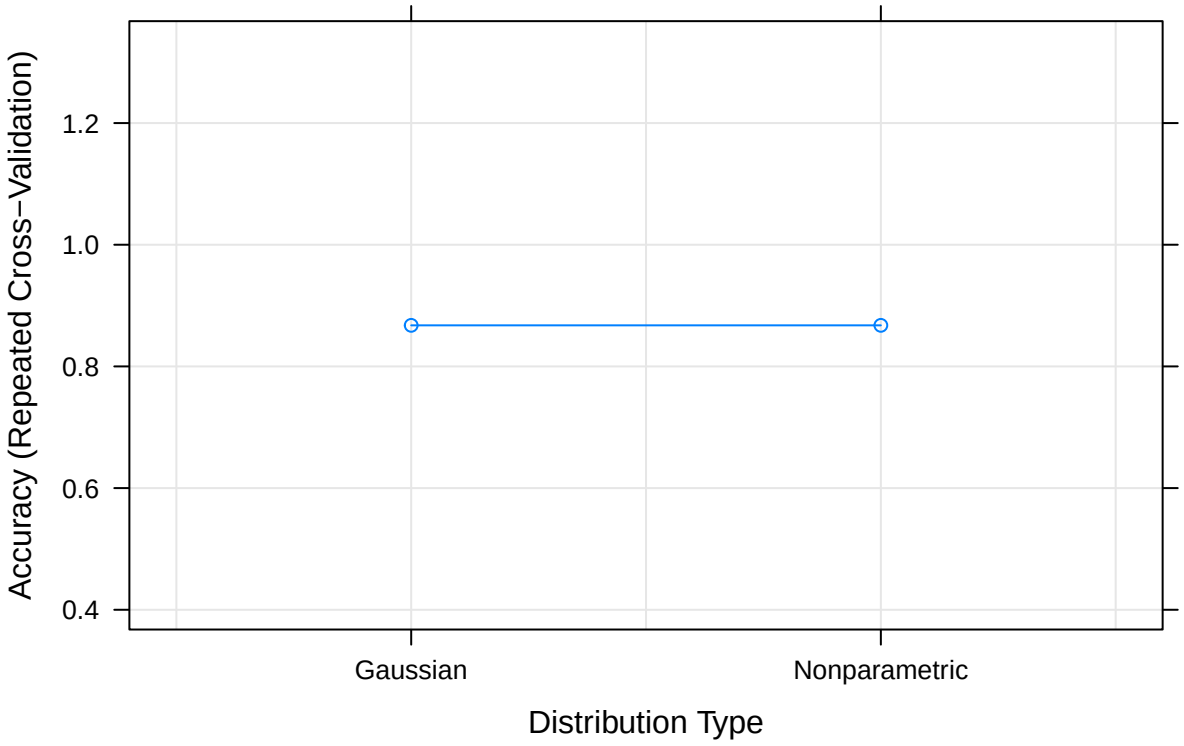

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73
```

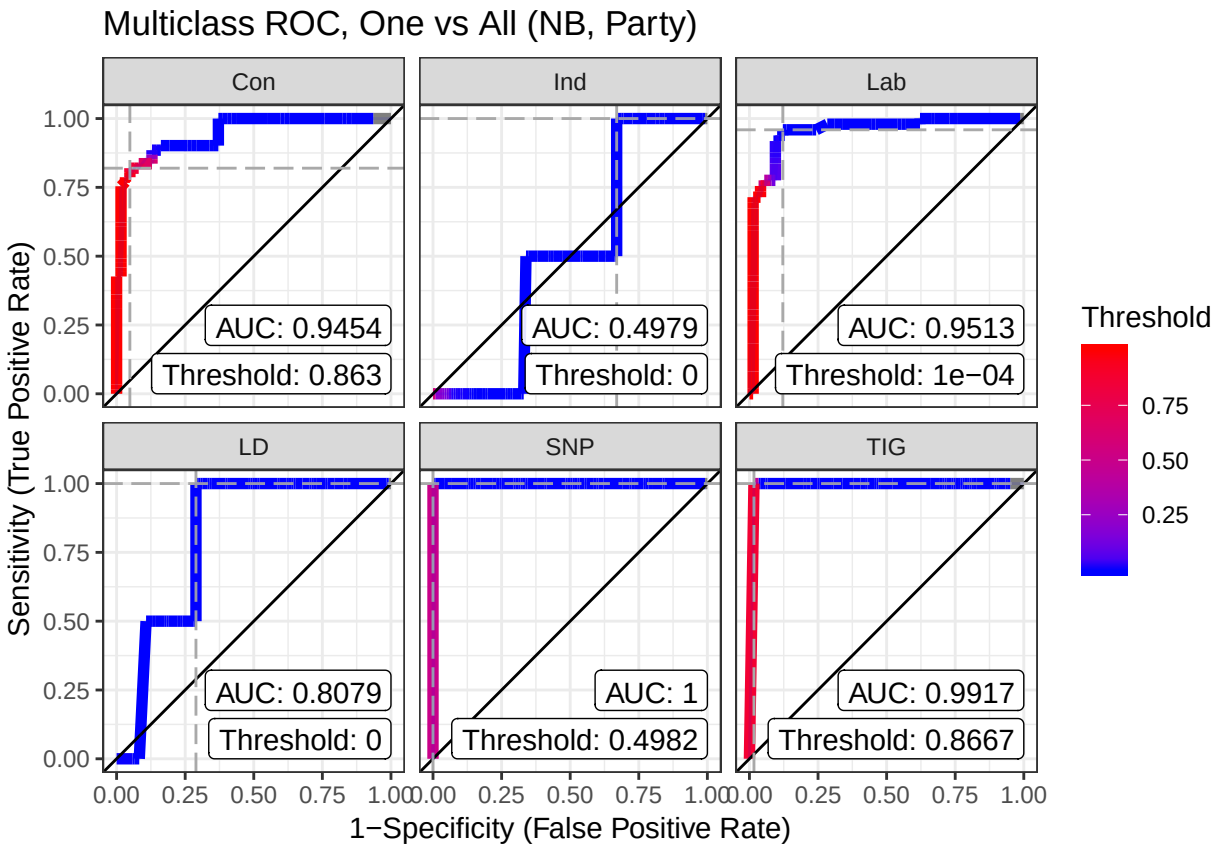


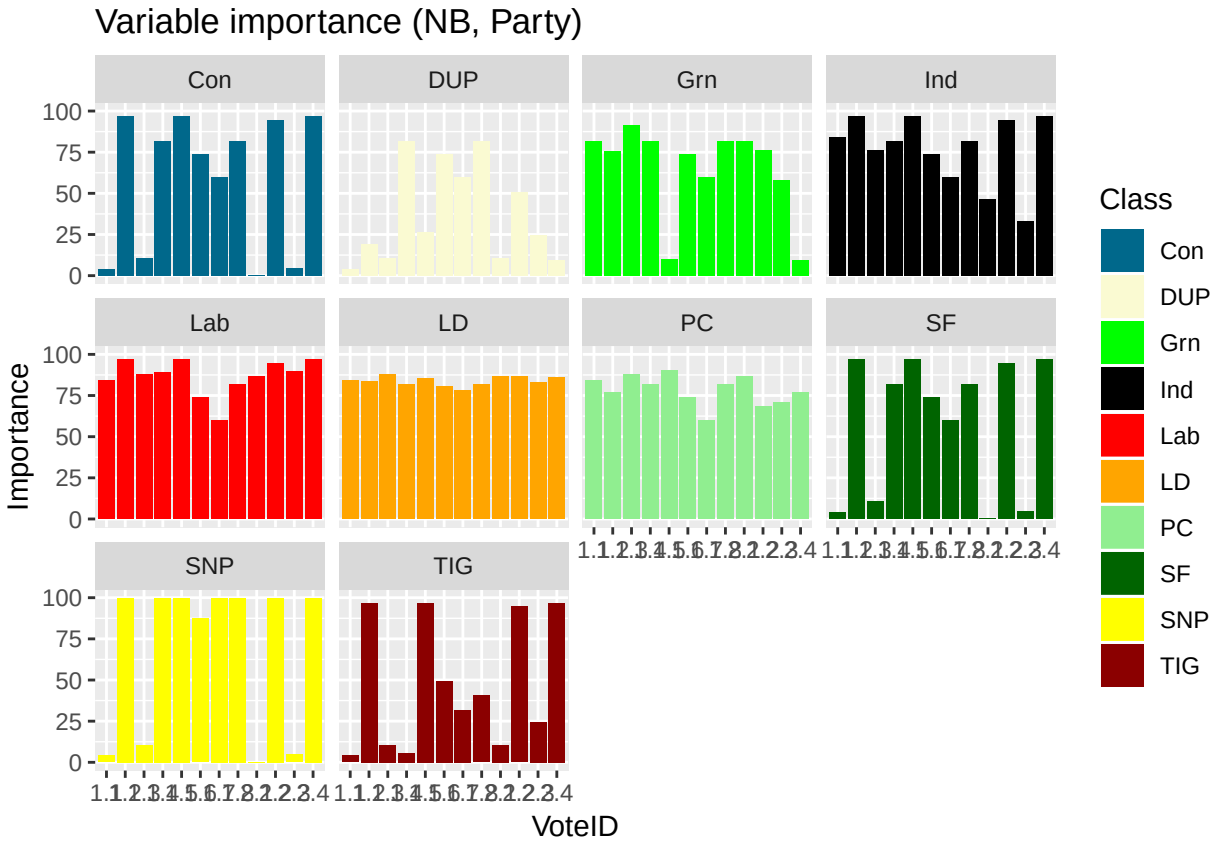

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con DUP Grn Ind Lab LD PC SF SNP TIG
##        Con  53   0   0   1   7  0  0  0   0   0
##        DUP   0   0   0   0   0  0  0  0   0   0
##        Grn   0   0   0   0   0  0  0  0   0   0
##        Ind   0   0   0   0   1  0  0  0   0   0
##        Lab   2   0   0   1  38  1  0  0   0   0
##        LD    0   0   0   0   0  0  0  0   0   0
##        PC    0   0   0   0   0  0  0  0   0   0
##        SF    6   0   0   0   0  0  0  0   0   0
##        SNP   0   0   0   0   0  0  0  0   7   0
##        TIG   0   0   0   0   3  1  0  0   0   2
##
## Overall Statistics
##
##                Accuracy : 0.813
##                  95% CI : (0.7328, 0.8776)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : 3.236e-13
##
##                   Kappa : 0.6954
##
```

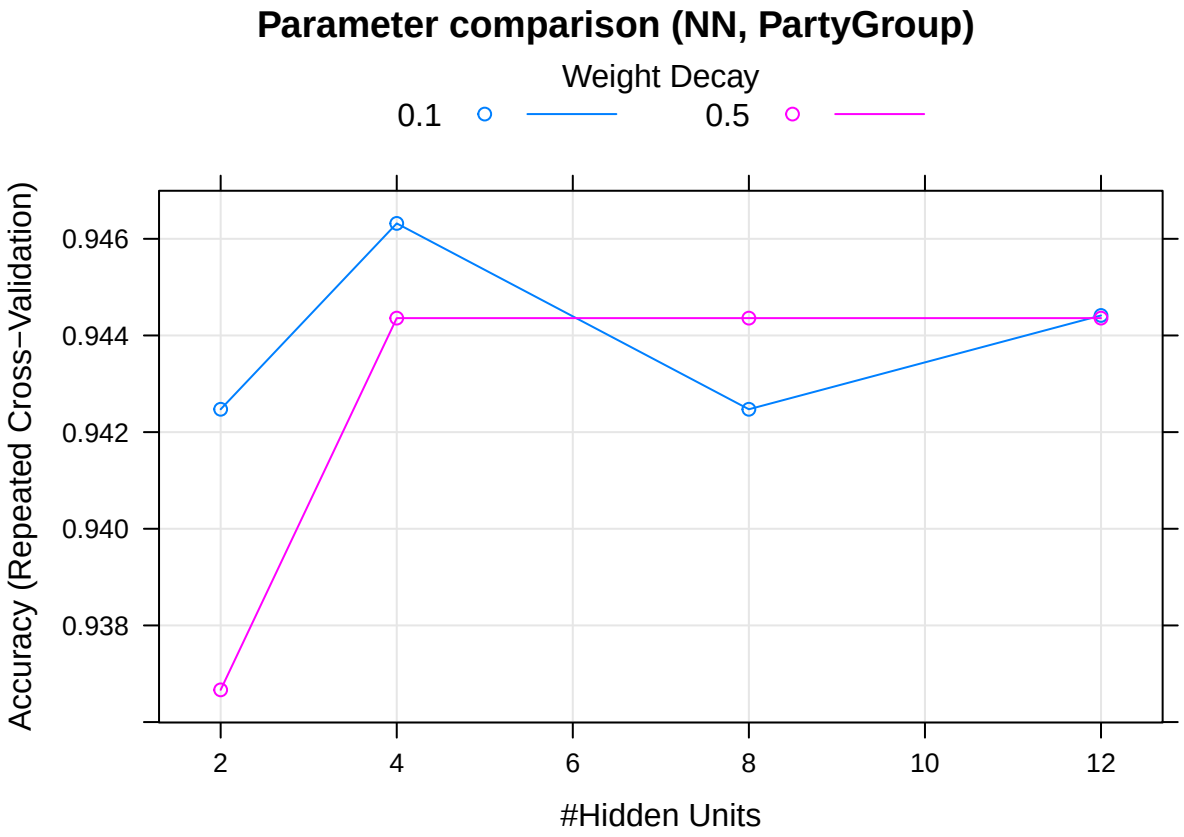
```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Con Class: DUP Class: Grn Class: Ind
## Sensitivity              0.8689         NA         NA    0.00000
## Specificity              0.8710          1          1    0.99174
## Pos Pred Value           0.8689         NA         NA    0.00000
## Neg Pred Value           0.8710         NA         NA    0.98361
## Prevalence               0.4959          0          0    0.01626
## Detection Rate           0.4309          0          0    0.00000
## Detection Prevalence     0.4959          0          0    0.00813
## Balanced Accuracy        0.8699         NA         NA    0.49587
##                      Class: Lab Class: LD Class: PC Class: SF Class: SNP
## Sensitivity              0.7755   0.00000        NA        NA    1.00000
## Specificity              0.9459   1.00000         1   0.95122    1.00000
## Pos Pred Value           0.9048       NaN        NA        NA    1.00000
## Neg Pred Value           0.8642   0.98374        NA        NA    1.00000
## Prevalence               0.3984   0.01626         0   0.00000    0.05691
## Detection Rate           0.3089   0.00000         0   0.00000    0.05691
## Detection Prevalence     0.3415   0.00000         0   0.04878    0.05691
## Balanced Accuracy        0.8607   0.50000        NA        NA    1.00000
##                      Class: TIG
## Sensitivity              1.00000
## Specificity              0.96694
## Pos Pred Value           0.33333
## Neg Pred Value           1.00000
## Prevalence               0.01626
## Detection Rate           0.01626
## Detection Prevalence     0.04878
## Balanced Accuracy        0.98347

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : No observation for response level(s): DUP, Grn, PC, SF

## Warning in multiclass.roc.multivariate(response, predictor, levels,
## percent, : The following classes were not found in 'response':
## DUP,Grn,PC,SF.
```




```
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 6 levels of response: Con, Ind, Lab, LD, SNP, TIG.
## Multi-class area under the curve: 0.7898
```

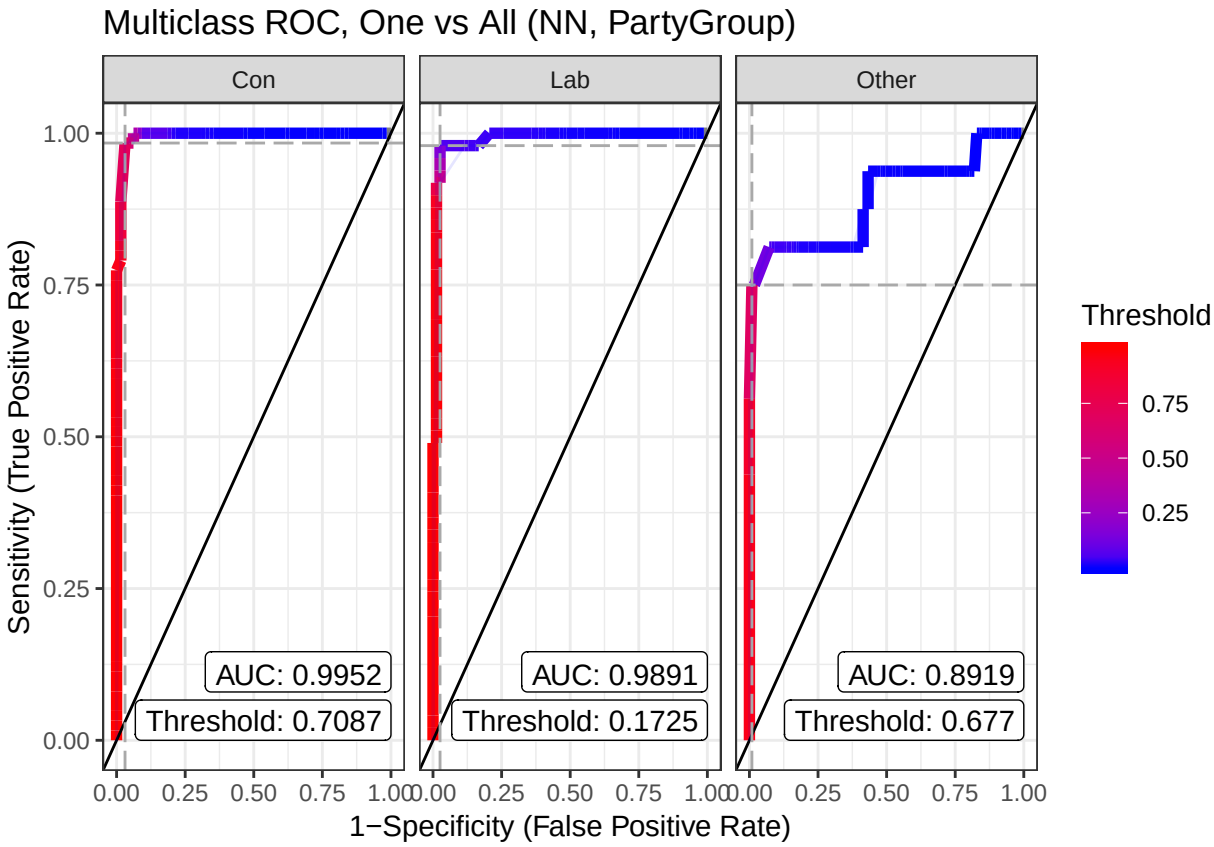

Multiclass ROC, One vs All (NB, Party)
Con
Ind
Lab
LD
SNP
TIG
AUC: 0.9454
Threshold: 0.863
AUC: 0.4979
Threshold: 0
AUC: 0.9513
Threshold: 1e−04
AUC: 0.8079
Threshold: 0
AUC: 1
Threshold: 0.4982
AUC: 0.9917
Threshold: 0.8667
Sensitivity (True Positive Rate)
1−Specificity (False Positive Rate)
0.00
0.25
0.50
0.75
1.00
Threshold
0.75
0.50
0.25

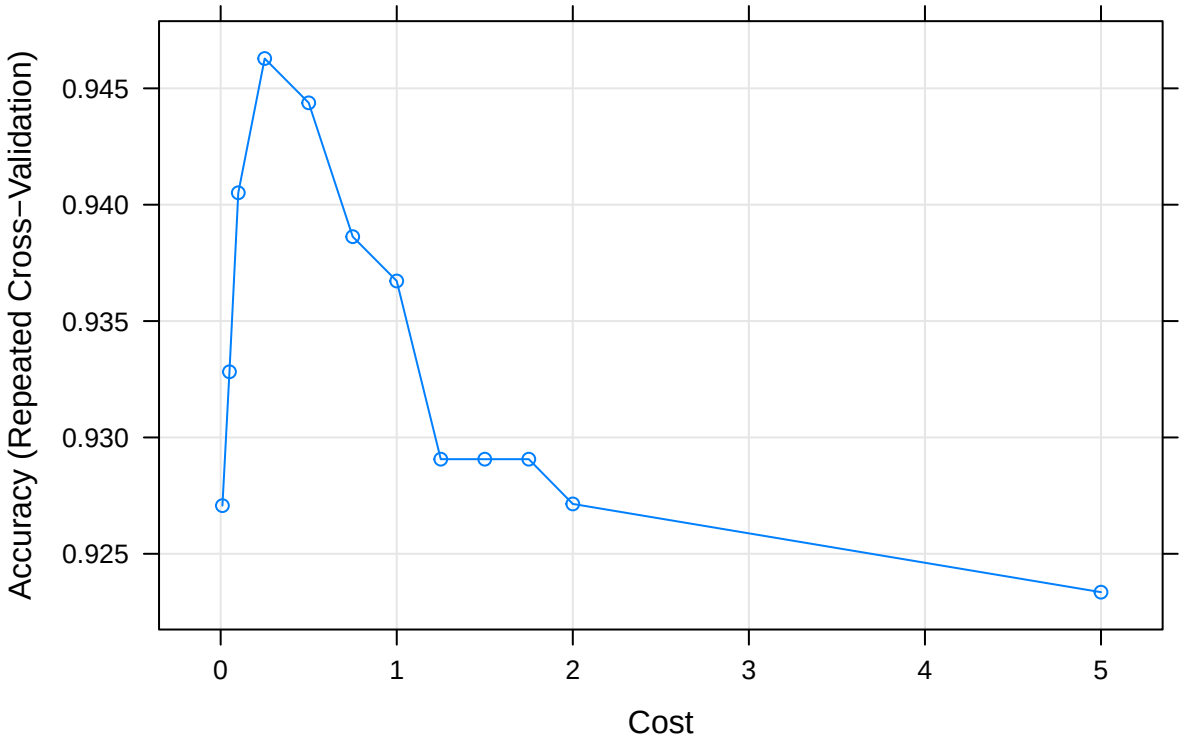

Variable importance (NB, Party)
Con
DUP
Grn
Ind
Lab
LD
PC
SF
SNP
TIG
Importance
VoteID
Class
Con
DUP
Grn
Ind
Lab
LD
PC
SF
SNP
TIG




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    62   2     2
##      Lab     0  46     2
##      Other   0   1    12
##
## Overall Statistics
##
##                Accuracy : 0.9449
##                  95% CI : (0.8897, 0.9776)
##     No Information Rate : 0.4882
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9062
##
##  Mcnemar's Test P-Value : 0.2276
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              1.0000     0.9388      0.75000
## Specificity              0.9385     0.9744      0.99099
## Pos Pred Value           0.9394     0.9583      0.92308
## Neg Pred Value           1.0000     0.9620      0.96491
```

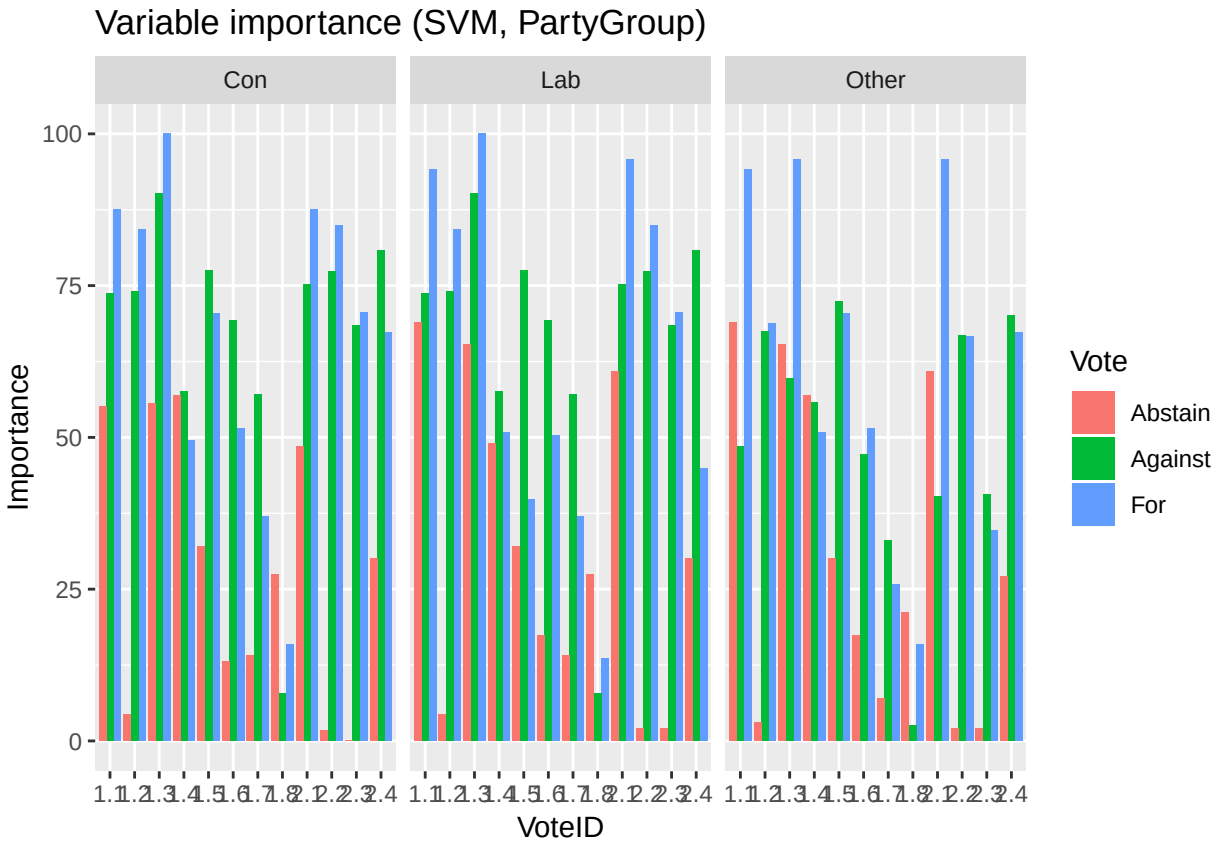
```
## Prevalence                0.4882     0.3858      0.12598
## Detection Rate            0.4882     0.3622      0.09449
## Detection Prevalence      0.5197     0.3780      0.10236
## Balanced Accuracy         0.9692     0.9566      0.87050
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.953
```

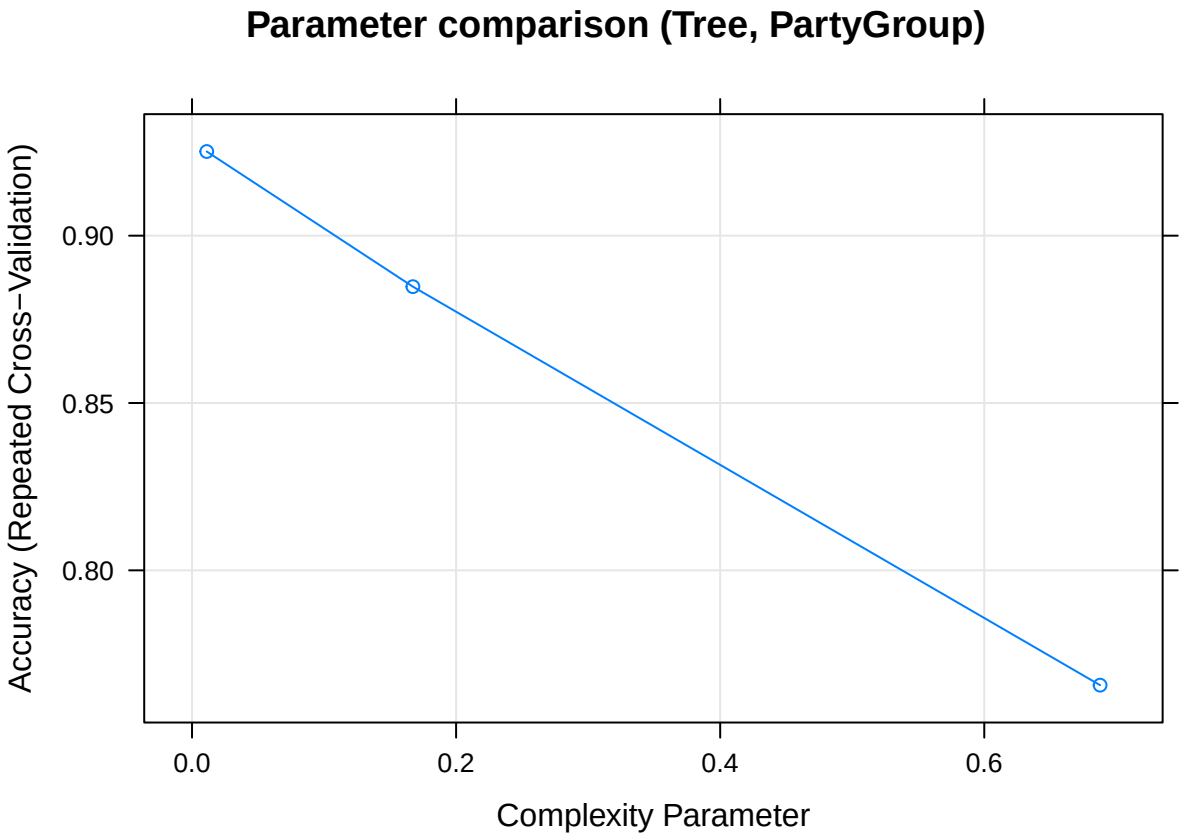


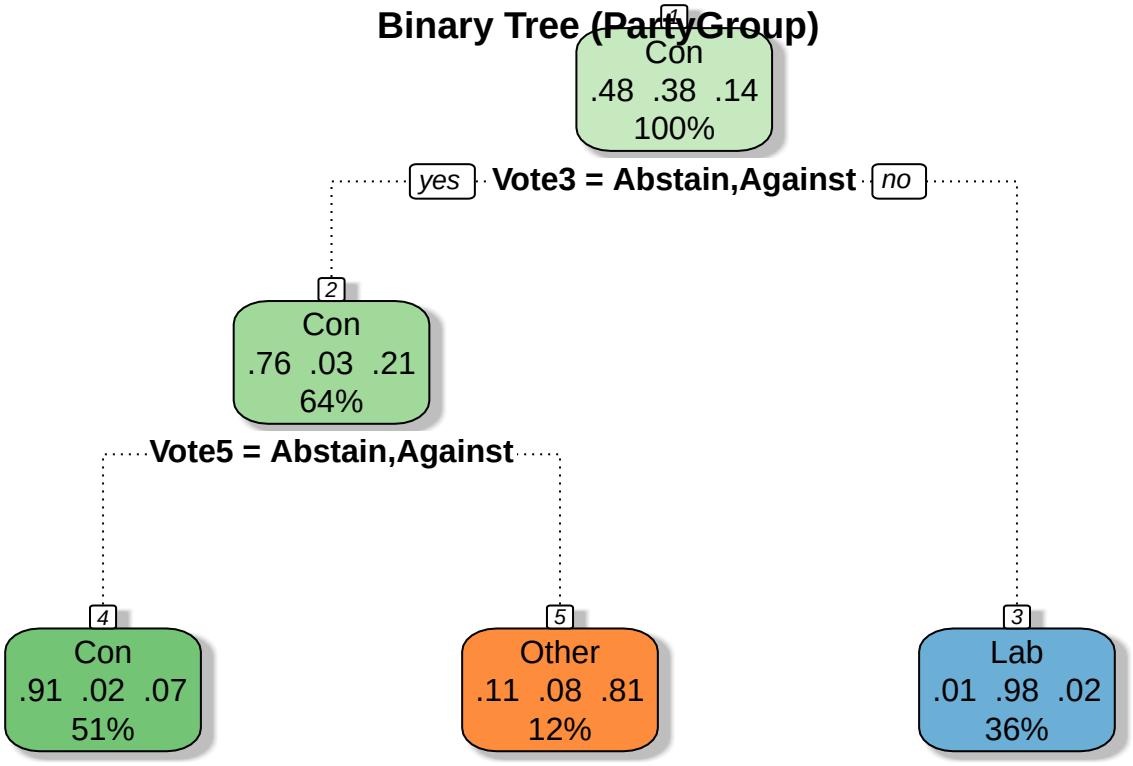



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    62   1     2
##      Lab     0  47     2
##      Other   0   1    12
##
## Overall Statistics
##
##                Accuracy : 0.9528
##                  95% CI : (0.9, 0.9825)
##     No Information Rate : 0.4882
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9197
##
##  Mcnemar's Test P-Value : 0.343
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              1.0000     0.9592      0.75000
## Specificity              0.9538     0.9744      0.99099
## Pos Pred Value           0.9538     0.9592      0.92308
## Neg Pred Value           1.0000     0.9744      0.96491
```

```
## Prevalence                0.4882      0.3858      0.12598
## Detection Rate            0.4882      0.3701      0.09449
## Detection Prevalence      0.5118      0.3858      0.10236
## Balanced Accuracy         0.9769      0.9668      0.87050
```

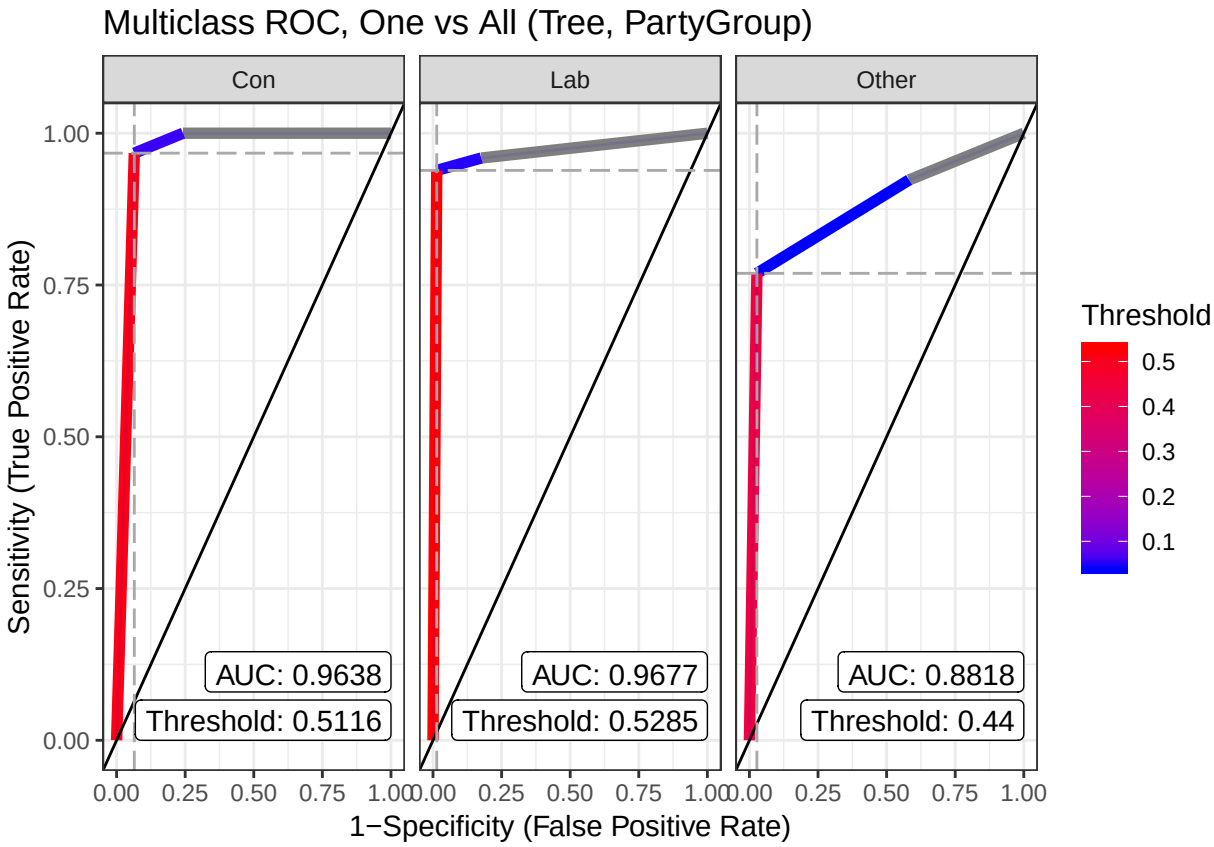


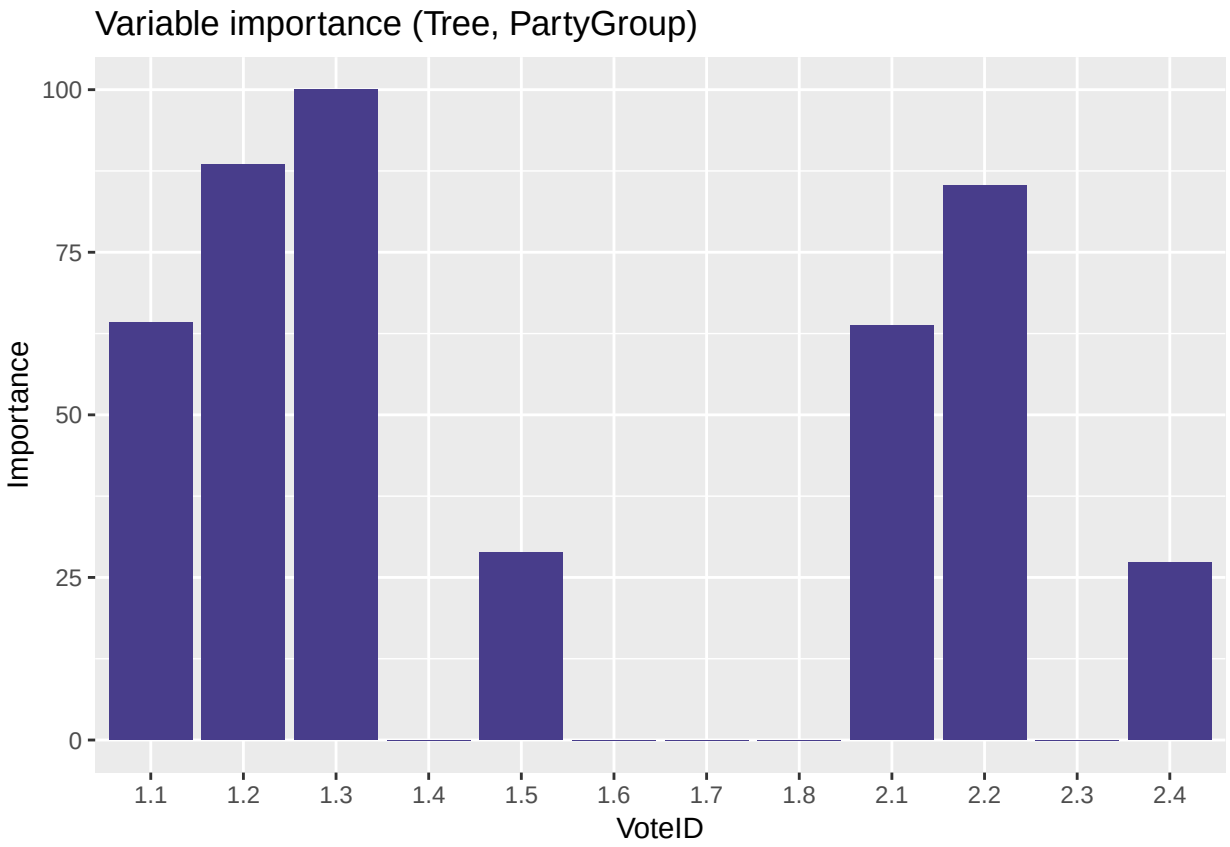

Parameter comparison (Tree, PartyGroup)
Accuracy (Repeated Cross−Validation)
0.90
0.85
0.80
0.0
0.2
0.4
0.6
Complexity Parameter

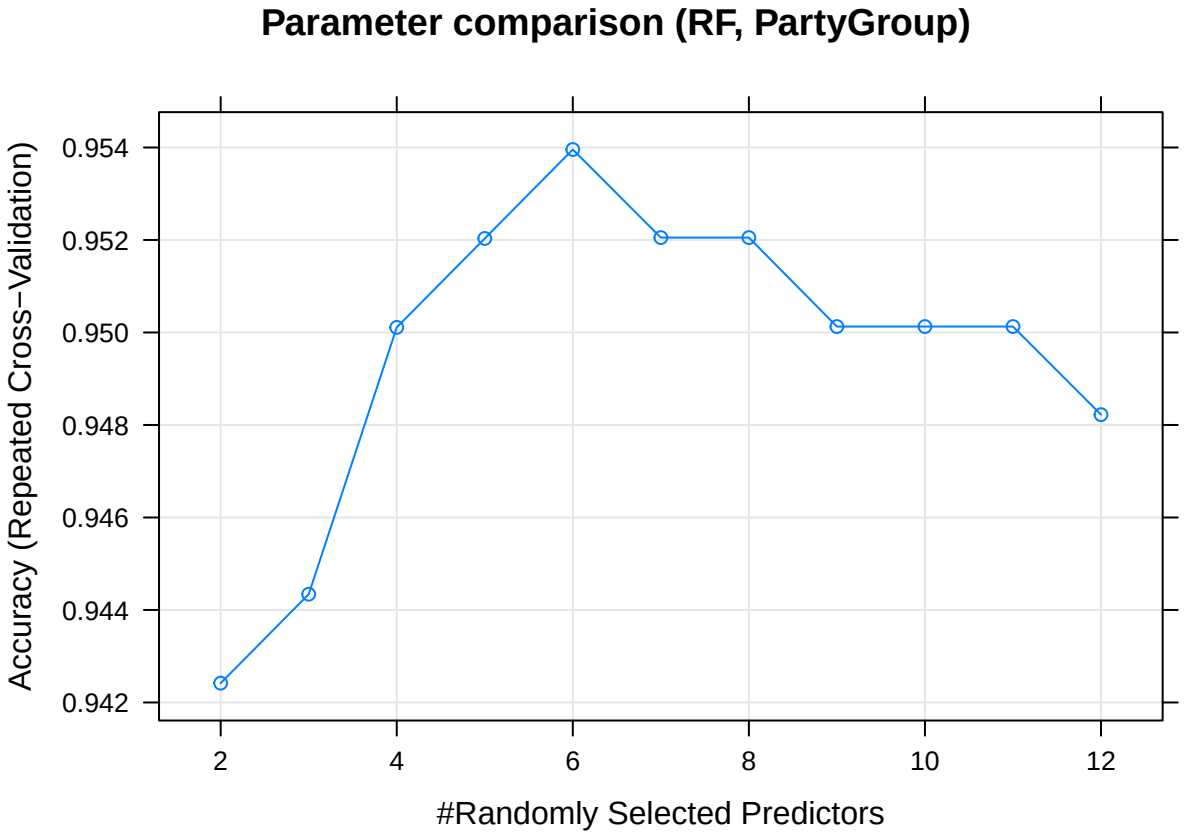


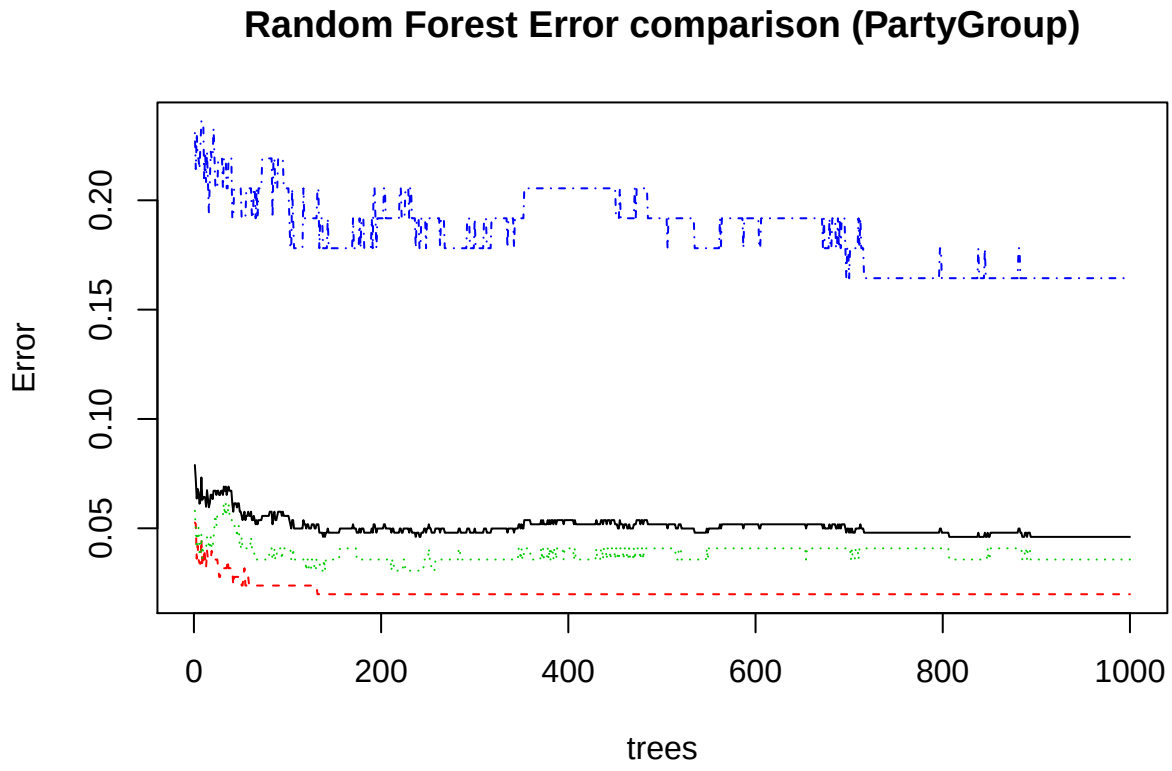

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    59   2     2
##      Lab     0  46     1
##      Other   2   1    10
##
## Overall Statistics
##
##                Accuracy : 0.935
##                  95% CI : (0.8759, 0.9715)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8884
##
##  Mcnemar's Test P-Value : 0.5724
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              0.9672     0.9388       0.7692
## Specificity              0.9355     0.9865       0.9727
## Pos Pred Value           0.9365     0.9787       0.7692
## Neg Pred Value           0.9667     0.9605       0.9727
```

```
## Prevalence                0.4959     0.3984       0.1057
## Detection Rate            0.4797     0.3740       0.0813
## Detection Prevalence      0.5122     0.3821       0.1057
## Balanced Accuracy         0.9513     0.9626       0.8710
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.9263
```



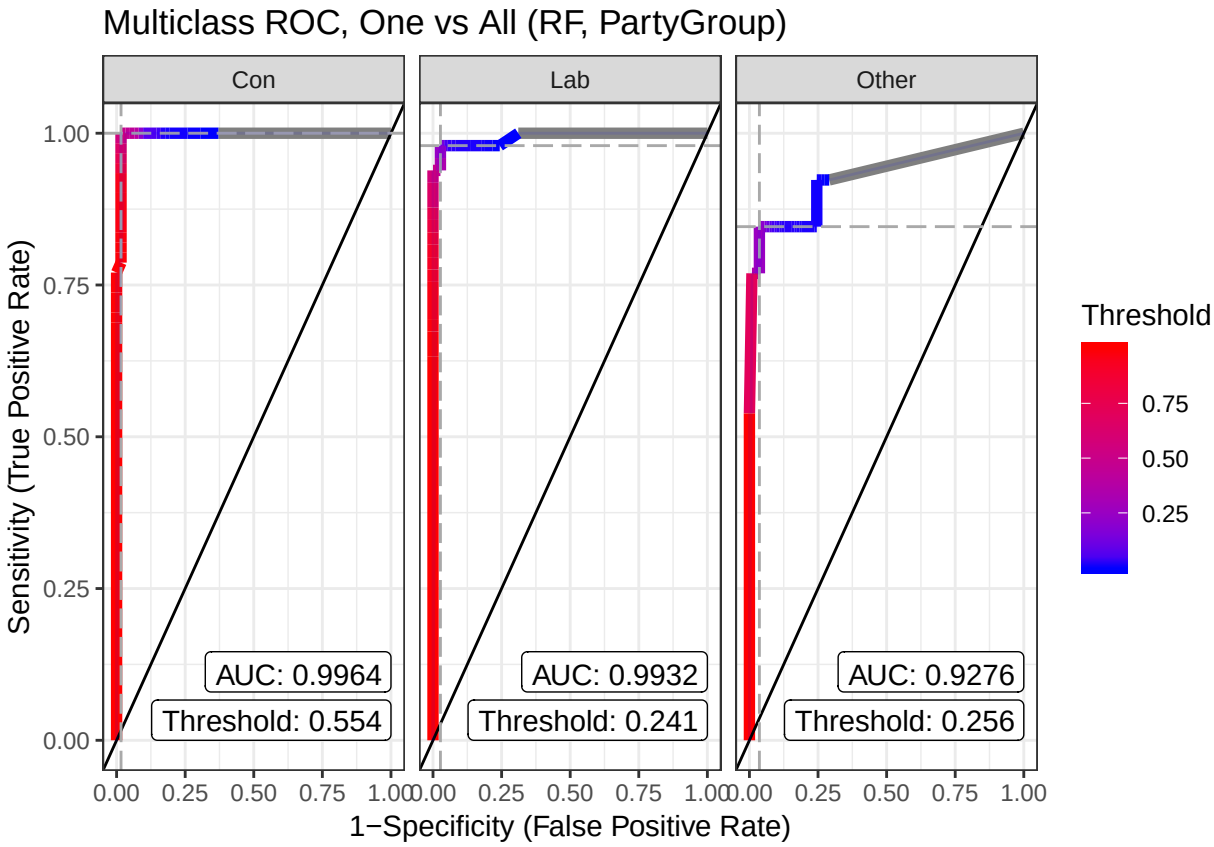

Variable importance (Tree, PartyGroup)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

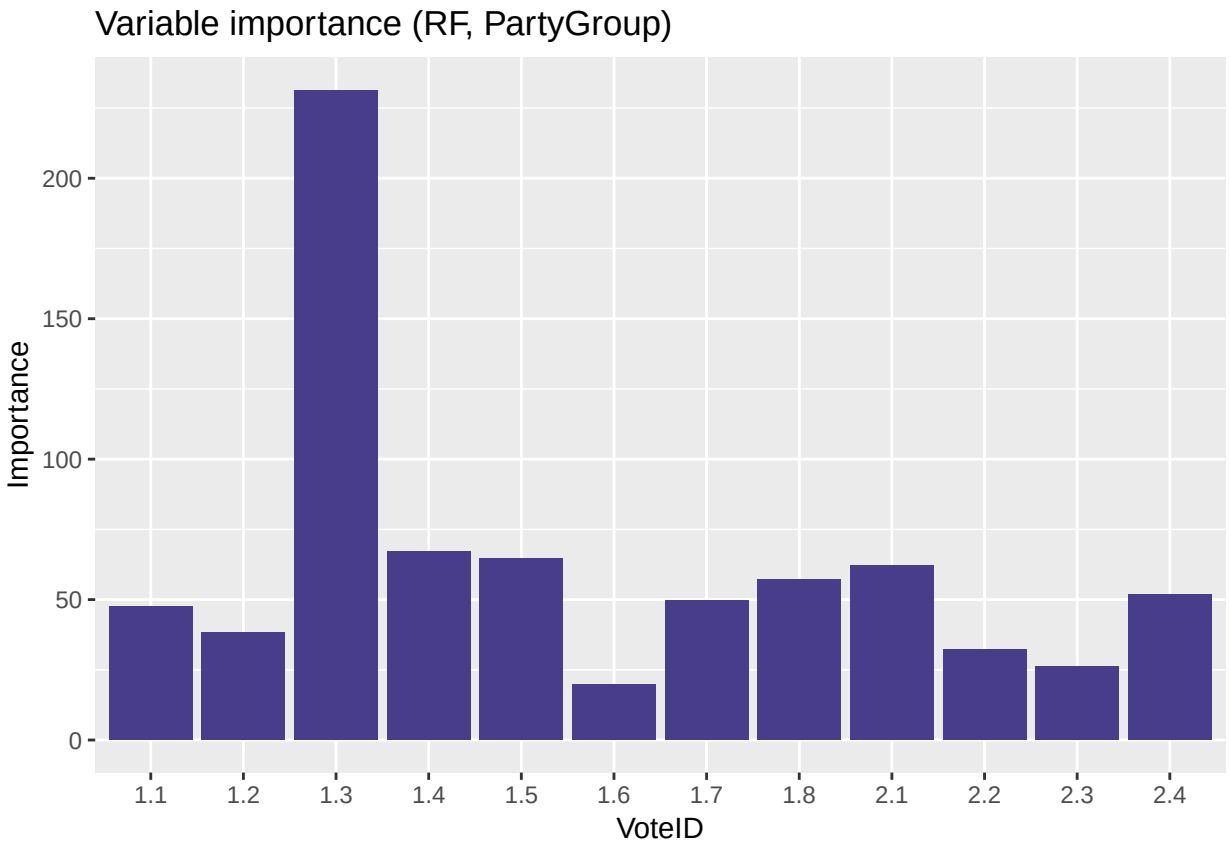

Parameter comparison (RF, PartyGroup)
Accuracy (Repeated Cross−Validation)
0.954
0.952
0.950
0.948
0.946
0.944
0.942
2
4
6
8
10
12
#Randomly Selected Predictors

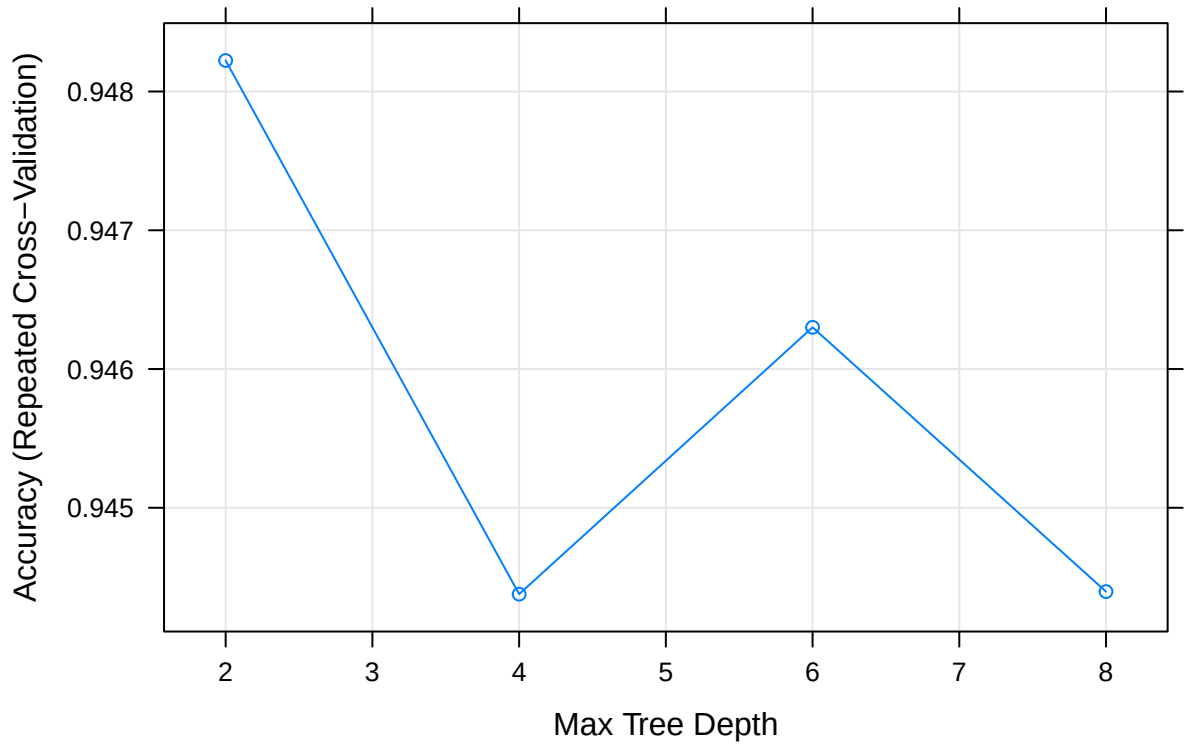



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    61   2     1
##      Lab     0  46     2
##      Other   0   1    10
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9155
##
##  Mcnemar's Test P-Value : 0.343
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              1.0000     0.9388      0.76923
## Specificity              0.9516     0.9730      0.99091
## Pos Pred Value           0.9531     0.9583      0.90909
## Neg Pred Value           1.0000     0.9600      0.97321
```

```
## Prevalence                0.4959     0.3984      0.10569
## Detection Rate            0.4959     0.3740      0.08130
## Detection Prevalence      0.5203     0.3902      0.08943
## Balanced Accuracy         0.9758     0.9559      0.88007
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.9696
```

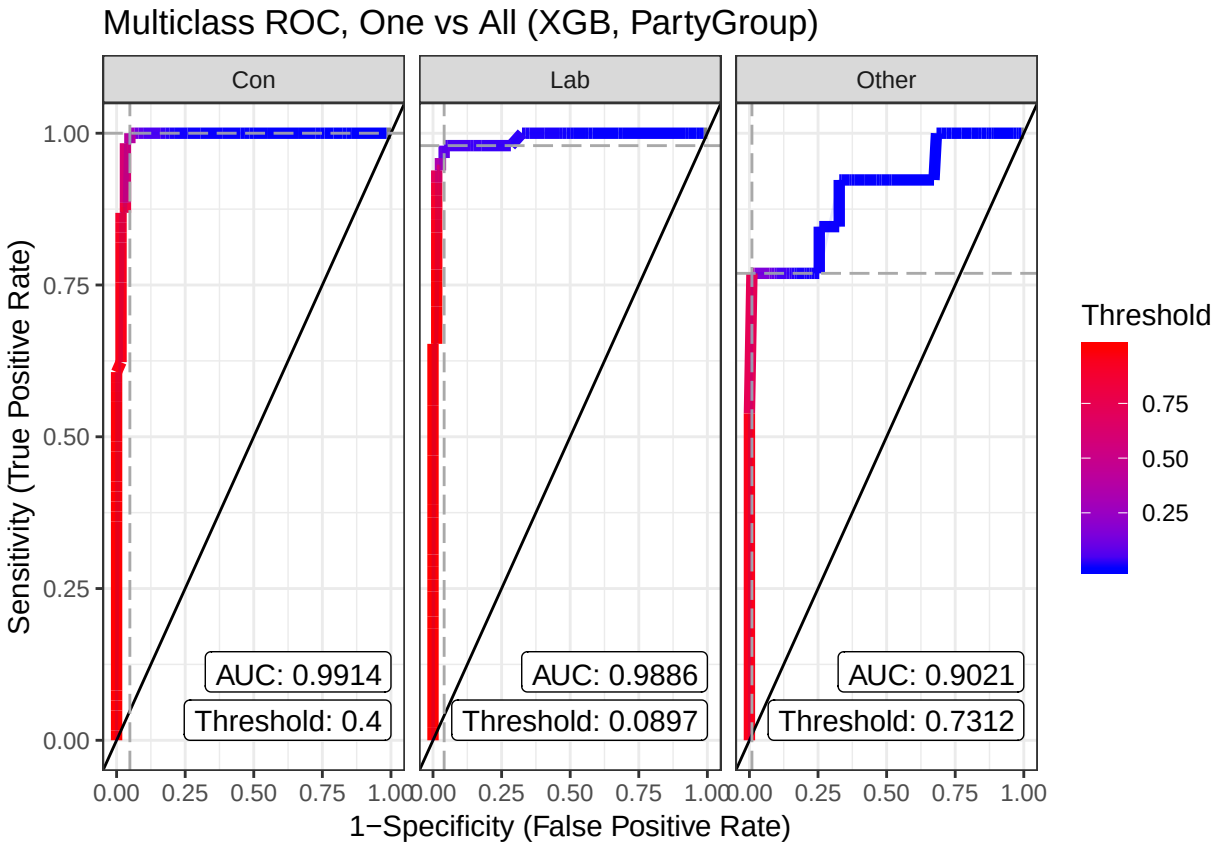


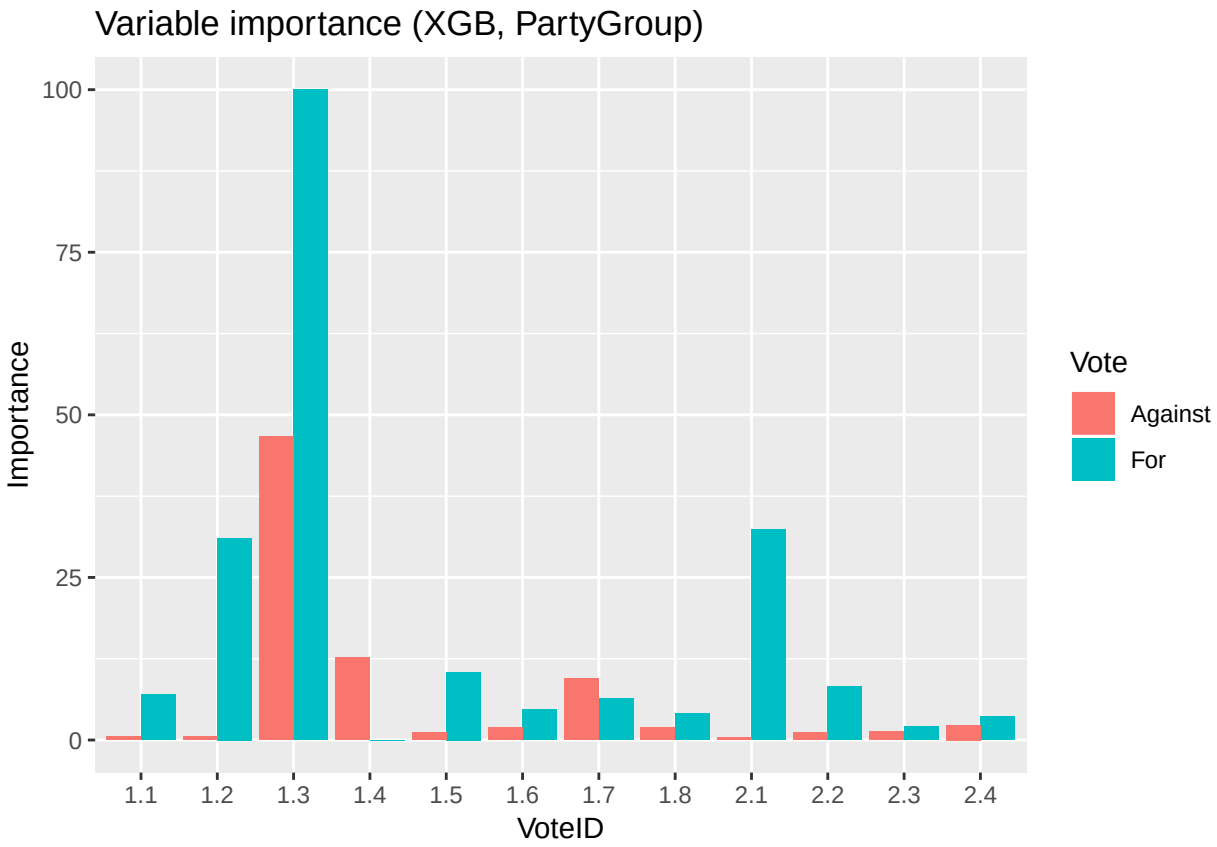

Variable importance (RF, PartyGroup)
Importance
200
150
100
50
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

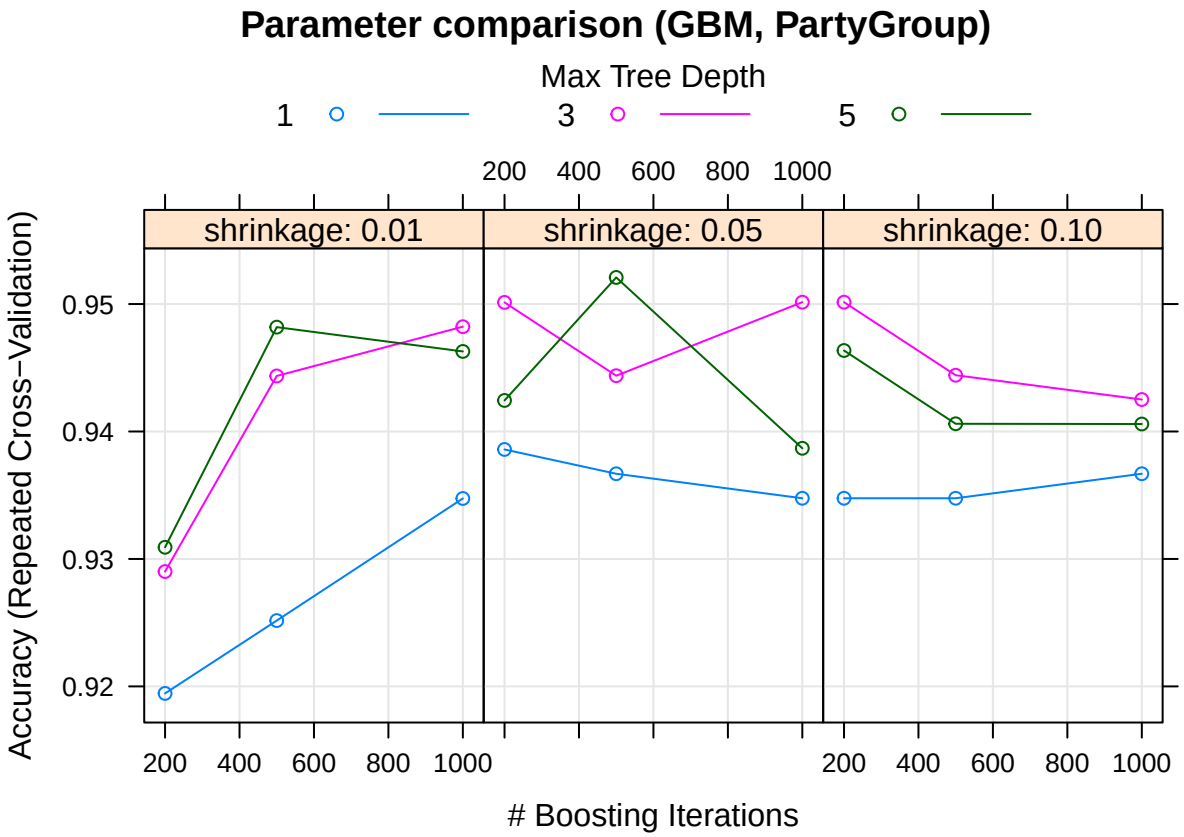



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    60   1     1
##      Lab     0  47     2
##      Other   1   1    10
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.916
##
##  Mcnemar's Test P-Value : 0.7212
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              0.9836     0.9592      0.76923
## Specificity              0.9677     0.9730      0.98182
## Pos Pred Value           0.9677     0.9592      0.83333
## Neg Pred Value           0.9836     0.9730      0.97297
```

```
## Prevalence                0.4959     0.3984      0.10569
## Detection Rate            0.4878     0.3821      0.08130
## Detection Prevalence      0.5041     0.3984      0.09756
## Balanced Accuracy         0.9757     0.9661      0.87552
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.9554
```

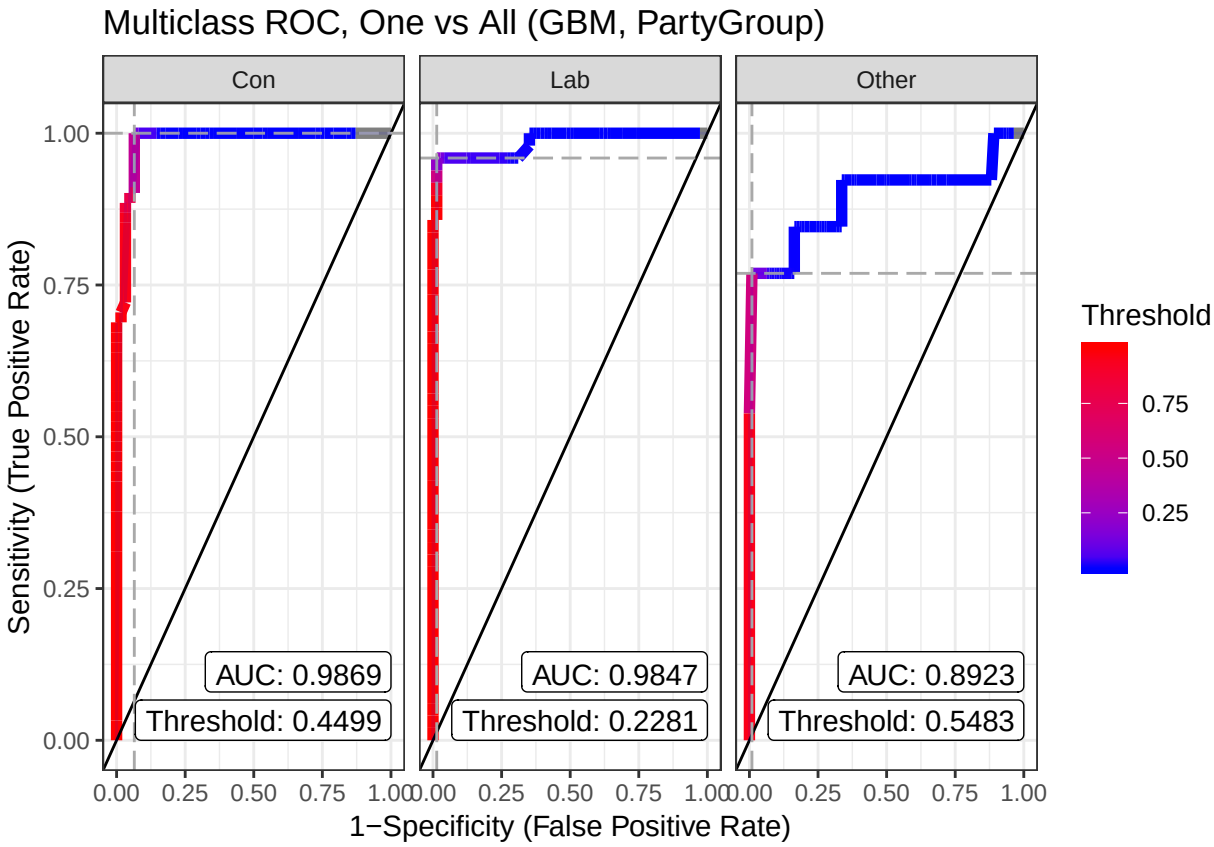


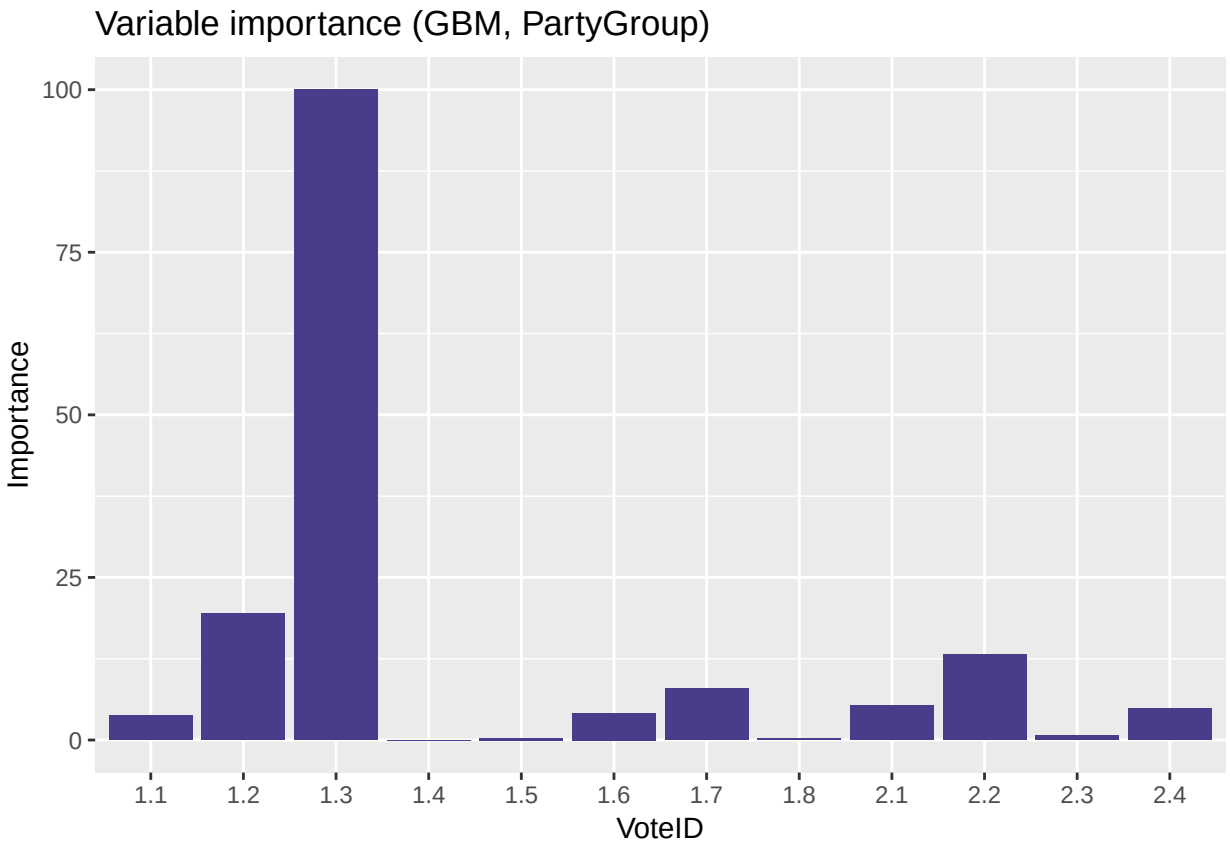

Variable importance (XGB, PartyGroup)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For

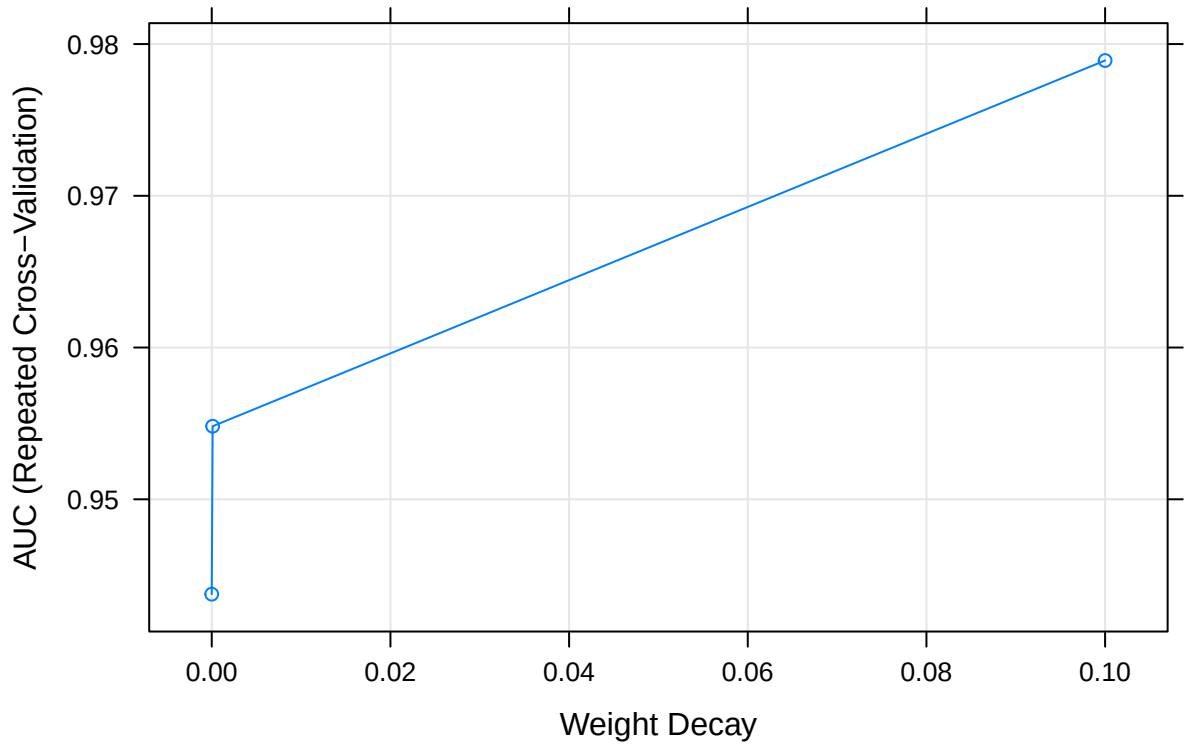



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    61   2     2
##      Lab     0  46     1
##      Other   0   1    10
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9153
##
##  Mcnemar's Test P-Value : 0.2615
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              1.0000     0.9388      0.76923
## Specificity              0.9355     0.9865      0.99091
## Pos Pred Value           0.9385     0.9787      0.90909
## Neg Pred Value           1.0000     0.9605      0.97321
```

```
## Prevalence                0.4959     0.3984      0.10569
## Detection Rate            0.4959     0.3740      0.08130
## Detection Prevalence      0.5285     0.3821      0.08943
## Balanced Accuracy         0.9677     0.9626      0.88007
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.9513
```

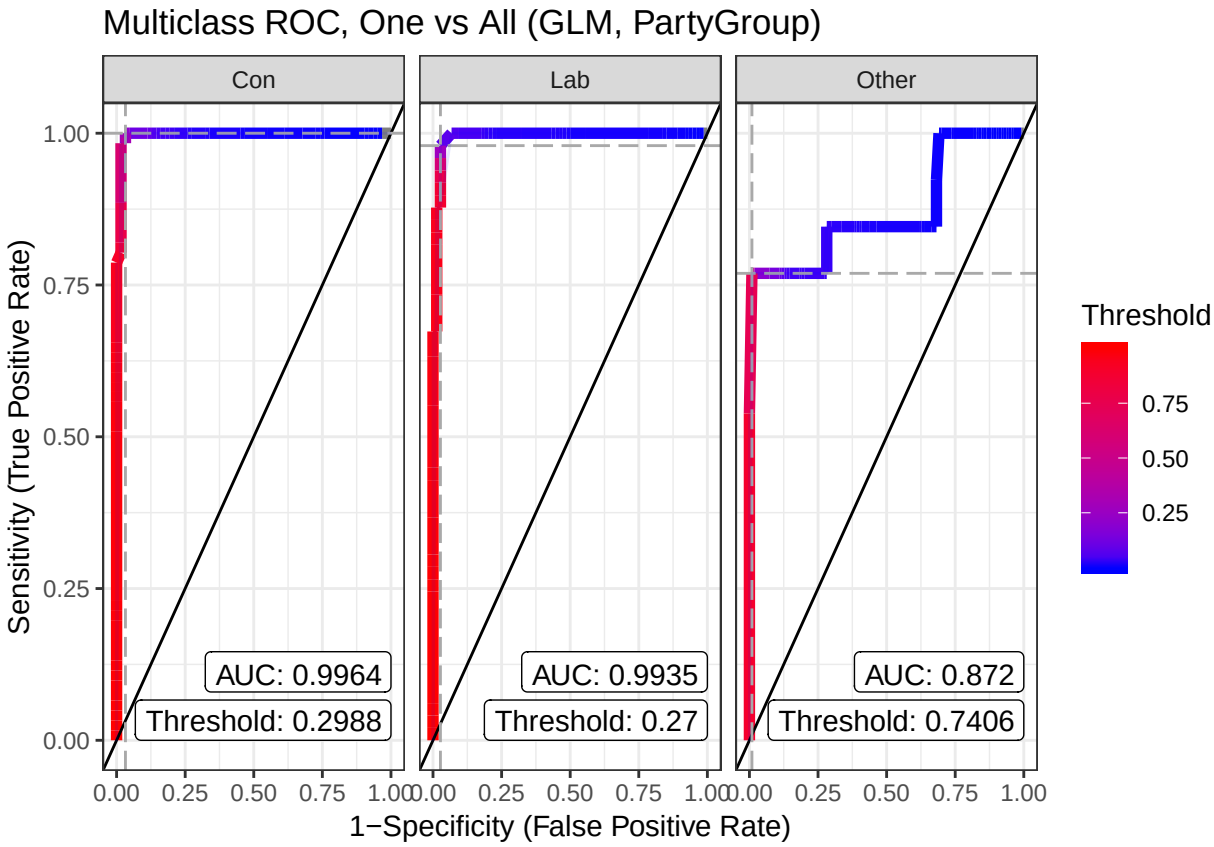



Variable importance (GBM, PartyGroup)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID



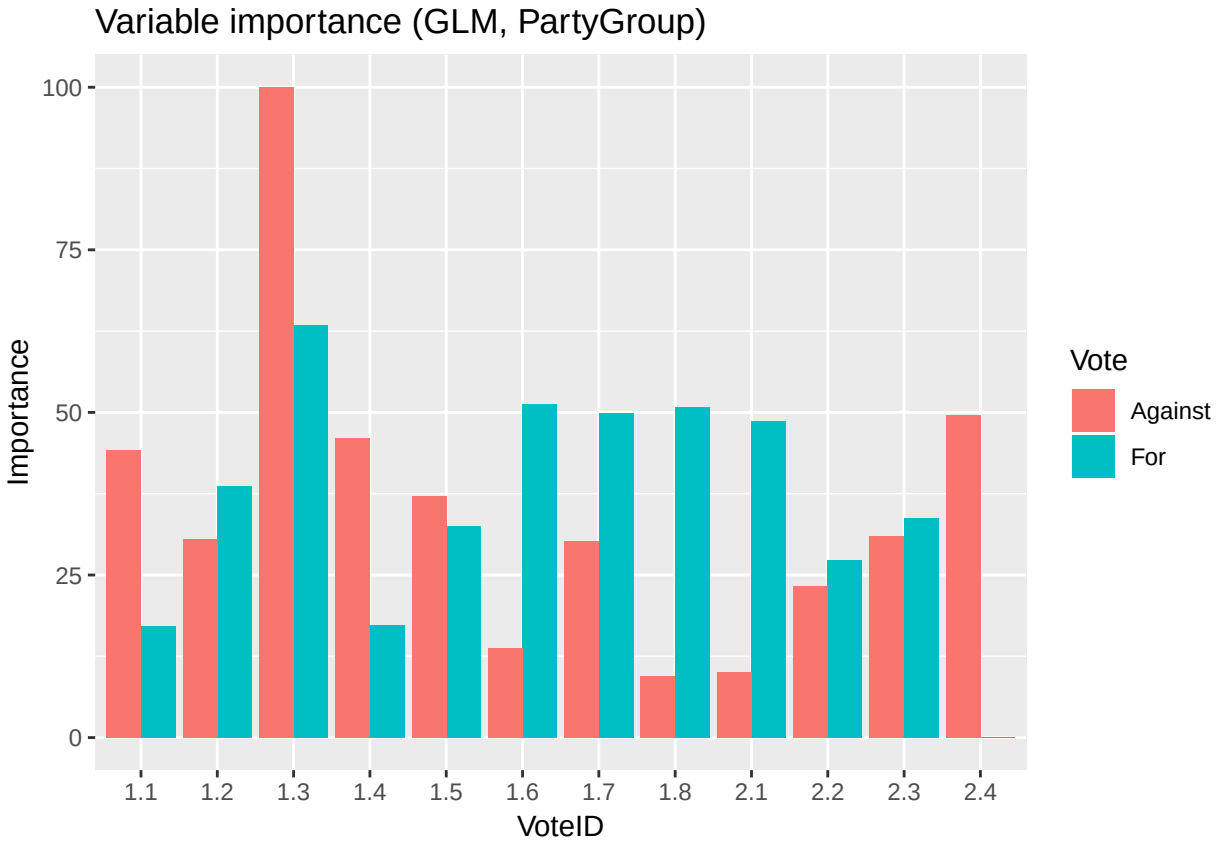

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    60   1     1
##      Lab     0  47     2
##      Other   1   1    10
##
## Overall Statistics
##
##                Accuracy : 0.9512
##                  95% CI : (0.8968, 0.9819)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.916
##
##  Mcnemar's Test P-Value : 0.7212
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              0.9836     0.9592      0.76923
## Specificity              0.9677     0.9730      0.98182
## Pos Pred Value           0.9677     0.9592      0.83333
## Neg Pred Value           0.9836     0.9730      0.97297
```

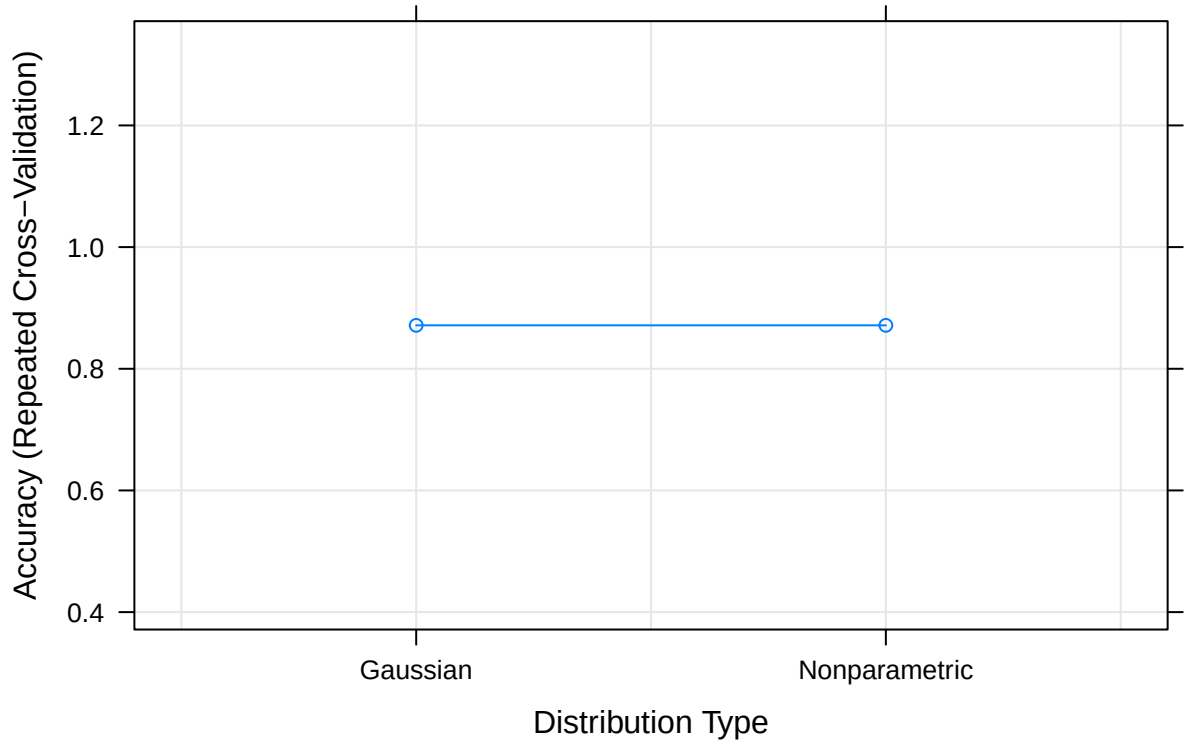
```
## Prevalence                0.4959     0.3984      0.10569
## Detection Rate            0.4878     0.3821      0.08130
## Detection Prevalence      0.5041     0.3984      0.09756
## Balanced Accuracy         0.9757     0.9661      0.87552
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.949
```




```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

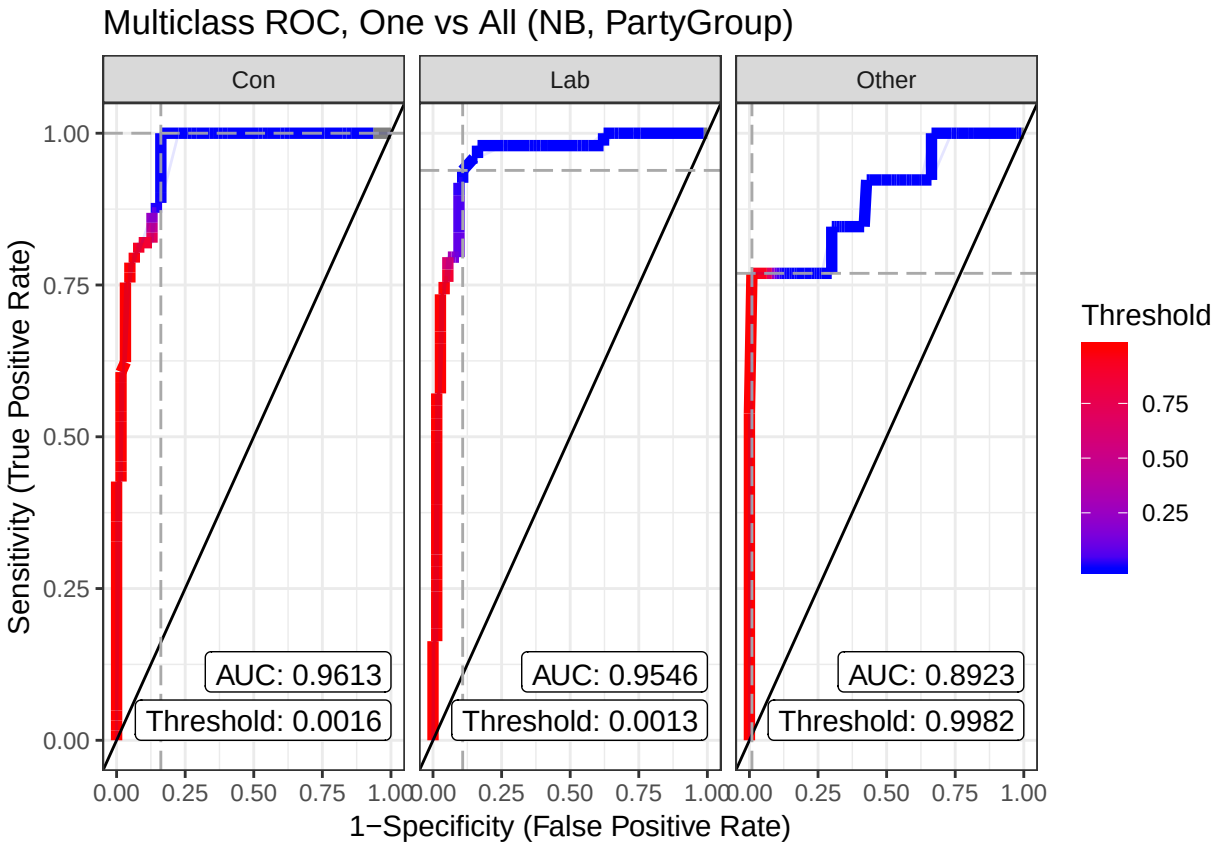
```
## observation 25
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73
```

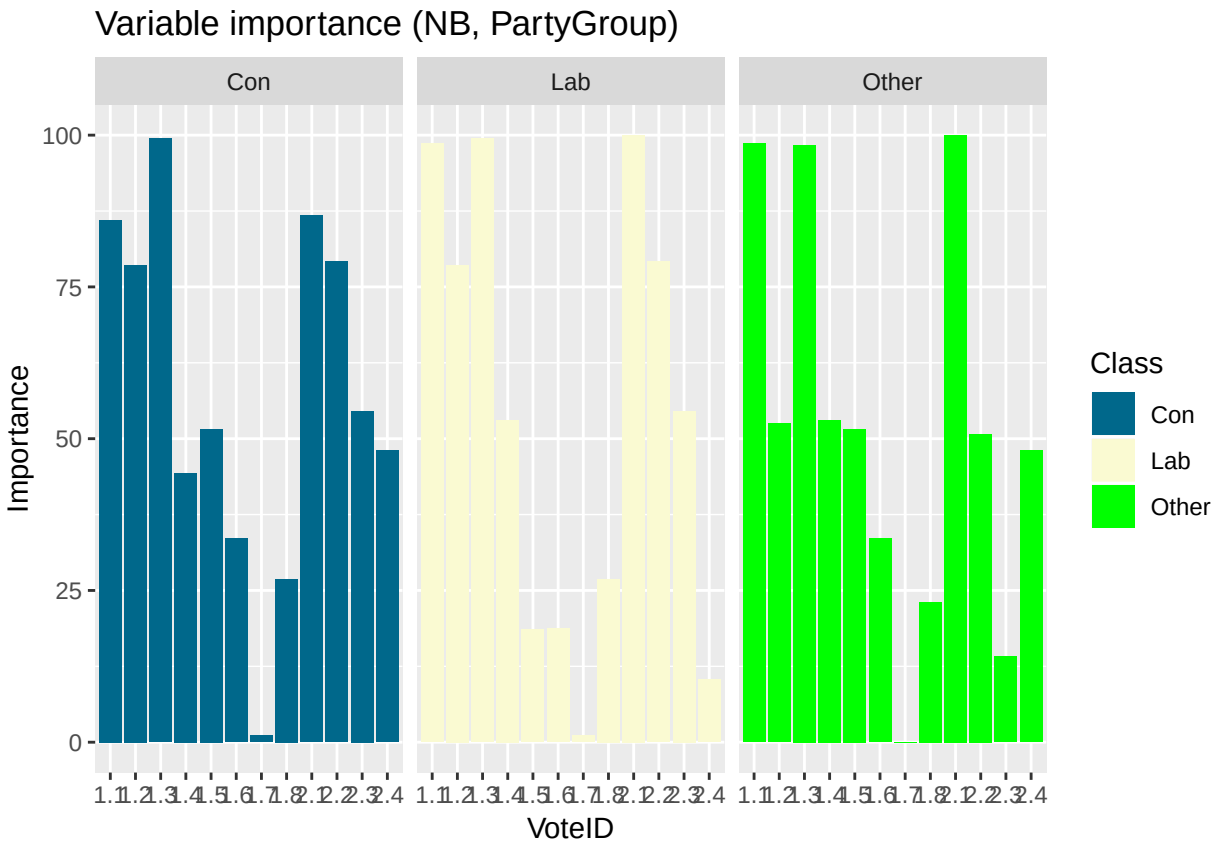


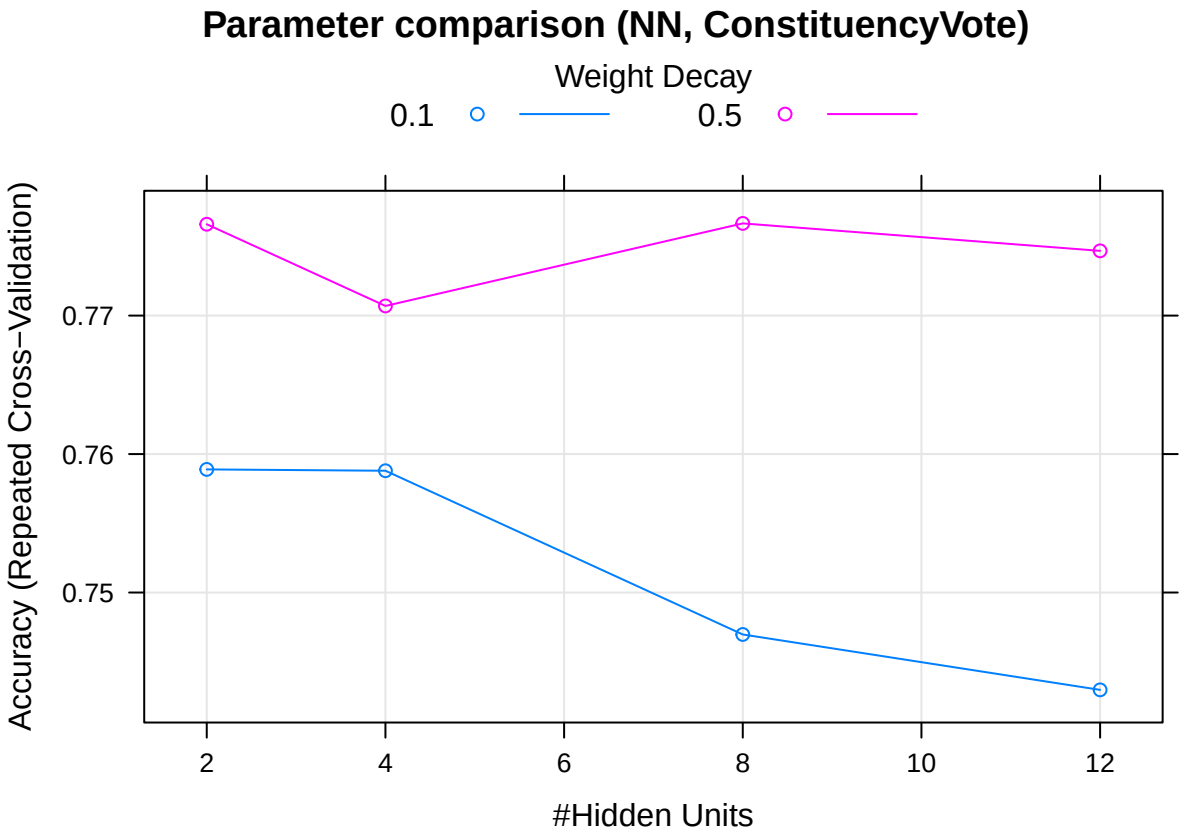



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Con Lab Other
##      Con    52   7     1
##      Lab     2  39     2
##      Other   7   3    10
##
## Overall Statistics
##
##                Accuracy : 0.8211
##                  95% CI : (0.7418, 0.8844)
##     No Information Rate : 0.4959
##     P-Value [Acc > NIR] : 7.143e-14
##
##                   Kappa : 0.7027
##
##  Mcnemar's Test P-Value : 0.05813
##
## Statistics by Class:
##
##                      Class: Con Class: Lab Class: Other
## Sensitivity              0.8525     0.7959       0.7692
## Specificity              0.8710     0.9459       0.9091
## Pos Pred Value           0.8667     0.9070       0.5000
## Neg Pred Value           0.8571     0.8750       0.9709
```

```
## Prevalence                0.4959     0.3984       0.1057
## Detection Rate            0.4228     0.3171       0.0813
## Detection Prevalence      0.4878     0.3496       0.1626
## Balanced Accuracy         0.8617     0.8709       0.8392
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Con, Lab, Other.
## Multi-class area under the curve: 0.9264
```

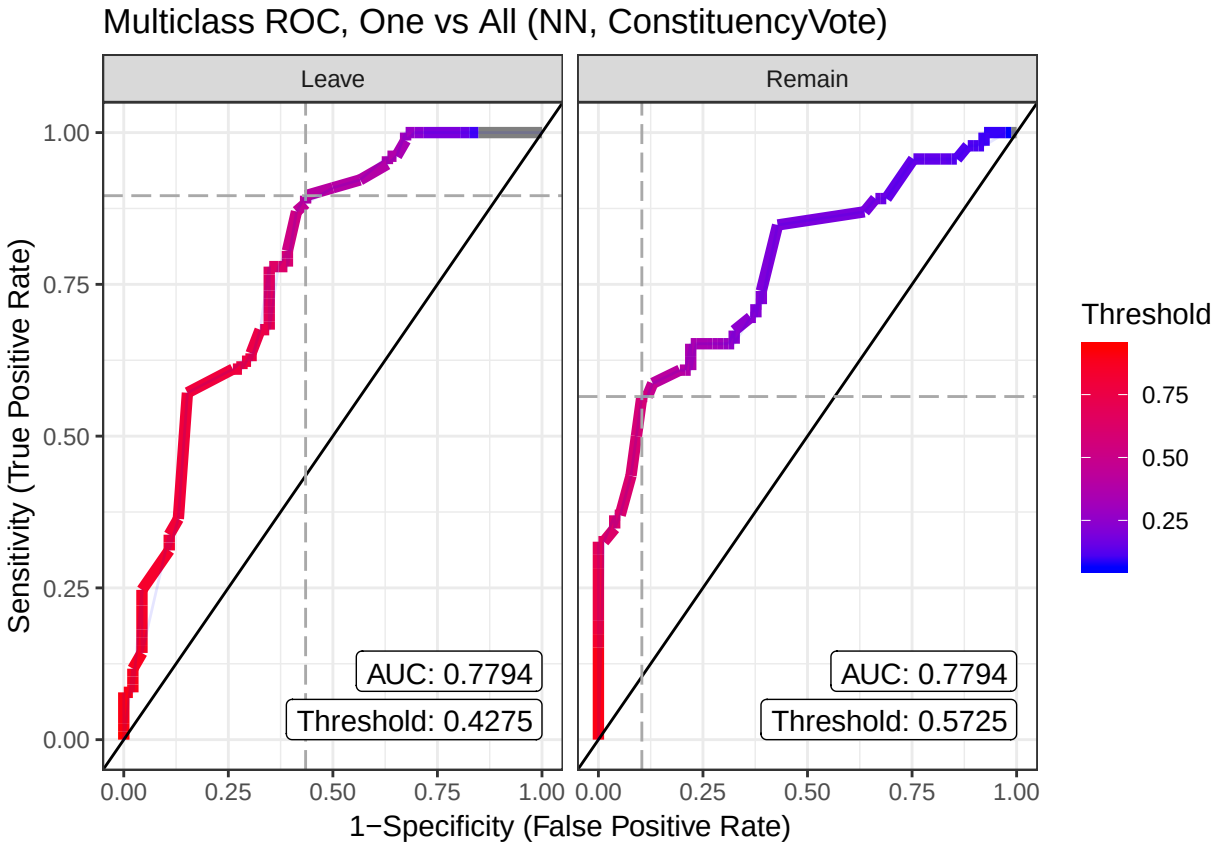


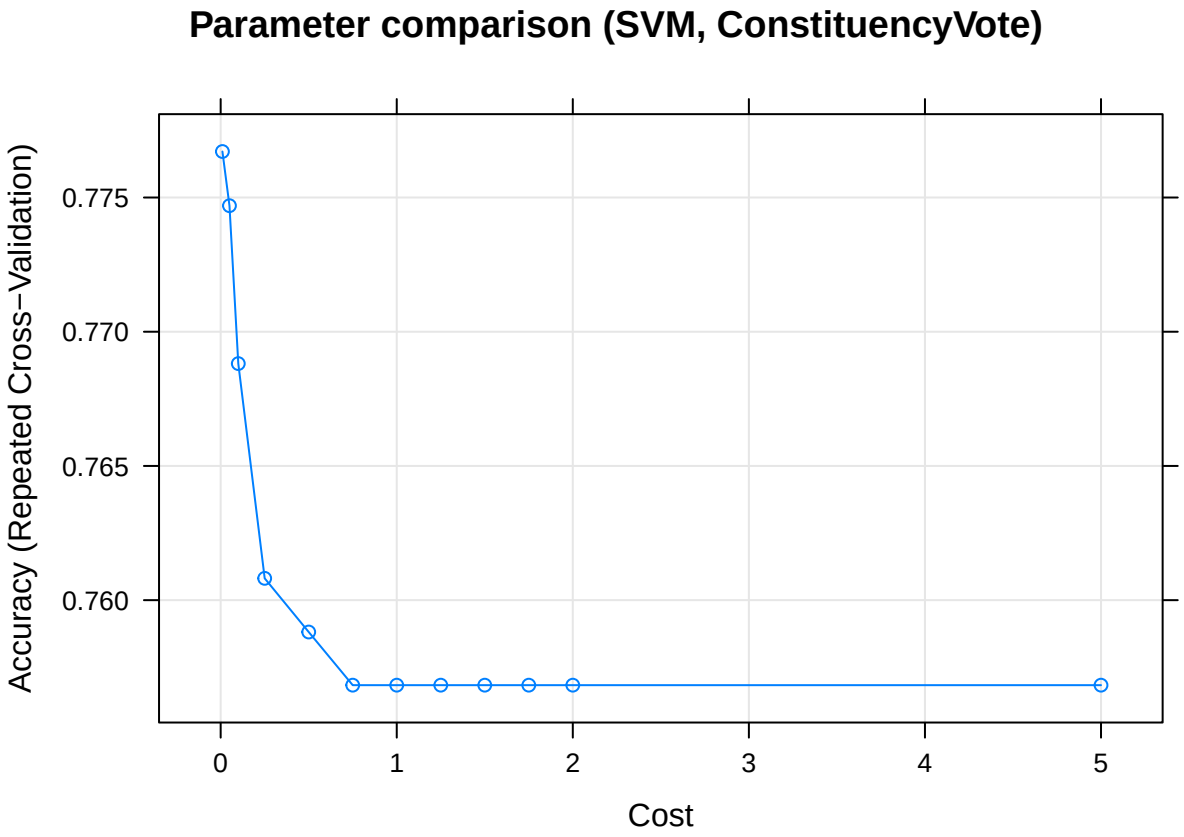

Variable importance (NB, PartyGroup)
Con
Lab
Other
Importance
100
75
50
25
0
VoteID
Class
Con
Lab
Other




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     67     19
##     Remain    10     27
##
##                Accuracy : 0.7642
##                  95% CI : (0.6793, 0.8361)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.0007699
##
##                   Kappa : 0.4758
##
##  Mcnemar's Test P-Value : 0.1373948
##
##             Sensitivity : 0.8701
##             Specificity : 0.5870
##          Pos Pred Value : 0.7791
##          Neg Pred Value : 0.7297
##              Prevalence : 0.6260
##          Detection Rate : 0.5447
##    Detection Prevalence : 0.6992
##       Balanced Accuracy : 0.7285
##
##        'Positive' Class : Leave
```

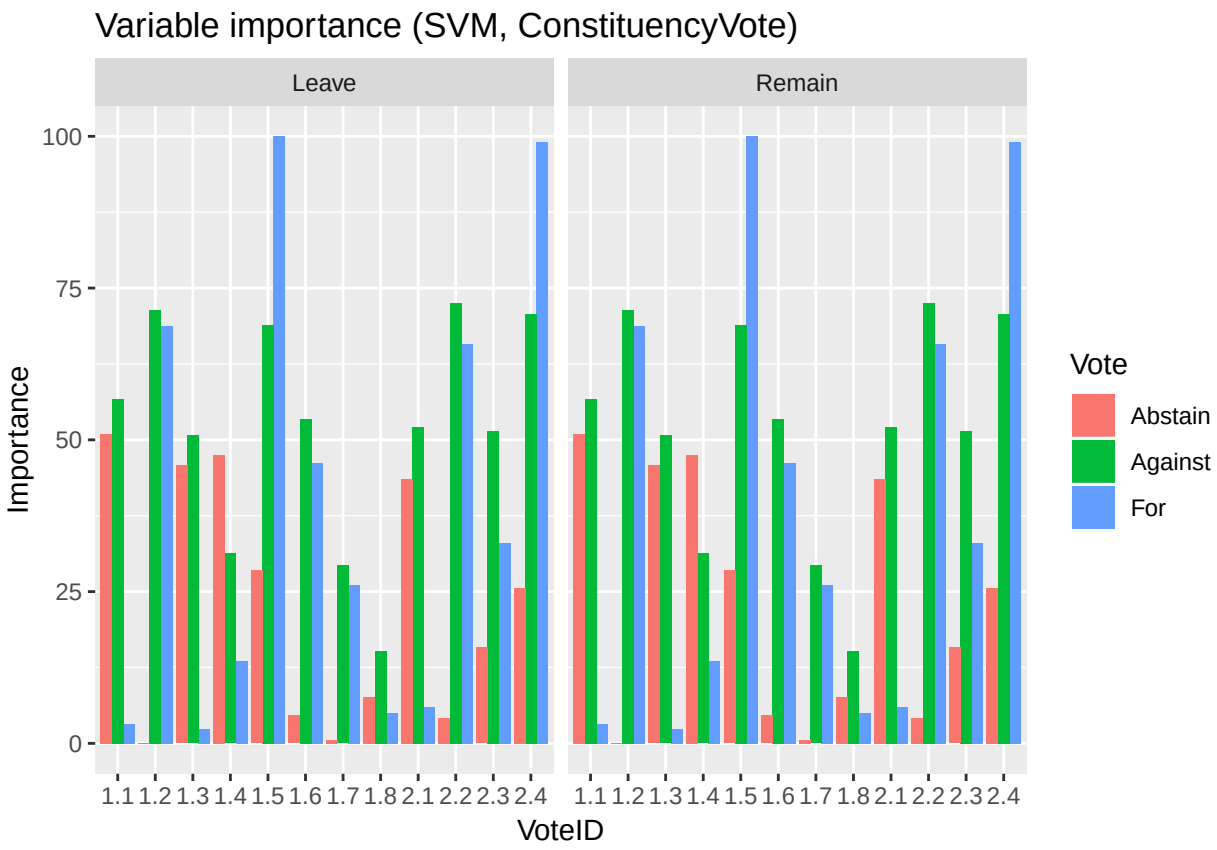
```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.7794
```

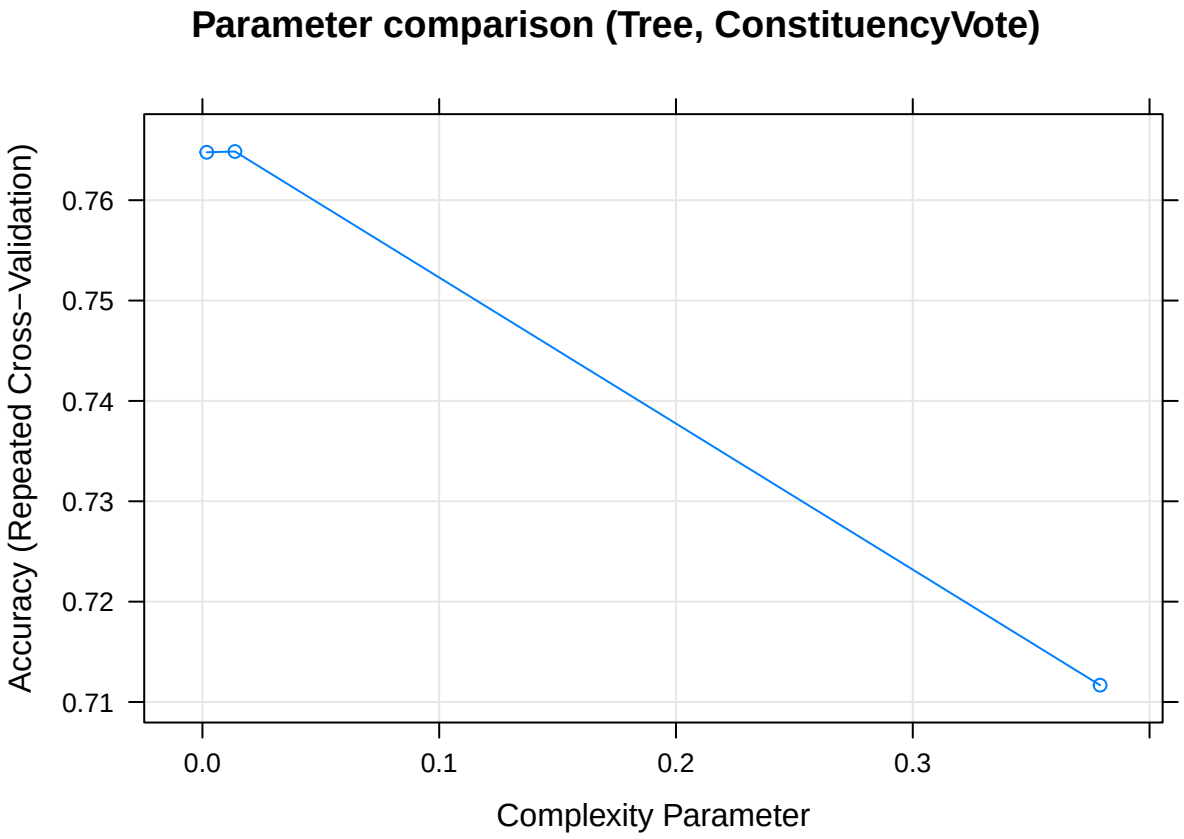


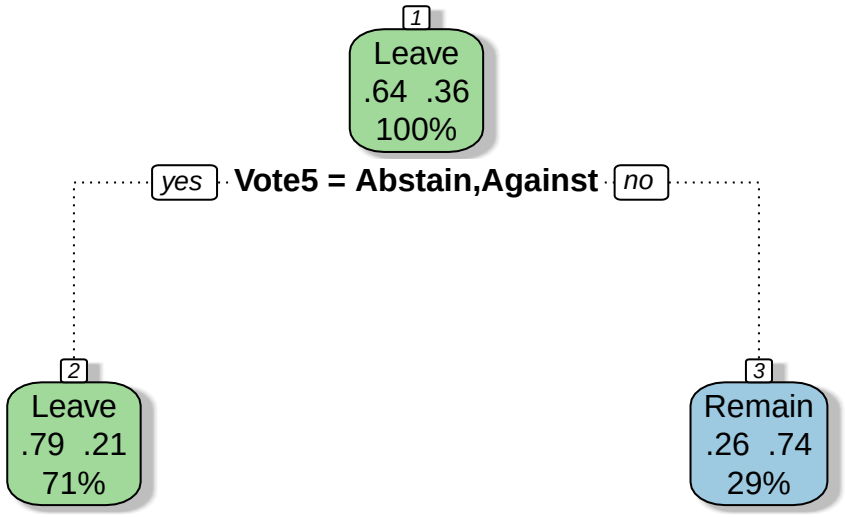



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     68     20
##     Remain     9     26
##
##                Accuracy : 0.7642
##                  95% CI : (0.6793, 0.8361)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.0007699
##
##                   Kappa : 0.471
##
##  Mcnemar's Test P-Value : 0.0633178
##
##             Sensitivity : 0.8831
##             Specificity : 0.5652
##          Pos Pred Value : 0.7727
##          Neg Pred Value : 0.7429
##              Prevalence : 0.6260
##          Detection Rate : 0.5528
##    Detection Prevalence : 0.7154
##       Balanced Accuracy : 0.7242
##
##        'Positive' Class : Leave
```

##

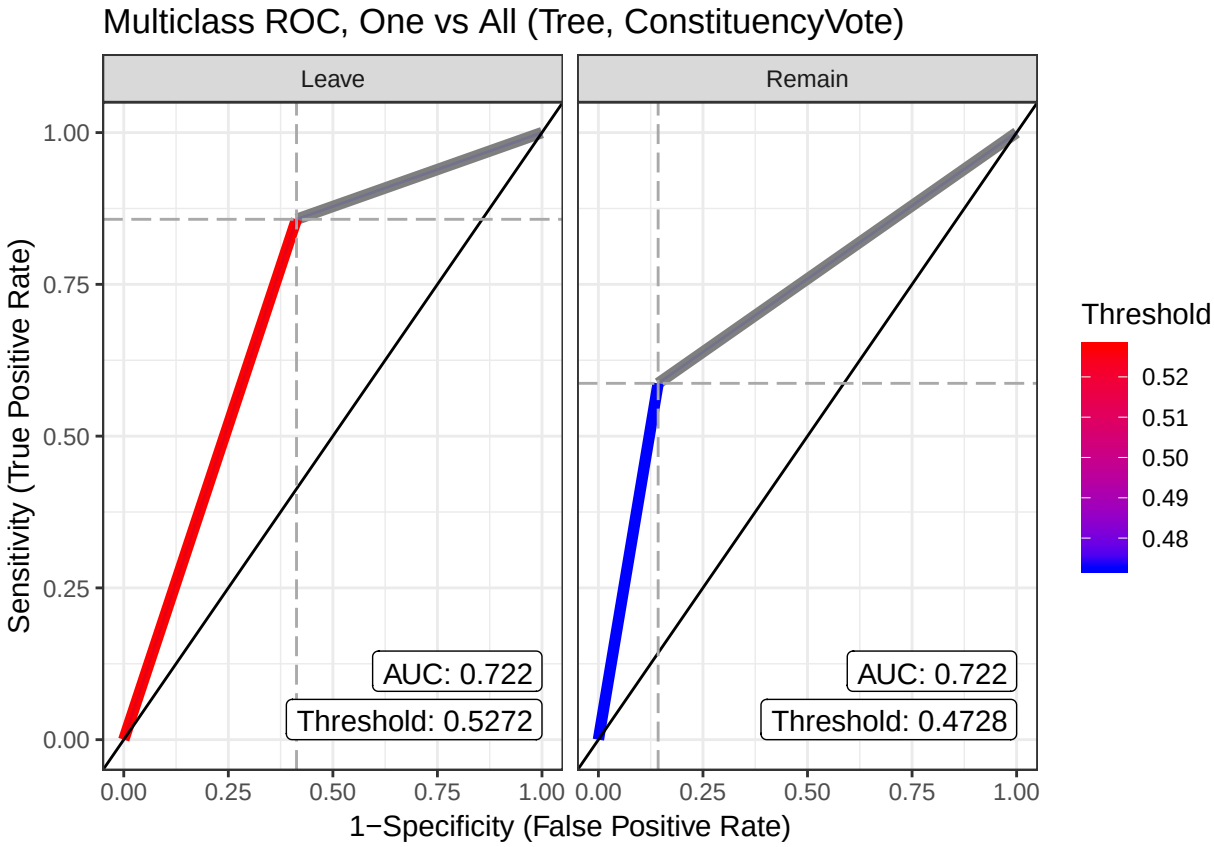

Variable importance (SVM, ConstituencyVote)
Leave
Remain
Importance
100
75
50
25
0
1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 2.1 2.2 2.3 2.4
VoteID
Vote
Abstain
Against
For

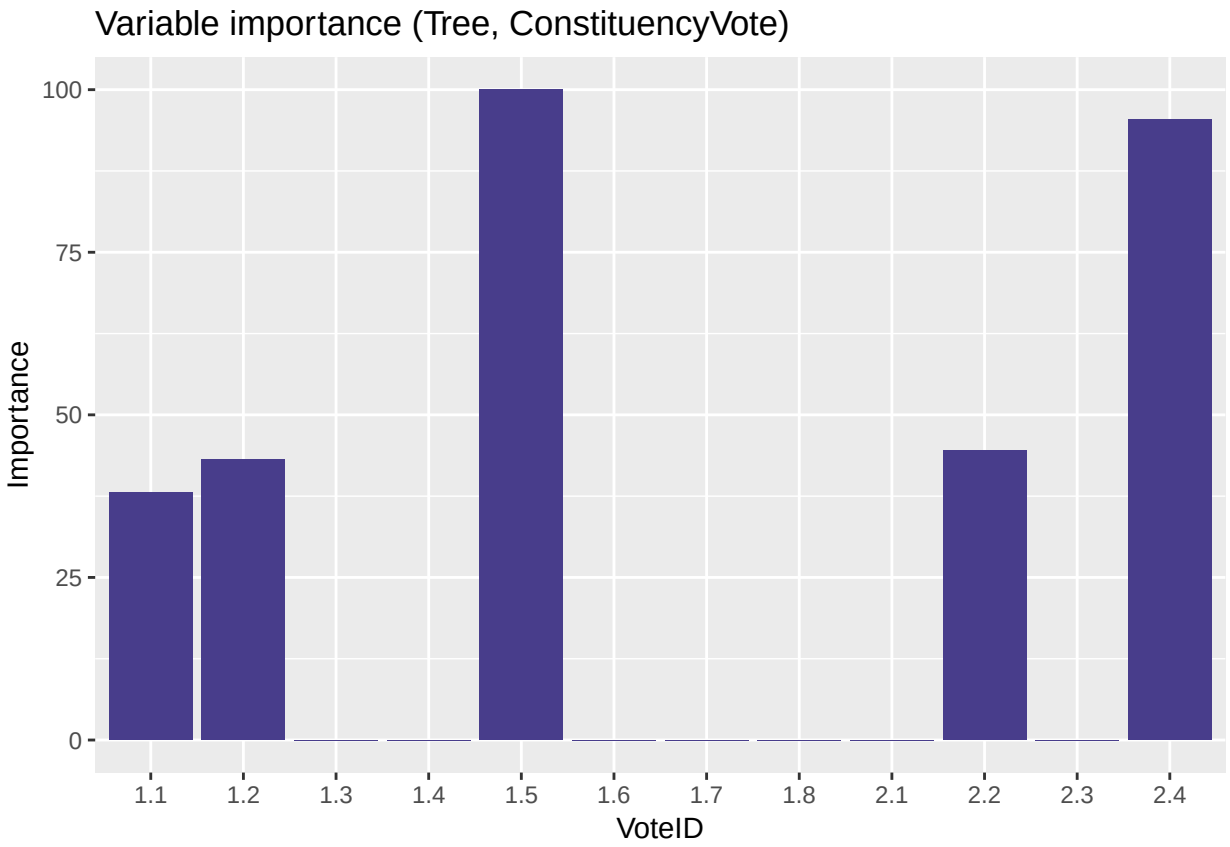

Parameter comparison (Tree, ConstituencyVote)
Accuracy (Repeated Cross−Validation)
0.76
0.75
0.74
0.73
0.72
0.71
0.0
0.1
0.2
0.3
Complexity Parameter

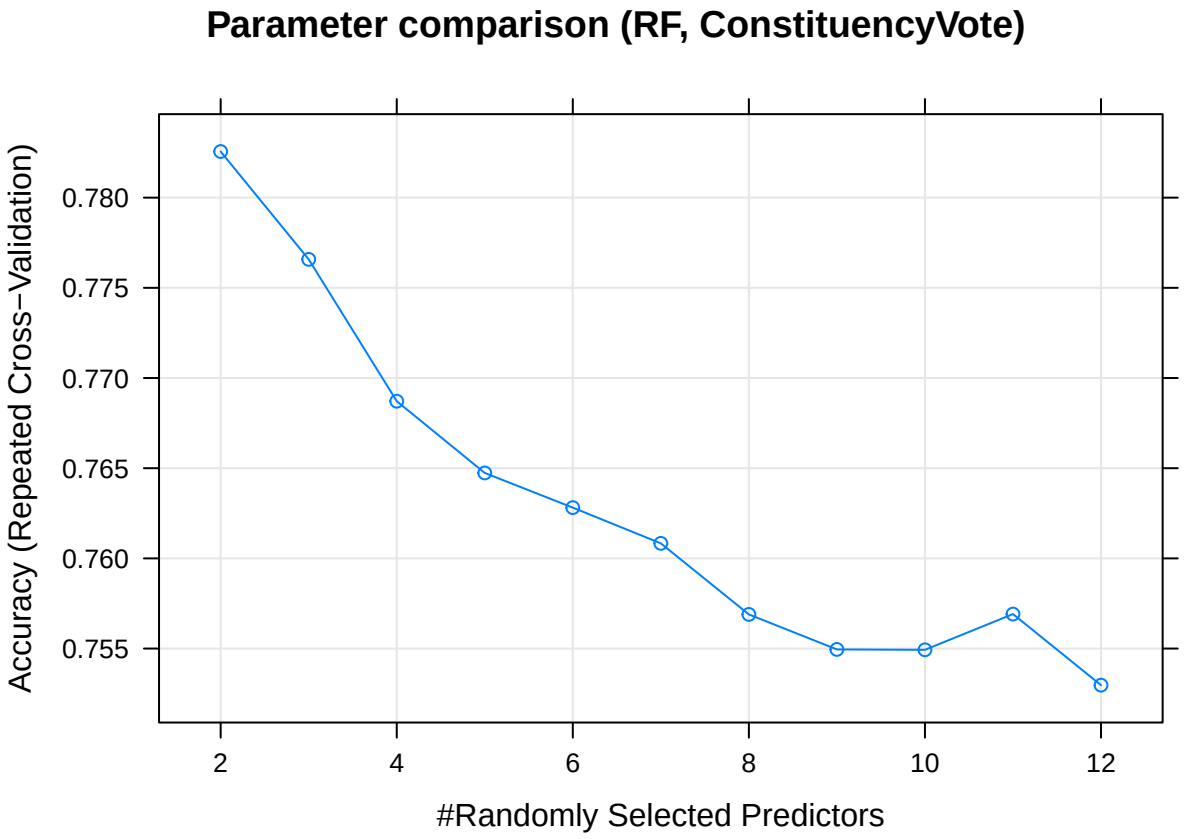


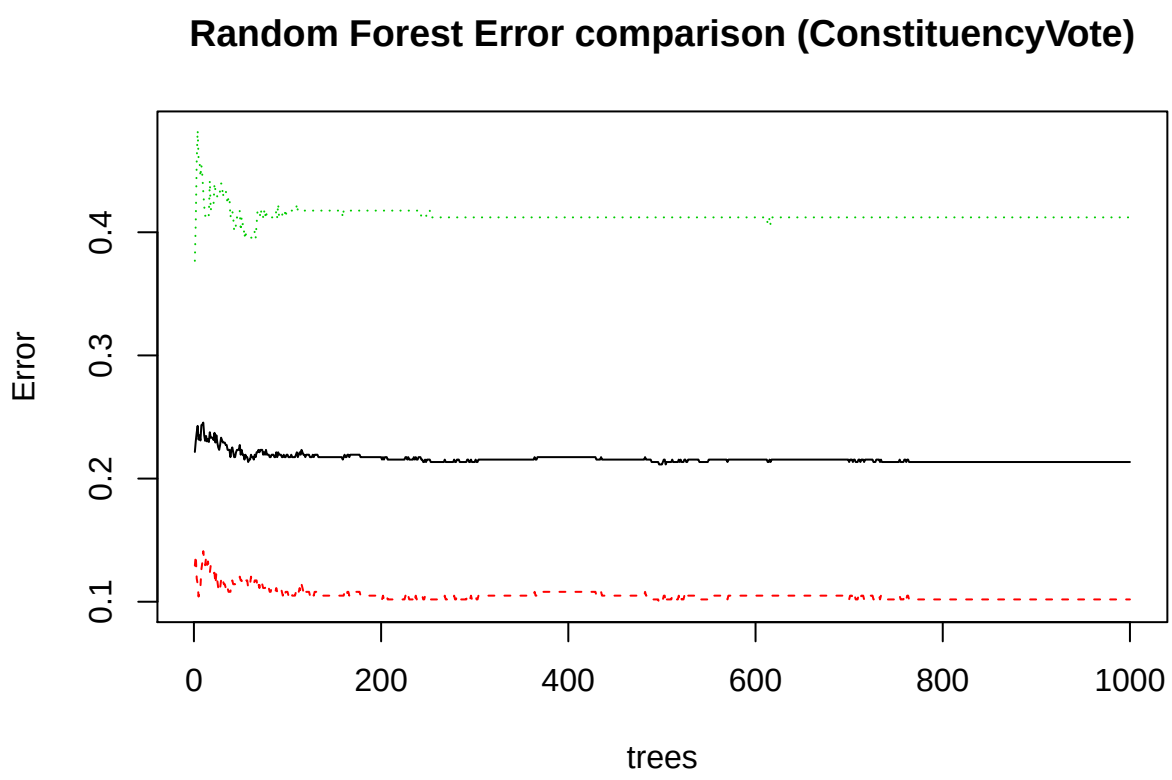

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     66     19
##     Remain    11     27
##
##                Accuracy : 0.7561
##                  95% CI : (0.6705, 0.829)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.001504
##
##                   Kappa : 0.4602
##
##  Mcnemar's Test P-Value : 0.201243
##
##             Sensitivity : 0.8571
##             Specificity : 0.5870
##          Pos Pred Value : 0.7765
##          Neg Pred Value : 0.7105
##              Prevalence : 0.6260
##          Detection Rate : 0.5366
##    Detection Prevalence : 0.6911
##       Balanced Accuracy : 0.7220
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.722
```



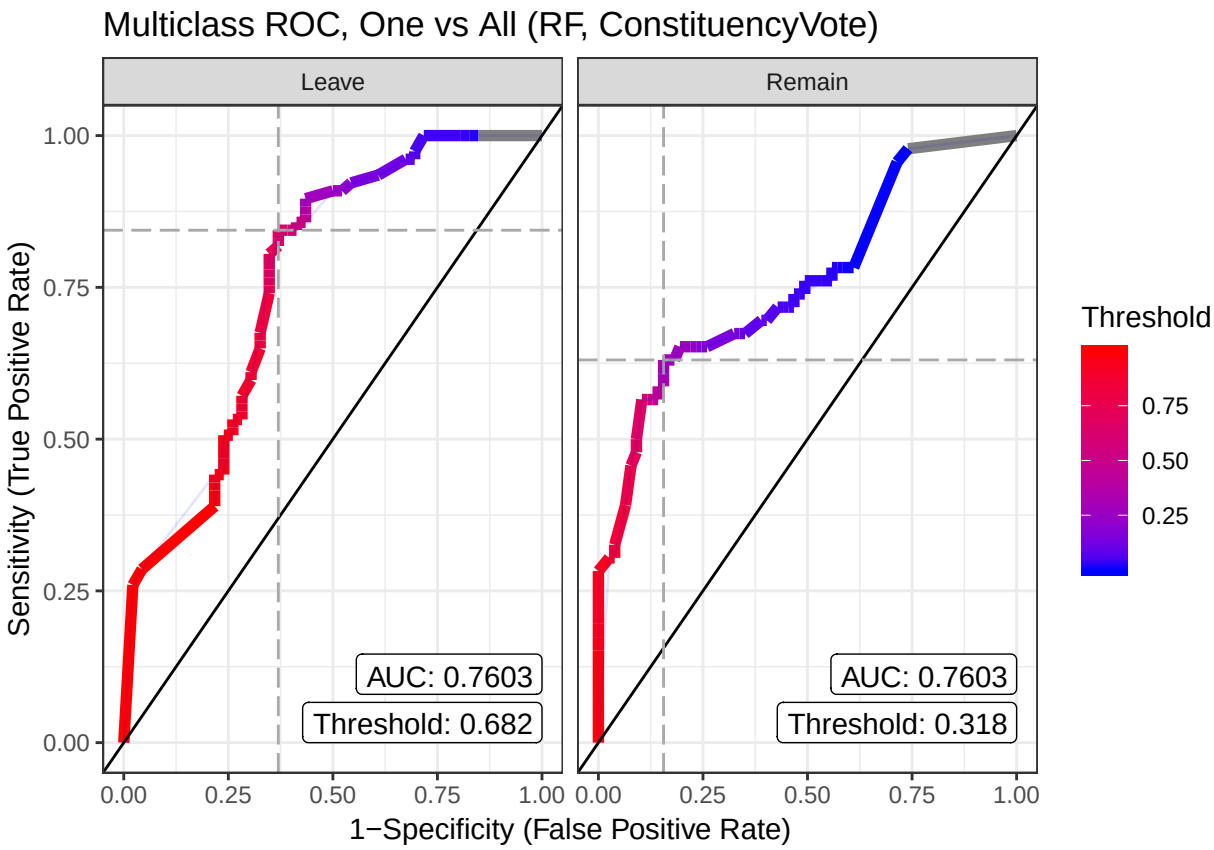

Variable importance (Tree, ConstituencyVote)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

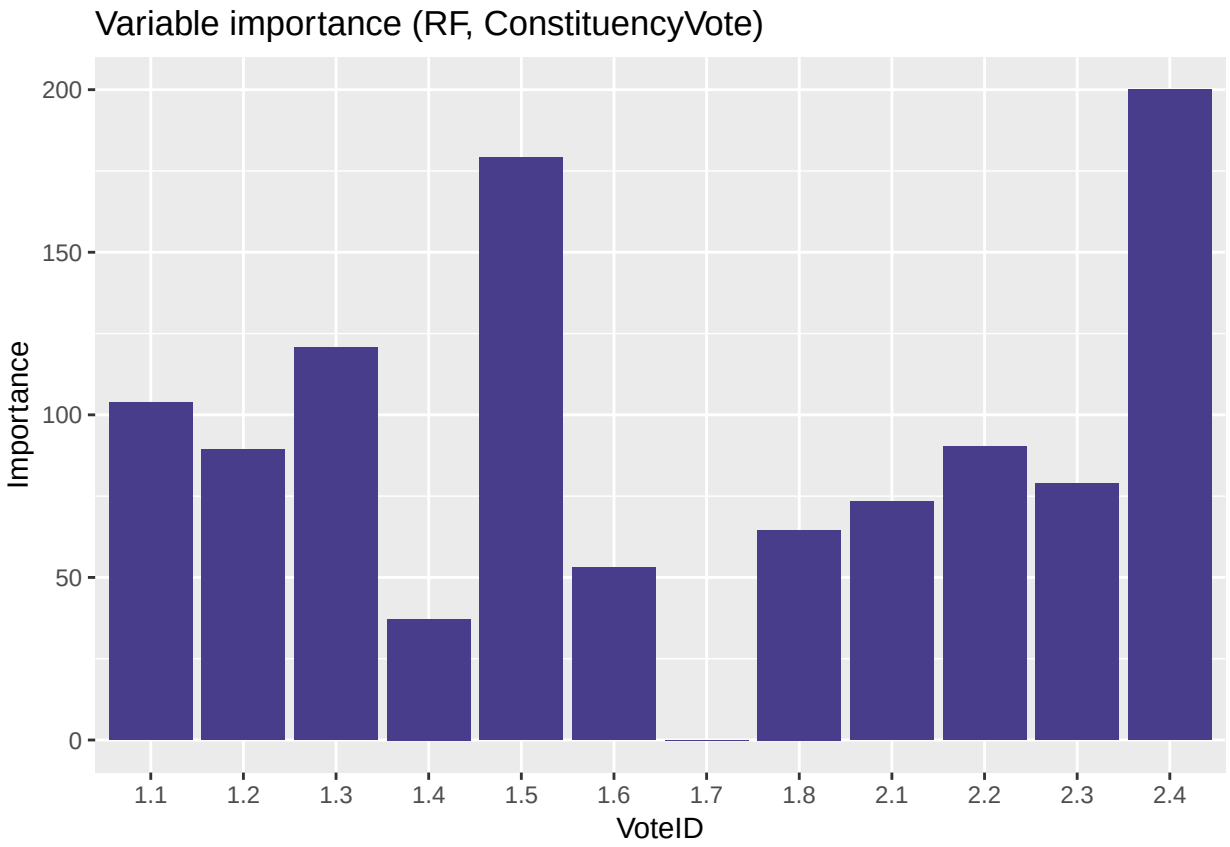

Parameter comparison (RF, ConstituencyVote)
Accuracy (Repeated Cross−Validation)
0.780
0.775
0.770
0.765
0.760
0.755
2
4
6
8
10
12
#Randomly Selected Predictors

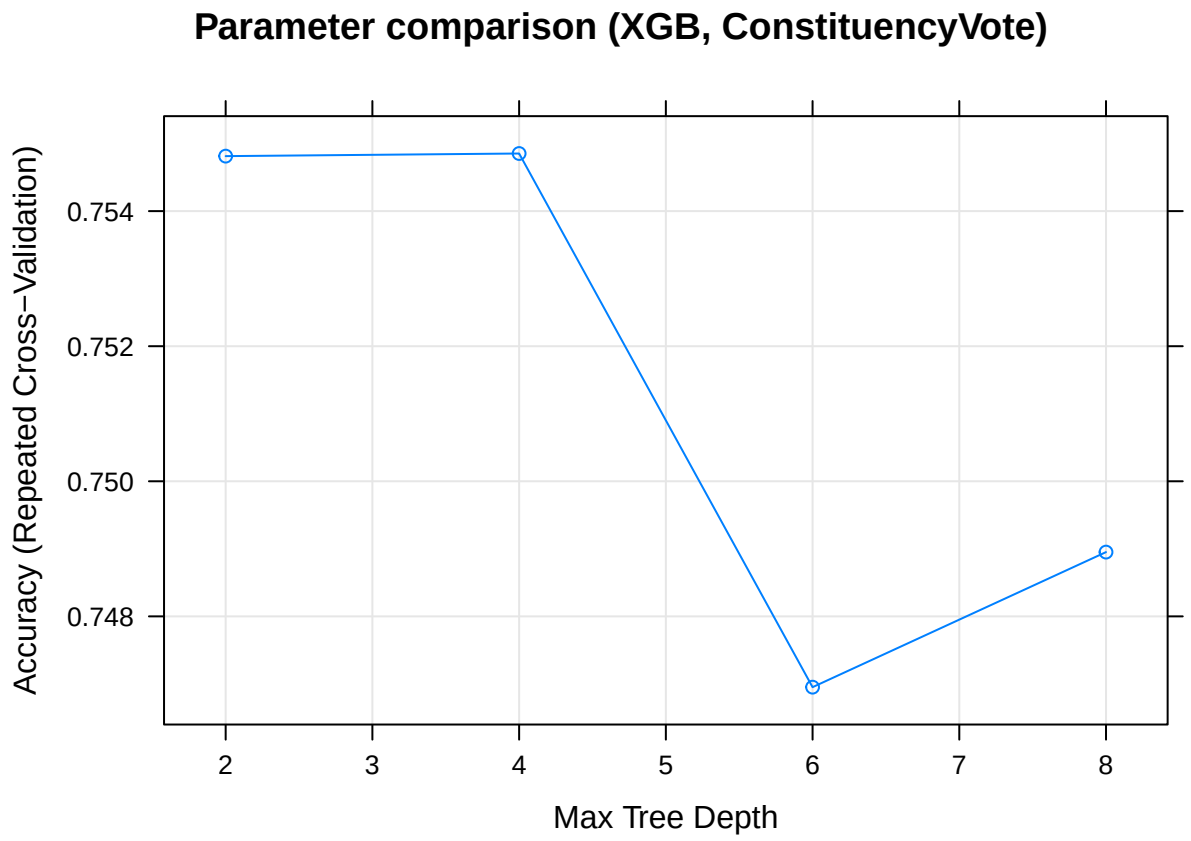



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     67     20
##     Remain    10     26
##
##                Accuracy : 0.7561
##                  95% CI : (0.6705, 0.829)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.001504
##
##                   Kappa : 0.4553
##
##  Mcnemar's Test P-Value : 0.100348
##
##             Sensitivity : 0.8701
##             Specificity : 0.5652
##          Pos Pred Value : 0.7701
##          Neg Pred Value : 0.7222
##              Prevalence : 0.6260
##          Detection Rate : 0.5447
##    Detection Prevalence : 0.7073
##       Balanced Accuracy : 0.7177
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.7603
```

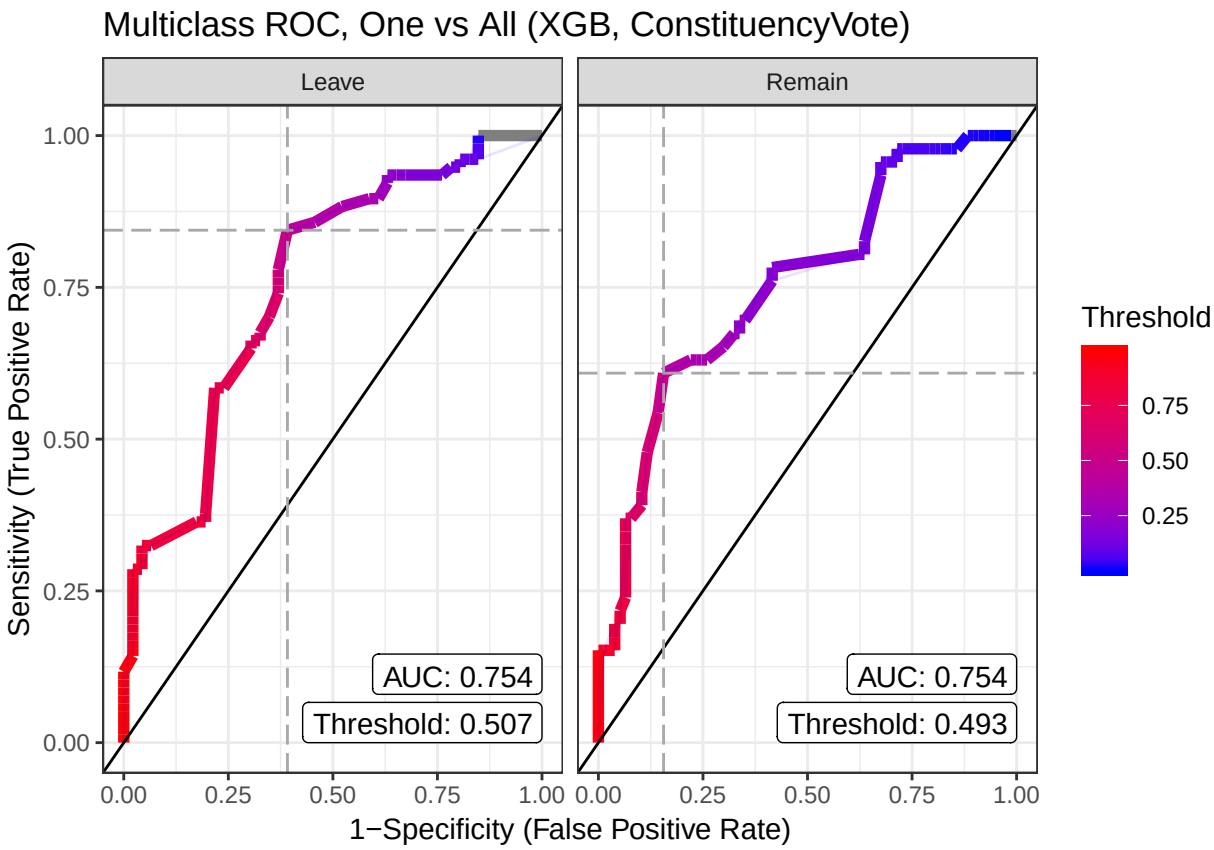


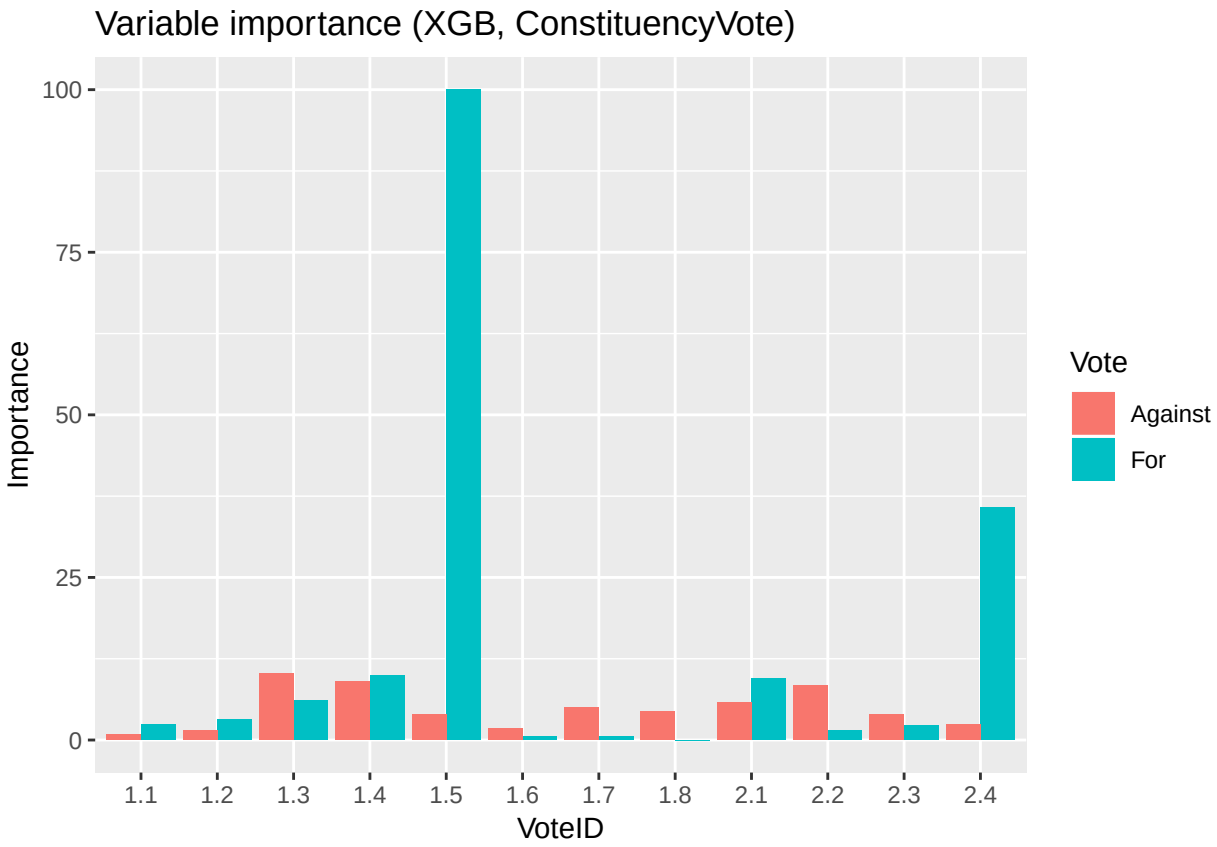

Variable importance (RF, ConstituencyVote)
Importance
200
150
100
50
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

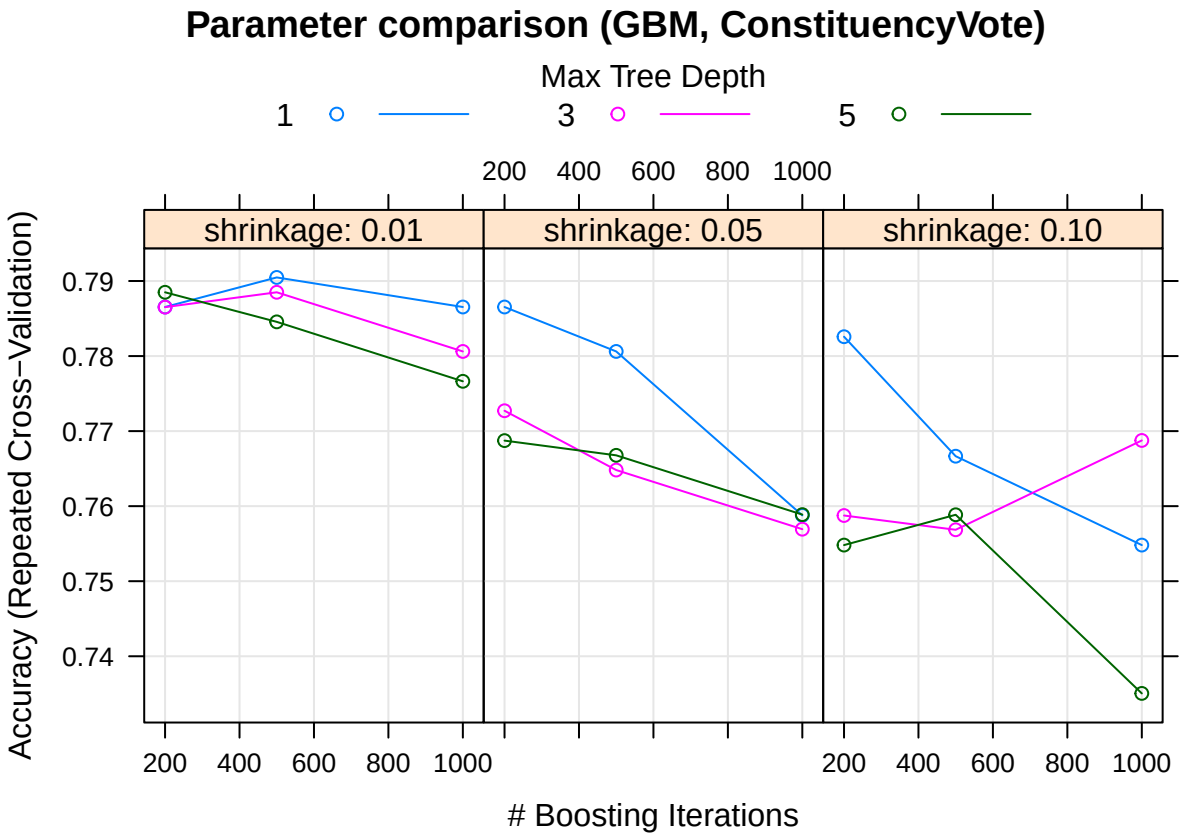



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     65     18
##     Remain    12     28
##
##                Accuracy : 0.7561
##                  95% CI : (0.6705, 0.829)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.001504
##
##                   Kappa : 0.4651
##
##  Mcnemar's Test P-Value : 0.361310
##
##             Sensitivity : 0.8442
##             Specificity : 0.6087
##          Pos Pred Value : 0.7831
##          Neg Pred Value : 0.7000
##              Prevalence : 0.6260
##          Detection Rate : 0.5285
##    Detection Prevalence : 0.6748
##       Balanced Accuracy : 0.7264
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.754
```

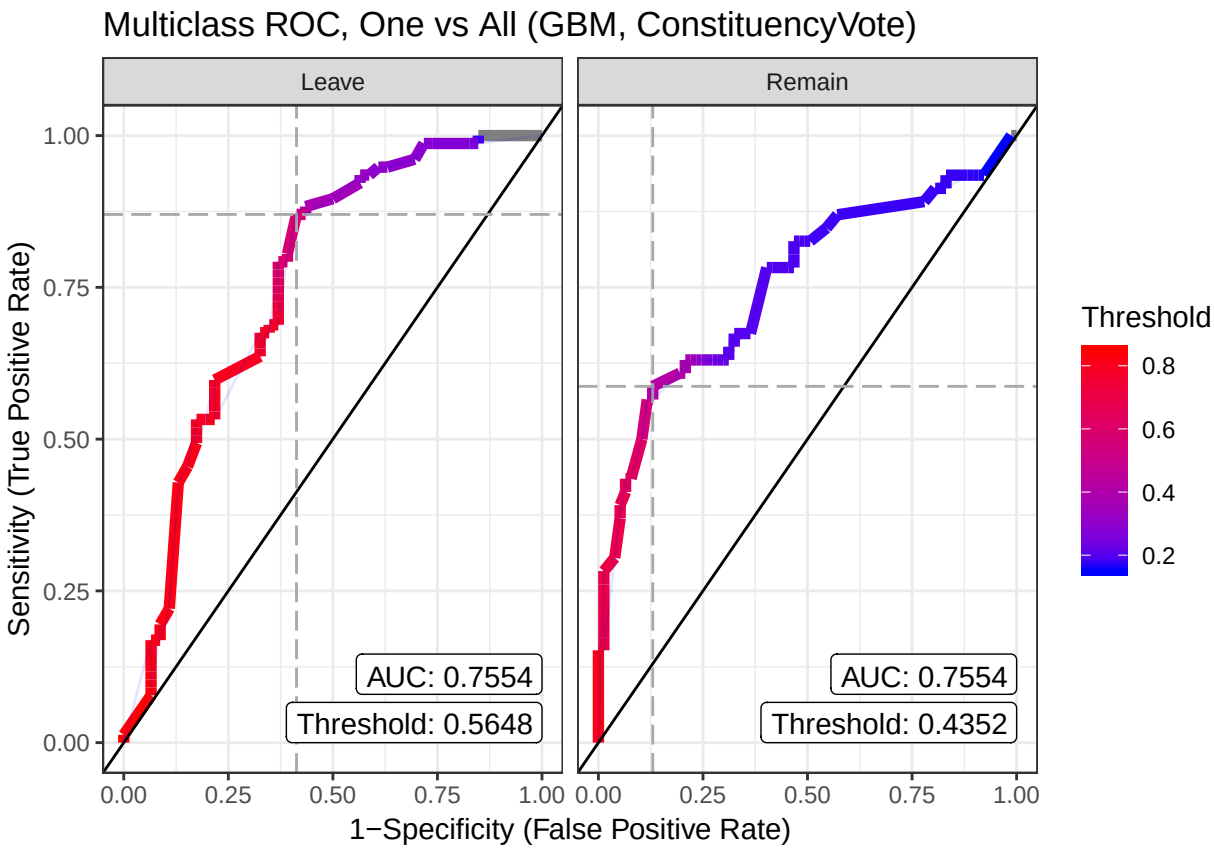


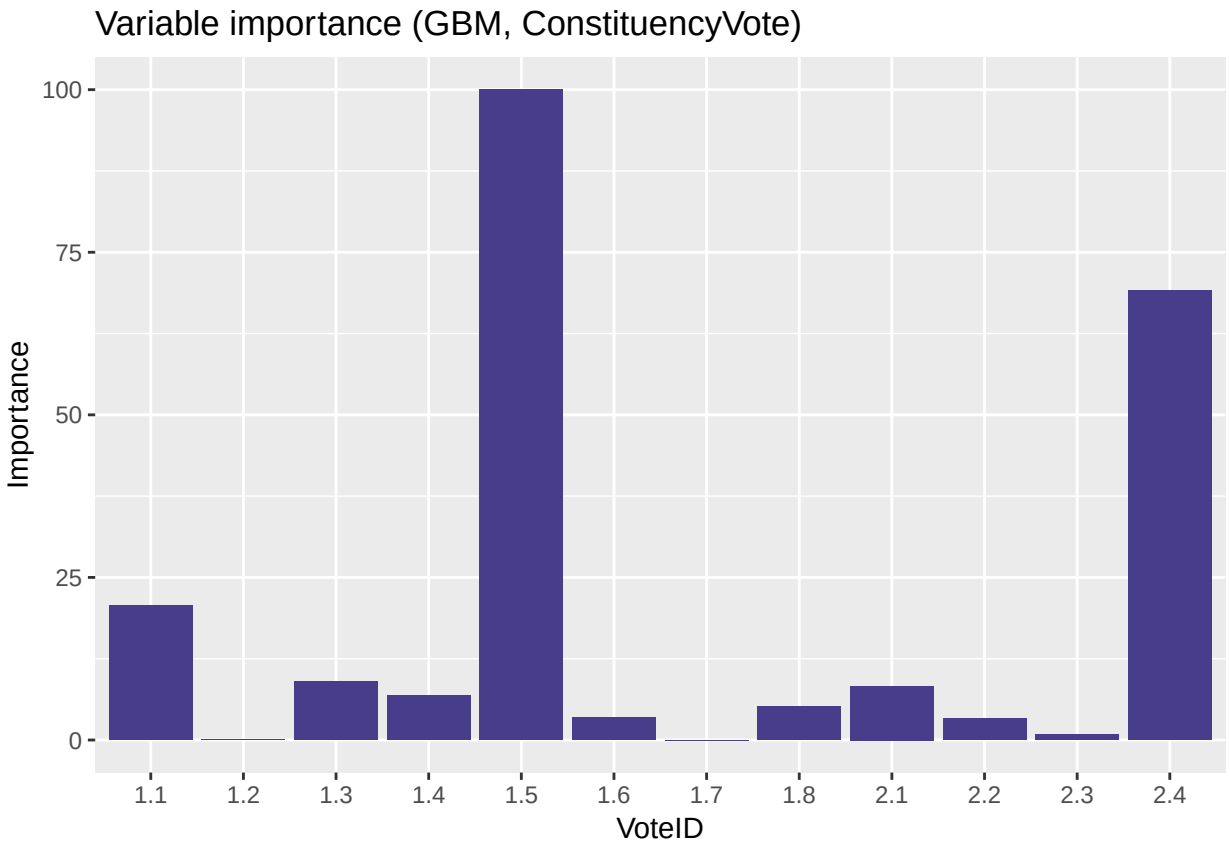

Variable importance (XGB, ConstituencyVote)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For

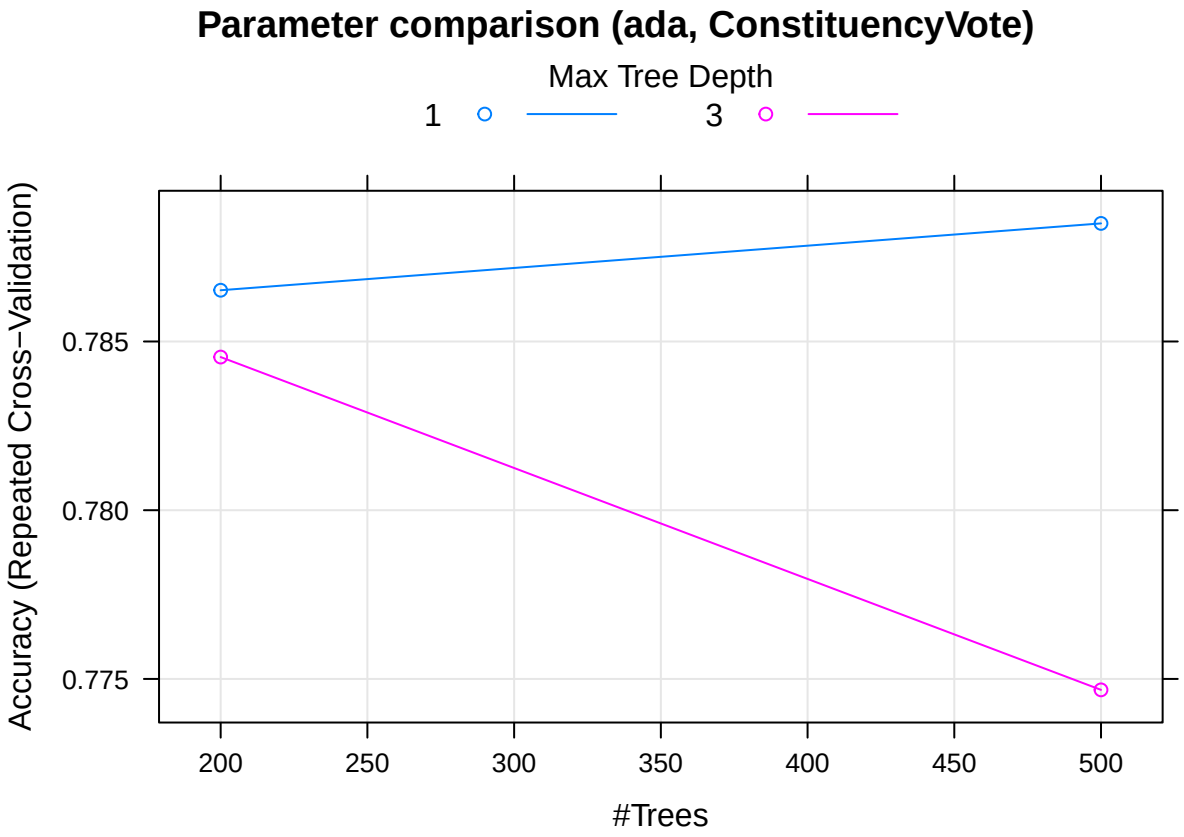



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     68     20
##     Remain     9     26
##
##                Accuracy : 0.7642
##                  95% CI : (0.6793, 0.8361)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.0007699
##
##                   Kappa : 0.471
##
##  Mcnemar's Test P-Value : 0.0633178
##
##             Sensitivity : 0.8831
##             Specificity : 0.5652
##          Pos Pred Value : 0.7727
##          Neg Pred Value : 0.7429
##              Prevalence : 0.6260
##          Detection Rate : 0.5528
##    Detection Prevalence : 0.7154
##       Balanced Accuracy : 0.7242
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.7554
```

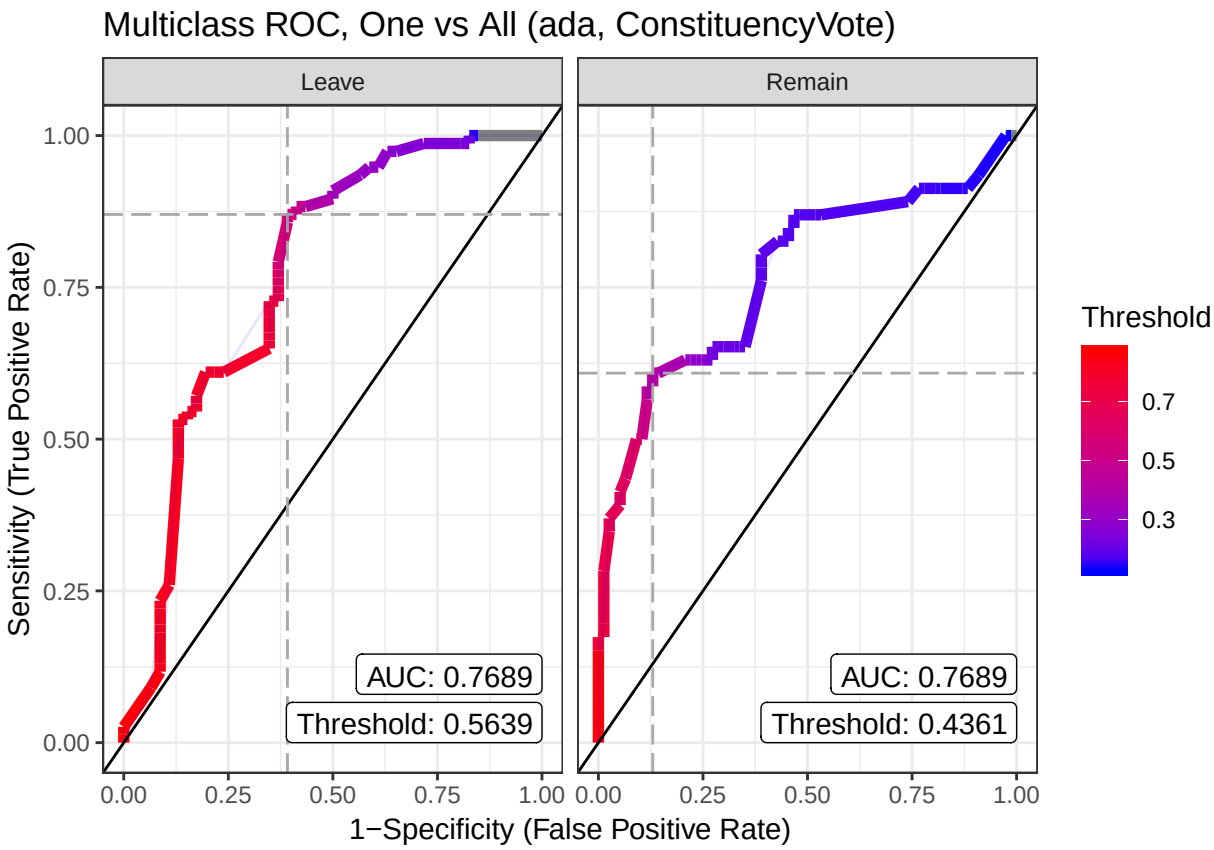


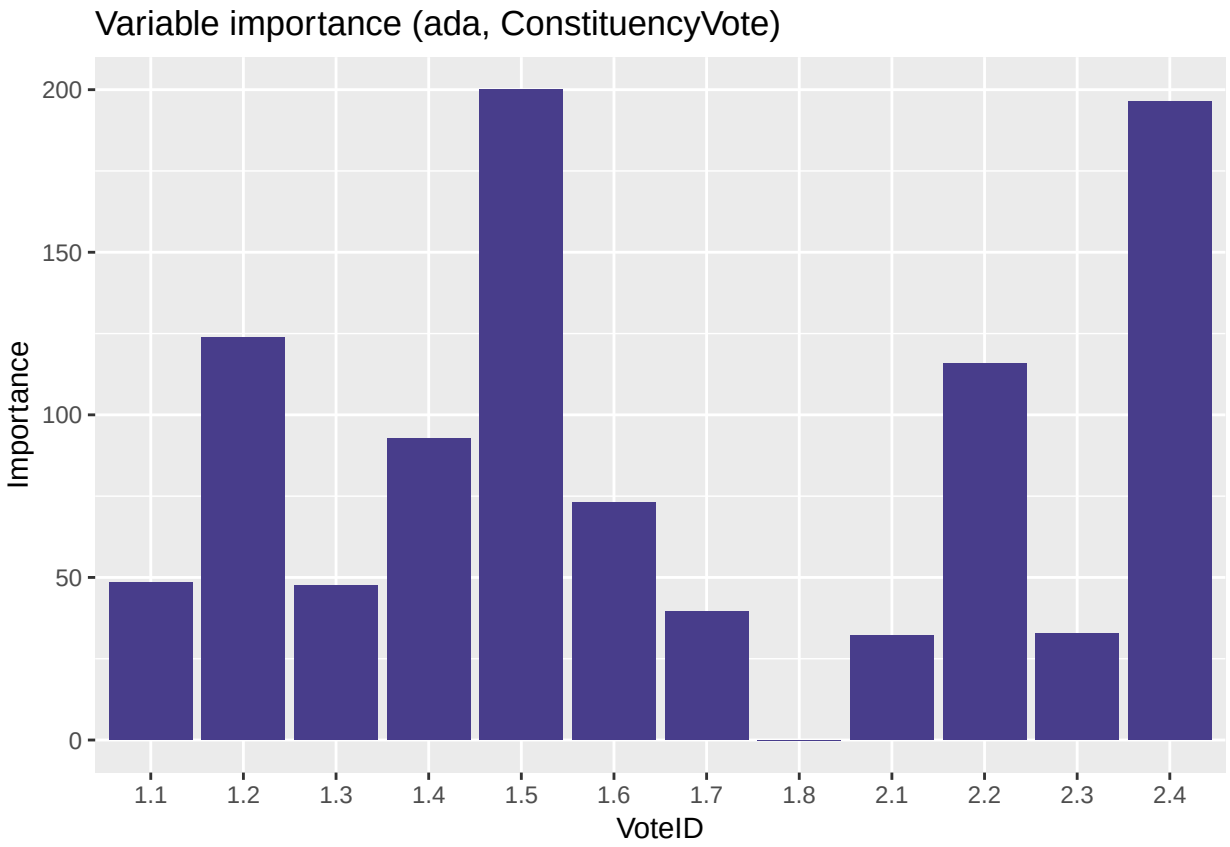

Variable importance (GBM, ConstituencyVote)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

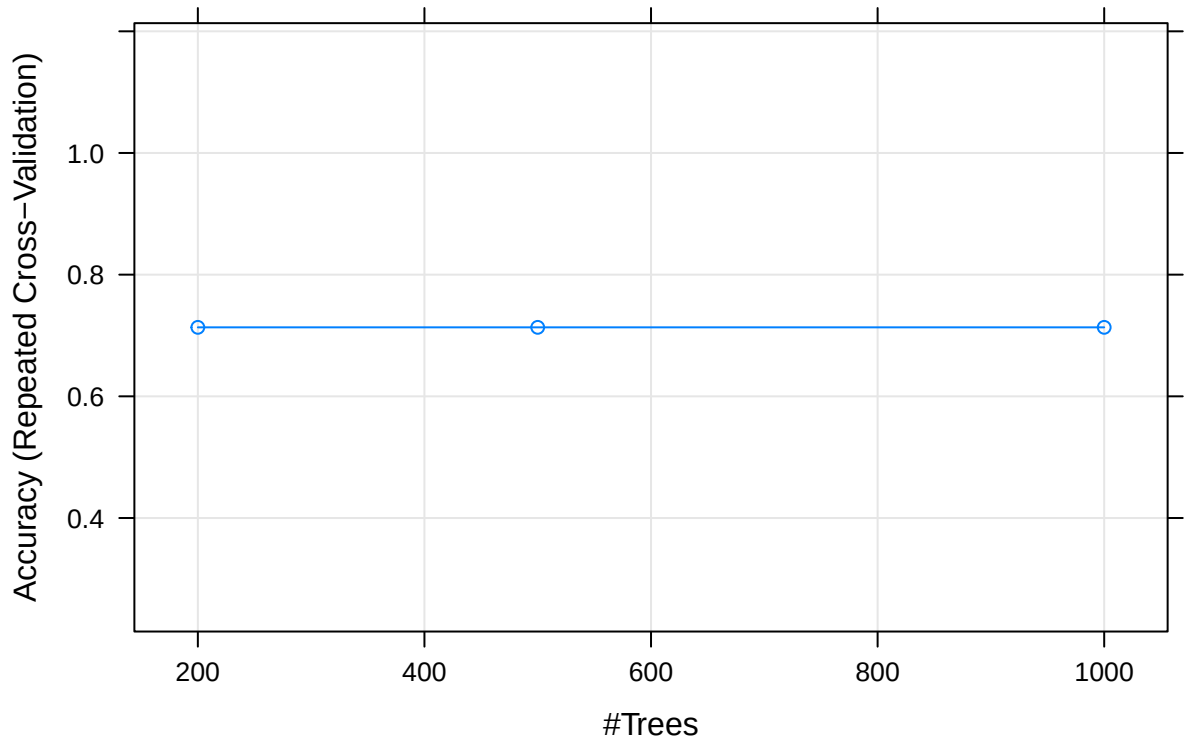



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     68     20
##     Remain     9     26
##
##                Accuracy : 0.7642
##                  95% CI : (0.6793, 0.8361)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.0007699
##
##                   Kappa : 0.471
##
##  Mcnemar's Test P-Value : 0.0633178
##
##             Sensitivity : 0.8831
##             Specificity : 0.5652
##          Pos Pred Value : 0.7727
##          Neg Pred Value : 0.7429
##              Prevalence : 0.6260
##          Detection Rate : 0.5528
##    Detection Prevalence : 0.7154
##       Balanced Accuracy : 0.7242
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.7689
```

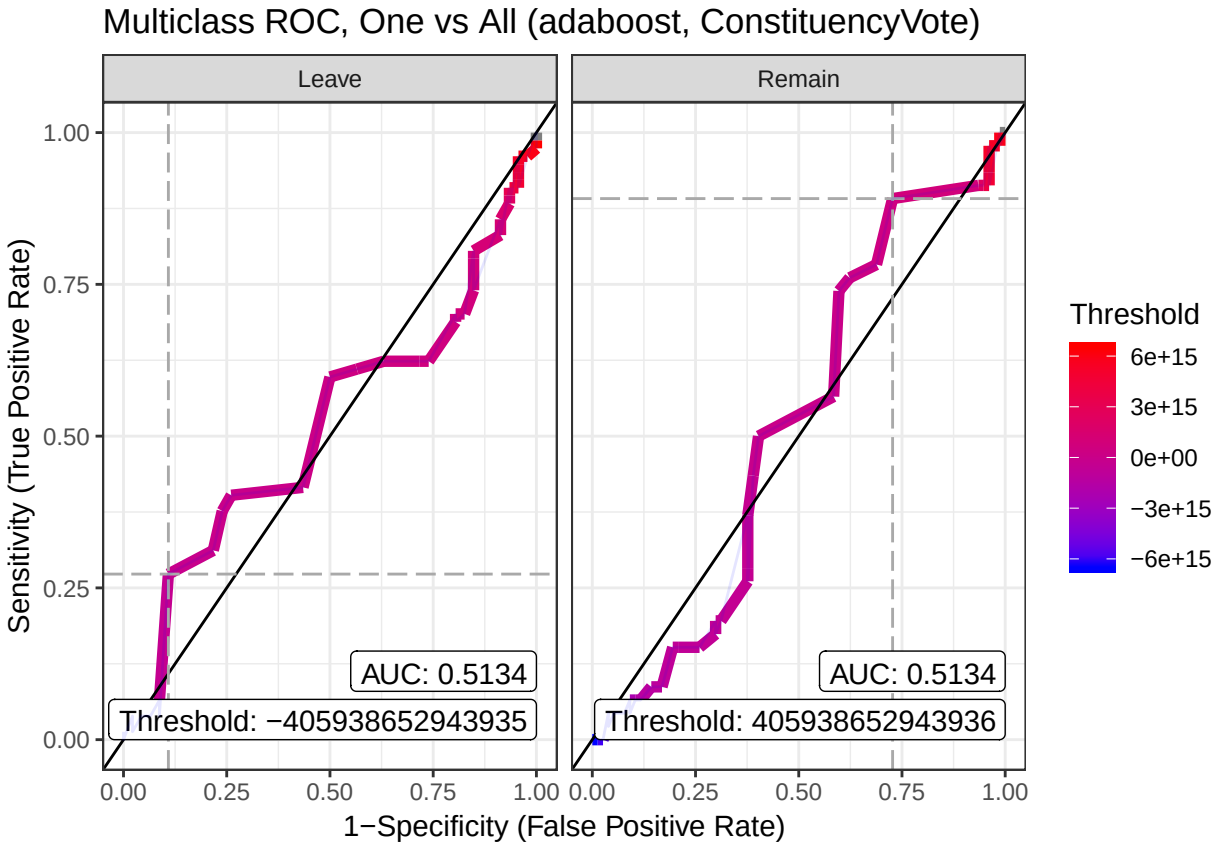


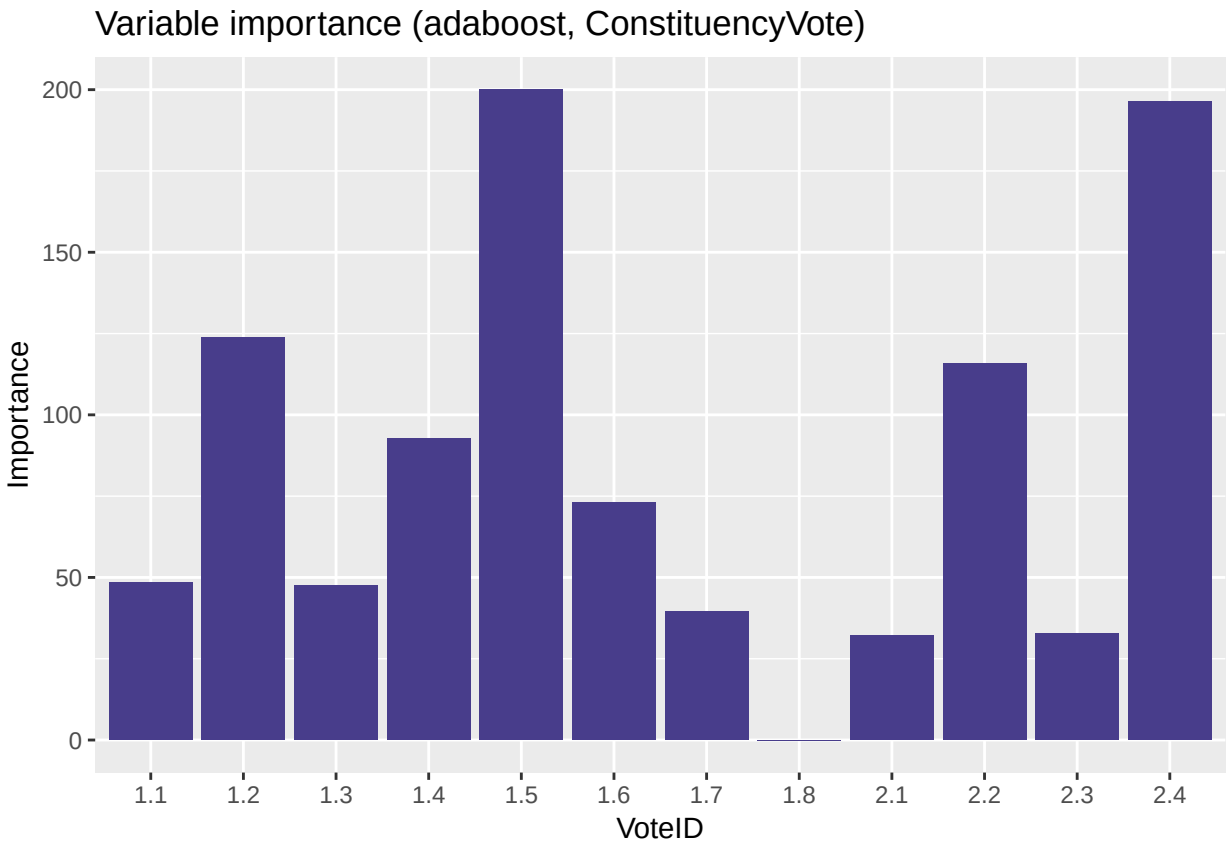

Variable importance (ada, ConstituencyVote)
Importance
200
150
100
50
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

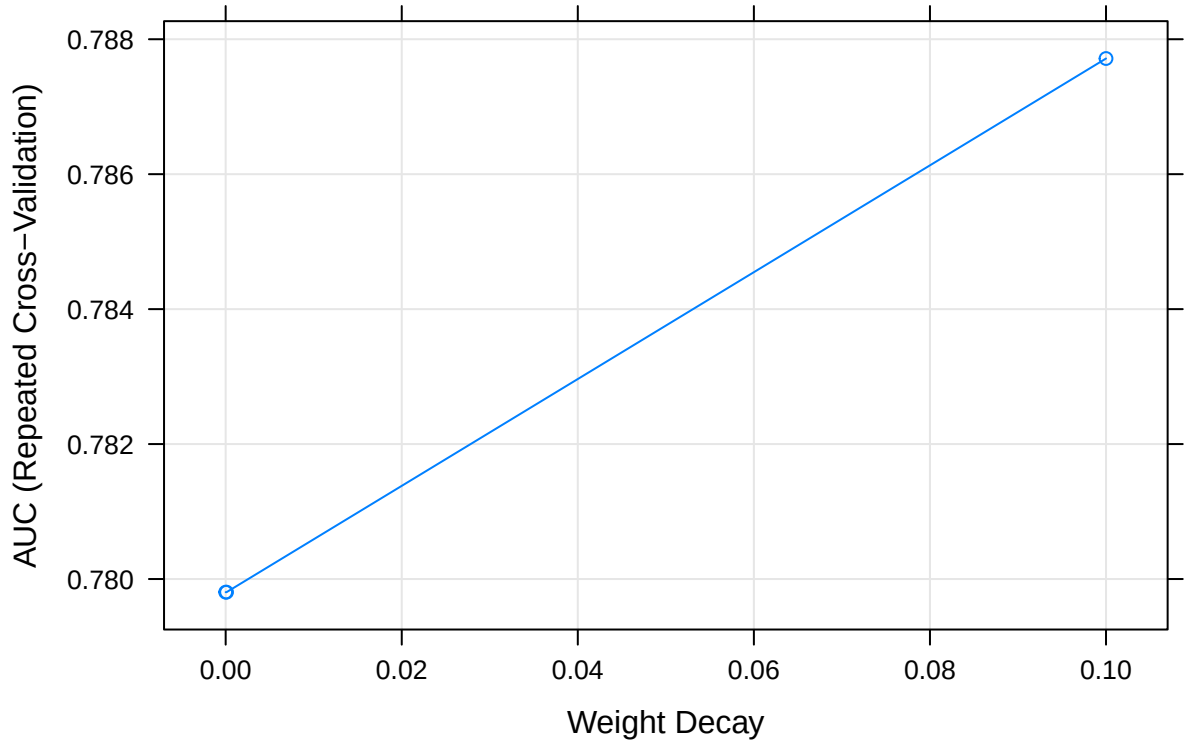



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     55     16
##     Remain    22     30
##
##                Accuracy : 0.6911
##                  95% CI : (0.6014, 0.7712)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.0798
##
##                   Kappa : 0.3571
##
##  Mcnemar's Test P-Value : 0.4173
##
##             Sensitivity : 0.7143
##             Specificity : 0.6522
##          Pos Pred Value : 0.7746
##          Neg Pred Value : 0.5769
##              Prevalence : 0.6260
##          Detection Rate : 0.4472
##    Detection Prevalence : 0.5772
##       Balanced Accuracy : 0.6832
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.4866
```

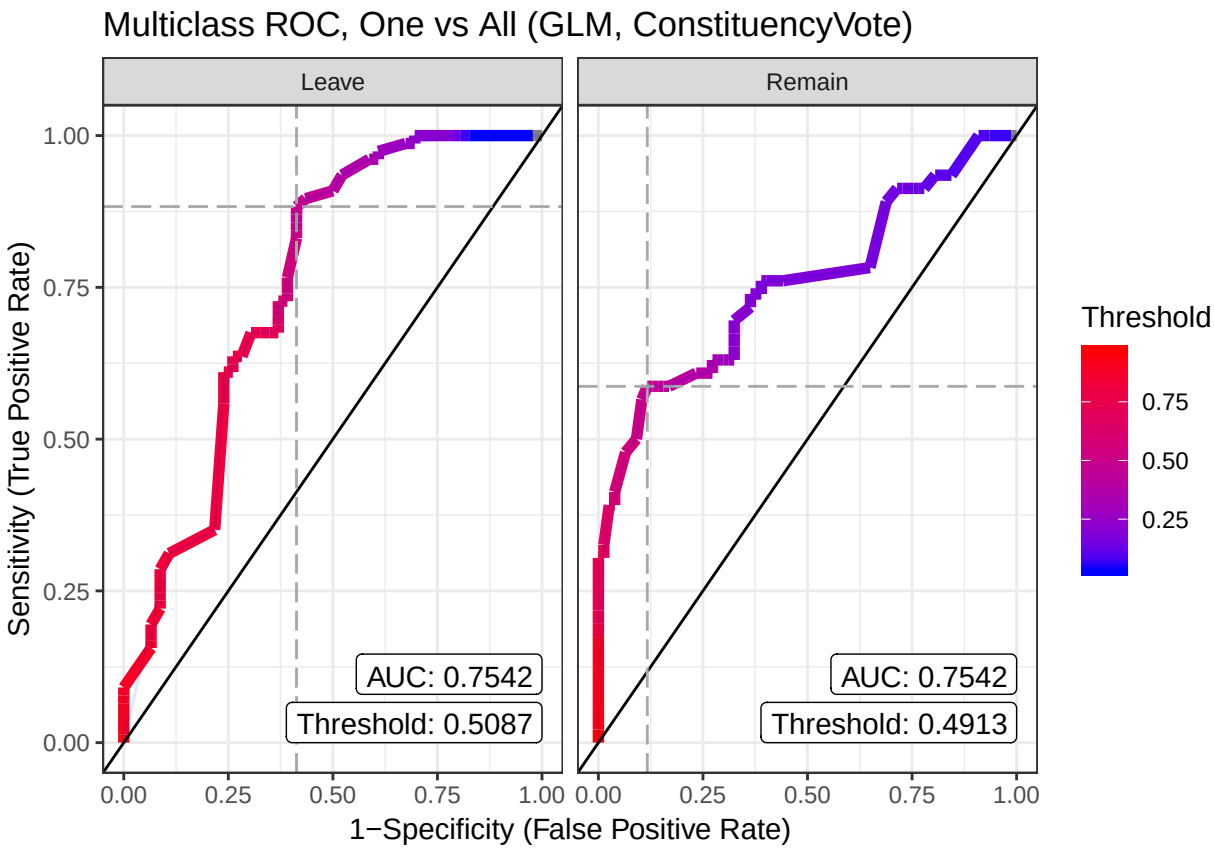



Variable importance (adaboost, ConstituencyVote)
Importance
200
150
100
50
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     69     20
##     Remain     8     26
##
##                Accuracy : 0.7724
##                  95% CI : (0.6881, 0.8431)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.0003779
##
##                   Kappa : 0.4869
##
##  Mcnemar's Test P-Value : 0.0376353
##
##             Sensitivity : 0.8961
##             Specificity : 0.5652
##          Pos Pred Value : 0.7753
##          Neg Pred Value : 0.7647
##              Prevalence : 0.6260
##          Detection Rate : 0.5610
##    Detection Prevalence : 0.7236
##       Balanced Accuracy : 0.7307
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.7542
```




```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 126
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86
```

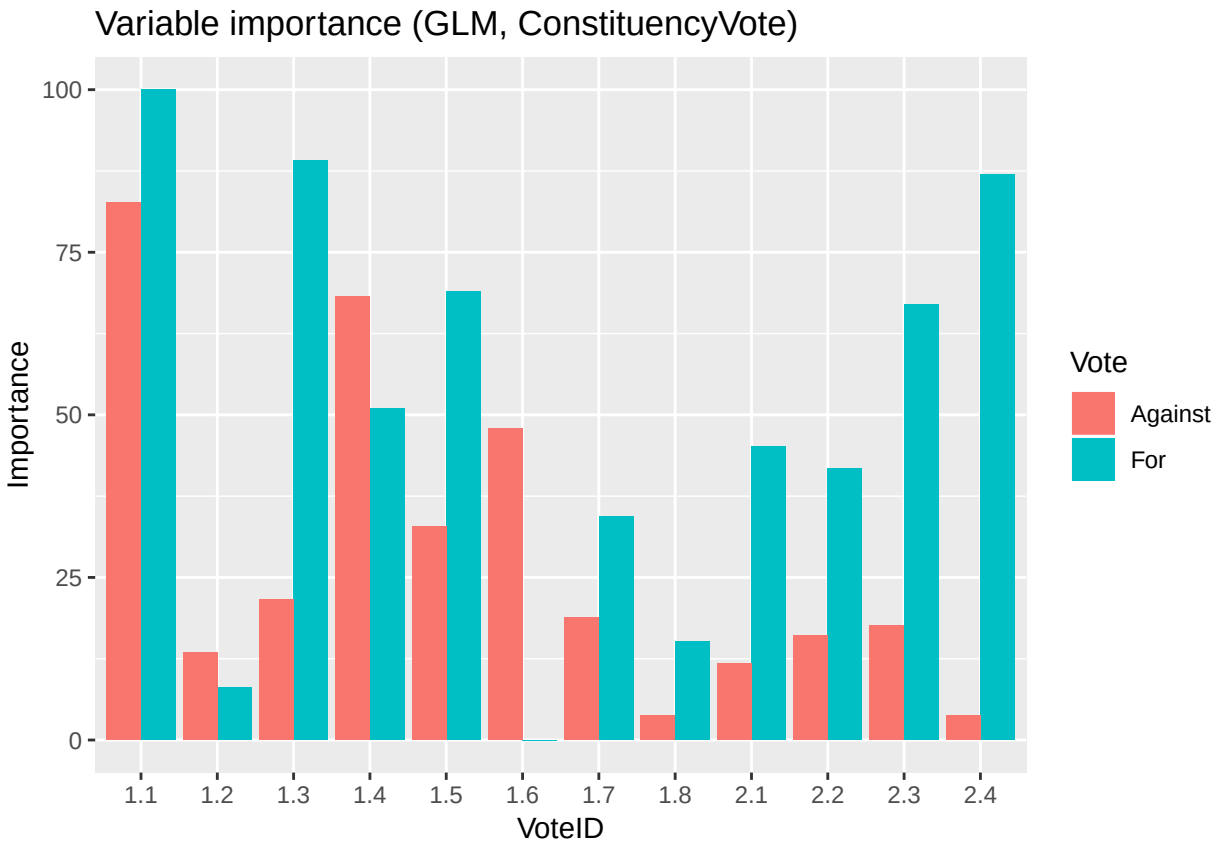
```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93
```

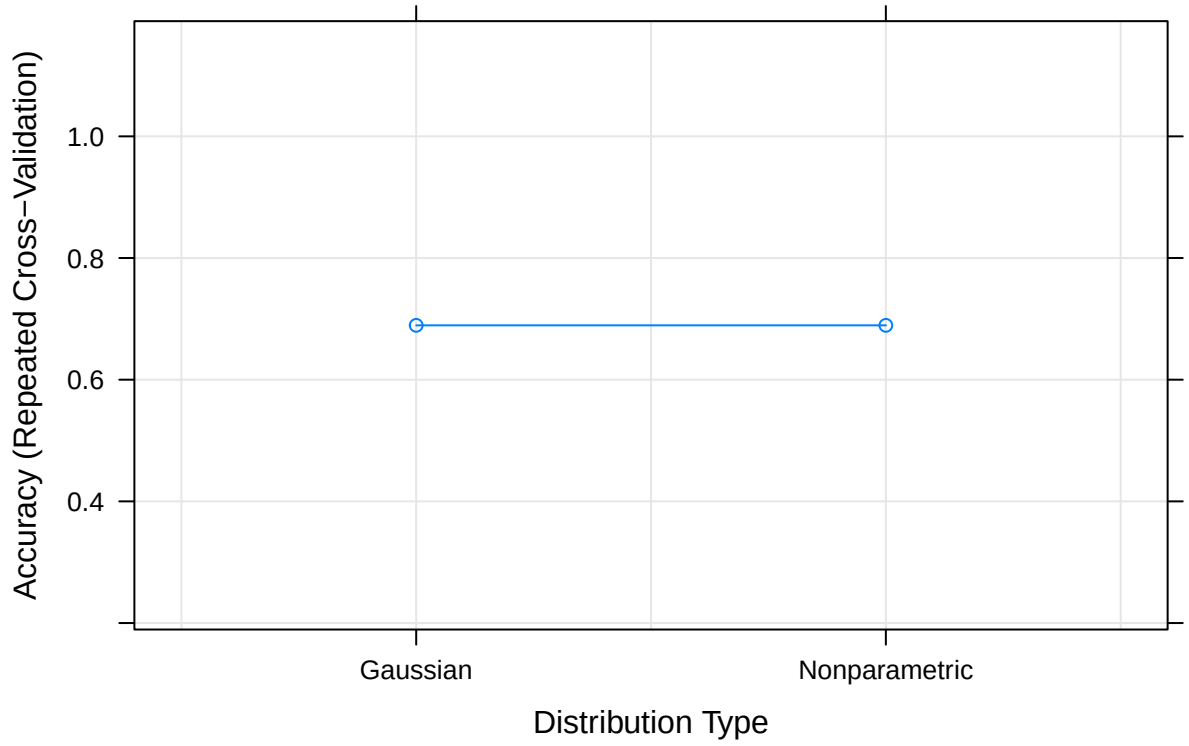
```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106
```

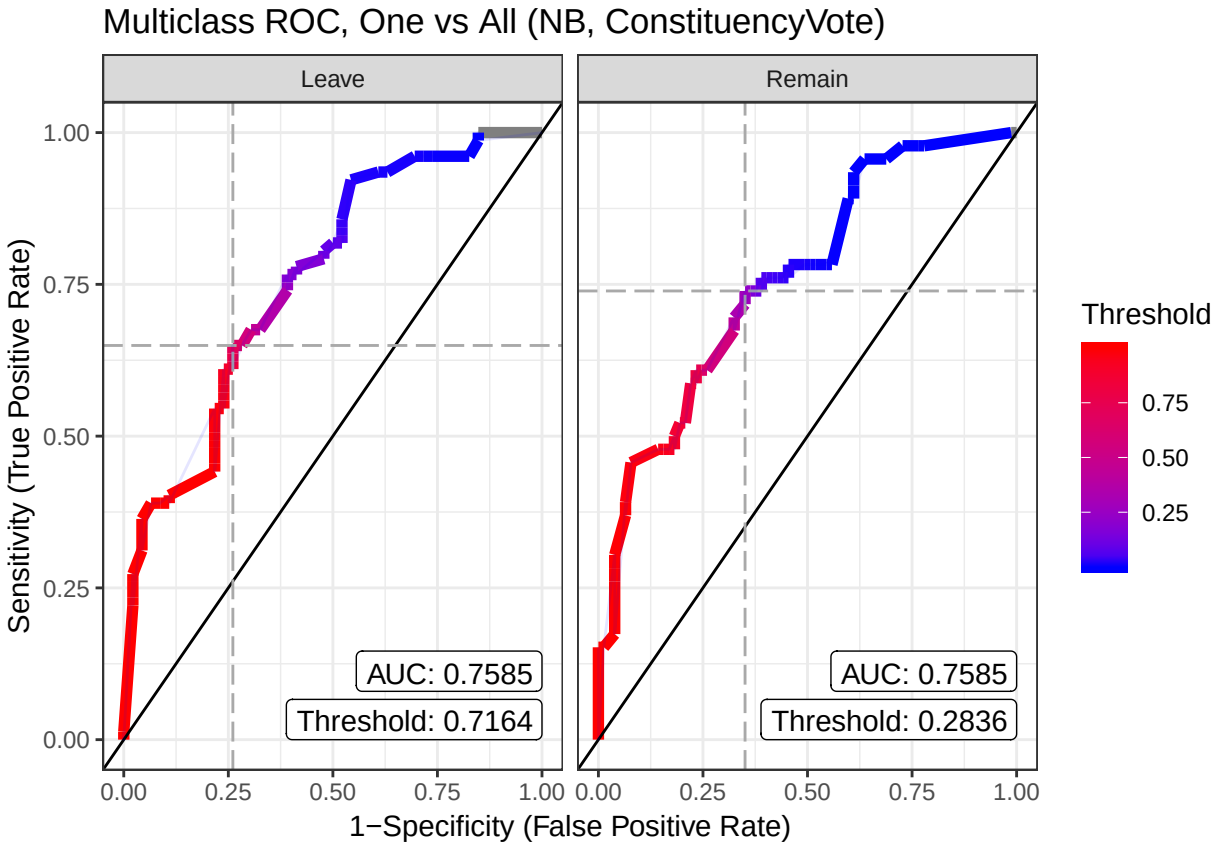
```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126
```

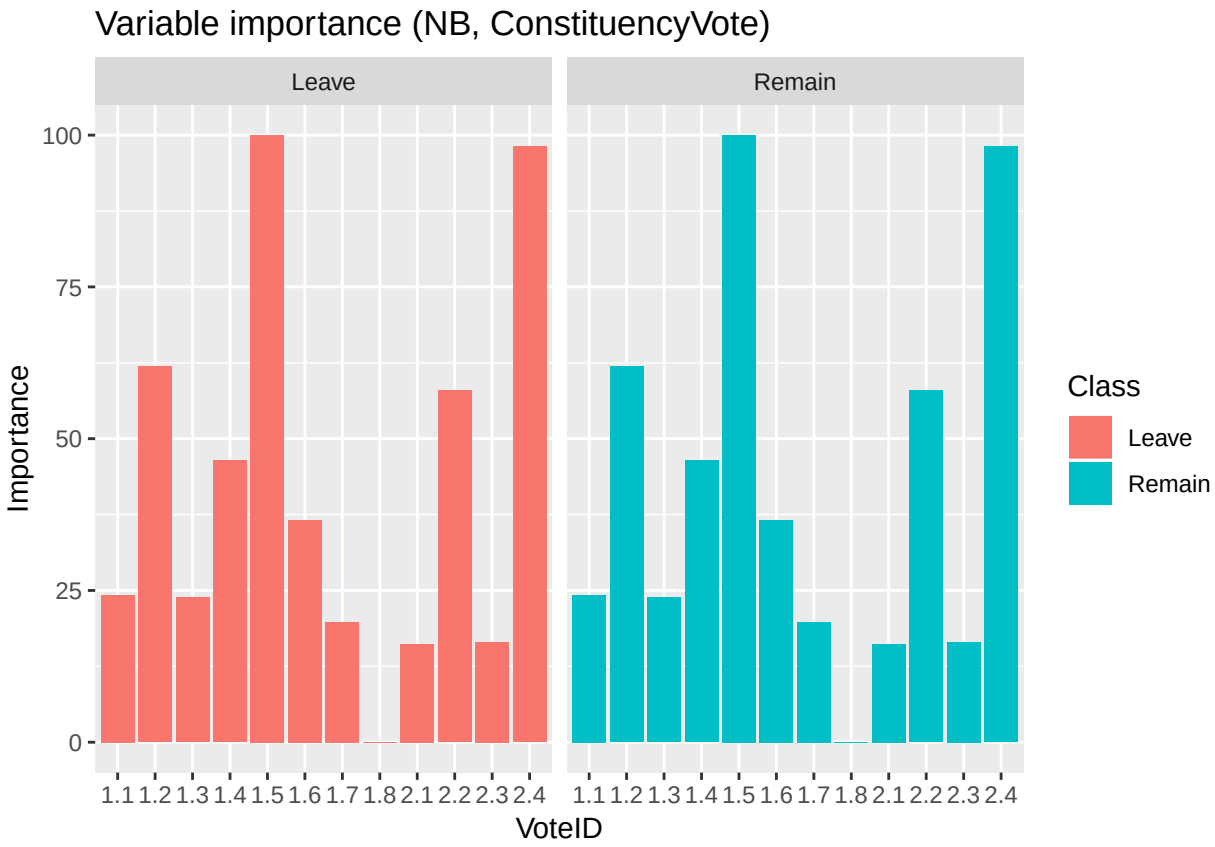

Variable importance (GLM, ConstituencyVote)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For

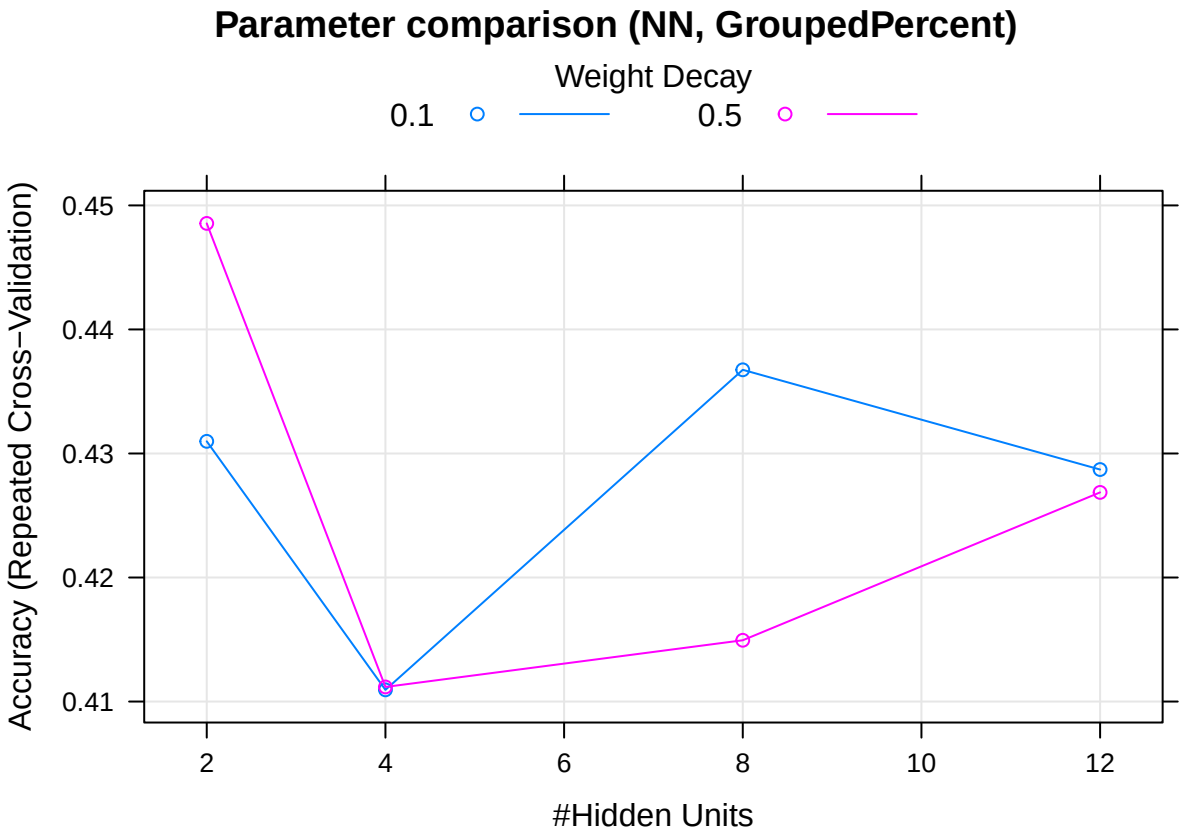



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Leave Remain
##     Leave     52     14
##     Remain    25     32
##
##                Accuracy : 0.6829
##                  95% CI : (0.5929, 0.7639)
##     No Information Rate : 0.626
##     P-Value [Acc > NIR] : 0.1121
##
##                   Kappa : 0.3539
##
##  Mcnemar's Test P-Value : 0.1093
##
##             Sensitivity : 0.6753
##             Specificity : 0.6957
##          Pos Pred Value : 0.7879
##          Neg Pred Value : 0.5614
##              Prevalence : 0.6260
##          Detection Rate : 0.4228
##    Detection Prevalence : 0.5366
##       Balanced Accuracy : 0.6855
##
##        'Positive' Class : Leave
```

```
##
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 2 levels of response: Leave, Remain.
## Multi-class area under the curve: 0.7585
```

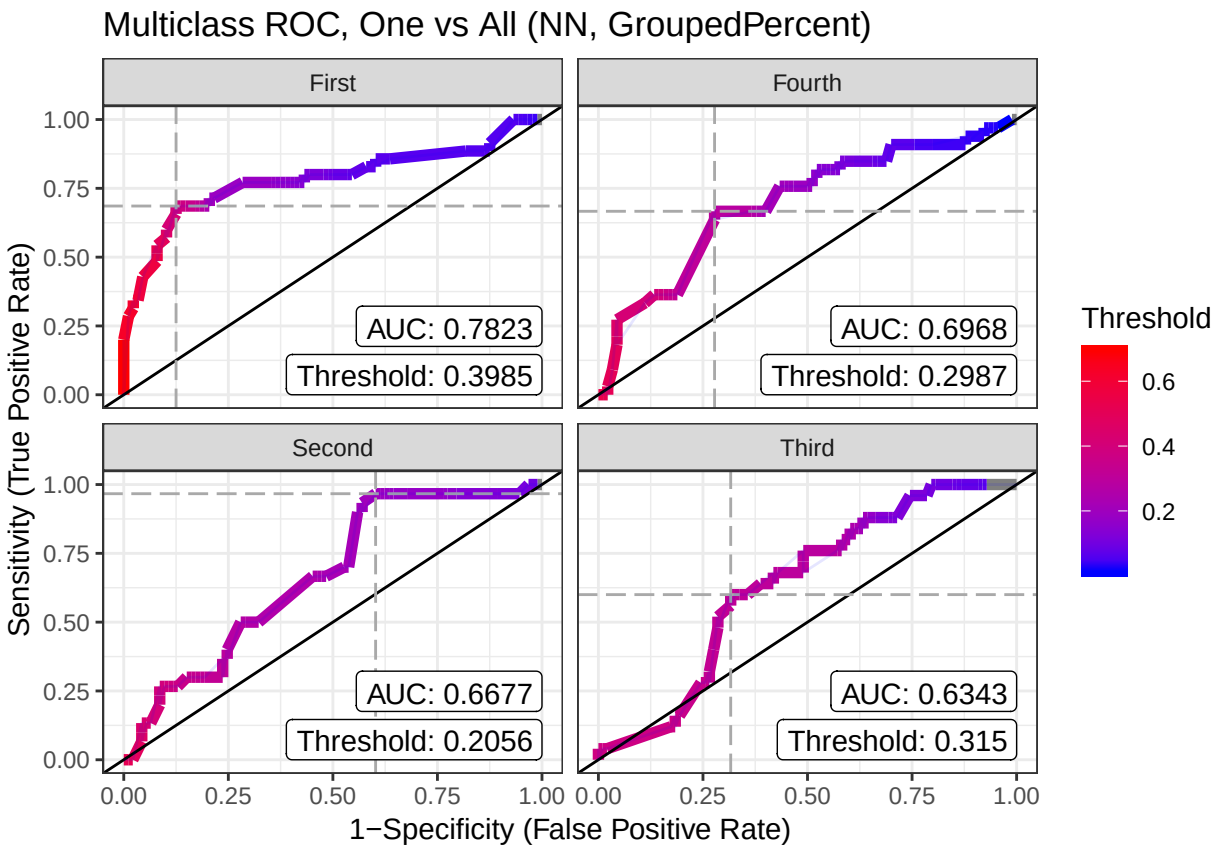


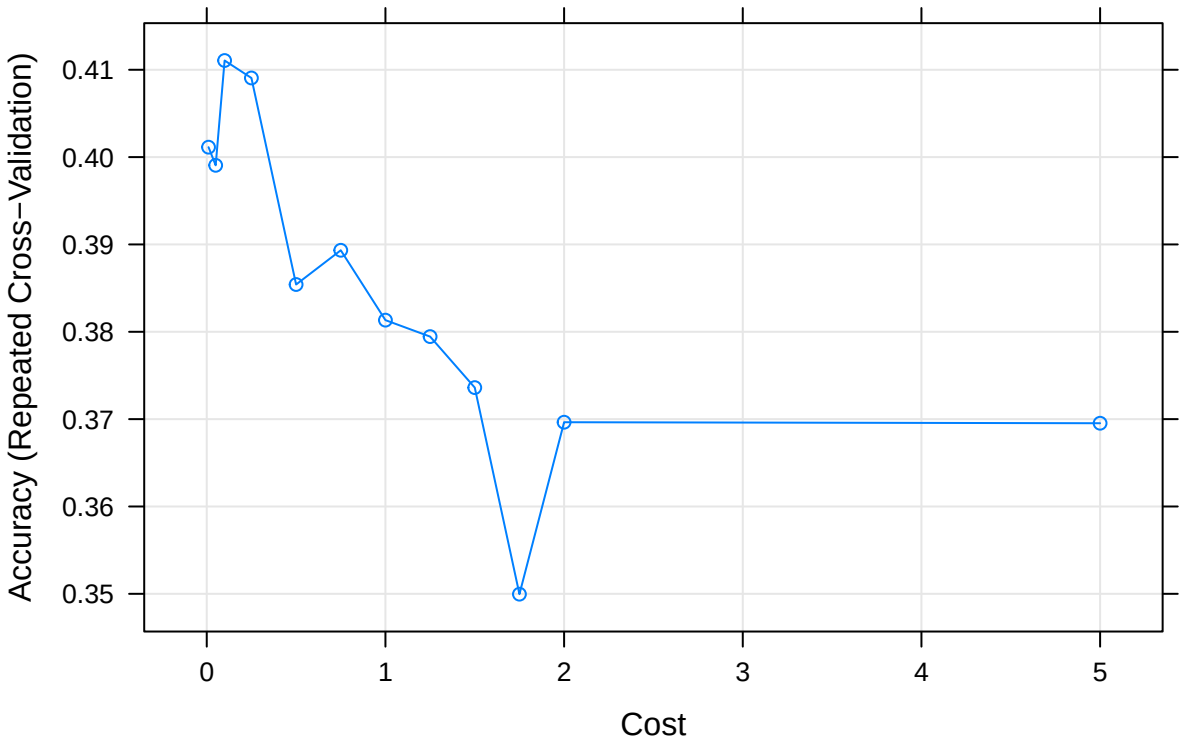

Variable importance (NB, ConstituencyVote)
Leave
Remain
Importance
100
75
50
25
0
1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 2.1 2.2 2.3 2.4
VoteID
Class
Leave
Remain




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     23      4      7     4
##     Fourth     6     13      7    10
##     Second     4      4      7     6
##     Third      2     12      9     5
##
## Overall Statistics
##
##                Accuracy : 0.3902
##                  95% CI : (0.3036, 0.4823)
##     No Information Rate : 0.2846
##     P-Value [Acc > NIR] : 0.007411
##
##                   Kappa : 0.1823
##
##  Mcnemar's Test P-Value : 0.745984
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.6571        0.3939       0.23333      0.20000
## Specificity                0.8295        0.7444       0.84946      0.76531
## Pos Pred Value             0.6053        0.3611       0.33333      0.17857
```

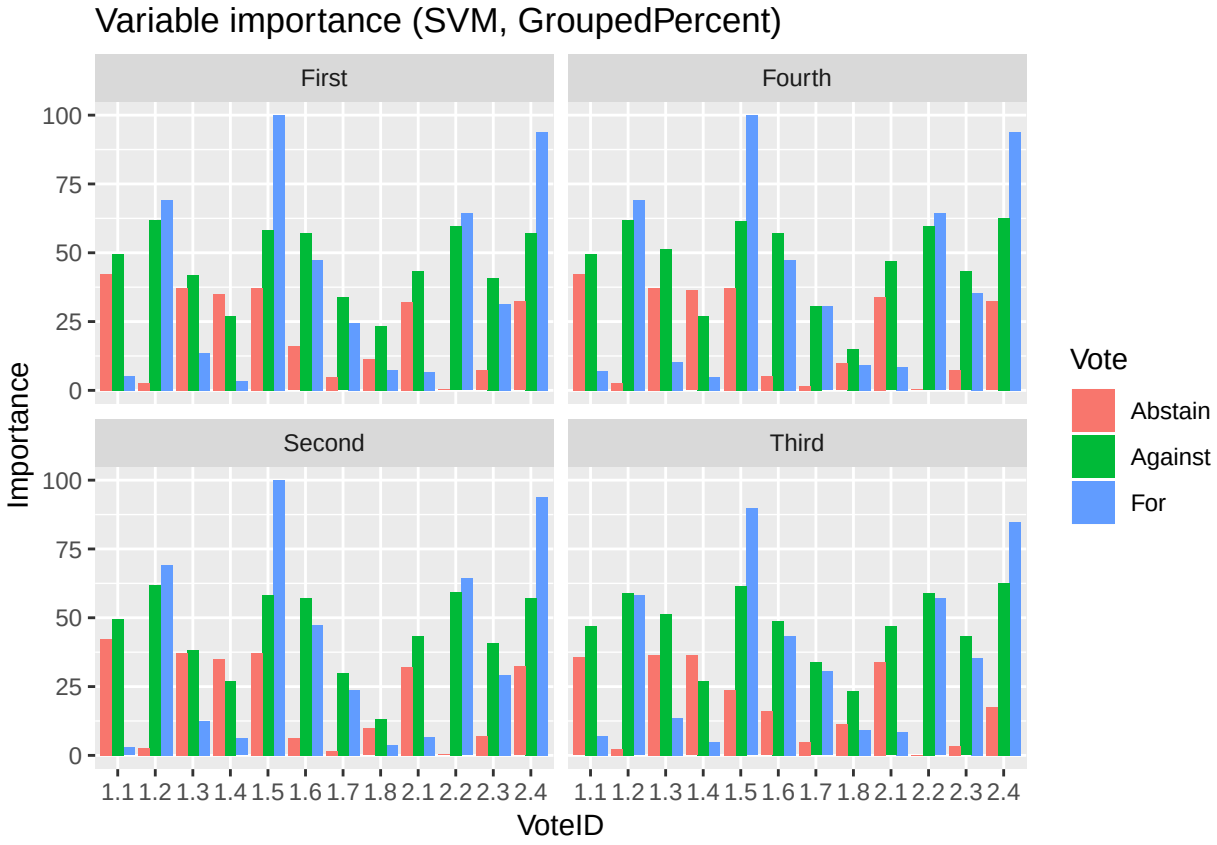
```
## Neg Pred Value              0.8588       0.7701       0.77451      0.78947
## Prevalence                  0.2846       0.2683       0.24390      0.20325
## Detection Rate              0.1870       0.1057       0.05691      0.04065
## Detection Prevalence        0.3089       0.2927       0.17073      0.22764
## Balanced Accuracy           0.7433       0.5692       0.54140      0.48265
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.6909
```

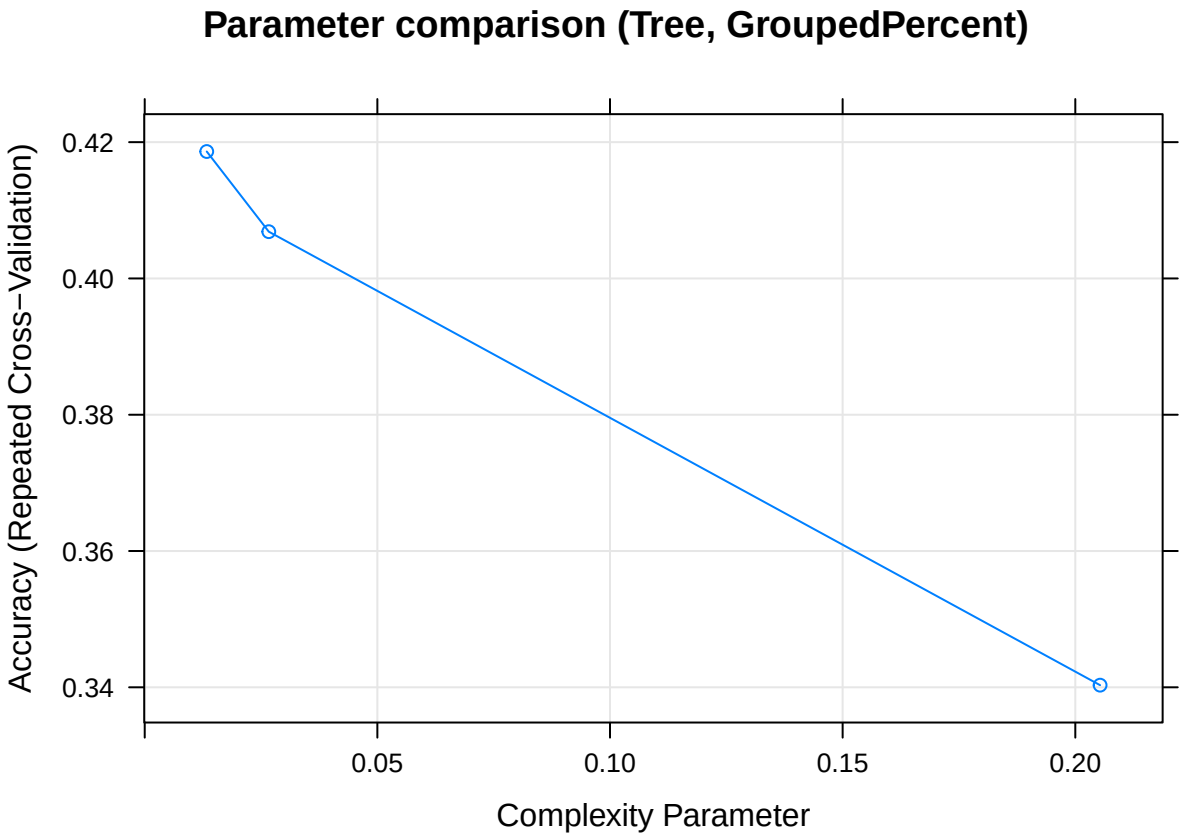


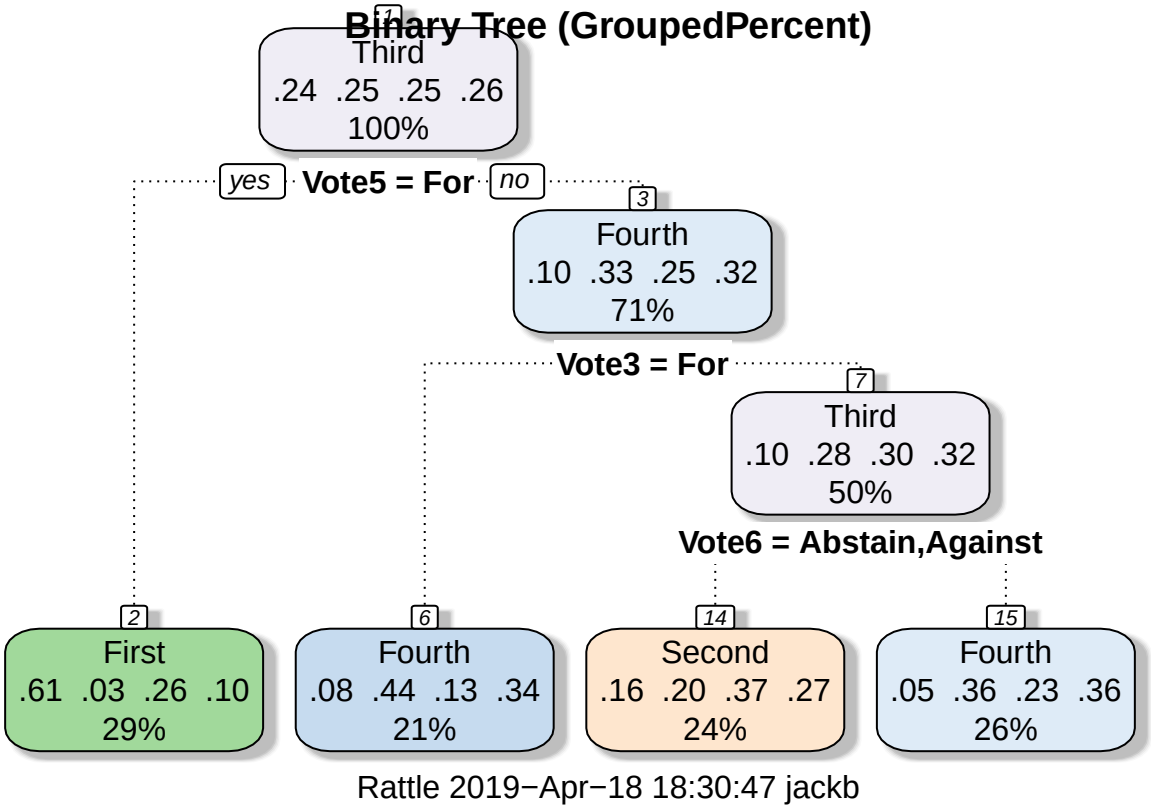



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     24      4      5     5
##     Fourth     5     21      6    11
##     Second     6      7     19     8
##     Third      0      1      0     1
##
## Overall Statistics
##
##                Accuracy : 0.5285
##                  95% CI : (0.4364, 0.6191)
##     No Information Rate : 0.2846
##     P-Value [Acc > NIR] : 1.164e-08
##
##                   Kappa : 0.359
##
##  Mcnemar's Test P-Value : 0.001423
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.6857        0.6364        0.6333      0.04000
## Specificity                0.8409        0.7556        0.7742      0.98980
## Pos Pred Value             0.6316        0.4884        0.4750      0.50000
```

```
## Neg Pred Value              0.8706          0.8500          0.8675          0.80165
## Prevalence                  0.2846          0.2683          0.2439          0.20325
## Detection Rate              0.1951          0.1707          0.1545          0.00813
## Detection Prevalence        0.3089          0.3496          0.3252          0.01626
## Balanced Accuracy           0.7633          0.6960          0.7038          0.51490
```

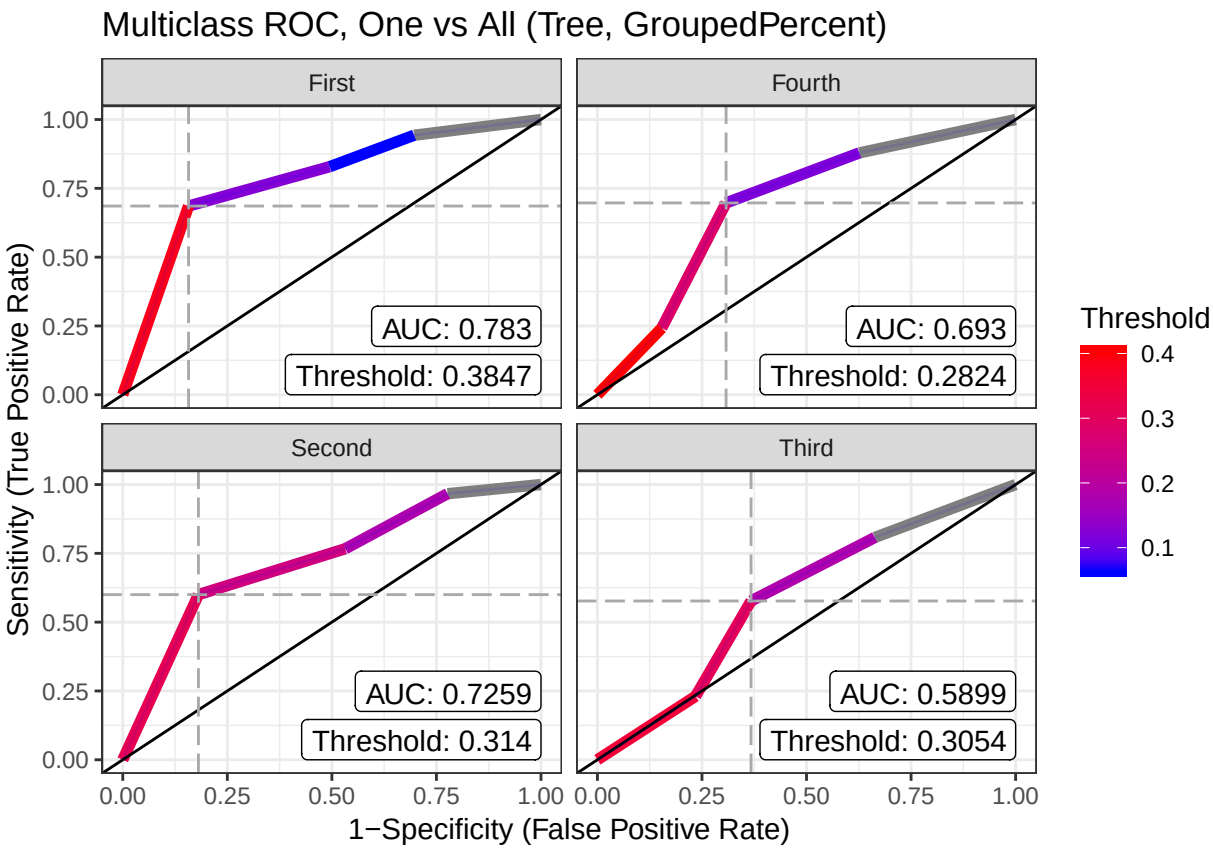


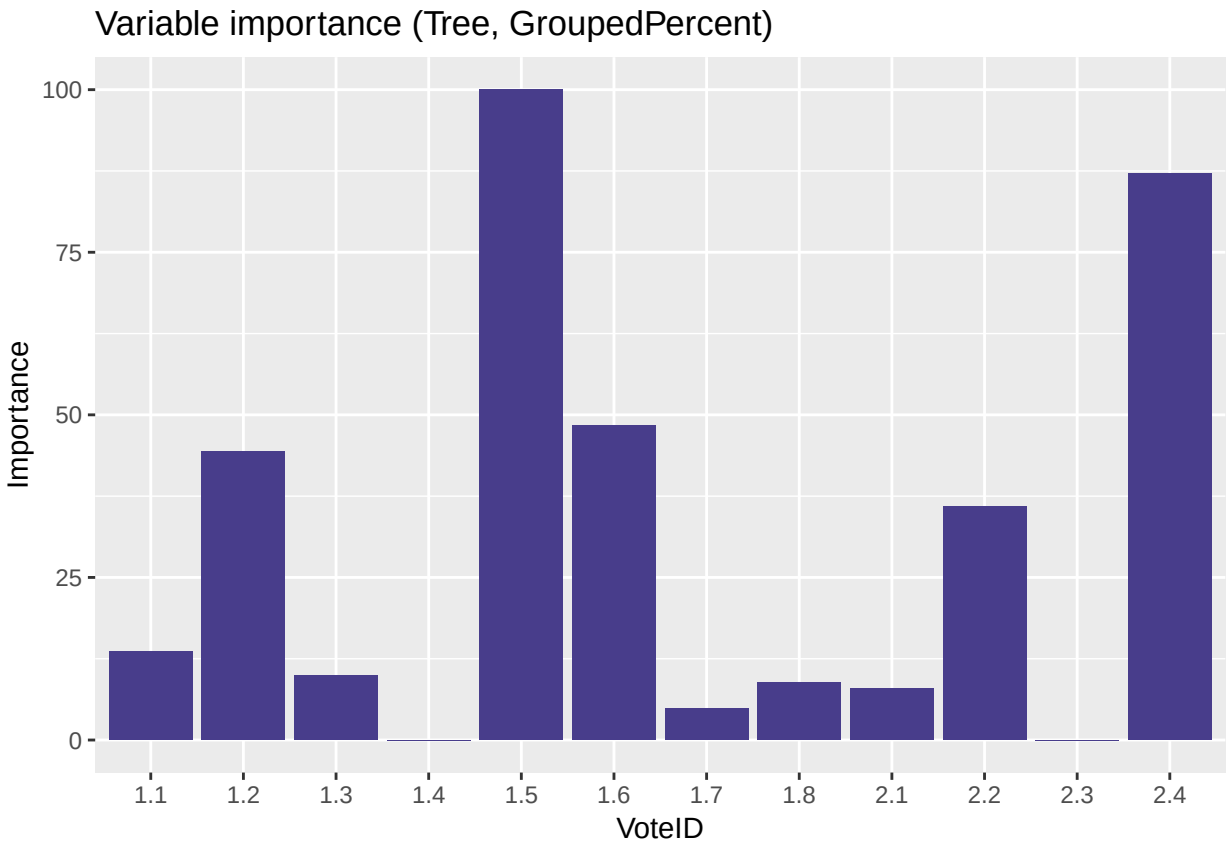

Parameter comparison (Tree, GroupedPercent)
Accuracy (Repeated Cross−Validation)
0.42
0.40
0.38
0.36
0.34
0.05
0.10
0.15
0.20
Complexity Parameter

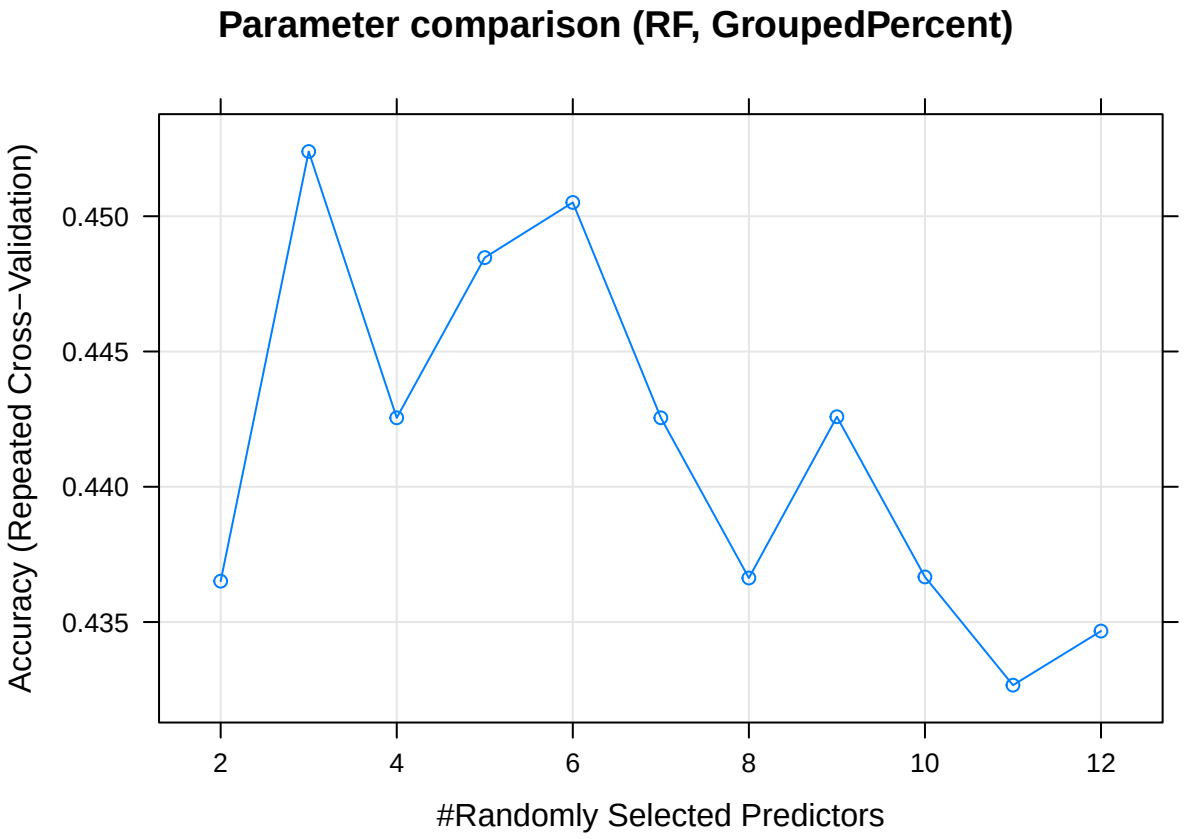


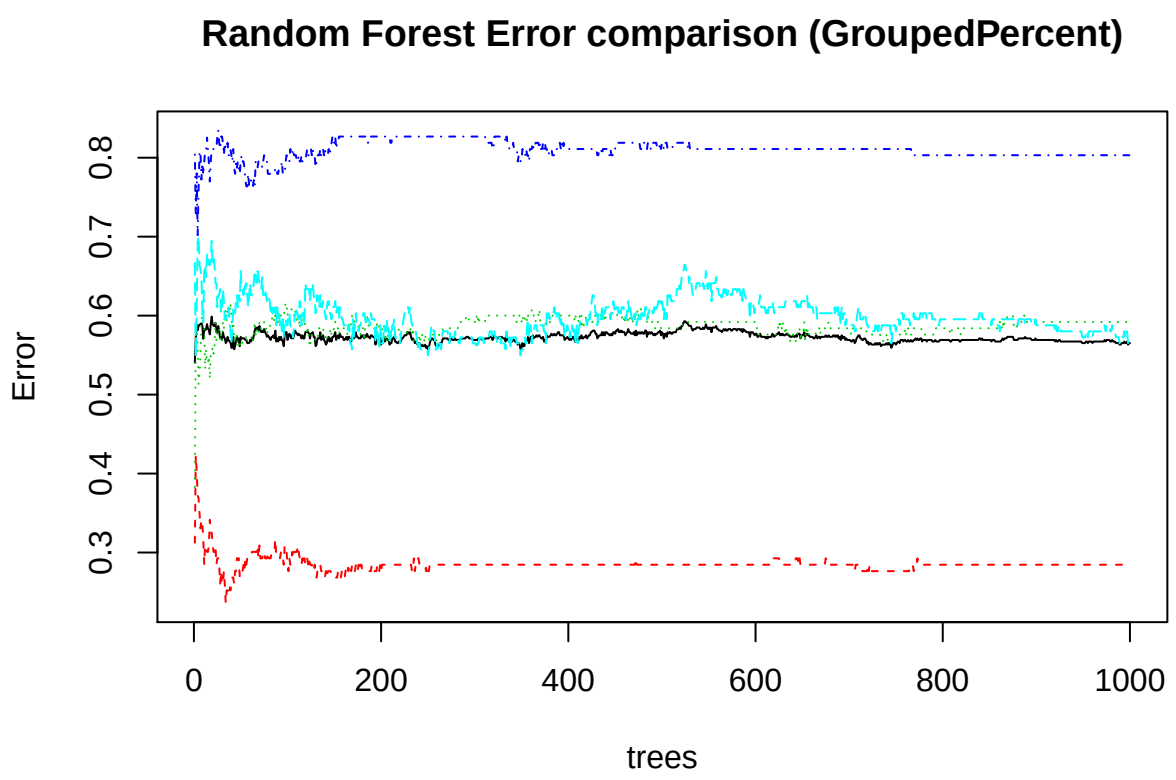

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     24      4      5     5
##     Fourth     6     23      7    15
##     Second     5      6     18     6
##     Third      0      0      0     0
##
## Overall Statistics
##
##                Accuracy : 0.5242
##                  95% CI : (0.4326, 0.6146)
##     No Information Rate : 0.2823
##     P-Value [Acc > NIR] : 1.253e-08
##
##                   Kappa : 0.3533
##
##  Mcnemar's Test P-Value : 0.0001814
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.6857        0.6970        0.6000       0.0000
## Specificity                0.8427        0.6923        0.8191       1.0000
## Pos Pred Value             0.6316        0.4510        0.5143          NaN
```

```
## Neg Pred Value              0.8721       0.8630       0.8652       0.7903
## Prevalence                  0.2823       0.2661       0.2419       0.2097
## Detection Rate              0.1935       0.1855       0.1452       0.0000
## Detection Prevalence        0.3065       0.4113       0.2823       0.0000
## Balanced Accuracy           0.7642       0.6946       0.7096       0.5000
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.6936
```



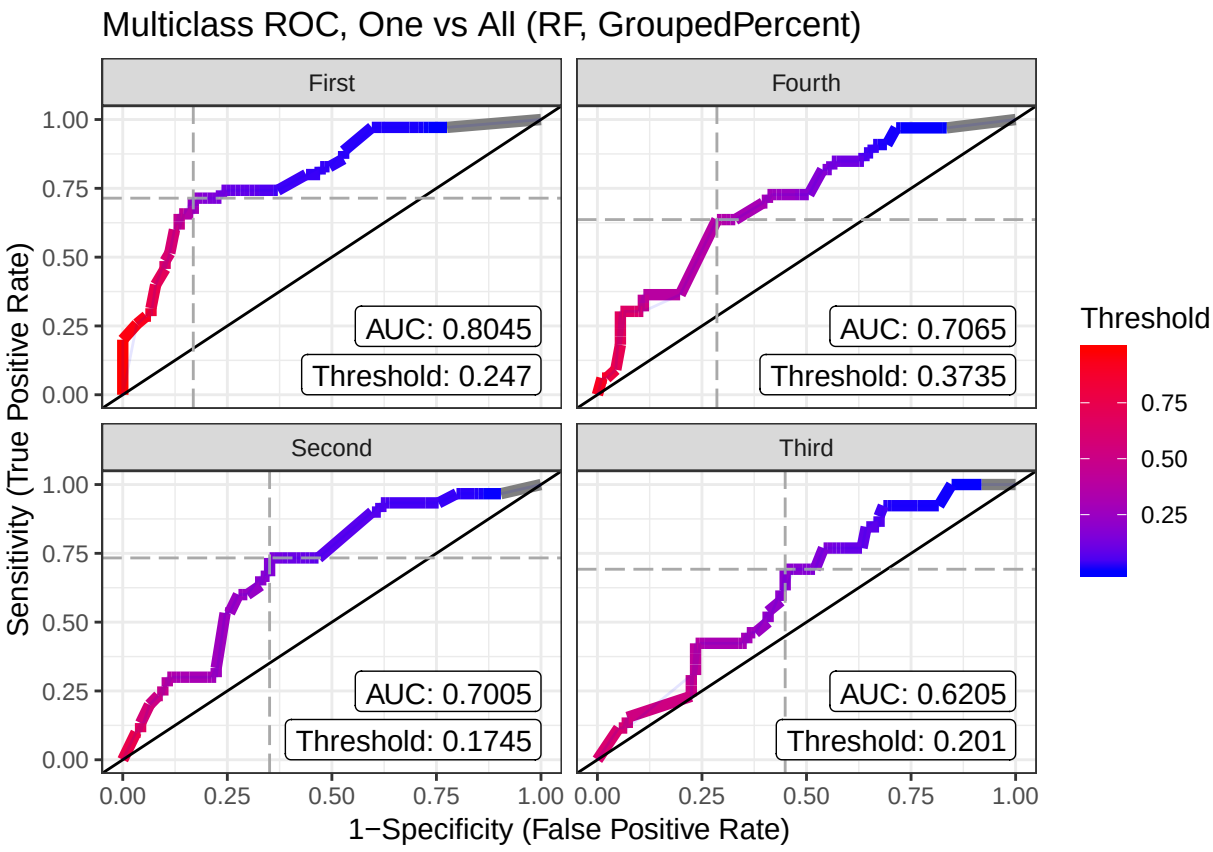

Variable importance (Tree, GroupedPercent)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

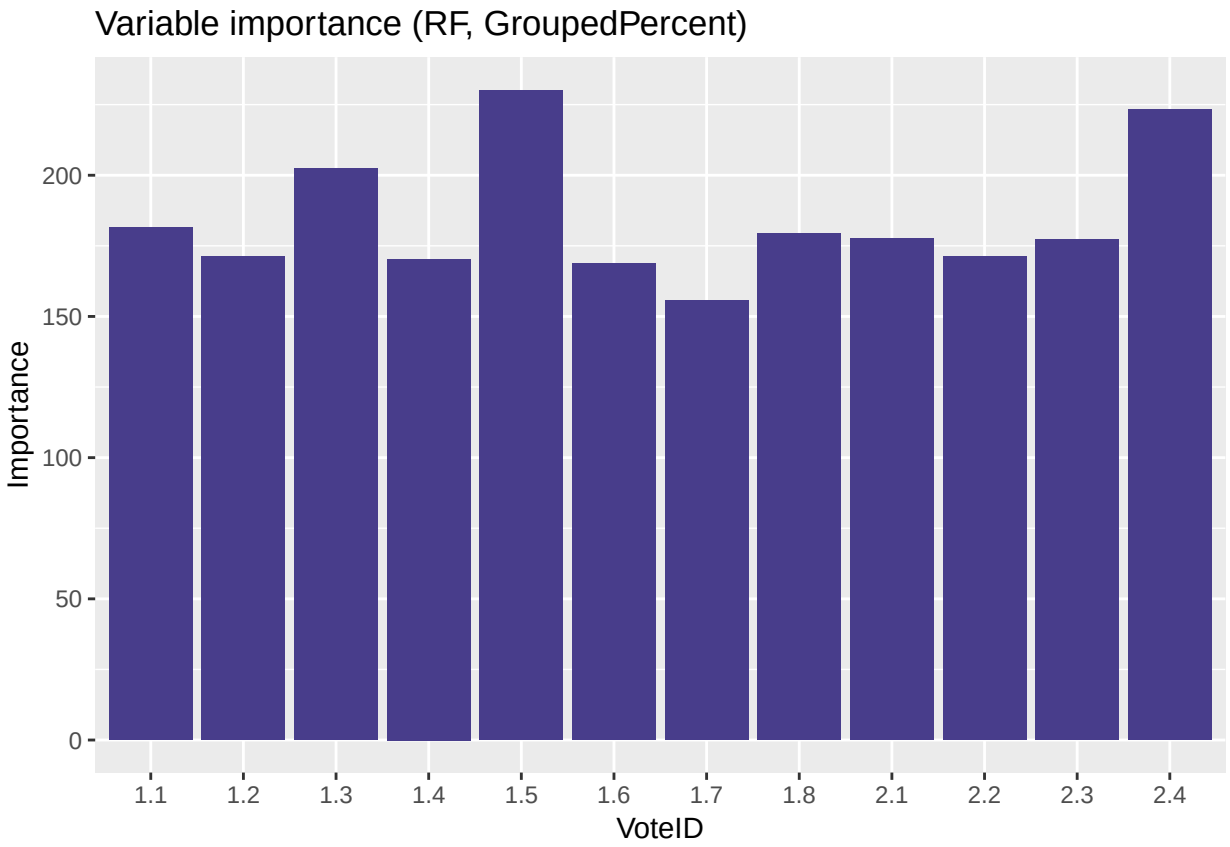

Parameter comparison (RF, GroupedPercent)
Accuracy (Repeated Cross−Validation)
0.450
0.445
0.440
0.435
2
4
6
8
10
12
#Randomly Selected Predictors

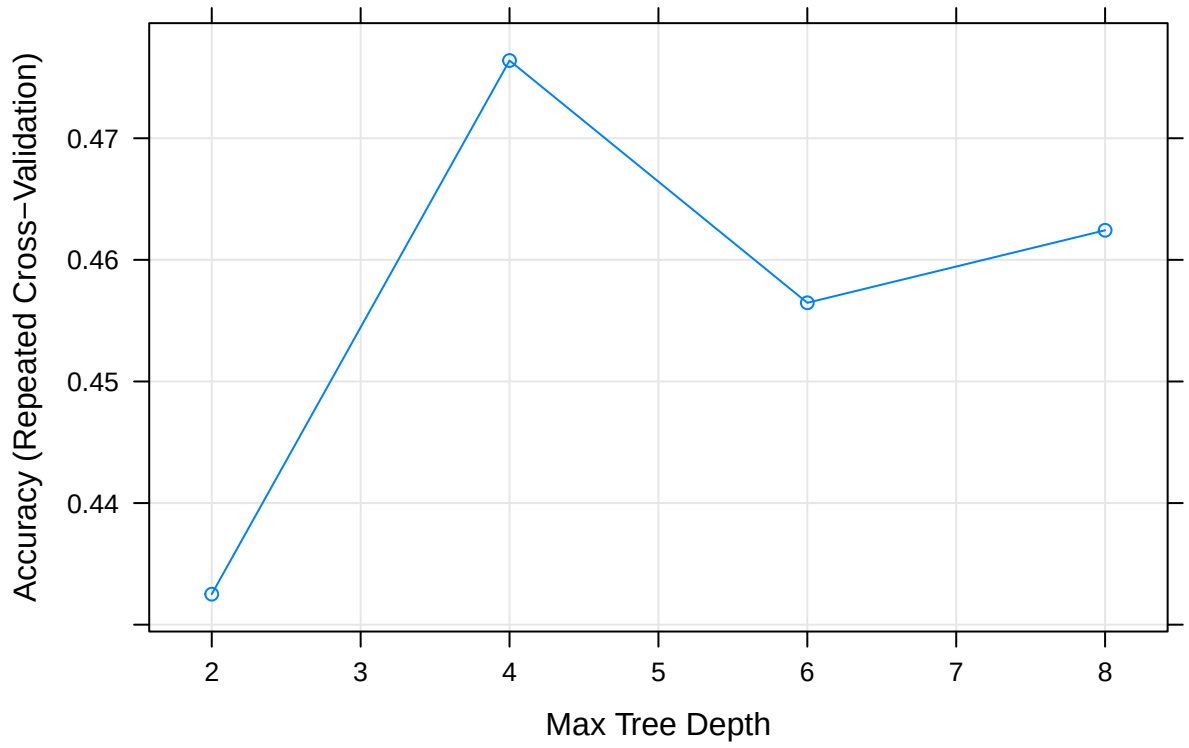



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     23      4      7     4
##     Fourth     4     12      7     8
##     Second     3      5      8     3
##     Third      5     12      8    11
##
## Overall Statistics
##
##                Accuracy : 0.4355
##                  95% CI : (0.3467, 0.5274)
##     No Information Rate : 0.2823
##     P-Value [Acc > NIR] : 0.0001898
##
##                   Kappa : 0.2463
##
##  Mcnemar's Test P-Value : 0.5288753
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.6571       0.36364       0.26667      0.42308
## Specificity                0.8315       0.79121       0.88298      0.74490
## Pos Pred Value             0.6053       0.38710       0.42105      0.30556
```

```
## Neg Pred Value              0.8605       0.77419       0.79048       0.82955
## Prevalence                  0.2823       0.26613       0.24194       0.20968
## Detection Rate              0.1855       0.09677       0.06452       0.08871
## Detection Prevalence        0.3065       0.25000       0.15323       0.29032
## Balanced Accuracy           0.7443       0.57742       0.57482       0.58399
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.7041
```

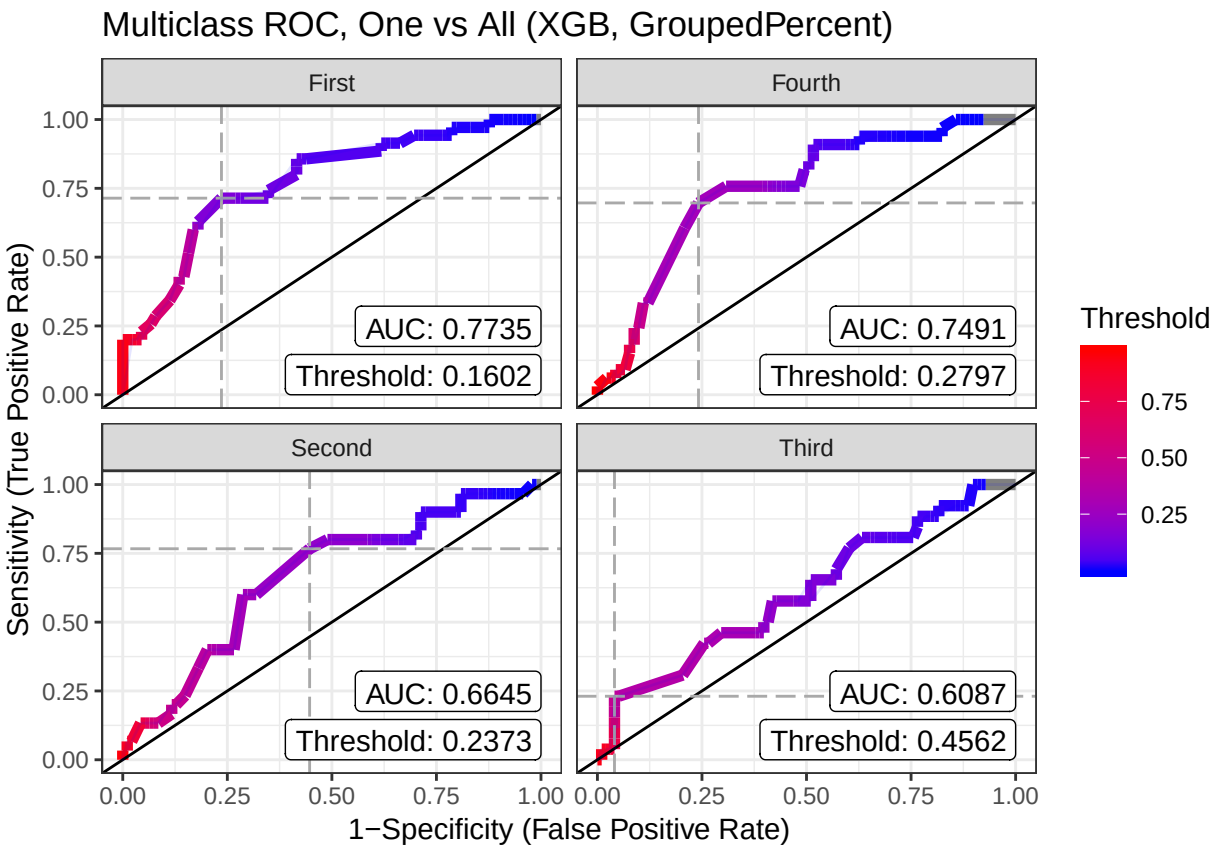


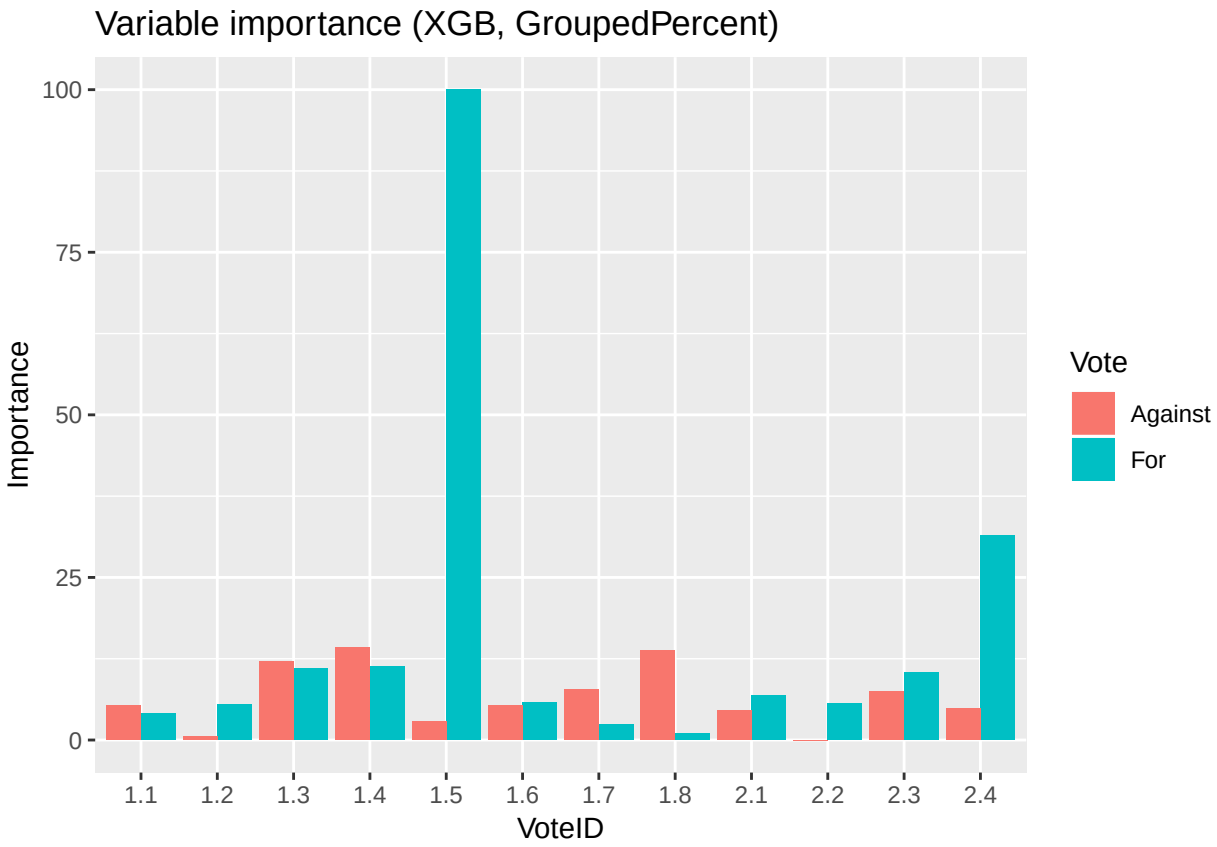

Variable importance (RF, GroupedPercent)
Importance
200
150
100
50
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

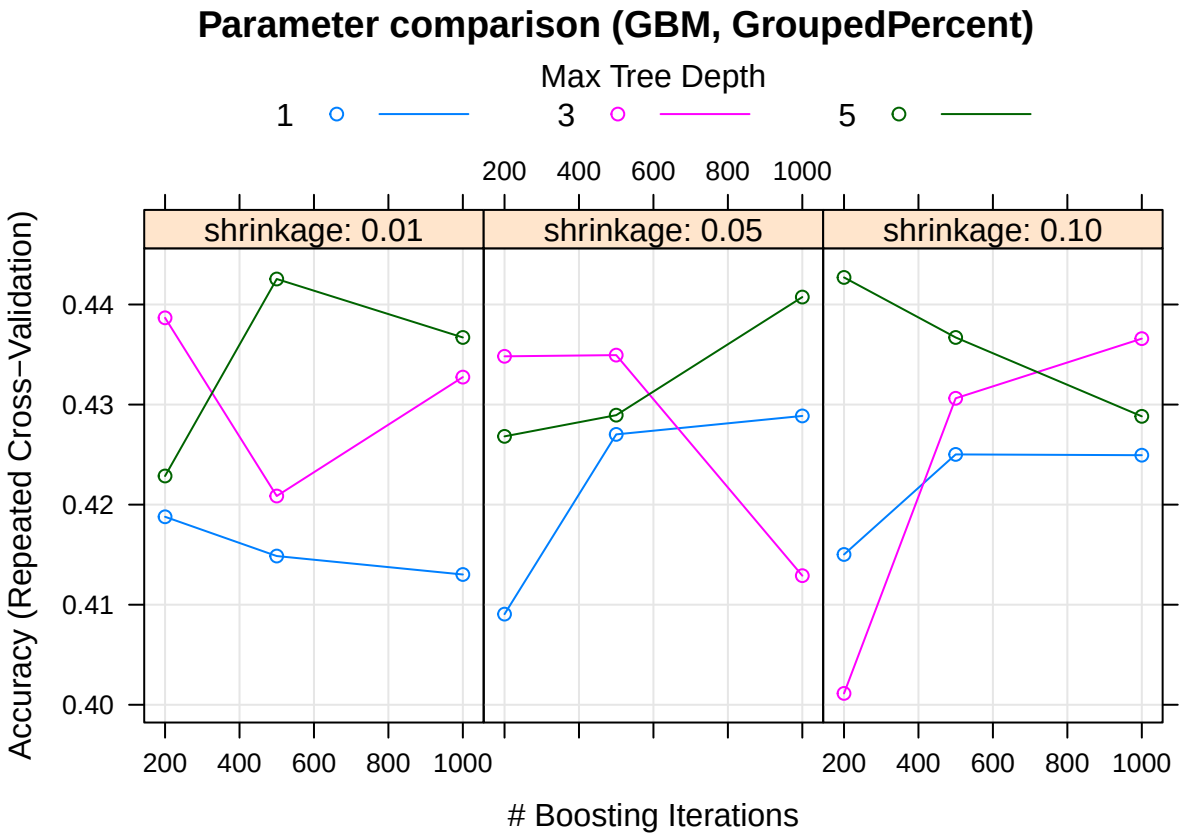



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     18      4      6     4
##     Fourth     1     11      1     8
##     Second     9      6     10     3
##     Third      7     12     13    11
##
## Overall Statistics
##
##                Accuracy : 0.4032
##                  95% CI : (0.3161, 0.4951)
##     No Information Rate : 0.2823
##     P-Value [Acc > NIR] : 0.00249
##
##                   Kappa : 0.2093
##
##  Mcnemar's Test P-Value : 0.03148
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.5143       0.33333       0.33333      0.42308
## Specificity                0.8427       0.89011       0.80851      0.67347
## Pos Pred Value             0.5625       0.52381       0.35714      0.25581
```

```
## Neg Pred Value              0.8152       0.78641       0.79167       0.81481
## Prevalence                  0.2823       0.26613       0.24194       0.20968
## Detection Rate              0.1452       0.08871       0.08065       0.08871
## Detection Prevalence        0.2581       0.16935       0.22581       0.34677
## Balanced Accuracy           0.6785       0.61172       0.57092       0.54827
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.6966
```

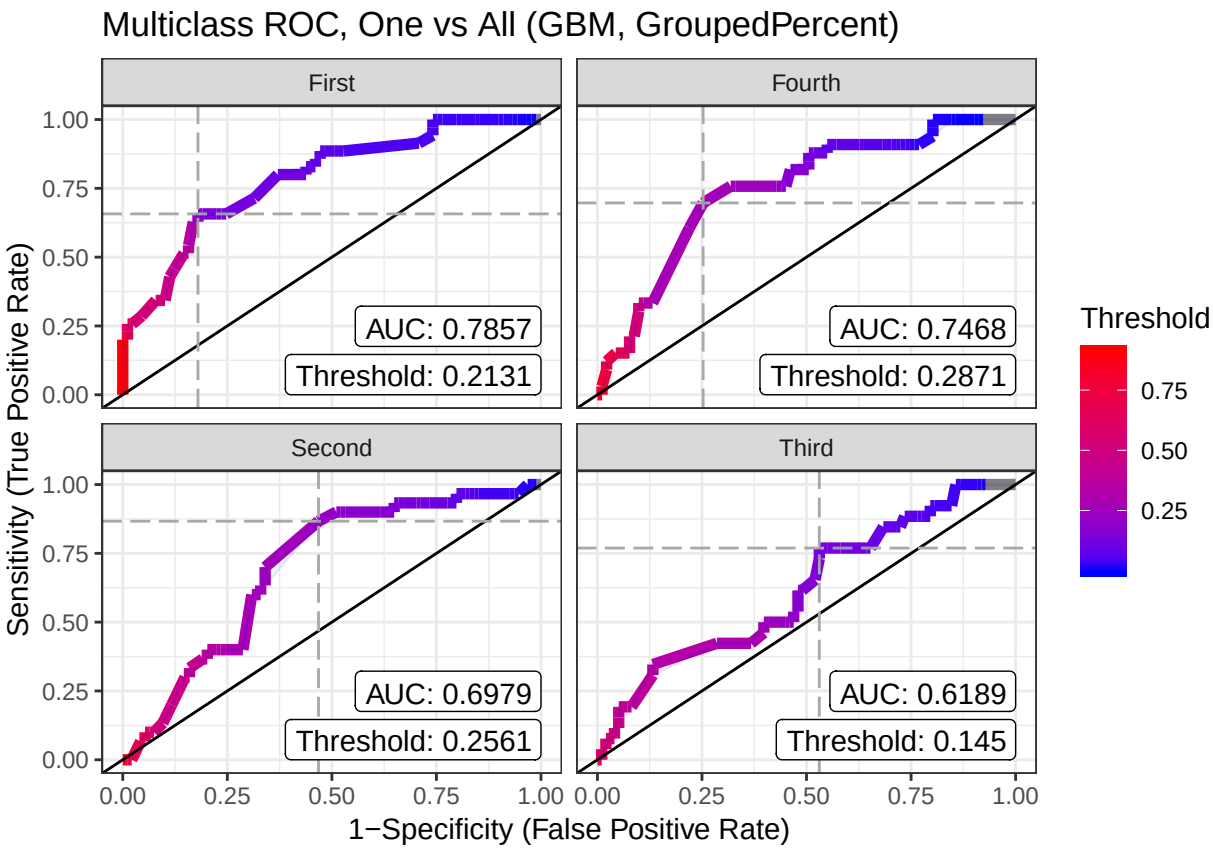


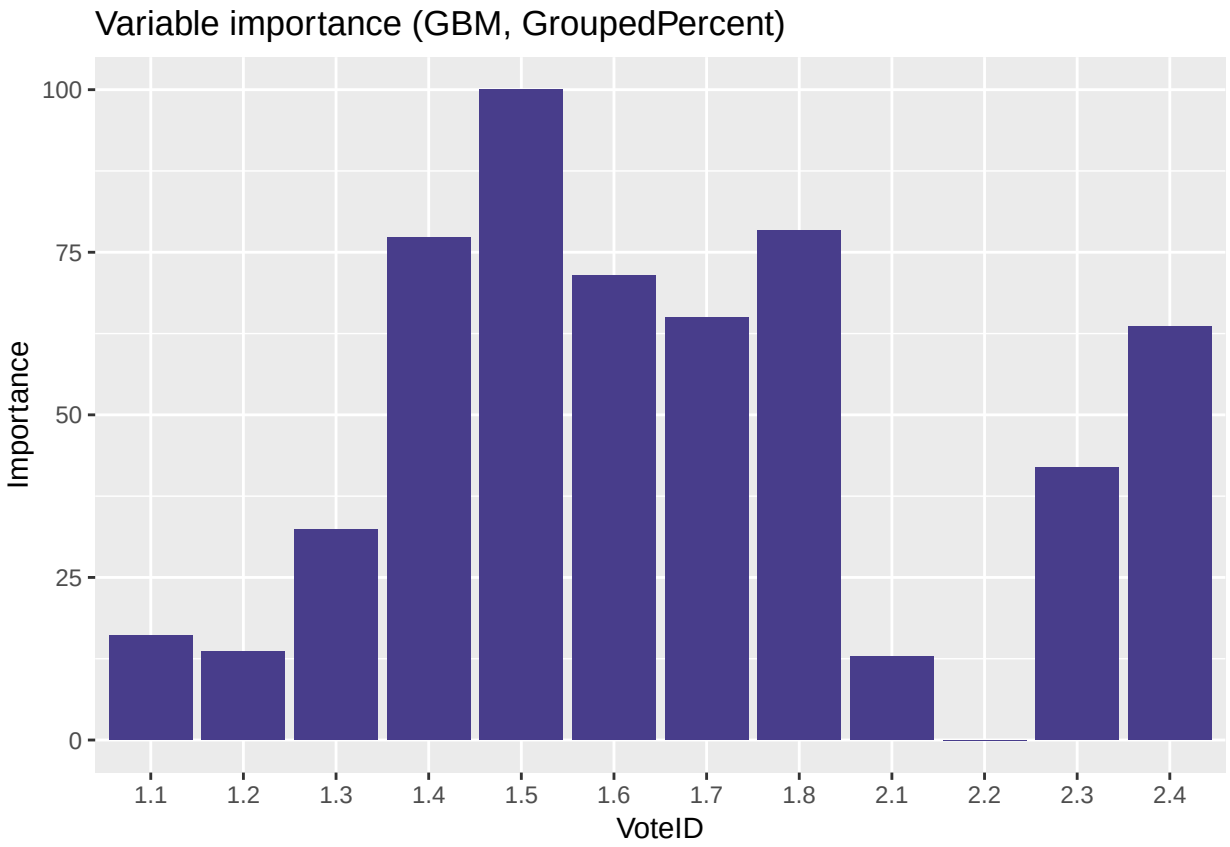

Variable importance (XGB, GroupedPercent)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For

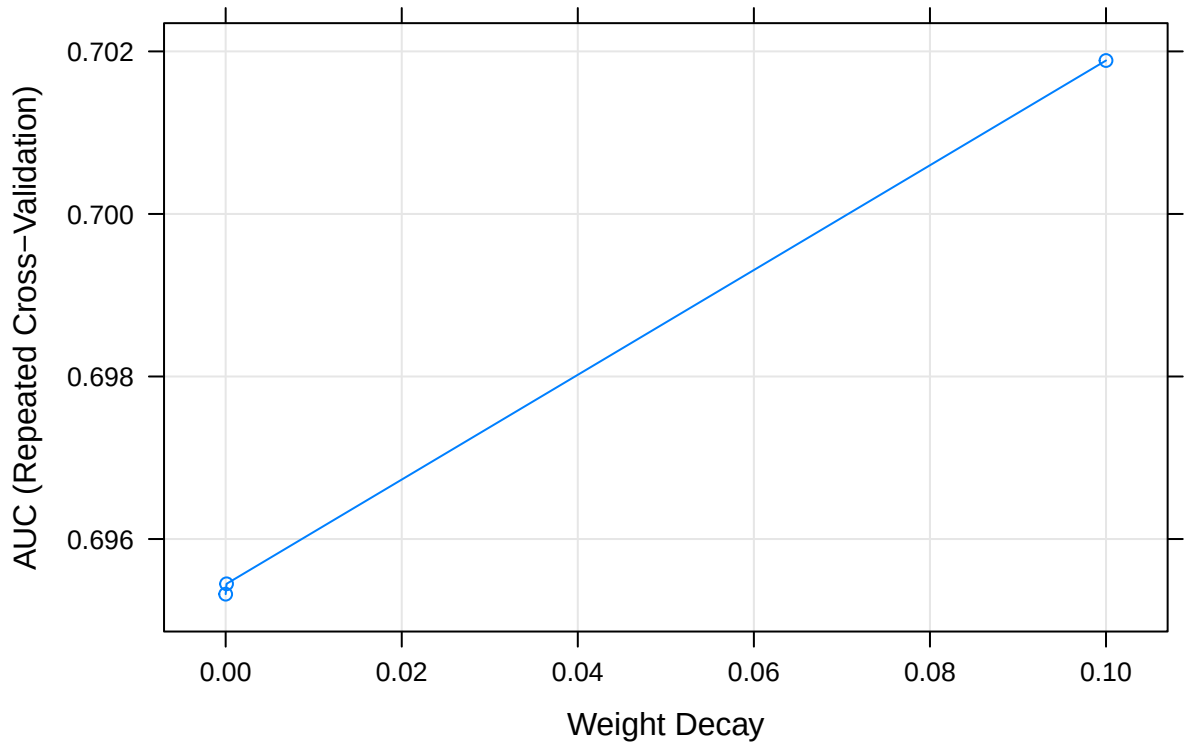



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     19      4      5     4
##     Fourth     1     11      1     8
##     Second     6      6      9     4
##     Third      9     12     15    10
##
## Overall Statistics
##
##                Accuracy : 0.3952
##                  95% CI : (0.3086, 0.4869)
##     No Information Rate : 0.2823
##     P-Value [Acc > NIR] : 0.004389
##
##                   Kappa : 0.1994
##
##  Mcnemar's Test P-Value : 0.024026
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.5429       0.33333       0.30000      0.38462
## Specificity                0.8539       0.89011       0.82979      0.63265
## Pos Pred Value             0.5938       0.52381       0.36000      0.21739
```

```
## Neg Pred Value              0.8261       0.78641       0.78788       0.79487
## Prevalence                  0.2823       0.26613       0.24194       0.20968
## Detection Rate              0.1532       0.08871       0.07258       0.08065
## Detection Prevalence        0.2581       0.16935       0.20161       0.37097
## Balanced Accuracy           0.6984       0.61172       0.56489       0.50863
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.7087
```

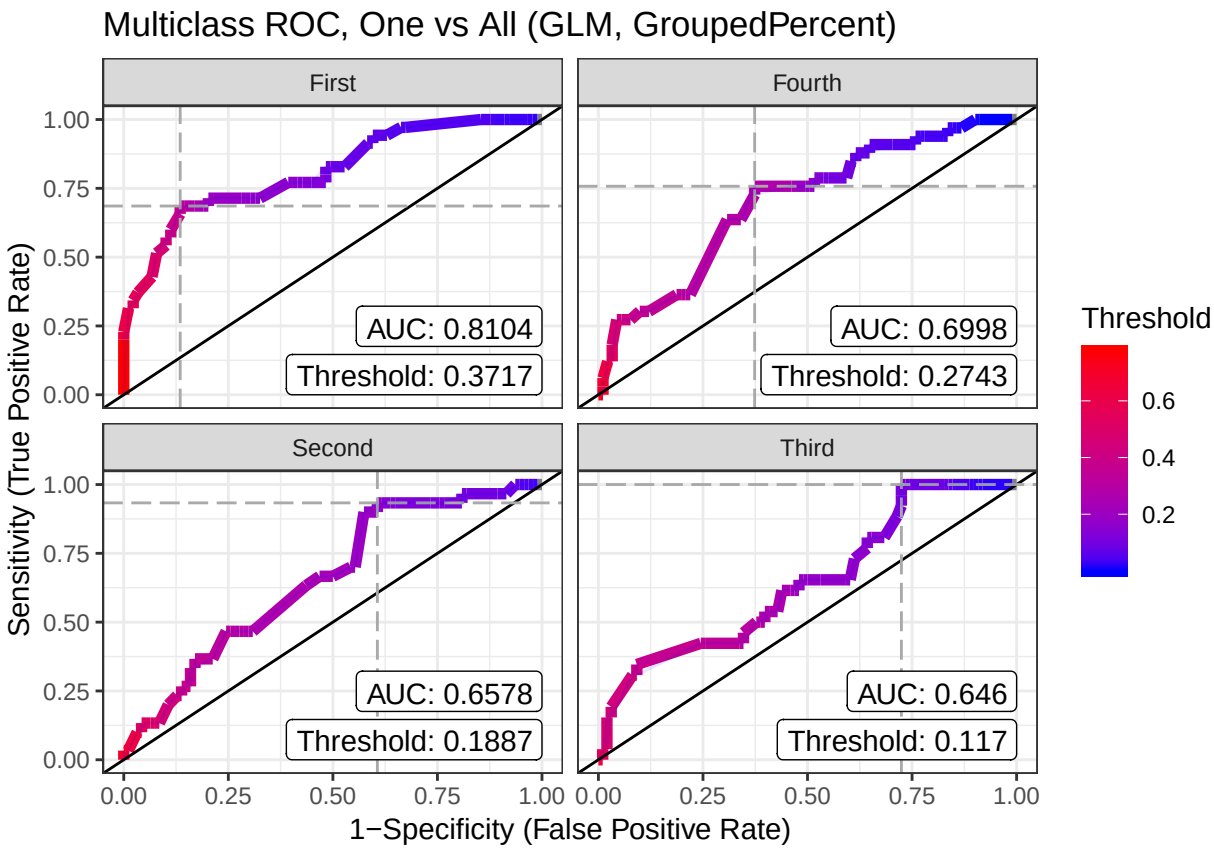



Variable importance (GBM, GroupedPercent)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID



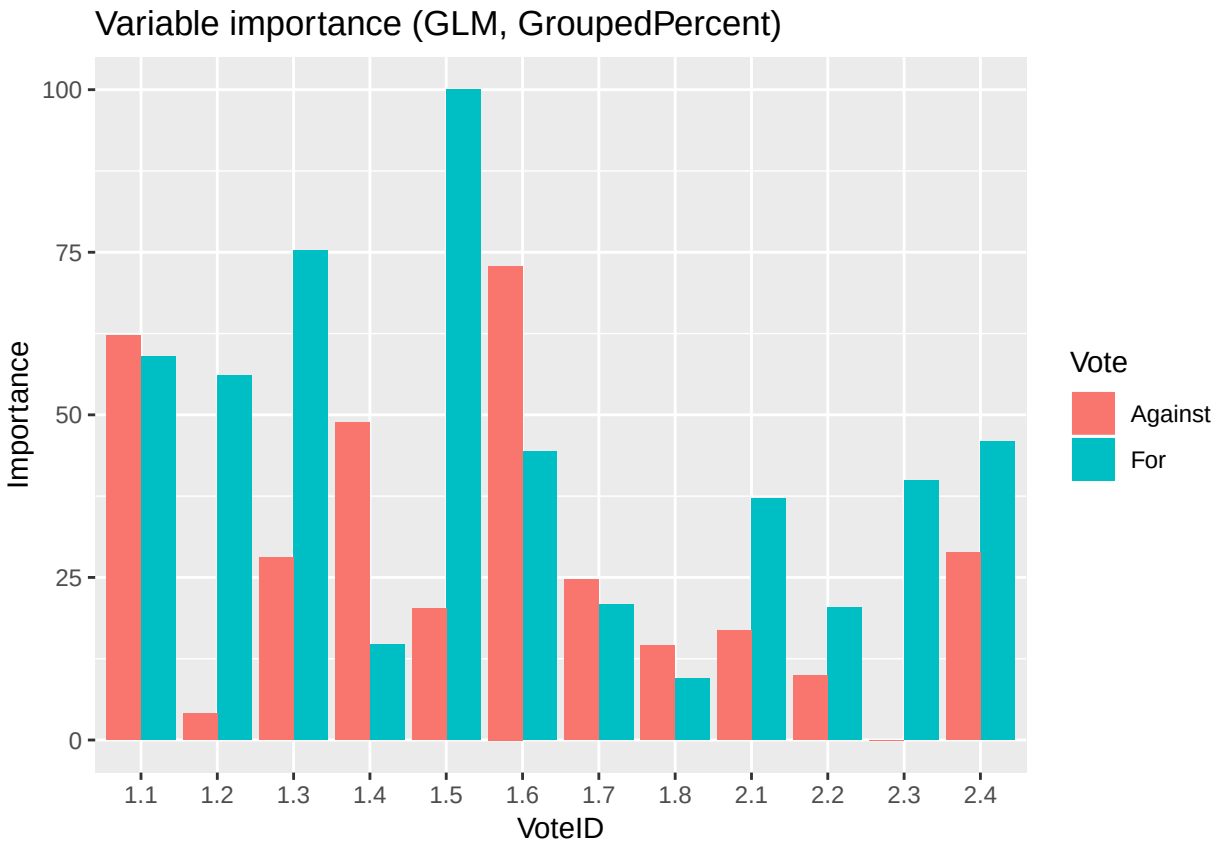

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     20      4      2     4
##     Fourth     1     10      2     6
##     Second     7      6     14     5
##     Third      7     13     12    11
##
## Overall Statistics
##
##                Accuracy : 0.4435
##                  95% CI : (0.3544, 0.5355)
##     No Information Rate : 0.2823
##     P-Value [Acc > NIR] : 9.234e-05
##
##                   Kappa : 0.2637
##
##  Mcnemar's Test P-Value : 0.04536
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.5714       0.30303        0.4667      0.42308
## Specificity                0.8876       0.90110        0.8085      0.67347
## Pos Pred Value             0.6667       0.52632        0.4375      0.25581
```

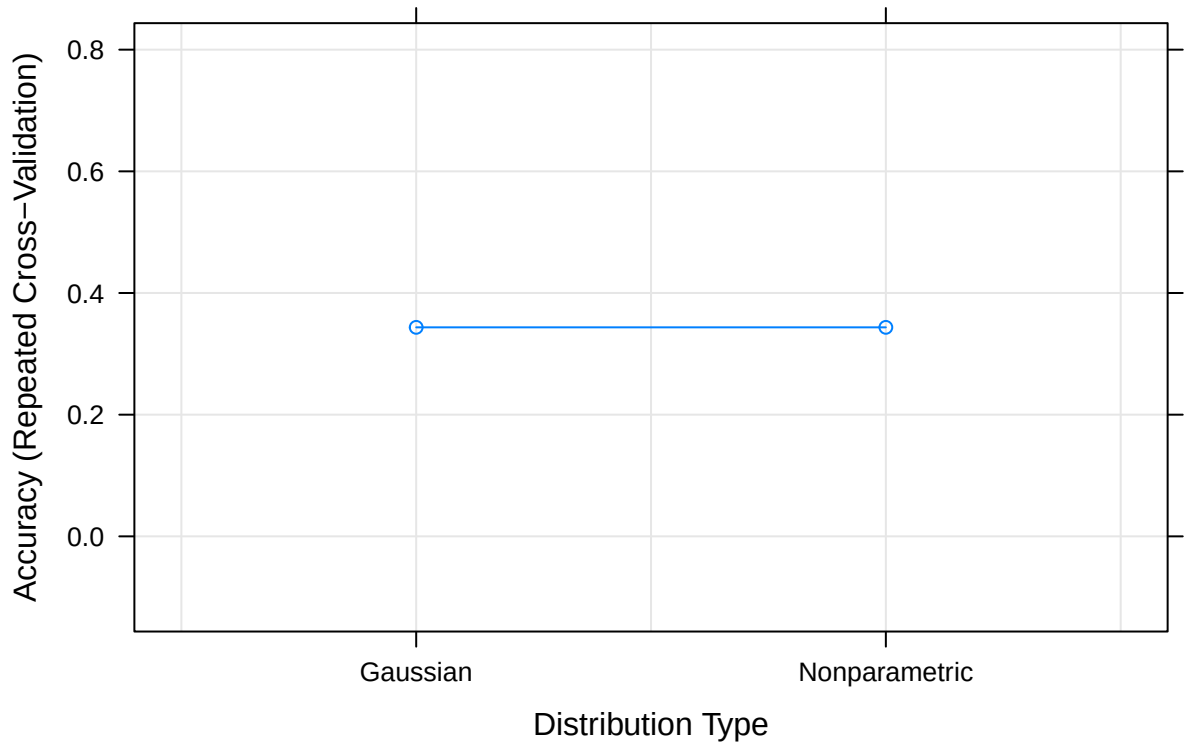
```
## Neg Pred Value              0.8404       0.78095       0.8261       0.81481
## Prevalence                  0.2823       0.26613       0.2419       0.20968
## Detection Rate              0.1613       0.08065       0.1129       0.08871
## Detection Prevalence        0.2419       0.15323       0.2581       0.34677
## Balanced Accuracy           0.7295       0.60206       0.6376       0.54827
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.6991
```




```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101
```

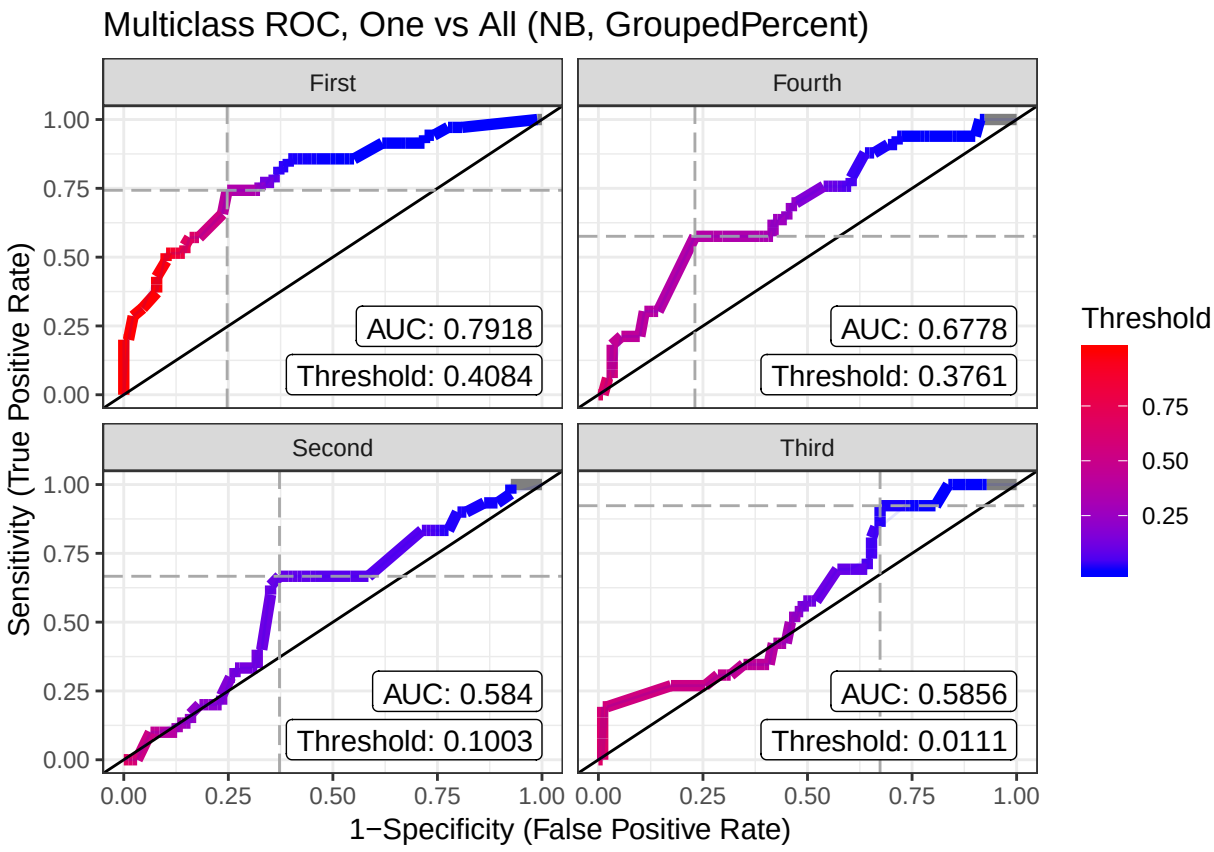
```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126
```

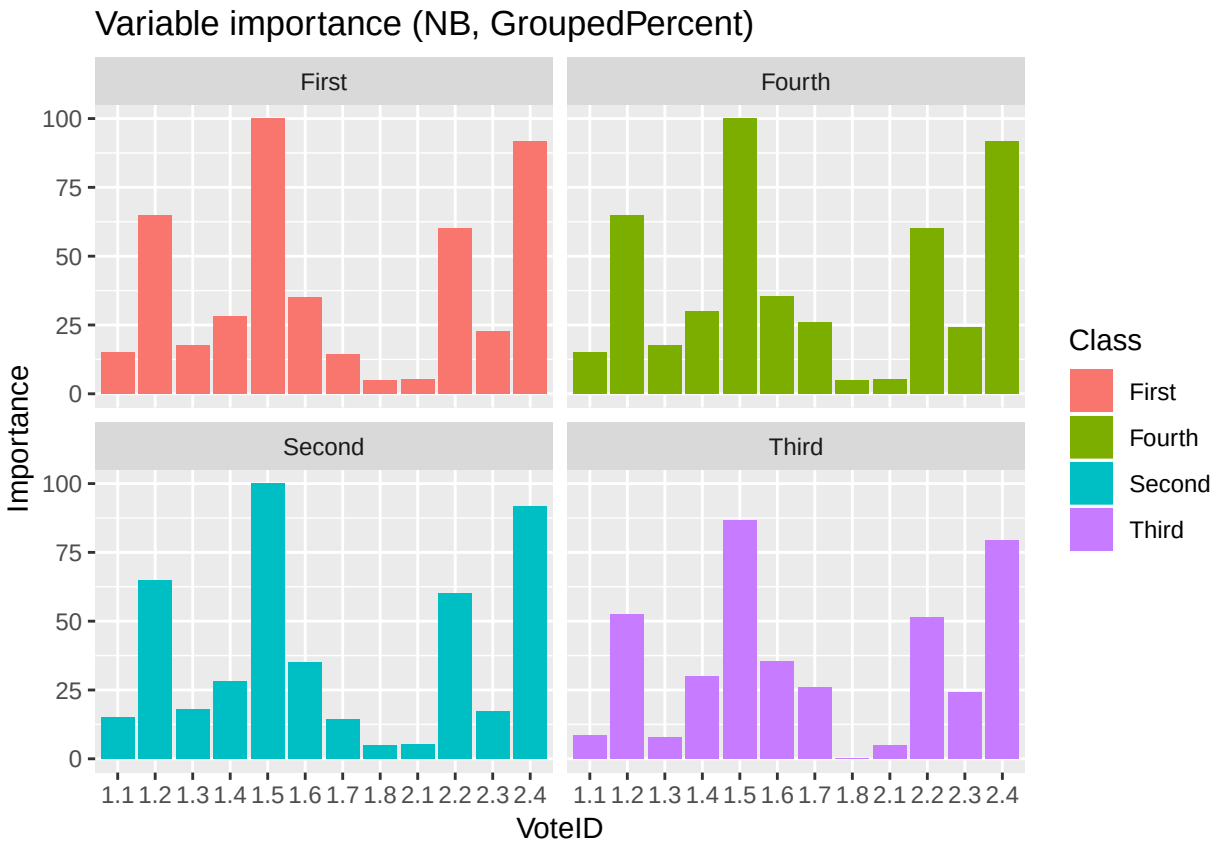

Variable importance (GLM, GroupedPercent)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For

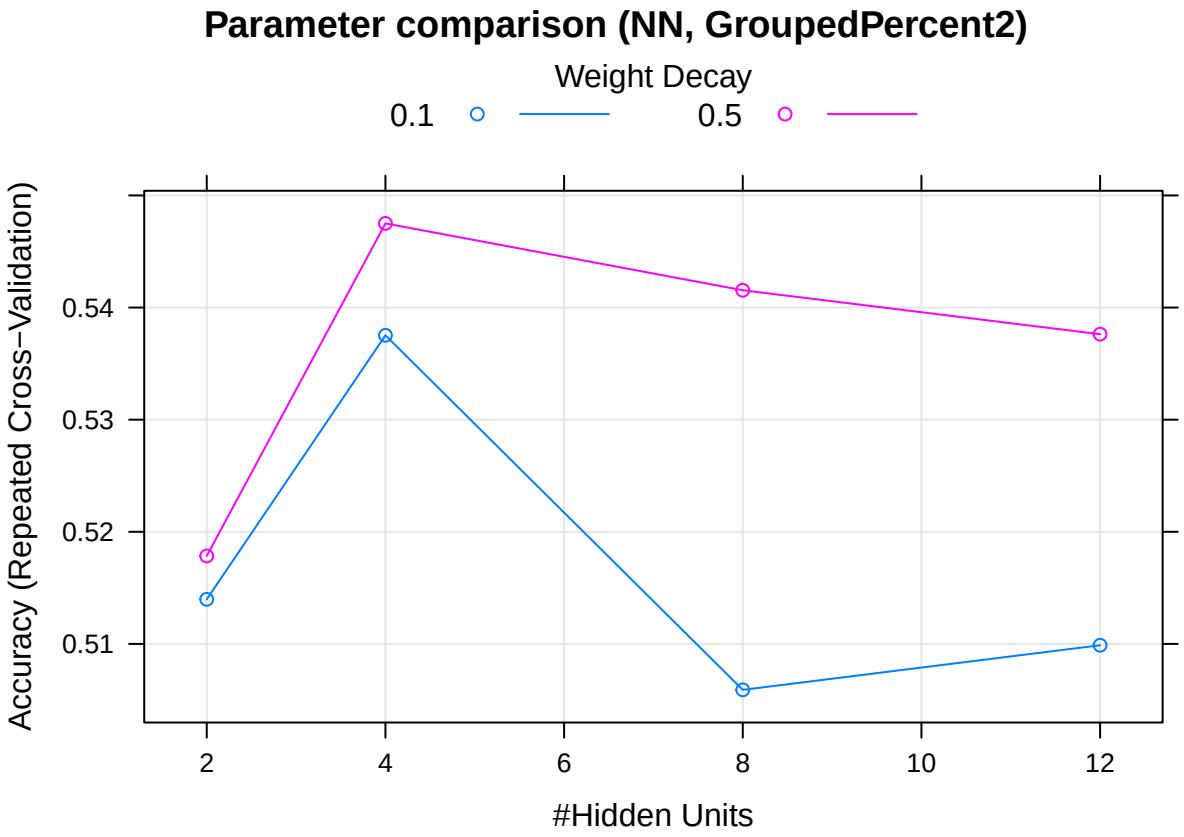



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction First Fourth Second Third
##     First     26      6      5    11
##     Fourth     1      7      3     6
##     Second     3      5      4     0
##     Third      5     15     18     9
##
## Overall Statistics
##
##                Accuracy : 0.371
##                  95% CI : (0.286, 0.4623)
##     No Information Rate : 0.2823
##     P-Value [Acc > NIR] : 0.02
##
##                   Kappa : 0.1628
##
##  Mcnemar's Test P-Value : 6.998e-05
##
## Statistics by Class:
##
##                      Class: First Class: Fourth Class: Second Class: Third
## Sensitivity                0.7429       0.21212       0.13333      0.34615
## Specificity                0.7528       0.89011       0.91489      0.61224
## Pos Pred Value             0.5417       0.41176       0.33333      0.19149
```

```
## Neg Pred Value              0.8816       0.75701       0.76786       0.77922
## Prevalence                  0.2823       0.26613       0.24194       0.20968
## Detection Rate              0.2097       0.05645       0.03226       0.07258
## Detection Prevalence        0.3871       0.13710       0.09677       0.37903
## Balanced Accuracy           0.7478       0.55112       0.52411       0.47920
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 4 levels of response: First, Fourth, Second, Third.
## Multi-class area under the curve: 0.6534
```

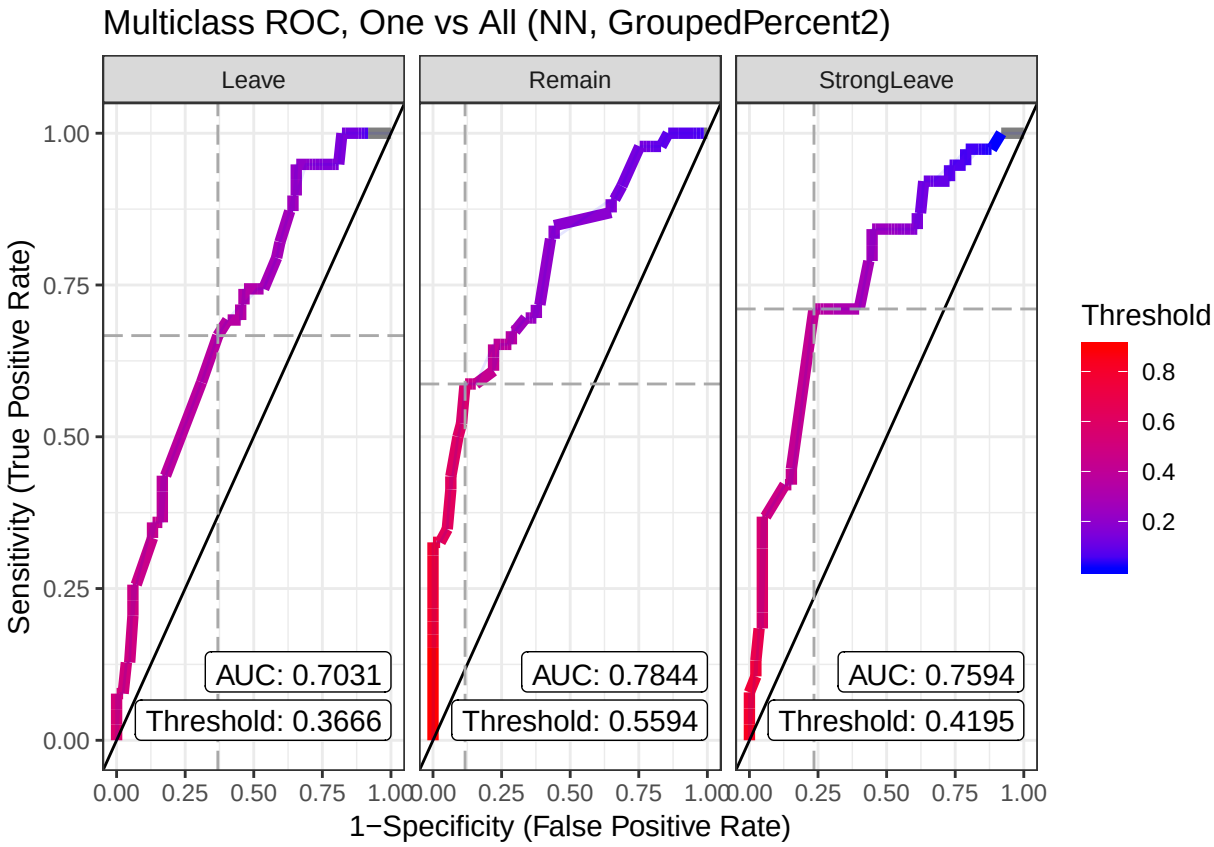


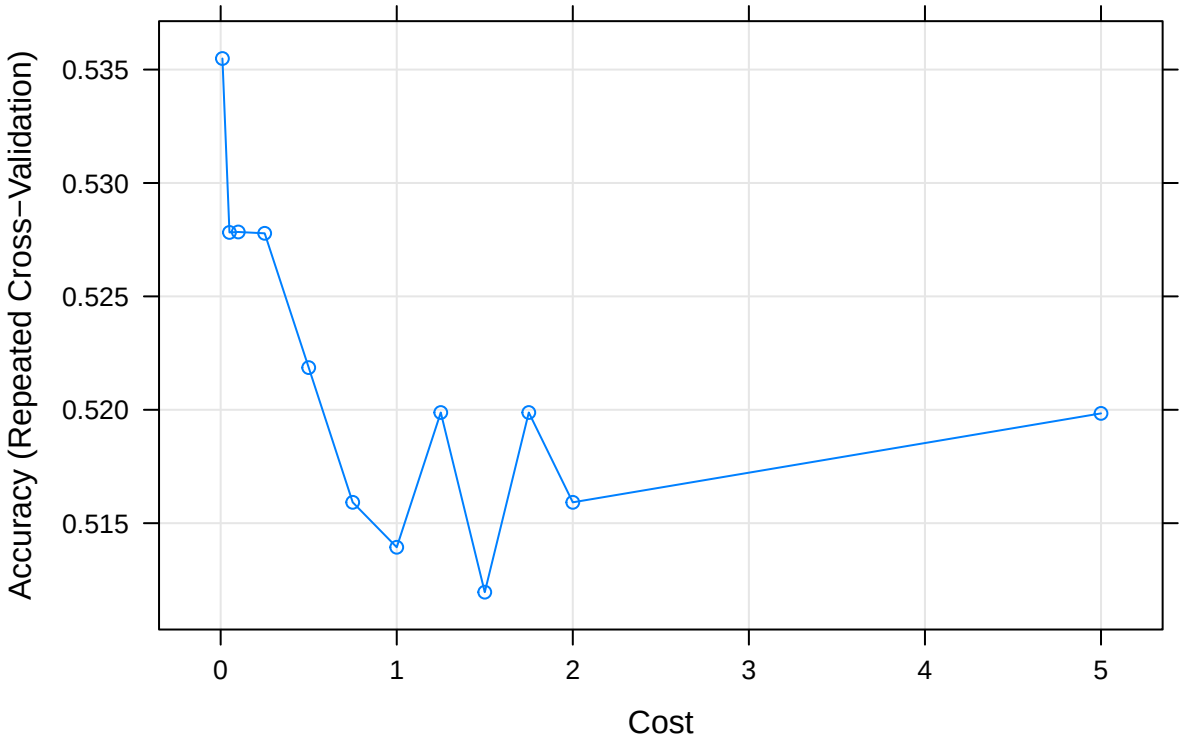

Variable importance (NB, GroupedPercent)
First
Fourth
Second
Third
Importance
100
75
50
25
0
1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 2.1 2.2 2.3 2.4
VoteID
Class
First
Fourth
Second
Third




```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          13     11           2
##   Remain         12     29           9
##   StrongLeave    14      6          27
##
## Overall Statistics
##
##                Accuracy : 0.561
##                  95% CI : (0.4687, 0.6503)
##     No Information Rate : 0.374
##     P-Value [Acc > NIR] : 1.947e-05
##
##                   Kappa : 0.3377
##
##  Mcnemar's Test P-Value : 0.02185
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity                0.3333        0.6304              0.7105
## Specificity                0.8452        0.7273              0.7647
## Pos Pred Value             0.5000        0.5800              0.5745
## Neg Pred Value             0.7320        0.7671              0.8553
```

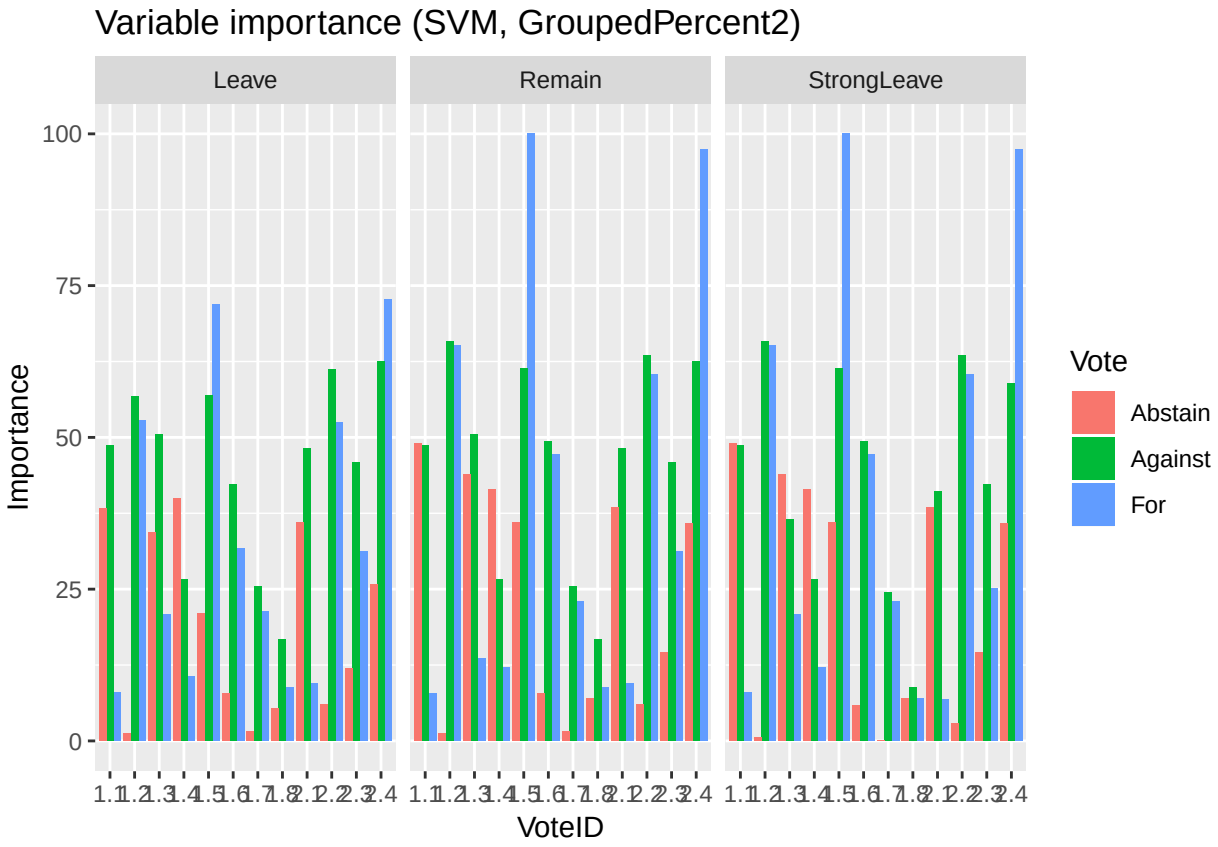
```
## Prevalence                0.3171          0.3740          0.3089
## Detection Rate            0.1057          0.2358          0.2195
## Detection Prevalence      0.2114          0.4065          0.3821
## Balanced Accuracy         0.5893          0.6789          0.7376
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.7469
```

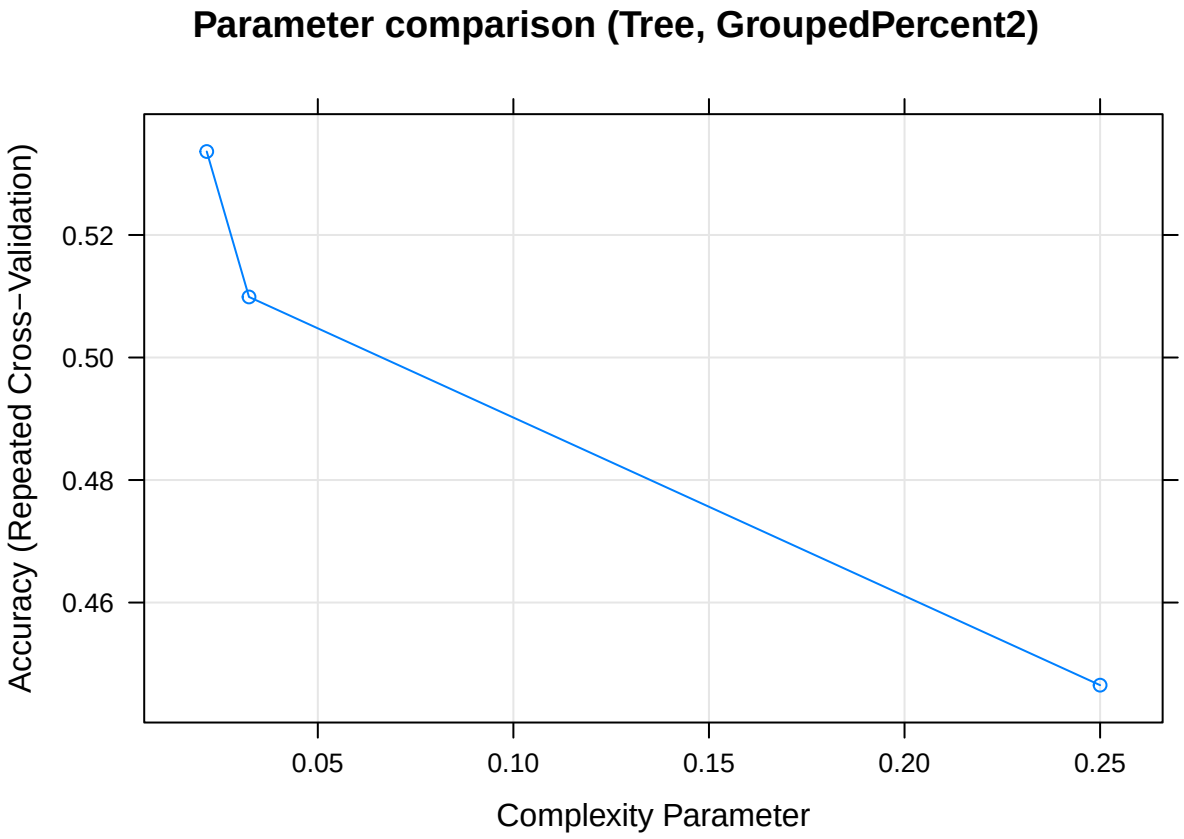


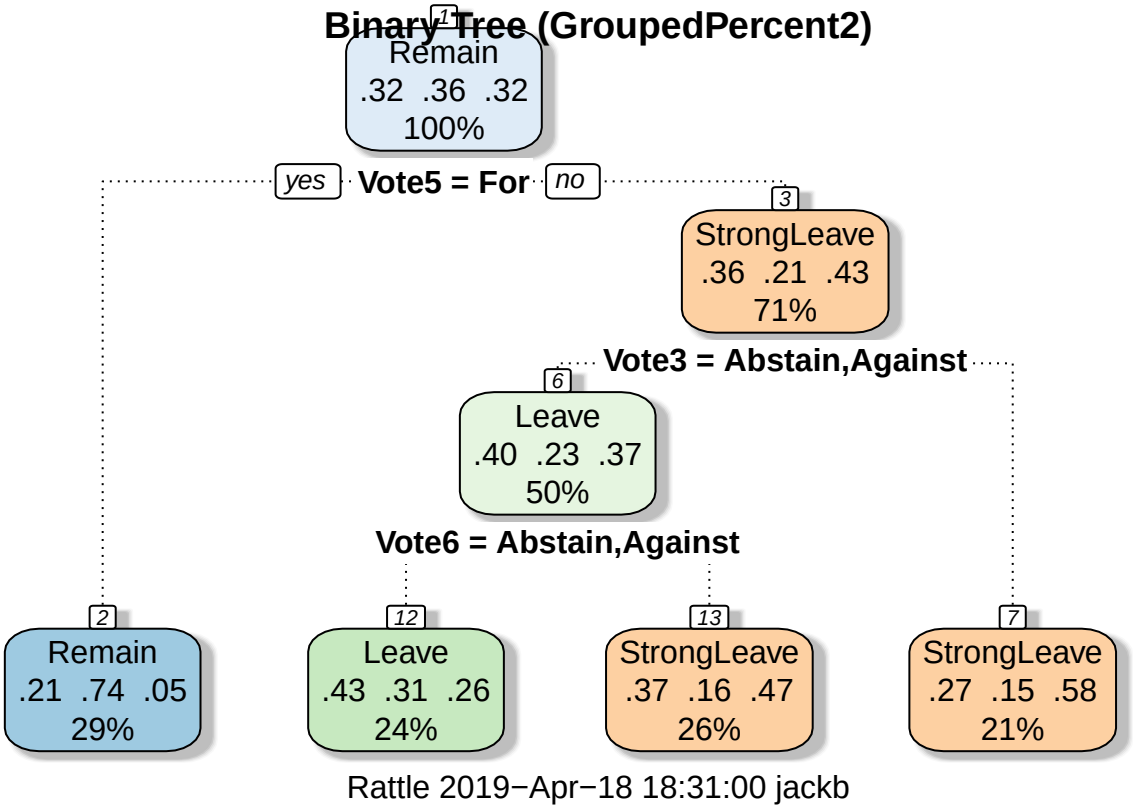



```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          16      9           3
##   Remain         11     29          10
##   StrongLeave    12      8          25
##
## Overall Statistics
##
##                Accuracy : 0.5691
##                  95% CI : (0.4768, 0.658)
##     No Information Rate : 0.374
##     P-Value [Acc > NIR] : 8.752e-06
##
##                   Kappa : 0.3499
##
##  Mcnemar's Test P-Value : 0.1206
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity                0.4103        0.6304              0.6579
## Specificity                0.8571        0.7273              0.7647
## Pos Pred Value             0.5714        0.5800              0.5556
## Neg Pred Value             0.7579        0.7671              0.8333
```

```
## Prevalence                0.3171          0.3740               0.3089
## Detection Rate            0.1301          0.2358               0.2033
## Detection Prevalence      0.2276          0.4065               0.3659
## Balanced Accuracy         0.6337          0.6789               0.7113
```

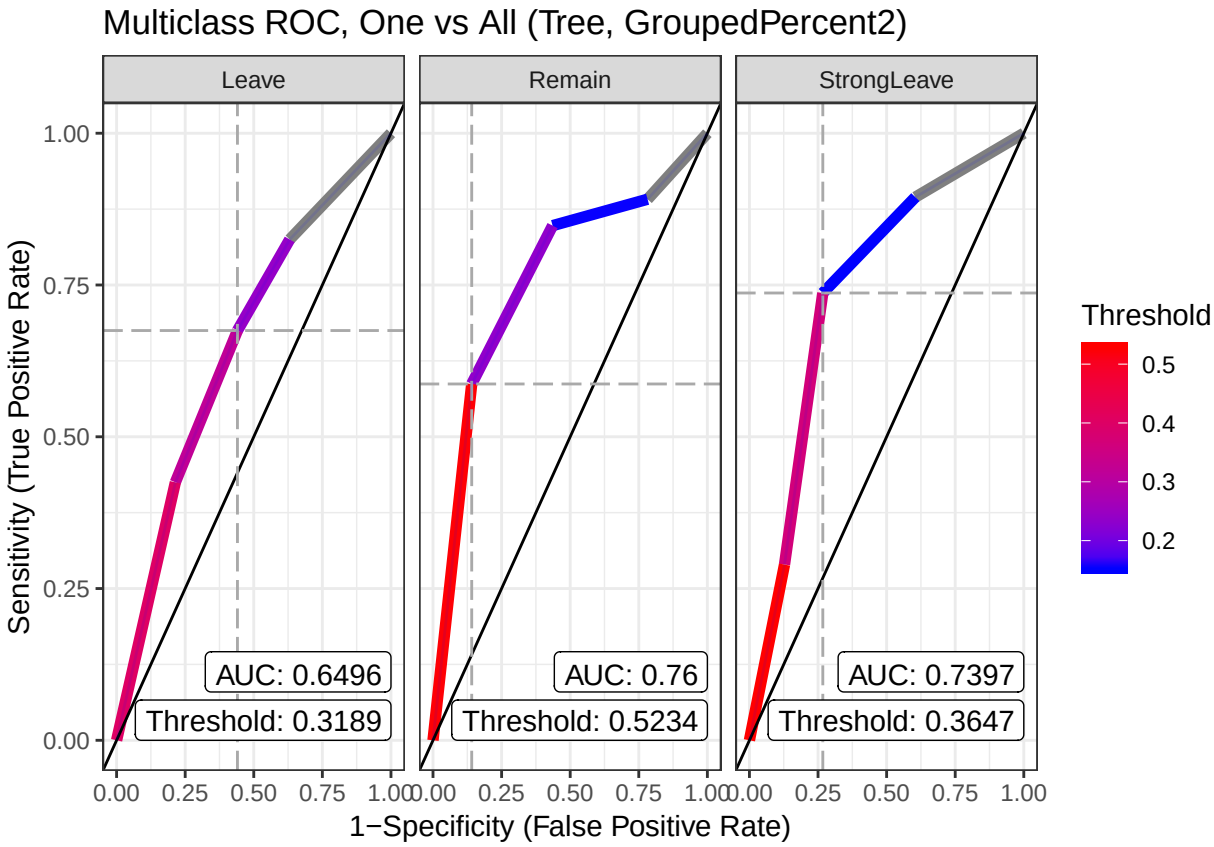


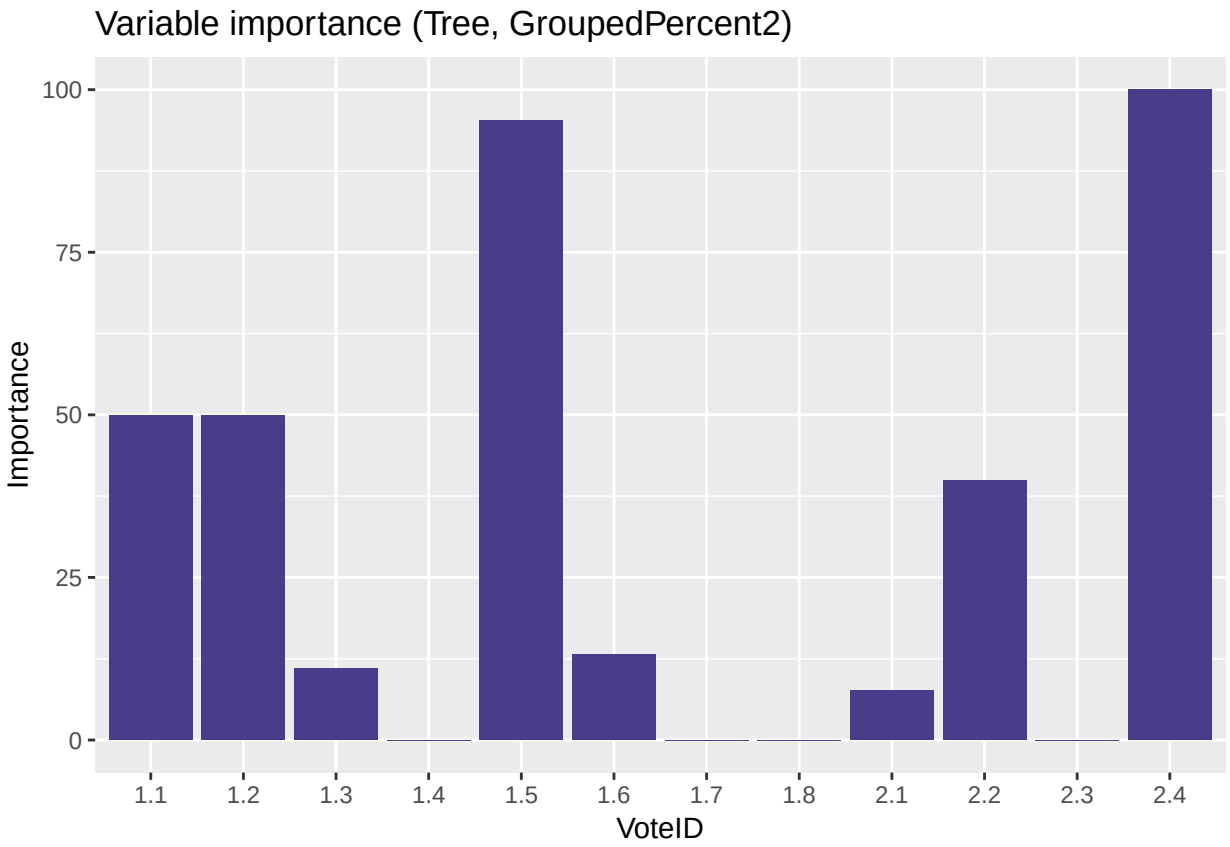

Parameter comparison (Tree, GroupedPercent2)
Accuracy (Repeated Cross−Validation)
0.52
0.50
0.48
0.46
0.05
0.10
0.15
0.20
0.25
Complexity Parameter

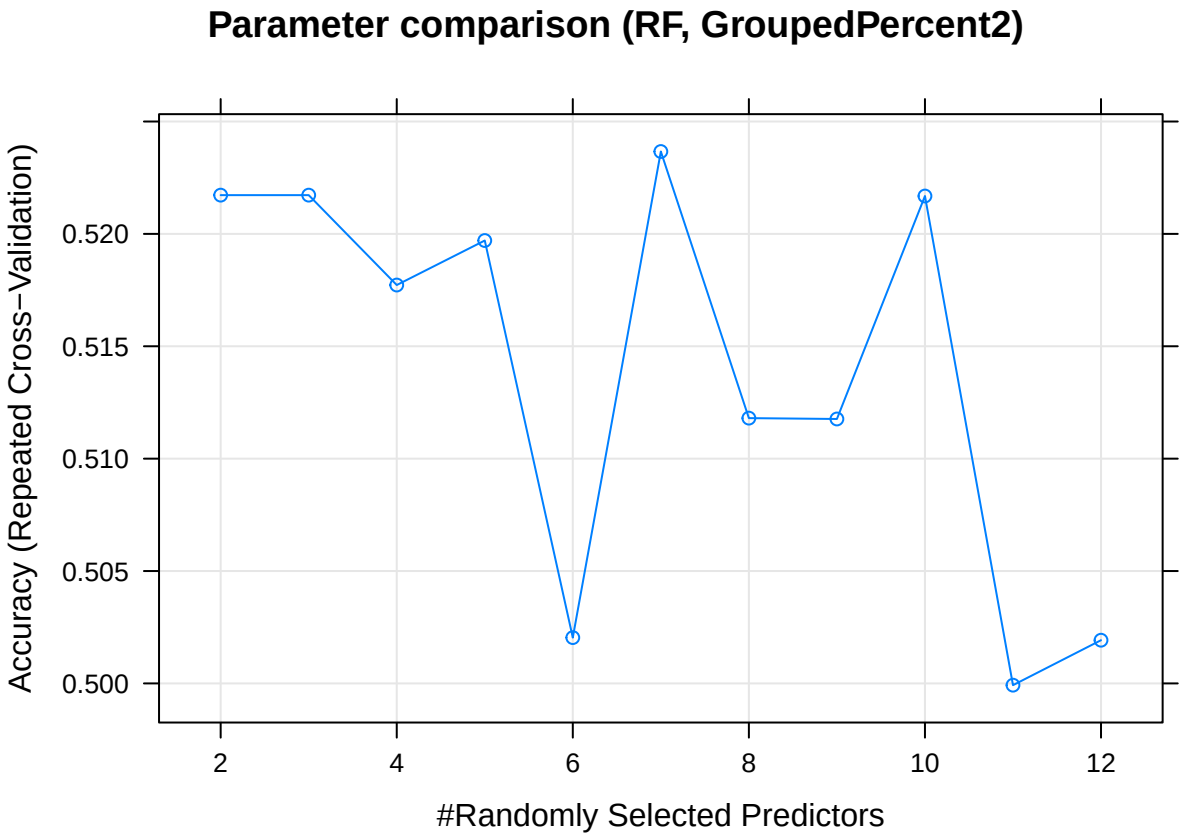


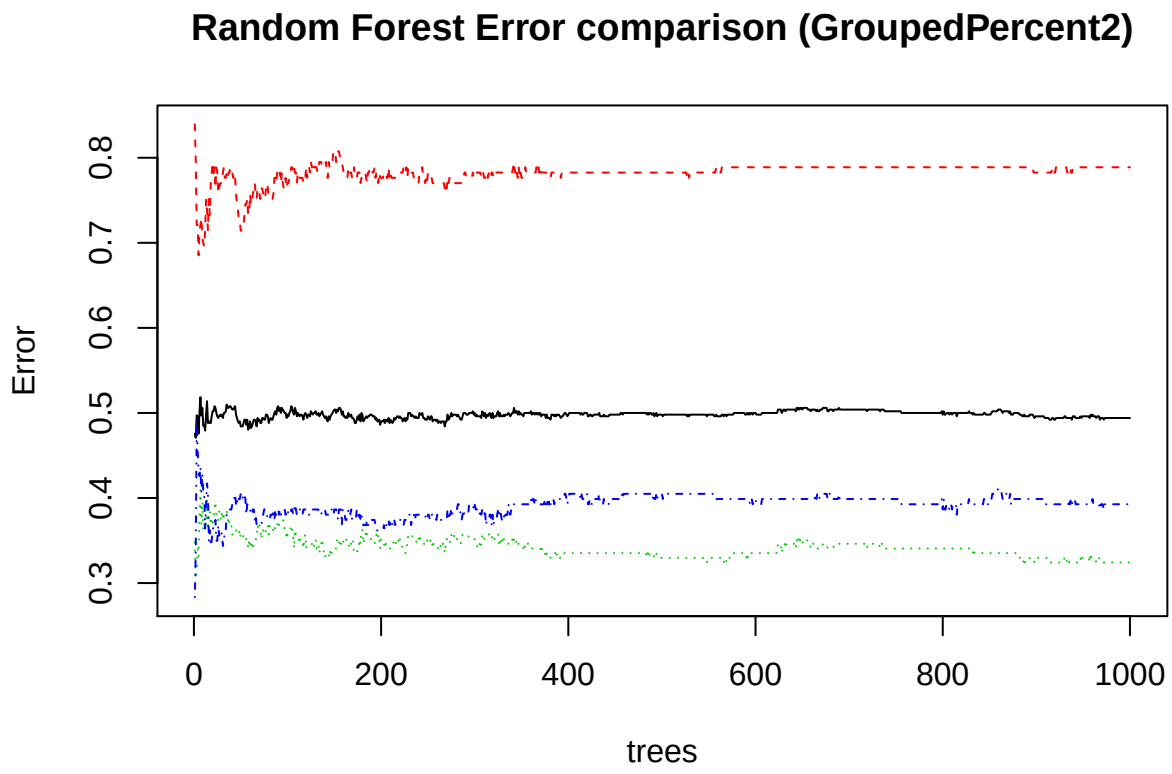

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          17     12           6
##   Remain          7     27           4
##   StrongLeave    16      7          28
##
## Overall Statistics
##
##                Accuracy : 0.5806
##                  95% CI : (0.4887, 0.6686)
##     No Information Rate : 0.371
##     P-Value [Acc > NIR] : 1.727e-06
##
##                   Kappa : 0.3734
##
##  Mcnemar's Test P-Value : 0.08285
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity                0.4250        0.5870              0.7368
## Specificity                0.7857        0.8590              0.7326
## Pos Pred Value             0.4857        0.7105              0.5490
## Neg Pred Value             0.7416        0.7791              0.8630
```

```
## Prevalence                0.3226          0.3710          0.3065
## Detection Rate            0.1371          0.2177          0.2258
## Detection Prevalence      0.2823          0.3065          0.4113
## Balanced Accuracy         0.6054          0.7230          0.7347
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.7135
```



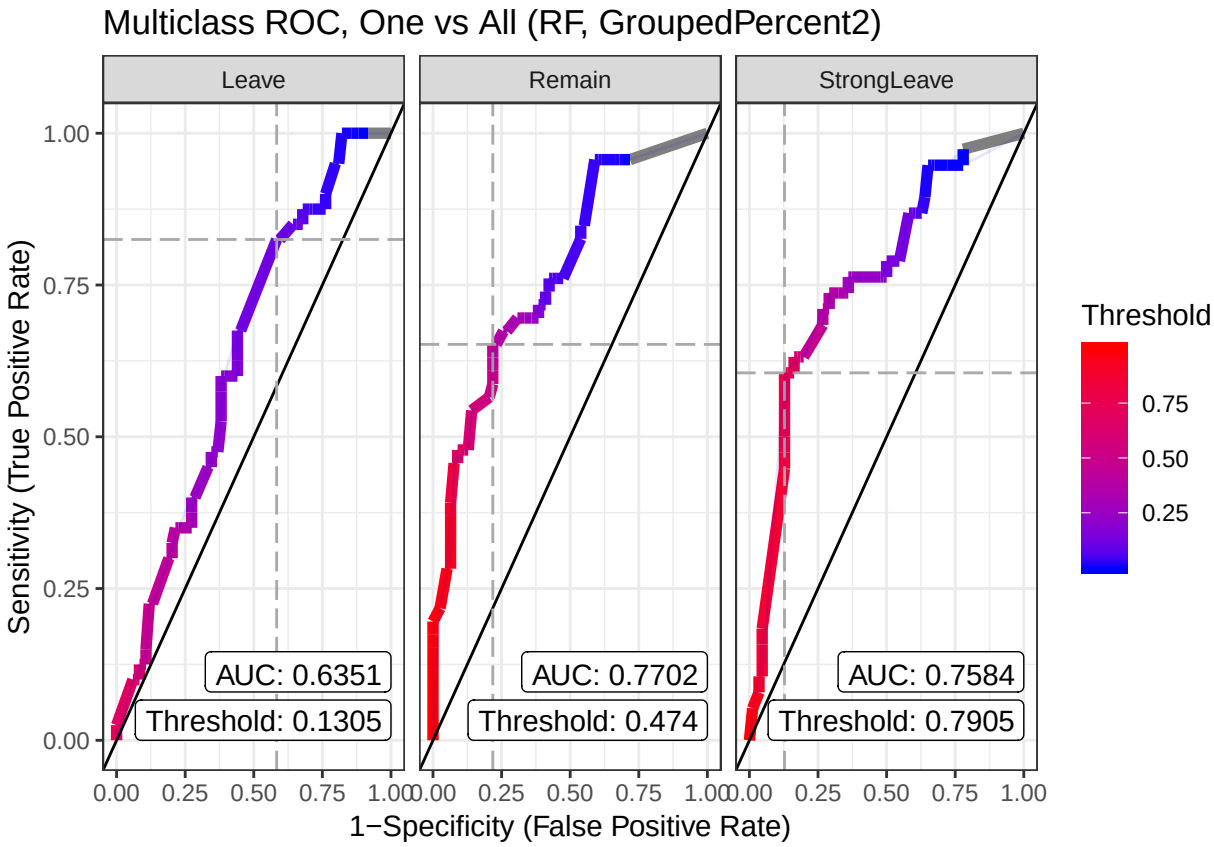

Variable importance (Tree, GroupedPercent2)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

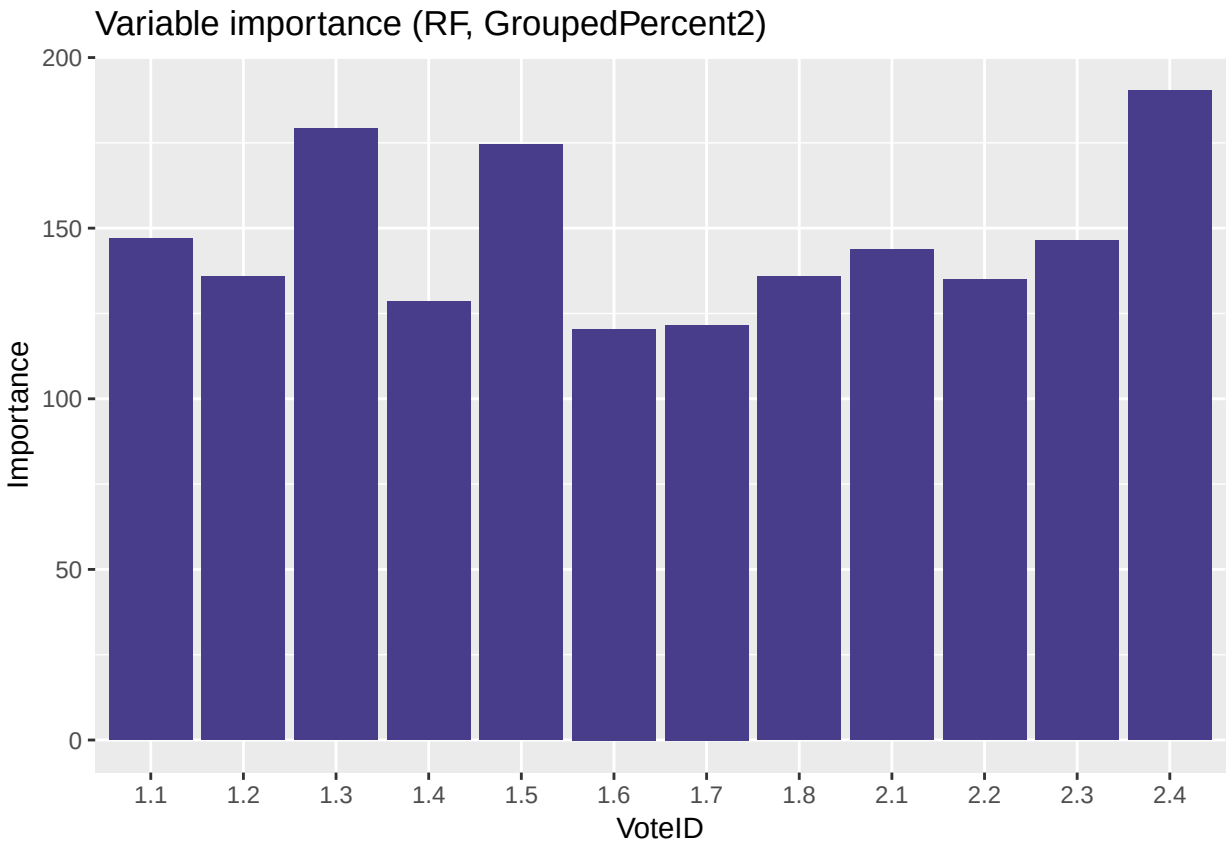

Parameter comparison (RF, GroupedPercent2)
Accuracy (Repeated Cross−Validation)
0.520
0.515
0.510
0.505
0.500
2
4
6
8
10
12
#Randomly Selected Predictors

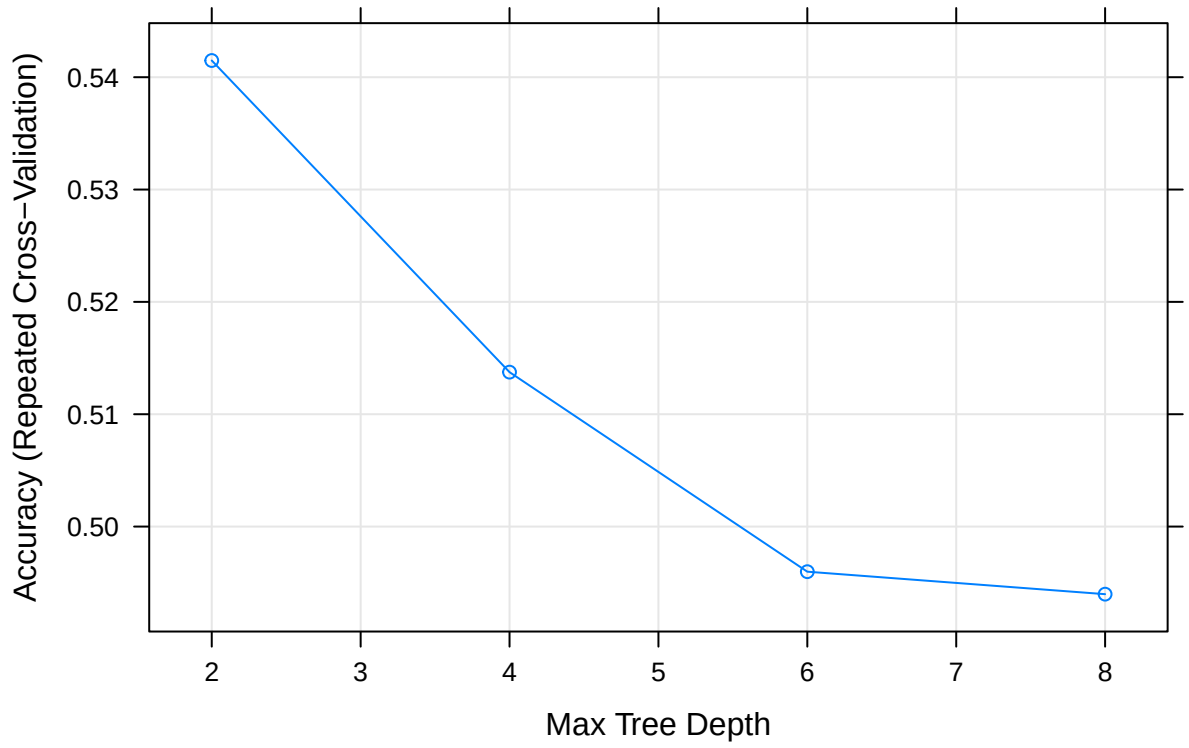



```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave           9      9           3
##   Remain         12     31           7
##   StrongLeave    19      6          28
##
## Overall Statistics
##
##                Accuracy : 0.5484
##                  95% CI : (0.4565, 0.6379)
##     No Information Rate : 0.371
##     P-Value [Acc > NIR] : 4.412e-05
##
##                   Kappa : 0.3207
##
##  Mcnemar's Test P-Value : 0.006913
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity               0.22500        0.6739             0.7368
## Specificity               0.85714        0.7564             0.7093
## Pos Pred Value            0.42857        0.6200             0.5283
## Neg Pred Value            0.69903        0.7973             0.8592
```

```
## Prevalence                0.32258         0.3710            0.3065
## Detection Rate            0.07258         0.2500            0.2258
## Detection Prevalence      0.16935         0.4032            0.4274
## Balanced Accuracy         0.54107         0.7152            0.7231
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.7202
```

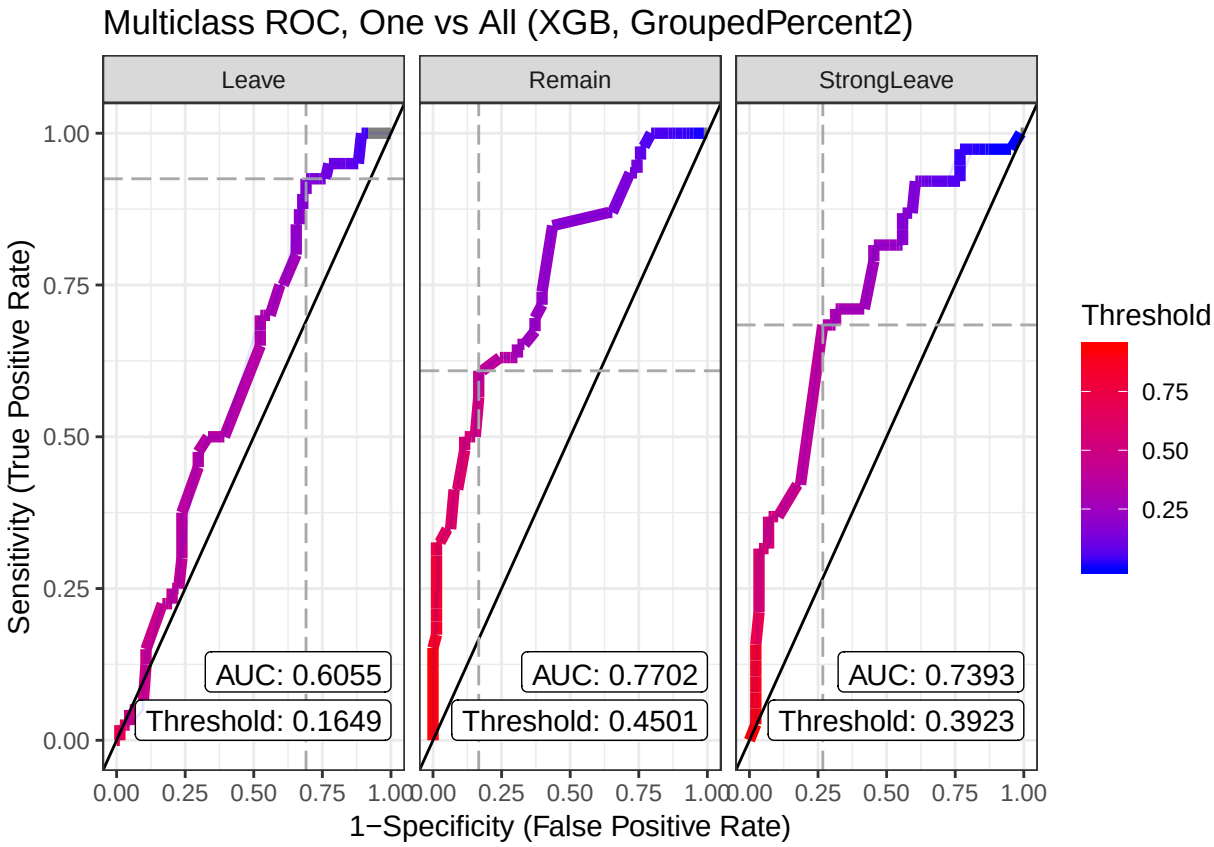


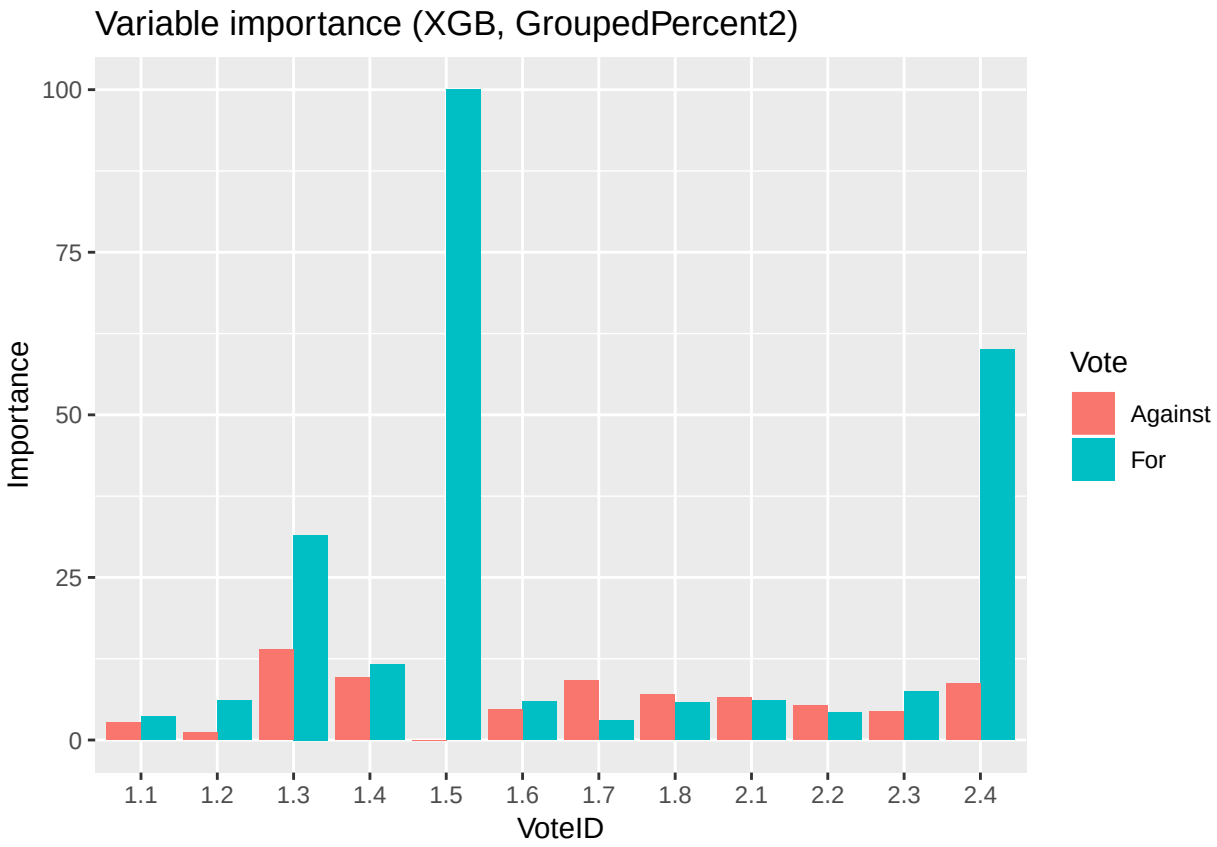

Variable importance (RF, GroupedPercent2)
Importance
200
150
100
50
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID

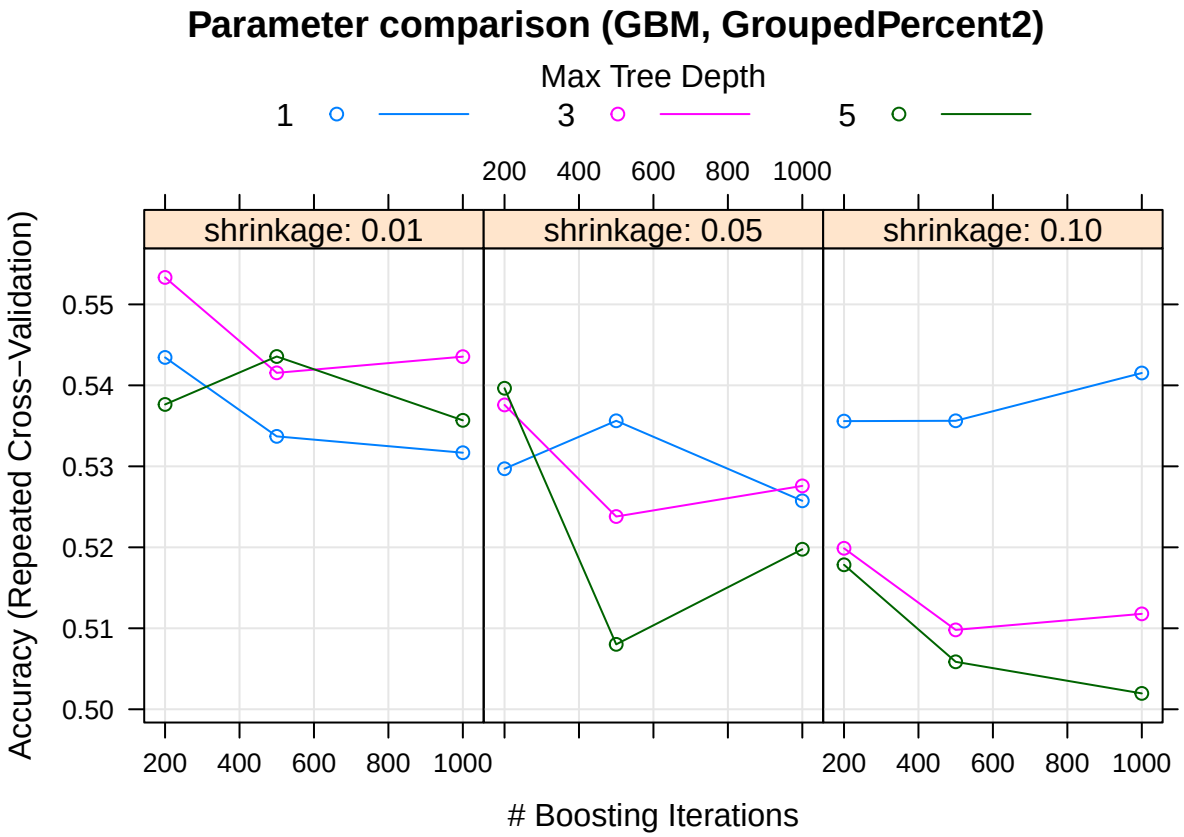



```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          12     11           4
##   Remain         11     29           8
##   StrongLeave    17      6          26
##
## Overall Statistics
##
##                Accuracy : 0.5403
##                  95% CI : (0.4485, 0.6301)
##     No Information Rate : 0.371
##     P-Value [Acc > NIR] : 9.177e-05
##
##                   Kappa : 0.3088
##
##  Mcnemar's Test P-Value : 0.0396
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity               0.30000        0.6304             0.6842
## Specificity               0.82143        0.7564             0.7326
## Pos Pred Value            0.44444        0.6042             0.5306
## Neg Pred Value            0.71134        0.7763             0.8400
```

```
## Prevalence                0.32258         0.3710            0.3065
## Detection Rate            0.09677         0.2339            0.2097
## Detection Prevalence      0.21774         0.3871            0.3952
## Balanced Accuracy         0.56071         0.6934            0.7084
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.7038
```

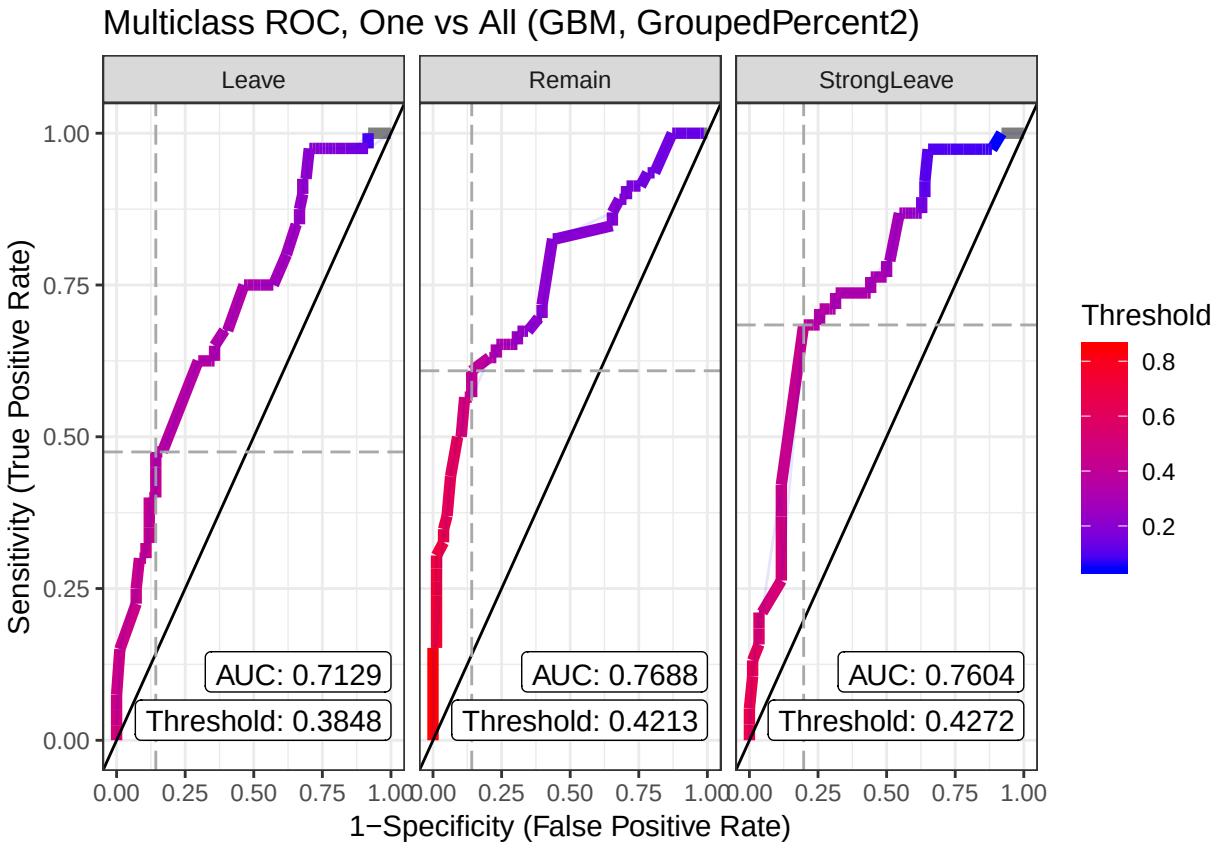


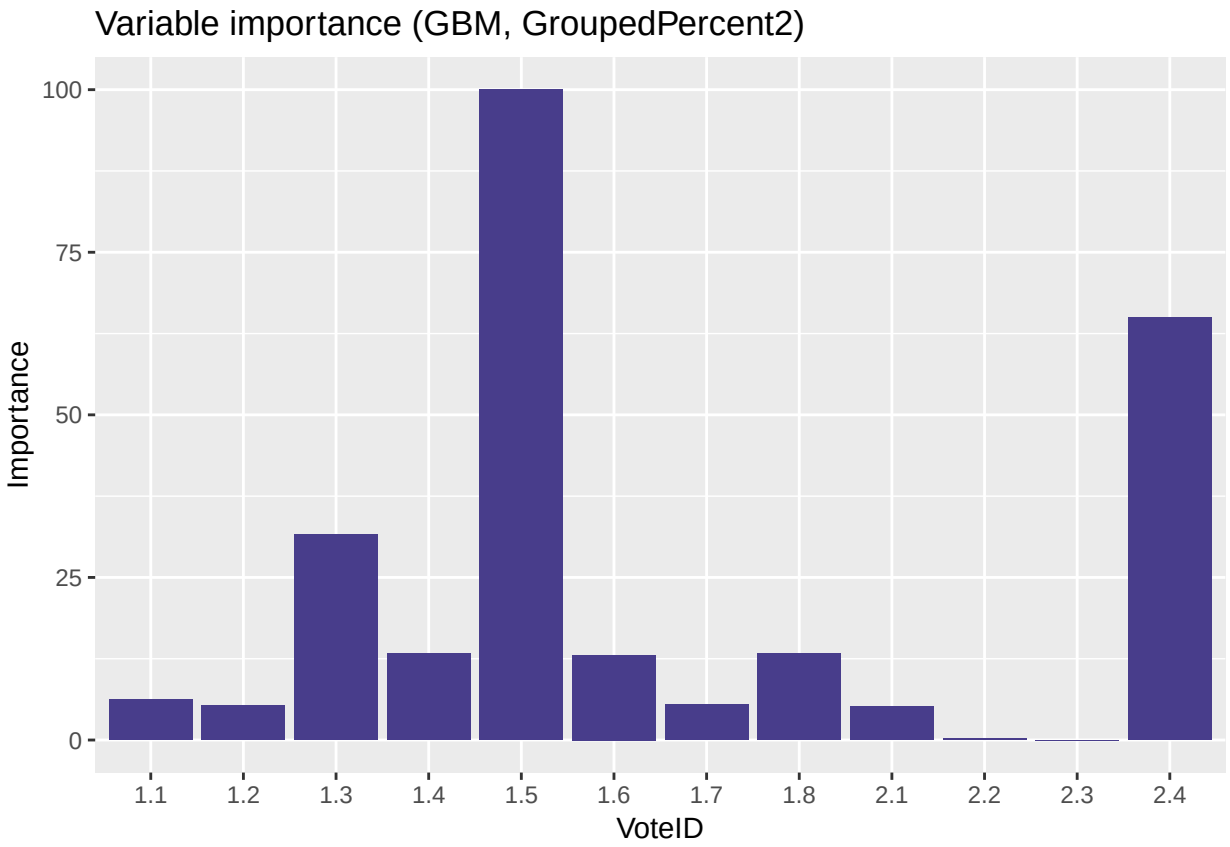

Variable importance (XGB, GroupedPercent2)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For

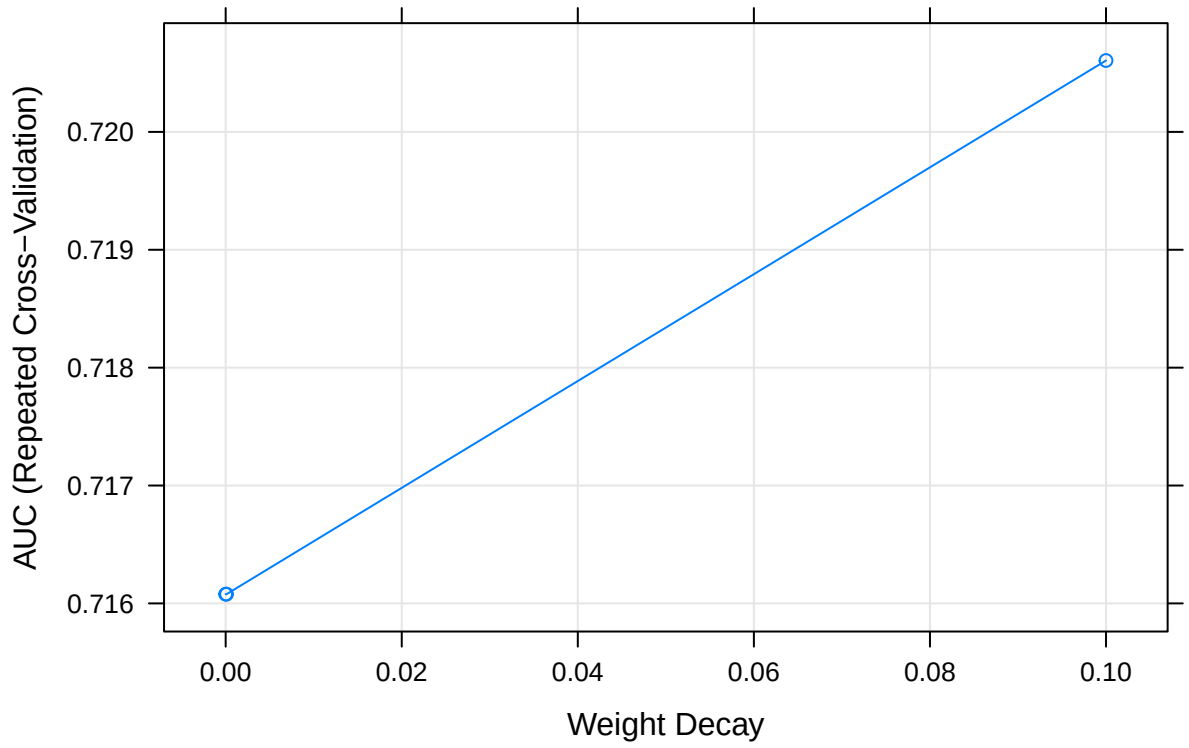



```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          20     12           3
##   Remain          8     28           8
##   StrongLeave    12      6          27
##
## Overall Statistics
##
##                Accuracy : 0.6048
##                  95% CI : (0.5131, 0.6914)
##     No Information Rate : 0.371
##     P-Value [Acc > NIR] : 1.089e-07
##
##                   Kappa : 0.4068
##
##  Mcnemar's Test P-Value : 0.09023
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity                0.5000        0.6087              0.7105
## Specificity                0.8214        0.7949              0.7907
## Pos Pred Value             0.5714        0.6364              0.6000
## Neg Pred Value             0.7753        0.7750              0.8608
```

```
## Prevalence                0.3226          0.3710             0.3065
## Detection Rate            0.1613          0.2258             0.2177
## Detection Prevalence      0.2823          0.3548             0.3629
## Balanced Accuracy         0.6607          0.7018             0.7506
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.746
```

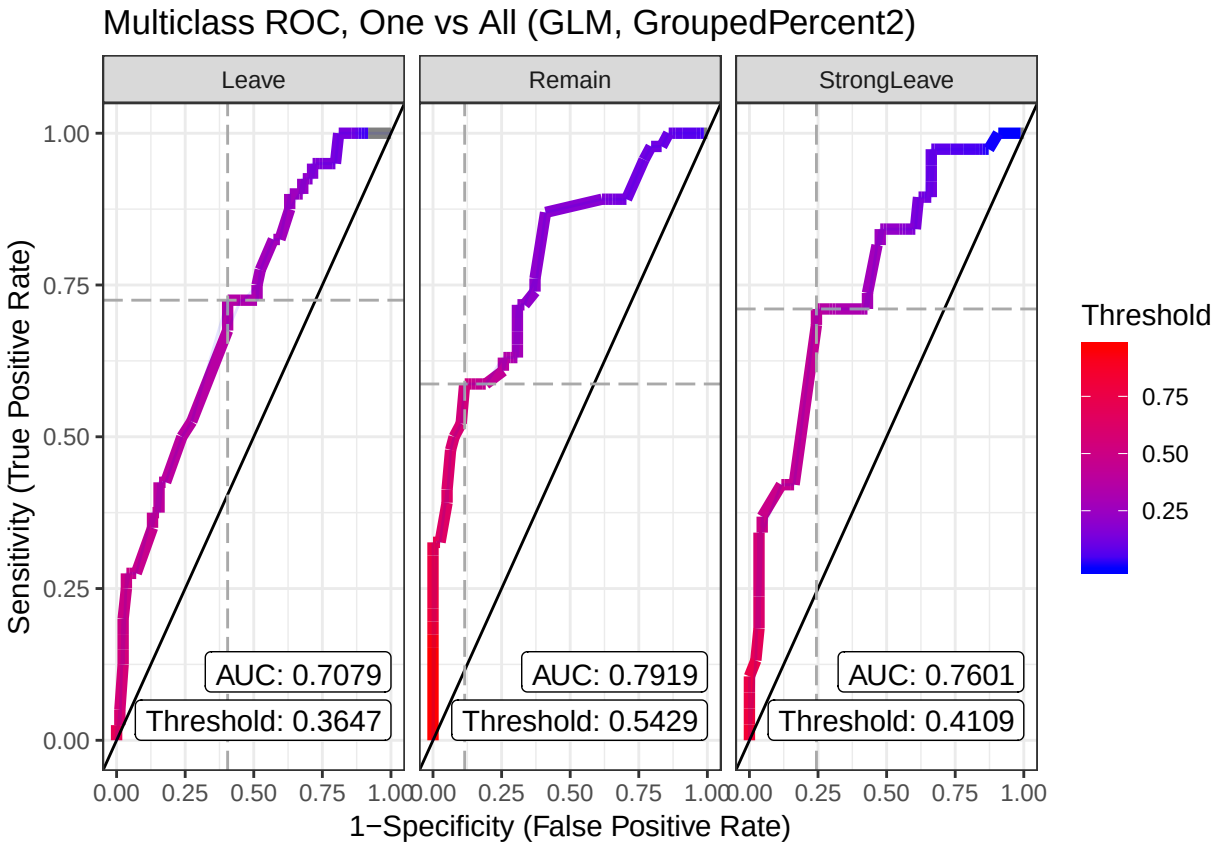



Variable importance (GBM, GroupedPercent2)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID



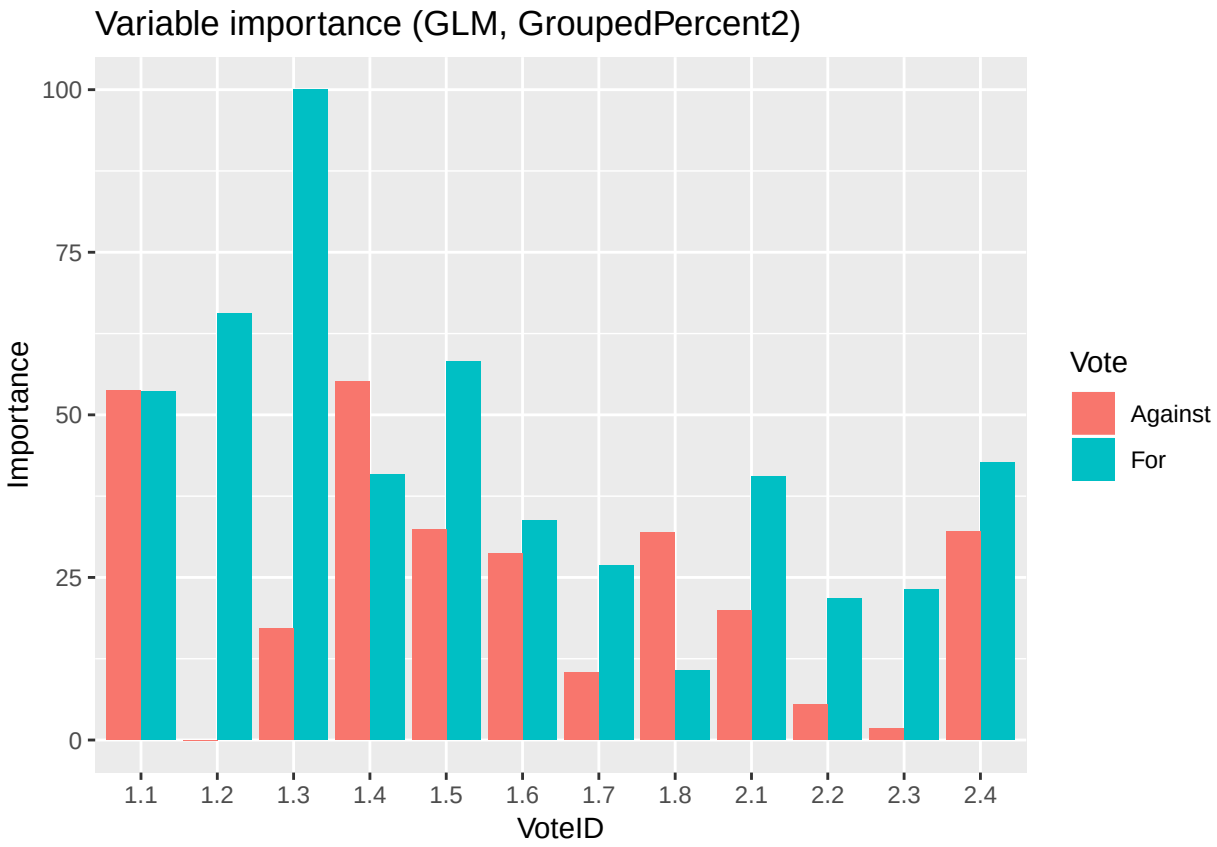

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          15     11           2
##   Remain         10     29          10
##   StrongLeave    15      6          26
##
## Overall Statistics
##
##                Accuracy : 0.5645
##                  95% CI : (0.4726, 0.6533)
##     No Information Rate : 0.371
##     P-Value [Acc > NIR] : 9.294e-06
##
##                   Kappa : 0.3446
##
##  Mcnemar's Test P-Value : 0.01179
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity                0.3750        0.6304              0.6842
## Specificity                0.8452        0.7436              0.7558
## Pos Pred Value             0.5357        0.5918              0.5532
## Neg Pred Value             0.7396        0.7733              0.8442
```

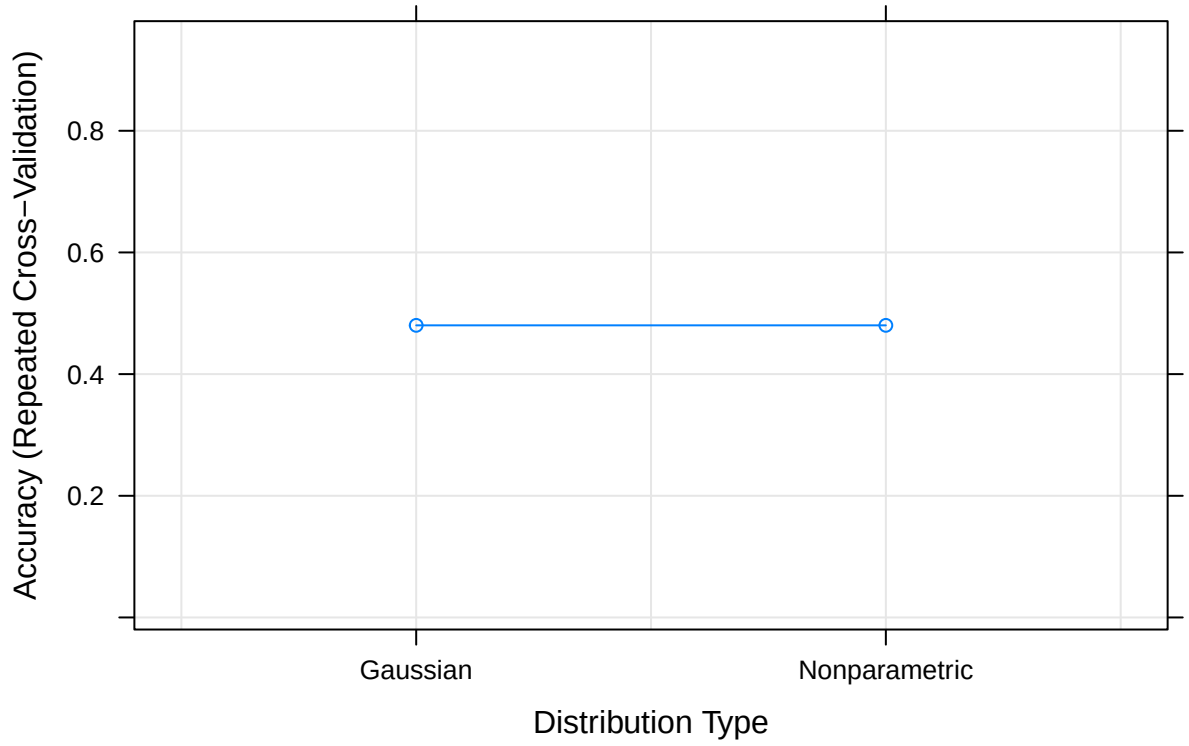
```
## Prevalence                0.3226          0.3710          0.3065
## Detection Rate            0.1210          0.2339          0.2097
## Detection Prevalence      0.2258          0.3952          0.3790
## Balanced Accuracy         0.6101          0.6870          0.7200
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.7522
```




```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

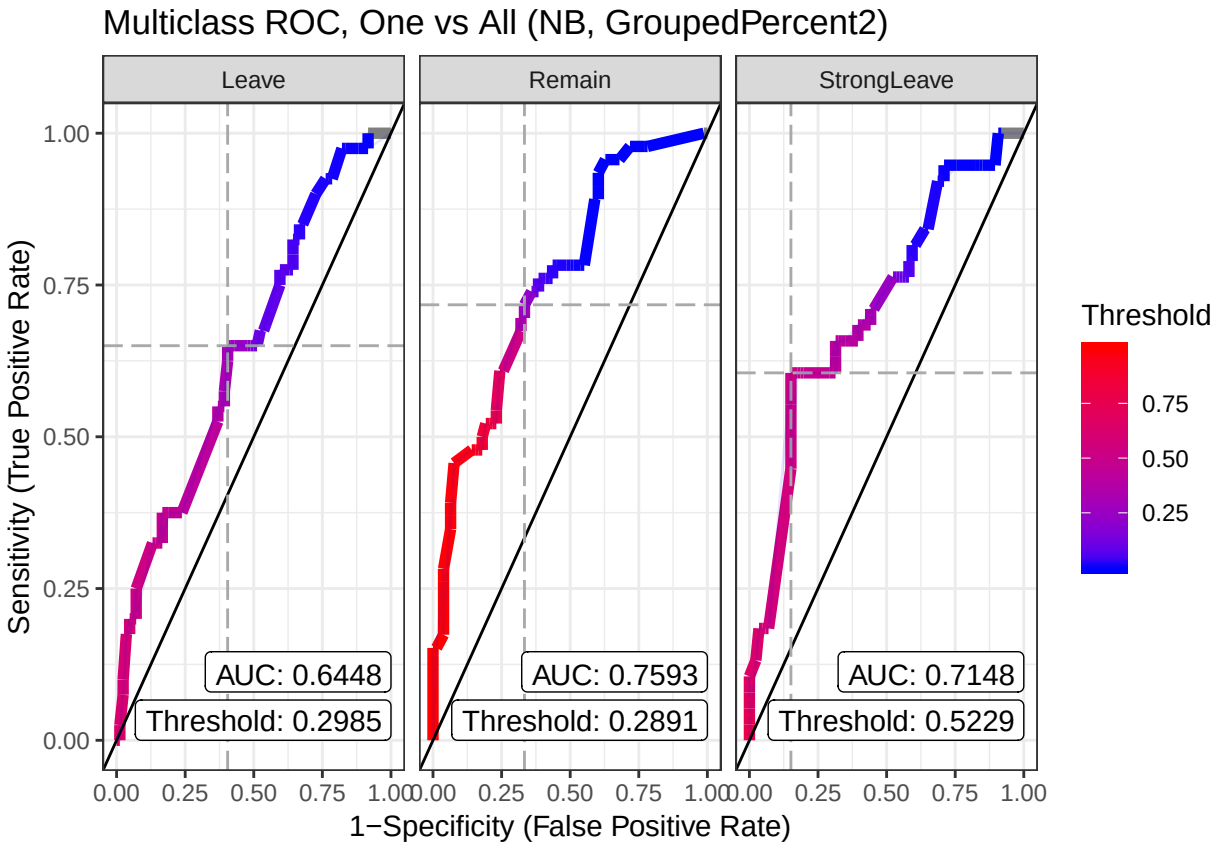
```
## observation 106
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126
```

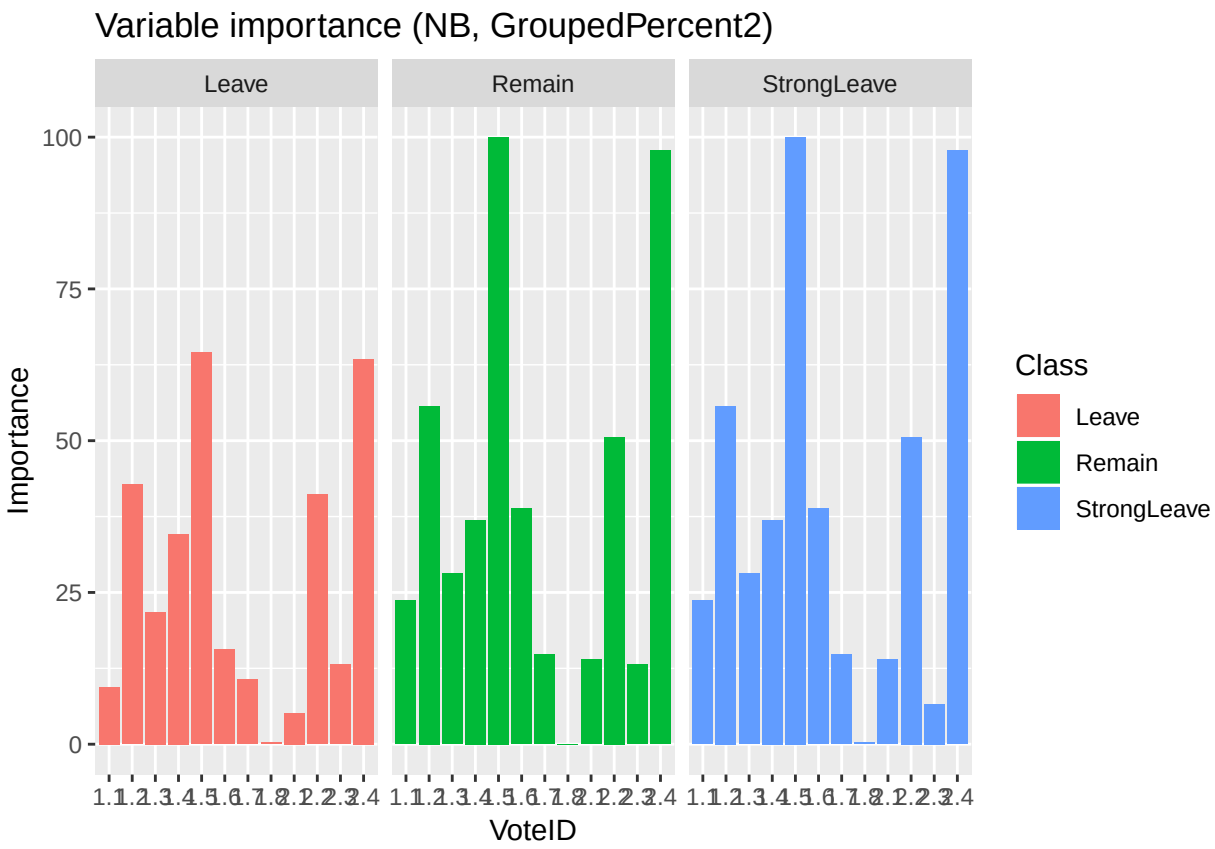

Variable importance (GLM, GroupedPercent2)
Importance
100
75
50
25
0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.1
2.2
2.3
2.4
VoteID
Vote
Against
For




```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Leave Remain StrongLeave
##   Leave          13     11           4
##   Remain         14     32          11
##   StrongLeave    13      3          23
##
## Overall Statistics
##
##                Accuracy : 0.5484
##                  95% CI : (0.4565, 0.6379)
##     No Information Rate : 0.371
##     P-Value [Acc > NIR] : 4.412e-05
##
##                   Kappa : 0.316
##
##  Mcnemar's Test P-Value : 0.02133
##
## Statistics by Class:
##
##                      Class: Leave Class: Remain Class: StrongLeave
## Sensitivity                0.3250        0.6957              0.6053
## Specificity                0.8214        0.6795              0.8140
## Pos Pred Value             0.4643        0.5614              0.5897
## Neg Pred Value             0.7188        0.7910              0.8235
```

```
## Prevalence                0.3226          0.3710          0.3065
## Detection Rate            0.1048          0.2581          0.1855
## Detection Prevalence      0.2258          0.4597          0.3145
## Balanced Accuracy         0.5732          0.6876          0.7096
##
## Call:
## multiclass.roc.default(response = response, predictor = preds.probs)
##
## Data: multivariate predictor preds.probs with 3 levels of response: Leave, Remain, StrongLeave.
## Multi-class area under the curve: 0.7022
```

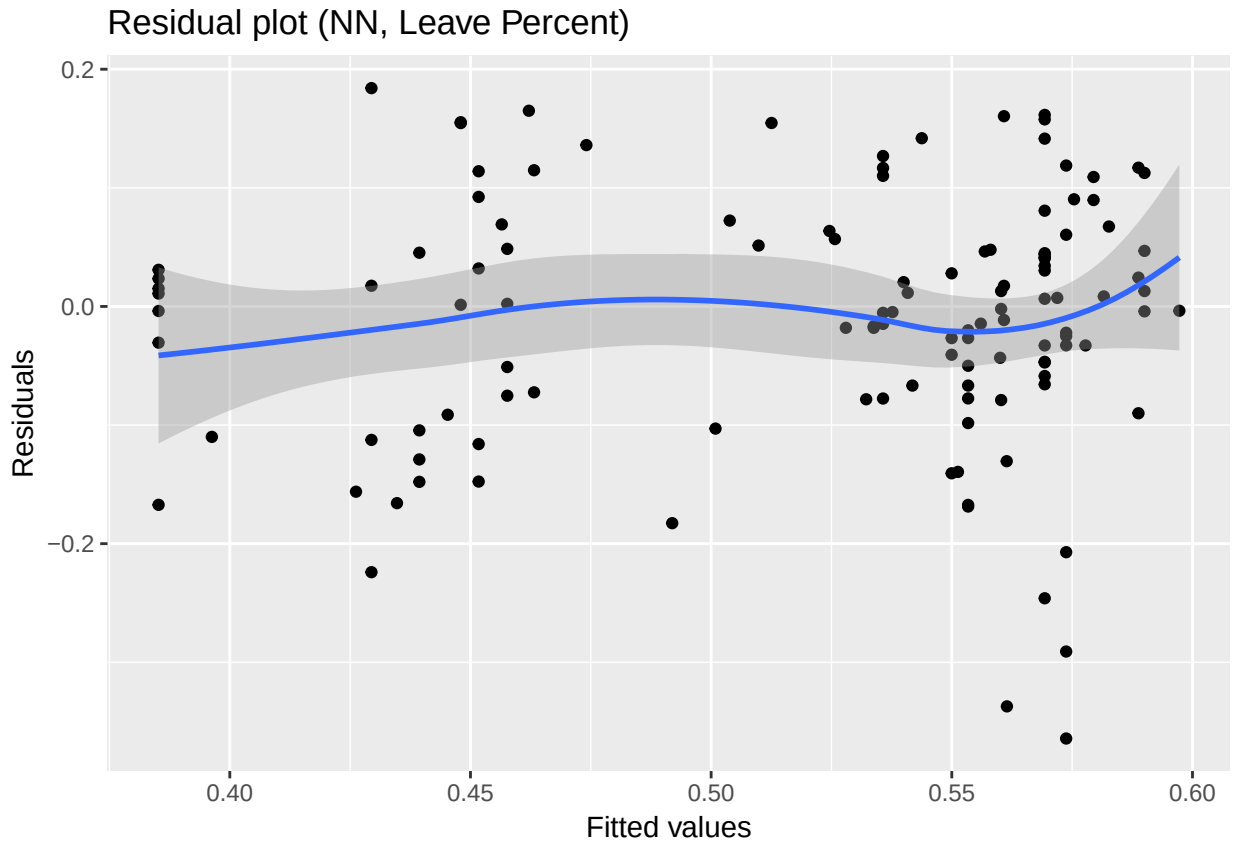


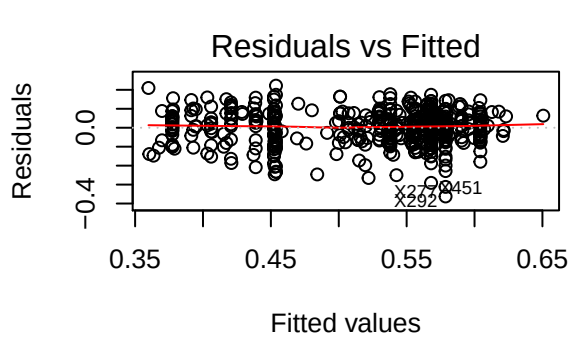


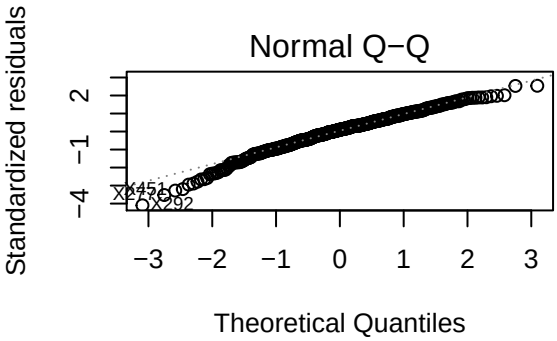

```
leave.percent.stats <- matrix(nrow=5, ncol=3)
colnames(leave.percent.stats) <- c("NN", "lm", "GBM")
rownames(leave.percent.stats) <- c("RMSE", "Rsquared","Best.Rsquared", "MAE", "Best.MAE")

for (i in 1:length(all.continuous.models)) {
  model.type <- names(all.continuous.models)[i]
  model <- all.continuous.models[[i]]
  if (model.type != "GBM") {
    test.data <- holdout.dummy.no.NA[,7:42]
    response <- holdout.dummy.no.NA[,"BetterLeavePercent"]
  } else {
    test.data <- holdout.no.NA[,7:18]
    response <- holdout.no.NA[,"BetterLeavePercent"]
  }
  leave.percent.stats["RMSE",model.type] <- min(na.omit(model$results[,"RMSE"]))
  leave.percent.stats["Rsquared", model.type] <-
    na.omit(model$results[,"Rsquared"])[which.min(na.omit(model$results[,"RMSE"]))]
  leave.percent.stats["MAE", model.type] <-
    na.omit(model$results[,"MAE"])[which.min(na.omit(model$results[,"RMSE"]))]

  leave.percent.stats["Best.Rsquared",model.type] <- min(na.omit(model$results[,"Rsquared"]))
  leave.percent.stats["Best.MAE",model.type] <- min(na.omit(model$results[,"MAE"]))

  if (model.type != "lm") plot(model)
  preds <- predict(model, test.data)
  if (model.type == "lm") {
```

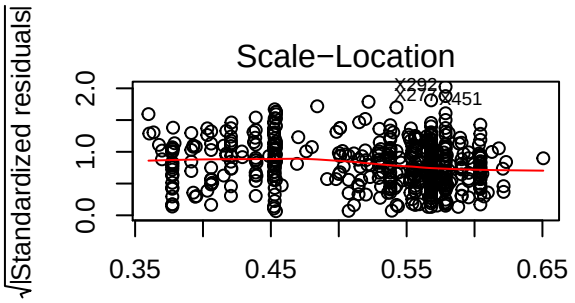
```
    par(mfrow=c(2,2))
    plot(model$finalModel)
    par(mfrow=c(1,1))
  } else {
    print(ggplot(mapping=aes(preds, response-preds)) + geom_point() + geom_smooth(method="loess") +
      labs(title=paste("Residual plot (", model.type, ", Leave Percent)", sep=""), x="Fitted values",
           y="Residuals"))
  }
}
```

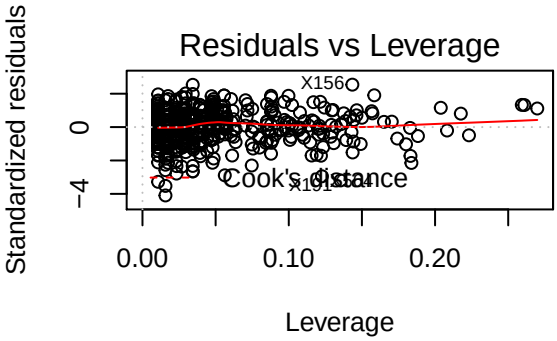
```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading
```

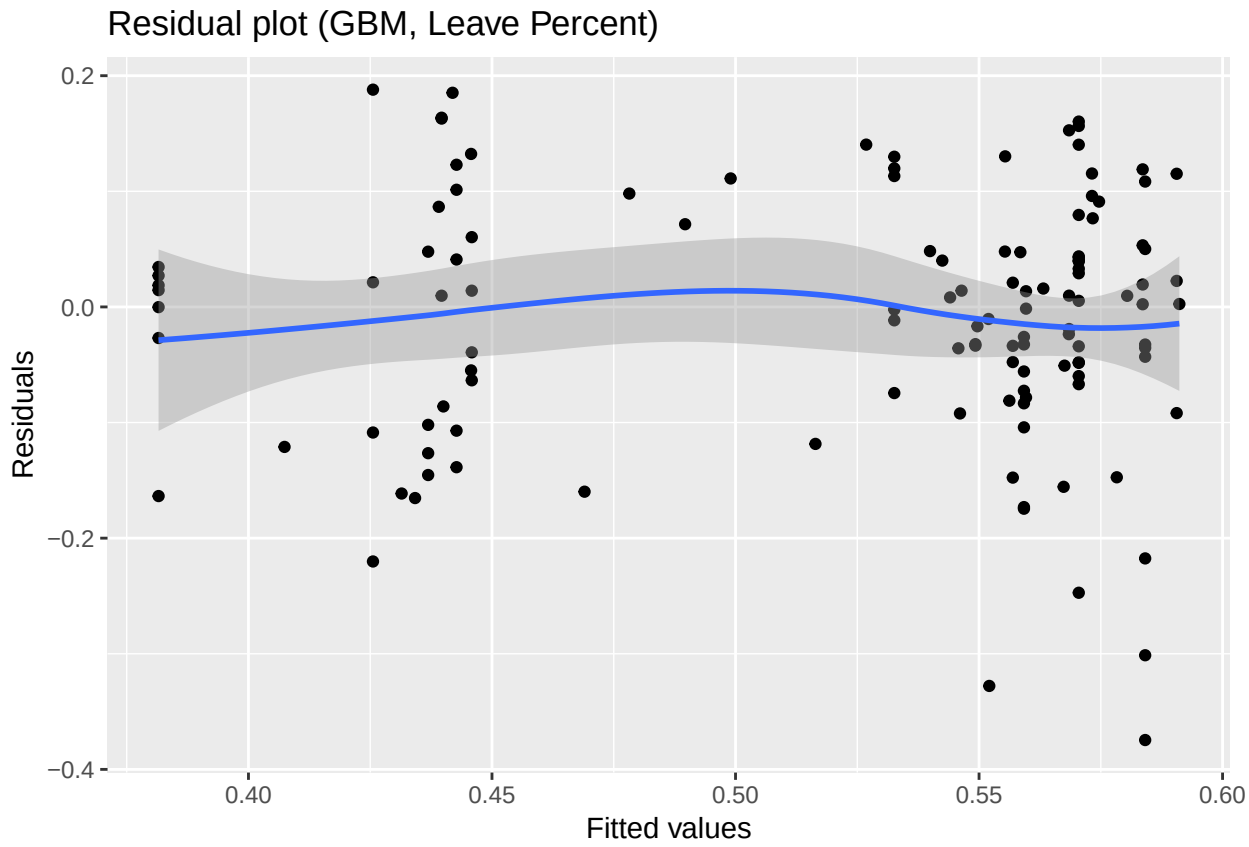



Residuals vs Fitted
Residuals
Fitted values
X292
X451


Normal Q−Q
Standardized residuals
Theoretical Quantiles
X451
X292


Scale−Location
√|Standardized residuals|
Fitted values
X292
X451


Residuals vs Leverage
Standardized residuals
Leverage
X156
Cook's distance




```
## Renaming responses as row name to be more descriptive
# Party -> Party
# PartyGroup -> PartyGroup
# VoteResult -> ConstituencyVote
# LeaveQuantile -> GroupedPercent
# LeaveDiscrete -> GroupedPercent2

knitr::kable(accuracy)
```

	NN	SVM	Tree	RF	XGB	GBM	GLM	NB	a
Party	0.8567191	0.8586609	0.8752888	0.9519437	0.9425074	0.9443938	0.9463169	0.8675214	
PartyGroup	0.9463187	0.9462821	0.9251465	0.9539560	0.9482234	0.9520879	0.9482051	0.8713919	
VoteResult	0.7766562	0.7767144	0.7648523	0.7825579	0.7548530	0.7904787	0.7567940	0.6893458	0.7884
LeaveQuantile	0.4485490	0.4110588	0.4186275	0.4523922	0.4763922	0.4427059	0.4289804	0.3435686	
LeaveDiscrete	0.5475053	0.5354883	0.5336245	0.5236653	0.5414871	0.5533295	0.5336440	0.4801786	

```
knitr::kable(kappa)
```

	NN	SVM	Tree	RF	XGB	GBM	GLM	NB	a
Party	0.7922887	0.7946047	0.7954155	0.9212703	0.9058560	0.9085556	0.9117721	0.7970506	
PartyGroup	0.9105885	0.9098658	0.8748942	0.9231684	0.9136020	0.9197100	0.9134991	0.7933818	
VoteResult	0.4955817	0.4884222	0.4681741	0.5008780	0.4547317	0.5209667	0.4518261	0.3733827	0.5170
LeaveQuantile	0.2647661	0.2169416	0.2259319	0.2693548	0.3010362	0.2570571	0.2378083	0.1253315	
LeaveDiscrete	0.3191040	0.3009852	0.2985146	0.2839235	0.3099596	0.3285201	0.2977946	0.2114471	

```
knitr::kable(auc)
```

	NN	SVM	Tree	RF	XGB	GBM	GLM	NB	ada
Party	0.8470475	NA	0.7174152	0.8597636	0.8228588	0.8024256	0.7981934	0.7898154	NA
PartyGroup	0.9529878	NA	0.9263324	0.9696430	0.9554349	0.9513301	0.9490268	0.9264097	NA
VoteResult	0.7793619	NA	0.7220497	0.7603049	0.7539526	0.7553642	0.7542349	0.7584698	0.7689159
LeaveQuantile	0.6909447	NA	0.6936242	0.7040646	0.6966273	0.7086948	0.6991306	0.6534315	NA
LeaveDiscrete	0.7469318	NA	0.7135107	0.7202350	0.7038330	0.7460050	0.7521834	0.7021548	NA

```
best.kappa # This is the best kappas if we had used kappa to choose rather than accuracy
```

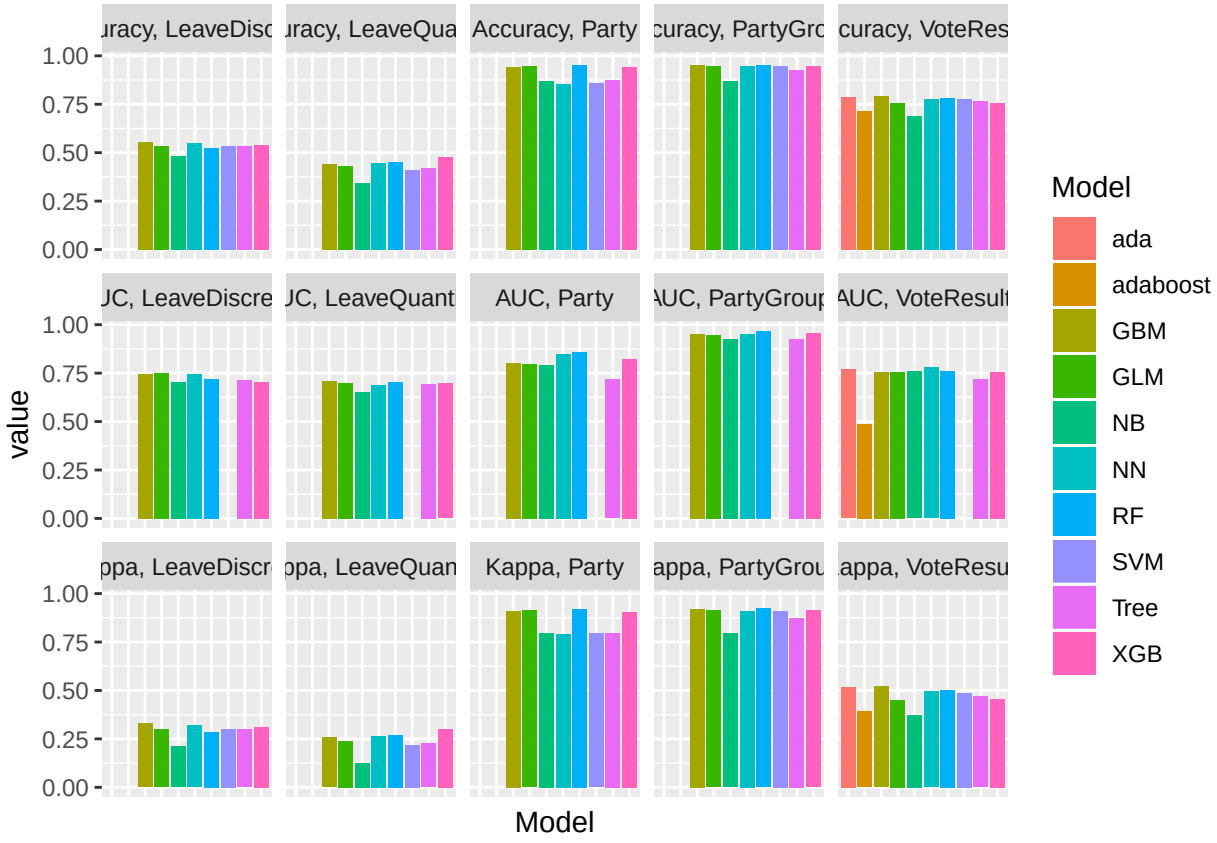
```
##                       NN       SVM      Tree        RF       XGB       GBM
## Party          0.7922887 0.7946047 0.7954155 0.9212703 0.9059558 0.9085556
## PartyGroup     0.9105885 0.9098658 0.8748942 0.9231684 0.9136020 0.9197100
## VoteResult     0.4955817 0.4884222 0.4681741 0.5008780 0.4547317 0.5209667
## LeaveQuantile 0.2647661 0.2169416 0.2259319 0.2693548 0.3010362 0.2570571
## LeaveDiscrete 0.3191040 0.3009852 0.2985146 0.2839235 0.3099596 0.3285201
##                      GLM        NB       ada  adaboost
## Party          0.9117721 0.7970506        NA        NA
## PartyGroup     0.9134991 0.7933818        NA        NA
## VoteResult     0.4518261 0.3733827 0.5170936 0.3904882
## LeaveQuantile 0.2378083 0.1253315        NA        NA
## LeaveDiscrete 0.2977946 0.2114471        NA        NA
```

```
            # (Btw, it's exactly the same, except for a TINY change in XGB for Party
knitr::kable(leave.percent.stats)
```

	NN	lm	GBM
RMSE	0.0908858	0.0921730	0.0907404
Rsquared	0.3421588	0.3278733	0.3460730
Best.Rsquared	0.3250428	0.3278733	0.3014126
MAE	0.0708292	0.0717047	0.0706026
Best.MAE	0.0708292	0.0717047	0.0706026

```
plot.metrics <- function(stat.matrices, stats) {
  all.stats <- data.frame("response"=NA, "stat"=NA, "Model"=NA, "value"=NA)

  for (i in 1:length(stat.matrices)) {
    matrix <- stat.matrices[[i]]
    stat <- stats[i]
    stat.df <- data.frame(matrix)
    responses <- rownames(matrix)
    stat.df$response <- rownames(matrix)
    stat.df$stat <- stat
    stat.df <- stat.df %>% gather("Model", "value", -response, -stat)
    all.stats <- rbind(all.stats, stat.df)
  }

  all.stats$type <- apply(all.stats, 1, function(x) return(paste(x[2], ", ", x[1], sep="")))
  all.stats <- all.stats[2:nrow(all.stats),]
  ggplot(na.omit(all.stats), aes(x=Model, y=value, fill=Model)) +
```

```
    geom_col() + facet_wrap(~type, ncol=5) + theme(axis.ticks.x=element_blank(),
                                                   axis.text.x=element_blank())
}

plot.metrics(list(accuracy, kappa, auc), c("Accuracy","Kappa", "AUC"))
```

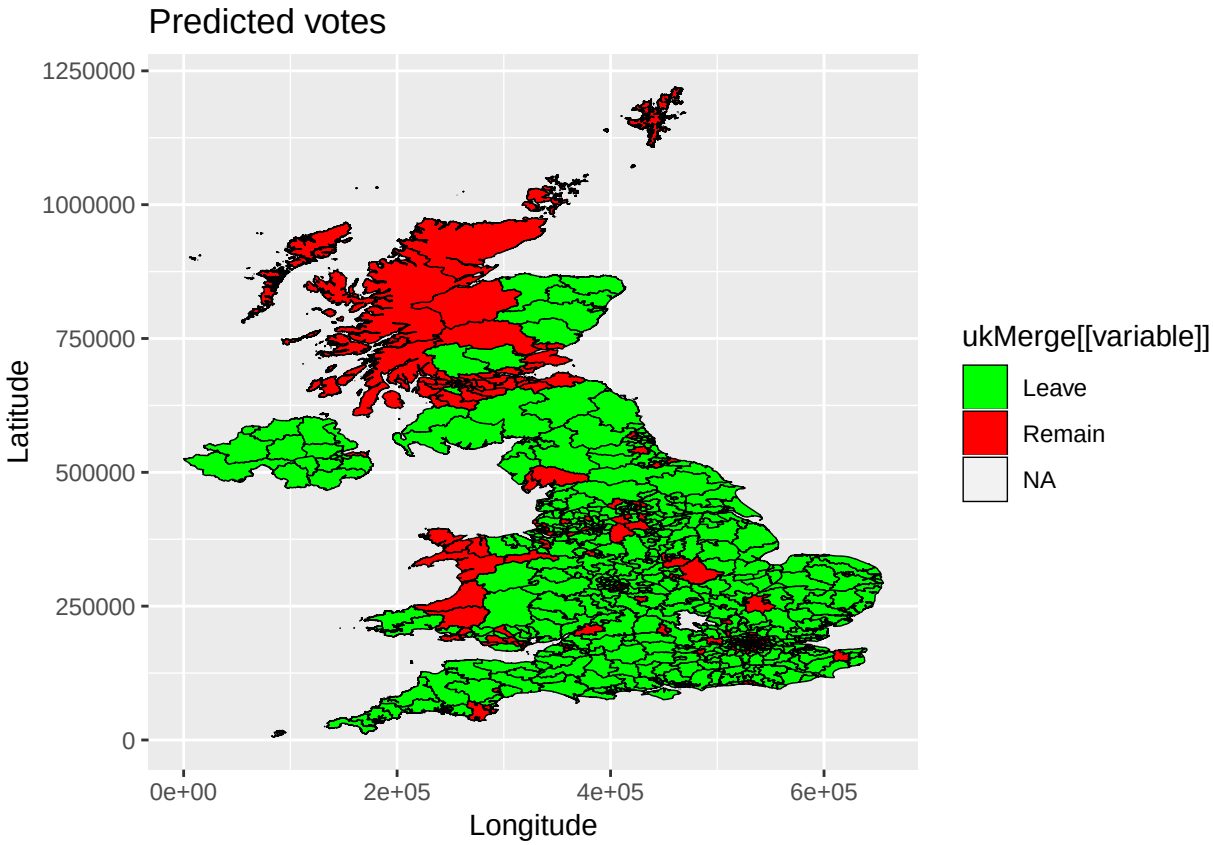



```
# Prediction Choropleths  ----------------------------------------------------------------
#### Importing data ####
mpdata <- rbind(train, holdout)
mpdata$party.predict <- predict(all.class.models$Party$RF,mpdata[,7:18])
mpdata$vote.predict <- predict(all.class.models$ConstituencyVote$GBM, mpdata[,7:18])

#### Choropleths ####
ukMap <- readOGR('Shapefiles/Westminster_Parliamentary_Constituencies_December_2017_Generalised_Clipped_

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\jackb\OneDrive - University of St Andrews\Everything\Uni\Data Intensive Analysis\I
## with 650 features
## It has 9 fields
## Integer64 fields read as strings:  bng_e bng_n objectid
ukMapf <- fortify(ukMap, region = "pcon17nm") %>% rename(Geography = id) %>% arrange(Geography)
mpdata$Geography <- mpdata$Constituency

# Fixing geographies
ukMapf$Geography <- gsub('[[:punct:] ]+',' ',ukMapf$Geography)
mpdata$Geography <- replace(mpdata$Geography, mpdata$Name=="Albert Owen", "Ynys Mon")
```

```
geos.map <- sort(unique(ukMapf$Geography))
geos.map <- geos.map[!geos.map == "Newport West"]
geos.map <- geos.map[!geos.map == "Buckingham"]
mpdata <- mpdata[order(mpdata$Geography),]
mpdata$Geography <- geos.map

vote.predict.map <- UKChoroplethFunction(ukMapf, mpdata, 'vote.predict', "Predicted votes", "Predicted
vote.predict.map <- vote.predict.map + scale_fill_manual(values=c("Green", "Red"))
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

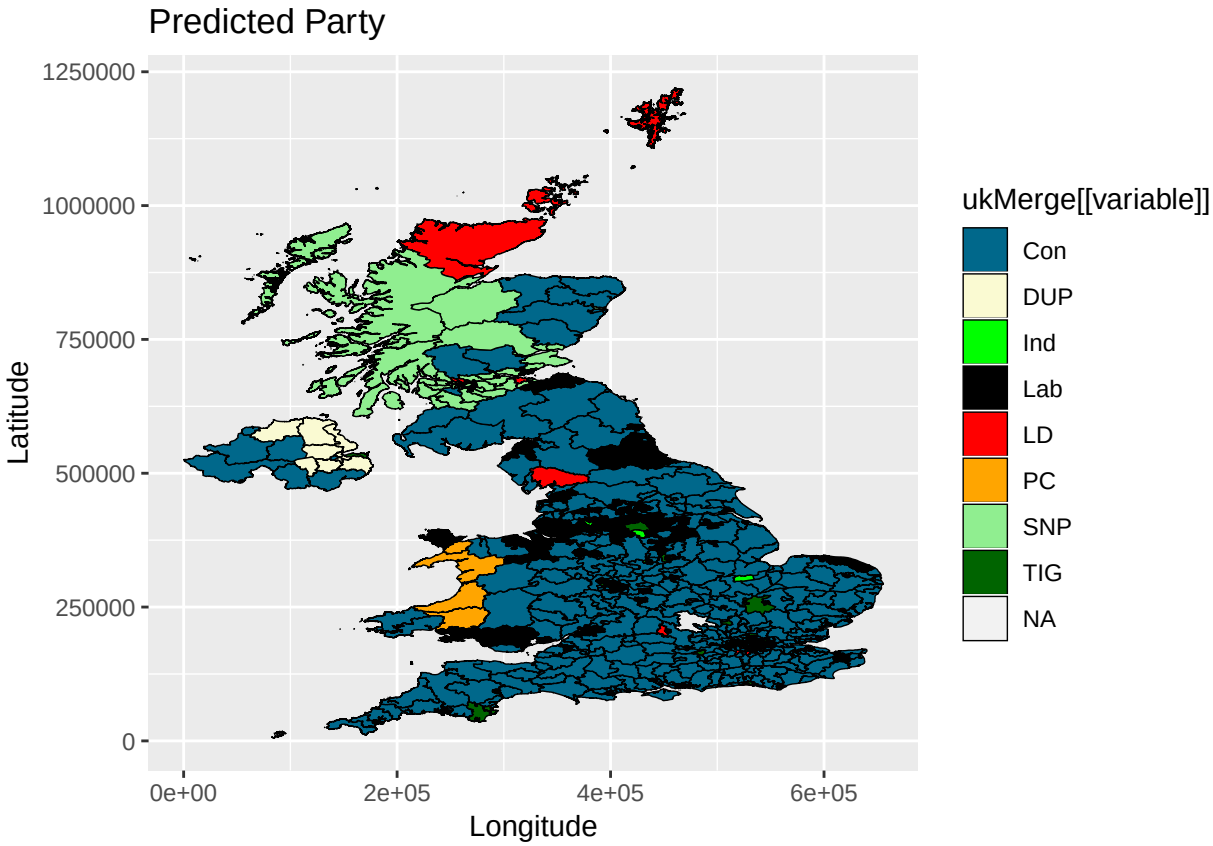
```
vote.predict.map
```




```
party.predict.map <- UKChoroplethFunction(ukMapf, mpdata, 'party.predict', "Predicted Party", "Predicte
party.predict.map <- party.predict.map + scale_fill_manual(values=party.color.scheme)
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

```
party.predict.map
```




```
#### OUR OWN VOTES ####
our.data <- data.frame(read_csv("data/ourvotes.csv"))
```

```
## Parsed with column specification:
## cols(
##   Name = col_character(),
##   Vote1 = col_character(),
##   Vote2 = col_character(),
##   Vote3 = col_character(),
##   Vote4 = col_character(),
##   Vote5 = col_character(),
##   Vote6 = col_character(),
##   Vote7 = col_character(),
##   Vote8 = col_character(),
##   SecondVote1 = col_character(),
##   SecondVote2 = col_character(),
##   SecondVote3 = col_character(),
##   SecondVote4 = col_character()
## )
```

```
for (i in 2:13) {
  our.data[,i] <- factor(our.data[,i], levels=c("For", "Abstain", "Against"))
}
#our.data <- as.data.frame(unclass(our.data))
our.data$vote <- predict(all.class.models$ConstituencyVote$GBM, our.data[,2:13])
our.data$party <- predict(all.class.models$Party$RF, our.data[,2:13])
knitr::kable(our.data)
```

Name	Vote1	Vote2	Vote3	Vote4	Vote5	Vote6	Vote7	Vote8	SecondVote1	SecondVote2
Amir	For	For	Abstain	Abstain	For	Against	Abstain	Against	For	For
Ossian	For	For	Abstain	Against	For	Against	Abstain	Against	For	For
Jack	Abstain	For	Abstain	Against	For	Against	For	Against	For	For
Sam	Against	Against	Against	For	Against	Abstain	Against	Abstain	Against	Against
Enoch	Against	For	Against	Against	For	Against	Against	Against	Against	For

```
#### Party ####
party.models <- list()
# names(party.models) <- c("NN", "SVM", "Tree", "RF", "XGB", "GBM", "GLM", "NB")
### Neural Net ###
party.models$NN <- nn.cv("Party", covars, train.dummy, nn.grid)
### SVM ###
party.models$SVM <- svm.cv("Party", covars, train.dummy, svm.grid)
### Binary Tree ###
party.models$Tree <- binary.tree.cv(train,"Party"], train[,7:18])
### Random Forest ###
party.models$RF <- rf.cv(train[,7:18], train[,1], na.omit)
### XGBoost Tree ###
party.models$XGB <- xg.cv2(1, train, na.omit)
### Generalized Boosted Regression Modelling ###
party.models$GBM <- gbm.caret.func(DM = train[, 7:18], y = train[, 1])
### GLM ###
party.models$GLM <- multinom.cv(train,"Party"], train[,7:18])
### Naive Bayes ###
party.models$NB <- nb.cv(train,"Party"], train[,7:18])

#### Party Group ####
party.grp.models <- list()
# names(party.grp.models) <- c("NN", "SVM", "Tree", "RF", "XGB", "GBM", "GLM", "NB")
### Neural Net ###
party.grp.models$NN <- nn.cv("PartyGroup", covars, train.dummy, nn.grid)
### SVM ###
party.grp.models$SVM <- svm.cv("PartyGroup", covars, train.dummy, svm.grid)
### Binary Tree ###
party.grp.models$Tree <- binary.tree.cv(train,"PartyGroup"], train[,7:18])
### Random Forest ###
party.grp.models$RF <- rf.cv(train[,7:18], train[,2], na.omit)
### XGBoost Tree ###
party.grp.models$XGB <- xg.cv2(2, train, na.omit)
### Generalized Boosted Regression Modelling ###
party.grp.models$GBM <- gbm.caret.func(DM = train[, 7:18], y = train[, 2])
### GLM ###
party.grp.models$GLM <- multinom.cv(train,"PartyGroup"], train[,7:18])
### Naive Bayes
party.grp.models$NB <- nb.cv(train,"PartyGroup"], train[,7:18])

#### Constituency Vote Result ####
ref.vote.models <- list()
# names(ref.vote.models) <- c("NN", "SVM", "RF", "XGB", "GBM", "ada", "adaboost", "GLM", "NB")
### Neural Net
ref.vote.models$NN <- nn.cv("ConstituencyVote", covars, train.dummy.no.NA, nn.grid, linout=FALSE)
### SVM ###
```

```
ref.vote.models$SVM <- svm.cv("ConstituencyVote", covars, train.dummy.no.NA, svm.grid)
### Binary Tree ###
ref.vote.models$Tree <- binary.tree.cv(train[,"ConstituencyVote"], train[,7:18])
### Random Forest ###
ref.vote.models$RF <- rf.cv(train.no.NA[,7:18], train.no.NA[,6], na.omit)
### XGBoost Tree ###
ref.vote.models$XGB <- xg.cv2(6, train.no.NA, na.omit)
### Generalized Boosted Regression Modelling ###
ref.vote.models$GBM <- gbm.caret.func(DM = train.no.NA[, 7:18], y = train.no.NA[, 6])
### Adaboost ###
ref.vote.models$ada <- ada.caret.func(DM = train.no.NA[ , 7:18], y = train.no.NA[, 6], Ada.method = 'ada
ref.vote.models$adaboost <- ada.caret.func(DM = train.no.NA[ , 7:18], y = train.no.NA[, 6], Ada.method =
### GLM ###
ref.vote.models$GLM <- multinom.cv(train.no.NA[,'ConstituencyVote'], train.no.NA[,7:18])
### Naive Bayes ###
ref.vote.models$NB <- nb.cv(train.no.NA[,'ConstituencyVote'], train.no.NA[,7:18])

#### Constituency Leave Percent - Discretised by Quantile ####
leave.quant.models <- list()
# names(leave.quant.models) <- c("NN", "SVM", "RF", "XGB", "GBM", "GLM", "NB")
### Neural Net ###
leave.quant.models$NN <- nn.cv("GroupedPercent", covars, train.dummy.no.NA, nn.grid)
### SVM ###
leave.quant.models$SVM <- svm.cv("GroupedPercent", covars, train.dummy.no.NA, svm.grid)
### Binary Tree ###
leave.quant.models$Tree <- binary.tree.cv(train[,"GroupedPercent"], train[,7:18])
### Random Forest ###
leave.quant.models$RF <- rf.cv(train.no.NA[,7:18], train.no.NA[,4], na.omit)
### XGBoost Tree ###
leave.quant.models$XGB <- xg.cv2(4, train.no.NA, na.omit)
### Generalized Boosted Regression Modelling ###
leave.quant.models$GBM <- gbm.caret.func(DM = train.no.NA[, 7:18], y = train.no.NA[,4])
### GLM ###
leave.quant.models$GLM <- multinom.cv(train.no.NA[,"GroupedPercent"], train.no.NA[,7:18])
### Naive Bayes ###
leave.quant.models$NB <- nb.cv(train.no.NA[,"GroupedPercent"], train.no.NA[,7:18])

#### Constituency Leave Percent - Discretised by Remain/Leave/StrongLeave ####
leave.disc.models <- list()
# names(leave.disc.models) <- c("NN", "SVM", "RF", "XGB", "GBM", "GLM", "NB")
### Neural Net ###
leave.disc.models$NN <- nn.cv("GroupedPercent2", covars, train.dummy.no.NA, nn.grid)
### SVM ###
leave.disc.models$SVM <- svm.cv("GroupedPercent2", covars, train.dummy.no.NA, svm.grid)
### Binary Tree ###
leave.disc.models$Tree <- binary.tree.cv(train[,"GroupedPercent2"], train[,7:18])
### Random Forest ###
leave.disc.models$RF <- rf.cv(train.no.NA[,7:18], train.no.NA[,5], na.omit)
### XGBoost Tree ###
leave.disc.models$XGB <- xg.cv2(5, train.no.NA, na.omit)
### Generalized Boosted Regression Modelling ###
leave.disc.models$GBM <- gbm.caret.func(DM = train.no.NA[, 7:18], y = train.no.NA[, 5])
### GLM ###
```

```
leave.disc.models$GLM <- multinom.cv(train.no.NA[,"GroupedPercent2"], train.no.NA[,7:18])
### Naive Bayes ###
leave.disc.models$NB <- nb.cv(train.no.NA[,"GroupedPercent2"], train.no.NA[,7:18])

#### Constituency Leave Percent - Continuous ####
leave.percent.models <- list()
# names(leave.percent.models) <- c("NN", "lm", "GBM")
### Neural Net ###
leave.percent.models$NN <- nn.cv("BetterLeavePercent", covars, train.dummy.no.NA, nn.grid)
### Linear Model ###
leave.percent.models$lm <- train(formulaMaker('BetterLeavePercent', covars), data = train.dummy.no.NA,
                                 method = "lm",
                                 trControl = trainControl(
                                   method = "repeatedcv", number = 5, repeats=10, verboseIter = 
                                   allowParallel = TRUE)
)
### Generalized Boosted Regression Model ###
leave.percent.models$GBM <- gbm.caret.func(DM = train.no.NA[, 7:18], y = train.no.NA[, 3])

stopCluster(cores)
```