IPIP Examples and different configurations

Fernando de la Gandara Fernandez

```
library(foreach)
library(doParallel)
```

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

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

```
library(DataExplorer)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
library(parallel)

options(mc.cores = detectCores())

#sink("FILE_OUTPUT.txt")
```

First of all we obtain the needed data and do the preprocessing to the necessary elements as well as set up the output variable.

```
data <- read.csv("../../data_tfg.csv")
ind.cualit <- c(which(names(data) == "SITUACION"),which(names(data)=="SEXO"),which(names(data)=="DM"):wh

 for(i in ind.cualit){
```

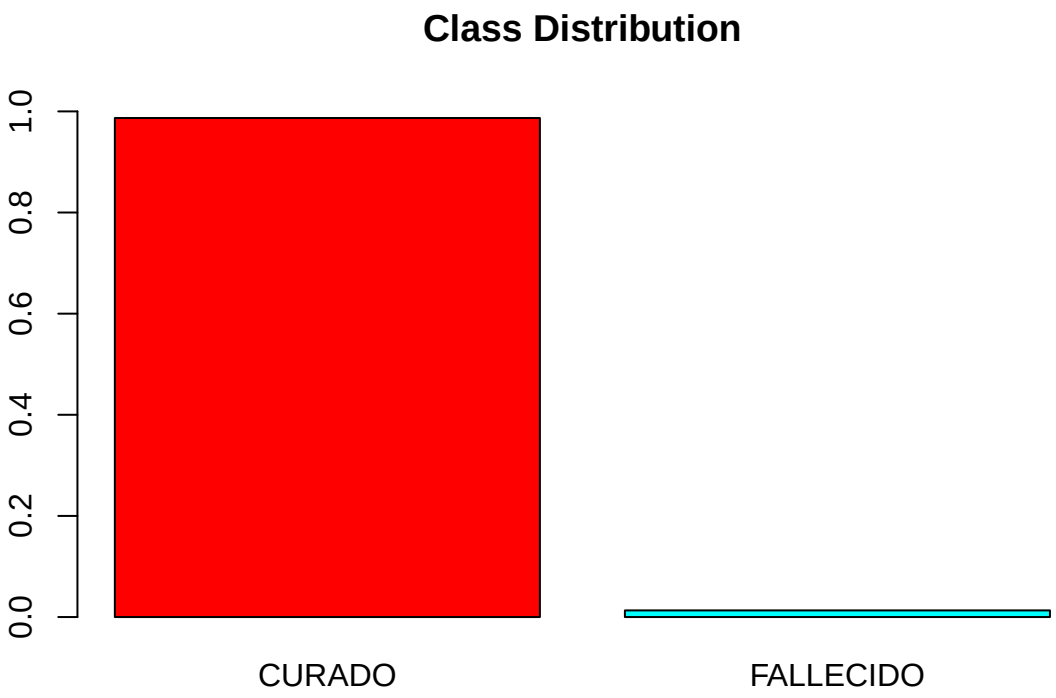
```
  data[,i] <- as.factor(data[, i])

}


OUTPUT_VAR = "SITUACION"
```

We look at imbalanced data in a table

```
barplot(prop.table(table(data[[OUTPUT_VAR]])),
        col = rainbow(2),
        ylim = c(0, 1.01),
        main = "Class Distribution")
```



And the number of elements per class

```
output_lev <- levels(data[[OUTPUT_VAR]])

lev_nrow <- c(nrow(data[data[[OUTPUT_VAR]] == output_lev[1],]), nrow(data[data[[OUTPUT_VAR]] == output_

if (lev_nrow[1] < lev_nrow[2]){
  OUTPUT_MIN = output_lev[1]
  OUTPUT_MAJ = output_lev[2]
  nmin = lev_nrow[1]
  nmax = lev_nrow[2]
}else{
  OUTPUT_MIN = output_lev[2]
  OUTPUT_MAJ = output_lev[1]
  nmin = lev_nrow[2]
  nmaj = lev_nrow[1]
}
print(sprintf("%s : %d", OUTPUT_MIN, nmin))
```

```
## [1] "FALLECIDO : 342"
```

```
print(sprintf("%s : %d", OUTPUT_MAJ, nmaj))
```

```
## [1] "CURADO : 25718"
```

We define the metrics to be measured, amongst which the kappa factor is a significant value to measure imbalanced data and will be the differential measure during the training. One of them will be used in the training and the other for measuring purposes

```
metrics <- function(data, lev = levels(as.factor(data$obs)), model = NULL){
      met <- c(KAPPA = psych::cohen.kappa(cbind(data[, "obs"],data[, "pred"]))$kappa)
  return(met)
}

metrics_all <- function(data, lev = levels(as.factor(data$obs)), model = NULL){

    met <- c(
      ACCURACY = MLmetrics::Accuracy(data[, "pred"], data[, "obs"]),
      SENS = sensitivity(data[, "pred"],data[, "obs"],positive=OUTPUT_MIN,negative=OUTPUT_MAJ),
      SPEC = specificity(data[, "pred"], data[, "obs"],positive=OUTPUT_MIN,negative=OUTPUT_MAJ),
      PPV = posPredValue(data[, "pred"], data[, "obs"],positive=OUTPUT_MIN,negative=OUTPUT_MAJ),
      NPV = negPredValue(data[, "pred"], data[, "obs"],positive=OUTPUT_MIN,negative=OUTPUT_MAJ),
      KAPPA = psych::cohen.kappa(cbind(data[, "obs"],data[, "pred"]))$kappa,
      BAL_ACC = (sensitivity(data[, "pred"],data[, "obs"],positive=OUTPUT_MIN,negative=OUTPUT_MAJ) + sp
    )
  return(met)
}
```

Let us remember the different parameters of our model:

- np: Elements of the minoritary class in each partition
- p: Number of partitions to be made
- b: Maximum number of models in each ensemble of each partition
- mt: Function which returns how many trials we must make in the greedy sequencial approach

In order to stablish these parameters we will create a function for each one of them:

First of all the one related to the “np” parameter which in our case will be reduced to a function which gets the 75% of the minoritary class.

```
calculate_np <- function( nmin, nmaj,  elbow_prob=0.75)
{
  np <- round(nmin*elbow_prob)
  return(np)
}
```

Now we move on to the number of partitions which, in our case, is obtained using the formula studied in the original work, based on applying the logarithmic division of values less than 1 (both negative) to obtain a ratio between the established alpha and the number of elements chosen per partition.

```
calculate_p <- function( np, prob_codo=0.75, alpha_p= .01)
{
  p <- ceiling(log(alpha_p)/(log(1-1/np)*np))
  return(p)
}
```

Subsequently, we approach the parameter b, which is responsible for determining how many models at most there will be in each ensemble of each partition made. To do this, we will take a similar approach to that

of the number of partitions, except that we will use a relationship between nmin and np, and not just the number of minority elements per partition (p). This way, b is always above the number of partitions made.

```
calculate_b <- function(np, nmin, nmaj, elbow_prob=0.75, alpha_b= .01)
{

  b <- ceiling(log(alpha_b)/(log(1-1/nmin)*np))
  return(b)
}
```

And lastly, we will define the "mt" function, which calculates how many tries at most should be made in order to enlarge each model ensemble in the sequencial approach. It can be shown that the "mt" number of tries will always be smaller than the "b" which are the maximum models for each ensemble.

```
mt <- function(b, n) { ceiling((b-n) / 3) }
```

This is an auxiliary function that just returns a continuous apply of a given configuration "b" times. Useful for multiple seed algorithm trainings (i.e five GBM models)

```
get_function_vector <- function(b,function_training){
  function_vector <- c()
  for(i in 1:b){
    function_vector <- append(function_vector, function_training)
  }


  return(function_vector)
}
```

Now let us set up some seed algorithms for the model. This function list will be made by four well known ones:

- Ranger
- Logistic regression
- SVM
- GBM

The idea is to train them in different approaches and see which IPIP variant works better, so we will make a k-fold of each of the next approaches and see which of them seem to be a better idea.

- Each seed algorithm alone (without IPIP)
- Sequential IPIP with many instances of the same seed algorithm
- Exhaustive IPIP with different seed algorithms (one of each at least)
- Exhaustive IPIP but only the best ensemble

```
seed_algorithms <- c(

#RFOREST
 function(df.train, metrics, OUTPUT) {


    tC <- trainControl(
      summaryFunction = metrics,
      method = "cv",
      number = 5,
      allowParallel = TRUE,
      classProbs = TRUE
    )
```

```
    method <- "ranger"
    metric <- "KAPPA"
    maximize <- T

    cl <- makeCluster(detectCores(), type="FORK", outfile="")
    clusterExport(cl, c("OUTPUT_MIN", "OUTPUT_MAJ"))
    registerDoParallel(cl)


    rf <- train(
      as.formula(sprintf("%s ~.", OUTPUT)),
      data = df.train,
      method = method,
      metric = metric,
      maximize = maximize,
      importance = "impurity",
      trControl = tC
    )

    stopCluster(cl)

    return(rf)
  },
#RLOG
 function(df.train, metrics, OUTPUT) {

    tC <- trainControl(
      summaryFunction = metrics,
      allowParallel = TRUE,
      classProbs = TRUE
    )

    method <- "glmnet"
    metric <- "KAPPA"
    maximize <- T


    cl <- makeCluster(detectCores(), type="FORK", outfile="")
    clusterExport(cl, c("OUTPUT_MIN", "OUTPUT_MAJ"))
    registerDoParallel(cl)


    rlog <- train(as.formula(sprintf("%s ~.", OUTPUT)),
      data = df.train,
      method = "glmnet",
      family = 'binomial',
      metric = "KAPPA",
      maximize = T,
      tuneGrid = expand.grid(
        alpha = 0:1,
        lambda = seq(0.0001, 1, length = 100)
      ),
```

```
      trControl = tC
    )


    stopCluster(cl)


    return(rlog)
  },
#SVM
function(df.train, metrics, OUTPUT) {

    tC <- trainControl(
      summaryFunction = metrics,
      allowParallel = TRUE,
      classProbs = TRUE
    )

    method <- "svmLinear"
    metric <- "KAPPA"
    maximize <- T




    cl <- makeCluster(detectCores(), type="FORK", outfile="")
    clusterExport(cl, c("OUTPUT_MIN", "OUTPUT_MAJ"))
    registerDoParallel(cl)


    svm <- train(
      as.formula(sprintf("%s ~.", OUTPUT)),
      data = df.train,
      method = method,
      metric = metric,
      maximize = maximize,
      trControl = tC
    )

      stopCluster(cl)

    return(svm)
  },
#GBM
 function(df.train, metrics, OUTPUT) {

    tC <- trainControl(
      summaryFunction = metrics,
      allowParallel = TRUE,
      classProbs = TRUE,
      verboseIter = FALSE

    )
```

```
    method <- "gbm"
    metric <- "KAPPA"
    maximize <- T


    cl <- makeCluster(detectCores(), type="FORK", outfile="")
    clusterExport(cl, c("OUTPUT_MIN", "OUTPUT_MAJ"))
    registerDoParallel(cl)

    gbm <- train(as.formula(sprintf("%s ~.", OUTPUT)),
                 data = df.train,
                 method = method,
                 metric = metric,
                 maximize = maximize,
                 verbose = FALSE,
                 trControl = tC
    )

    stopCluster(cl)

    return(gbm)
  })
```

Now that we have the configurations, we proceed to obtain a separation into balanced subsets. To do this, I obtain a series of folds with the created function 'imbalancedFold' that allow applying that division so that each subset of the k-fold has the imbalanced subsets distributed

```
imbalancedFold <- function(data, n_folds, target, minority_class) {
  n_samples <- nrow(data)
  n_majority_total <- n_samples - sum(data[[target]] == minority_class)
  n_minority_total <- sum(data[[target]] == minority_class)

  n_minority_per_fold <- ceiling(n_minority_total / n_folds)
  n_majority_per_fold <- ceiling(n_samples / n_folds - n_minority_per_fold)

  fold_indices <- list()

  for (i in 1:n_folds) {

    #We choose the samples of the majoritary class
    majority_indices <- which(data[[target]] != minority_class)
    used_majority_indices <- unlist(fold_indices)
    available_majority_indices <- setdiff(majority_indices, used_majority_indices)

    n_available_majority <- length(available_majority_indices)
    n_majority_this_fold <- min(n_majority_per_fold, n_available_majority)
    selected_majority_indices <- sample(available_majority_indices, size = n_majority_this_fold)

    #We take the samples of the minoritary class

    minority_indices <- which(data[[target]] == minority_class)
    used_minority_indices <- setdiff(used_majority_indices, available_majority_indices)
    available_minority_indices <- setdiff(minority_indices, used_minority_indices)
    n_available_minority <- length(available_minority_indices)
```

```
    n_minority_this_fold <- min(n_minority_per_fold, n_available_minority)
    selected_minority_indices <- sample(available_minority_indices, size = n_minority_this_fold)

    #We collect and combine both indexes
    selected_indices <- c(selected_majority_indices, selected_minority_indices)

    fold_indices[[i]] <- selected_indices
  }

  return(fold_indices)
}
```

In this case we will make a k-fold of k=5

```
folds <-imbalancedFold(data, 5,OUTPUT_VAR, OUTPUT_MIN)
```

Now we define the function that performs the training. It follows the ideas explained in the work. The process consists of creating p partitions where we ensure in each one a quantity of np elements of the minority class, and then training multiple models on these elements.

To do this, we have a list called "dfs" that contains the indices of each well-balanced partition. We will perform iterative training on each of these partitions:

- First, the model is trained on the first function of the function vector, and its metric results are stored.
- Then, we try to expand the model until the "mt" attempt function limits it. To expand the model, it is trained with the next training function given by the parameter of functions, and the value obtained with that new function is compared with the value obtained without it. If an improvement is achieved, it is incorporated into the ensemble; otherwise, we move on to a new attempt.
- Once the number of attempts for the current ensemble is complete, it is considered finished and is incorporated as the final model given for the partition.

We will continue this process for the p partitions, and we will have the model as our final response.

```
train_IPIP <- function( prop.maj, OUTPUT, min_str, maj_str, train.set, test.set,
                        configuration, prediction, metrics, b, np,  p, mt, seed=42){


  set.seed(seed)
  nmin = sum(train.set[[OUTPUT]] == min_str)
  nmaj = sum(train.set[[OUTPUT]] == maj_str)



  majSubSize <-  round(np*prop.maj/(1-prop.maj))
  minSubSize <-  np

  #Incluye en cada posicion los valores de los elementos de dicha particion, de 1 a p
  dfs <- list()

  minoritary <- subset(train.set, train.set[[OUTPUT]] == min_str)
  majoritary<- subset(train.set, train.set[[OUTPUT]] == maj_str)

  for(k in 1:p){
    id.minoritary <- sample(x = 1:nmin, size = minSubSize) #Index for minoritary class for each subset
    id.majoritary <- sample(x= 1:nmaj, size = majSubSize) #Indexes for majoritary class for each subset
```

```
  dfs[[k]] <- rbind(minoritary[id.minoritary,],majoritary[id.majoritary,])
}

# #Full maj class approach
# vals = 1:nmaj
# p=0
# while(length(vals)>majSubSize){
#   p = p+1
#   id.minoritary <- sample(x = 1:nmin, size = minSubSize) #Index for minoritary class for each subse
#   id.majoritary <- sample(vals, size = majSubSize, replace = FALSE) #Indexes for majoritary class f
#
#   vals <- vals[!(vals %in% id.majoritary)]
#   dfs[[p]] <- rbind(minoritary[id.minoritary,],majoritary[id.majoritary,])
#
# }

E <- list() #Final model (ensemble of ensembles)


for(k in 1:p){
  Ek <- list() # k-esim ensemble
  i <- 0 #Counter for number of tries of enlarging the ensemble

  #Balanced partition

  df <- dfs[[k]]
  model_i = 1
  while(length(Ek)<=b && i<mt(b,length(Ek))){
    #We select the elements for the training

    majoritary <- which(df[[OUTPUT]] == maj_str)
    minoritary <- which(df[[OUTPUT]] == min_str)
    ind.train <- c(
      sample(majoritary, size = majSubSize, replace = TRUE),
      sample(minoritary, size = minSubSize, replace = TRUE)
    )

    #We train with the models in configuration with a sequential approach
    cat(sprintf("Training element %d in %d possible values\n", i+1,mt(b,length(Ek))))
    model <- configuration[[length(Ek)+1]](df[ind.train,], metrics, OUTPUT)

    metrics.ensemble <-
      if (length(Ek)==0){
        u <- -Inf;
        names(u) <- "KAPPA";
        u;
      } else{
        metrics(data.frame(
        obs = test.set[[OUTPUT]],
        pred = as.factor(prediction(Ek, test.set[colnames(test.set)!=OUTPUT]))
        ))
      }
```

```
      Ek[[length(Ek)+1]] <- model
      metrics.ensemble.new <- metrics(data.frame(
      obs = test.set[[OUTPUT]],
      pred= as.factor(prediction(Ek, test.set[colnames(test.set)!=OUTPUT]))
      ))

      #We check if the new model changes the result. If it does not we start trying again
      if(metrics.ensemble.new["KAPPA"] <= metrics.ensemble["KAPPA"]){
        i <- i+1
        Ek[[length(Ek)]] <- NULL
      } else{
        #If the ensemble tries to enlarge again, we restart the enlarging posibilities.
        i <- 0
      }
    } # End of the k-esim ensemble building

    E[[length(E)+1]] <- Ek

  }
return(E);
}
```

Now we have another possible approach of IPIP based on an exhaustive approach of the models.

```
powerset = function(s, test.set){
    len = length(s)
    l = vector(mode="list",length=2^len);l[[1]]=numeric()
    counter = 1L
    for(x in 1L:len){
        xUpred <- list("model"= s[x], "prediction"=predict(s[x], test.set))
        for(subset in 1L:counter){
            counter=counter+1L
            l[[counter]] = c(l[[subset]],xUpred)
        }
    }
    return(tail(l, length(l)-1))
}


train_IPIP_exhaustive <- function( prop.maj, OUTPUT, min_str, maj_str, train.set, test.set, configurati


  #Set random seed
  set.seed(seed)

  nmin = sum(train.set[[OUTPUT]] == min_str)
  nmaj = sum(train.set[[OUTPUT]] == maj_str)



  majSubSize <-  round(np*prop.maj/(1-prop.maj))
  minSubSize <-  np

  #Incluye en cada posicion los valores de los elementos de dicha particion, de 1 a p
```

```r
dfs <- list()

minoritary <- subset(train.set, train.set[[OUTPUT]] == min_str)
majoritary <- subset(train.set, train.set[[OUTPUT]] == maj_str)

for(k in 1:p){
  id.minoritary <- sample(x = 1:nmin, size = minSubSize) #Index for minoritary class for each subset
  id.majoritary <- sample(x= 1:nmaj, size = majSubSize) #Indexes for majoritary class for each subset

  dfs[[k]] <- rbind(minoritary[id.minoritary,],majoritary[id.majoritary,])
}

# #Full set approach
# vals = 1:nmaj
# p=0
# while(length(vals)>majSubSize){
#   p = p+1
#   id.minoritary <- sample(x = 1:nmin, size = minSubSize) #Index for minoritary class for each subse
#   id.majoritary <- sample(vals, size = majSubSize, replace = FALSE) #Indexes for majoritary class f
#
#   vals <- vals[!(vals %in% id.majoritary)]
#   dfs[[p]] <- rbind(minoritary[id.minoritary,],majoritary[id.majoritary,])
#
# }


E <- list() #Final model (ensemble of ensembles)


for(k in 1:p){

  cat(sprintf("Ensemble number %d of %d\n", k,p))
  Ek <- list() # k-esim ensemble
  init = Sys.time()

  #Balanced partition

  df <- dfs[[k]]
  majoritary <- which(df[[OUTPUT]] == maj_str)
  minoritary <- which(df[[OUTPUT]] == min_str)

    ind.train <- c(
      sample(majoritary, size = majSubSize, replace = TRUE),
      sample(minoritary, size = minSubSize, replace = TRUE)
    )

    Ek <- list()
    for(conf in 1:length(configuration)){
      cat(sprintf("Configuration number %d\n", conf))
      Ek[[length(Ek)+1]] <- configuration[[conf]](df[ind.train,], metrics, OUTPUT)

    }
    all_sets <- powerset(Ek, as.data.frame(test.set[, -which(names(test.set) == OUTPUT)]))
```

```
      max_kappa <- function(model){
        pred = prediction.predicted(model)
          metrics(data.frame(
              obs = test.set[[OUTPUT]],
              pred = as.factor(pred)
          ))["KAPPA"]
      }



      #Run the kappa check in parallel mode
      kappa_list <- lapply(all_sets, max_kappa)
      #print(cbind(lapply(all_sets, function(x) length(x)/2), kappa_list))
      # find the maximum kappa value and corresponding set

      max_kappa_value <- max(unlist(kappa_list))
      max_set <- all_sets[[which.max(unlist(kappa_list))]]
      cat(sprintf("Max kappa of ensemble is %f with length %d\n", max_kappa_value, length(max_set$model

      E[[length(E)+1]] <- max_set$model

  }
return(E);
}
```

Now the prediction function is missing. In our case it consists of two functions.

The first one is applied to the ensemble of models of a specific partition. On each established model, the specific prediction (predict method of the same) is made. Then a table is made that assigns to each example the proportion of models that have predicted in that way. If more than q (75% by default, that is, at least three-quarters of the models predict that it is cured) have been predicted, it is established that it is predicted as OUTPUT_MAJ and if not, it will be deceased.

Finally, the same thing is done but instead of using the predict function of each model, the given prediction function on each ensemble is used, the proportion is made on the ensemble sets and if the prediction proportion value is >50%, it is assumed as OUTPUT_MAJ being if not OUTPUT_MIN.

```
prediction <- function(conj.model, x, q = 0.75){ #q=0.75, pero se deberían probar valores como 0.5, 0.2
  pred <- data.frame(matrix(nrow=nrow(x),ncol=0))
  for(model in conj.model) pred <- cbind(pred, predict(model,x))
  nElems = ncol(pred)
  nElems <- ncol(pred)
  counts <- rowSums(pred == OUTPUT_MAJ)
  pred <- counts / nElems
  ifelse(is.na(pred) | pred<q, OUTPUT_MIN, OUTPUT_MAJ)
}

prediction.final <- function(ensemble, x, q = 0.75){
  # Colocamos en cada fila de un conjunto de data todas las predictiones para una muestra
  pred <- as.data.frame(lapply(ensemble, function(e) prediction(e,x)))
  pred <- apply(pred, 1, function(x) prop.table(table(x))[OUTPUT_MAJ])
  ifelse(is.na(pred) | pred<q, OUTPUT_MIN, OUTPUT_MAJ)
}

prediction.predicted <- function(predicted, q = 0.75){
```

```
  pred <- data.frame(matrix(nrow=length(predicted$prediction),ncol=0))
  pred <- cbind(pred,predicted$prediction)
  nElems = ncol(pred)
  nElems <- ncol(pred)
  counts <- rowSums(pred == OUTPUT_MAJ)
  pred <- counts / nElems
  ifelse(is.na(pred) | pred<q, OUTPUT_MIN, OUTPUT_MAJ)
}


prop.maj = 0.5
```

Now we move on to performing the training for each configuration established in the configuration list. To do this, we iterate through each configuration and apply k-fold cross-validation. For each fold, we provide a vector function associated with the maximum number of models b for an ensemble. This approach allows for the creation of vectors of variable functions

First we will start with seed algorithms performing with the direct data

```
mean_metrics.seed <- list()
mean_time.seed <- list()


cat("######## Training seed algorithms ######## \n")
```

```
## ######## Training seed algorithms ########
```

```
for(alg in 1:length(seed_algorithms)){
    mean_time.seed[[alg]] = 0

    cat(sprintf("Seed algorithm: %d\n", alg))
    for (i in 1:length(folds)) {
        cat(sprintf("Fold %d out of %d\n", i, length(folds)))
        cat("---------------------------------------\n")


        metrics.final.seed <- list()

        train.set <- data[unlist(folds[i]),]
        test.set <- data[-unlist(folds[i]),]


        start_time <- Sys.time()
        model_seed <- seed_algorithms[[alg]](train.set, metrics, OUTPUT_VAR)
        end_time <- Sys.time()
        mean_time.seed[[alg]]= mean_time.seed[[alg]] +
        end_time - start_time
        print(end_time - start_time)

        metrics.final.seed <- append( metrics.final.seed, metrics_all(data.frame(
          obs = test.set[[OUTPUT_VAR]],
          pred = predict(model_seed,test.set)
        )))
        cat("---------------------------------------\n")
    }
```

```
  mean_metrics.seed <- append(mean_metrics.seed,
        apply(matrix(unlist(metrics.final.seed), ncol= 7, byrow=T), 2, mean))
}
```

```
## Seed algorithm: 1
## Fold 1 out of 5
## ------------------------------------
## Time difference of 9.155498 secs
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
## Time difference of 8.283856 secs
## ------------------------------------
## Fold 3 out of 5
## ------------------------------------
## Time difference of 10.80912 secs
## ------------------------------------
## Fold 4 out of 5
## ------------------------------------
## Time difference of 10.78391 secs
## ------------------------------------
## Fold 5 out of 5
## ------------------------------------
## Time difference of 12.83436 secs
## ------------------------------------
## Seed algorithm: 2
## Fold 1 out of 5
## ------------------------------------
## Time difference of 5.539548 secs
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
## Time difference of 6.020092 secs
## ------------------------------------
## Fold 3 out of 5
## ------------------------------------
## Time difference of 5.110557 secs
## ------------------------------------
## Fold 4 out of 5
## ------------------------------------
## Time difference of 5.475613 secs
## ------------------------------------
## Fold 5 out of 5
## ------------------------------------
## Time difference of 5.35451 secs
## ------------------------------------
## Seed algorithm: 3
## Fold 1 out of 5
## ------------------------------------
## maximum number of iterations reached 0.007917966 0.007766793Time difference of 8.46624 secs
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
```

```
## maximum number of iterations reached 0.0127347 0.01186344Time difference of 7.294417 secs
## -------------------------------------
## Fold 3 out of 5
## -------------------------------------
## maximum number of iterations reached 0.01269138 0.01150269Time difference of 7.034344 secs
## -------------------------------------
## Fold 4 out of 5
## -------------------------------------
## maximum number of iterations reached 0.01516531 0.01331701Time difference of 6.073045 secs
## -------------------------------------
## Fold 5 out of 5
## -------------------------------------
## maximum number of iterations reached 0.003853056 0.00383101Time difference of 6.069788 secs
## -------------------------------------
## Seed algorithm: 4
## Fold 1 out of 5
## -------------------------------------
## Time difference of 6.517729 secs
## -------------------------------------
## Fold 2 out of 5
## -------------------------------------
## Time difference of 6.57786 secs
## -------------------------------------
## Fold 3 out of 5
## -------------------------------------
## Time difference of 6.82507 secs
## -------------------------------------
## Fold 4 out of 5
## -------------------------------------
## Time difference of 6.372718 secs
## -------------------------------------
## Fold 5 out of 5
## -------------------------------------
## Time difference of 6.359823 secs
## -------------------------------------
```

```
matrix_mean.seed <- matrix(mean_metrics.seed, nrow = 4, ncol = 7, byrow = TRUE)
matrix_mean.seed<- cbind(matrix_mean.seed, as.numeric(mean_time.seed, units = "secs")/length(folds))

col_names <- c("ACCURACY", "SENS", "SPEC", "PPV", "NPV", "KAPPA", "BAL_ACC", "TIME")
row_names <- c("RANGER", "RLOG", "SVM", "GBM")

colnames(matrix_mean.seed) <- col_names
rownames(matrix_mean.seed) <- row_names


cat("SEED ALGORITHM\n")
```

```
## SEED ALGORITHM
```

```
print(matrix_mean.seed)
```

```
##        ACCURACY  SENS       SPEC      PPV       NPV       KAPPA      BAL_ACC
## RANGER 0.984701  0.04710145 0.9972783 0.1884058 0.9873448 0.07044057 0.5221898
## RLOG   0.9866193 0.04347826 0.999271  0.4444444 0.9873223 0.07703061 0.5213746
```

```
## SVM    0.9867632 0          1         NaN       0.9867632 0          0.5
## GBM    0.9848449 0.06521739 0.997181  0.2368421 0.9875812 0.09711193 0.5311992
##        TIME
## RANGER 10.37335
## RLOG   5.500064
## SVM    6.987567
## GBM    6.53064
```

Now we will perform the training in the mixed algorithm exhaustive approach of IPIP

```
mean_metrics.IPIPexhaustMixed <- list()
mean_time.IPIPexhaustMixed = 0
metrics.final.IPIPexhaustMixed <- list()
cat("######## Training mixed exhaustive IPIP ######## \n")
```

```
## ######## Training mixed exhaustive IPIP ########
```

```
 for (i in 1:length(folds)) {
    cat(sprintf("Fold %d out of %d\n",i, length(folds)))
    cat("---------------------------------------\n")



    train.set <- data[unlist(folds[i]),]
    test.set <- data[-unlist(folds[i]),]

    nmin = sum(train.set[[OUTPUT_VAR]] == OUTPUT_MIN)
    nmaj = sum(train.set[[OUTPUT_VAR]] == OUTPUT_MAJ)

    np <- calculate_np( nmin, nmaj)
    p <- calculate_p(np)
    b <- calculate_b(np, nmin, nmaj)


    start_time <- Sys.time()
    ensemble.fold <- train_IPIP_exhaustive(prop.maj, OUTPUT_VAR, OUTPUT_MIN,
          OUTPUT_MAJ, train.set, test.set, seed_algorithms, prediction,  metrics,np, p)

    end_time <- Sys.time()
    mean_time.IPIPexhaustMixed = mean_time.IPIPexhaustMixed + as.numeric(end_time - start_time)

    cat(sprintf("Ensembles length in exhaustive MIXED fold: %d\nLength of ensembles:", i))
    cat(unlist(lapply(ensemble.fold,length)))
    cat("\n")

    metrics.final.IPIPexhaustMixed <- append( metrics.final.IPIPexhaustMixed,
      metrics_all(data.frame(
      obs = test.set[[OUTPUT_VAR]],
      pred= as.factor(prediction.final(ensemble.fold,
                                       test.set[,names(test.set) != OUTPUT_VAR]))
      )))
      cat("---------------------------------------\n")

 }
```

```
## Fold 1 out of 5
```

```
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.179178 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.139501 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.113958 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.111354 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.187675 with length 1
## Ensembles length in exhaustive MIXED fold: 1
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.168608 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.157071 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.164824 with length 1
```

```
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.168682 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.129207 with length 1
## Ensembles length in exhaustive MIXED fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.206487 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.178343 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.103920 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.129421 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.146885 with length 1
## Ensembles length in exhaustive MIXED fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
```

```
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.204033 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.173570 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.126960 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.127182 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.222175 with length 1
## Ensembles length in exhaustive MIXED fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.101640 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.126693 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.134417 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
```

```
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.173458 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Max kappa of ensemble is 0.139198 with length 1
## Ensembles length in exhaustive MIXED fold: 5
## Length of ensembles:1 1 1 1 1
## -------------------------------------
```

```
mean_metrics.IPIPexhaustMixed <- append(mean_metrics.IPIPexhaustMixed,
  apply(matrix(unlist(metrics.final.IPIPexhaustMixed), ncol= 7, byrow=T), 2, mean))

matrix_mean.IPIPexhaustMixed <- matrix(mean_metrics.IPIPexhaustMixed, nrow = 1, ncol = 7, byrow = TRUE)
matrix_mean.IPIPexhaustMixed <- cbind(matrix_mean.IPIPexhaustMixed, as.numeric(mean_time.IPIPexhaustMix

col_names <- c("ACCURACY", "SENS", "SPEC", "PPV", "NPV", "KAPPA", "BAL_ACC", "TIME")


colnames(matrix_mean.IPIPexhaustMixed) <- col_names

cat("IPIP EXHAUST MIXED\n")
```

```
## IPIP EXHAUST MIXED
```

```
print(matrix_mean.IPIPexhaustMixed)
```

```
##      ACCURACY SENS      SPEC      PPV         NPV       KAPPA     BAL_ACC
## [1,] 0.858274 0.9443781 0.8571276 0.08132061 0.9991435 0.1285999 0.9007528
##      TIME
## [1,] 34.24929
```

We then try the exhaustive method applied to a group of algorithms of the same kind

```
mean_metrics.IPIPexhaustRepeat <- list()
mean_time.IPIPexhaustRepeat <- list()

cat("######## Training exhaustive repeat IPIP ######## \n")
```

```
## ######## Training exhaustive repeat IPIP ########
```

```
for(alg in 1:length(seed_algorithms)){

   cat(sprintf("Seed algorithm: %d\n", alg))

   mean_time.IPIPexhaustRepeat[[alg]] = 0
   metrics.final.IPIPexhaustRepeat <- list()
   for (i in 1:length(folds)) {
      cat(sprintf("Fold %d out of %d\n",i, length(folds)))
      cat("---------------------------------------\n")

      train.set <- data[unlist(folds[i]),]
      test.set <- data[-unlist(folds[i]),]
```

```
      nmin = sum(train.set[[OUTPUT_VAR]] == OUTPUT_MIN)
      nmaj = sum(train.set[[OUTPUT_VAR]] == OUTPUT_MAJ)

      np <- calculate_np( nmin, nmaj)
      p <- calculate_p(np)
      b <- calculate_b(np, nmin, nmaj)

      conf<- get_function_vector(b, seed_algorithms[alg])

      start_time <- Sys.time()

      ensemble.fold <- train_IPIP_exhaustive(prop.maj, OUTPUT_VAR, OUTPUT_MIN, OUTPUT_MAJ, train.set, t

      end_time <- Sys.time()
      mean_time.IPIPexhaustRepeat[[alg]]= mean_time.IPIPexhaustRepeat[[alg]] + end_time - start_time

      cat(sprintf("Ensembles length in exhaustive REPEAT fold: %d\nLength of ensembles:", i))
      cat(unlist(lapply(ensemble.fold,length)))
      cat("\n")


      metrics.final.IPIPexhaustRepeat <- append( metrics.final.IPIPexhaustRepeat,
        metrics_all(data.frame(
        obs = test.set[[OUTPUT_VAR]],
        pred= as.factor(prediction.final(ensemble.fold,
                                         test.set[,names(test.set) != OUTPUT_VAR]))
        )))
          cat("-------------------------------------\n")

   }
   mean_metrics.IPIPexhaustRepeat <- append(mean_metrics.IPIPexhaustRepeat,
    apply(matrix(unlist(metrics.final.IPIPexhaustRepeat), ncol= 7, byrow=T), 2, mean))
}
```

```
## Seed algorithm: 1
## Fold 1 out of 5
## -------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.174828 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
```

```
## Max kappa of ensemble is 0.165538 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.174964 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.129584 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.177338 with length 1
## Ensembles length in exhaustive REPEAT fold: 1
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.174625 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.144064 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
```

```
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.169728 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.164704 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.171854 with length 1
## Ensembles length in exhaustive REPEAT fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.185006 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.112012 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
```

```
## Configuration number 7
## Max kappa of ensemble is 0.190088 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.101294 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.130453 with length 1
## Ensembles length in exhaustive REPEAT fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.211205 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.179790 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.140660 with length 1
## Ensemble number 4 of 5
## Configuration number 1
```

```
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.129162 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.133415 with length 1
## Ensembles length in exhaustive REPEAT fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.073723 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.115094 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.138050 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
```

```
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.155147 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.160969 with length 1
## Ensembles length in exhaustive REPEAT fold: 5
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Seed algorithm: 2
## Fold 1 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.162394 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.153677 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.142396 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.157102 with length 1
```

```
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.144070 with length 1
## Ensembles length in exhaustive REPEAT fold: 1
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 2 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.151378 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.083545 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.175007 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.169345 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
```

```
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.139959 with length 1
## Ensembles length in exhaustive REPEAT fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.130625 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.145277 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.107653 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.129421 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
```

```
## Max kappa of ensemble is 0.162267 with length 1
## Ensembles length in exhaustive REPEAT fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.158305 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.198792 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.113325 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.185373 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.166772 with length 1
## Ensembles length in exhaustive REPEAT fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
```

```
## Fold 5 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.106367 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.084650 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.165917 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.140565 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.133726 with length 1
## Ensembles length in exhaustive REPEAT fold: 5
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Seed algorithm: 3
## Fold 1 out of 5
## -----------------------------------
## Ensemble number 1 of 5
```

```
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.173114 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.162787 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.121479 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.117586 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.107668 with length 1
## Ensembles length in exhaustive REPEAT fold: 1
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
```

```
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.175406 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.131888 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.188867 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.166137 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.146396 with length 1
## Ensembles length in exhaustive REPEAT fold: 2
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Fold 3 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.216489 with length 1
```

```
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.142078 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.194726 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.097783 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.139067 with length 1
## Ensembles length in exhaustive REPEAT fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.191810 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
```

```
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.209478 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.175733 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.163179 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.150965 with length 1
## Ensembles length in exhaustive REPEAT fold: 4
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Fold 5 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.102388 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
```

```
## Max kappa of ensemble is 0.043493 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.091361 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.113425 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.152199 with length 1
## Ensembles length in exhaustive REPEAT fold: 5
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Seed algorithm: 4
## Fold 1 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.194564 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.200021 with length 1
## Ensemble number 3 of 5
## Configuration number 1
```

```
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.140246 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.149238 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.148603 with length 1
## Ensembles length in exhaustive REPEAT fold: 1
## Length of ensembles:1 1 1 1 1
## ------------------------------------
## Fold 2 out of 5
## ------------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.169510 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.144109 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
```

```
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.204164 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.167128 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.201589 with length 1
## Ensembles length in exhaustive REPEAT fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.165909 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.145600 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.148354 with length 1
## Ensemble number 4 of 5
```

```
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.111883 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.162129 with length 1
## Ensembles length in exhaustive REPEAT fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.201064 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.159515 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.175156 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
```

```
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.148347 with length 1
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.186238 with length 1
## Ensembles length in exhaustive REPEAT fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Ensemble number 1 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.082338 with length 1
## Ensemble number 2 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.101916 with length 1
## Ensemble number 3 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.179073 with length 1
## Ensemble number 4 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.176960 with length 1
```

```
## Ensemble number 5 of 5
## Configuration number 1
## Configuration number 2
## Configuration number 3
## Configuration number 4
## Configuration number 5
## Configuration number 6
## Configuration number 7
## Max kappa of ensemble is 0.162165 with length 1
## Ensembles length in exhaustive REPEAT fold: 5
## Length of ensembles:1 1 1 1 1
## -------------------------------------
```

```
matrix_mean.IPIPexhaustRepeat <- matrix(mean_metrics.IPIPexhaustRepeat, nrow = 4, ncol = 7, byrow = TRU
matrix_mean.IPIPexhaustRepeat <- cbind(matrix_mean.IPIPexhaustRepeat, as.numeric(mean_time.IPIPexhaustR

col_names <- c("ACCURACY", "SENS", "SPEC", "PPV", "NPV", "KAPPA", "BAL_ACC", "TIME")
row_names <- c("RANGER", "RLOG", "SVM", "GBM")

colnames(matrix_mean.IPIPexhaustRepeat) <- col_names
rownames(matrix_mean.IPIPexhaustRepeat) <- row_names

cat("IPIP EXHAUST REPEATED\n")
```

```
## IPIP EXHAUST REPEATED
```

```
print(matrix_mean.IPIPexhaustRepeat)
```

```
##        ACCURACY  SENS      SPEC      PPV         NPV       KAPPA     BAL_ACC
## RANGER 0.8602221 0.9304587 0.8592855 0.082886   0.9989381 0.1309501 0.8948721
## RLOG   0.8571999 0.9407469 0.8560875 0.08085972 0.9990842 0.1277106 0.8984172
## SVM    0.8446253 0.9399506 0.8433536 0.0784049  0.9990771 0.1229564 0.8916521
## GBM    0.87299   0.9304427 0.8722236 0.08964336 0.9989511 0.1427782 0.9013332
##        TIME
## RANGER 0.8409693
## RLOG   1.664061
## SVM    1.347163
## GBM    0.9503123
```

Let us finally execute the code on the sequential greedy original approach of IPIP

```
mean_metrics.IPIPrepeated <- list()
mean_time.IPIPrepeated <- list()

cat("######## Training sequential repeat IPIP ######## \n")
```

```
## ######## Training sequential repeat IPIP ########
```

```
for(alg in 1:length(seed_algorithms)){

   cat(sprintf("Seed algorithm: %d\n", alg))
   mean_time.IPIPrepeated[[alg]] = 0
   metrics.final.IPIPrepeated <- list()

   for (i in 1:length(folds)) {
      cat(sprintf("Fold %d out of %d\n",i, length(folds)))
      cat("---------------------------------\n")
```

```
      train.set <- data[unlist(folds[i]),]
      test.set <- data[-unlist(folds[i]),]

      nmin = sum(train.set[[OUTPUT_VAR]] == OUTPUT_MIN)
      nmaj = sum(train.set[[OUTPUT_VAR]] == OUTPUT_MAJ)

      np <- calculate_np( nmin, nmaj)
      p <- calculate_p(np)
      b <- calculate_b(np, nmin, nmaj)

      conf<- get_function_vector(b, seed_algorithms[alg])

      start_time <- Sys.time()
      ensemble.fold <- train_IPIP(prop.maj, OUTPUT_VAR, OUTPUT_MIN, OUTPUT_MAJ, train.set, test.set, co
      end_time <- Sys.time()
      mean_time.IPIPrepeated[[alg]]= mean_time.IPIPrepeated[[alg]] +
              end_time - start_time


      cat(sprintf("Ensembles length in sequential in fold: %d\nLength of ensembles:", i))
      cat(unlist(lapply(ensemble.fold,length)))
      cat("\n")


      metrics.final.IPIPrepeated <- append( metrics.final.IPIPrepeated,
        metrics_all(data.frame(
        obs = test.set[[OUTPUT_VAR]],
        pred= as.factor(prediction.final(ensemble.fold,
                                         test.set[,names(test.set) != OUTPUT_VAR]))
        )))
        cat("-----------------------------------\n")

  }

  mean_metrics.IPIPrepeated <- append(mean_metrics.IPIPrepeated, apply(matrix(unlist(metrics.final.IPIPr

}
```

```
## Seed algorithm: 1
## Fold 1 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
```

```
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 1
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 2 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
```

```
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 5
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Seed algorithm: 2
## Fold 1 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 1
## Length of ensembles:1 1 2 1 1
## -----------------------------------
```

```
## Fold 2 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 3
## Length of ensembles:1 1 1 1 2
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
```

```
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 5
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Seed algorithm: 3
## Fold 1 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 1 in 1 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 1
## Length of ensembles:1 1 1 4 1
## -----------------------------------
## Fold 2 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
```

```
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 3
## Length of ensembles:1 2 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
```

```
## Ensembles length in sequential in fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 5
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Seed algorithm: 4
## Fold 1 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 1
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 2 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
```

```
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 2
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 3 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 3
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 4 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 4
## Length of ensembles:1 1 1 1 1
## -----------------------------------
## Fold 5 out of 5
## -----------------------------------
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
```

```
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 1 in 2 possible values
## Training element 1 in 1 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Training element 1 in 3 possible values
## Training element 1 in 2 possible values
## Training element 2 in 2 possible values
## Ensembles length in sequential in fold: 5
## Length of ensembles:1 4 1 1 1
## -------------------------------------
```

```
matrix_mean.IPIPrepeated <- matrix(mean_metrics.IPIPrepeated, nrow = 4, ncol = 7, byrow = TRUE)
matrix_mean.IPIPrepeated <- cbind(matrix_mean.IPIPrepeated, as.numeric(mean_time.IPIPrepeated, units = "

col_names <- c("ACCURACY", "SENS", "SPEC", "PPV", "NPV", "KAPPA", "BAL_ACC", "TIME")
row_names <- c("RANGER", "RLOG", "SVM", "GBM")



colnames(matrix_mean.IPIPrepeated) <- col_names
rownames(matrix_mean.IPIPrepeated) <- row_names


cat("IPIP SEQUENTIAL REPEAT\n")
```

```
## IPIP SEQUENTIAL REPEAT
```

```
print(matrix_mean.IPIPrepeated)
```

```
##        ACCURACY  SENS      SPEC      PPV         NPV       KAPPA     BAL_ACC
## RANGER 0.8310989 0.9487418 0.829531  0.0725546  0.9991988 0.112718  0.8891364
## RLOG   0.8408635 0.9502389 0.839407  0.07507122 0.9992207 0.117398  0.894823
## SVM    0.8502928 0.9465759 0.8490109 0.07770886 0.9991696 0.1222223 0.8977934
## GBM    0.8494405 0.9421564 0.8482041 0.0797514  0.9991129 0.1254388 0.8951803
##        TIME
## RANGER 0.5071666
## RLOG   0.7250962
## SVM    0.9007185
## GBM    0.3088666
```

After executing the code, we will show the comparison of relevant metrics:

```
cat("IPIP SEQUENTIAL REPEAT\n")
print(matrix_mean.IPIPrepeated)

cat("IPIP EXHAUST REPEATED\n")
print(matrix_mean.IPIPexhaustRepeat)
```

```
cat("IPIP EXHAUST MIXED\n")
print(matrix_mean.IPIPexhaustMixed)

cat("SEED ALGORITHM\n")
print(matrix_mean.seed)
```

```
## IPIP SEQUENTIAL REPEAT
##        ACCURACY  SENS      SPEC      PPV         NPV       KAPPA     BAL_ACC
## RANGER 0.8310989 0.9487418 0.829531  0.0725546  0.9991988 0.112718  0.8891364
## RLOG   0.8408635 0.9502389 0.839407  0.07507122 0.9992207 0.117398  0.894823
## SVM    0.8502928 0.9465759 0.8490109 0.07770886 0.9991696 0.1222223 0.8977934
## GBM    0.8494405 0.9421564 0.8482041 0.0797514  0.9991129 0.1254388 0.8951803
##        TIME
## RANGER 0.5071666
## RLOG   0.7250962
## SVM    0.9007185
## GBM    0.3088666
## IPIP EXHAUST REPEATED
##        ACCURACY  SENS      SPEC      PPV         NPV       KAPPA     BAL_ACC
## RANGER 0.8602221 0.9304587 0.8592855 0.082886   0.9989381 0.1309501 0.8948721
## RLOG   0.8571999 0.9407469 0.8560875 0.08085972 0.9990842 0.1277106 0.8984172
## SVM    0.8446253 0.9399506 0.8433536 0.0784049  0.9990771 0.1229564 0.8916521
## GBM    0.87299   0.9304427 0.8722236 0.08964336 0.9989511 0.1427782 0.9013332
##        TIME
## RANGER 0.8409693
## RLOG   1.664061
## SVM    1.347163
## GBM    0.9503123
## IPIP EXHAUST MIXED
##      ACCURACY SENS      SPEC      PPV         NPV       KAPPA     BAL_ACC
## [1,] 0.858274 0.9443781 0.8571276 0.08132061 0.9991435 0.1285999 0.9007528
##      TIME
## [1,] 34.24929
## SEED ALGORITHM
##        ACCURACY  SENS       SPEC      PPV       NPV       KAPPA      BAL_ACC
## RANGER 0.984701  0.04710145 0.9972783 0.1884058 0.9873448 0.07044057 0.5221898
## RLOG   0.9866193 0.04347826 0.999271  0.4444444 0.9873223 0.07703061 0.5213746
## SVM    0.9867632 0          1         NaN       0.9867632 0          0.5
## GBM    0.9848449 0.06521739 0.997181  0.2368421 0.9875812 0.09711193 0.5311992
##        TIME
## RANGER 10.37335
## RLOG   5.500064
## SVM    6.987567
## GBM    6.53064
```