FINAL PROJECT TWO

Data classification and prediction of breast cancer

URL LINK GITHUB: <https://github.com/CB560/CB560Repository>

library(ggplot2)  
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(knitr)  
library(ggthemes)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

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

library(corrplot)

## corrplot 0.84 loaded

library(reshape)

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':  
##   
## rename

library(caret)

## Loading required package: lattice

library(caretEnsemble)

##   
## Attaching package: 'caretEnsemble'

## The following object is masked from 'package:ggplot2':  
##   
## autoplot

library(caTools)  
library(data.table)

##   
## Attaching package: 'data.table'

## The following object is masked from 'package:reshape':  
##   
## melt

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

library(xgboost)

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

library(gbm)

## Loaded gbm 2.1.8

library(kernlab)

##   
## Attaching package: 'kernlab'

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

library(C50)  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:reshape':  
##   
## expand

## Loaded glmnet 4.1-1

library(magrittr)

# Read the data  
df <- read.csv("data.csv", stringsAsFactors = TRUE)  
  
str(df)

## 'data.frame': 569 obs. of 33 variables:  
## $ id : int 842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...  
## $ diagnosis : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...  
## $ radius\_mean : num 18 20.6 19.7 11.4 20.3 ...  
## $ texture\_mean : num 10.4 17.8 21.2 20.4 14.3 ...  
## $ perimeter\_mean : num 122.8 132.9 130 77.6 135.1 ...  
## $ area\_mean : num 1001 1326 1203 386 1297 ...  
## $ smoothness\_mean : num 0.1184 0.0847 0.1096 0.1425 0.1003 ...  
## $ compactness\_mean : num 0.2776 0.0786 0.1599 0.2839 0.1328 ...  
## $ concavity\_mean : num 0.3001 0.0869 0.1974 0.2414 0.198 ...  
## $ concave.points\_mean : num 0.1471 0.0702 0.1279 0.1052 0.1043 ...  
## $ symmetry\_mean : num 0.242 0.181 0.207 0.26 0.181 ...  
## $ fractal\_dimension\_mean : num 0.0787 0.0567 0.06 0.0974 0.0588 ...  
## $ radius\_se : num 1.095 0.543 0.746 0.496 0.757 ...  
## $ texture\_se : num 0.905 0.734 0.787 1.156 0.781 ...  
## $ perimeter\_se : num 8.59 3.4 4.58 3.44 5.44 ...  
## $ area\_se : num 153.4 74.1 94 27.2 94.4 ...  
## $ smoothness\_se : num 0.0064 0.00522 0.00615 0.00911 0.01149 ...  
## $ compactness\_se : num 0.049 0.0131 0.0401 0.0746 0.0246 ...  
## $ concavity\_se : num 0.0537 0.0186 0.0383 0.0566 0.0569 ...  
## $ concave.points\_se : num 0.0159 0.0134 0.0206 0.0187 0.0188 ...  
## $ symmetry\_se : num 0.03 0.0139 0.0225 0.0596 0.0176 ...  
## $ fractal\_dimension\_se : num 0.00619 0.00353 0.00457 0.00921 0.00511 ...  
## $ radius\_worst : num 25.4 25 23.6 14.9 22.5 ...  
## $ texture\_worst : num 17.3 23.4 25.5 26.5 16.7 ...  
## $ perimeter\_worst : num 184.6 158.8 152.5 98.9 152.2 ...  
## $ area\_worst : num 2019 1956 1709 568 1575 ...  
## $ smoothness\_worst : num 0.162 0.124 0.144 0.21 0.137 ...  
## $ compactness\_worst : num 0.666 0.187 0.424 0.866 0.205 ...  
## $ concavity\_worst : num 0.712 0.242 0.45 0.687 0.4 ...  
## $ concave.points\_worst : num 0.265 0.186 0.243 0.258 0.163 ...  
## $ symmetry\_worst : num 0.46 0.275 0.361 0.664 0.236 ...  
## $ fractal\_dimension\_worst: num 0.1189 0.089 0.0876 0.173 0.0768 ...  
## $ X : logi NA NA NA NA NA NA ...

# Remove id and X columns since they are not useful and have no values  
df$id <- NULL  
df$X <- NULL  
# Sort rows alphabetically and put target at the end  
df <- df[, order(names(df))]  
df <- df[, c(1:12, 14:31, 13)]

Data exploration

Looking at the data, there are more cases of benign than malignant breast mass, so upsampling could be useful for the models. Looking at the boxplots of the scaled data shows that most of the attributes are skewed upwards and have outliers in the upper tail, and the area and radius have less variation, especially compared to concavity and smoothness.

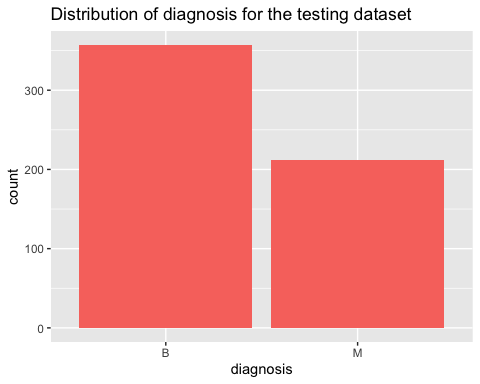
By looking at the plots, it shows that the variables split by malignant and benign reveal the two features of symmetry and fractal dimension are not useful at separating the classes. The features like area, concavity, and radius all separate the data better.

By plotting the correlation between the variables, it shows that there is a high correlation between some variables since there is the mean, standard error, and worst case of the values, which should all be correlated since they all measure the spread of the same attributes. Principal Component Analysis would be a good way to reduce complexity and deal with the highly correlated variables.

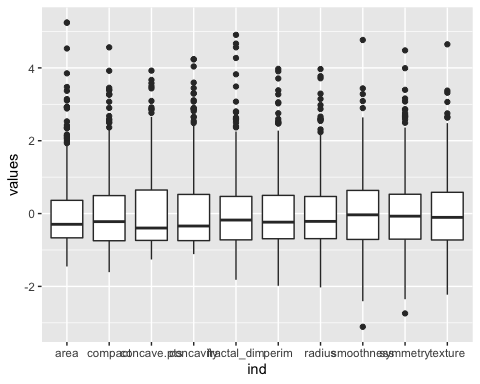
# Show summary of data  
summary(df)

## area\_mean area\_se area\_worst compactness\_mean   
## Min. : 143.5 Min. : 6.802 Min. : 185.2 Min. :0.01938   
## 1st Qu.: 420.3 1st Qu.: 17.850 1st Qu.: 515.3 1st Qu.:0.06492   
## Median : 551.1 Median : 24.530 Median : 686.5 Median :0.09263   
## Mean : 654.9 Mean : 40.337 Mean : 880.6 Mean :0.10434   
## 3rd Qu.: 782.7 3rd Qu.: 45.190 3rd Qu.:1084.0 3rd Qu.:0.13040   
## Max. :2501.0 Max. :542.200 Max. :4254.0 Max. :0.34540   
## compactness\_se compactness\_worst concave.points\_mean concave.points\_se   
## Min. :0.002252 Min. :0.02729 Min. :0.00000 Min. :0.000000   
## 1st Qu.:0.013080 1st Qu.:0.14720 1st Qu.:0.02031 1st Qu.:0.007638   
## Median :0.020450 Median :0.21190 Median :0.03350 Median :0.010930   
## Mean :0.025478 Mean :0.25427 Mean :0.04892 Mean :0.011796   
## 3rd Qu.:0.032450 3rd Qu.:0.33910 3rd Qu.:0.07400 3rd Qu.:0.014710   
## Max. :0.135400 Max. :1.05800 Max. :0.20120 Max. :0.052790   
## concave.points\_worst concavity\_mean concavity\_se concavity\_worst   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.06493 1st Qu.:0.02956 1st Qu.:0.01509 1st Qu.:0.1145   
## Median :0.09993 Median :0.06154 Median :0.02589 Median :0.2267   
## Mean :0.11461 Mean :0.08880 Mean :0.03189 Mean :0.2722   
## 3rd Qu.:0.16140 3rd Qu.:0.13070 3rd Qu.:0.04205 3rd Qu.:0.3829   
## Max. :0.29100 Max. :0.42680 Max. :0.39600 Max. :1.2520   
## fractal\_dimension\_mean fractal\_dimension\_se fractal\_dimension\_worst  
## Min. :0.04996 Min. :0.0008948 Min. :0.05504   
## 1st Qu.:0.05770 1st Qu.:0.0022480 1st Qu.:0.07146   
## Median :0.06154 Median :0.0031870 Median :0.08004   
## Mean :0.06280 Mean :0.0037949 Mean :0.08395   
## 3rd Qu.:0.06612 3rd Qu.:0.0045580 3rd Qu.:0.09208   
## Max. :0.09744 Max. :0.0298400 Max. :0.20750   
## perimeter\_mean perimeter\_se perimeter\_worst radius\_mean   
## Min. : 43.79 Min. : 0.757 Min. : 50.41 Min. : 6.981   
## 1st Qu.: 75.17 1st Qu.: 1.606 1st Qu.: 84.11 1st Qu.:11.700   
## Median : 86.24 Median : 2.287 Median : 97.66 Median :13.370   
## Mean : 91.97 Mean : 2.866 Mean :107.26 Mean :14.127   
## 3rd Qu.:104.10 3rd Qu.: 3.357 3rd Qu.:125.40 3rd Qu.:15.780   
## Max. :188.50 Max. :21.980 Max. :251.20 Max. :28.110   
## radius\_se radius\_worst smoothness\_mean smoothness\_se   
## Min. :0.1115 Min. : 7.93 Min. :0.05263 Min. :0.001713   
## 1st Qu.:0.2324 1st Qu.:13.01 1st Qu.:0.08637 1st Qu.:0.005169   
## Median :0.3242 Median :14.97 Median :0.09587 Median :0.006380   
## Mean :0.4052 Mean :16.27 Mean :0.09636 Mean :0.007041   
## 3rd Qu.:0.4789 3rd Qu.:18.79 3rd Qu.:0.10530 3rd Qu.:0.008146   
## Max. :2.8730 Max. :36.04 Max. :0.16340 Max. :0.031130   
## smoothness\_worst symmetry\_mean symmetry\_se symmetry\_worst   
## Min. :0.07117 Min. :0.1060 Min. :0.007882 Min. :0.1565   
## 1st Qu.:0.11660 1st Qu.:0.1619 1st Qu.:0.015160 1st Qu.:0.2504   
## Median :0.13130 Median :0.1792 Median :0.018730 Median :0.2822   
## Mean :0.13237 Mean :0.1812 Mean :0.020542 Mean :0.2901   
## 3rd Qu.:0.14600 3rd Qu.:0.1957 3rd Qu.:0.023480 3rd Qu.:0.3179   
## Max. :0.22260 Max. :0.3040 Max. :0.078950 Max. :0.6638   
## texture\_mean texture\_se texture\_worst diagnosis  
## Min. : 9.71 Min. :0.3602 Min. :12.02 B:357   
## 1st Qu.:16.17 1st Qu.:0.8339 1st Qu.:21.08 M:212   
## Median :18.84 Median :1.1080 Median :25.41   
## Mean :19.29 Mean :1.2169 Mean :25.68   
## 3rd Qu.:21.80 3rd Qu.:1.4740 3rd Qu.:29.72   
## Max. :39.28 Max. :4.8850 Max. :49.54

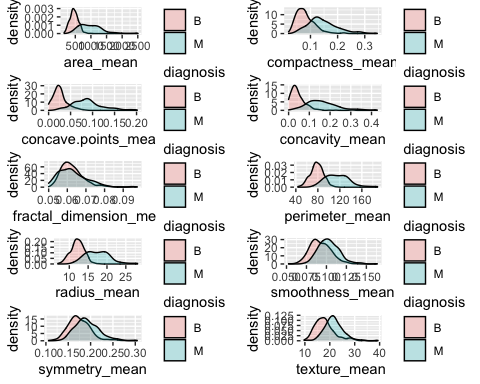
ggplot(df, aes(x = diagnosis)) +   
 geom\_bar(aes(fill = 'pink')) +   
 ggtitle("Distribution of diagnosis for the testing dataset") +   
 theme(legend.position="none")



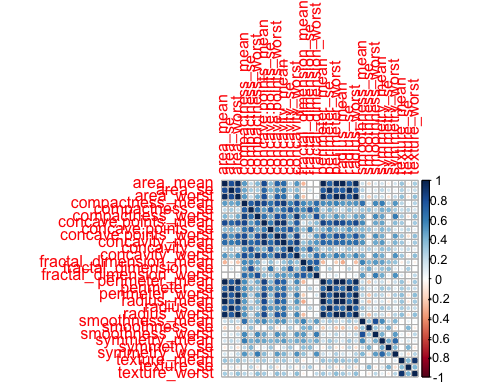
# Boxplot of scaled values  
df\_box <- df[,c("area\_mean", "compactness\_mean", "concave.points\_mean", "concavity\_mean", "fractal\_dimension\_mean", "perimeter\_mean", "radius\_mean", "smoothness\_mean", "symmetry\_mean", "texture\_mean")]  
colnames(df\_box) <- c("area", "compact", "concave.pts", "concavity", "fractal\_dim", "perim", "radius", "smoothness", "symmetry", "texture")  
ggplot(stack(data.frame(scale(df\_box))), aes(x = ind, y = values)) +  
 geom\_boxplot()



# Plot differences between malignant and benign  
g1 <- ggplot(df, aes(x=area\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g2 <- ggplot(df, aes(x=compactness\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)   
g3 <- ggplot(df, aes(x=concave.points\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g4 <- ggplot(df, aes(x=concavity\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g5 <- ggplot(df, aes(x=fractal\_dimension\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g6 <- ggplot(df, aes(x=perimeter\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g7 <- ggplot(df, aes(x=radius\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g8 <- ggplot(df, aes(x=smoothness\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g9 <- ggplot(df, aes(x=symmetry\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
g10 <- ggplot(df, aes(x=texture\_mean, fill=diagnosis)) + geom\_density(alpha=0.25)  
grid.arrange(g1,g2,g3,g4,g5,g6,g7,g8,g9,g10, ncol=2)



# Plot correlation matrix to see al correlating values  
corrplot(cor(df[,-31]), method="circle",)

 METHOD

When building the models, malignant is assigned positive and benign is assigned negative.

Data is randomly split in a 70/30 train and test split.

12 different models are used with10-fold cross validation on the training data set. The individual models are then compared by their performance.

The 12 models:

Tree-Based Models: “C5.0”, “rf”, “rpart” Boosting Models: “xgbTree”, “xgbLinear”, “gbm” Clustering Models: “knn” Linear Models: “glm”, “glmnet” Non-linear Models: “nb”, “svmPoly”, “nnet”

Pre-processing methods for all twelve models:

There are less cases of malignant than benign so upsampling is used. The data was centered and scaled to make sure the algorithms are performed correctly. The use of nzv, rorr and pca is to handle all highly correlated variables in the data set as shown in the above correlation matrix.

ROC is optimized for these models to show a two-class summary report that shows sensitivity. Sensitivity is important because it has to do with cancer, an extremely small cell that is malignant can lead to the potential of the cancer spreading and then death. Maximizing recall over precision is very important.

MODEL PERFORMANCE:

rpart performed the worst in all three metrics of ROC, sensitivity, and specificity  
Naiive bayes had poor sensitivity  
Tree-based models had high variability in recall The svm models had more variability in specificity. This is one of the models that was a best performer of the 12 different models. Linear models had more variability in specificity Neural network showed a good representations of the best performers as well.

The correlation between the models is compared and will be used for the stack ensemble approach.

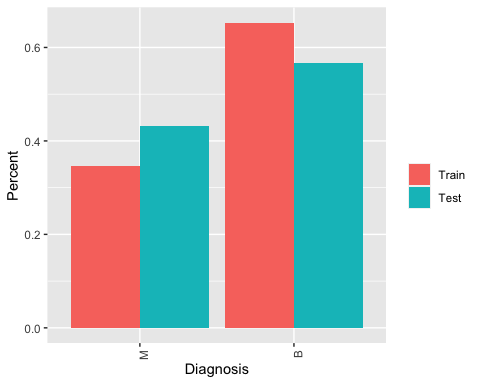
Next compared the correlation between the models and decided not to remove any because the stacked ensemble model blends the strengths of each of these models as meta-models.

PREDICTIVE MODEL PREPARATION

# Split data into train set and test set  
  
index <- sample(1:nrow(df), 0.7 \* nrow(df))  
train <- df[index,]  
test <- df[-index,]  
  
# This makes sure that malignant is positive and benign is negative  
  
df$diagnosis <- factor(df$diagnosis, levels = c("M", "B"))  
train$diagnosis <- factor(train$diagnosis, levels = c("M", "B"))  
test$diagnosis <- factor(test$diagnosis, levels = c("M", "B"))  
  
# Compare distribution of train and test data  
  
DistributionCompare <- cbind(prop.table(table(train$diagnosis)), prop.table(table(test$diagnosis)))  
colnames(DistributionCompare) <- c("Train", "Test")  
meltedDComp <- melt(DistributionCompare)

## Warning in melt(DistributionCompare): The melt generic in data.table has  
## been passed a matrix and will attempt to redirect to the relevant reshape2  
## method; please note that reshape2 is deprecated, and this redirection is now  
## deprecated as well. To continue using melt methods from reshape2 while both  
## libraries are attached, e.g. melt.list, you can prepend the namespace like  
## reshape2::melt(DistributionCompare). In the next version, this warning will  
## become an error.

ggplot(meltedDComp, aes(x=Var1, y=value)) + geom\_bar(aes(fill=Var2), stat = "identity", position = "dodge") + theme(axis.text.x = element\_text(angle = 90, hjust = 1)) + xlab("Diagnosis") + ylab("Percent") + labs(fill="")

 TRAINING TWELVE MODELS 10-fold CV, center, scale, pca, upsampling and compare performance

econtrol <- trainControl(method="cv", number=10, summaryFunction = twoClassSummary,   
 savePredictions = TRUE, classProbs = TRUE, sampling = "up")  
models <- caretList(diagnosis ~., data=train,  
 methodList=c("svmPoly", "nnet", "gbm", "xgbTree", "knn", "glm", "rf", "C5.0",   
 "nb", "rpart", "xgbLinear", "glmnet"),  
 preProcess = c("center", "scale", "nzv","corr", "pca"),  
 trControl = econtrol,  
 metric = "ROC")

## Warning in trControlCheck(x = trControl, y = target): x$savePredictions == TRUE  
## is depreciated. Setting to 'final' instead.

## Warning in trControlCheck(x = trControl, y = target): indexes not defined in  
## trControl. Attempting to set them ourselves, so each model in the ensemble will  
## have the same resampling indexes.

## # weights: 13  
## initial value 342.403715   
## iter 10 value 35.199576  
## iter 20 value 24.773989  
## iter 30 value 20.510256  
## iter 40 value 11.852474  
## iter 50 value 11.618472  
## iter 60 value 11.602609  
## iter 70 value 11.599431  
## iter 80 value 11.590293  
## iter 90 value 11.550992  
## iter 100 value 11.547611  
## final value 11.547611   
## stopped after 100 iterations  
## # weights: 37  
## initial value 367.410915   
## iter 10 value 27.108253  
## iter 20 value 14.230704  
## iter 30 value 7.753317  
## iter 40 value 3.927643  
## iter 50 value 2.977769  
## iter 60 value 2.818505  
## iter 70 value 2.779599  
## iter 80 value 2.773975  
## iter 90 value 2.772936  
## iter 100 value 2.772853  
## final value 2.772853   
## stopped after 100 iterations  
## # weights: 61  
## initial value 297.652704   
## iter 10 value 33.825904  
## iter 20 value 15.635508  
## iter 30 value 13.964666  
## iter 40 value 13.568083  
## iter 50 value 11.743124  
## iter 60 value 11.614586  
## iter 70 value 11.553218  
## iter 80 value 11.461862  
## iter 90 value 11.255333  
## iter 100 value 10.139586  
## final value 10.139586   
## stopped after 100 iterations  
## # weights: 13  
## initial value 326.989923   
## iter 10 value 59.643985  
## iter 20 value 38.642544  
## iter 30 value 37.504909  
## final value 37.504114   
## converged  
## # weights: 37  
## initial value 300.965181   
## iter 10 value 27.868553  
## iter 20 value 24.561407  
## iter 30 value 23.036765  
## iter 40 value 22.876486  
## iter 50 value 22.832148  
## final value 22.832110   
## converged  
## # weights: 61  
## initial value 394.336086   
## iter 10 value 35.350333  
## iter 20 value 27.830579  
## iter 30 value 26.559957  
## iter 40 value 26.301538  
## iter 50 value 26.276079  
## iter 60 value 26.264754  
## iter 70 value 26.261852  
## iter 80 value 26.261588  
## final value 26.261584   
## converged  
## # weights: 13  
## initial value 334.259428   
## iter 10 value 110.852937  
## iter 20 value 26.419174  
## iter 30 value 19.934314  
## iter 40 value 16.486270  
## iter 50 value 16.472540  
## iter 60 value 16.471774  
## iter 70 value 16.470334  
## iter 80 value 16.469246  
## iter 90 value 16.469032  
## iter 100 value 16.468728  
## final value 16.468728   
## stopped after 100 iterations  
## # weights: 37  
## initial value 337.299576   
## iter 10 value 21.007415  
## iter 20 value 14.164997  
## iter 30 value 12.165206  
## iter 40 value 11.807977  
## iter 50 value 11.761202  
## iter 60 value 11.698096  
## iter 70 value 11.344734  
## iter 80 value 10.736926  
## iter 90 value 7.913369  
## iter 100 value 6.672180  
## final value 6.672180   
## stopped after 100 iterations  
## # weights: 61  
## initial value 300.251132   
## iter 10 value 14.097951  
## iter 20 value 5.238763  
## iter 30 value 1.006203  
## iter 40 value 0.293584  
## iter 50 value 0.250886  
## iter 60 value 0.233334  
## iter 70 value 0.208388  
## iter 80 value 0.194578  
## iter 90 value 0.181532  
## iter 100 value 0.173437  
## final value 0.173437   
## stopped after 100 iterations  
## # weights: 13  
## initial value 317.811799   
## iter 10 value 45.120709  
## iter 20 value 25.108539  
## iter 30 value 21.795989  
## iter 40 value 20.392024  
## iter 50 value 20.310259  
## final value 20.310122   
## converged  
## # weights: 37  
## initial value 366.953874   
## iter 10 value 34.470331  
## iter 20 value 27.978501  
## iter 30 value 27.660467  
## iter 40 value 27.398116  
## iter 50 value 27.203477  
## iter 60 value 26.899821  
## iter 70 value 24.621309  
## iter 80 value 23.260465  
## iter 90 value 17.891157  
## iter 100 value 15.604241  
## final value 15.604241   
## stopped after 100 iterations  
## # weights: 61  
## initial value 296.323425   
## iter 10 value 16.456351  
## iter 20 value 4.186873  
## iter 30 value 0.114244  
## iter 40 value 0.003114  
## final value 0.000076   
## converged  
## # weights: 13  
## initial value 388.716516   
## iter 10 value 122.687751  
## iter 20 value 44.541627  
## iter 30 value 35.725191  
## final value 35.704397   
## converged  
## # weights: 37  
## initial value 341.794333   
## iter 10 value 46.972842  
## iter 20 value 35.280450  
## iter 30 value 33.517041  
## iter 40 value 31.227660  
## iter 50 value 30.992030  
## iter 60 value 30.948366  
## iter 70 value 30.897074  
## final value 30.896966   
## converged  
## # weights: 61  
## initial value 416.770819   
## iter 10 value 38.135327  
## iter 20 value 31.665332  
## iter 30 value 30.740272  
## iter 40 value 30.512121  
## iter 50 value 30.410288  
## iter 60 value 30.361494  
## iter 70 value 30.356359  
## iter 80 value 30.356138  
## final value 30.356122   
## converged  
## # weights: 13  
## initial value 320.278158   
## iter 10 value 39.100835  
## iter 20 value 32.755367  
## iter 30 value 24.149882  
## iter 40 value 21.736282  
## iter 50 value 20.844437  
## iter 60 value 20.825746  
## iter 70 value 20.814991  
## iter 80 value 20.809839  
## iter 90 value 20.806941  
## iter 100 value 20.804690  
## final value 20.804690   
## stopped after 100 iterations  
## # weights: 37  
## initial value 303.791367   
## iter 10 value 24.491521  
## iter 20 value 14.407142  
## iter 30 value 10.149477  
## iter 40 value 9.654599  
## iter 50 value 9.079426  
## iter 60 value 8.506011  
## iter 70 value 8.104744  
## iter 80 value 7.853569  
## iter 90 value 7.383266  
## iter 100 value 7.258836  
## final value 7.258836   
## stopped after 100 iterations  
## # weights: 61  
## initial value 356.194191   
## iter 10 value 31.413165  
## iter 20 value 21.602043  
## iter 30 value 17.698587  
## iter 40 value 14.424595  
## iter 50 value 7.897792  
## iter 60 value 5.167189  
## iter 70 value 4.733337  
## iter 80 value 4.262813  
## iter 90 value 0.529273  
## iter 100 value 0.424352  
## final value 0.424352   
## stopped after 100 iterations  
## # weights: 13  
## initial value 356.882464   
## iter 10 value 31.253069  
## iter 20 value 23.030347  
## iter 30 value 20.416608  
## iter 40 value 20.375855  
## iter 50 value 20.367338  
## iter 60 value 20.356196  
## iter 70 value 20.353059  
## iter 80 value 20.343726  
## iter 90 value 20.329811  
## iter 100 value 20.326013  
## final value 20.326013   
## stopped after 100 iterations  
## # weights: 37  
## initial value 309.897621   
## iter 10 value 18.674275  
## iter 20 value 6.749468  
## iter 30 value 4.146861  
## iter 40 value 4.113171  
## iter 50 value 4.067947  
## iter 50 value 4.067947  
## final value 4.067935   
## converged  
## # weights: 61  
## initial value 432.807977   
## iter 10 value 18.415756  
## iter 20 value 9.389162  
## iter 30 value 1.452689  
## iter 40 value 0.217308  
## iter 50 value 0.013328  
## final value 0.000067   
## converged  
## # weights: 13  
## initial value 319.908916   
## iter 10 value 41.687908  
## iter 20 value 33.397558  
## iter 30 value 32.365373  
## iter 40 value 32.348717  
## iter 40 value 32.348717  
## iter 40 value 32.348717  
## final value 32.348717   
## converged  
## # weights: 37  
## initial value 345.888318   
## iter 10 value 58.011085  
## iter 20 value 44.888459  
## iter 30 value 38.474760  
## iter 40 value 35.941934  
## iter 50 value 34.673179  
## iter 60 value 34.490704  
## iter 70 value 34.483303  
## final value 34.483262   
## converged  
## # weights: 61  
## initial value 397.360416   
## iter 10 value 49.752012  
## iter 20 value 30.499415  
## iter 30 value 27.564545  
## iter 40 value 26.023344  
## iter 50 value 25.065335  
## iter 60 value 23.988791  
## iter 70 value 23.523195  
## iter 80 value 23.337995  
## iter 90 value 23.320598  
## iter 100 value 23.318401  
## final value 23.318401   
## stopped after 100 iterations  
## # weights: 13  
## initial value 292.791647   
## iter 10 value 32.487940  
## iter 20 value 24.843315  
## iter 30 value 24.495589  
## iter 40 value 24.414179  
## iter 50 value 24.400888  
## iter 60 value 24.398642  
## iter 70 value 24.397382  
## iter 80 value 24.396200  
## iter 90 value 24.396069  
## iter 100 value 24.395995  
## final value 24.395995   
## stopped after 100 iterations  
## # weights: 37  
## initial value 306.620502   
## iter 10 value 22.838185  
## iter 20 value 18.376402  
## iter 30 value 11.941482  
## iter 40 value 3.444939  
## iter 50 value 1.311114  
## iter 60 value 0.905019  
## iter 70 value 0.836681  
## iter 80 value 0.780659  
## iter 90 value 0.723419  
## iter 100 value 0.671309  
## final value 0.671309   
## stopped after 100 iterations  
## # weights: 61  
## initial value 390.543534   
## iter 10 value 24.927862  
## iter 20 value 9.237354  
## iter 30 value 4.888252  
## iter 40 value 3.952242  
## iter 50 value 3.685743  
## iter 60 value 3.440218  
## iter 70 value 2.794566  
## iter 80 value 2.172402  
## iter 90 value 1.979201  
## iter 100 value 1.920331  
## final value 1.920331   
## stopped after 100 iterations  
## # weights: 13  
## initial value 366.471035   
## iter 10 value 32.163480  
## iter 20 value 23.355572  
## iter 30 value 20.510482  
## iter 40 value 20.452848  
## iter 50 value 20.393269  
## iter 60 value 20.350966  
## iter 70 value 20.323139  
## iter 80 value 20.318022  
## iter 90 value 20.317231  
## iter 100 value 20.316739  
## final value 20.316739   
## stopped after 100 iterations  
## # weights: 37  
## initial value 327.693881   
## iter 10 value 22.792857  
## iter 20 value 8.286234  
## iter 30 value 5.228975  
## iter 40 value 4.850357  
## iter 50 value 4.517527  
## iter 60 value 4.278137  
## iter 70 value 4.198222  
## iter 80 value 4.197233  
## iter 90 value 4.194492  
## iter 100 value 4.179883  
## final value 4.179883   
## stopped after 100 iterations  
## # weights: 61  
## initial value 338.919461   
## iter 10 value 36.668726  
## iter 20 value 14.342767  
## iter 30 value 7.906206  
## iter 40 value 5.978603  
## iter 50 value 5.592530  
## iter 60 value 5.409030  
## iter 70 value 4.188718  
## iter 80 value 0.166364  
## iter 90 value 0.038531  
## iter 100 value 0.021760  
## final value 0.021760   
## stopped after 100 iterations  
## # weights: 13  
## initial value 331.756804   
## iter 10 value 56.010153  
## iter 20 value 44.875893  
## iter 30 value 42.213595  
## iter 40 value 41.883315  
## final value 41.878223   
## converged  
## # weights: 37  
## initial value 351.113231   
## iter 10 value 37.338583  
## iter 20 value 30.766908  
## iter 30 value 30.025190  
## iter 40 value 29.706489  
## iter 50 value 29.444952  
## iter 60 value 29.429890  
## final value 29.429663   
## converged  
## # weights: 61  
## initial value 327.072746   
## iter 10 value 47.044778  
## iter 20 value 35.524174  
## iter 30 value 29.613501  
## iter 40 value 27.383892  
## iter 50 value 27.162069  
## iter 60 value 27.097821  
## iter 70 value 27.032129  
## iter 80 value 26.762524  
## iter 90 value 26.737737  
## iter 100 value 26.717253  
## final value 26.717253   
## stopped after 100 iterations  
## # weights: 13  
## initial value 336.491291   
## iter 10 value 79.811540  
## iter 20 value 36.602843  
## iter 30 value 31.120645  
## iter 40 value 28.279715  
## iter 50 value 19.935478  
## iter 60 value 12.998974  
## iter 70 value 11.964479  
## iter 80 value 11.838017  
## iter 90 value 11.828893  
## iter 100 value 11.826200  
## final value 11.826200   
## stopped after 100 iterations  
## # weights: 37  
## initial value 331.478873   
## iter 10 value 23.515819  
## iter 20 value 13.333962  
## iter 30 value 11.440860  
## iter 40 value 10.591944  
## iter 50 value 9.576344  
## iter 60 value 9.050998  
## iter 70 value 8.893325  
## iter 80 value 7.827916  
## iter 90 value 6.505996  
## iter 100 value 5.679315  
## final value 5.679315   
## stopped after 100 iterations  
## # weights: 61  
## initial value 309.526603   
## iter 10 value 24.824534  
## iter 20 value 6.876222  
## iter 30 value 0.721769  
## iter 40 value 0.364831  
## iter 50 value 0.338107  
## iter 60 value 0.306055  
## iter 70 value 0.280931  
## iter 80 value 0.263023  
## iter 90 value 0.251218  
## iter 100 value 0.245296  
## final value 0.245296   
## stopped after 100 iterations  
## # weights: 13  
## initial value 359.205447   
## iter 10 value 97.809183  
## iter 20 value 35.757603  
## iter 30 value 24.822389  
## iter 40 value 24.300698  
## iter 50 value 24.293610  
## iter 60 value 24.292781  
## iter 70 value 24.291861  
## iter 80 value 24.291375  
## iter 90 value 24.290317  
## iter 100 value 24.289799  
## final value 24.289799   
## stopped after 100 iterations  
## # weights: 37  
## initial value 314.822660   
## iter 10 value 23.666136  
## iter 20 value 12.258598  
## iter 30 value 10.270669  
## iter 40 value 9.583576  
## iter 50 value 9.136972  
## iter 60 value 8.318010  
## iter 70 value 8.312768  
## iter 80 value 8.304410  
## iter 90 value 6.807477  
## iter 100 value 6.738726  
## final value 6.738726   
## stopped after 100 iterations  
## # weights: 61  
## initial value 276.923581   
## iter 10 value 14.262418  
## iter 20 value 9.052328  
## iter 30 value 2.355768  
## iter 40 value 0.053027  
## iter 50 value 0.004051  
## iter 60 value 0.000312  
## iter 70 value 0.000199  
## final value 0.000072   
## converged  
## # weights: 13  
## initial value 332.517724   
## iter 10 value 64.983863  
## iter 20 value 46.371951  
## iter 30 value 46.355637  
## iter 30 value 46.355637  
## iter 30 value 46.355637  
## final value 46.355637   
## converged  
## # weights: 37  
## initial value 337.272370   
## iter 10 value 71.981921  
## iter 20 value 29.619779  
## iter 30 value 26.865311  
## iter 40 value 26.515082  
## iter 50 value 26.403878  
## iter 60 value 26.379523  
## iter 70 value 26.378198  
## iter 80 value 26.378050  
## iter 90 value 26.377994  
## final value 26.377993   
## converged  
## # weights: 61  
## initial value 320.581805   
## iter 10 value 50.486480  
## iter 20 value 27.768845  
## iter 30 value 24.679181  
## iter 40 value 24.002966  
## iter 50 value 23.838725  
## iter 60 value 23.095960  
## iter 70 value 22.868822  
## iter 80 value 22.841889  
## iter 90 value 22.836520  
## final value 22.836453   
## converged  
## # weights: 13  
## initial value 358.045834   
## iter 10 value 65.233489  
## iter 20 value 34.878805  
## iter 30 value 21.440166  
## iter 40 value 20.528924  
## iter 50 value 20.452083  
## iter 60 value 20.448184  
## iter 70 value 20.444586  
## iter 80 value 20.442896  
## iter 90 value 20.441746  
## iter 100 value 20.440547  
## final value 20.440547   
## stopped after 100 iterations  
## # weights: 37  
## initial value 329.848939   
## iter 10 value 23.339738  
## iter 20 value 7.929827  
## iter 30 value 4.105292  
## iter 40 value 2.806659  
## iter 50 value 2.749087  
## iter 60 value 2.732169  
## iter 70 value 2.724011  
## iter 80 value 2.699773  
## iter 90 value 2.335358  
## iter 100 value 2.106914  
## final value 2.106914   
## stopped after 100 iterations  
## # weights: 61  
## initial value 423.916173   
## iter 10 value 14.148994  
## iter 20 value 2.712923  
## iter 30 value 0.367938  
## iter 40 value 0.310020  
## iter 50 value 0.266870  
## iter 60 value 0.241693  
## iter 70 value 0.228604  
## iter 80 value 0.208135  
## iter 90 value 0.200336  
## iter 100 value 0.194198  
## final value 0.194198   
## stopped after 100 iterations  
## # weights: 13  
## initial value 325.402006   
## iter 10 value 40.157445  
## iter 20 value 24.226924  
## iter 30 value 23.555507  
## iter 40 value 22.258827  
## iter 50 value 20.314201  
## final value 20.310234   
## converged  
## # weights: 37  
## initial value 324.828986   
## iter 10 value 16.432815  
## iter 20 value 7.573676  
## iter 30 value 4.496245  
## iter 40 value 4.200825  
## iter 50 value 4.188470  
## iter 60 value 4.187899  
## final value 4.187893   
## converged  
## # weights: 61  
## initial value 283.507853   
## iter 10 value 13.089213  
## iter 20 value 4.780344  
## iter 30 value 2.841973  
## iter 40 value 1.801195  
## iter 50 value 1.420224  
## iter 60 value 1.389616  
## iter 70 value 1.386346  
## iter 80 value 1.386313  
## final value 1.386306   
## converged  
## # weights: 13  
## initial value 350.132393   
## iter 10 value 50.564262  
## iter 20 value 38.982330  
## iter 30 value 36.192387  
## final value 36.192377   
## converged  
## # weights: 37  
## initial value 320.570240   
## iter 10 value 79.363485  
## iter 20 value 34.940053  
## iter 30 value 26.380974  
## iter 40 value 24.916217  
## iter 50 value 24.601927  
## iter 60 value 24.549851  
## iter 70 value 24.548137  
## iter 80 value 24.547888  
## final value 24.547856   
## converged  
## # weights: 61  
## initial value 421.387738   
## iter 10 value 72.292403  
## iter 20 value 30.163336  
## iter 30 value 25.982232  
## iter 40 value 24.776688  
## iter 50 value 24.375657  
## iter 60 value 24.065797  
## iter 70 value 24.036433  
## iter 80 value 24.032153  
## iter 90 value 24.030657  
## iter 100 value 24.029931  
## final value 24.029931   
## stopped after 100 iterations  
## # weights: 13  
## initial value 363.405313   
## iter 10 value 19.498217  
## iter 20 value 12.224729  
## iter 30 value 11.847847  
## iter 40 value 11.844335  
## iter 50 value 11.843152  
## iter 60 value 11.841807  
## iter 70 value 11.841343  
## iter 80 value 11.840910  
## iter 90 value 11.840455  
## iter 100 value 11.840388  
## final value 11.840388   
## stopped after 100 iterations  
## # weights: 37  
## initial value 336.458197   
## iter 10 value 27.055162  
## iter 20 value 12.081380  
## iter 30 value 4.223898  
## iter 40 value 2.568632  
## iter 50 value 1.877439  
## iter 60 value 1.768356  
## iter 70 value 1.750248  
## iter 80 value 1.727691  
## iter 90 value 1.698969  
## iter 100 value 1.673258  
## final value 1.673258   
## stopped after 100 iterations  
## # weights: 61  
## initial value 375.128465   
## iter 10 value 15.956923  
## iter 20 value 6.152236  
## iter 30 value 0.455397  
## iter 40 value 0.331905  
## iter 50 value 0.288907  
## iter 60 value 0.261803  
## iter 70 value 0.242587  
## iter 80 value 0.226754  
## iter 90 value 0.219259  
## iter 100 value 0.212430  
## final value 0.212430   
## stopped after 100 iterations  
## # weights: 13  
## initial value 351.013860   
## iter 10 value 60.178877  
## iter 20 value 36.431812  
## iter 30 value 35.352690  
## iter 40 value 34.997019  
## iter 50 value 34.552211  
## iter 60 value 34.467051  
## iter 70 value 34.465302  
## iter 80 value 34.464924  
## final value 34.464705   
## converged  
## # weights: 37  
## initial value 322.874204   
## iter 10 value 27.049066  
## iter 20 value 13.844394  
## iter 30 value 10.135515  
## iter 40 value 9.834404  
## iter 50 value 9.815058  
## iter 60 value 9.809896  
## iter 70 value 9.809640  
## final value 9.809639   
## converged  
## # weights: 61  
## initial value 296.012687   
## iter 10 value 14.139560  
## iter 20 value 6.606238  
## iter 30 value 3.833619  
## iter 40 value 0.282080  
## iter 50 value 0.018885  
## iter 60 value 0.004284  
## iter 70 value 0.001409  
## iter 80 value 0.001027  
## iter 90 value 0.000582  
## iter 100 value 0.000318  
## final value 0.000318   
## stopped after 100 iterations  
## # weights: 13  
## initial value 351.390284   
## iter 10 value 45.409254  
## iter 20 value 37.106911  
## iter 30 value 33.350612  
## final value 33.350408   
## converged  
## # weights: 37  
## initial value 300.606300   
## iter 10 value 79.394255  
## iter 20 value 37.697159  
## iter 30 value 28.746290  
## iter 40 value 27.365645  
## iter 50 value 27.057364  
## iter 60 value 27.044461  
## final value 27.044199   
## converged  
## # weights: 61  
## initial value 349.258442   
## iter 10 value 65.557422  
## iter 20 value 31.163902  
## iter 30 value 26.073469  
## iter 40 value 25.104624  
## iter 50 value 24.618740  
## iter 60 value 24.509693  
## iter 70 value 24.508849  
## final value 24.508802   
## converged  
## # weights: 13  
## initial value 366.579982   
## iter 10 value 236.655055  
## iter 20 value 56.862079  
## iter 30 value 31.571028  
## iter 40 value 24.592234  
## iter 50 value 24.190964  
## iter 60 value 23.349243  
## iter 70 value 22.847655  
## iter 80 value 22.389673  
## iter 90 value 22.267372  
## iter 100 value 22.255117  
## final value 22.255117   
## stopped after 100 iterations  
## # weights: 37  
## initial value 322.598792   
## iter 10 value 27.195805  
## iter 20 value 13.905618  
## iter 30 value 6.585478  
## iter 40 value 5.126913  
## iter 50 value 4.948059  
## iter 60 value 4.904044  
## iter 70 value 4.890582  
## iter 80 value 4.886386  
## iter 90 value 4.869623  
## iter 100 value 4.772565  
## final value 4.772565   
## stopped after 100 iterations  
## # weights: 61  
## initial value 338.311983   
## iter 10 value 23.994532  
## iter 20 value 7.775227  
## iter 30 value 3.207626  
## iter 40 value 2.246278  
## iter 50 value 2.215575  
## iter 60 value 2.199987  
## iter 70 value 2.191780  
## iter 80 value 2.184668  
## iter 90 value 2.179597  
## iter 100 value 2.176160  
## final value 2.176160   
## stopped after 100 iterations  
## # weights: 13  
## initial value 328.367251   
## iter 10 value 29.724170  
## iter 20 value 19.384564  
## iter 30 value 17.305352  
## iter 40 value 16.575563  
## iter 50 value 16.542638  
## iter 60 value 16.502115  
## iter 70 value 16.353534  
## iter 80 value 16.345924  
## iter 90 value 16.328858  
## iter 100 value 16.266988  
## final value 16.266988   
## stopped after 100 iterations  
## # weights: 37  
## initial value 314.933848   
## iter 10 value 37.372221  
## iter 20 value 14.670467  
## iter 30 value 5.930486  
## iter 40 value 4.332273  
## iter 50 value 3.800209  
## iter 60 value 2.311203  
## iter 70 value 0.906686  
## iter 80 value 0.389321  
## iter 90 value 0.257288  
## iter 100 value 0.121538  
## final value 0.121538   
## stopped after 100 iterations  
## # weights: 61  
## initial value 358.111251   
## iter 10 value 24.177597  
## iter 20 value 7.029010  
## iter 30 value 2.388836  
## iter 40 value 1.817013  
## iter 50 value 1.413706  
## iter 60 value 1.398262  
## iter 70 value 1.350341  
## iter 80 value 0.040657  
## iter 90 value 0.023592  
## iter 100 value 0.015643  
## final value 0.015643   
## stopped after 100 iterations  
## # weights: 13  
## initial value 345.143491   
## iter 10 value 63.530788  
## iter 20 value 39.784995  
## iter 30 value 37.379928  
## final value 37.359847   
## converged  
## # weights: 37  
## initial value 338.760829   
## iter 10 value 25.975037  
## iter 20 value 24.213679  
## iter 30 value 23.176393  
## iter 40 value 23.077504  
## iter 50 value 23.042751  
## iter 60 value 23.041983  
## iter 70 value 23.041847  
## iter 70 value 23.041847  
## iter 70 value 23.041847  
## final value 23.041847   
## converged  
## # weights: 61  
## initial value 332.382331   
## iter 10 value 36.667448  
## iter 20 value 28.699938  
## iter 30 value 25.471257  
## iter 40 value 23.360416  
## iter 50 value 22.184705  
## iter 60 value 21.920652  
## iter 70 value 21.903142  
## iter 80 value 21.901652  
## final value 21.901645   
## converged  
## # weights: 13  
## initial value 329.647925   
## iter 10 value 22.932653  
## iter 20 value 20.644492  
## iter 30 value 20.519037  
## iter 40 value 20.496699  
## iter 50 value 20.491081  
## iter 60 value 20.488091  
## iter 70 value 20.485949  
## iter 80 value 20.484932  
## iter 90 value 20.484517  
## iter 100 value 20.484113  
## final value 20.484113   
## stopped after 100 iterations  
## # weights: 37  
## initial value 311.659226   
## iter 10 value 29.972693  
## iter 20 value 22.280964  
## iter 30 value 20.236700  
## iter 40 value 19.301552  
## iter 50 value 18.870026  
## iter 60 value 18.463590  
## iter 70 value 18.379288  
## iter 80 value 18.359682  
## iter 90 value 18.332723  
## iter 100 value 18.322589  
## final value 18.322589   
## stopped after 100 iterations  
## # weights: 61  
## initial value 379.648380   
## iter 10 value 14.559133  
## iter 20 value 8.970022  
## iter 30 value 6.965105  
## iter 40 value 4.134512  
## iter 50 value 3.767934  
## iter 60 value 2.343307  
## iter 70 value 2.261796  
## iter 80 value 2.219872  
## iter 90 value 1.797476  
## iter 100 value 1.721290  
## final value 1.721290   
## stopped after 100 iterations  
## # weights: 13  
## initial value 344.170729   
## iter 10 value 34.425364  
## iter 20 value 30.277100  
## iter 30 value 25.672485  
## iter 40 value 25.295770  
## iter 50 value 24.293143  
## iter 60 value 24.282715  
## final value 24.282700   
## converged  
## # weights: 37  
## initial value 372.452327   
## iter 10 value 36.040666  
## iter 20 value 14.877901  
## iter 30 value 3.784047  
## iter 40 value 0.033142  
## iter 50 value 0.000315  
## final value 0.000052   
## converged  
## # weights: 61  
## initial value 321.227185   
## iter 10 value 25.389185  
## iter 20 value 6.728883  
## iter 30 value 2.462478  
## iter 40 value 1.522655  
## iter 50 value 1.437334  
## iter 60 value 1.400264  
## iter 70 value 1.392250  
## iter 80 value 1.390868  
## iter 90 value 1.387255  
## iter 100 value 1.386917  
## final value 1.386917   
## stopped after 100 iterations  
## # weights: 13  
## initial value 345.542016   
## iter 10 value 47.143793  
## iter 20 value 41.372525  
## final value 41.340494   
## converged  
## # weights: 37  
## initial value 345.075278   
## iter 10 value 50.426150  
## iter 20 value 34.152026  
## iter 30 value 32.456783  
## iter 40 value 31.208408  
## iter 50 value 30.752629  
## iter 60 value 30.747141  
## iter 70 value 30.745958  
## final value 30.745956   
## converged  
## # weights: 61  
## initial value 350.786710   
## iter 10 value 53.122162  
## iter 20 value 33.910076  
## iter 30 value 27.609417  
## iter 40 value 25.089882  
## iter 50 value 24.709833  
## iter 60 value 24.622906  
## iter 70 value 24.619902  
## final value 24.619893   
## converged  
## # weights: 13  
## initial value 334.438398   
## iter 10 value 30.111844  
## iter 20 value 26.505572  
## iter 30 value 25.556059  
## iter 40 value 21.441288  
## iter 50 value 20.690985  
## iter 60 value 20.659208  
## iter 70 value 20.651792  
## iter 80 value 20.650177  
## iter 90 value 20.649288  
## iter 100 value 20.648936  
## final value 20.648936   
## stopped after 100 iterations  
## # weights: 37  
## initial value 297.184218   
## iter 10 value 36.170623  
## iter 20 value 29.004167  
## iter 30 value 28.057809  
## iter 40 value 27.472937  
## iter 50 value 27.385140  
## iter 60 value 27.336869  
## iter 70 value 27.317521  
## iter 80 value 27.304473  
## iter 90 value 27.291536  
## iter 100 value 27.281972  
## final value 27.281972   
## stopped after 100 iterations  
## # weights: 61  
## initial value 335.029308   
## iter 10 value 20.070682  
## iter 20 value 7.829686  
## iter 30 value 1.344078  
## iter 40 value 0.473487  
## iter 50 value 0.395712  
## iter 60 value 0.364631  
## iter 70 value 0.337813  
## iter 80 value 0.310545  
## iter 90 value 0.270106  
## iter 100 value 0.255386  
## final value 0.255386   
## stopped after 100 iterations  
## # weights: 13  
## initial value 346.221848   
## iter 10 value 30.373199  
## iter 20 value 20.455102  
## iter 30 value 14.704077  
## iter 40 value 12.947260  
## iter 50 value 11.540463  
## final value 11.532907   
## converged  
## # weights: 37  
## initial value 335.476015   
## iter 10 value 24.018100  
## iter 20 value 6.086399  
## iter 30 value 0.808613  
## iter 40 value 0.014051  
## iter 50 value 0.001317  
## final value 0.000092   
## converged  
## # weights: 61  
## initial value 357.832152   
## iter 10 value 24.161192  
## iter 20 value 17.483477  
## iter 30 value 13.514075  
## iter 40 value 13.033251  
## iter 50 value 12.959804  
## iter 60 value 12.954205  
## iter 70 value 12.953429  
## iter 80 value 12.952617  
## iter 90 value 12.951870  
## iter 100 value 12.943896  
## final value 12.943896   
## stopped after 100 iterations  
## # weights: 13  
## initial value 341.267082   
## iter 10 value 303.483936  
## iter 20 value 241.972336  
## iter 30 value 87.908924  
## iter 40 value 41.119824  
## iter 50 value 35.902599  
## iter 60 value 35.877921  
## final value 35.876588   
## converged  
## # weights: 37  
## initial value 347.971307   
## iter 10 value 36.022360  
## iter 20 value 29.547381  
## iter 30 value 28.633041  
## iter 40 value 28.377614  
## iter 50 value 28.366092  
## iter 60 value 28.359718  
## final value 28.359712   
## converged  
## # weights: 61  
## initial value 417.921440   
## iter 10 value 34.378510  
## iter 20 value 26.830402  
## iter 30 value 25.575014  
## iter 40 value 25.329444  
## iter 50 value 25.276214  
## iter 60 value 25.252778  
## iter 70 value 25.251276  
## final value 25.251275   
## converged  
## # weights: 13  
## initial value 358.189835   
## iter 10 value 32.283749  
## iter 20 value 17.165879  
## iter 30 value 12.756388  
## iter 40 value 12.016281  
## iter 50 value 11.807308  
## iter 60 value 11.800750  
## iter 70 value 11.798359  
## iter 80 value 11.796396  
## iter 90 value 11.795694  
## iter 100 value 11.794709  
## final value 11.794709   
## stopped after 100 iterations  
## # weights: 37  
## initial value 366.479861   
## iter 10 value 28.456330  
## iter 20 value 20.496418  
## iter 30 value 15.044849  
## iter 40 value 10.350132  
## iter 50 value 7.817596  
## iter 60 value 7.100493  
## iter 70 value 6.640634  
## iter 80 value 6.180825  
## iter 90 value 5.902330  
## iter 100 value 5.828123  
## final value 5.828123   
## stopped after 100 iterations  
## # weights: 61  
## initial value 322.036395   
## iter 10 value 26.395046  
## iter 20 value 5.930049  
## iter 30 value 0.427538  
## iter 40 value 0.373526  
## iter 50 value 0.338080  
## iter 60 value 0.320509  
## iter 70 value 0.308547  
## iter 80 value 0.293747  
## iter 90 value 0.284250  
## iter 100 value 0.269689  
## final value 0.269689   
## stopped after 100 iterations  
## # weights: 61  
## initial value 373.394157   
## iter 10 value 76.675932  
## iter 20 value 34.909270  
## iter 30 value 28.892246  
## iter 40 value 25.843308  
## iter 50 value 24.764867  
## iter 60 value 24.651577  
## iter 70 value 24.486064  
## iter 80 value 24.398555  
## iter 90 value 24.387508  
## final value 24.387479   
## converged  
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2599 nan 0.1000 0.0651  
## 2 1.1504 nan 0.1000 0.0529  
## 3 1.0619 nan 0.1000 0.0424  
## 4 0.9891 nan 0.1000 0.0360  
## 5 0.9293 nan 0.1000 0.0274  
## 6 0.8734 nan 0.1000 0.0277  
## 7 0.8229 nan 0.1000 0.0241  
## 8 0.7795 nan 0.1000 0.0204  
## 9 0.7419 nan 0.1000 0.0161  
## 10 0.7092 nan 0.1000 0.0147  
## 20 0.5408 nan 0.1000 0.0028  
## 40 0.3859 nan 0.1000 0.0023  
## 60 0.3008 nan 0.1000 0.0008  
## 80 0.2468 nan 0.1000 -0.0002  
## 100 0.2143 nan 0.1000 -0.0001  
## 120 0.1853 nan 0.1000 -0.0004  
## 140 0.1637 nan 0.1000 -0.0008  
## 150 0.1564 nan 0.1000 0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2521 nan 0.1000 0.0697  
## 2 1.1375 nan 0.1000 0.0545  
## 3 1.0417 nan 0.1000 0.0445  
## 4 0.9598 nan 0.1000 0.0384  
## 5 0.8883 nan 0.1000 0.0345  
## 6 0.8307 nan 0.1000 0.0280  
## 7 0.7781 nan 0.1000 0.0233  
## 8 0.7301 nan 0.1000 0.0230  
## 9 0.6894 nan 0.1000 0.0187  
## 10 0.6510 nan 0.1000 0.0180  
## 20 0.4136 nan 0.1000 0.0039  
## 40 0.2455 nan 0.1000 0.0009  
## 60 0.1726 nan 0.1000 0.0001  
## 80 0.1342 nan 0.1000 -0.0012  
## 100 0.1077 nan 0.1000 -0.0005  
## 120 0.0829 nan 0.1000 -0.0008  
## 140 0.0660 nan 0.1000 -0.0001  
## 150 0.0591 nan 0.1000 -0.0004  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2446 nan 0.1000 0.0668  
## 2 1.1226 nan 0.1000 0.0589  
## 3 1.0222 nan 0.1000 0.0473  
## 4 0.9278 nan 0.1000 0.0430  
## 5 0.8553 nan 0.1000 0.0321  
## 6 0.7921 nan 0.1000 0.0279  
## 7 0.7294 nan 0.1000 0.0299  
## 8 0.6804 nan 0.1000 0.0236  
## 9 0.6300 nan 0.1000 0.0235  
## 10 0.5899 nan 0.1000 0.0187  
## 20 0.3535 nan 0.1000 0.0052  
## 40 0.1926 nan 0.1000 -0.0004  
## 60 0.1187 nan 0.1000 -0.0001  
## 80 0.0851 nan 0.1000 -0.0001  
## 100 0.0608 nan 0.1000 0.0003  
## 120 0.0465 nan 0.1000 -0.0000  
## 140 0.0355 nan 0.1000 -0.0002  
## 150 0.0314 nan 0.1000 -0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2546 nan 0.1000 0.0637  
## 2 1.1529 nan 0.1000 0.0531  
## 3 1.0671 nan 0.1000 0.0418  
## 4 0.9895 nan 0.1000 0.0372  
## 5 0.9244 nan 0.1000 0.0313  
## 6 0.8715 nan 0.1000 0.0270  
## 7 0.8273 nan 0.1000 0.0220  
## 8 0.7885 nan 0.1000 0.0196  
## 9 0.7526 nan 0.1000 0.0166  
## 10 0.7194 nan 0.1000 0.0140  
## 20 0.5496 nan 0.1000 0.0046  
## 40 0.4039 nan 0.1000 0.0026  
## 60 0.3207 nan 0.1000 0.0007  
## 80 0.2631 nan 0.1000 0.0008  
## 100 0.2161 nan 0.1000 -0.0001  
## 120 0.1858 nan 0.1000 -0.0005  
## 140 0.1645 nan 0.1000 -0.0001  
## 150 0.1496 nan 0.1000 0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2466 nan 0.1000 0.0674  
## 2 1.1240 nan 0.1000 0.0587  
## 3 1.0236 nan 0.1000 0.0508  
## 4 0.9383 nan 0.1000 0.0416  
## 5 0.8663 nan 0.1000 0.0332  
## 6 0.8000 nan 0.1000 0.0318  
## 7 0.7484 nan 0.1000 0.0218  
## 8 0.6982 nan 0.1000 0.0225  
## 9 0.6561 nan 0.1000 0.0199  
## 10 0.6130 nan 0.1000 0.0202  
## 20 0.3867 nan 0.1000 0.0052  
## 40 0.2181 nan 0.1000 -0.0004  
## 60 0.1449 nan 0.1000 0.0004  
## 80 0.1051 nan 0.1000 0.0002  
## 100 0.0798 nan 0.1000 0.0003  
## 120 0.0644 nan 0.1000 0.0001  
## 140 0.0493 nan 0.1000 -0.0003  
## 150 0.0446 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2346 nan 0.1000 0.0718  
## 2 1.1125 nan 0.1000 0.0604  
## 3 1.0110 nan 0.1000 0.0466  
## 4 0.9242 nan 0.1000 0.0436  
## 5 0.8499 nan 0.1000 0.0359  
## 6 0.7776 nan 0.1000 0.0316  
## 7 0.7250 nan 0.1000 0.0244  
## 8 0.6772 nan 0.1000 0.0215  
## 9 0.6201 nan 0.1000 0.0284  
## 10 0.5764 nan 0.1000 0.0193  
## 20 0.3323 nan 0.1000 0.0015  
## 40 0.1619 nan 0.1000 0.0011  
## 60 0.0970 nan 0.1000 -0.0003  
## 80 0.0641 nan 0.1000 -0.0001  
## 100 0.0432 nan 0.1000 -0.0002  
## 120 0.0308 nan 0.1000 0.0000  
## 140 0.0226 nan 0.1000 0.0001  
## 150 0.0197 nan 0.1000 -0.0001  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2684 nan 0.1000 0.0535  
## 2 1.1675 nan 0.1000 0.0507  
## 3 1.0784 nan 0.1000 0.0406  
## 4 1.0045 nan 0.1000 0.0359  
## 5 0.9454 nan 0.1000 0.0299  
## 6 0.8921 nan 0.1000 0.0219  
## 7 0.8455 nan 0.1000 0.0221  
## 8 0.8035 nan 0.1000 0.0180  
## 9 0.7697 nan 0.1000 0.0163  
## 10 0.7420 nan 0.1000 0.0125  
## 20 0.5793 nan 0.1000 0.0053  
## 40 0.4209 nan 0.1000 0.0025  
## 60 0.3351 nan 0.1000 0.0007  
## 80 0.2783 nan 0.1000 0.0001  
## 100 0.2371 nan 0.1000 0.0004  
## 120 0.2039 nan 0.1000 0.0002  
## 140 0.1832 nan 0.1000 -0.0009  
## 150 0.1719 nan 0.1000 -0.0003  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2544 nan 0.1000 0.0667  
## 2 1.1500 nan 0.1000 0.0507  
## 3 1.0509 nan 0.1000 0.0471  
## 4 0.9709 nan 0.1000 0.0382  
## 5 0.9047 nan 0.1000 0.0301  
## 6 0.8463 nan 0.1000 0.0257  
## 7 0.7903 nan 0.1000 0.0235  
## 8 0.7485 nan 0.1000 0.0192  
## 9 0.7045 nan 0.1000 0.0204  
## 10 0.6647 nan 0.1000 0.0182  
## 20 0.4512 nan 0.1000 0.0016  
## 40 0.2643 nan 0.1000 0.0011  
## 60 0.1860 nan 0.1000 0.0004  
## 80 0.1373 nan 0.1000 0.0002  
## 100 0.1031 nan 0.1000 0.0001  
## 120 0.0790 nan 0.1000 -0.0006  
## 140 0.0607 nan 0.1000 -0.0001  
## 150 0.0546 nan 0.1000 0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2379 nan 0.1000 0.0732  
## 2 1.1259 nan 0.1000 0.0524  
## 3 1.0126 nan 0.1000 0.0508  
## 4 0.9310 nan 0.1000 0.0372  
## 5 0.8594 nan 0.1000 0.0336  
## 6 0.7947 nan 0.1000 0.0293  
## 7 0.7403 nan 0.1000 0.0250  
## 8 0.6832 nan 0.1000 0.0271  
## 9 0.6387 nan 0.1000 0.0189  
## 10 0.5938 nan 0.1000 0.0211  
## 20 0.3564 nan 0.1000 0.0081  
## 40 0.1807 nan 0.1000 0.0012  
## 60 0.1108 nan 0.1000 0.0010  
## 80 0.0779 nan 0.1000 -0.0001  
## 100 0.0538 nan 0.1000 -0.0003  
## 120 0.0397 nan 0.1000 0.0001  
## 140 0.0296 nan 0.1000 -0.0001  
## 150 0.0259 nan 0.1000 -0.0001  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2568 nan 0.1000 0.0628  
## 2 1.1505 nan 0.1000 0.0516  
## 3 1.0642 nan 0.1000 0.0442  
## 4 0.9887 nan 0.1000 0.0355  
## 5 0.9287 nan 0.1000 0.0323  
## 6 0.8750 nan 0.1000 0.0277  
## 7 0.8252 nan 0.1000 0.0232  
## 8 0.7848 nan 0.1000 0.0176  
## 9 0.7482 nan 0.1000 0.0175  
## 10 0.7123 nan 0.1000 0.0143  
## 20 0.5344 nan 0.1000 0.0051  
## 40 0.3921 nan 0.1000 0.0017  
## 60 0.3054 nan 0.1000 -0.0004  
## 80 0.2509 nan 0.1000 0.0005  
## 100 0.2153 nan 0.1000 -0.0005  
## 120 0.1835 nan 0.1000 0.0005  
## 140 0.1600 nan 0.1000 0.0003  
## 150 0.1508 nan 0.1000 -0.0009  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2535 nan 0.1000 0.0637  
## 2 1.1391 nan 0.1000 0.0534  
## 3 1.0442 nan 0.1000 0.0447  
## 4 0.9661 nan 0.1000 0.0365  
## 5 0.9014 nan 0.1000 0.0290  
## 6 0.8405 nan 0.1000 0.0263  
## 7 0.7869 nan 0.1000 0.0259  
## 8 0.7424 nan 0.1000 0.0209  
## 9 0.6986 nan 0.1000 0.0184  
## 10 0.6596 nan 0.1000 0.0171  
## 20 0.4369 nan 0.1000 0.0049  
## 40 0.2524 nan 0.1000 0.0006  
## 60 0.1757 nan 0.1000 0.0008  
## 80 0.1272 nan 0.1000 -0.0000  
## 100 0.0943 nan 0.1000 0.0002  
## 120 0.0758 nan 0.1000 -0.0001  
## 140 0.0578 nan 0.1000 -0.0005  
## 150 0.0521 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2332 nan 0.1000 0.0730  
## 2 1.1110 nan 0.1000 0.0560  
## 3 1.0209 nan 0.1000 0.0436  
## 4 0.9330 nan 0.1000 0.0426  
## 5 0.8604 nan 0.1000 0.0363  
## 6 0.7997 nan 0.1000 0.0295  
## 7 0.7423 nan 0.1000 0.0266  
## 8 0.6952 nan 0.1000 0.0219  
## 9 0.6530 nan 0.1000 0.0195  
## 10 0.6135 nan 0.1000 0.0193  
## 20 0.3682 nan 0.1000 0.0082  
## 40 0.1942 nan 0.1000 0.0006  
## 60 0.1218 nan 0.1000 0.0000  
## 80 0.0829 nan 0.1000 0.0003  
## 100 0.0578 nan 0.1000 0.0001  
## 120 0.0420 nan 0.1000 0.0000  
## 140 0.0311 nan 0.1000 0.0001  
## 150 0.0268 nan 0.1000 0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2576 nan 0.1000 0.0661  
## 2 1.1464 nan 0.1000 0.0539  
## 3 1.0580 nan 0.1000 0.0433  
## 4 0.9803 nan 0.1000 0.0372  
## 5 0.9147 nan 0.1000 0.0301  
## 6 0.8584 nan 0.1000 0.0273  
## 7 0.8095 nan 0.1000 0.0232  
## 8 0.7676 nan 0.1000 0.0187  
## 9 0.7329 nan 0.1000 0.0157  
## 10 0.7020 nan 0.1000 0.0151  
## 20 0.5248 nan 0.1000 0.0059  
## 40 0.3787 nan 0.1000 0.0019  
## 60 0.2979 nan 0.1000 -0.0005  
## 80 0.2416 nan 0.1000 0.0000  
## 100 0.2013 nan 0.1000 -0.0001  
## 120 0.1734 nan 0.1000 0.0004  
## 140 0.1497 nan 0.1000 0.0003  
## 150 0.1398 nan 0.1000 0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2491 nan 0.1000 0.0691  
## 2 1.1331 nan 0.1000 0.0547  
## 3 1.0421 nan 0.1000 0.0467  
## 4 0.9561 nan 0.1000 0.0404  
## 5 0.8888 nan 0.1000 0.0318  
## 6 0.8253 nan 0.1000 0.0312  
## 7 0.7699 nan 0.1000 0.0271  
## 8 0.7219 nan 0.1000 0.0210  
## 9 0.6797 nan 0.1000 0.0181  
## 10 0.6418 nan 0.1000 0.0173  
## 20 0.4026 nan 0.1000 0.0073  
## 40 0.2319 nan 0.1000 0.0005  
## 60 0.1562 nan 0.1000 -0.0010  
## 80 0.1164 nan 0.1000 0.0004  
## 100 0.0842 nan 0.1000 0.0001  
## 120 0.0680 nan 0.1000 -0.0002  
## 140 0.0559 nan 0.1000 -0.0000  
## 150 0.0496 nan 0.1000 0.0001  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2419 nan 0.1000 0.0686  
## 2 1.1119 nan 0.1000 0.0616  
## 3 1.0092 nan 0.1000 0.0516  
## 4 0.9199 nan 0.1000 0.0417  
## 5 0.8470 nan 0.1000 0.0337  
## 6 0.7875 nan 0.1000 0.0272  
## 7 0.7325 nan 0.1000 0.0261  
## 8 0.6800 nan 0.1000 0.0234  
## 9 0.6282 nan 0.1000 0.0233  
## 10 0.5932 nan 0.1000 0.0152  
## 20 0.3388 nan 0.1000 0.0075  
## 40 0.1731 nan 0.1000 0.0016  
## 60 0.1078 nan 0.1000 -0.0000  
## 80 0.0719 nan 0.1000 0.0003  
## 100 0.0508 nan 0.1000 -0.0001  
## 120 0.0374 nan 0.1000 0.0000  
## 140 0.0277 nan 0.1000 -0.0001  
## 150 0.0238 nan 0.1000 -0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2549 nan 0.1000 0.0646  
## 2 1.1460 nan 0.1000 0.0528  
## 3 1.0638 nan 0.1000 0.0423  
## 4 0.9888 nan 0.1000 0.0370  
## 5 0.9248 nan 0.1000 0.0295  
## 6 0.8740 nan 0.1000 0.0265  
## 7 0.8220 nan 0.1000 0.0229  
## 8 0.7811 nan 0.1000 0.0177  
## 9 0.7441 nan 0.1000 0.0169  
## 10 0.7096 nan 0.1000 0.0158  
## 20 0.5301 nan 0.1000 0.0035  
## 40 0.3734 nan 0.1000 0.0031  
## 60 0.2953 nan 0.1000 0.0002  
## 80 0.2361 nan 0.1000 0.0004  
## 100 0.1948 nan 0.1000 -0.0002  
## 120 0.1607 nan 0.1000 -0.0000  
## 140 0.1365 nan 0.1000 -0.0001  
## 150 0.1280 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2501 nan 0.1000 0.0676  
## 2 1.1381 nan 0.1000 0.0538  
## 3 1.0412 nan 0.1000 0.0458  
## 4 0.9605 nan 0.1000 0.0380  
## 5 0.8910 nan 0.1000 0.0333  
## 6 0.8314 nan 0.1000 0.0288  
## 7 0.7746 nan 0.1000 0.0237  
## 8 0.7351 nan 0.1000 0.0163  
## 9 0.6971 nan 0.1000 0.0157  
## 10 0.6556 nan 0.1000 0.0170  
## 20 0.4279 nan 0.1000 0.0049  
## 40 0.2496 nan 0.1000 -0.0002  
## 60 0.1586 nan 0.1000 0.0001  
## 80 0.1156 nan 0.1000 -0.0005  
## 100 0.0857 nan 0.1000 -0.0004  
## 120 0.0618 nan 0.1000 0.0000  
## 140 0.0481 nan 0.1000 -0.0000  
## 150 0.0430 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2421 nan 0.1000 0.0709  
## 2 1.1131 nan 0.1000 0.0637  
## 3 1.0194 nan 0.1000 0.0429  
## 4 0.9282 nan 0.1000 0.0425  
## 5 0.8581 nan 0.1000 0.0307  
## 6 0.7936 nan 0.1000 0.0293  
## 7 0.7354 nan 0.1000 0.0274  
## 8 0.6853 nan 0.1000 0.0229  
## 9 0.6346 nan 0.1000 0.0243  
## 10 0.5960 nan 0.1000 0.0177  
## 20 0.3451 nan 0.1000 0.0054  
## 40 0.1711 nan 0.1000 0.0006  
## 60 0.1083 nan 0.1000 0.0001  
## 80 0.0710 nan 0.1000 0.0003  
## 100 0.0492 nan 0.1000 -0.0002  
## 120 0.0360 nan 0.1000 -0.0001  
## 140 0.0266 nan 0.1000 -0.0001  
## 150 0.0231 nan 0.1000 0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2624 nan 0.1000 0.0626  
## 2 1.1591 nan 0.1000 0.0507  
## 3 1.0691 nan 0.1000 0.0436  
## 4 0.9907 nan 0.1000 0.0357  
## 5 0.9274 nan 0.1000 0.0303  
## 6 0.8747 nan 0.1000 0.0226  
## 7 0.8286 nan 0.1000 0.0237  
## 8 0.7866 nan 0.1000 0.0195  
## 9 0.7537 nan 0.1000 0.0157  
## 10 0.7217 nan 0.1000 0.0151  
## 20 0.5473 nan 0.1000 0.0038  
## 40 0.3975 nan 0.1000 0.0020  
## 60 0.3169 nan 0.1000 0.0001  
## 80 0.2565 nan 0.1000 0.0005  
## 100 0.2163 nan 0.1000 0.0003  
## 120 0.1813 nan 0.1000 0.0000  
## 140 0.1559 nan 0.1000 -0.0002  
## 150 0.1465 nan 0.1000 0.0003  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2561 nan 0.1000 0.0642  
## 2 1.1440 nan 0.1000 0.0491  
## 3 1.0502 nan 0.1000 0.0453  
## 4 0.9738 nan 0.1000 0.0387  
## 5 0.9065 nan 0.1000 0.0306  
## 6 0.8487 nan 0.1000 0.0278  
## 7 0.7947 nan 0.1000 0.0247  
## 8 0.7491 nan 0.1000 0.0199  
## 9 0.7090 nan 0.1000 0.0186  
## 10 0.6785 nan 0.1000 0.0128  
## 20 0.4493 nan 0.1000 0.0050  
## 40 0.2704 nan 0.1000 -0.0001  
## 60 0.1839 nan 0.1000 -0.0008  
## 80 0.1332 nan 0.1000 0.0003  
## 100 0.1044 nan 0.1000 0.0002  
## 120 0.0839 nan 0.1000 -0.0000  
## 140 0.0675 nan 0.1000 -0.0001  
## 150 0.0612 nan 0.1000 -0.0001  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2397 nan 0.1000 0.0698  
## 2 1.1165 nan 0.1000 0.0603  
## 3 1.0190 nan 0.1000 0.0458  
## 4 0.9403 nan 0.1000 0.0364  
## 5 0.8653 nan 0.1000 0.0324  
## 6 0.7918 nan 0.1000 0.0311  
## 7 0.7297 nan 0.1000 0.0269  
## 8 0.6859 nan 0.1000 0.0195  
## 9 0.6453 nan 0.1000 0.0169  
## 10 0.6036 nan 0.1000 0.0189  
## 20 0.3628 nan 0.1000 0.0059  
## 40 0.1815 nan 0.1000 0.0013  
## 60 0.1157 nan 0.1000 0.0004  
## 80 0.0761 nan 0.1000 -0.0000  
## 100 0.0537 nan 0.1000 -0.0002  
## 120 0.0399 nan 0.1000 -0.0005  
## 140 0.0300 nan 0.1000 -0.0001  
## 150 0.0259 nan 0.1000 -0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2566 nan 0.1000 0.0631  
## 2 1.1584 nan 0.1000 0.0508  
## 3 1.0721 nan 0.1000 0.0428  
## 4 0.9964 nan 0.1000 0.0361  
## 5 0.9323 nan 0.1000 0.0303  
## 6 0.8778 nan 0.1000 0.0264  
## 7 0.8330 nan 0.1000 0.0202  
## 8 0.7903 nan 0.1000 0.0189  
## 9 0.7559 nan 0.1000 0.0173  
## 10 0.7269 nan 0.1000 0.0144  
## 20 0.5573 nan 0.1000 0.0020  
## 40 0.4025 nan 0.1000 0.0005  
## 60 0.3201 nan 0.1000 0.0002  
## 80 0.2619 nan 0.1000 0.0011  
## 100 0.2177 nan 0.1000 0.0001  
## 120 0.1844 nan 0.1000 0.0009  
## 140 0.1580 nan 0.1000 0.0003  
## 150 0.1450 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2458 nan 0.1000 0.0659  
## 2 1.1304 nan 0.1000 0.0523  
## 3 1.0412 nan 0.1000 0.0458  
## 4 0.9610 nan 0.1000 0.0374  
## 5 0.8927 nan 0.1000 0.0298  
## 6 0.8340 nan 0.1000 0.0293  
## 7 0.7802 nan 0.1000 0.0261  
## 8 0.7327 nan 0.1000 0.0224  
## 9 0.6913 nan 0.1000 0.0153  
## 10 0.6530 nan 0.1000 0.0186  
## 20 0.4271 nan 0.1000 0.0037  
## 40 0.2531 nan 0.1000 -0.0018  
## 60 0.1765 nan 0.1000 -0.0000  
## 80 0.1314 nan 0.1000 -0.0003  
## 100 0.1007 nan 0.1000 0.0000  
## 120 0.0787 nan 0.1000 -0.0003  
## 140 0.0617 nan 0.1000 -0.0001  
## 150 0.0550 nan 0.1000 0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2320 nan 0.1000 0.0745  
## 2 1.1194 nan 0.1000 0.0562  
## 3 1.0186 nan 0.1000 0.0472  
## 4 0.9267 nan 0.1000 0.0434  
## 5 0.8484 nan 0.1000 0.0383  
## 6 0.7854 nan 0.1000 0.0296  
## 7 0.7305 nan 0.1000 0.0243  
## 8 0.6724 nan 0.1000 0.0267  
## 9 0.6272 nan 0.1000 0.0221  
## 10 0.5899 nan 0.1000 0.0161  
## 20 0.3428 nan 0.1000 0.0074  
## 40 0.1694 nan 0.1000 0.0017  
## 60 0.1028 nan 0.1000 0.0011  
## 80 0.0697 nan 0.1000 0.0001  
## 100 0.0467 nan 0.1000 -0.0002  
## 120 0.0329 nan 0.1000 0.0000  
## 140 0.0229 nan 0.1000 0.0001  
## 150 0.0198 nan 0.1000 -0.0001  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2699 nan 0.1000 0.0627  
## 2 1.1616 nan 0.1000 0.0507  
## 3 1.0778 nan 0.1000 0.0426  
## 4 1.0005 nan 0.1000 0.0357  
## 5 0.9422 nan 0.1000 0.0290  
## 6 0.8887 nan 0.1000 0.0247  
## 7 0.8396 nan 0.1000 0.0221  
## 8 0.7989 nan 0.1000 0.0197  
## 9 0.7664 nan 0.1000 0.0156  
## 10 0.7349 nan 0.1000 0.0146  
## 20 0.5736 nan 0.1000 0.0028  
## 40 0.4114 nan 0.1000 0.0020  
## 60 0.3150 nan 0.1000 0.0021  
## 80 0.2586 nan 0.1000 0.0000  
## 100 0.2181 nan 0.1000 0.0001  
## 120 0.1910 nan 0.1000 -0.0001  
## 140 0.1637 nan 0.1000 -0.0007  
## 150 0.1570 nan 0.1000 0.0003  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2634 nan 0.1000 0.0609  
## 2 1.1517 nan 0.1000 0.0532  
## 3 1.0632 nan 0.1000 0.0391  
## 4 0.9884 nan 0.1000 0.0355  
## 5 0.9223 nan 0.1000 0.0294  
## 6 0.8639 nan 0.1000 0.0279  
## 7 0.8116 nan 0.1000 0.0212  
## 8 0.7691 nan 0.1000 0.0172  
## 9 0.7257 nan 0.1000 0.0188  
## 10 0.6870 nan 0.1000 0.0171  
## 20 0.4624 nan 0.1000 0.0058  
## 40 0.2851 nan 0.1000 0.0004  
## 60 0.1993 nan 0.1000 0.0009  
## 80 0.1505 nan 0.1000 -0.0003  
## 100 0.1150 nan 0.1000 -0.0002  
## 120 0.0920 nan 0.1000 -0.0001  
## 140 0.0730 nan 0.1000 -0.0002  
## 150 0.0642 nan 0.1000 -0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2493 nan 0.1000 0.0656  
## 2 1.1291 nan 0.1000 0.0593  
## 3 1.0300 nan 0.1000 0.0487  
## 4 0.9451 nan 0.1000 0.0389  
## 5 0.8701 nan 0.1000 0.0346  
## 6 0.8011 nan 0.1000 0.0308  
## 7 0.7459 nan 0.1000 0.0256  
## 8 0.6899 nan 0.1000 0.0250  
## 9 0.6457 nan 0.1000 0.0208  
## 10 0.6090 nan 0.1000 0.0164  
## 20 0.3659 nan 0.1000 0.0050  
## 40 0.1916 nan 0.1000 0.0006  
## 60 0.1215 nan 0.1000 -0.0003  
## 80 0.0835 nan 0.1000 -0.0003  
## 100 0.0613 nan 0.1000 -0.0006  
## 120 0.0459 nan 0.1000 -0.0002  
## 140 0.0354 nan 0.1000 -0.0001  
## 150 0.0311 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2643 nan 0.1000 0.0605  
## 2 1.1652 nan 0.1000 0.0489  
## 3 1.0792 nan 0.1000 0.0400  
## 4 1.0050 nan 0.1000 0.0345  
## 5 0.9481 nan 0.1000 0.0260  
## 6 0.8920 nan 0.1000 0.0239  
## 7 0.8497 nan 0.1000 0.0199  
## 8 0.8042 nan 0.1000 0.0203  
## 9 0.7701 nan 0.1000 0.0163  
## 10 0.7413 nan 0.1000 0.0151  
## 20 0.5627 nan 0.1000 0.0052  
## 40 0.4023 nan 0.1000 0.0024  
## 60 0.3133 nan 0.1000 0.0001  
## 80 0.2561 nan 0.1000 0.0001  
## 100 0.2149 nan 0.1000 0.0008  
## 120 0.1811 nan 0.1000 0.0003  
## 140 0.1534 nan 0.1000 0.0001  
## 150 0.1463 nan 0.1000 -0.0004  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2517 nan 0.1000 0.0657  
## 2 1.1455 nan 0.1000 0.0495  
## 3 1.0515 nan 0.1000 0.0437  
## 4 0.9686 nan 0.1000 0.0394  
## 5 0.9016 nan 0.1000 0.0322  
## 6 0.8432 nan 0.1000 0.0297  
## 7 0.7924 nan 0.1000 0.0246  
## 8 0.7438 nan 0.1000 0.0222  
## 9 0.7046 nan 0.1000 0.0180  
## 10 0.6704 nan 0.1000 0.0149  
## 20 0.4297 nan 0.1000 0.0063  
## 40 0.2535 nan 0.1000 0.0007  
## 60 0.1743 nan 0.1000 0.0007  
## 80 0.1275 nan 0.1000 0.0004  
## 100 0.0958 nan 0.1000 0.0000  
## 120 0.0713 nan 0.1000 0.0002  
## 140 0.0570 nan 0.1000 -0.0004  
## 150 0.0494 nan 0.1000 -0.0002  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2461 nan 0.1000 0.0615  
## 2 1.1257 nan 0.1000 0.0588  
## 3 1.0197 nan 0.1000 0.0495  
## 4 0.9378 nan 0.1000 0.0397  
## 5 0.8622 nan 0.1000 0.0340  
## 6 0.7998 nan 0.1000 0.0268  
## 7 0.7372 nan 0.1000 0.0280  
## 8 0.6877 nan 0.1000 0.0226  
## 9 0.6436 nan 0.1000 0.0177  
## 10 0.6082 nan 0.1000 0.0147  
## 20 0.3659 nan 0.1000 0.0068  
## 40 0.1872 nan 0.1000 0.0015  
## 60 0.1173 nan 0.1000 0.0003  
## 80 0.0808 nan 0.1000 -0.0000  
## 100 0.0599 nan 0.1000 -0.0001  
## 120 0.0436 nan 0.1000 0.0000  
## 140 0.0324 nan 0.1000 0.0000  
## 150 0.0277 nan 0.1000 -0.0000  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 1.2396 nan 0.1000 0.0733  
## 2 1.1197 nan 0.1000 0.0560  
## 3 1.0182 nan 0.1000 0.0478  
## 4 0.9305 nan 0.1000 0.0400  
## 5 0.8614 nan 0.1000 0.0305  
## 6 0.7986 nan 0.1000 0.0289  
## 7 0.7435 nan 0.1000 0.0254  
## 8 0.6906 nan 0.1000 0.0248  
## 9 0.6447 nan 0.1000 0.0193  
## 10 0.6064 nan 0.1000 0.0174  
## 20 0.3672 nan 0.1000 0.0077  
## 40 0.1976 nan 0.1000 0.0006  
## 60 0.1258 nan 0.1000 0.0000  
## 80 0.0876 nan 0.1000 0.0002  
## 100 0.0653 nan 0.1000 -0.0000  
## 120 0.0507 nan 0.1000 -0.0001  
## 140 0.0366 nan 0.1000 -0.0002  
## 150 0.0321 nan 0.1000 -0.0000

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 30

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 30

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 16

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 21

## 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 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 16

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 21

## 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 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 21

## 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 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 21

## 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 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## 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 9

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 29

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## 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 9

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 29

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## 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 9

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## 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 9

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## 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 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## 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 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## 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 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## 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 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 17

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 17

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## 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 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 38

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## 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 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 38

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## 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 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 38

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## 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 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 38

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 19

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 19

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## 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 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 19

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 3

## 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 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 19

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 17

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 14

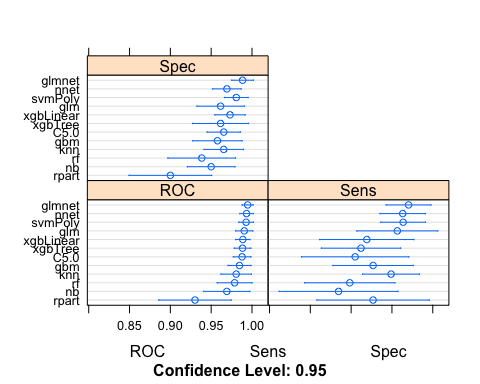
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 17

RESULTS FOR ALL 12 MODELS

results <- resamples(models)  
summary(results)

##   
## Call:  
## summary.resamples(object = results)  
##   
## Models: svmPoly, nnet, gbm, xgbTree, knn, glm, rf, C5.0, nb, rpart, xgbLinear, glmnet   
## Number of resamples: 10   
##   
## ROC   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## svmPoly 0.9615385 0.9941885 0.9985207 0.9927937 1.0000000 1.0000000 0  
## nnet 0.9615385 0.9945055 0.9971471 0.9933221 0.9993132 1.0000000 0  
## gbm 0.9423077 0.9781276 0.9917582 0.9847210 1.0000000 1.0000000 0  
## xgbTree 0.9587912 0.9896978 0.9917582 0.9884193 0.9985207 1.0000000 0  
## knn 0.9258242 0.9665311 0.9928149 0.9806847 0.9989698 1.0000000 0  
## glm 0.9587912 0.9896450 0.9972527 0.9905325 1.0000000 1.0000000 0  
## rf 0.9038462 0.9752747 0.9898563 0.9786982 0.9965659 1.0000000 0  
## C5.0 0.9519231 0.9852071 0.9917582 0.9878381 0.9993132 1.0000000 0  
## nb 0.8791209 0.9574176 0.9862637 0.9691462 0.9955621 1.0000000 0  
## rpart 0.7664835 0.9202768 0.9498626 0.9301881 0.9637574 0.9807692 0  
## xgbLinear 0.9670330 0.9783390 0.9928149 0.9887997 1.0000000 1.0000000 0  
## glmnet 0.9697802 0.9941885 0.9986264 0.9946746 1.0000000 1.0000000 0  
##   
## Sens   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## svmPoly 0.9230769 0.9285714 0.9642857 0.9637363 1.0000000 1 0  
## nnet 0.9230769 0.9285714 0.9642857 0.9631868 1.0000000 1 0  
## gbm 0.7857143 0.9244505 0.9285714 0.9269231 0.9821429 1 0  
## xgbTree 0.7857143 0.8667582 0.9285714 0.9120879 0.9285714 1 0  
## knn 0.8571429 0.9244505 0.9285714 0.9489011 1.0000000 1 0  
## glm 0.7857143 0.9285714 1.0000000 0.9565934 1.0000000 1 0  
## rf 0.7142857 0.8736264 0.9285714 0.8983516 0.9285714 1 0  
## C5.0 0.7142857 0.9244505 0.9285714 0.9049451 0.9285714 1 0  
## nb 0.6428571 0.8736264 0.9258242 0.8846154 0.9285714 1 0  
## rpart 0.7142857 0.8736264 0.9642857 0.9269231 1.0000000 1 0  
## xgbLinear 0.7857143 0.8489011 0.9285714 0.9192308 1.0000000 1 0  
## glmnet 0.9230769 0.9285714 1.0000000 0.9703297 1.0000000 1 0  
##   
## Spec   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## svmPoly 0.9615385 0.9615385 0.9807692 0.9807692 1.0000000 1 0  
## nnet 0.9230769 0.9615385 0.9615385 0.9692308 0.9903846 1 0  
## gbm 0.8461538 0.9615385 0.9615385 0.9576923 0.9615385 1 0  
## xgbTree 0.8846154 0.9326923 0.9807692 0.9615385 1.0000000 1 0  
## knn 0.9230769 0.9326923 0.9615385 0.9653846 1.0000000 1 0  
## glm 0.8846154 0.9326923 0.9615385 0.9615385 1.0000000 1 0  
## rf 0.8076923 0.9230769 0.9615385 0.9384615 0.9615385 1 0  
## C5.0 0.9230769 0.9615385 0.9615385 0.9653846 0.9903846 1 0  
## nb 0.8846154 0.9230769 0.9423077 0.9500000 0.9903846 1 0  
## rpart 0.7692308 0.8846154 0.9038462 0.9000000 0.9519231 1 0  
## xgbLinear 0.9230769 0.9615385 0.9615385 0.9730769 1.0000000 1 0  
## glmnet 0.9615385 0.9711538 1.0000000 0.9884615 1.0000000 1 0

dotplot(results)

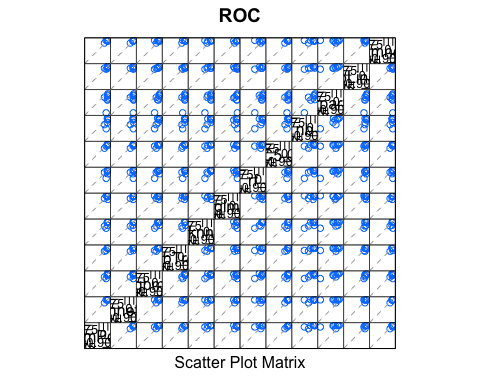


MODEL CORRELATION

mcr <-modelCor(results)  
mcr

## svmPoly nnet gbm xgbTree knn glm  
## svmPoly 1.0000000 0.89711599 0.7818420 0.9205614 0.72888760 0.35342046  
## nnet 0.8971160 1.00000000 0.4615973 0.7443031 0.52666052 0.58990830  
## gbm 0.7818420 0.46159733 1.0000000 0.8638332 0.78471110 0.11126875  
## xgbTree 0.9205614 0.74430307 0.8638332 1.0000000 0.73782635 0.36651268  
## knn 0.7288876 0.52666052 0.7847111 0.7378264 1.00000000 0.06942758  
## glm 0.3534205 0.58990830 0.1112687 0.3665127 0.06942758 1.00000000  
## rf 0.5016434 0.21849995 0.7702786 0.5641396 0.92187063 -0.06555896  
## C5.0 0.9690995 0.90148616 0.7611625 0.8597705 0.76034664 0.42990753  
## nb 0.7372641 0.44171243 0.9029507 0.8406941 0.80213479 0.11961809  
## rpart 0.3986854 0.01540654 0.8587200 0.5613380 0.67635824 -0.09085851  
## xgbLinear 0.6875864 0.40265784 0.9017704 0.6824972 0.71192663 0.11145903  
## glmnet 0.8163915 0.97308004 0.3533068 0.6290544 0.41105357 0.64552000  
## rf C5.0 nb rpart xgbLinear glmnet  
## svmPoly 0.50164335 0.9690995 0.7372641 0.39868536 0.6875864 0.8163915  
## nnet 0.21849995 0.9014862 0.4417124 0.01540654 0.4026578 0.9730800  
## gbm 0.77027858 0.7611625 0.9029507 0.85872003 0.9017704 0.3533068  
## xgbTree 0.56413960 0.8597705 0.8406941 0.56133804 0.6824972 0.6290544  
## knn 0.92187063 0.7603466 0.8021348 0.67635824 0.7119266 0.4110536  
## glm -0.06555896 0.4299075 0.1196181 -0.09085851 0.1114590 0.6455200  
## rf 1.00000000 0.5453594 0.8111701 0.85342040 0.7591158 0.1011451  
## C5.0 0.54535939 1.0000000 0.6868831 0.39099401 0.7383193 0.8590774  
## nb 0.81117012 0.6868831 1.0000000 0.84670547 0.8105586 0.2875802  
## rpart 0.85342040 0.3909940 0.8467055 1.00000000 0.8152487 -0.1065164  
## xgbLinear 0.75911580 0.7383193 0.8105586 0.81524866 1.0000000 0.3519603  
## glmnet 0.10114515 0.8590774 0.2875802 -0.10651640 0.3519603 1.0000000

splom(results)



STACKED MODEL USING BOOSTED TREE MODEL ON META-MODELS

stack <- caretStack(models, method="xgbTree", metric="Sens", verbose = FALSE,  
 trControl = trainControl(method="boot", number=15, savePredictions="final",  
 classProbs=TRUE, summaryFunction=twoClassSummary)  
)  
  
summary(stack)

## Length Class Mode   
## handle 1 xgb.Booster.handle externalptr  
## raw 30284 -none- raw   
## niter 1 -none- numeric   
## call 6 -none- call   
## params 8 -none- list   
## callbacks 0 -none- list   
## feature\_names 12 -none- character   
## nfeatures 1 -none- numeric   
## xNames 12 -none- character   
## problemType 1 -none- character   
## tuneValue 7 data.frame list   
## obsLevels 2 -none- character   
## param 1 -none- list

PREDICT ON TEST DATA SET

test$nnet <- predict(models$nnet, test)  
test$svm <- predict(models$svmPoly, test)  
pred <- predict(stack, test, type="prob")

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 6

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 9

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 47

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 61

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 62

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 63

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 77

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 85

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 102

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 110

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 115

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 122

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 124

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 163

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 164

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 169

threshold <- 0.5  
test$stack <- ifelse(pred>threshold, "M", "B")  
test$stack <- factor(test$stack, levels = c("M", "B"))  
  
summary(test$nnet)

## M B   
## 72 99

summary(test$svm)

## M B   
## 71 100

summary(test$stack)

## M B   
## 71 100

RESULTS

The best performing model the stacked ensemble model trained on the 12 meta-models was a boosted decision tree algorithm.

Comparing individual models with the stacked model, the nn and svm models have the best sensitivity out of all of the individual models and still maintained a high ROC.

The confusion matrix below shows:

nn model had a recall of 0.9091. svm model had a recall of 0.9545. Stacked model had a recall of 1.

Looking at the stacked model it was able to identify all of the malignant cases or breast cancer correctly, it had the highest accuracy compared to the nn and svm and even though a higher recall means that precision worsens, the stacked model did not sacrifice precision because it had the highest precision of the three models as well, at 0.9041.

The stacked model was superior in all metrics and using so many meta-models gave it much for flexibility in identifying the difficult to predict observations that any one model had difficulty predicting correctly.

CONFUSION MATRIX

cmstackpreds <- confusionMatrix(test$nnet, test$diagnosis, positive="M", mode="everything")  
cmstackpreds

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction M B  
## M 71 1  
## B 3 96  
##   
## Accuracy : 0.9766   
## 95% CI : (0.9412, 0.9936)  
## No Information Rate : 0.5673   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9522   
##   
## Mcnemar's Test P-Value : 0.6171   
##   
## Sensitivity : 0.9595   
## Specificity : 0.9897   
## Pos Pred Value : 0.9861   
## Neg Pred Value : 0.9697   
## Precision : 0.9861   
## Recall : 0.9595   
## F1 : 0.9726   
## Prevalence : 0.4327   
## Detection Rate : 0.4152   
## Detection Prevalence : 0.4211   
## Balanced Accuracy : 0.9746   
##   
## 'Positive' Class : M   
##

cmstackpreds <- confusionMatrix(test$svm, test$diagnosis, positive="M", mode="everything")  
cmstackpreds

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction M B  
## M 71 0  
## B 3 97  
##   
## Accuracy : 0.9825   
## 95% CI : (0.9496, 0.9964)  
## No Information Rate : 0.5673   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9641   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 0.9595   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9700   
## Precision : 1.0000   
## Recall : 0.9595   
## F1 : 0.9793   
## Prevalence : 0.4327   
## Detection Rate : 0.4152   
## Detection Prevalence : 0.4152   
## Balanced Accuracy : 0.9797   
##   
## 'Positive' Class : M   
##

cmstackpreds <- confusionMatrix(test$stack, test$diagnosis, positive="M", mode="everything")  
cmstackpreds

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction M B  
## M 71 0  
## B 3 97  
##   
## Accuracy : 0.9825   
## 95% CI : (0.9496, 0.9964)  
## No Information Rate : 0.5673   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9641   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 0.9595   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9700   
## Precision : 1.0000   
## Recall : 0.9595   
## F1 : 0.9793   
## Prevalence : 0.4327   
## Detection Rate : 0.4152   
## Detection Prevalence : 0.4152   
## Balanced Accuracy : 0.9797   
##   
## 'Positive' Class : M   
##

ANALYSIS AND RECOMMENDATION

The stacked ensemble model had a high precision of accuracy in its ability to detect malignant cases which is more important in the analysis and detection than benign cases. This is important because the higher accuracy rate has the potential of saving three or more lives compared to the individual svm and nn. Each model had almost the same in the ability to identify benign cases. This precision is an example of how detecting and treating malignant breast mass is important before it spreads. May God help everyone who suffers from cancer. Please donate to charities that support cancer research.

REFERENCES

“U.S. Breast Cancer Statistics.” Breastcancer.org, <https://www.breastcancer.org/symptoms/understand_bc/statistics>. Street, Nick (1995).

UCI Machine Learning Repository [<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>]. Irvine, CA: University of California, School of Information and Computer Science.