team3-Final

Lorena Dorado Madeline Chang Marvin Moran

Introduction

CRY1 cryptochrome circadian regulator 1 is a gene that helps regulate the circadian clock or the body's internal clock (NIH.gov). The gene is activated by proteins including CLOCK/ARNTL and is turned off in a feedback loop with proteins such as PER/CRY. Therefore, changes in the genes can affect sleep patterns.

The Dataset

The dataset includes records of 90 small molecules created for CRY1. Per dataset creators, 27 of these molecules make the circadian rhythm longer while the remaining 63 molecules have no affect. There are 1177 molecular discriptive features.

Objective

Perform EDA, feature selection and classification tasks. Use Gradient Boosting ML algorithms to identify the best set of molecular descriptors for the model. Finally, perform comprehensive predictive modeling techniques on the data.

EDA

Load Libraries

```r
library(AppliedPredictiveModeling)
library(e1071)
library(caret)
library(rpart)
library(rpart.plot)
library(partykit)
library(earth)
library(kernlab)
library(mlbench)
library(randomForest)
library(dplyr)
library(corrplot)
library(pROC)
library(RANN)
library(glmnet)
library(gt)
```

Read data

```
# Read data
changer_data <- read.csv("data.csv", stringsAsFactors = F)
```

```
# Check for and remove zero variance predictors
degen<- nearZeroVar(changer_data)
changer_new<- changer_data[,-degen]
```

```
# Remove highy correlated variables of 75%
corr<- cor(changer_new[,1:1094])
high_corr <- findCorrelation(corr, cutoff = 0.75)
changer_new<- changer_new[,-high_corr]
```

```
# Count instances per class
changer_new |>
  group_by(Class) |>
  summarise(n = n()) |>
  gt::gt()
```

Class	n
Changer	27
NoChanger	63

```
set.seed(720)

# Create stratified random splits of the data
trainingRows <- createDataPartition(changer_new$Class, p = .5, list = FALSE)
train <- changer_new[trainingRows, ]
```

```
test <- changer_new[-trainingRows, ]

# Create control methods for cross validation
ctrl <- trainControl(method = "cv",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)
```

Required index?

```
# Train models and evaluate with ROC
class_function<- function(method){
  model<- train(x = train[,1:216],
               y = train$Class,
               method = method,
               preProc = c("center", "scale"),
               metric = "ROC",
               trControl = ctrl)
}

# Train models and evaluate hyperparameters and tune with ROC
class_function_tune<- function(method, grid){
  model<- train(x = train[,1:126],
               y = train$Class,
               method = method,
               preProc = c("center", "scale"),
               tuneGrid = grid,
               metric = "ROC",
               trControl = ctrl)
}
```

```
set.seed(720)

# Grid for train to use
glmnGrid <- expand.grid(alpha = c(0,  .1,  .2, .4, .6, .8, 1),
                        lambda = seq(.01, .2, length = 10))

# NSC tuning parameters
nsc_grid<- data.frame(threshold = seq(0, 25, length = 30))
```

```r
lr<- class_function("glm") # Linear Regression
lda<- class_function("lda") # Linear Discriminant Analysis
glmn<- class_function_tune("glmnet", glmnGrid) # Penalized LogReg
nsc<- class_function_tune("pam", nsc_grid) # Nearest Shrunken Centroids
```

```
11111111111
```

```r
# Predict test data set and corresponding ROC
model_roc <- function(model){
  roc(response = model$pred$obs,
             predictor = model$pred$Changer,
             levels = rev(levels(model$pred$obs)))
}

model_roc(lr)
```

```
Setting direction: controls < cases


Call:
roc.default(response = model$pred$obs, predictor = model$pred$Changer,     levels = rev(level

Data: model$pred$Changer in 32 controls (model$pred$obs NoChanger) < 14 cases (model$pred$obs
Area under the curve: 0.6652
```

```r
model_roc(lda)
```

```
Setting direction: controls < cases


Call:
roc.default(response = model$pred$obs, predictor = model$pred$Changer,     levels = rev(level

Data: model$pred$Changer in 32 controls (model$pred$obs NoChanger) < 14 cases (model$pred$obs
Area under the curve: 0.6228
```

```r
model_roc(glmn)
```

```
Setting direction: controls < cases
```

```

Call:
roc.default(response = model$pred$obs, predictor = model$pred$Changer,     levels = rev(leve

Data: model$pred$Changer in 2240 controls (model$pred$obs NoChanger) < 980 cases (model$pred$
Area under the curve: 0.6237
```

```r
model_roc(nsc)
```

```
Setting direction: controls > cases
```

```

Call:
roc.default(response = model$pred$obs, predictor = model$pred$Changer,     levels = rev(leve

Data: model$pred$Changer in 960 controls (model$pred$obs NoChanger) > 420 cases (model$pred$
Area under the curve: 0.5704
```