Brain.R

#setwd(" ")  
load("BloodBrain.RData")  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(RANN)  
####################################################################  
#Remove near zero variance predictors  
####################################################################  
#One interesting aspect of this dataset is that it contains many variables   
#and many of these variables have extemely low variances  
  
# Identify near zero variance predictors: remove\_cols  
remove\_cols <- nearZeroVar(bloodbrain\_x, names = TRUE,   
 freqCut = 2, uniqueCut = 20)  
# Get all column names from bloodbrain\_x: all\_cols  
all\_cols <- names(bloodbrain\_x)  
# Remove from data: bloodbrain\_x\_small  
bloodbrain\_x\_small <- bloodbrain\_x[ , setdiff(all\_cols, remove\_cols)]  
  
####################################################################  
#Fit model on reduced blood-brain data  
####################################################################  
# Fit model on reduced data: model  
model <- train(x = bloodbrain\_x\_small, y = bloodbrain\_y, method = "glm")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

# Print model to console  
model

## Generalized Linear Model   
##   
## 208 samples  
## 112 predictors  
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 208, 208, 208, 208, 208, 208, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 1.739134 0.1042736 1.143086

#Result  
model$results$RMSE

## [1] 1.739134

#RMSE 1.80  
  
####################################################################  
#Using PCA as an alternative to nearZeroVar()  
####################################################################  
#alternative to removing low-variance predictors is to run PCA on your dataset.  
#preferable because it does not throw out all of your data  
#This creates an ideal dataset for linear regression modeling,  
  
# Fit glm model using PCA: model  
model <- train(  
 x = bloodbrain\_x, y = bloodbrain\_y,  
 method = "glm", preProcess = "pca"  
)  
  
# Print model to console  
model

## Generalized Linear Model   
##   
## 208 samples  
## 132 predictors  
##   
## Pre-processing: principal component signal extraction (132),  
## centered (132), scaled (132)   
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 208, 208, 208, 208, 208, 208, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 0.5935343 0.4374797 0.4472667