BCF.R

US16120

Wed Sep 26 14:28:19 2018

library(rcdk)

## Loading required package: rcdklibs

## Loading required package: rJava

library(tidyverse)

## -- Attaching packages ------------------------------------------------------ tidyverse 1.2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.6  
## v tidyr 0.8.1 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts --------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::matches() masks rcdk::matches()

library(magrittr)

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

library(purrr)  
library(stringr)  
library(caret)

## Loading required package: lattice

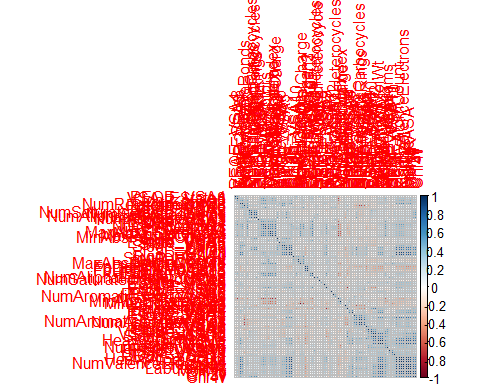
##   
## Attaching package: 'caret'

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

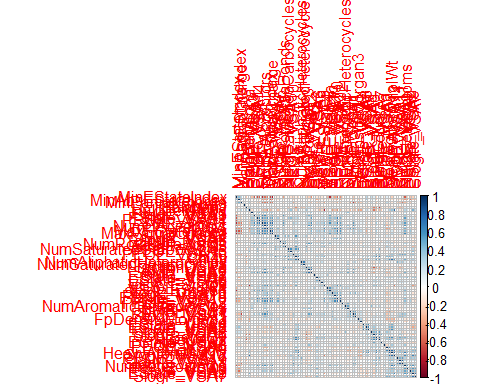
library(corrplot)

## corrplot 0.84 loaded

library(ggplot2)  
library(ggthemes)  
  
# read data  
  
## training data  
train <-  
 read.csv('cache/TR\_BCF\_469\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogBCF, everything()) %>%  
 na.omit()  
  
X\_train <- train %>%  
 select(-LogBCF)  
y\_train <- train %>%  
 select(LogBCF) %>%  
 data.frame()  
  
## test data  
test <-  
 read.csv('cache/TST\_BCF\_157\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogBCF, everything()) %>%  
 na.omit()  
  
X\_test <- test %>%  
 select(-LogBCF)  
y\_test <- test %>%  
 select(LogBCF) %>%  
 data.frame()  
  
# curate data  
  
## near-zero variance descriptors  
  
nzv <- nearZeroVar(X\_train, freqCut = 100/0)  
X\_train <- X\_train[ , -nzv]  
### and  
X\_test <- X\_test[ , -nzv]  
  
## highly correlated descriptors  
  
correlations <- cor(X\_train)  
corrplot::corrplot(correlations, order = 'hclust')



highCorr <- findCorrelation(correlations, cutoff = 0.85)  
X\_train <- X\_train[ , -highCorr]  
### and  
X\_test <- X\_test[ , -highCorr]  
  
correlations <- cor(X\_train)  
corrplot::corrplot(correlations, order = 'hclust')



## linear combinations  
  
comboInfo <- findLinearCombos(X\_train) # returns NULL  
# X\_train <- X\_train[ , -comboInfo$remove]  
# ### and  
# X\_test <- X\_test[ , -nzv]  
  
## center & scale descriptors  
  
preProcValues <- preProcess(X\_train, method = c("center", "scale"))  
  
X\_trainTransformed <- predict(preProcValues, X\_train)  
### and  
X\_testTransformed <- predict(preProcValues, X\_test)  
  
### PCA  
  
# pca <- preProcess(X\_trainTransformed, method = c('pca'))  
# X\_train\_pca <- predict(pca, X\_trainTransformed)  
# X\_test\_pca <- predict(pca, X\_testTransformed)  
#   
# train\_pca <- X\_train\_pca %>%  
# select(PC1, PC2) %>%  
# mutate(dataset = 'train')  
# test\_pca <- X\_test\_pca %>%  
# select(PC1, PC2) %>%  
# mutate(dataset = 'test')  
# pcaPts <- rbind(train\_pca, test\_pca)  
#   
# p <-  
# ggplot(pcaPts, aes(PC1, PC2)) +  
# geom\_point(aes(colour = factor(dataset), shape = factor(dataset))) +  
# ggthemes::theme\_tufte()  
# p  
  
# models  
  
fitControl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 repeats = 5)  
  
set.seed(350)  
  
## multiple linear regression  
  
trainSet <- cbind(y\_train, X\_trainTransformed)  
  
mlr <- train(LogBCF ~ .,  
 data = trainSet,  
 method = 'lm',  
 trControl = fitControl)

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

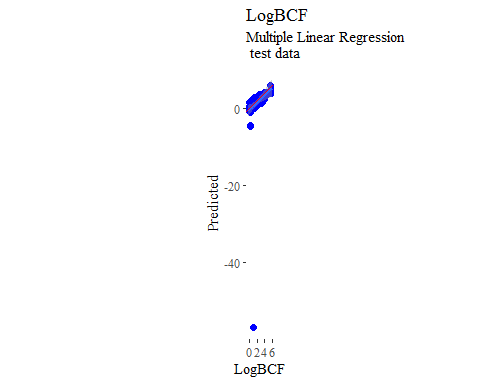
y\_predict <- predict(mlr, newdata = X\_testTransformed) %>%  
 data.frame()

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

colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -57.431 -0.046 0.447 0.812 2.134   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.4571 0.6553 -0.698 0.486536   
## LogBCF 0.9491 0.2728 3.479 0.000654 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.724 on 154 degrees of freedom  
## Multiple R-squared: 0.07287, Adjusted R-squared: 0.06685   
## F-statistic: 12.1 on 1 and 154 DF, p-value: 0.0006543

p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Multiple Linear Regression\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



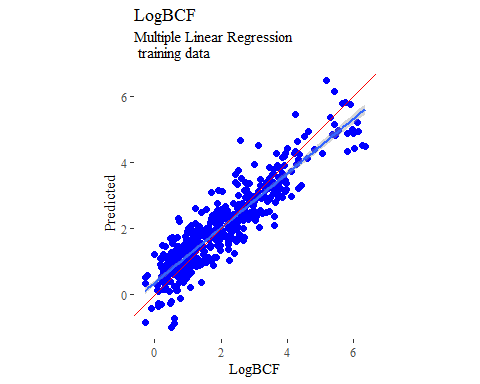
y\_predict <- predict(mlr, newdata = X\_trainTransformed) %>%  
 data.frame()

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

colnames(y\_predict) <- c('Predicted')  
  
mlrPR <- postResample(pred = y\_predict, obs = X\_trainTransformed)  
rmse\_train = c(mlrPR[1])  
r2\_train = c(mlrPR[2])  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.74342 -0.33047 -0.01832 0.34323 2.16910   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.34836 0.04286 8.128 3.96e-15 \*\*\*  
## LogBCF 0.82551 0.01758 46.953 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.532 on 466 degrees of freedom  
## Multiple R-squared: 0.8255, Adjusted R-squared: 0.8251   
## F-statistic: 2205 on 1 and 466 DF, p-value: < 2.2e-16

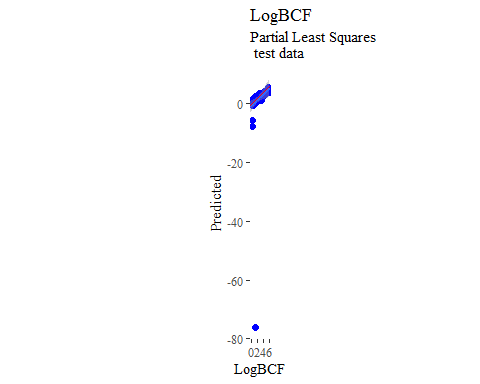
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Multiple Linear Regression\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## partial least squares  
  
plsModel <- train(  
 LogBCF ~ .,  
 data = trainSet,  
 method = 'pls',  
 tuneLength = 20,  
 trControl = fitControl  
)  
  
y\_predict <- predict(plsModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -76.477 0.109 0.569 1.135 2.046   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.6537 0.8698 -0.752 0.45348   
## LogBCF 0.9875 0.3621 2.727 0.00713 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.27 on 154 degrees of freedom  
## Multiple R-squared: 0.04607, Adjusted R-squared: 0.03987   
## F-statistic: 7.437 on 1 and 154 DF, p-value: 0.007133

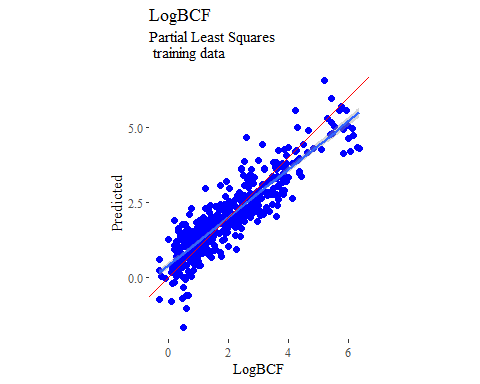
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Partial Least Squares\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(plsModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.45334 -0.33036 -0.02036 0.34090 2.20842   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.39836 0.04513 8.827 <2e-16 \*\*\*  
## LogBCF 0.80046 0.01851 43.236 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5602 on 466 degrees of freedom  
## Multiple R-squared: 0.8005, Adjusted R-squared: 0.8   
## F-statistic: 1869 on 1 and 466 DF, p-value: < 2.2e-16

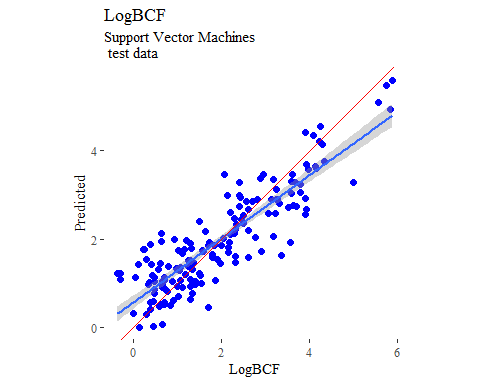
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Partial Least Squares\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## support vector machines  
  
svmModel <- train(  
 LogBCF ~ .,  
 data = trainSet,  
 method = 'svmRadial',  
 # tuneLength = 14,  
 trControl = fitControl  
)  
  
y\_predict <- predict(svmModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.36183 -0.41706 -0.00957 0.34481 1.41859   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.55925 0.07624 7.335 1.18e-11 \*\*\*  
## LogBCF 0.72170 0.03174 22.738 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5496 on 154 degrees of freedom  
## Multiple R-squared: 0.7705, Adjusted R-squared: 0.769   
## F-statistic: 517 on 1 and 154 DF, p-value: < 2.2e-16

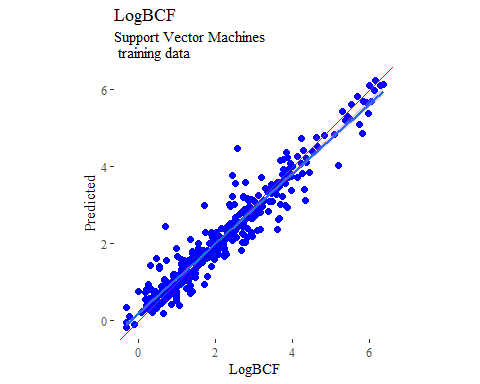
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Support Vector Machines\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(svmModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.09337 -0.16491 -0.00571 0.13352 1.96263   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.18767 0.02696 6.96 1.16e-11 \*\*\*  
## LogBCF 0.90702 0.01106 82.00 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3347 on 466 degrees of freedom  
## Multiple R-squared: 0.9352, Adjusted R-squared: 0.935   
## F-statistic: 6724 on 1 and 466 DF, p-value: < 2.2e-16

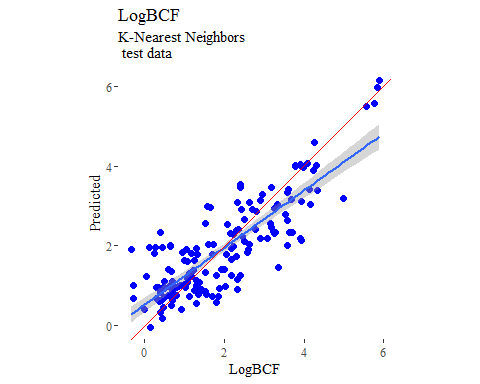
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Support Vector Machines\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## k-nearest neighbors  
  
knnModel <- train(  
 LogBCF ~ .,  
 data = trainSet,  
 method = 'knn',  
 tuneGrid = data.frame(.k = 1:20),  
 trControl = fitControl  
)  
  
y\_predict <- predict(knnModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4781 -0.4704 -0.1196 0.4158 1.6263   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.52441 0.09217 5.69 6.24e-08 \*\*\*  
## LogBCF 0.71637 0.03837 18.67 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6644 on 154 degrees of freedom  
## Multiple R-squared: 0.6936, Adjusted R-squared: 0.6916   
## F-statistic: 348.6 on 1 and 154 DF, p-value: < 2.2e-16

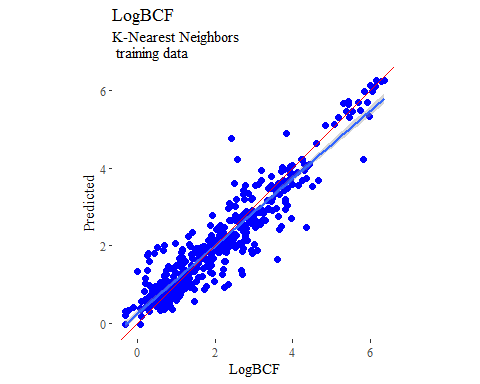
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'K-Nearest Neighbors\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(knnModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.75545 -0.24130 -0.02372 0.24133 2.39580   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.24736 0.03688 6.708 5.73e-11 \*\*\*  
## LogBCF 0.87112 0.01513 57.587 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4578 on 466 degrees of freedom  
## Multiple R-squared: 0.8768, Adjusted R-squared: 0.8765   
## F-statistic: 3316 on 1 and 466 DF, p-value: < 2.2e-16

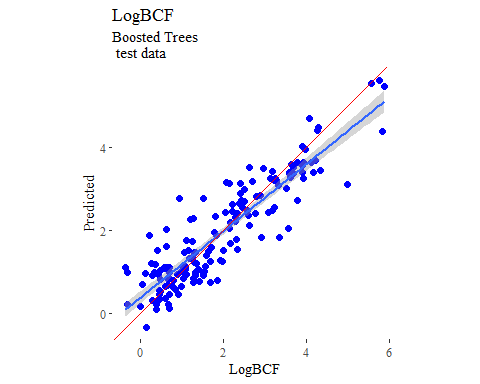
p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'K-Nearest Neighbors\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



## boosted trees  
  
gbmGrid <- expand.grid(  
 .interaction.depth = seq(1, 7, by = 2),  
 .n.trees = seq(100, 1000, by = 50),  
 .shrinkage = c(0.001, 0.1),  
 .n.minobsinnode = 3  
)  
  
treeModel <- train(  
 LogBCF ~ .,  
 data = trainSet,  
 method = 'gbm',  
 tuneGrid = gbmGrid,  
 verbose = FALSE  
)  
  
y\_predict <- predict(treeModel, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.28503 -0.42366 -0.00033 0.31566 1.63184   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.38235 0.07543 5.069 1.13e-06 \*\*\*  
## LogBCF 0.80387 0.03140 25.602 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5437 on 154 degrees of freedom  
## Multiple R-squared: 0.8097, Adjusted R-squared: 0.8085   
## F-statistic: 655.5 on 1 and 154 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Boosted Trees\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



y\_predict <- predict(treeModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogBCF, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogBCF, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.39910 -0.10028 -0.00422 0.10134 0.41701   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.053720 0.012348 4.351 1.67e-05 \*\*\*  
## LogBCF 0.973213 0.005065 192.133 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1533 on 466 degrees of freedom  
## Multiple R-squared: 0.9875, Adjusted R-squared: 0.9875   
## F-statistic: 3.692e+04 on 1 and 466 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogBCF, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogBCF',  
 subtitle = 'Boosted Trees\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p

