KOC.R

US16120

Wed Sep 26 16:55:44 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

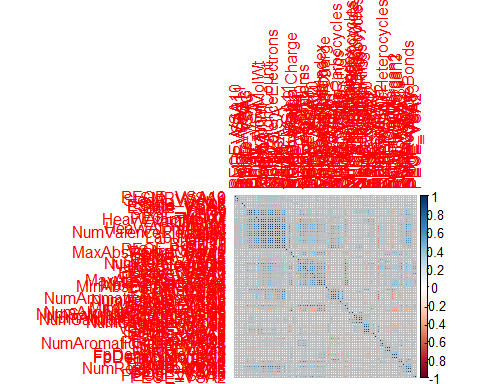
##   
## Attaching package: 'caret'

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

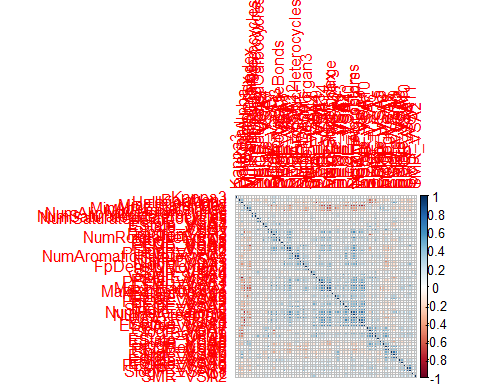
library(corrplot)

## corrplot 0.84 loaded

library(ggplot2)  
library(ggthemes)  
  
# read data  
  
## training data  
train <-  
 read.csv('cache/TR\_KOC\_545\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogKOC, everything()) %>%  
 na.omit()  
  
X\_train <- train %>%  
 select(-LogKOC)  
y\_train <- train %>%  
 select(LogKOC) %>%  
 data.frame()  
  
## test data  
test <-  
 read.csv('cache/TST\_KOC\_184\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogKOC, everything()) %>%  
 na.omit()  
  
X\_test <- test %>%  
 select(-LogKOC)  
y\_test <- test %>%  
 select(LogKOC) %>%  
 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(LogKOC ~ .,  
 data = trainSet,  
 method = 'lm',  
 trControl = fitControl)

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

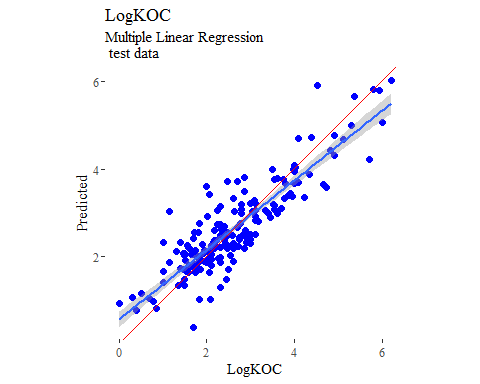
y\_predict <- predict(mlr, newdata = X\_testTransformed) %>%  
 data.frame()

## 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 ~ LogKOC, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.52295 -0.29552 -0.05049 0.27471 1.74040   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.57041 0.08978 6.353 1.66e-09 \*\*\*  
## LogKOC 0.79309 0.03165 25.059 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4903 on 181 degrees of freedom  
## Multiple R-squared: 0.7763, Adjusted R-squared: 0.775   
## F-statistic: 627.9 on 1 and 181 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogKOC, 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 = 'LogKOC',  
 subtitle = 'Multiple Linear Regression\n test data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p



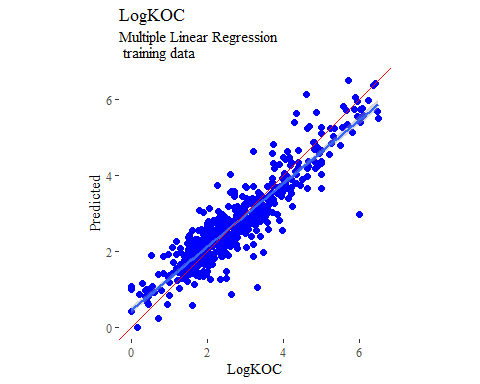
y\_predict <- predict(mlr, newdata = X\_trainTransformed) %>%  
 data.frame()

## 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 ~ LogKOC, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.47053 -0.27568 -0.00295 0.25321 1.84334   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.45209 0.04806 9.406 <2e-16 \*\*\*  
## LogKOC 0.83406 0.01602 52.050 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4676 on 539 degrees of freedom  
## Multiple R-squared: 0.8341, Adjusted R-squared: 0.8338   
## F-statistic: 2709 on 1 and 539 DF, p-value: < 2.2e-16

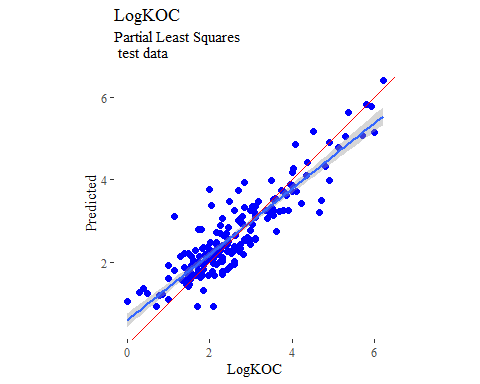
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'Multiple Linear Regression\n training data') +  
 ggthemes::theme\_tufte()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



## partial least squares  
  
plsModel <- train(  
 LogKOC ~ .,  
 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 ~ LogKOC, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.32982 -0.29309 -0.03586 0.26838 1.59708   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.59709 0.08268 7.222 1.37e-11 \*\*\*  
## LogKOC 0.79680 0.02914 27.340 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4515 on 181 degrees of freedom  
## Multiple R-squared: 0.8051, Adjusted R-squared: 0.804   
## F-statistic: 747.5 on 1 and 181 DF, p-value: < 2.2e-16

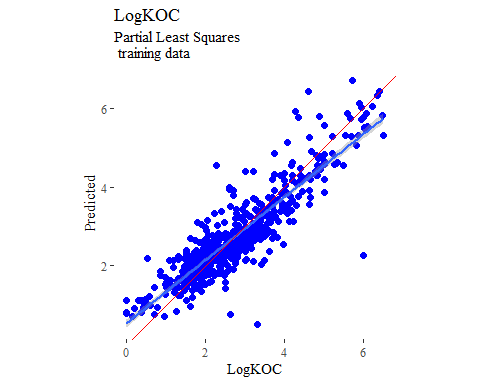
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'Partial Least Squares\n test data') +  
 ggthemes::theme\_tufte()  
p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

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.08367 -0.29460 -0.01871 0.22551 2.19354   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.54927 0.05183 10.6 <2e-16 \*\*\*  
## LogKOC 0.79840 0.01728 46.2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5042 on 539 degrees of freedom  
## Multiple R-squared: 0.7984, Adjusted R-squared: 0.798   
## F-statistic: 2135 on 1 and 539 DF, p-value: < 2.2e-16

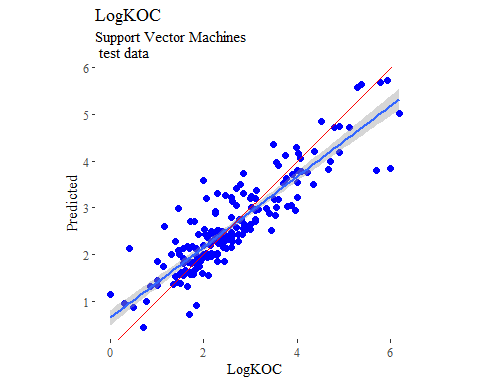
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'Partial Least Squares\n training data') +  
 ggthemes::theme\_tufte()  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



## support vector machines  
  
svmModel <- train(  
 LogKOC ~ .,  
 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 ~ LogKOC, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.3246 -0.2831 -0.1057 0.2398 1.4323   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.64801 0.08405 7.709 8.1e-13 \*\*\*  
## LogKOC 0.75553 0.02963 25.499 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.459 on 181 degrees of freedom  
## Multiple R-squared: 0.7822, Adjusted R-squared: 0.781   
## F-statistic: 650.2 on 1 and 181 DF, p-value: < 2.2e-16

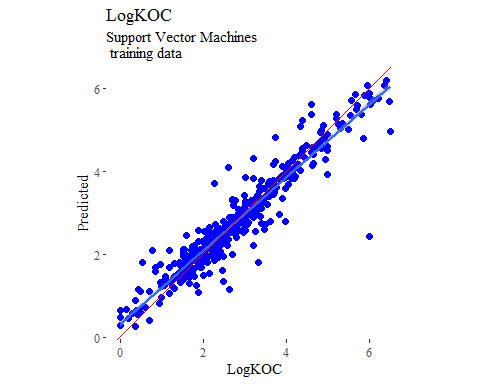
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'Support Vector Machines\n test data') +  
 ggthemes::theme\_tufte()  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

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.1836 -0.1249 -0.0014 0.1391 1.4805   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.31465 0.03539 8.892 <2e-16 \*\*\*  
## LogKOC 0.88230 0.01180 74.788 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3442 on 539 degrees of freedom  
## Multiple R-squared: 0.9121, Adjusted R-squared: 0.9119   
## F-statistic: 5593 on 1 and 539 DF, p-value: < 2.2e-16

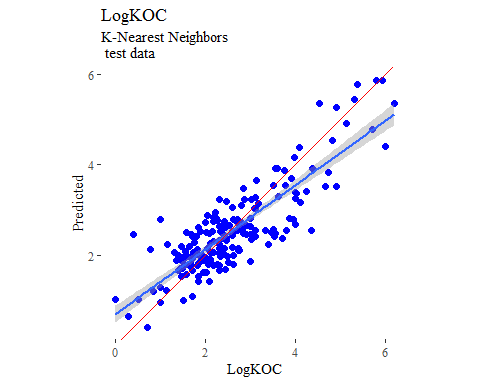
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'Support Vector Machines\n training data') +  
 ggthemes::theme\_tufte()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



## k-nearest neighbors  
  
knnModel <- train(  
 LogKOC ~ .,  
 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 ~ LogKOC, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.26581 -0.36707 -0.02109 0.32428 1.47473   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.68683 0.09391 7.314 8.13e-12 \*\*\*  
## LogKOC 0.71609 0.03311 21.631 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5128 on 181 degrees of freedom  
## Multiple R-squared: 0.7211, Adjusted R-squared: 0.7195   
## F-statistic: 467.9 on 1 and 181 DF, p-value: < 2.2e-16

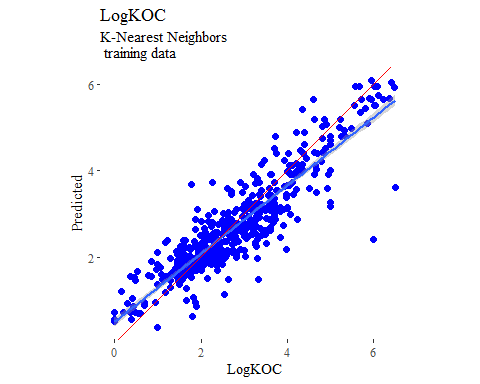
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'K-Nearest Neighbors\n test data') +  
 ggthemes::theme\_tufte()  
p <-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

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.8329 -0.2485 -0.0029 0.2419 1.7752   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.50410 0.04875 10.34 <2e-16 \*\*\*  
## LogKOC 0.78914 0.01625 48.55 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4743 on 539 degrees of freedom  
## Multiple R-squared: 0.8139, Adjusted R-squared: 0.8135   
## F-statistic: 2357 on 1 and 539 DF, p-value: < 2.2e-16

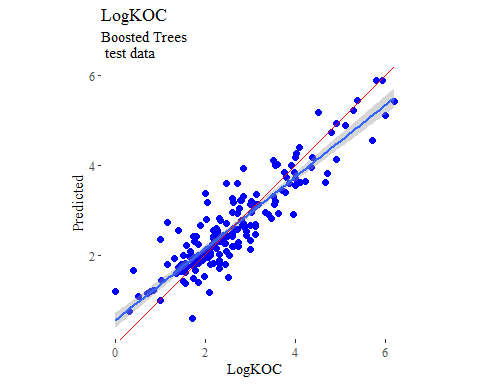
p <-  
 ggplot(data2plot, aes(LogKOC, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogKOC',  
 subtitle = 'K-Nearest Neighbors\n training data') +  
 ggthemes::theme\_tufte()  
p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

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.32238 -0.23256 -0.05424 0.25146 1.25167   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.55317 0.07989 6.924 7.39e-11 \*\*\*  
## LogKOC 0.79654 0.02816 28.283 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4363 on 181 degrees of freedom  
## Multiple R-squared: 0.8155, Adjusted R-squared: 0.8145   
## F-statistic: 799.9 on 1 and 181 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogKOC, 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 = 'LogKOC',  
 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 ~ LogKOC, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOC, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.19428 -0.20619 -0.01445 0.18003 1.28913   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.32648 0.03408 9.581 <2e-16 \*\*\*  
## LogKOC 0.88149 0.01136 77.594 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3315 on 539 degrees of freedom  
## Multiple R-squared: 0.9178, Adjusted R-squared: 0.9177   
## F-statistic: 6021 on 1 and 539 DF, p-value: < 2.2e-16

p <-  
 ggplot(data2plot, aes(LogKOC, 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 = 'LogKOC',  
 subtitle = 'Boosted Trees\n training data') +  
 ggthemes::theme\_tufte()  
p <- p + geom\_abline(intercept = 0,  
 slope = 1,  
 colour = 'red')  
p

