KOA.R

US16120

Wed Sep 26 16:48:05 2018

library(rcdk)

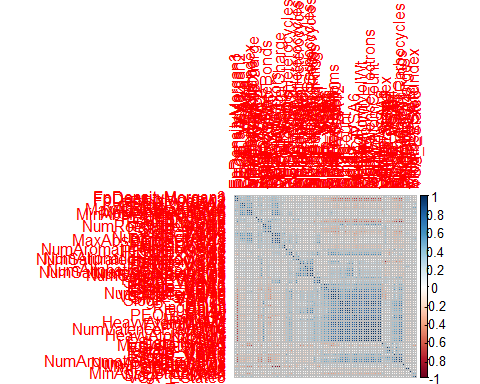
library(tidyverse)

library(magrittr)

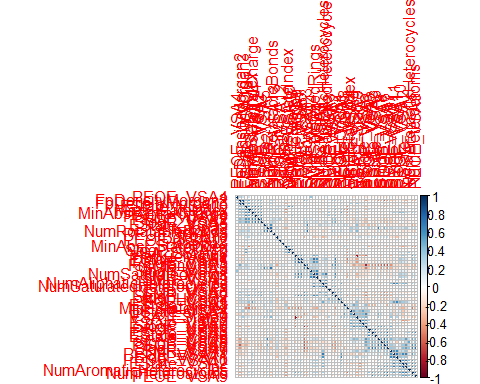
library(purrr)  
library(stringr)  
library(caret)

library(corrplot)

library(ggplot2)  
library(ggthemes)  
  
# read data  
  
## training data  
train <-  
 read.csv('cache/TR\_KOA\_202\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogKOA, everything()) %>%  
 na.omit()  
  
X\_train <- train %>%  
 select(-LogKOA)  
y\_train <- train %>%  
 select(LogKOA) %>%  
 data.frame()  
  
## test data  
test <-  
 read.csv('cache/TST\_KOA\_68\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogKOA, everything()) %>%  
 na.omit()  
  
X\_test <- test %>%  
 select(-LogKOA)  
y\_test <- test %>%  
 select(LogKOA) %>%  
 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')

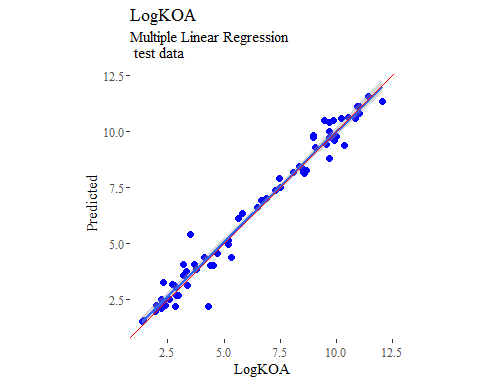


## linear combinations  
  
comboInfo <- findLinearCombos(X\_train) # returns NULL  
# X\_train <- X\_train[ , -comboInfo$remove]  
# ### and  
# X\_test <- X\_test[ , -comboInfo$remove]  
  
## center & scale descriptors  
  
preProcValues <- preProcess(X\_train, method = c("center", "scale"))  
  
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(LogKOA ~ .,  
 data = trainSet,  
 method = 'lm',  
 trControl = fitControl)

y\_predict <- predict(mlr, newdata = X\_testTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogKOA, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.21041 -0.26963 0.05195 0.24500 1.77858   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.18294 0.14621 1.251 0.215   
## LogKOA 0.98054 0.02075 47.260 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5519 on 65 degrees of freedom  
## Multiple R-squared: 0.9717, Adjusted R-squared: 0.9713   
## F-statistic: 2234 on 1 and 65 DF, p-value: < 2.2e-16

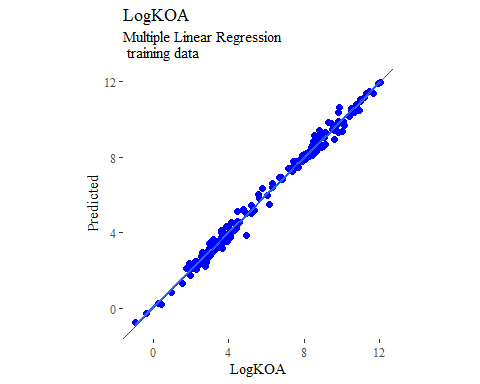
p <-  
 ggplot(data2plot, aes(LogKOA, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOA',  
 subtitle = 'Multiple Linear Regression\n test data') +  
 ggthemes::theme\_tufte()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



y\_predict <- predict(mlr, newdata = X\_trainTransformed) %>%  
 data.frame()  
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 ~ LogKOA, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.12212 -0.16886 -0.01541 0.13341 0.81513   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.047987 0.042413 1.131 0.259   
## LogKOA 0.992074 0.006286 157.821 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2654 on 199 degrees of freedom  
## Multiple R-squared: 0.9921, Adjusted R-squared: 0.992   
## F-statistic: 2.491e+04 on 1 and 199 DF, p-value: < 2.2e-16

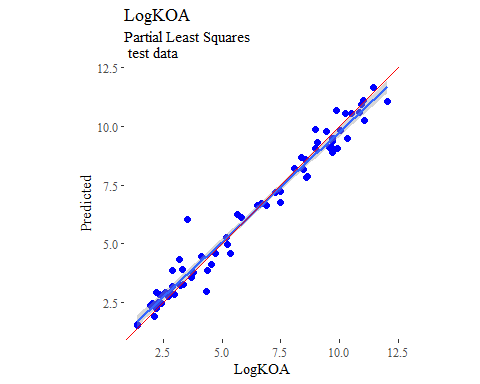
p <-  
 ggplot(data2plot, aes(LogKOA, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogKOA',  
 subtitle = 'Multiple Linear Regression\n training data') +  
 ggthemes::theme\_tufte()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



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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.49209 -0.27994 -0.02858 0.27943 2.36691   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.39337 0.14649 2.685 0.00919 \*\*   
## LogKOA 0.93801 0.02079 45.123 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.553 on 65 degrees of freedom  
## Multiple R-squared: 0.9691, Adjusted R-squared: 0.9686   
## F-statistic: 2036 on 1 and 65 DF, p-value: < 2.2e-16

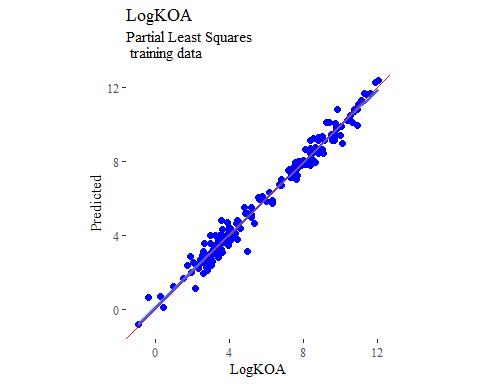
p <-  
 ggplot(data2plot, aes(LogKOA, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOA',  
 subtitle = 'Partial Least Squares\n test data') +  
 ggthemes::theme\_tufte()  
p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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.86258 -0.25394 0.00024 0.25952 1.16115   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.12342 0.06759 1.826 0.0693 .   
## LogKOA 0.97961 0.01002 97.790 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.423 on 199 degrees of freedom  
## Multiple R-squared: 0.9796, Adjusted R-squared: 0.9795   
## F-statistic: 9563 on 1 and 199 DF, p-value: < 2.2e-16

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

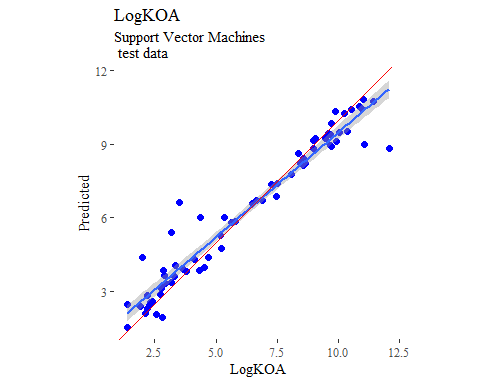


## support vector machines  
  
svmModel <- train(  
 LogKOA ~ .,  
 data = trainSet,  
 method = 'svmRadial',  
 # tuneLength = 14,  
 trControl = fitControl  
)

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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.44189 -0.32741 -0.04353 0.29408 2.71014   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.91919 0.19461 4.723 1.28e-05 \*\*\*  
## LogKOA 0.85618 0.02762 31.004 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7346 on 65 degrees of freedom  
## Multiple R-squared: 0.9367, Adjusted R-squared: 0.9357   
## F-statistic: 961.2 on 1 and 65 DF, p-value: < 2.2e-16

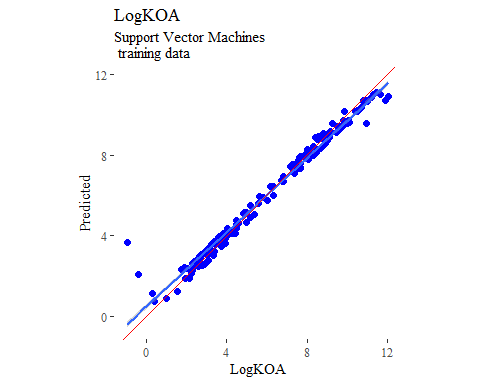
p <-  
 ggplot(data2plot, aes(LogKOA, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOA',  
 subtitle = 'Support Vector Machines\n test data') +  
 ggthemes::theme\_tufte()  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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.9842 -0.1534 0.0032 0.1143 4.0421   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.475658 0.065643 7.246 9.24e-12 \*\*\*  
## LogKOA 0.923747 0.009729 94.947 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4108 on 199 degrees of freedom  
## Multiple R-squared: 0.9784, Adjusted R-squared: 0.9783   
## F-statistic: 9015 on 1 and 199 DF, p-value: < 2.2e-16

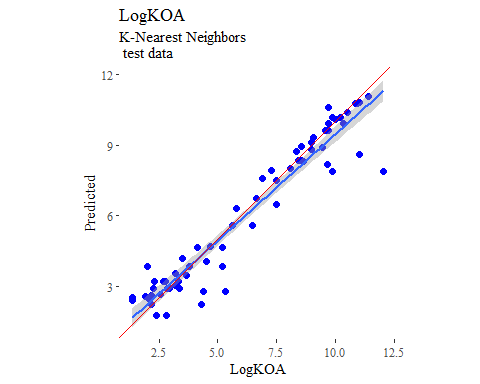
p <-  
 ggplot(data2plot, aes(LogKOA, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'LogKOA',  
 subtitle = 'Support Vector Machines\n training data') +  
 ggthemes::theme\_tufte()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



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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.4538 -0.2268 0.2484 0.5397 1.6034   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.41900 0.23910 1.752 0.0844 .   
## LogKOA 0.90380 0.03393 26.638 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9026 on 65 degrees of freedom  
## Multiple R-squared: 0.9161, Adjusted R-squared: 0.9148   
## F-statistic: 709.6 on 1 and 65 DF, p-value: < 2.2e-16

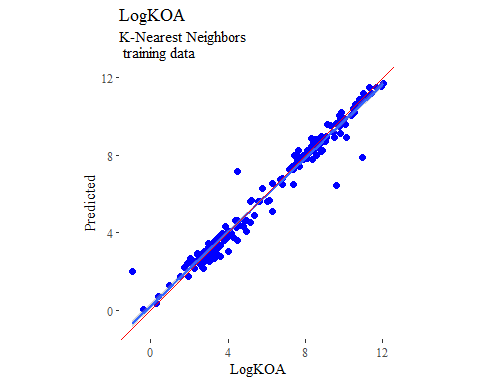
p <-  
 ggplot(data2plot, aes(LogKOA, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 # xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'LogKOA',  
 subtitle = 'K-Nearest Neighbors\n test data') +  
 ggthemes::theme\_tufte()  
p <-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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.95458 -0.21892 0.07923 0.18503 2.69711   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.21523 0.08217 2.619 0.00949 \*\*   
## LogKOA 0.95509 0.01218 78.428 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5142 on 199 degrees of freedom  
## Multiple R-squared: 0.9687, Adjusted R-squared: 0.9685   
## F-statistic: 6151 on 1 and 199 DF, p-value: < 2.2e-16

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



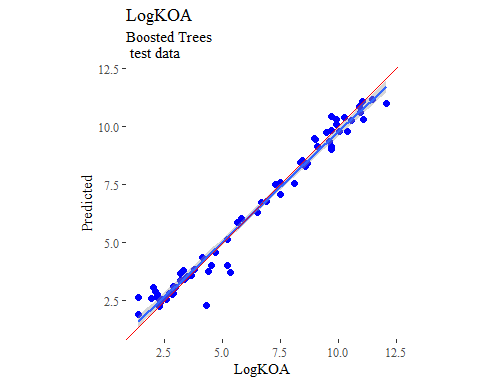
## boosted trees  
  
gbmGrid <- expand.grid(  
 .interaction.depth = seq(1, 7, by = 2),  
 .n.trees = seq(100, 1000, by = 50),  
 .shrinkage = c(0.001, 0.1),  
 .n.minobsinnode = 3  
)  
  
treeModel <- train(  
 LogKOA ~ .,  
 data = trainSet,  
 method = 'gbm',  
 tuneGrid = gbmGrid,  
 verbose = FALSE  
)

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 37: PEOE\_VSA3 has no variation.

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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.10253 -0.18843 -0.03465 0.31008 1.02493   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.27625 0.13780 2.005 0.0492 \*   
## LogKOA 0.94924 0.01955 48.545 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5202 on 65 degrees of freedom  
## Multiple R-squared: 0.9732, Adjusted R-squared: 0.9727   
## F-statistic: 2357 on 1 and 65 DF, p-value: < 2.2e-16

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

##   
## Call:  
## lm(formula = Predicted ~ LogKOA, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.996e-03 -7.170e-04 -1.969e-05 6.771e-04 2.686e-03   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.553e-06 1.638e-04 -0.028 0.978   
## LogKOA 1.000e+00 2.427e-05 41202.591 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.001025 on 199 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 1.698e+09 on 1 and 199 DF, p-value: < 2.2e-16

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

