BioHL.R

US16120

Wed Sep 26 13:51:18 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

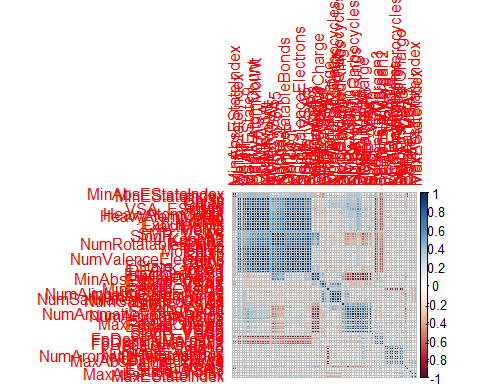
##   
## Attaching package: 'caret'

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

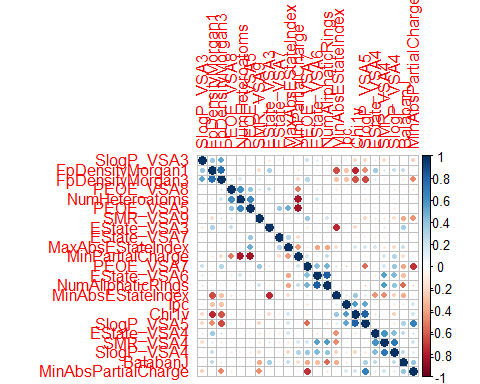
library(corrplot)

## corrplot 0.84 loaded

library(ggplot2)  
library(ggthemes)  
  
# read data  
  
## training data  
train <-  
 read.csv('cache/TR\_BioHL\_112\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogHalfLife, everything()) %>%  
 na.omit()  
  
X\_train <- train %>%  
 select(-LogHalfLife)  
y\_train <- train %>%  
 select(LogHalfLife) %>%  
 data.frame()  
  
## test data  
test <-  
 read.csv('cache/TST\_BioHL\_38\_descrs.csv',  
 header = TRUE,  
 stringsAsFactors = FALSE) %>%  
 select(-X,-CAS,-ROMol,-SMILES,-ID) %>%  
 select(LogHalfLife, everything()) %>%  
 na.omit()  
  
X\_test <- test %>%  
 select(-LogHalfLife)  
y\_test <- test %>%  
 select(LogHalfLife) %>%  
 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')



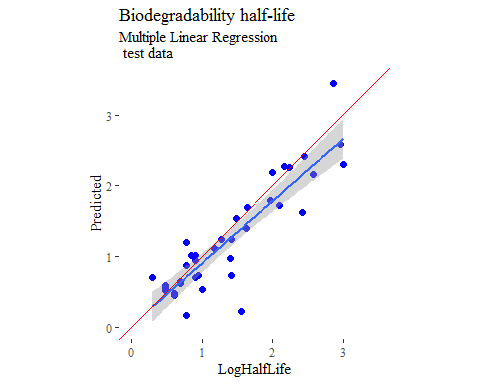
## linear combinations  
  
comboInfo <- findLinearCombos(X\_train) # returns NULL  
# X\_train <- X\_train[ , -comboInfo$remove]  
# ### and  
# X\_test <- X\_test[ , -nzv]  
  
## center & scale descriptors  
  
preProcValues <- preProcess(X\_train, method = c("center", "scale"))  
  
X\_trainTransformed <- 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(LogHalfLife ~ .,  
 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()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_test, y\_predict)  
  
summary(lm(Predicted ~ LogHalfLife, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.16679 -0.13525 0.04206 0.20199 0.91022   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.02377 0.12387 0.192 0.849   
## LogHalfLife 0.88116 0.07741 11.383 2.58e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3623 on 35 degrees of freedom  
## Multiple R-squared: 0.7873, Adjusted R-squared: 0.7812   
## F-statistic: 129.6 on 1 and 35 DF, p-value: 2.575e-13

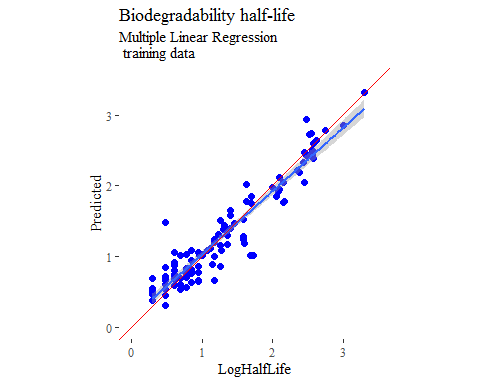
p <-  
 ggplot(data2plot, aes(LogHalfLife, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'Biodegradability half-life',  
 subtitle = 'Multiple Linear Regression\n test data') +  
 ggthemes::theme\_tufte()  
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 ~ LogHalfLife, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.67094 -0.11301 0.01151 0.12722 0.92639   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.13091 0.04297 3.046 0.00291 \*\*   
## LogHalfLife 0.89812 0.02897 30.998 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2262 on 109 degrees of freedom  
## Multiple R-squared: 0.8981, Adjusted R-squared: 0.8972   
## F-statistic: 960.9 on 1 and 109 DF, p-value: < 2.2e-16

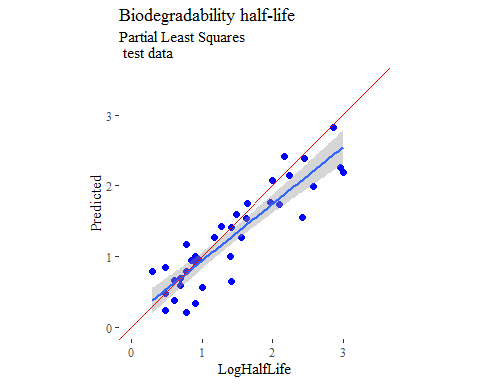
p <-  
 ggplot(data2plot, aes(LogHalfLife, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'Biodegradability half-life',  
 subtitle = 'Multiple Linear Regression\n training data') +  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



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

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.62975 -0.24266 0.04881 0.25678 0.54050   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.13520 0.10549 1.282 0.208   
## LogHalfLife 0.80491 0.06593 12.209 3.59e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3085 on 35 degrees of freedom  
## Multiple R-squared: 0.8098, Adjusted R-squared: 0.8044   
## F-statistic: 149 on 1 and 35 DF, p-value: 3.587e-14

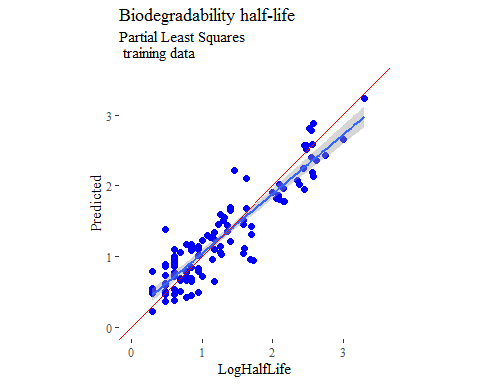
p <-  
 ggplot(data2plot, aes(LogHalfLife, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'Biodegradability half-life',  
 subtitle = 'Partial Least Squares\n test data') +  
 ggthemes::theme\_tufte()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

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.71397 -0.18106 0.00383 0.19994 0.78415   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.20435 0.05195 3.933 0.000148 \*\*\*  
## LogHalfLife 0.84097 0.03503 24.008 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2734 on 109 degrees of freedom  
## Multiple R-squared: 0.841, Adjusted R-squared: 0.8395   
## F-statistic: 576.4 on 1 and 109 DF, p-value: < 2.2e-16

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



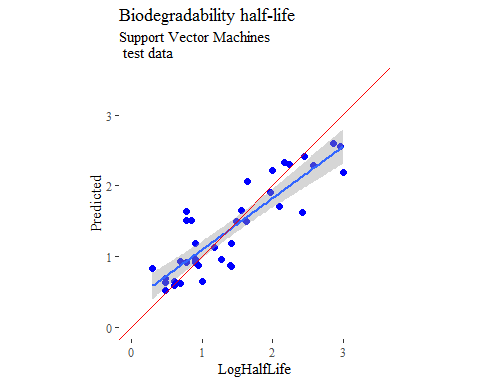
## support vector machines  
  
svmModel <- train(  
 LogHalfLife ~ .,  
 data = trainSet,  
 method = 'svmRadial',  
 # tuneLength = 14,  
 trControl = fitControl  
)

## Warning in .local(x, ...): Variable(s) `' constant. Cannot scale data.

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

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.54633 -0.19495 -0.02926 0.16079 0.70957   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.36366 0.10927 3.328 0.00207 \*\*   
## LogHalfLife 0.73121 0.06829 10.707 1.38e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3196 on 35 degrees of freedom  
## Multiple R-squared: 0.7661, Adjusted R-squared: 0.7594   
## F-statistic: 114.6 on 1 and 35 DF, p-value: 1.377e-12

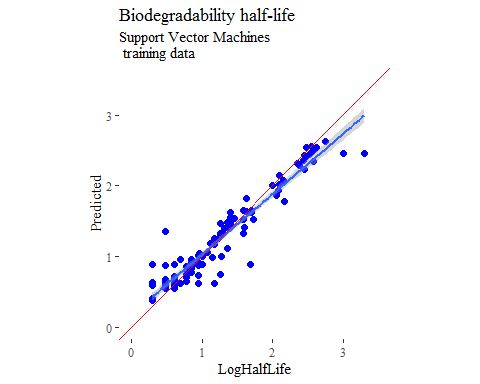
p <-  
 ggplot(data2plot, aes(LogHalfLife, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method = 'lm') +  
 labs(title = 'Biodegradability half-life',  
 subtitle = 'Support Vector Machines\n test data') +  
 ggthemes::theme\_tufte()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

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.72730 -0.07583 0.00194 0.10601 0.77608   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.17270 0.03615 4.778 5.56e-06 \*\*\*  
## LogHalfLife 0.85552 0.02437 35.106 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1902 on 109 degrees of freedom  
## Multiple R-squared: 0.9187, Adjusted R-squared: 0.918   
## F-statistic: 1232 on 1 and 109 DF, p-value: < 2.2e-16

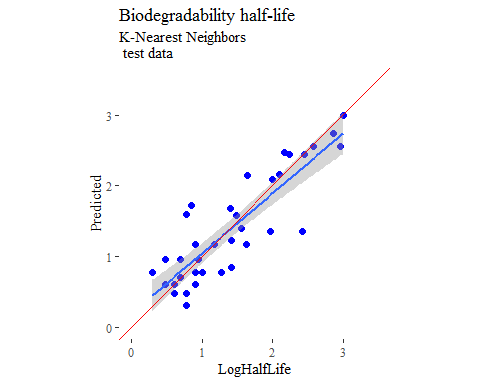
p <-  
 ggplot(data2plot, aes(LogHalfLife, Predicted)) +  
 geom\_point(colour = "blue", size = 2) +  
 coord\_equal() +  
 xlim(c(0, 3.5)) + ylim(c(0, 3.5)) +  
 geom\_smooth(method='lm') +  
 labs(title = 'Biodegradability half-life',  
 subtitle = 'Support Vector Machines\n training data') +  
 ggthemes::theme\_tufte()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



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

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.88514 -0.22811 0.00319 0.21493 0.81245   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.19320 0.12964 1.49 0.145   
## LogHalfLife 0.85037 0.08102 10.49 2.36e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3792 on 35 degrees of freedom  
## Multiple R-squared: 0.7589, Adjusted R-squared: 0.752   
## F-statistic: 110.2 on 1 and 35 DF, p-value: 2.357e-12

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

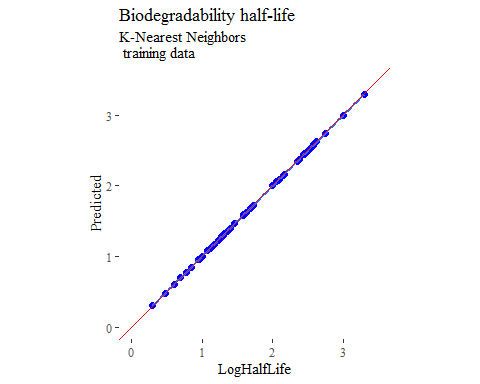


y\_predict <- predict(knnModel, newdata = X\_trainTransformed) %>%  
 data.frame()  
colnames(y\_predict) <- c('Predicted')  
  
data2plot <- cbind(y\_train, y\_predict)  
  
summary(lm(Predicted ~ LogHalfLife, data = data2plot))

## Warning in summary.lm(lm(Predicted ~ LogHalfLife, data = data2plot)):  
## essentially perfect fit: summary may be unreliable

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.645e-16 -2.605e-17 -5.910e-18 4.104e-17 4.065e-16   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.372e-16 1.381e-17 -2.442e+01 <2e-16 \*\*\*  
## LogHalfLife 1.000e+00 9.309e-18 1.074e+17 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.266e-17 on 109 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 1.154e+34 on 1 and 109 DF, p-value: < 2.2e-16

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



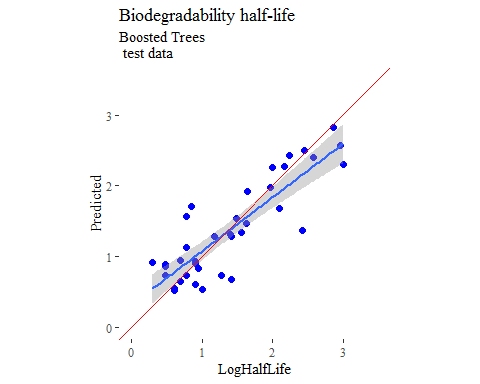
## boosted trees  
  
gbmGrid <- expand.grid(  
 .interaction.depth = seq(1, 7, by = 2),  
 .n.trees = seq(100, 1000, by = 50),  
 .shrinkage = c(0.001, 0.1),  
 .n.minobsinnode = 3  
)  
  
treeModel <- train(  
 LogHalfLife ~ .,  
 data = trainSet,  
 method = 'gbm',  
 tuneGrid = gbmGrid,  
 verbose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

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 15: NumHeteroatoms 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 ~ LogHalfLife, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.7799 -0.2153 0.0134 0.2264 0.7511   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.31501 0.12191 2.584 0.0141 \*   
## LogHalfLife 0.76107 0.07619 9.989 8.73e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3566 on 35 degrees of freedom  
## Multiple R-squared: 0.7403, Adjusted R-squared: 0.7329   
## F-statistic: 99.78 on 1 and 35 DF, p-value: 8.729e-12

p <-  
 ggplot(data2plot, aes(LogHalfLife, 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 = 'Biodegradability half-life',  
 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 ~ LogHalfLife, data = data2plot))

##   
## Call:  
## lm(formula = Predicted ~ LogHalfLife, data = data2plot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.06550 -0.01522 0.00081 0.01652 0.06375   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.011793 0.004685 2.517 0.0133 \*   
## LogHalfLife 0.991359 0.003159 313.831 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.02466 on 109 degrees of freedom  
## Multiple R-squared: 0.9989, Adjusted R-squared: 0.9989   
## F-statistic: 9.849e+04 on 1 and 109 DF, p-value: < 2.2e-16

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

