LR4

Kovel

30 РѕРєС‚СЏР±СЂСЏ 2020 Рі

# Download the files

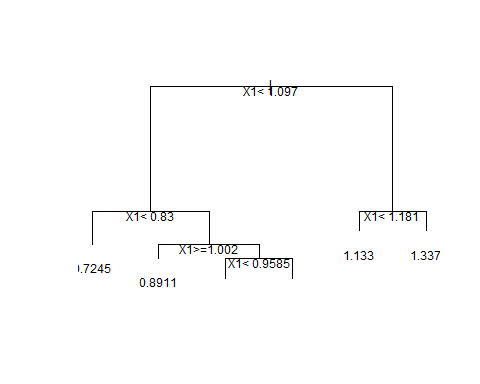
f\_train <- read.csv2('indeks\_train.csv', header = TRUE, encoding = 'UNICOD')  
f\_test <- read.csv2('indeks\_test.csv', header = TRUE, encoding = 'UNICOD')

## Висновок: окремо завантажені навчальна і тестова вибірки.

# Decision Tree Regression

# Fitting simple tree

# install.packages('rpart')  
library(rpart)  
dt <- rpart(Y ~ X1, f\_train, control = rpart.control(minsplit = 20))  
plot(dt)  
text(dt, pos = 1, cex = .75, col = 1, font = 1)

 ##Висновок:побудовано дерево рішень, екзогенна змінна – X1. #Predicting

p\_dt <- predict(dt, f\_test)  
  
train\_mse\_dt <- sum((f\_train$Y-predict(dt, f\_train))^2) /length(f\_train$Y)  
test\_mse\_dt <- sum((f\_test$Y-p\_dt)^2)/length(p\_dt)  
  
train\_mse\_dt

## [1] 0.01301742

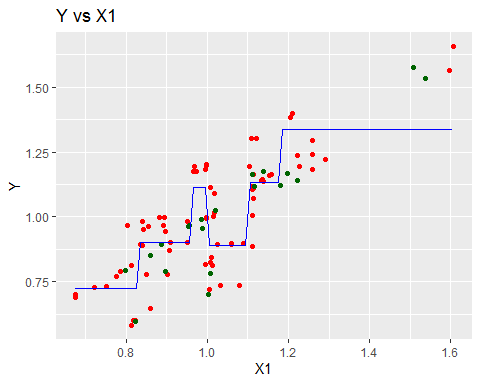
test\_mse\_dt

## [1] 0.02230507

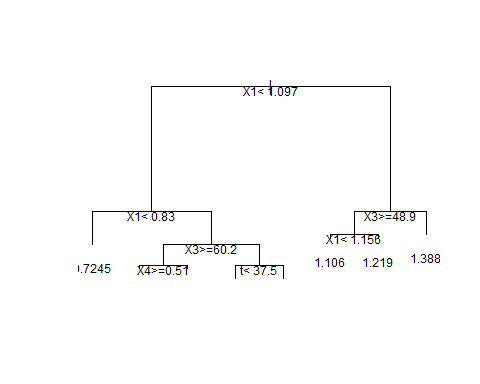
## Висновок:значення середньоквадратичної похибки трохи покращилися на навчальній вибірці – 0.01301742 (було 0.01334668), погіршилися на тестовій вибірці – 0.02230507(було 0.008486525). Модель перенавчено.

# Visualising

library(ggplot2)  
x\_grid <- seq(min(f\_train$X1), max(f\_train$X1), 0.01)  
ggplot() +  
 geom\_point(aes(f\_train$X1, f\_train$Y),colour = 'red') +  
 geom\_point(aes(f\_test$X1, f\_test$Y),colour = 'dark green') +  
 geom\_line(aes(x\_grid, predict(dt, data.frame(X1 = x\_grid))),colour = 'blue') +  
 ggtitle('Y vs X1') +  
 xlab('X1') +  
 ylab('Y')

 ##Висновок:Висновок: на графіку червоним позначені точки навчальної вибірки, зеленим – точки тестової вибірки, синім – модельні значення. #Fitting full tree

# install.packages('rpart')  
library(rpart)  
dt <- rpart(Y ~ t + X1 + X3 + X4, f\_train, control = rpart.control(minsplit = 20))  
plot(dt)  
text(dt, pos = 1, cex = .75, col = 1, font = 1)

 ##Висновок:побудовано дерево рішень, екзогенні змінні – t,X1,X3.X4 #Predicting

p\_dt <- predict(dt, f\_test)  
  
train\_mse\_dt <- sum((f\_train$Y-predict(dt, f\_train))^2)/length(f\_train$Y)  
test\_mse\_dt <- sum((f\_test$Y-p\_dt)^2)/length(p\_dt)  
  
train\_mse\_dt

## [1] 0.009771367

test\_mse\_dt

## [1] 0.0137824

## Висновок:значення середньоквадратичної похибки трохи покращилися на навчальній вибірці – 0.009771367 (було 0.01301742), також є покращення і на тестовій вибірці – 0.0137824(було 0.02230507). Та модель все одно перенавчено.

# Visualising

# Random forest

# Fitting

# install.packages('randomForest')  
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

set.seed(1234)  
rf = randomForest(x = f\_train['X1'],  
 y = f\_train$Y,  
 ntree = 5)

## Висновок: побудовано віпадковий ліс із 5 дерев, екзогенна змінна – X1.

# Predicting

p\_rf <- predict(rf, f\_test)  
  
train\_mse\_rf <- sum((f\_train$Y-predict(rf, f\_train))^2)/length(f\_train$Y)  
test\_mse\_rf <- sum((f\_test$Y-p\_rf)^2)/length(p\_rf)  
  
train\_mse\_rf

## [1] 0.005588631

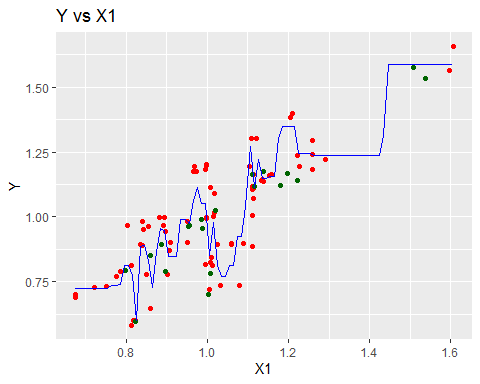
test\_mse\_rf

## [1] 0.01614003

## Висновок: значення середньоквадратичної похибки покращилися на навчальній вибірці – 0.005588631 (було 0.009771367), трошки погіршилися на тестовій вибірці – 0.01614003 (було 0.0137824). Модель перенавчено.

# Visualising

ggplot() +  
 geom\_point(aes(f\_train$X1, f\_train$Y),colour = 'red') +  
 geom\_point(aes(f\_test$X1, f\_test$Y),colour = 'dark green') +  
 geom\_line(aes(x\_grid, predict(rf, data.frame(X1 = x\_grid))),colour = 'blue') +  
 ggtitle('Y vs X1') +  
 xlab('X1') +  
 ylab('Y')

 ##Висновок: на графіку червоним позначені точки навчальної вибірки, зеленим – точки тестової вибірки, синім – модельні значення. #Saving results

fit <- read.csv2('indeks\_fit.csv', header = TRUE, encoding = 'UNICOD')  
fit$p\_dt <- p\_dt  
fit$p\_rf <- p\_rf  
head(fit)

## X p\_sr p\_mr p\_pr p\_dt p\_rf  
## 1 1 1.0018696 0.9860102 0.9975055 0.9307778 0.9910933  
## 2 2 1.5320579 1.5320162 1.5481294 1.3884286 1.5872500  
## 3 3 1.5030476 1.5107325 1.5186496 1.3884286 1.5872500  
## 4 4 1.2159456 1.1488504 1.2187034 1.3884286 1.2449100  
## 5 5 0.9818625 0.9048637 0.9774612 0.9307778 1.0513100  
## 6 6 0.8528166 0.7167507 0.8522472 0.8140000 0.7291400

write.csv2(fit[-1], file = "indeks\_fit.csv")

## Висновок:збережено результати