Project 2 - phase 3

#### ***Magda Młynarczyk, Dariusz Matlak, Zuzanna Kasprowicz***

* 1 Intorduction and preparing dataset
* 2 In searching for the best model
  + 2.1 Linear model
  + 2.2 Lasso
  + 2.3 Generalized Boosted Regression Model
  + 2.4 Random forest
  + 2.5 Support vector machine - SVM
  + 2.6 k-Nearest Neighbors - kNN
  + 2.7 The Besag York and Molli ́e model - BYM

**1** Intorduction and preparing dataset

Our goal, in third phase, was to create the best model that can predict number of cancer cases in future - we tried different approaches. The dataset, which we used, contains information about number of cancer cases for different regions, genders and age groups. To our analysis we chose only **breast cancer**. We also decided to work with additional factors such that:

1. Gaseous pollution.
2. Dust pollution.
3. Salaries.

We used data from BANK DANYCH LOKALNYCH. Gaseous and dust pollution data contains information about the number of different pollution in each counties. Salaries is a data which contain average salary for each county.

First, we prepared our three additionaly data sets to make them useful during prediction. Data about air pollution was normalized - divided by area of each county.

What is more, using information about number of population in each age group and each counties we normalized new cases of breast cancer.

Finally we got data: *pred*, which we used to perform prediction:

head(pred)

## X year TERYT wynagrodzenia dwutlenek.siarki tlenki.azotu tlenek.węgla  
## 1 1 2010 1001 3958.4 75.94525 43.26446 9.994835  
## 2 2 2010 1001 3958.4 75.94525 43.26446 9.994835  
## 3 3 2010 1001 3958.4 75.94525 43.26446 9.994835  
## 4 4 2010 1001 3958.4 75.94525 43.26446 9.994835  
## 5 5 2010 1001 3958.4 75.94525 43.26446 9.994835  
## 6 6 2010 1001 3958.4 75.94525 43.26446 9.994835  
## pył age sex newcases  
## 1 1.97 <75-84> M 0.000000000  
## 2 1.97 <0-44> M 0.000000000  
## 3 1.97 85+ K 0.000000000  
## 4 1.97 <45-54> K 0.001803905  
## 5 1.97 <65-74> K 0.002056404  
## 6 1.97 <75-84> K 0.001091703

**2** In searching for the best model

In this section we were going to find the best model which can predict the number of new diseases cases.

Firstly we divided our data set into two: training and testing set. For training set we chose \(75 \%\) of all observations - the rest for testing set.

samp <- createDataPartition(pred$newcases, 1, 0.75)  
  
train <- pred[samp[[1]],]  
test <- pred[-samp[[1]],]

**2.1** Linear model

Let us try some *linear model*!

To fit an ordinary *linear model* with new-cases as the response and salaries, gaseous pollution, dust pollution, age-group and sex as predictors, we tried lm function:

lm\_Mod <- lm(newcases~., data = train[, c(4:11)])

Next, we performed prediction and calculate **MSE** (mean square error) - average of the squares of the errors, that is, the difference between the estimator and what is estimated:

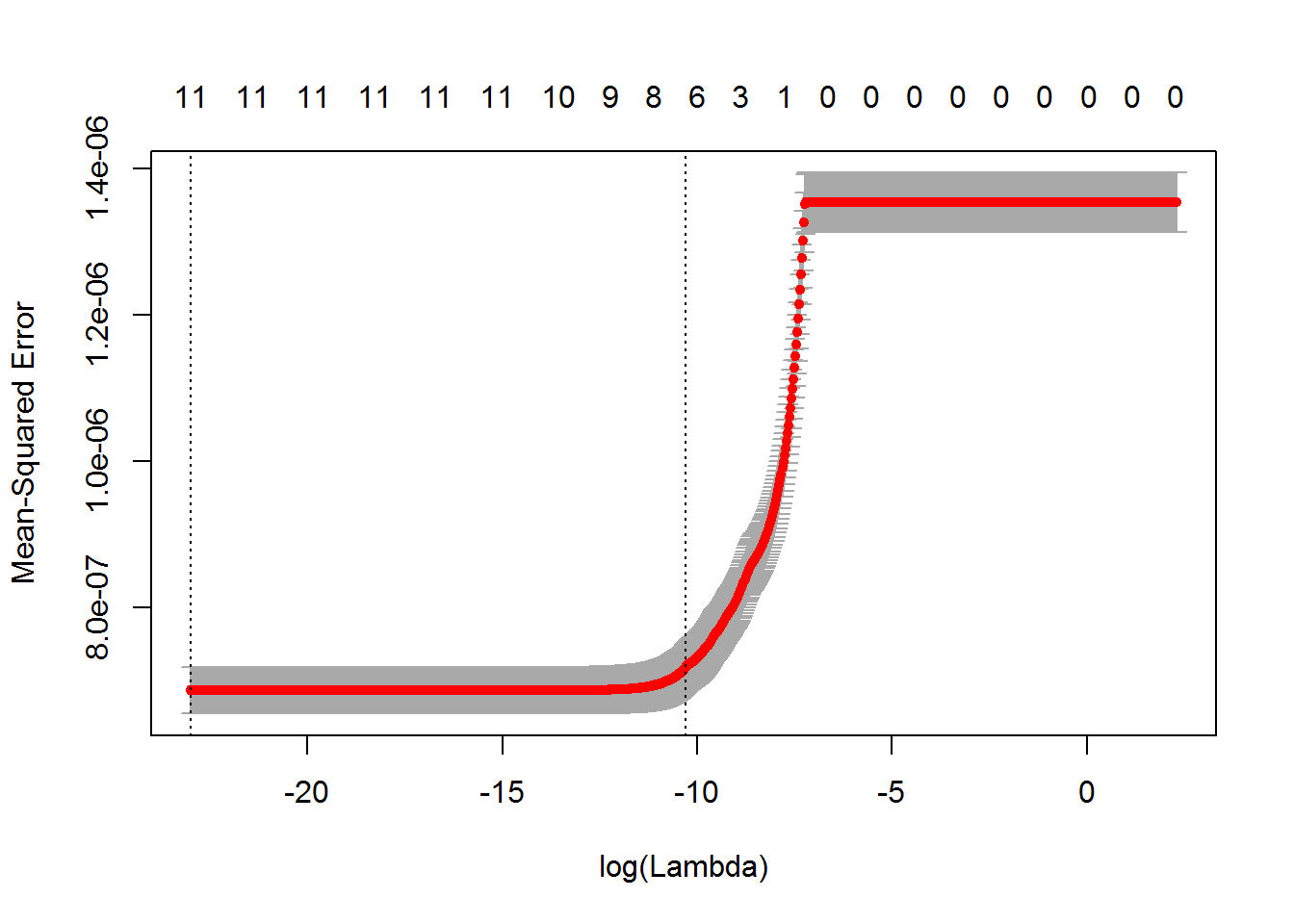
lm\_Pred <- predict(lm\_Mod, test[, c(4:10)])  
mean((lm\_Pred - test$newcases)^2)

## [1] 6.818084e-07

**2.2** Lasso

Next model is the *lasso*. We used cross-validation to set the parameter - lambda. Let’s see the plot which includes the cross-validation curve (red dotted line), and upper and lower standard deviation curves along the \(\lambda\) sequence (error bars). Two selected \(\lambda\)’s are indicated by the vertical dotted lines:

grid.lasso <- 10^seq(1,-10,length = 1000)  
x.lasso <- model.matrix(newcases~., train[, c(4:11)])  
y.lasso <- train$newcases  
newx.lasso <- model.matrix(newcases~., test[, c(4:11)])  
  
lasso\_Mod <- cv.glmnet(x.lasso,y.lasso, alpha = 1, type.measure = "mse", lambda = grid.lasso)  
#lasso\_Mod$lambda.min  
#coef(lasso\_Mod, s = "lambda.min")  
plot(lasso\_Mod)



We can view the selected \(\lambda\)’s, named lambda.min: 1.0256779\times 10^{-10}. *Lambda.min* is the value of \(\lambda\) that gives minimum mean cross-validated error.

lasso\_Pred <- predict(lasso\_Mod, newx.lasso, s = "lambda.min")  
lasso\_err <- mean((lasso\_Pred-test$newcases)^2)

For that \(\lambda\) we performed prediction and calculate the **MSE**, which is: 6.8180839\times 10^{-7}.

**2.3** Generalized Boosted Regression Model

Another model is GBM - Generalized Boosted Model.

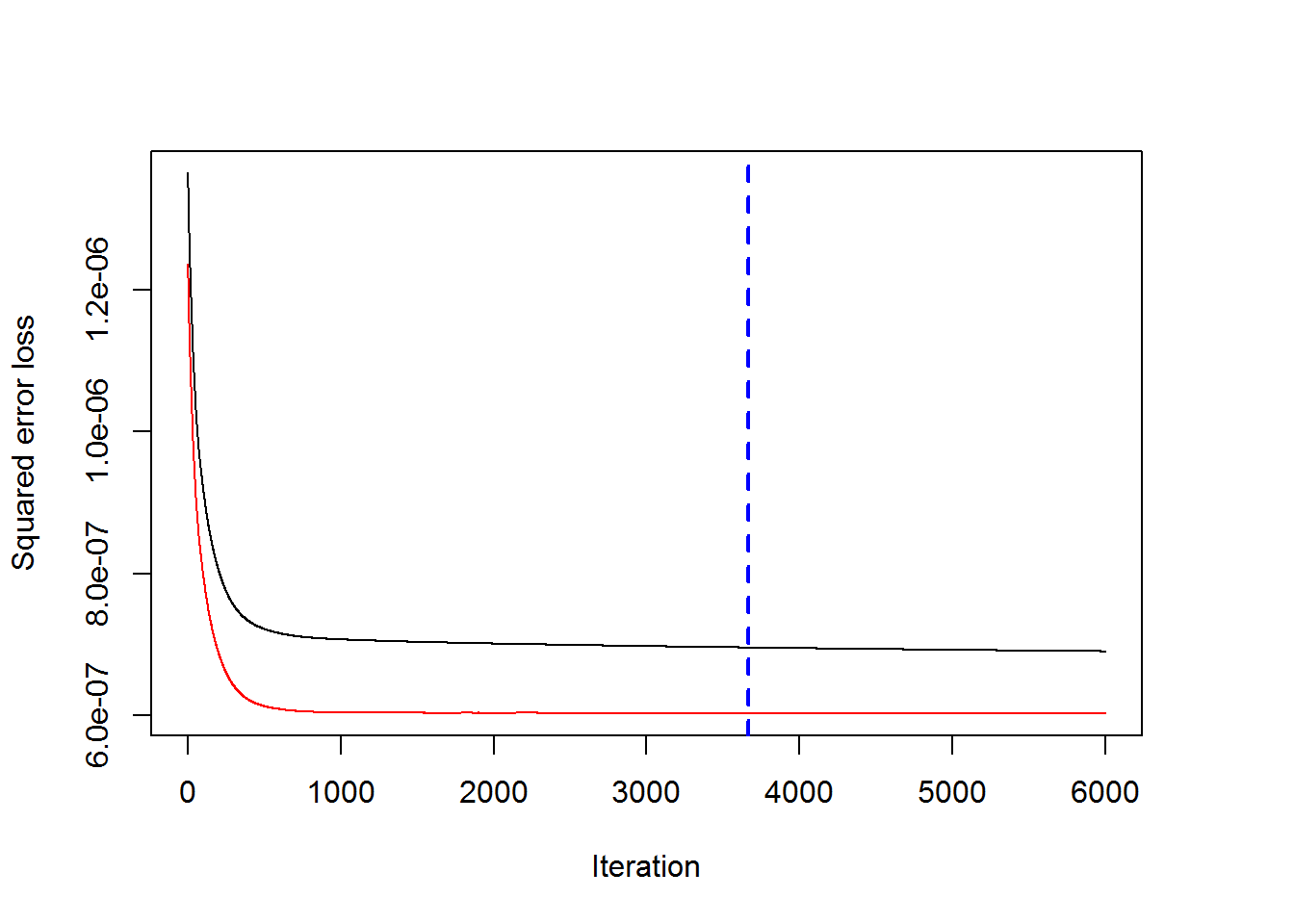
First we fit the model using gbm function. Next used gbm.perf(...,method="test") to obtain an estimate of the optimal number of iterations using the held out test set and optionally plotted various performance measures. We got:

gbm.perf(gbm\_Mod)

## Using test method...

## [1] 3663

gbm.perf(gbm\_Mod, method = "test")

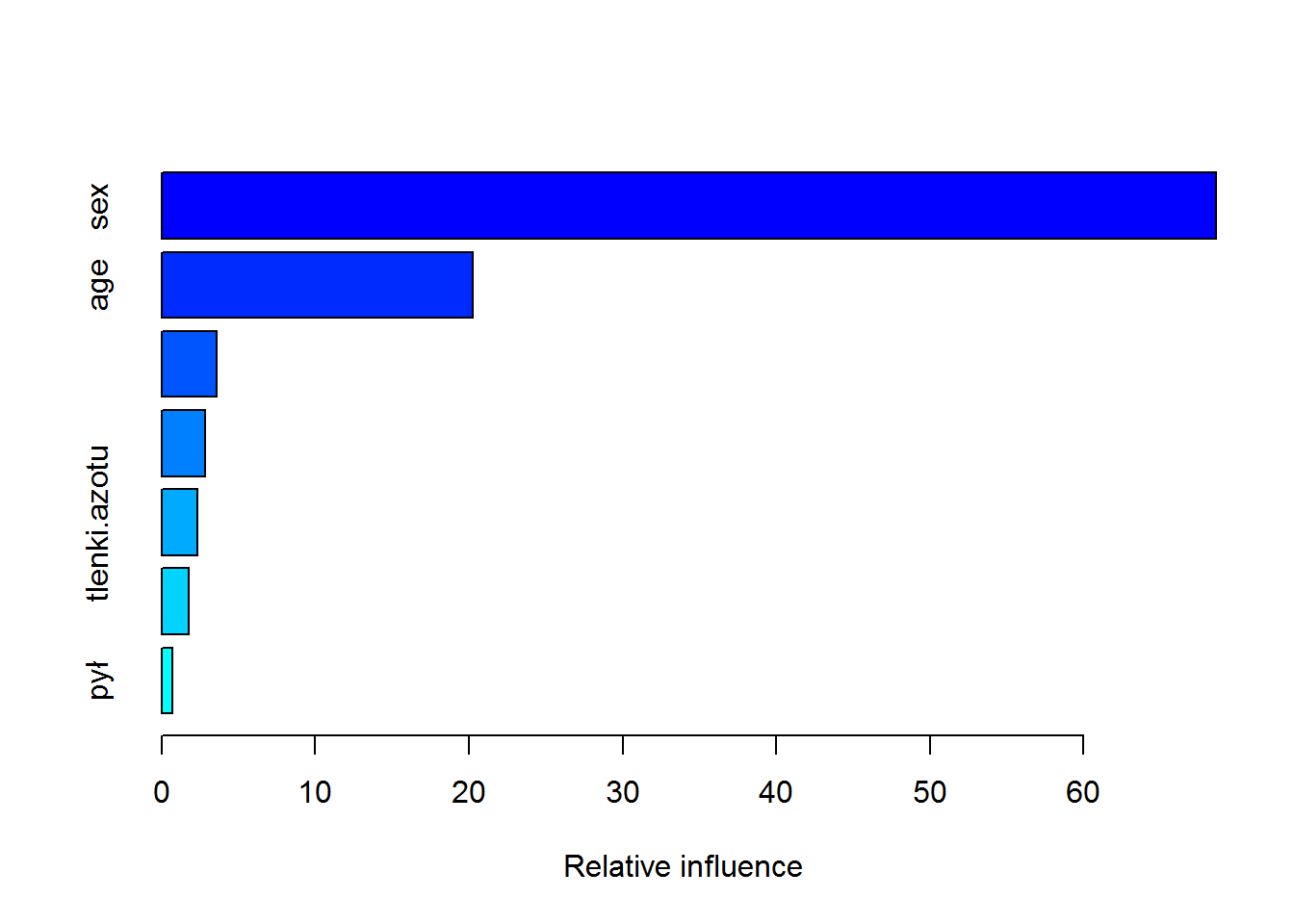


## [1] 3663

m\_gbm <- mean(gbm\_Mod$valid.error)

What is important we also calculated **MSE** of this model: 6.134245\times 10^{-7}. At the end of this model, we can see the plot which shows relative influence of each predictors:

#pretty.gbm.tree(gbm\_Mod, i.tree = 1)  
#print(gbm\_Mod)  
summary(gbm\_Mod)



## var rel.inf  
## sex sex 68.6138600  
## age age 20.2179482  
## wynagrodzenia wynagrodzenia 3.6098641  
## dwutlenek.siarki dwutlenek.siarki 2.7961314  
## tlenki.azotu tlenki.azotu 2.3332246  
## tlenek.węgla tlenek.węgla 1.7541635  
## pył pył 0.6748082

We can observe that the biggest infuence has sex and the lowest dust pollution.

**2.4** Random forest

Next method which we used for prediction is Random forest. randomForest implements Breiman’s random forest algorithm for classification and regression. As before, salaries, gaseous pollution, dust pollution, age-group and sex were used as predictors and new-cases as the response.

random.forests\_Test <- random.forests\_Mod$test  
  
pred\_rf <- random.forests\_Mod$predicted  
pred\_rf <- as.data.frame(pred\_rf)  
  
mrf <- mean(random.forests\_Test$mse)  
  
rsq <- mean(random.forests\_Test$rsq)

The **MSE** is: 6.0395191\times 10^{-8}.

**2.5** Support vector machine - SVM

Alternate method to perform prediction is also SVM. Let’s see it:

svm <- svm(newcases~., train[,c(4:11)], kernel = "linear", cross = 5)  
  
svm\_Pred <- predict(svm, test[, c(4:11)])  
m\_svm <- mean((svm\_Pred - test$newcases)^2)

And the **MSE** for this model is: 7.5047967\times 10^{-7}

**2.6** k-Nearest Neighbors - kNN

We used also for prediction - kNN.

knnData$newcases <- 1000\*knnData$newcases  
knnTrain <- knnData[knnData$year %in% c(2010,2011), ]  
knnTest <- knnData[knnData$year == 2012, ]  
knnData <- split(knnData, knnData$TERYT4)  
knnData <- lapply(knnData, function(x) c(x$year[1], x$TERYT4[1], x$X[1], x$Y[1], mean(x$newcases)))  
knnData <- as.matrix(as.data.frame(knnData))  
knnData <- as.data.frame(t(knnData))  
knnData <- knnData[-1]  
colnames(knnData) <- c("TERYT4", "X", "Y", "newcases")  
knnTest <- knnTest[, c(3,4,5,8)]  
  
knnMSE <- numeric(20)  
for (i in 1:20){   
knnModel <- knnreg(x = knnData[!is.na(knnData[, 4]), 2:3], y=knnData[!is.na(knnData[, 4]), 4], k=i)  
preds2 <- predict(knnModel, knnTest[, 2:3])  
names(preds2) <- knnData$TERYT4  
knnMSE[i] <- mean((knnTest$newcases-preds2)^2)  
}  
  
preds2 <- preds2[order(names(preds2))]  
preds2 <- round(preds2, digits = 3)  
#counties@data$preds2 <- preds2

**2.7** The Besag York and Molli ́e model - BYM

The final model which will be demonstrated is the Besag, York, and Molli ́e (1991) model (BYM), useful for modelling disease case counts in polygons.

library(raster)  
library(diseasemapping)  
library(INLA)  
counties <- shapefile("powiaty.shp")  
colnames(counties@data)[5] <- "TERYT"  
counties@data$TERYT <- as.integer(as.character(counties@data$TERYT))  
predTrain <- pred[pred$year %in% c(2010,2011), ]  
predListTrain <- split(predTrain, predTrain$TERYT)  
predListTrain <- lapply(predListTrain, function(x) c(x$TERYT[1], mean(x$wynagrodzenia), mean(x$dwutlenek.siarki), mean(x$tlenki.azotu), mean(x$tlenek.węgla), mean(x$pył), mean(x$newcases)))  
predListTrain <- as.data.frame(t(as.matrix(as.data.frame(predListTrain))))  
colnames(predListTrain) <- c("TERYT", "wynagrodzenia", "dwutlenek.siarki", "tlenki.azotu", "tlenek.węgla", "pył", "newcases")  
counties@data <- merge(counties@data, predListTrain, by = "TERYT", all = T)  
predTest <- pred[pred$year == 2012, ]  
predListTest <- split(predTest, predTest$TERYT)  
predListTest <- lapply(predListTest, function(x) c(x$TERYT[1], mean(x$wynagrodzenia), mean(x$dwutlenek.siarki), mean(x$tlenki.azotu), mean(x$tlenek.węgla), mean(x$pył), mean(x$newcases)))  
predListTest <- as.data.frame(t(as.matrix(as.data.frame(predListTest))))  
colnames(predListTest) <- c("TERYT", "wynagrodzenia", "dwutlenek.siarki", "tlenki.azotu", "tlenek.węgla", "pył", "newcases")  
  
  
bymModel <- bym(newcases~wynagrodzenia+dwutlenek.siarki+tlenki.azotu+tlenek.węgla+pył, data = counties, priorCI = list(sdSpatial = c(0.1, 5), sdIndep = c(0.1, 5)))  
  
predListTest <- predListTest[, c("TERYT", "newcases")]  
colnames(predListTest)[2] <- "newcases2012"  
counties@data <- merge(counties@data, predListTest, by = "TERYT", all = T)  
mean((bymModel$inla$summary.random$region.indexS$mode-counties@data$newcases2012)^2, na.rm=T)

## [1] 2.270663e-06