Daugiamatė Statistinė Analizė Lab 2, 6 variantas

Vytautas Kraujalis

2021-11-30

Table of Contents

# 1 Required packages

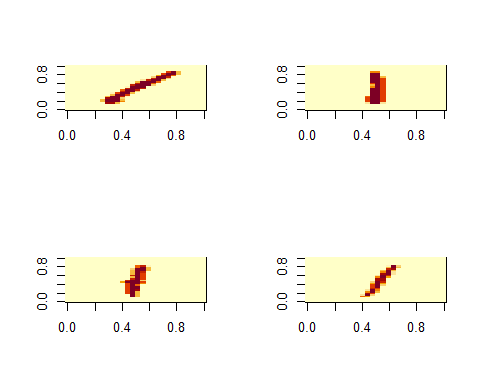
library(dplyr)  
library(factoextra)  
library(caret)  
library(glmnet)

# 2 Uzd. 1

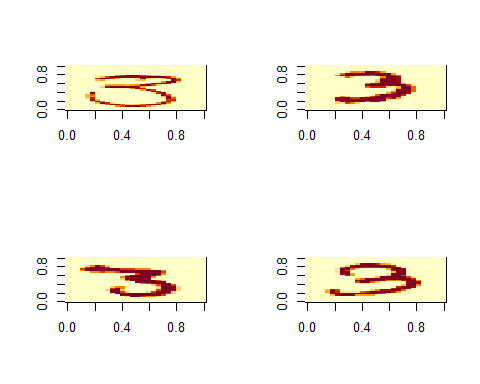
set.seed(123)  
  
data <- read.table("mnist.txt", sep = ",", header = T) %>%   
 filter(label %in% c(1, 3, 5, 7)) # 6 var.: 1357

# 3 Uzd. 2

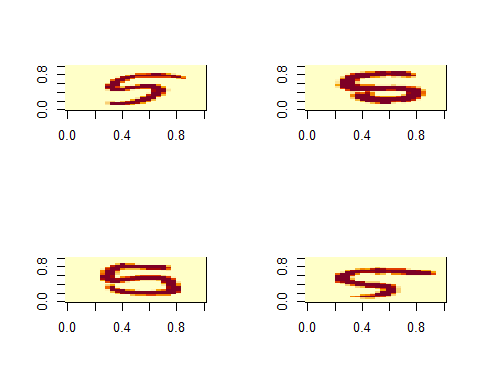
spausdinti\_grafika <- function(skaicius, kuri\_irasa){  
 image(t(matrix(data %>% filter(label == skaicius) %>% slice(kuri\_irasa) %>% select(-label) %>% t() %>% as.vector(), 28, byrow = T))[1:28, 28:1])  
}  
  
# vienetai:  
par(mfrow=c(2,2))  
spausdinti\_grafika(1, 1)  
spausdinti\_grafika(1, 2)  
spausdinti\_grafika(1, 3)  
spausdinti\_grafika(1, 4)



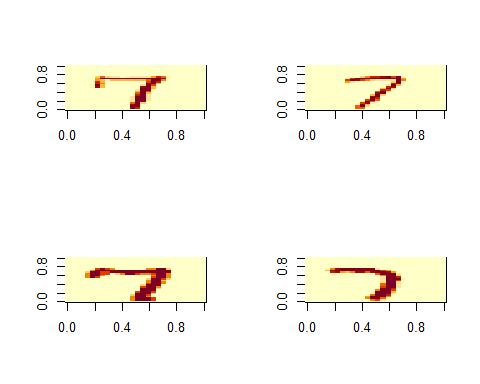
# trejetai:  
par(mfrow=c(2,2))  
spausdinti\_grafika(3, 1)  
spausdinti\_grafika(3, 2)  
spausdinti\_grafika(3, 3)  
spausdinti\_grafika(3, 4)



# penketai:  
par(mfrow=c(2,2))  
spausdinti\_grafika(5, 1)  
spausdinti\_grafika(5, 2)  
spausdinti\_grafika(5, 3)  
spausdinti\_grafika(5, 4)

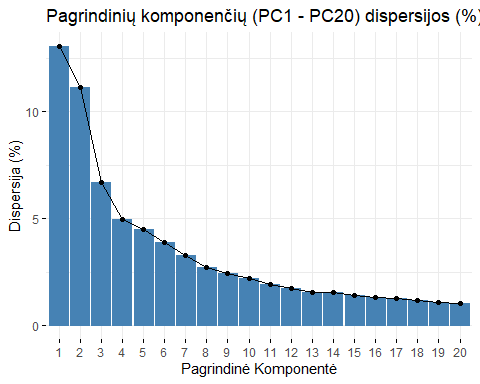


# septynetai:  
par(mfrow=c(2,2))  
spausdinti\_grafika(7, 1)  
spausdinti\_grafika(7, 2)  
spausdinti\_grafika(7, 3)  
spausdinti\_grafika(7, 4)



# 4 Uzd. 3

pca <- prcomp(data %>% select(-label), scale = FALSE)  
fviz\_eig(pca, ncp = 20, main = "Pagrindinių komponenčių (PC1 - PC20) dispersijos (%)", ylab = "Dispersija (%)", xlab = "Pagrindinė Komponentė")



# 5 Uzd. 4

pca\_var <- pca$sdev^2  
pca\_var <- pca\_var / sum(pca\_var)  
  
print(paste0("Pirmos 4 pagrindinės komponentės sumiškai savyje turi ", round(sum(pca\_var[1:4]) \* 100, digits = 2), "% dispersijos."))

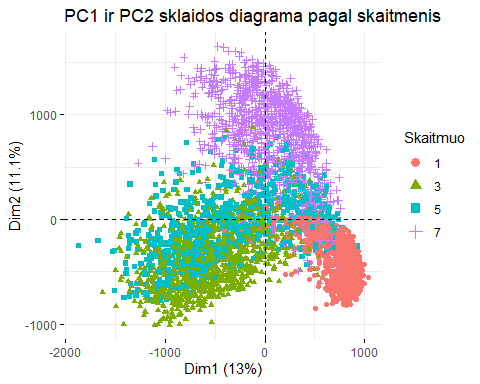
## [1] "Pirmos 4 pagrindinės komponentės sumiškai savyje turi 35.79% dispersijos."

dispersija <- 0  
kiek\_pca <- 0  
while(dispersija < 0.9){  
 kiek\_pca <- kiek\_pca + 1  
 dispersija <- sum(pca\_var[1:kiek\_pca])  
}  
  
print(paste0("Tam, kad pasiektume 90% suminės dispersijos, mums reikia panaudoti pirmas ", kiek\_pca, " pagrindines komponentes, su šiomis komponentinėmis, suminė dispersija sieka ", round(dispersija \* 100, digits = 2), "%"))

## [1] "Tam, kad pasiektume 90% suminės dispersijos, mums reikia panaudoti pirmas 78 pagrindines komponentes, su šiomis komponentinėmis, suminė dispersija sieka 90.08%"

# 6 Uzd. 5

skaitmenys <- as.factor(data$label)  
fviz\_pca\_ind(pca,  
 col.ind = skaitmenys, # color by groups  
 legend.title = "Skaitmuo",  
 label = "none",  
 title = "PC1 ir PC2 sklaidos diagrama pagal skaitmenis"  
 )



# 7 Uzd. 6

uzpildyti\_pixeliai <- data %>%   
 sapply(function(x){sum(x > 0)}) %>%   
 sort(decreasing = TRUE)  
  
uzpildyti\_pixeliai <- uzpildyti\_pixeliai[uzpildyti\_pixeliai / nrow(data) > 0.1]  
uzpildyti\_pixeliai <- uzpildyti\_pixeliai[-1]

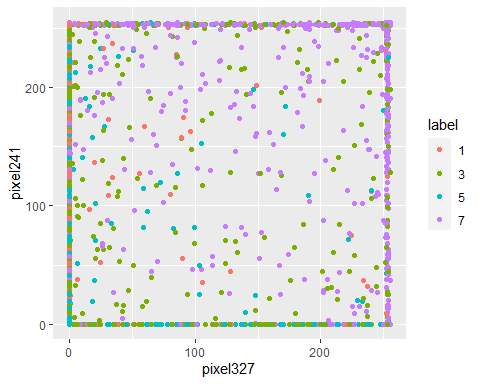
Kadangi yra stulpelių (pixelių), kurie jokiose eilutėse nebūna užpildyti, atfiltruosim tik tuos stulpelius, kurių užpildyti (reikšmė > 0) bent 10% iš stebėjimų. Iš gautų stulpelių atrinksime atsitiktines 3 poras ir atvaizduosime grafiškai.

Atrenkam 6 atsitiktinius stulpelius:

atsitiktines\_poros <- sample(names(uzpildyti\_pixeliai), 6, replace = F)

Atvaizduojame 3 atsitiktines poras:

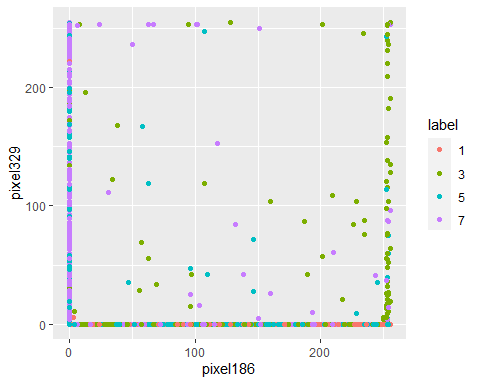
atsitiktiniai\_data <- data %>%   
 select(all\_of(atsitiktines\_poros), label) %>%   
 mutate(label = as.factor(label))  
atsitiktiniai\_data\_colnames <- colnames(atsitiktiniai\_data)  
colnames(atsitiktiniai\_data) <- c("V1", "V2", "V3", "V4", "V5", "V6", "label")  
  
atsitiktiniai\_data %>%   
 ggplot(aes(x = V1, y = V2, color = label)) +  
 geom\_point() +  
 xlab(atsitiktiniai\_data\_colnames[1]) +  
 ylab(atsitiktiniai\_data\_colnames[2])



atsitiktiniai\_data %>%   
 ggplot(aes(x = V3, y = V4, color = label)) +  
 geom\_point() +  
 xlab(atsitiktiniai\_data\_colnames[3]) +  
 ylab(atsitiktiniai\_data\_colnames[4])



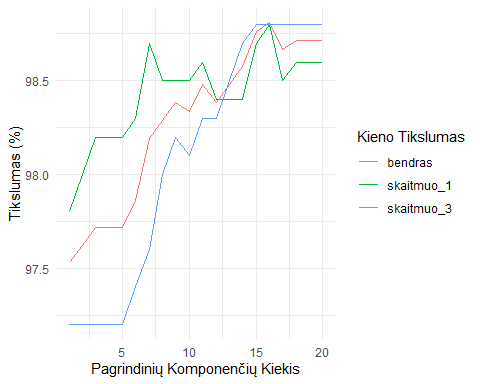
atsitiktiniai\_data %>%   
 ggplot(aes(x = V5, y = V6, color = label)) +  
 geom\_point() +  
 xlab(atsitiktiniai\_data\_colnames[5]) +  
 ylab(atsitiktiniai\_data\_colnames[6])



# 8 Uzd. 7

data <- data %>%   
 filter(label %in% c(1, 3)) %>%   
 mutate(label = as.factor(label))  
  
pca <- prcomp(data %>% select(-label), scale = FALSE)  
  
data\_pcs <- pca$x %>%   
 bind\_cols(label = data$label)  
  
# 2 klasių tikslumo ištraukimui iš sumaišymo matricos  
classAcc <- function(confusionMatrix) {  
 class1 <- round(confusionMatrix$table[1, 1] / sum(confusionMatrix$table[, 1]) \* 100, 1)  
 class2 <- round(confusionMatrix$table[2, 2] / sum(confusionMatrix$table[, 2]) \* 100, 1)  
 bendras <- confusionMatrix$overall["Accuracy"] \* 100  
 acc <- c(class1, class2, bendras)  
 names(acc) <- c(colnames(confusionMatrix$table), "Bendras")  
 return(acc)  
}  
  
# Susidarom list'ą su 20 formulių  
formules <- NULL  
formules[[1]] <- "label ~ PC1"  
for(i in 2:20){  
 formules[[i]] <- "label ~ PC1"  
 for(j in 2:i){  
 formules[[i]] <- paste0(formules[[i]], " + PC", j)  
 }  
}  
  
formules <- lapply(formules, as.formula)

tikslumai <- NULL  
for(i in 1:20){  
 glm <- glm(formula = formules[[i]], data = data\_pcs, family = "binomial")  
 prob\_pred = predict(glm, type = "response", newdata = data\_pcs %>% select(-label))  
 y\_pred = ifelse(prob\_pred > 0.5, 3, 1) %>%   
 as.factor()  
 confusion\_matrix <- confusionMatrix(y\_pred, data\_pcs$label)  
 tikslumai[[i]] <- classAcc(confusion\_matrix)  
}  
  
  
tikslumai\_data <- do.call(rbind.data.frame, tikslumai) %>%   
 bind\_cols(kiek\_pca = 1:20)  
colnames(tikslumai\_data) <- c("skaitmuo\_1", "skaitmuo\_3", "bendras", "kiek\_pca")  
  
  
tikslumai\_data %>%   
 tidyr::pivot\_longer(c(-kiek\_pca), names\_to = "Kieno Tikslumas", values\_to = "Tikslumas") %>%   
 ggplot(aes(x = kiek\_pca, y = Tikslumas, color = `Kieno Tikslumas`)) +  
 geom\_line() +  
 theme\_minimal() +  
 xlab("Pagrindinių Komponenčių Kiekis") +  
 ylab("Tikslumas (%)")



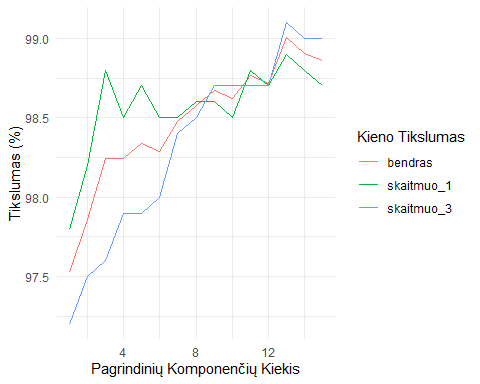
glm <- glm(label ~ ., data = data\_pcs[,c(1:20, 785)], family = "binomial")  
glm\_tuscias <- glm(label ~ 1, data = data\_pcs, family = "binomial")  
# Forward  
forward\_glm <- step(glm\_tuscias, direction = "forward", scope = formula(glm), trace=0)  
summary(forward\_glm)

##   
## Call:  
## glm(formula = label ~ PC1 + PC6 + PC7 + PC8 + PC16 + PC11 + PC13 +   
## PC15 + PC12 + PC20 + PC9 + PC5 + PC10 + PC4 + PC14, family = "binomial",   
## data = data\_pcs)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.5705 -0.0384 -0.0098 0.0058 1.9808   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.4421196 0.4398288 3.279 0.001042 \*\*   
## PC1 -0.0124006 0.0012118 -10.233 < 2e-16 \*\*\*  
## PC6 0.0050536 0.0009631 5.247 1.54e-07 \*\*\*  
## PC7 -0.0043165 0.0010998 -3.925 8.67e-05 \*\*\*  
## PC8 -0.0039978 0.0010385 -3.850 0.000118 \*\*\*  
## PC16 0.0068052 0.0016223 4.195 2.73e-05 \*\*\*  
## PC11 0.0037038 0.0010448 3.545 0.000392 \*\*\*  
## PC13 -0.0022650 0.0011576 -1.957 0.050400 .   
## PC15 -0.0034864 0.0013118 -2.658 0.007868 \*\*   
## PC12 0.0035853 0.0012632 2.838 0.004538 \*\*   
## PC20 0.0043398 0.0014496 2.994 0.002755 \*\*   
## PC9 0.0034400 0.0013232 2.600 0.009329 \*\*   
## PC5 -0.0018997 0.0007620 -2.493 0.012664 \*   
## PC10 0.0028300 0.0010981 2.577 0.009963 \*\*   
## PC4 -0.0011987 0.0006533 -1.835 0.066548 .   
## PC14 -0.0020457 0.0012475 -1.640 0.101038   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2913.25 on 2103 degrees of freedom  
## Residual deviance: 142.65 on 2088 degrees of freedom  
## AIC: 174.65  
##   
## Number of Fisher Scoring iterations: 10

forward\_glm$anova

## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 NA NA 2103 2913.2471 2915.2471  
## 2 + PC1 -1 2616.350401 2102 296.8967 300.8967  
## 3 + PC6 -1 45.647348 2101 251.2494 257.2494  
## 4 + PC7 -1 27.478649 2100 223.7707 231.7707  
## 5 + PC8 -1 14.408248 2099 209.3625 219.3625  
## 6 + PC16 -1 9.569141 2098 199.7934 211.7934  
## 7 + PC11 -1 10.733620 2097 189.0597 203.0597  
## 8 + PC13 -1 12.397102 2096 176.6626 192.6626  
## 9 + PC15 -1 9.337983 2095 167.3247 185.3247  
## 10 + PC12 -1 5.194648 2094 162.1300 182.1300  
## 11 + PC20 -1 3.116652 2093 159.0134 181.0134  
## 12 + PC9 -1 3.870922 2092 155.1424 179.1424  
## 13 + PC5 -1 3.400437 2091 151.7420 177.7420  
## 14 + PC10 -1 3.545717 2090 148.1963 176.1963  
## 15 + PC4 -1 2.788664 2089 145.4076 175.4076  
## 16 + PC14 -1 2.760514 2088 142.6471 174.6471

formules\_forward <- NULL  
formules\_forward[[1]] <- "label ~ PC1"  
for(i in 3:nrow(forward\_glm$anova)){  
 formules\_forward[[i-1]] <- paste0(formules\_forward[[i-2]], forward\_glm$anova[i,1])  
}  
  
formules\_forward <- lapply(formules\_forward, as.formula)  
  
  
tikslumai\_forward <- NULL  
for(i in 1:length(formules\_forward)){  
 glm <- glm(formula = formules\_forward[[i]], data = data\_pcs, family = "binomial")  
 prob\_pred = predict(glm, type = "response", newdata = data\_pcs %>% select(-label))  
 y\_pred = ifelse(prob\_pred > 0.5, 3, 1) %>%   
 as.factor()  
 confusion\_matrix <- confusionMatrix(y\_pred, data\_pcs$label)  
 tikslumai\_forward[[i]] <- classAcc(confusion\_matrix)  
}  
  
  
tikslumai\_forward\_data <- do.call(rbind.data.frame, tikslumai\_forward) %>%   
 bind\_cols(kiek\_pca = 1:length(formules\_forward))  
colnames(tikslumai\_forward\_data) <- c("skaitmuo\_1", "skaitmuo\_3", "bendras", "kiek\_pca")  
  
  
tikslumai\_forward\_data %>%   
 tidyr::pivot\_longer(c(-kiek\_pca), names\_to = "Kieno Tikslumas", values\_to = "Tikslumas") %>%   
 ggplot(aes(x = kiek\_pca, y = Tikslumas, color = `Kieno Tikslumas`)) +  
 geom\_line() +  
 theme\_minimal() +  
 xlab("Pagrindinių Komponenčių Kiekis") +  
 ylab("Tikslumas (%)")



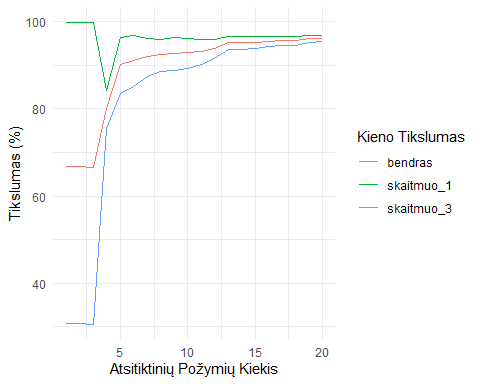
# 9 Uzd. 8

forward\_glm$anova$Step

## [1] "" "+ PC1" "+ PC6" "+ PC7" "+ PC8" "+ PC16" "+ PC11" "+ PC13"  
## [9] "+ PC15" "+ PC12" "+ PC20" "+ PC9" "+ PC5" "+ PC10" "+ PC4" "+ PC14"

# 10 Uzd. 9

atsitiktines\_originalios\_poros <- sample(names(uzpildyti\_pixeliai), 20, replace = F)  
  
formules\_originalios <- NULL  
formules\_originalios[[1]] <- paste0("label ~ ", atsitiktines\_originalios\_poros[1])  
for(i in 2:length(atsitiktines\_originalios\_poros)){  
 formules\_originalios[[i]] <- paste0(formules\_originalios[[i-1]], "+", atsitiktines\_originalios\_poros[i])  
}  
  
formules\_originalios <- lapply(formules\_originalios, as.formula)  
  
tikslumai\_originalios <- NULL  
for(i in 1:length(formules\_originalios)){  
 glm <- glm(formula = formules\_originalios[[i]], data = data, family = "binomial")  
 prob\_pred = predict(glm, type = "response", newdata = data %>% select(-label))  
 y\_pred = ifelse(prob\_pred > 0.5, 3, 1) %>%   
 as.factor()  
 confusion\_matrix <- confusionMatrix(y\_pred, data$label)  
 tikslumai\_originalios[[i]] <- classAcc(confusion\_matrix)  
}  
  
  
tikslumai\_originalios\_data <- do.call(rbind.data.frame, tikslumai\_originalios) %>%   
 bind\_cols(kiek\_pozymiu = 1:length(formules\_originalios))  
colnames(tikslumai\_originalios\_data) <- c("skaitmuo\_1", "skaitmuo\_3", "bendras", "kiek\_pozymiu")  
  
  
tikslumai\_originalios\_data %>%   
 tidyr::pivot\_longer(c(-kiek\_pozymiu), names\_to = "Kieno Tikslumas", values\_to = "Tikslumas") %>%   
 ggplot(aes(x = kiek\_pozymiu, y = Tikslumas, color = `Kieno Tikslumas`)) +  
 geom\_line() +  
 theme\_minimal() +  
 xlab("Atsitiktinių Požymių Kiekis") +  
 ylab("Tikslumas (%)")



# 11 Uzd. 10

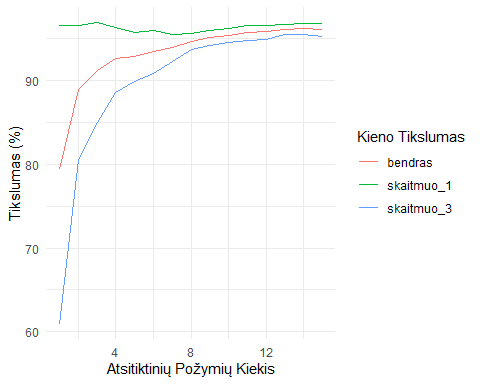
glm <- glm(label ~ ., data = data %>% select(all\_of(atsitiktines\_originalios\_poros), label), family = "binomial")  
glm\_tuscias <- glm(label ~ 1, data = data, family = "binomial")  
# Forward  
originalios\_forward\_glm <- step(glm\_tuscias, direction = "forward", scope = formula(glm), trace=0)  
summary(originalios\_forward\_glm)

##   
## Call:  
## glm(formula = label ~ pixel465 + pixel374 + pixel231 + pixel566 +   
## pixel576 + pixel294 + pixel379 + pixel153 + pixel653 + pixel581 +   
## pixel682 + pixel160 + pixel679 + pixel208 + pixel298, family = "binomial",   
## data = data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.6793 -0.1459 -0.0333 0.0265 2.7637   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.161338 0.393487 -2.951 0.003163 \*\*   
## pixel465 0.023228 0.002029 11.447 < 2e-16 \*\*\*  
## pixel374 0.019398 0.002347 8.265 < 2e-16 \*\*\*  
## pixel231 0.043406 0.007901 5.494 3.94e-08 \*\*\*  
## pixel566 0.016655 0.002737 6.086 1.16e-09 \*\*\*  
## pixel576 0.009425 0.001436 6.563 5.28e-11 \*\*\*  
## pixel294 -0.011864 0.001532 -7.743 9.70e-15 \*\*\*  
## pixel379 -0.013235 0.001507 -8.782 < 2e-16 \*\*\*  
## pixel153 0.009900 0.001412 7.013 2.33e-12 \*\*\*  
## pixel653 0.007573 0.001677 4.516 6.30e-06 \*\*\*  
## pixel581 0.022174 0.004296 5.161 2.45e-07 \*\*\*  
## pixel682 0.007399 0.001792 4.130 3.63e-05 \*\*\*  
## pixel160 -0.008603 0.002309 -3.726 0.000194 \*\*\*  
## pixel679 0.007149 0.002888 2.475 0.013328 \*   
## pixel208 -0.002863 0.001392 -2.057 0.039699 \*   
## pixel298 0.003095 0.001660 1.864 0.062299 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2913.2 on 2103 degrees of freedom  
## Residual deviance: 465.8 on 2088 degrees of freedom  
## AIC: 497.8  
##   
## Number of Fisher Scoring iterations: 9

originalios\_forward\_glm$anova

## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 NA NA 2103 2913.2471 2915.2471  
## 2 + pixel465 -1 951.811642 2102 1961.4355 1965.4355  
## 3 + pixel374 -1 665.141353 2101 1296.2942 1302.2942  
## 4 + pixel231 -1 215.045339 2100 1081.2488 1089.2488  
## 5 + pixel566 -1 154.920136 2099 926.3287 936.3287  
## 6 + pixel576 -1 85.484613 2098 840.8441 852.8441  
## 7 + pixel294 -1 88.227055 2097 752.6170 766.6170  
## 8 + pixel379 -1 89.211270 2096 663.4057 679.4057  
## 9 + pixel153 -1 60.410358 2095 602.9954 620.9954  
## 10 + pixel653 -1 46.547777 2094 556.4476 576.4476  
## 11 + pixel581 -1 42.865065 2093 513.5825 535.5825  
## 12 + pixel682 -1 21.356897 2092 492.2256 516.2256  
## 13 + pixel160 -1 11.887309 2091 480.3383 506.3383  
## 14 + pixel679 -1 6.478036 2090 473.8603 501.8603  
## 15 + pixel208 -1 4.547572 2089 469.3127 499.3127  
## 16 + pixel298 -1 3.513351 2088 465.7994 497.7994

formules\_originalios\_forward <- NULL  
formules\_originalios\_forward[[1]] <- paste0("label ~ ", originalios\_forward\_glm$anova[2,1])  
for(i in 3:nrow(originalios\_forward\_glm$anova)){  
 formules\_originalios\_forward[[i-1]] <- paste0(formules\_originalios\_forward[[i-2]], originalios\_forward\_glm$anova[i,1])  
}  
  
formules\_originalios\_forward <- lapply(formules\_originalios\_forward, as.formula)  
  
  
tikslumai\_originalios\_forward <- NULL  
for(i in 1:length(formules\_originalios\_forward)){  
 glm <- glm(formula = formules\_originalios\_forward[[i]], data = data, family = "binomial")  
 prob\_pred = predict(glm, type = "response", newdata = data %>% select(-label))  
 y\_pred = ifelse(prob\_pred > 0.5, 3, 1) %>%   
 as.factor()  
 confusion\_matrix <- confusionMatrix(y\_pred, data$label)  
 tikslumai\_originalios\_forward[[i]] <- classAcc(confusion\_matrix)  
}  
  
  
tikslumai\_originalios\_forward\_data <- do.call(rbind.data.frame, tikslumai\_originalios\_forward) %>%   
 bind\_cols(kiek\_pozymiu = 1:length(formules\_originalios\_forward))  
colnames(tikslumai\_originalios\_forward\_data) <- c("skaitmuo\_1", "skaitmuo\_3", "bendras", "kiek\_pozymiu")  
  
  
tikslumai\_originalios\_forward\_data %>%   
 tidyr::pivot\_longer(c(-kiek\_pozymiu), names\_to = "Kieno Tikslumas", values\_to = "Tikslumas") %>%   
 ggplot(aes(x = kiek\_pozymiu, y = Tikslumas, color = `Kieno Tikslumas`)) +  
 geom\_line() +  
 theme\_minimal() +  
 xlab("Atsitiktinių Požymių Kiekis") +  
 ylab("Tikslumas (%)")



# 12 Uzd. 11

tikslumai\_data

## skaitmuo\_1 skaitmuo\_3 bendras kiek\_pca  
## 1 97.8 97.2 97.52852 1  
## 2 98.0 97.2 97.62357 2  
## 3 98.2 97.2 97.71863 3  
## 4 98.2 97.2 97.71863 4  
## 5 98.2 97.2 97.71863 5  
## 6 98.3 97.4 97.86122 6  
## 7 98.7 97.6 98.19392 7  
## 8 98.5 98.0 98.28897 8  
## 9 98.5 98.2 98.38403 9  
## 10 98.5 98.1 98.33650 10  
## 11 98.6 98.3 98.47909 11  
## 12 98.4 98.3 98.38403 12  
## 13 98.4 98.5 98.47909 13  
## 14 98.4 98.7 98.57414 14  
## 15 98.7 98.8 98.76426 15  
## 16 98.8 98.8 98.81179 16  
## 17 98.5 98.8 98.66920 17  
## 18 98.6 98.8 98.71673 18  
## 19 98.6 98.8 98.71673 19  
## 20 98.6 98.8 98.71673 20

tikslumai\_forward\_data

## skaitmuo\_1 skaitmuo\_3 bendras kiek\_pca  
## 1 97.8 97.2 97.52852 1  
## 2 98.2 97.5 97.86122 2  
## 3 98.8 97.6 98.24144 3  
## 4 98.5 97.9 98.24144 4  
## 5 98.7 97.9 98.33650 5  
## 6 98.5 98.0 98.28897 6  
## 7 98.5 98.4 98.47909 7  
## 8 98.6 98.5 98.57414 8  
## 9 98.6 98.7 98.66920 9  
## 10 98.5 98.7 98.62167 10  
## 11 98.8 98.7 98.76426 11  
## 12 98.7 98.7 98.71673 12  
## 13 98.9 99.1 99.00190 13  
## 14 98.8 99.0 98.90684 14  
## 15 98.7 99.0 98.85932 15

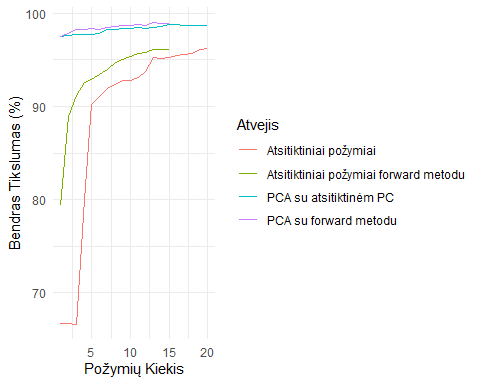
tikslumai\_originalios\_data

## skaitmuo\_1 skaitmuo\_3 bendras kiek\_pozymiu  
## 1 99.8 30.7 66.68251 1  
## 2 99.8 30.8 66.73004 2  
## 3 99.8 30.6 66.63498 3  
## 4 84.1 75.6 80.03802 4  
## 5 96.4 83.4 90.20913 5  
## 6 96.7 85.0 91.11217 6  
## 7 96.1 87.4 91.92015 7  
## 8 95.9 88.5 92.34791 8  
## 9 96.4 88.7 92.72814 9  
## 10 96.2 89.2 92.82319 10  
## 11 95.9 90.1 93.10837 11  
## 12 95.8 91.6 93.77376 12  
## 13 96.6 93.7 95.19962 13  
## 14 96.6 93.6 95.15209 14  
## 15 96.5 93.9 95.24715 15  
## 16 96.6 94.3 95.48479 16  
## 17 96.6 94.4 95.57985 17  
## 18 96.6 94.6 95.67490 18  
## 19 96.9 95.2 96.10266 19  
## 20 96.9 95.4 96.19772 20

tikslumai\_originalios\_forward\_data

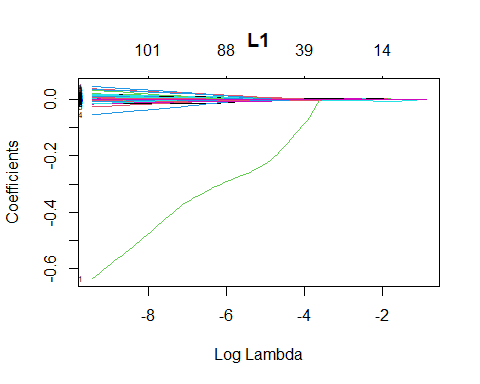
## skaitmuo\_1 skaitmuo\_3 bendras kiek\_pozymiu  
## 1 96.5 60.9 79.42015 1  
## 2 96.5 80.6 88.87833 2  
## 3 96.9 85.0 91.20722 3  
## 4 96.3 88.6 92.58555 4  
## 5 95.7 89.8 92.87072 5  
## 6 95.9 90.8 93.44106 6  
## 7 95.5 92.2 93.91635 7  
## 8 95.6 93.7 94.67681 8  
## 9 95.9 94.2 95.05703 9  
## 10 96.2 94.5 95.38973 10  
## 11 96.5 94.7 95.67490 11  
## 12 96.5 94.9 95.76996 12  
## 13 96.6 95.4 96.05513 13  
## 14 96.8 95.4 96.15019 14  
## 15 96.8 95.2 96.05513 15

tikslumai\_visi <- data.frame(  
 Bendras\_Tikslumas = c(tikslumai\_data$bendras, tikslumai\_forward\_data$bendras, tikslumai\_originalios\_data$bendras, tikslumai\_originalios\_forward\_data$bendras),  
 Kiek\_Pozymiu = c(tikslumai\_data$kiek\_pca, tikslumai\_forward\_data$kiek\_pca, tikslumai\_originalios\_data$kiek\_pozymiu, tikslumai\_originalios\_forward\_data$kiek\_pozymiu),  
 Atvejis = c(rep("PCA su atsitiktinėm PC", nrow(tikslumai\_data)), rep("PCA su forward metodu", nrow(tikslumai\_forward\_data)), rep("Atsitiktiniai požymiai", nrow(tikslumai\_originalios\_data)), rep("Atsitiktiniai požymiai forward metodu", nrow(tikslumai\_originalios\_forward\_data)))  
)  
  
tikslumai\_visi %>%   
 ggplot(aes(x = Kiek\_Pozymiu, y = Bendras\_Tikslumas, color = Atvejis)) +  
 geom\_line() +  
 theme\_minimal() +  
 xlab("Požymių Kiekis") +  
 ylab("Bendras Tikslumas (%)")

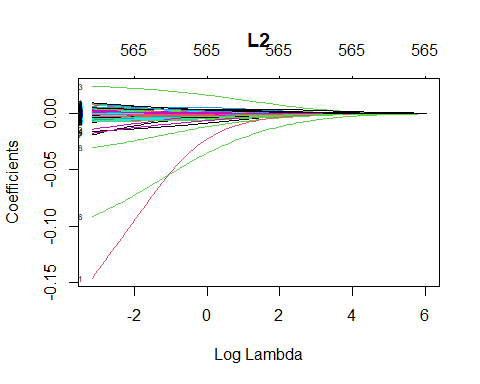


# 13 Uzd. 12

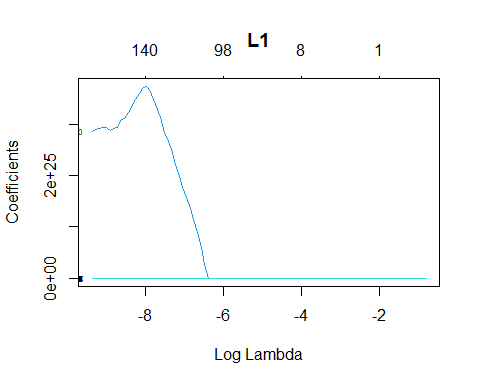
lasso\_L1 <- glmnet(data %>% select(-label), data$label, family = "binomial", alpha = 1)  
ridge\_L2 <- glmnet(data %>% select(-label), data$label, family = "binomial", alpha = 0)  
  
plot(lasso\_L1, label=TRUE,xvar="lambda", main="L1")



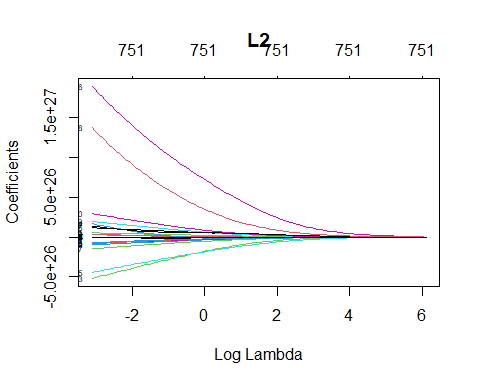
plot(ridge\_L2,label=TRUE,xvar="lambda", main="L2")



lasso\_L1\_pca <- glmnet(data\_pcs %>% select(-label), data\_pcs$label, family = "binomial", alpha = 1)  
ridge\_L2\_pca <- glmnet(data\_pcs %>% select(-label), data\_pcs$label, family = "binomial", alpha = 0)  
plot(lasso\_L1\_pca, label=TRUE,xvar="lambda", main="L1")



plot(ridge\_L2\_pca, label=TRUE,xvar="lambda", main="L2")



lambdas <- lasso\_L1$lambda  
  
lambda <- NULL  
  
for(i in 1:length(lambdas)){  
 lassoCoef = coef(lasso\_L1, s = lambdas[i])  
 nonzero\_coef = lassoCoef[lassoCoef[,1] != 0,]  
 if(length(nonzero\_coef) - 1 == 10){  
 lambda <- lambdas[i]  
 break  
 }  
}  
  
  
  
lassoCoef <- coef(lasso\_L1, s = lambda)  
lassoCoef <- lassoCoef[lassoCoef[,1] != 0,]  
lambda

## [1] 0.1498659

lassoCoef

## (Intercept) pixel179 pixel375 pixel409 pixel462   
## 6.670124e-01 8.914077e-04 8.535919e-04 7.285605e-04 -9.622494e-04   
## pixel489 pixel490 pixel494 pixel517 pixel522   
## -7.135796e-03 -7.732842e-04 4.888314e-04 -6.193543e-05 5.149765e-04   
## pixel550   
## 4.514332e-04

lambdas <- lasso\_L1\_pca$lambda  
  
lambda\_pca <- NULL  
  
for(i in 1:length(lambdas)){  
 lassoCoef = coef(lasso\_L1\_pca, s = lambdas[i])  
 nonzero\_coef = lassoCoef[lassoCoef[,1] != 0,]  
 if(length(nonzero\_coef) - 1 == 10){  
 lambda\_pca <- lambdas[i]  
 break  
 }  
}  
  
  
  
lassoCoef <- coef(lasso\_L1\_pca, s = lambda\_pca)  
lassoCoef <- lassoCoef[lassoCoef[,1] != 0,]  
lambda\_pca

## [1] 0.01575323

lassoCoef

## (Intercept) PC1 PC6 PC7 PC8   
## 4.543868e-01 -5.777272e-03 1.076897e-03 -9.677499e-05 -3.201329e-04   
## PC11 PC13 PC30 PC565 PC591   
## 4.236738e-04 -1.635797e-04 -1.236879e-04 -7.053515e+11 3.061386e+11   
## PC623   
## 1.065933e+11