Raport 3

Eksploracja danych

Mikołaj Langner, Marcin Kostrzewa
nr albumów: 255716, 255749

2021-04-19

Spis treści

1 Wstęp

2 Zadanie 1

2.1 Wczytanie danych i podział na zbiór uczący i testowy

Wczytajmy dane o irysach i podzielmy je na zbiór uczący i testowy w proporcji 1 : 2.

```
data(iris)
n <- dim(iris)[1]

train.set.index <- sample(1:n, 2/3*n)
train.set <- iris %>% slice(train.set.index) %>% arrange(Species)
test.set <- iris %>% slice(-train.set.index) %>% arrange(Species)
```

```
dummies <- dummyVars(" ~ .", data=iris)

train.dummies <- predict(dummies, newdata = train.set)
train.X <- as.matrix(cbind(rep(1, nrow(train.dummies)), train.dummies[, 1:4]))
train.Y <- train.dummies[, 5:7]
```

```
test.dummies <- predict(dummies, newdata = test.set)
test.X <- as.matrix(cbind(rep(1, nrow(test.dummies)), test.dummies[, 1:4]))
test.Y <- test.dummies[, 5:7]
```

```
Y.hat <- solve(t(train.X) %*% train.X) %*% t(train.X) %*% train.Y
```

```
train.proba <- train.X %*% Y.hat
train.prediction <- colnames(train.proba)[apply(train.proba, 1, which.max)]

test.proba <- test.X %*% Y.hat
test.prediction <- colnames(test.proba)[apply(test.proba, 1, which.max)]
```

```
train.confusion <- table(train.set$Species, train.prediction)
train.confusion
```

```
##             train.prediction
##              Species.setosa Species.versicolor Species.virginica
##   setosa                 33                  0                 0
##   versicolor              0                 21                12
##   virginica               0                  4                30
```

```
sum(diag(train.confusion)) / length(train.prediction)
```

```
## [1] 0.84
```

```
test.confusion <- table(test.set$Species, test.prediction)
test.confusion
```

```
##             test.prediction
##              Species.setosa Species.versicolor Species.virginica
##   setosa                 17                  0                 0
##   versicolor              0                 12                 5
##   virginica               0                  1                15
```

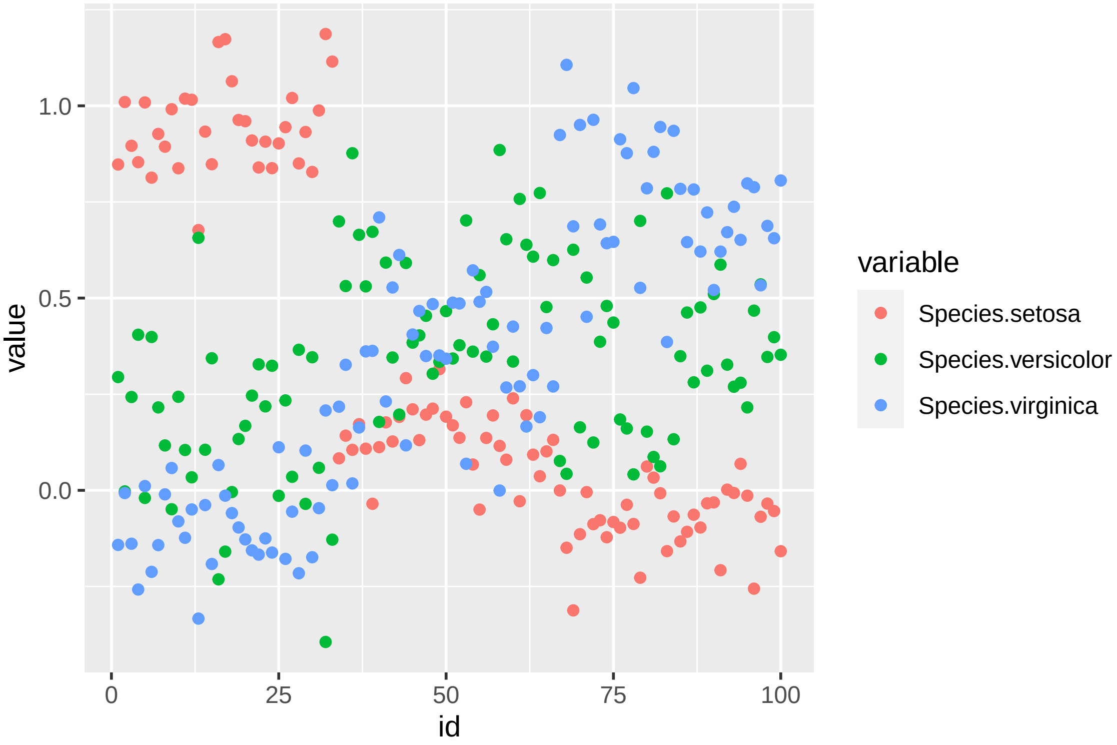
```
sum(diag(test.confusion)) / length(test.prediction)
```

```
## [1] 0.88
```

```
train.plot <- melt(as.data.frame(train.proba))
```

```
## No id variables; using all as measure variables
```

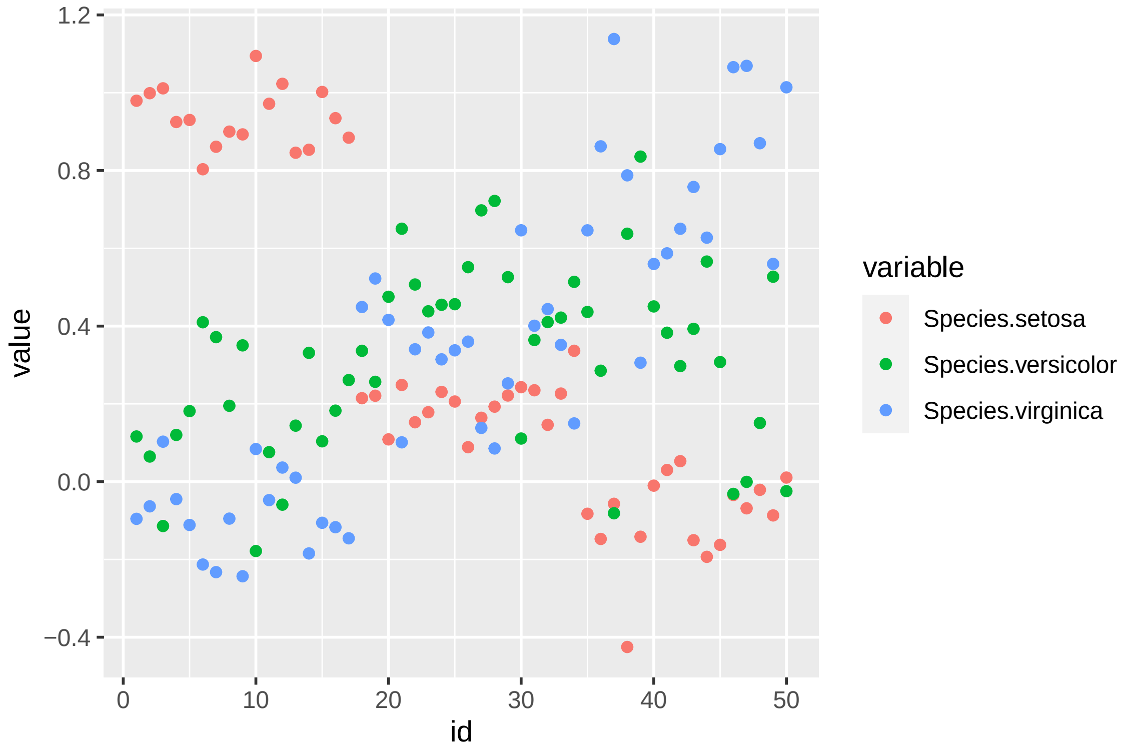
```
train.plot$id <- as.integer(rownames(train.proba))
ggplot(train.plot, aes(x=id, y=value, color=variable)) +
  geom_point()
```



```
test.plot <- melt(as.data.frame(test.proba))
```

```
## No id variables; using all as measure variables
```

```
test.plot$id <- as.integer(rownames(test.proba))
ggplot(test.plot, aes(x=id, y=value, color=variable)) +
  geom_point()
```



```
iris.quad <- (iris %>% select(-Species))^2
colnames(iris.quad) <- c("SL", "SW", "PL", "PW")
iris <- cbind(iris, combn(iris %>% select(-Species), 2, FUN = Reduce, f = `*`), iris.qu
```

```
train.set.index <- sample(1:n, 2/3*n)
train.set <- iris %>% slice(train.set.index) %>% arrange(Species)
test.set <- iris %>% slice(-train.set.index) %>% arrange(Species)
```

```
dummies <- dummyVars(" ~ .", data=iris)

train.dummies <- predict(dummies, newdata = train.set)
train.X <- as.matrix(cbind(rep(1, nrow(train.dummies)), train.dummies[, -c(5:7)]))
train.Y <- train.dummies[, 5:7]

test.dummies <- predict(dummies, newdata = test.set)
test.X <- as.matrix(cbind(rep(1, nrow(test.dummies)), test.dummies[, -c(5:7)]))
test.Y <- test.dummies[, 5:7]
```

```
Y.hat <- solve(t(train.X) %*% train.X) %*% t(train.X) %*% train.Y
```

```
train.proba <- train.X %*% Y.hat
train.prediction <- colnames(train.proba)[apply(train.proba, 1, which.max)]

test.proba <- test.X %*% Y.hat
test.prediction <- colnames(test.proba)[apply(test.proba, 1, which.max)]
```

```r
train.confusion <- table(train.set$Species, train.prediction)
train.confusion
```

```
##             train.prediction
##              Species.setosa Species.versicolor Species.virginica
##   setosa                 33                  0                 0
##   versicolor              0                 33                 0
##   virginica               0                  0                34
```

```r
sum(diag(train.confusion)) / length(train.prediction)
```

```
## [1] 1
```

```r
test.confusion <- table(test.set$Species, test.prediction)
test.confusion
```

```
##             test.prediction
##              Species.setosa Species.versicolor Species.virginica
##   setosa                 17                  0                 0
##   versicolor              0                 16                 1
##   virginica               0                  1                15
```

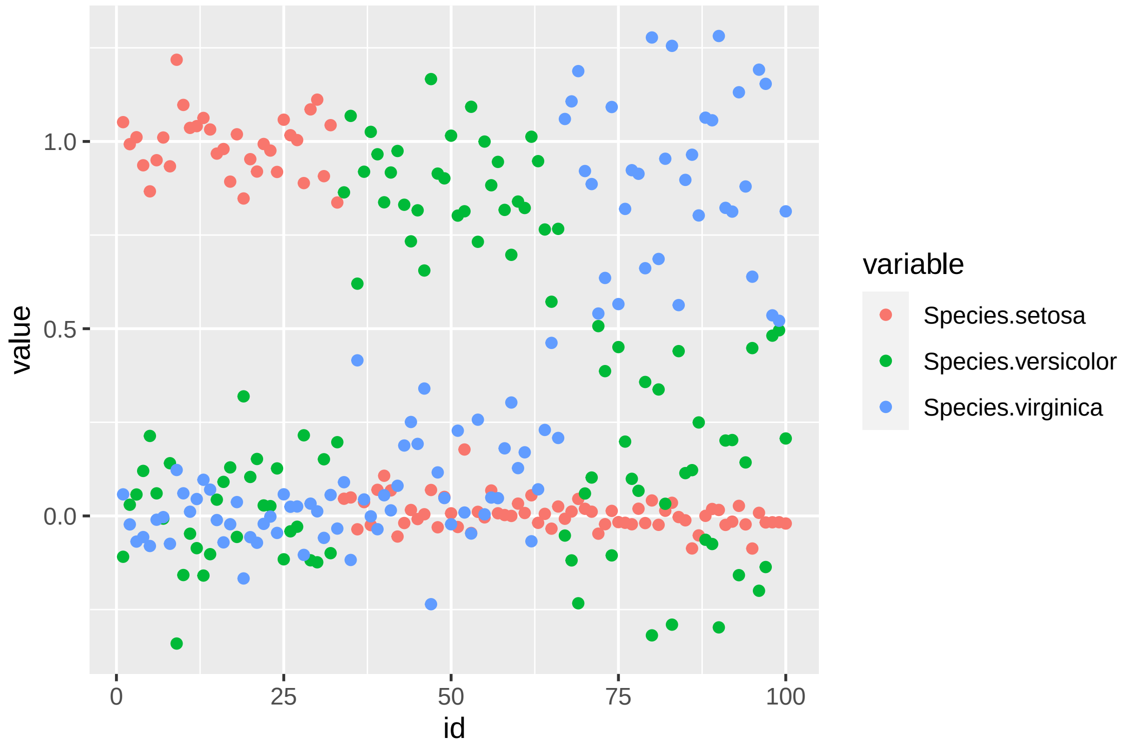
```r
sum(diag(test.confusion)) / length(test.prediction)
```

```
## [1] 0.96
```

```r
train.plot <- melt(as.data.frame(train.proba))
```

```
## No id variables; using all as measure variables
```

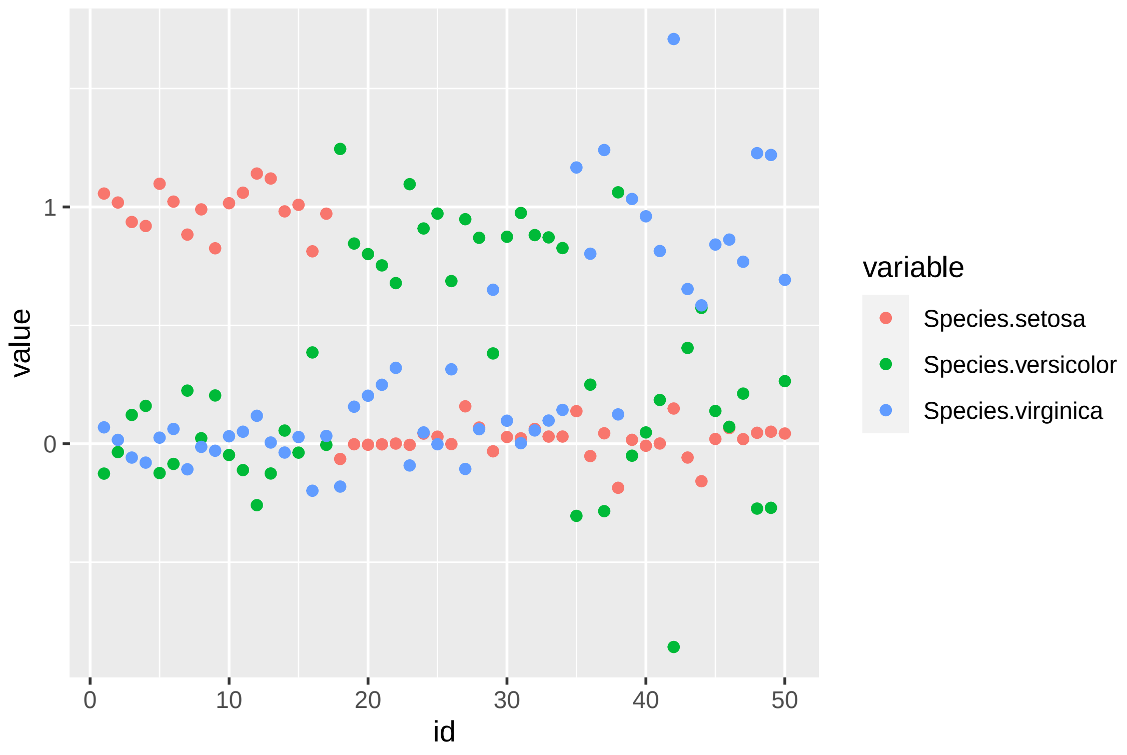
```r
train.plot$id <- as.integer(rownames(train.proba))
ggplot(train.plot, aes(x=id, y=value, color=variable)) +
  geom_point()
```



```
test.plot <- melt(as.data.frame(test.proba))
```

```
## No id variables; using all as measure variables
```

```
test.plot$id <- as.integer(rownames(test.proba))
ggplot(test.plot, aes(x=id, y=value, color=variable)) +
  geom_point()
```



3 Zadanie 2

```
data("BreastCancer")
n <- dim(BreastCancer)[1]
```

```
BreastCancer <- BreastCancer %>% select(-Id)
BreastCancer <- drop_na(BreastCancer)


for (column in colnames(BreastCancer)) {
  if (is.factor(BreastCancer[, column]) & column != "Class") {
    BreastCancer[, column] <- ordered(BreastCancer[, column])
  }
}
```

```
train.index <- sample(n, n/7)
train.data <- BreastCancer %>% slice(train.index)
test.data <- BreastCancer %>% slice(-train.index)
```

```
cv <- trainControl(method="cv", number=6)
```

```
model <- train(Class ~ ., data = train.data, method = "knn", trControl = cv)
confusion <- table(test.data$Class, predict(model, test.data))
confusion
```

```
##
##             benign malignant
##   benign       369         6
##   malignant     39       172
```

```
sum(diag(confusion)) / nrow(test.data)
```

```
## [1] 0.9232082
```

```
model <- train(Class ~ ., data = train.data, method = "rpart", trControl = cv)
confusion <- table(test.data$Class, predict(model, test.data))
confusion
```

```
##
##             benign malignant
##   benign       340        35
##   malignant     11       200
```

```
sum(diag(confusion)) / nrow(test.data)
```

```
## [1] 0.9215017
```

```
model <- train(Class ~ ., data = train.data, method = "naive_bayes", trControl = cv)
confusion <- table(test.data$Class, predict(model, test.data))
confusion
```

```
##
##             benign malignant
##   benign       326        49
##   malignant      1       210
```

```
sum(diag(confusion)) / nrow(test.data)
```

```
## [1] 0.9146758
```