Raport 3

Eksploracja danych

Mikołaj Langner, Marcin Kostrzewa
nr albumów: 255716, 255749

2021-04-19

Spis treści

1 Wstęp

Raport zawiera rozwiązania listy 3.

W zadaniu pierwszym budujemy klasyfikator na bazie metody regresji liniowej i oceniamy jego skuteczność i dokładność.

W zadaniu drugim ....Porównamy ze sobą rezultaty zastosowania:

- metoda k-najblizszych sasiadów (*k-Nearest Neighbors*),
- drzewa klasyfikacyjne (*classification trees*),
- naiwny klasyfikator bayesowski (*naïve Bayes classifier*).

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)
```

2.2 Konstrukcja klasyfikatora i wyznaczenie prognoz

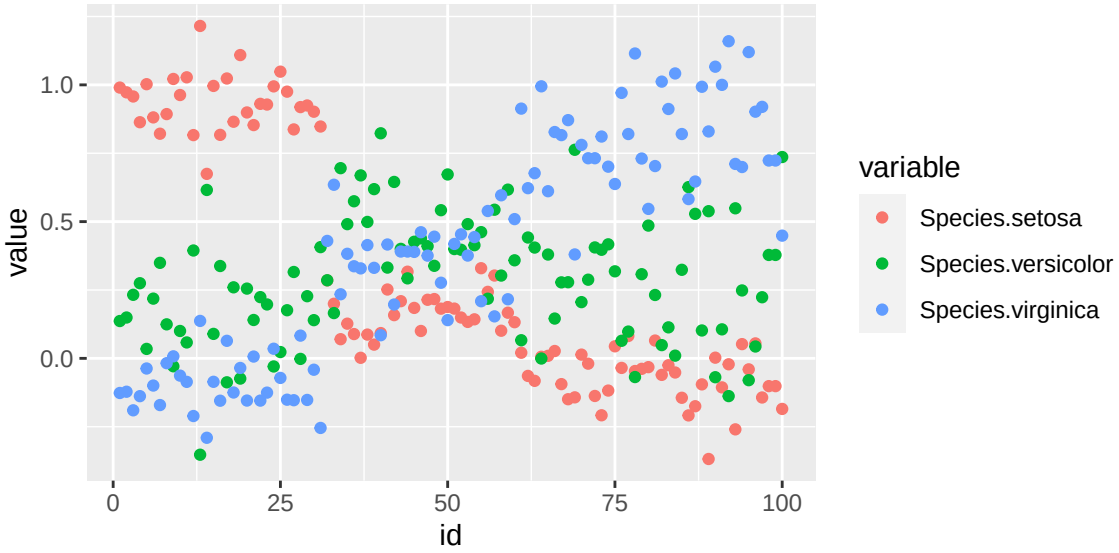
Stworzymy teraz macierze eksperymentu i wskaźnikową zarówno dla zbioru uczącego, jak i testowego. W tym celu wykorzystamy funckję `dummyVars` z pakietu Caret.

```
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]
```

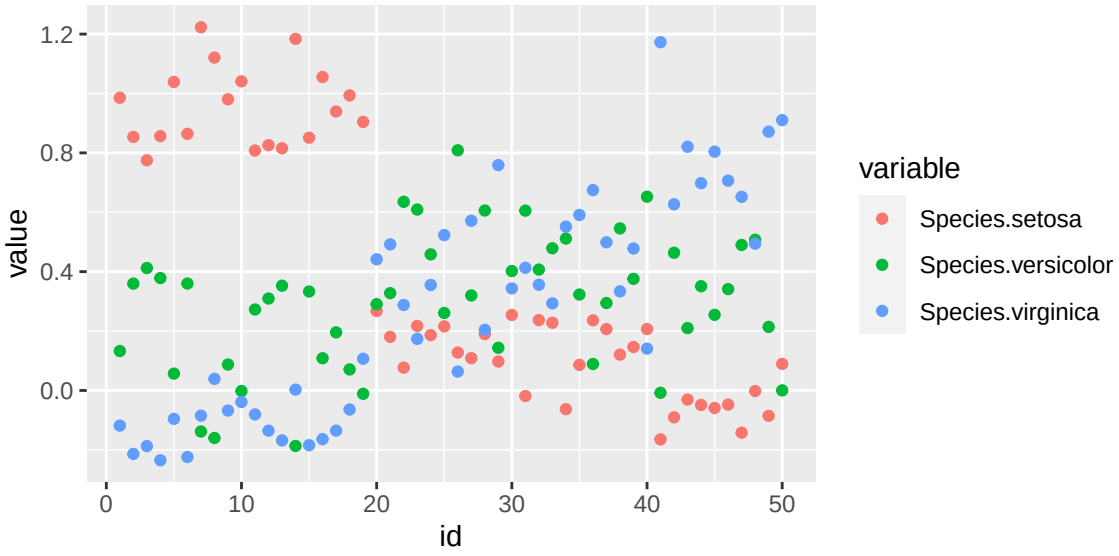
Wykorzystując metodę najmniejszych kwadratów, wyznaczamy przewidywane prognozy klas dla obu zbiorów.

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

train.proba <- train.X %*% Y.hat
test.proba <- test.X %*% Y.hat
```

Przedstawmy prognozy klas na wykresach.



Rysunek 1: Prognozy klas dla zbioru uczacego.



Rysunek 2: Prognozy klas dla zbioru testowego.

2.3 Ocena jakości klasyfikacji

Wyznaczmy teraz macierz pomyłek dla zbioru uczącego.

Tabela 1: Macierz pomylek dla zbioru uczacego.

	Species.setosa	Species.versicolor	Species.virginica
setosa	31	0	0
versicolor	0	17	12
virginica	0	3	37

Błąd klasyfikacji to 0.15.

Tabela 2: Macierz pomylek dla zbioru testowego.

	Species.setosa	Species.versicolor	Species.virginica
setosa	19	0	0
versicolor	0	11	10
virginica	0	1	9

Błąd klasyfikacji wynosi 0.22.

Wnioski i napomnienie o maskowaniu

2.4 Zastosowanie regresji liniowej do modelu o rozszerzonej ilości cech

Najpierw uzupełnijmy dane o irysach o składniki wielomianowe stopnia 2.

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

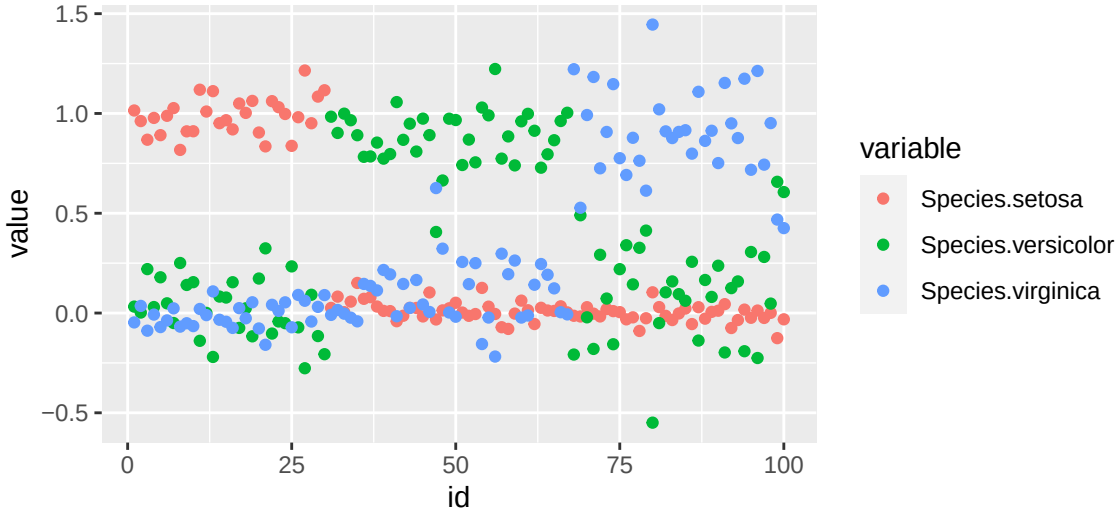
Podobnie jak poprzednio podzielimy dane na zbiory: uczący i testowy, a następnie utworzymy macierze: eksperymentu i indykatorów.

```
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]
```

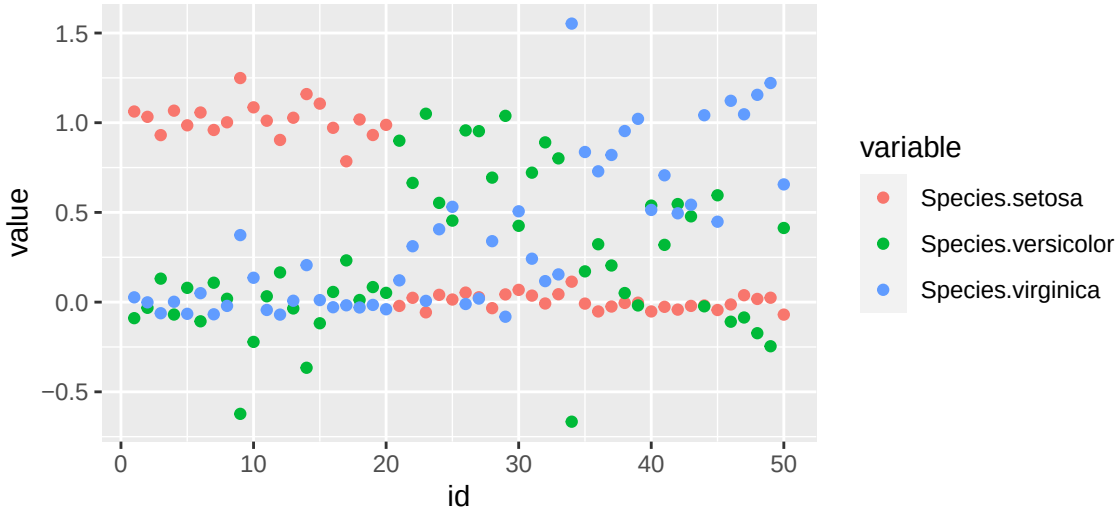
```
test.dummies <- predict(dummies, newdata = test.set)
test.X <- as.matrix(cbind(rep(1, nrow(test.dummies)), test.dummies[, -c(5:7)]))
test.Y <- test.dummies[, 5:7]
```

Ponownie, wyznaczymy prognozy klas i zwizualizujemy to przypisanie na wykresach.

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

train.proba <- train.X %*% Y.hat
test.proba <- test.X %*% Y.hat
```



Rysunek 3: Prognozy klas dla zbioru uczacego o rozszerzonej liczbie cech.



Rysunek 4: Prognozy klas dla zbioru uczacego o rozszerzonej liczbie cech.

Wyznaczymy także macierze pomyłek i błędy klasyfikacji.

Tabela 3: Macierz pomylek dla zbioru uczacego dla przypadku o rozszerzonej liczbie cech.

	Species.setosa	Species.versicolor	Species.virginica
setosa	30	0	0
versicolor	0	36	1
virginica	0	2	31

Błąd klasyfkacji wynosi 0.03.

Tabela 4: Macierz pomylek dla zbioru testowego dla przypadku o rozszerzonej liczbie cech.

	Species.setosa	Species.versicolor	Species.virginica
setosa	20	0	0
versicolor	0	11	2
virginica	0	3	14

Błąd klasyfikacji wynosi 0.1.

Wnioski i napomnienie o maskowaniu

3 Zadanie 2

3.1 Więcej treści na pierwszej stronie

3.2 Więcej treści na pierwszej stronie

3.3 Więcej treści na pierwszej stronie

3.4 Więcej treści na pierwszej stronie

```
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       370         6
##   malignant     40       171
```

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

```
## [1] 0.9216354
```

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

```
##
##             benign malignant
##   benign       359        17
##   malignant     31       180
```

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

```
## [1] 0.9182283
```

```
model <- train(Class ~ ., data = train.data, method = "naive_bayes", trControl = cv)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

```
confusion <- table(test.data$Class, predict(model, test.data))
confusion
```

```
##
##             benign malignant
##   benign       336        40
##   malignant      1       210
```

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

```
## [1] 0.9301533
```