Projekt1

Anton Lenartovich

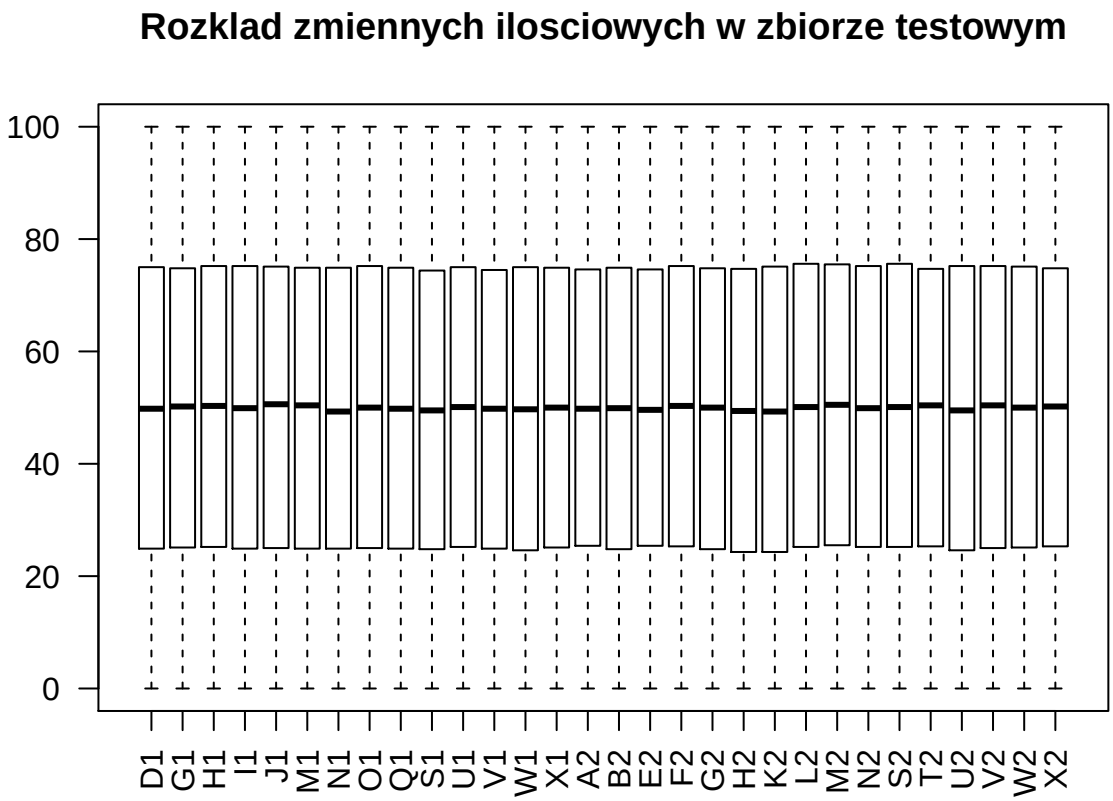
October 14, 2017

Wstępna analiza danych

Na etapie wstępnej analizy danych warto się przyjrzeć typom i wartościom danych oraz rozkładom poszczególnych zmiennych.

```
train <- read.csv("zbior_uczacy.txt", header=TRUE, sep=";")
train <- na.omit(train)
```

Zbiór składa się z 20 zmiennych typu *factor* oraz 30 zmiennych ilościowych. Zbiór wydaje się być w miarę zrównoważony dla zmiennych ilościowych i nominalnych.



Selekcja zmiennych

Dla porównania zostały wybrane dwie metody selekcji zmiennych: selekcja na podstawie kryterium BIC oraz selekcja przy pomocy lasów losowych. Spośród obu metod wybrana zostanie najlepsza.

Kryterium BIC

W pierwszym przypadku wybrane zmienne to:

$A1, B1, E1, L1, M1, P1, Q1, T1, W1, Y1, Z1, K2, N2, P2, T2, U2$.

```
m <- step(glm(y ~., train, family="binomial"), direction = "backward")
```

RandomForest

Żeby znaleźć najbardziej znaczące zmienne w drugim przypadku zostały użyte metody z pakietu *carot*. Na podstawienie wyników funkcji $varImp()$ zostaly wybrane zmienne, ktore mają największe "znaczenie":

$M1, W1, U2, T2, Q1, F2, J1, E1$.

```
rf <- randomForest(y ~ ., train)
```

Jak widać otrzymaliśmy dwa zbiory o różnych zmiennych. Wynikiem zastosowania kryterium oceny BIC jest wybór 16 zmiennych, w przypadku lasów losowych - 8 zmiennych.

```
features.rf = c("y", "M1", "W1", "U2", "T2", "Q1", "F2", "J1", "E1")
features.BIC = c("y", "A1", "B1", "E1", "L1", "M1", "P1", "Q1", "T1", "W1",
                 "Y1", "Z1", "K2", "N2", "P2", "T2", "U2")
```

Klasyfikacja

Zbiór treningowy został podzielony na 2 rozłączne zbiory: zbiór testowy (20% obserwacji) oraz zbiór treningowy(80% obserwacji).

```
train.ind <- sample(nrow(train), size = 0.8 * nrow(train), replace = FALSE)

trainSet <- train[train.ind,]
testSet <- train[-train.ind,]
```

glm

```
  model <- glm(y ~ ., trainSet, family="binomial")
  test$score <- predict(model, testSet[,-1], type="response")
```

```
##   features.rf features.BIC
## 1      0.6965       0.7294
```

randomForest

```
  model <- randomForest(y ~., data=trainSet)
  testSet$score <- predict(model, testSet[,-1], type="prob")[,2]
```

```
##   features.rf features.BIC
## 1      0.8541       0.8447
```

xgboost

```
model <- train(y ~., data=trainSet, method="xgbTree" )
class.test$score <- predict(model, testSet[,-1], type="prob")[,2]
```

```
##   features.rf features.BIC
## 1      0.8376         0.88
```

Podsumowanie

Porównując otrzymane wyniki widzimy, że najlepiej poradził sobie model *xgBoost*. Funkcja oceniająca poprawność klasyfikacji sprawdzała wynik dla 10% danych o największym współczynniku *score*. Warto również zauważyć, że w większości przypadków lepsze wyniki były otrzymywane dla modeli o zmiennych wybranych za pomocą kryterium BIC.

	rf	BIC
glm	0.6965	0.7294
randomForest	0.8541	0.8447
xgBoost	0.8376	0.88